MSC.Marc Mentat Help Reference

Version 2003
Copyright © 2003 MSC.Software Corporation

All rights reserved. Printed in U.S.A.

Corporate
MSC.Software Corporation
2 MacArthur Place
Santa Ana, CA 92707
Telephone: (714) 540-8900
Fax: (714) 784-4056

Europe
MSC.Software GmbH
Am Moosfeld
81829 München, GERMANY
Telephone: (49) (89) 431 987 0
Fax: (49) (89) 436 1716

Asia Pacific
MSC Japan Ltd.
Entsuji-Gadelius Building
2-39, Akasaka 5-chome
Minato-ku, Tokyo 107-0052, JAPAN
Telephone: (81) (3) 3505 0266
Fax: (81) (3) 3505 0914

Part Number: MA*V2003*Z*Z*Z*DC-REF

Disclaimer
THE CONCEPTS, METHODS, AND EXAMPLES PRESENTED IN THE DOCUMENTATION ARE FOR ILLUSTRATIVE AND
EDUCATIONAL PURPOSES ONLY, AND ARE NOT INTENDED TO BE EXHAUSTIVE OR TO APPLY TO ANY PARTICULAR
ENGINEERING PROBLEM OR DESIGN. USER ASSUMES ALL RISKS AND LIABILITY FOR RESULTS OBTAINED BY THE
USE OF THE COMPUTER PROGRAMS DESCRIBED HEREIN. IN NO EVENT SHALL MSC.SOFTWARE CORPORATION BE
LIABLE TO ANYONE FOR ANY SPECIAL, COLLATERAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES
ARISING OUT OF, RESULTING FROM, OR IN CONNECTION WITH USE OF THE CONTENTS OR INFORMATION IN THE
DOCUMENTATION.

MSC.SOFTWARE CORPORATION ASSUMES NO LIABILITY OR RESPONSIBILITY FOR ANY ERRORS THAT MAY
APPEAR IN THE DOCUMENTATION. THE DOCUMENTATION IS PROVIDED ON AN “AS-IS” BASIS AND ALL EXPRESS
AND IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT
SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

MSC.SOFTWARE CORPORATION RESERVES THE RIGHT TO MAKE CHANGES IN SPECIFICATIONS AND OTHER INFOR-
MATION CONTAINED IN THE DOCUMENTATION WITHOUT PRIOR NOTICE.

Trademarks
MSC, Dytran, MARC, and Patran are registered trademarks of MSC.Software Corporation or its subsidiaries in the United States and/or
other countries. MSC., MSC.Dytran, MSC.Marc, and MSC.Patran are trademarks of MSC.Software Corporation.

NASTRAN is a registered trademark of the National Aeronautics and Space Administration. MSC.Nastran is an enhanced proprietary
version developed and maintained by MSC.Software Corporation. All other trademarks are the property of their respective owners.

Third Party Software Program Credits
METIS is copyrighted by the regents of the University of Minnesota.
NT-MPICH is developed by Lehrstuhl für Betriebssysteme der RWTH Aachen. Copyright © 1992-2003 Lehrstuhl für
Betriebssysteme der RWTH Aachen.

Government Use
Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer
Software) and DFARS 227.7202 (Commercial Computer Software and Commercial Computer Software Documentation), as applicable.
Interfaces, 17
MSC.Marc Input File, 18
Plot Menu, 21
View Menu, 22
Static Commands, Continued, 23
List Commands, 25

A
Commands Begin with A
Mentat Help Commands in A, 1

B
Commands Begin with B
Mentat Help Commands in B, 75

C
Commands Begin with C
Mentat Help Commands in C, 83

D
Commands Begin with D
Mentat Help Commands in D, 193

E
Commands Begin with E
Mentat Help Commands in E, 231

F
Commands Begin with F
Mentat Help Commands in F, 267

G
Commands Begin with G
Mentat Help Commands in G, 281
<table>
<thead>
<tr>
<th>Commands Begin with</th>
<th>Mentat Help Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
</tr>
</tbody>
</table>

- Mentat Help Commands in H, 299
- Mentat Help Commands in I, 313
- Mentat Help Commands in J, 355
- Mentat Help Commands in K, 437
- Mentat Help Commands in L, 439
- Mentat Help Commands in M, 537
- Mentat Help Commands in N, 631
- Mentat Help Commands in O, 659
- Mentat Help Commands in P, 669
- Mentat Help Commands in Q, 727
<table>
<thead>
<tr>
<th></th>
<th>Commands Begin with</th>
<th>Mentat Help Commands in</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R, 729</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S, 823</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T, 999</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>U, 1041</td>
</tr>
<tr>
<td>V</td>
<td>V</td>
<td>V, 1075</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>W, 1087</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X, 1089</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y, 1116</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z, 1117</td>
</tr>
</tbody>
</table>
Appendix A

MSC.Marc Mentat
Arithmetic and Database Functions

- Arithmetic and Database Functions, 2
  - Arithmetic Functions, 2
  - Database Functions, 4
  - Post File Database Functions, 7
  - Set Functions, 8
CHAPTER 1

Mechanics of MSC.Marc Mentat

- MSC.Marc Mentat Window Layout
- How MSC.Marc Mentat Communicates with You
- How You Communicate with MSC.Marc Mentat
- Menu Structure
- List Specification
- Identifiers
Chapter Overview

This Help Reference manual will provide you with a description of the commands associated with the Mentat Product. The information is provided in a logical manner based upon the menu layout. This manual gives a description of the menu buttons. Associated with each button is a Mentat command. These commands are displayed in the scroll area, and may optionally be typed in. You should note that the commands may be executed at any point in the Mentat session.

There are fundamentally two types of commands:

– Those which set parameters or values such as entering the number of subdivisions.
– Those which result in an action such as actually subdividing the mesh or removing an entity.

In either case, the database associated with the problem is altered. In the case of defining a parameter, if an incorrect value is defined, simply enter a new value. In the case that an unanticipated result occurred due to the action, use the UNDO command to recover the previous state of the database.

This manual is referential in nature. The results of a command are graphically depicted in a few sections. The MSC.Marc User’s Guide should be used to learn how to be proficient with the product.

Upon completion of this chapter, you should have a clearer understanding of the following areas:

• The basic window layout
• How Mentat communicates with you
• How you communicate with Mentat
• The menu system
MSC.Marc Mentat Window Layout

The starting point for all communication with Mentat is the window shown in Figure 1-1 that appears at the start of the program.

![Basic Mentat Window](image)

Figure 1-1  Basic Mentat Window

The Mentat window that appears on your screen is divided into three major areas:

- Graphics
- Menu
- Dialogue

The **graphics** area is used to display the current state of the database. When you start Mentat, the graphics area is blank to indicate that the database is empty.

The **menu** area is reserved to show the selectable menu-items and is divided into two sub-areas, the static and dynamic menus. The contents of the dynamic menu area change as the menu-items are selected. In contrast, the static menu is always present and contains items that are applicable and selectable at all times.

The **dialogue** area is a scrollable area of about five visible lines where all program prompts, warnings, and responses appear, and where the user can input data or commands. Within the dialogue area is the status area which is reserved to communicate the state of the program to the user. Either **working** or **ready** appears in the status area to reflect the current state of the program.
How MSC.Marc Mentat Communicates with You

Mentat communicates with you via prompts and messages and other visual queues. Mentat’s prompts urge you to take action through the input of data or commands. These prompts have 3 types of trailing punctuation marks to indicate the required type of input:

: enter numeric data;

> enter a character string, typically a command, file name or set name;

? enter a YES or NO answer.

If you misspell a keyword or enter an incorrect response, Mentat warns you through a message posted in the dialogue area. Mentat does not require that you complete every action you initiate. For example, if you are prompted for a filename, and you change your mind, entering a \(<\text{CR}\>\) instead of typing in the filename will tell Mentat to abort the action. If the program is waiting for a list of items to operate on, and instead you enter a command that also requires a list of items or any additional data, Mentat will ignore your original request and process the command. If the command you enter does not request additional data, you are returned to the original data request from before the interrupt.

The program assumes at all times that you want to repeat the previous operation on a new set of items and will prompt you for a new list to operate on. This process repeats itself until you indicate otherwise, typically by entering a new command or a \(<\text{CR}\>\).
How You Communicate with MSC.Marc Mentat

All interaction with Mentat is done through the mouse, keyboard, or a combination of both. This section first discusses the usage of the mouse, followed by a discussion on how to use the keyboard as a means to enter commands and data.

The Mouse

The mouse is used to select items from the menu area or to point at items in the graphics area. It is important to make a distinction between using the mouse in the menu area versus the graphics area because the three mouse buttons have very different functions in each area. Figure 1-2 is a graphical representation of the mouse, mouse buttons, and corresponding cursor.

We have developed name conventions for each mouse button. The left button is represented by <ML>, the middle button by <MM>, and the right button by <MR>. The movements and location of the mouse are visually communicated to the user by the mouse cursor which is represented by a <^> in this guide.

*Depress* refers to the action of pressing a mouse button and keeping it depressed until we explicitly ask you to *release* it. *Click* refers to a quick single depress-release action.
Using the Mouse to Select a Menu Item

To select a menu item with the mouse, move the $\uparrow$ over the item that you want to select and click the $\text{ML}$. To return to the previous menu, move the $\uparrow$ over the menu area, and click the $\text{MR}$. Alternatively, you can click on the RETURN button in the menu area using $\text{ML}$. Clicking on the MAIN button takes you to the main menu.

On-line Help

An important companion for novice users in Mentat is the on-line help feature. Each menu item has a help panel with a short description and explanation of the function of that menu item. To activate the help feature, position the $\uparrow$ over the menu item on which you require help, followed by a click of the $\text{MM}$. The help panel disappears the moment you select another menu item.

Using the Mouse to Point

The mouse is used in two ways to operate in the graphics area: to point to, or pick, existing items as well as to point to, or pick, the location of yet to be created items.

1. The need to identify existing items displayed in the graphics area occurs frequently during a user’s interaction with the Mentat program. The mouse is used for this by moving the $\uparrow$ over the item to be identified followed by a click of the $\text{ML}$. In the remainder of the manual we describe this action by clicking on an item. If you inadvertently click on an item, you can undo that action by clicking the $\text{MM}$ anywhere in the graphics area.

![Figure 1-3 Using the Mouse in the Menu Area](image-url)
At times, you will need to identify more than a single item. A list of items must be terminated by a click of the \(<\text{MR}>\) with the \(<\uparrow>\) positioned anywhere in the graphics area. Alternatively, you can click on the END LIST button in the menu area using \(<\text{ML}>\).

2. In order to use the mouse for creating a new item you need to define the relation between the position on the graphics screen and its location in global coordinates. In Mentat, it is possible to define a grid that is positioned in space and where the grid consists of points that can be pointed to. If you click in the vicinity of a grid point, the coordinates of the item that you created will be snapped to that grid point. In addition, you can also pick an existing node, point, or surface-grid-point to specify a location.

![Figure 1-4 Using the Mouse in the Graphics Area](image-url)

**Keyboard Input**

Not all data can be entered through the mouse; **numerical** and **literal data** *must* be entered via the keyboard. The program mode prescribes the specific requirements for proper entry of each type of data. The program can be in **data mode** or in **command/literal data mode** and is described under the following two headings.

**Numerical Data**

You must use the keyboard for numerical data entry. The program interprets the data entry according to the context in which it is used. If the program expects a real number and you enter an integer, Mentat will automatically convert the number to its floating point value. Conversely, if a floating point format number is entered where an integer is expected, the program will convert the real number to an integer.
Scientific notation for real numbers is allowed in the following formats:

- 0.12345e-01
- -0.12345e-01

The interpreter does not allow imbedded blanks in the format. Whenever the program encounters an illegal format, the message `bad float!` will appear in the dialogue area. The prompt for numerical data is a colon (:).

**Literal Data**

Literal data is used for file, set and macro names. A literal data string may *not* be abbreviated. Commands as introduced in the beginning on page 4 are considered string data (as opposed to literal string data) and *can* be abbreviated as long as the character string is unique within the Mentat command library. For example, `*add_elements` cannot be abbreviated to `*add` because of the other commands that start with the same characters such as `*add_nodes` and `*add_curves`. The program checks the input for validity against the internal library of valid responses. For example, if you enter an ambiguous or misspelled command, Mentat responds by listing all the valid entries that start with the same first letter of the command. The prompt for literal data is a greater-than symbol (>).

If the program is in data mode which is identified by the : prompt, you must enter a command preceded by an * (asterisk) to instruct the program that you are entering a command.

For example: `Enter node (1): *add_nodes`

If you enter a command without the asterisk when the program is in data mode, Mentat responds with an error message in the dialogue area.

The asterisk *can* be omitted when the program is in command or literal data mode which is indicated by the greater-than symbol (>).

For example: `Command > add_nodes`

**Editing the Input Line**

The experienced user can enter a sequence of commands or requests in a single 160-column input line. Note that anything typed beyond the input line limit is lost! Use `<CR>` to avoid this. You must use a blank space to separate entries when you are entering multiple responses on a single input line. All entries in the buffer are processed sequentially.
Mentat maintains a history of lines that are entered and offers limited recall and editing capabilities for the command line. The arrow keys ∧ and ∨ on the keyboard can be used to scroll up and down in the dialogue area to make these lines visible. Use CTRL-p (that is, hold down the CTRL key and press the p key) to recall a previously entered input line. Repeat the CTRL-p sequence to recall as many lines as you need. Use CTRL-n to move to the next line in the history of command lines. (By the way, p and n stand for previous and next respectively in these control sequences.)

Edit functions for the current line are: backspace for character delete and CTRL-u for line delete. The left and right arrow keys are used to position the cursor at the desired location to overwrite or insert characters. The TAB key is used as a toggle to switch from insert to overwrite mode and vice versa. For example, if you type *view_viewpont 0.0 0.0 1.0, the program responds with the message unknown command in the dialogue area. To correct the entry, recall the line using CTRL-p, use the left arrow key to move the cursor to the letter n of view_viewpoint, press the TAB key, type i, and press <CR> to enter the line. The command will now be *view_viewpoint.
Menu Structure

This section focuses on the menu system as a means to communicate with Mentat. The first sub-section discusses the structure of menus that constitute the program. The second sub-section analyzes the components of each menu.

Menu System

The kernel of the Mentat program consists of a set of processors in a parallel configuration that operate on the database. The database is the most compact, yet complete, description of the current state of the model you are analyzing. Typical examples of processors are \texttt{SUBDIVIDE} and \texttt{PATH PLOT}.

Every processor may depend on a number of parameters that influence the process. The combined number of processors and parameters in Mentat is too large to show in one menu. To help you in the scheduling of tasks, we have structured menus around the processors that lead you through the steps from top down. Figure 1-1 shows you the organization of the main menu that appears when you start Mentat and how it corresponds to the main tasks of the analysis cycle depicted in Figure 1-1 of this chapter.

For your convenience, the menu items have been grouped in panels by the four main tasks: preprocessing, analysis, postprocessing, and configuration. The menu items and sub-tasks on each of these panels represent yet another group of corresponding tasks. It is important to realize that most of the menus for the global tasks do not contain processors; these menus are for navigation purposes only and are not part of the kernel of the program!
Figure 1-5  Organization of Main Menu

A task and corresponding sub-task is selected by clicking on a menu item of that menu. After the (sub-)task is accomplished, it is necessary to traverse the menus in the opposite direction. There are two ways to do this:

1. Click on the RETURN or MAIN menu items in the static menu area. RETURN takes you to the previous menu and MAIN takes you to the main menu.

2. Move the <↑> over the menu area and click the <MR>. The result of this sequence is equivalent to clicking on the RETURN menu item.
Buttons

Positioned on the panel are flat and raised rectangles. The raised rectangles in the released state suggest a light shining directly from above. The task is printed on the raised rectangle and is selectable by clicking on it with the <ML>. Flat rectangles are not selectable; they convey the setting of parameters. The program does not respond to clicking the <ML> or <MM> on the flat rectangles.

Mentat contains 5 types of raised rectangles. Throughout the remainder of this document we will use the term button for raised rectangle. Below is a list of the different types of buttons and their functions.

The Submenu Button
As mentioned before, this button represents a gateway to a submenu. It is recognized by a ( ) symbol on the right hand side of the button.

The Cycle Button
A cycle button ( ) is used to set a parameter to a value when there is a choice of three or more alternatives. The parameter is set to the value that is currently displayed on the button. Clicking on this button will change the displayed value to the next consecutive value in the list of alternatives. If the list is exhausted, the process will start over again with the first alternative. Note that the symbol is indicative of the unidirectional way the list of alternatives is traversed.

The Toggle Button
A special type of cycle button is the toggle button ( ) where the number of alternatives is limited to two. It is a switch that connotates a state of on or off; a button is depressed to flag on or active, and released (or raised) to flag that the listed parameter is off or inactive.

The Tabular Button
A tabular button represents a combination of a parameter button and a flat rectangle. They show one or more numerical or alpha-numerical values that are associated with the parameter represented by the button. Clicking on this button type usually implies that you have to enter data through the keyboard, which is then displayed in the rectangular fields after the keyboard input is completed.

Tabular buttons may contain a large number of numerical data fields. There are instances where the tabular buttons pop-up over the graphics area. If this is the case, you need to confirm that all entries have been completed by clicking on the OK button. Before returning to regular menu selection, you can clear all entries by clicking on the RESET button which usually appears in the lower left hand side of the panel. The
pop-up table then disappears from the graphics area and the original graphics area is restored. Typical examples of these compounded tabular buttons can be found in the boundary conditions and material properties menus.

**The One-Only Button Group**

The alternative values of cycle buttons are also represented as individual toggles under a one-only button group. In a cycle, only one value can be selected, hence if a button in a one-only group of buttons is depressed, another is released. The one-only button sequence is identified by a \( \Diamond \) symbol shown on each button of that sequence.

As a typical example of a menu, the Coordinate System panel of the Mesh Generation menu as shown in Figure 1-6 will be discussed. These buttons are also summarized in Figure 1-9.

![Coordinate System Panel](attachment:image1.png)

**Figure 1-6**  Coordinate System Panel

The GRID button is a toggle; it can be switched on or off. The default position for this button is the raised or released state which means that the grid is off. Clicking it will turn the grid on and leave the button in a depressed state.

![Toggle Button States](attachment:image2.png)

**Figure 1-7**  Released and Depressed States of a Toggle Button

The button next to it displays RECTANGULAR and has the \( \ns \) symbol which implies a cycle. In contrast to the toggle, a cycle button has more than two values. In this example the button is an adjective to grid and specifies the type of grid to be used. Again, in contrast to the toggle, this button will not stay depressed. A click on this button changes the value of the parameter displayed on the button. The default value, RECTANGULAR, is changed into CYLINDRICAL. Clicking on it again changes the value to SPHERICAL, to be followed by RECTANGULAR again if this is repeated.
For three items in the cycle list this is still a viable way of setting the value of a parameter. If there are more than three, it becomes a tedious task to cycle through the alternatives. Therefore you will often find a submenu button combined with a cycle button. For example, the SET button is a gateway to a submenu as can be seen by the ➤ symbol. By clicking on the SET button you are taken to a submenu where the cycle that describes the type of grid to be used is represented by a one-only group of buttons.

![Example of Simple Tabular Buttons](image)

**Figure 1-8** Example of Simple Tabular Buttons

*Figure 1-8* gives you examples of tabular buttons that are found in the SET submenu.
List Specification

*Menu Structure* on page 10 discussed the difference between menu buttons that are used to navigate through the menus and buttons that represent processors. Processors generally require two types of data:

- Parameters associated with the process
- A list of items to operate on.

If the list to operate on consists of only one item, you can use the mouse to point to that item on the graphics screen (see page 6 on pointing). If the list of items contains twenty items, pointing to each item individually becomes a cumbersome task and; if the list contains a hundred items, pointing becomes an impossible task. This section concentrates on the capabilities in Mentat to specify a list of items.

The Mentat program recognizes the following items:

- Points
- Curves
- Surfaces
- Solids
- Vertices of solids
- Edges of solids
- Faces of solids
- Nodes
- Elements
- Edges of elements
- Faces of elements.
A simple example of how to generate a list follows. You can then extrapolate from what you have learned in this section to do more intricate examples. Assume you want to subdivide an existing element that is already displayed in the graphics area of the Mentat window. The processor to use is **SUBDIVIDE**, and it operates on elements only.

![Figure 1-9 Locating the SUBDIVIDE Processor in the Mesh Generation Menu](image)

After you activate the **SUBDIVIDE** processor and click on the **ELEMENTS** button in the subdivide submenu, the following program prompt appears in the dialogue area:

```
Enter subdivide element list:
```
Chances are that you don’t know the element number (nor should you care at this point). For this reason, answering this question by typing a number in the dialogue area may be possible but is not necessarily a viable option. Instead, move the mouse over the graphics area and use the <ML> to click on the center of the element. You have now entered the first element in the list. The program keeps prompting you for more elements; if this is the only one you want to subdivide, you must let the program know that this is the end of the list. This can be done in one of three ways:

1. Press the END LIST button in the menu area,
2. Type a ‘#’ sign in the dialogue area, or
3. Click <MR> with <↑> anywhere over the graphics area.

The most convenient way of ending the list is of course to click <MR> since the <↑> is most likely already over the graphics area and saves you a keystroke from the keyboard.

**Using a Box to Specify a List**

Suppose the number of subdivisions was set to 20 by 20, creating 400 elements. Assume you want to enter the left 200 elements in a list by creating a rectangle to fence in those elements. Position the <↑> at one of the corners of the box. Depress the <ML> and move the <↑> to the opposite corner of the box you want to create. The rectangle that appears tells you exactly which elements are included in the box. Once you have reached the desired position, release <ML>. Every element that is completely inside the box is included in the list specification.
Step 1: Position the cursor

Step 2: Hold down the `<ML>` and drag the cursor to the desired position

Step 3: Release the `<ML>` button

Figure 1-10  Selecting an Element Using the Box Pick Method

There are times when you may need the guidance of cross hairs to help you determine what is to be included in your selection. To activate the cross hairs, press the SHIFT key while moving the `<↑>` in the graphics area.
Note: You can relax the completely inside constraint mentioned previously by using the PARTIAL button on the picking panel under DEVICE.

Using a Polygon to Specify a List (CTRL Key + <ML>)

An alternative to using the box pick for list specification is to use a polygon around the elements that you want to include in the list. As with the box pick method, only those elements that are completely inside the polygon are entered into the list. To use the polygon pick, move the <↑> to the first corner point of the polygon. Click <ML> while holding down the CTRL key on the keyboard. Move to the next vertex of the polygon and click <ML> again, continue to hold the CTRL key down. Repeat this process until a closed loop is formed. The last point needs to be in the vicinity of the starting point and must be clicked on to end the selection. A variation on this polygon pick is the lasso pick. This is done by holding down the CTRL key and the <ML> down simultaneously while slowly moving the mouse, until the elements to be selected are surrounded by the lasso. With either approach, a final click on <ML> is required at the position near the beginning of the polygon or lasso.

Note: The PARTIAL and COMPLETE buttons mentioned under the Box Pick Method also apply to the Polygon Pick Method.

Table 1-1 at the end of this chapter summarizes the mouse selection options in both the graphics and menu areas.

Press Ctrl key, and click <ML> at each vertex of the polygon

Figure 1-11 Selecting an Element Using the Polygon Pick Method
The LIST Buttons

For your ease of use, we have pre-programmed some of the more common list options and assigned them buttons which are located in the lower left hand side of the static menu.

The LIST buttons are:

- all: EXISTING
- all: SELECTED
- all: UNSELECTED
- all: VISIBLE
- all: INVISIBLE
- all: OUTLINE (all nodes and edges on the outline)
- all: SURFACE (all nodes, edges and faces on the outer surface)

Press Ctrl key, and hold <M> down while dragging the <↑> until a closed loop is formed around the elements to be selected.

**Figure 1-12** Selecting an Element Using the Lasso Pick Method

**All: EXISTING**

Perhaps the most used list button is all EXISTING. It specifies all existing elements, nodes, curves, points, or surfaces (whichever is applicable), to be operated on by the processor that requested the list.

The contents of the selected/unselected, visible/invisible list are determined by the two operators: SELECT and VISIBLE. The meaning of each and their connection is explained in the next paragraphs.
**All: SELECTED/UNSELECTED**

The **SELECT** operator is a very powerful way to separate specific items from others. The methods by which items are selected range from a single item to a path of nodes, a box of items, or all items on a plane, and are connected by Boolean operators such as *and*, *except*, *invert*, and *intersect*. An example of this syntax is:

```
(use) single [items] and (a) box (of) [items] except single [items]
```

where the words *use*, *a*, *of*, and *item* are implied because they do not appear as buttons. A powerful feature of the **SELECT** processor is the ability to name a group of items, and refer to them by that name in list specifications. The **STORE** command facilitates this process.

![Location of LIST Buttons](image)

**Figure 1-13** Location of LIST Buttons

**All: VISIBLE/INVISIBLE**

Sometimes the model may be so complex that it takes an unacceptably long time to update the graphics screen every time the database changes. It is advantageous to focus on the items that you are working on. By activating and deactivating items from the display list, you can minimize the items that are displayed. Note that activating or deactivating does not imply that they are removed from the database. The **PLOT** processor facilitates this activation and deactivation process by using the **VISIBLE** and **INVISIBLE** commands.
Table 1-1 and Table 1-2 summarize the functions of the three-mouse buttons in the graphics and menu areas.

### Table 1-1  Mouse Button Functions in Graphics Area

<table>
<thead>
<tr>
<th></th>
<th>&lt;ML&gt;</th>
<th>&lt;MM&gt;</th>
<th>&lt;MR&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>single pick or box pick</td>
<td>single pick or box pick</td>
<td>unpick</td>
<td>end of list</td>
</tr>
<tr>
<td>SHIFT</td>
<td>single pick or box pick with cross hairs</td>
<td>unpick</td>
<td>end of list</td>
</tr>
<tr>
<td>CTRL</td>
<td>polygon pick or lasso pick</td>
<td>unpick</td>
<td>end of list</td>
</tr>
</tbody>
</table>

### Table 1-2  Mouse Button Functions in Menu Area

<table>
<thead>
<tr>
<th></th>
<th>&lt;ML&gt;</th>
<th>&lt;MM&gt;</th>
<th>&lt;MR&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>command selection</td>
<td>command selection</td>
<td>on-line help</td>
<td>return</td>
</tr>
</tbody>
</table>
Identifiers

In many applications, an identifier is associated with a group of data. These applications include material properties, link properties, geometric properties, boundary conditions, initial conditions, tables, transformations, beam sections, loadcases, and jobs. The identifier can be any name, if none is given a default name is given. These id names are then referenced in other commands. The use of ids is detailed below.

When using many of the menus, the following buttons will appear.

The **NEW** button creates a new entry in the list of applications and makes it the current application.

The **REM** button removes the current application id and the associated data.

The **NAME** button allows you to provide a name to the current application.

The **COPY** button creates a new entry in the list of applications by copying the current application id; the new entry becomes the current application.

The **PREV** button selects the previous id and makes it the current application.

The **NEXT** button selects the next id and makes it the current application.

The **EDIT** button displays a list of the id’s and allows you to select a particular id. The selected id becomes the current one.
Main Menu Overview
Preprocessing Menus
Analysis Menus
Postprocessing Menus
Configuration Menus
Quit Command
Static Menus
Utilities Menus
Files Menu
Plot Menu
View Menu
Static Commands, Continued
List Commands
Main Menu Overview

To begin a Mentat session, move to the directory into which you would like your files to be saved, or in which previous files exist, and type `mentat`. The directory to which you have moved `cd` or `setdir` is your working directory.

The main menu is displayed on the screen. This menu consists of preprocessing, analysis, postprocessing, and configuration menus along the left of the screen, and the universal command menus along the bottom.

![Main Menu Diagram](image)

**Figure 2-1** Main Menu
Preprocessing Menus

**MESH GENERATION**

This menu contains commands that are used to create and/or edit geometry and the finite element mesh. Functionality includes adding, removing, and editing nodes, elements, points, curves, surfaces, and solids of new or existing meshes. In addition to these basic functions, the MESH GENERATION menu contains several submenus with commands for transforming the finite element mesh or the geometry.

**BOUNDARY CONDITIONS**

This menu contains commands that apply boundary conditions to the finite element mesh or the underlying geometry. The types of boundary conditions that can be applied depend upon the type of analysis being performed. Examples of boundary conditions include:

- fixed displacements
- pressures
- heat fluxes
- voltages
- potentials
- loads
- temperatures
- velocities
- currents
- charges

**INITIAL CONDITIONS**

This menu contains commands that are used to define the initial conditions. The type of initial conditions that can be applied depend upon the type of analysis being performed. Example of initial conditions include:

- Initial Velocity
- Initial Stress
- Initial Void Ratio
- Initial State Variables
- Initial Temperatures

**LINKS**

This menu contains commands for defining links or multi-point constraints (MPCs). The types of links that can be created are nodal ties, servo links, and springs/dashpots.
This menu contains commands that specify material properties to be used in the analysis. Material properties may be specified for the following analysis types:

- mechanical
- heat transfer
- joule heating
- acoustic
- bearing
- electrostatic
- magnetostatic
- electromagnetic

Typical material properties include: Young’s modulus, Poisson’s ratio, mass density, conductivity, specific heat, viscosity, and resistivity.

Default material properties are used if they are not specified. It is required that all the elements are assigned some material properties for a proper analysis.

This menu contains commands for defining sets of geometric properties and applying them to elements in the model. The types of properties that can be defined include truss area, beam properties, and plate/shell thickness. Additionally, element behavior such as constant dilatation and assumed strain formulation may be activated.

This menu contains commands for defining contact bodies in a contact analysis. The contact bodies can be classified as either rigid or deformable.

This menu contains commands for defining the fracture mechanics application which use the J-integral for investigating cracks in a structure.

This menu contains commands for defining the remeshing characteristics to be used when remeshing is performed in the analysis.
With this menu, the user enters the area where the design variables and design constraints can be specified for a design optimization or design sensitivity analysis job.

Design optimization refers to the process by which certain parameters (design variables) of a finite element model are modified in order to reach a feasible and desirable (improved) design. Design sensitivity, in the context of Marc, refers to the process whereby first-order derivatives of response quantities with respect to each of the design variables are computed and output. Element contributions to each response quantity are also computed and are made available.

### Analysis Menus

This menu contains commands that specify the parameters needed to perform one or more increments of analysis using the Marc program. The operations that may be performed include load selection, specification of load distribution defining the time step and specifying the convergence requirements. It is not required to set up the loadcase for a simple linear analysis.

The main objective of the menu is to combine individual boundary conditions into a loadcase.

This menu contains commands that define, run, and monitor the analysis. Multiple loadcases may be assembled into one analysis job. The output is controlled and analysis parameters are defined.
Postprocessing Menus

This menu contains commands that control the viewing of the analysis results. There are commands for selecting and manipulating analysis results files, selecting the results quantity for viewing, and for setting the viewing style and associated parameters. This includes deformed plots, contour plots, vector plots, X-Y displays, and time-history plots.

Configuration Menus

The DEVICE menu contains commands used for setting up and/or changing the appearance of the Mentat menu fonts. Control over the mouse accuracy is also specified here.

The VISUALIZATION menu is used to control the graphics display. What is to be displayed, the colors, the orientation and the lighting can be controlled here.

Quit Command

This button terminates a Mentat session, after confirming that you really want to exit.

Static Menus

The menus used most frequently in Mentat are called Static menus and are located below the graphics area of the Mentat window. These are the UTILITIES, FILES, PLOT, and VIEW menus. The PLOT and VIEW menus are also available in the VISUALIZATION menu. The remaining Static Commands are documented beginning on page 2-23.
The first of the Static menus is the UTILITIES menu. It contains commands which allow you to print the screen image, create or execute a procedure file, create command aliases, define parameters, determine the size of the database, annotate the display, and obtain geometric and coordinate data.

![Utilities Menu, Part I](image)

When the MORE button is selected, the remainder of the UTILITIES menu is displayed.
These PostScript commands produce a copy of graphics area on either a color or black and white PostScript printer. You can select any one of the printers that are available at your site. This is configured at the time of installation. For additional details, see the *MSC.Marc and Marc Mentat Installation and Operations Guide*. The COLOR FILE and GRAY FILE options produce a PostScript file.

Upon selecting SETTINGS, the following menu appears.

**Figure 2-3**  PostScript Settings Submenu

**PAGE WIDTH**

This command allows you to specify the width of the page in inches, for subsequent PostScript plotting of image files.
PAGE WIDTH
This command allows you to specify the height of the page in inches, for subsequent PostScript plotting of image files.

X ORIGIN
This command allows you to specify the X origin of the image in inches, for subsequent PostScript plotting of image files.

Y ORIGIN
This command allows you to specify the Y origin of the image in inches, for subsequent PostScript plotting of image files.

76 DPI, 150 DPI, and 300 DPI
These commands allows you to specify the resolution of the image to be printed.

COMPRESSED
Normally, compression is used to minimize the size of file when printing. Some systems do not support this, in which case it should be disabled.

PORTRAIT
This command aligns the image with the page in subsequent PostScript files (no rotation of the image). Thus, the current page width is used for the image’s width.

LANDSCAPE
This command rotates the image 90 degrees in subsequent PostScript files. Thus, the current page height is used for the image’s width, and this often results in a larger printed image. This will be the effect when the graphics window width is greater than its height, and when the current page height is greater than its width.

RASTER
This command selects the method of producing PostScript files. By default, it draws the Mentat geometry to a virtual graphics window of arbitrary size (resolution). With this method the final PostScript file contains a raster representation of the drawn image.

VECTOR
This command selects the method of producing PostScript files. It simply writes raw geometric primitives such as lines and text to the output file.
**THIN LINES**

This command turns the thin line option for raster PostScript output ON or OFF. This feature (which is off by default) makes all the drawn lines have a width of one dot or pixel. This can be desirable for high resolution images that have many lines (such as a mesh with many thousands of elements). When this option is off, a thicker line width is used, which compensates for varying resolutions.

**Predefined Colormaps**

These commands sets the current color map for the program to be 1 through 8. There are eight predefined collections of colors that the program can be made to use for menus and ordinary geometry with this command. Each of these collections maps a red, green, and blue value to each color index. Clarification of the colormaps are 1 - 4 = color, 5, 6 = gray, and 7, 8 = black and white. By default color map 1 is used.

**Predefined Contourmaps**

These commands sets the current contour map for the program to be 1 through 8. There are eight predefined collections of colors that the program can be made to use for displaying contours with this command. Each of these collections maps a red, green, and blue value to each color index. Clarification of the contourmaps are 1 - 4 = color, 5, 6 = gray, and 7, 8 = black and white. By default color map 1 is used.

**Utilities Continued**

**SNAPSHOT**

This command activates the menu used to capture the graphical window into a file. This file may be displayed, printed, or transferred to other desktop publishing programs.

**ANIMATION**

This command controls the creation and playback of the animation file.

**XDUMP 1 and XDUMP 2**

Both of these commands controls the plotting of the complete screen image. If selecting the XDUMP option, a file name must be given. This file is then processed outside of Mentat.
FULL WINDOW
This command controls the creation and playback of the animation file.

PARAMETERS
These commands define parameters. These parameters may be used at any point in the session, or in a procedure file.

ALIASES
These commands control defining aliasing of commands.

PROCEDURES
This command allows you to save commands into a file, or to replay a file containing previously created commands.

PYTHON
These menus allow you to select a Python script to run and whether the script is to be run as a separate process which allows the graphics to be updated as the script proceeds. It also allows for the initialization of a connection so that a Python script may be started in an external window.

CURRENT DIRECTORY
This command lists the files in a directory. To obtain files in the current directory, type “.”

EDIT FILE
This command edits a file with the editor.

LIST DIRECTORY
This command allows you to enter a directory path.

SYSTEM COMMAND
This command enters one system command at a time.

SYSTEM SHELL
This command changes the modes between C-shell and Bourne shell.
DISTANCE
This command calculates the distance between two points, and the angles from the global axes.

CALCULATE
This command calculates the value of an expression.

SIZES
This command brings up the submenu that reports on the size of the database, the number of nodes, elements, points, curves, surfaces, and solids in the geometry, the number of material and boundary condition ids, etc.

SAMPLE ELEMENT
This command returns the selected element number and the coordinates of an arbitrary point.

RESET PROGRAM
This command resets the program flags to their default values. This does not modify the database nor does it reset the view menu commands. To initialize the database, use the NEW command in the FILES menu.

UNDO
This command sets the undo capability on or off. If set to off, you cannot undo a command. The program will not backup the database in this mode.

GENERALIZED XY PLOT
This command has the ability to collect plots from various plotters: History plot, Response Gradient/Design Variable plot, Path plot, Tables and Data Fit plot. The overlapping feature allows the user to compare plots.

Utilities, More Menu

Annotations
The following commands are used to add annotations to the graphical display. Annotations can currently be added to the display of the model. They cannot be added to path or history plots. Annotations are tied to a view and will not appear in all views.
DISPLAY ANNOTATIONS
This command sets the annotations on or off.

ADD
This command adds an annotation; type in the annotation and give the location.

REMOVE
This command removes an annotation. Select the annotation to be removed.

SHOW
This command lists all active annotations.

EDIT
This command edits an existing annotation. Select the annotation to be edited.

MOVE
This command redefines the location of the annotation. Select the annotation, and give the new location.

COPY
This command copies an annotation to a new location. Select the annotation, and give the location.

CLEAR
This command removes all annotations.

Length, Area, & Volume Calculations

Direct Methods

EDGE LENGTH
This command determines the total length of a list of element edges.

FACE AREA
This command determines the total area of a list of element faces.

ELEMENT VOLUME
This command determines the total volume of a list of solid elements.
ELEMENT MASS
This command determines the mass of an element.

SOLID AREA
This command determines the solid surface area of element.

SOLID VOLUME
This command determines the solid volume of an element.

Enclosing Methods

EDGE AREA
This command calculates the area enclosed by a list of edges.

FACE VOLUME
This command calculates the volume enclosed by a list of faces.
Files Menu

The next Universal menu is the FILES menu. This menu controls the saving of the database, interfacing with CAD systems and the Marc input file.

![Figure 2-4 Files Submenu]

**Model**

The name of the current database is given here.

**Binary**

This command selects between the Mentat database being stored as a binary file (.mud) or an ASCII text file (.mfd).

**Description**

This command brings up a menu that allows the user to set the description for the current model. The text field is used for entering the description. Either a return typed into the text field or click on the OK button will cause the `model_description` command to be executed and close the popup menu.
This command removes all contents of the in-core memory data structure, and initialize all counters to zero.

This command opens an existing database and reads it into memory.

This command opens an existing database and merges the contents with the current database. The elements, nodes, etc. will be renumbered and be given the higher numbers. To eliminate nodes and points at identical positions, use the SWEEP option in the MESH GENERATION menu.

This command controls the renumbering of model entities during a model merge operation. When a model is renumbered, all entities will be assigned new ids based upon their creation order. A merged model will receive higher ids than the existing model.

This command saves the existing database to a file.

This command saves the existing database to a new file. The new name becomes the current name of this session.

This command specifies the format of the model file. It only affects the save_as_model (SAVE AS) command. The formats are: 2001 Style, 2000 Style, 3.3 Style, 3.2 Style, 3.1 Style, and 2.3.1 Style.

This command restores the last written database into memory.
Interfaces

Import

This command invokes the menu that controls reading in geometric and/or finite element data from external programs. Upon selecting IMPORT, the following menu appears.

![Files Import Submenu](image)

Export

This command invokes the menu that controls writing of geometric and finite element data in a form to be used by external programs. Upon selecting EXPORT, the following menu appears.

![Files Export Submenu](image)
MSC.Marc Input File

Whenever you request to either read or write a file, the Mentat file browser menu will appear. This allows you to move around in the directory system, select an existing file, or associate a name with a new file. The following menu appear.

![File Browser Menu](image)

**Figure 2-7** File Browser Menu

This allows you to search through a directory targeting files with a certain prefix and/or suffix.

In Mentat the following filters will be used by default.

<table>
<thead>
<tr>
<th>Reading/Writing File Type</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reading/Writing File Type</td>
</tr>
<tr>
<td>Database</td>
<td>.mfd</td>
</tr>
<tr>
<td>Procedure file</td>
<td>.proc</td>
</tr>
<tr>
<td>MARC data file</td>
<td>.dat</td>
</tr>
<tr>
<td>ACIS geometry</td>
<td>.sat</td>
</tr>
<tr>
<td>AutoCad</td>
<td>.dxf</td>
</tr>
<tr>
<td>I-DEAS</td>
<td>.unv</td>
</tr>
<tr>
<td>IGES</td>
<td>.igs</td>
</tr>
<tr>
<td>Nastran</td>
<td>.nas</td>
</tr>
<tr>
<td>Patran neutral file</td>
<td>.pat</td>
</tr>
</tbody>
</table>
**Chapter 2: Main Menu**

### Directories

This group selects the directory to move to.

To move up in the directory structure select `..`

### Files

This group selects the file to be read from or written to.

If the file is to be written to, and it already exists, you will be prompted to overwrite the existing file.

### Reading/Writing

<table>
<thead>
<tr>
<th>File Type</th>
<th>Text File</th>
<th>Binary file</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDAFS</td>
<td>*.vda</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Restart</td>
<td>*.t08</td>
<td></td>
</tr>
<tr>
<td>Post file</td>
<td>*.t19</td>
<td>*.t16</td>
</tr>
<tr>
<td>PostScript</td>
<td>*.ps</td>
<td></td>
</tr>
<tr>
<td>User subroutine</td>
<td>*.f</td>
<td></td>
</tr>
<tr>
<td>Radiation viewfactor</td>
<td>*.vfs</td>
<td></td>
</tr>
</tbody>
</table>

This filed allows you to enters the name of the file to be read from or written to, if you cannot pick on it. This is the way to enter new file names.

This command cancels the file browser. This will also cancel the requested read/write operation.
This command resets the file browser to the users working directory.

This command rescans the current file system to update the list of files with those files created or removed since the last time this directory was read.

This command accepts the name of the file given in the SELECTION area, and proceed to do the requested I/O operation.

This command renumbers all entities in the model (nodes, elements, etc.) starting at the start id (default 1), and incrementing by the specified increment (default 1).

This command allows you to change the program’s current directory. The current directory is initially the directory from which Mentat was started. Relative path and file names specified to Mentat begin at the current directory.

This command opens a window with a text editor which enables you to edit a given file.

This command resets all program controls except viewing to the default values. The contents of the database are not changed. To remove the contents of the database, use the NEW command.
Plot Menu

The PLOT menu controls the style of the plot geometry. It is typically used to control what type of geometric entities are to be displayed, labelled, whether they are in wireframe or filled mode.

Figure 2-8  Plot Menus
View Menu

The VIEW menu contains commands which control the viewing of the model.

![View Menu](image)

**Figure 2-9** View Menu
Static Commands, Continued

The remainder of the Static files are as follows:

**UNDO**

This command restores the database to its last saved position. This can be used to recover from an inadvertently typed command.

**DRAW**

This command re-draws the display. This is often used when one is in DRAW UPDATE MANUAL mode.

**FILL**

This command centers and enlarges the image so that it fills the display region.

**RESET VIEW**

This command resets the view to its original (program startup) state. The view’s camera position and settings and model viewing transformations are all reset. This command does not alter translation, rotation, scale, or zoom increments.

**DYN. MODEL**

This command turns dynamic model on and off. When dynamic model is enabled, you can move your model on the screen directly using the mouse, rather than by repeatedly using the translate (trans_) and rotate (rot_) commands.

With dynamic model on, holding the left mouse button <ML> down and moving the cursor in the graphics area translates the model, holding the middle mouse <MM> button down and moving the cursor rotates the model, and holding the right mouse button <MR> down and moving the cursor scales the model.

As an alternative to toggling the dyn. model button, the user can hold the ALT key and use the mouse buttons, except on Windows/95 and Windows/NT workstations.

**TX+ TX- TY+ TY- TZ+ TZ-**

These commands translates the model along the camera space X, Y, and Z axes, each in the positive or negative direction, and each by a distance set with the trans_model_increment command. This command acts on all the currently active views.
These commands rotates the model in the specified directions with respect to the view space.

This command allows you to construct a box using the mouse to zoom in on a part of the model.

This command moves the camera in.

This command moves the camera out.

This command bring up a user-defined menu, specified in user.ms, that provides a convenient location to place buttons for often-used commands and shortcuts to other menus.

These menus provide information on how to use the help system and contains links to all of the online documentation.
List Commands

The following common commands do not appear in the MAIN menu, but appear in nearly all other Mentat menus. Since they are globally located, they are documented in this chapter alone for clarity.

These pseudo-commands, which are always available regardless of which Mentat menu you are using, are shortcuts for providing entity lists to the command line. They may be used whenever a command expects a list of entities. Each pseudo-command generates a list of entities of the requested type and sends the list to the command requesting the list. The types of lists generated are as follows:

- **ALL**: This command lists all entities of the requested type.
- **SELECT**: This command lists all entities of the requested type that *are* currently selected.
- **UNSEL**: This command lists all entities of the requested type that *are not* currently selected.
- **VISIBLE**: This command lists all entities of the requested type that are *visible*.
- **INVIS**: This command lists all entities of the requested type that are *invisible*.
- **OUTLINE**: This command lists all entities of the requested type on the *outline* of the model.
- **SURFAC**: This command lists all entities of the requested type on the *surface* of the model.
Select Processor

The SELECT menu is used to manage lists of entities. A variety of filters, methods, and modes are available which provide convenient means for the generation of the lists.

The SELECT commands are used to place entities into a selected state and are used in conjunction with the ALL SELECTED and ALL UNSELECTED list macros to generate input to commands requiring a list of entities.

The SELECT processor may also be used to control the visibility of entities.

This command lists the currently defined sets.

This command terminates the current list of entries being entered by sending a pound sign (#) to the command line. When entering a list of items with the mouse, clicking on the left button while the mouse is in the graphics area has the same effect.
Mentat Help Commands in A

Command: activate_all_views

Description: This command (along with activate_view and deactivate_view) is used for multiple view management. Only active views are affected by view setting commands. The visibility of views is controlled by show_view and the show_all_views commands. Views may be visible but not active and vice versa. When the program is started, view 1 is the only active and visible view.

This command will activate all views.

When a view is made visible, it is automatically activated. When a view is made invisible, it is automatically deactivated. The three commands activate_all_views, activate_view, and deactivate_view allow you to override this.

Also see: activate_view, deactivate_view, and show_view.

Keyboard Command Sequence:

activate_all_views
Command: activate_view
deactivate_view

Description: These commands are used for multiple view management. Only active views are affected by view setting commands. The visibility of views is controlled by show_view and the show_all_views commands. Views may be visible but not active and vice versa. When the program is started, view 1 is the only active and visible view.

You must specify the view to be operated on. Views are specified by number, 1-4.

When a view is made visible, it is automatically activated, and when a view is made invisible, it is automatically deactivated. The three commands activate_all_views, activate_view, and deactivate_view allow you to override this.

Also see activate_all_views and show_view.

Keyboard Command Sequence:

activate_view <view number>
deactivate_view <view number>

Other Buttons with the same description:

2
3
4

Command: adapg_name

Description: This command sets or changes the name of the current entry in the list of remeshing criteria.

Keyboard Command Sequence:

adapg_name <adapg name>
**RESET**

**Command:** `adapg_reset`

**Description:** This command resets the settings for the current remeshing criteria to its default values.

**Keyboard Command Sequence:**

`adapg_reset`

---

**REMESH BODY**

**Command:** `adapg_rmsh_body`

**Description:** This command selects which contact body to remesh using the current settings.

The contact body must have been previously defined; see menus under `MAIN->CONTACT`.

**Keyboard Command Sequence:**

`adapg_rmsh_body`

---

**ANGLE DEVIATION**

**Command:** `adapg_option`

**Description:** This command selects which remeshing criteria to use.

- **ELEMENT DISTORTION**: Remesh if the element is too distorted or anticipating the elements will be too distorted. Check the angle of deviation.
- **PENETRATION**: Remesh if penetration between contact bodies is larger than the user input limit or 2 times the contact tolerance distance.
- **INCREMENT**: Remesh at increments specified by `FREQUENCY`.
- **ANGLE DEVIATION**: Remesh when element inner angle change from the undeformed mesh is greater than the input value.
Also remesh if element inner angle is greater than 175 degrees or less than 5 degrees.

**IMMEDIATE**
Remesh before the analysis begins.

**STRAIN CHANGE**
Remesh if strain change from the last remeshing in any element of the current contact body has reached the specified maximum value.

**VOLUME RATIO**
Remesh if any volume ratio associated with an element corner node is less than the input value. The volume ratio is defined to measure the flatness of a 3-D element. For tetrahedral element, 1 is a good element and 0 is a flat element. For hexahedral element, 1 is a good corner and 0 is a flat corner or very sharp corner.

*Note:* See Volume A or Volume C for more detailed information on remeshing criteria.

**Keyboard Command Sequence:**

```
adapg_option distortion_crit:<on/off>
adapg_option penetration_crit:<on/off>
adapg_option increment_crit:<on/off>
adapg_option angle_dev_crit:<on/off>
adapg_option strainchange_crit:<on/off>
adapg_option volume_ratio_crit:<on/off>
```

Other Buttons with the same description:

- **ELEMENT DISTORTION**
- **FROM**
- **IMMEDIATE**
- **INCREMENT**
- **PENETRATION**
- **USER LIMIT**

**Command:** `adapg_option`

**Description:** This command sets the element size for the current remeshing.

**ELEMENT EDGE LENGTH**
This element edge length is used as the target element edge length. The new elements will have edges of approximately this length.
# ELEMENTS
Use the target number of elements to control element size. Only the approximate number of the elements can be reached. It is an alternate to the element edge length.

PREVIOUS # ELEMENTS
Use the number of elements in the previous step as the target number of elements.

**Keyboard Command Sequence:**
```plaintext
adapg_option nelems_ctrl:<off/on/previous>
```

Other Buttons with the same description:
- # ELEMENTS
- PREVIOUS # LENGTH

---

**INSIDE COARSENING LEVELS**

**Command:** `adapg_param`

**Description:** This command sets the parameters for the current overlay remeshing criterion.

- **MAX ELEMENT EDGE LENGTH**
  This length is used as a target element edge length by the overlay mesher. The new elements will have edges of approximately this length.

- **OUTSIDE REFINING LEVELS**
  This number is used to split elements on the boundary to better represent the geometry. Maximum 2 levels can be used.

- **INSIDE REFINING LEVELS**
  This number is used to merge elements inside the contact bodies, therefore reduce number of elements used in the analysis. Maximum 1 level is allowed.

- **# ELEMENTS**
  This is the target number of elements required after remeshing. It is an alternate to the element edge length.

- **CHANGE OF # ELEMENTS(%)**
  Percentage change of the number of elements allowed after remeshing. This is used to avoid big change of # elements after remeshing.
**Keyboard Command Sequence:**

```
adapg_param max_edge_len <value>
adapg_apram refine_levels <value>
adapg_param coarsen_levels <value>
adapg_param nelems <value>
adapg_param dnelems_pct <value>
```

Another Button with the same description:

OUTSIDE REFINING LEVELS

---

**Command:** `adapg_param`

**Description:** This command sets the parameters for the current remeshing criterion.

- **ELEMENT EDGE LENGTH**
  
  This length is used as a target element edge length by the overlay mesher. The new elements will have edges of approximately this length.

- **MIN. ELEMENT EDGE LENGTH**
  
  Minimum element edge length is allowed in remeshing. By default, it is 1/3 of the target element edge length.

- **# ELEMENTS**
  
  This is the target number of elements required after remeshing. It is an alternate to the element edge length.

- **CHANGE OF # ELEMENTS(%)**
  
  Percentage change of the number of elements allowed after remeshing. This is used to avoid big change of # elements after remeshing.

- **CURVATURE CONTROL**
  
  This is the number of divisions of a circle. Use this number to control refinement of outline with different curvatures.

- **SMOOTHING RATIO**
  
  This ratio is used to control the outline smooth for remeshing. 0 means no smoothing and 1 means uniform.

- **FEATURE VERTEX ANGLE**
  
  An angle between two edge vectors pointing outward of a point. Any point with the vertex angle smaller than this
angle will be considered as a hard point. A hard point will be kept as an element node after remeshing. Users are recommended to use the default value.

**Keyboard Command Sequence:**

```plaintext
adapg_param adv_edge_len <value>
adapg_param min_edge_len <value>
adapg_param nelems <value>
adapg_param dnelems_pct <value>
adapg_param curvature_ndiv <value>
adapg_param smooth_ratio <value>
adapg_param feat_vert_ang_2d <value>
adapg_param refine_levels <value>
adapg_param coarsen_levels <value>
```

Other Buttons with the same description:

- #DIV MIN. ELEMENT EDGE LENGTH
- CHANGE OF # ELEMENTS (%) PREVIOUS # ELEMENTS
- CURVATURE CONTROL SET
- ELEMENT EDGE LENGTH SMOOTHING RATIO

**Command:** **adapg_param**

**Description:** This command sets the parameters for the Patran remeshing criterion.

- **ELEMENT EDGE LENGTH**
  This length is used as a target element edge length by the surface mesher. The new elements will have edges of approximately this length.

- **FEATURE EDGE ANGLE**
  An angle between normal vectors of two neighboring surfaces. Any edge with the edge angle larger than this angle will be considered as a soft feature edge. A soft edge will be kept after the remeshing but new nodes can be placed on the edge. Users are recommended to use the default value.

- **FEATURE VERTEX ANGLE**
  An angle between two edge vectors pointing outward of a point. Any point with the vertex angle smaller than this angle will be considered as a hard point. A hard point will be kept as an
element node after remeshing. Users are recommended to use the default value.

COARSENING FACTOR  This factor can be used to gradually enlarge the tetrahedral element size from the surface to the interior region during the remeshing. It will help reduce the total number of elements in the mesh. Users are recommended to use the default value.

MINIMUM EDGE LENGTH  This element size is used in local refinement to control the smallest element size.

MAXIMUM EDGE LENGTH  This element size is used in local refinement to control the largest element size.

CURVATURE CONTROL  This parameter controls local refinement based on the curvature of the surfaces and curves. The number of divisions represents the division of a cycle.

CHANGE ELEMENT TYPE  Patran Tetra mesher supports only Tetrahedral element type 157. If you need to convert element from Hex element you need to change the element type to 157 together with the IMMEDIATE command.

Keyboard Command Sequence:

```
adapg_param del3_edge_len <value>
adapg_param feat_edge_ang <value>
adapg_param feat_vert_ang <value>
adapg_param coarsen_factor <value>
adapg_param pat3_min_edge_len <value>
adapg_param pat3_max_edge_len <value>
adapg_param pat3_curvature_ndiv <value>
```

Other buttons with the same description:

SET
MIN. ELEMENT EDGE LENGTH
**ADVANCING FRONT QUAD**

**Command:** `adapg_type`

**Description:** This command selects the type of mesher to use for remeshing. The menu below these menus are used for defining the remeshing criteria to use.

- **ADVANCING FRONT QUAD**: Quadrilateral mesh using the advancing front method.
- **OVERLAY QUAD**: Quadrilateral mesh using the overlay method.
- **ADVANCING FRONT TRIA**: Triangular mesh using the advancing front method.
- **DELAUNAY TRIA**: Triangular mesh using the Delaunay method.
- **PATRAN TETRA**: Patran 3-D tetrahedral mesh.

For quadrilateral meshes, the advancing front method is recommended while the Delaunay method is recommended for triangular meshes.

**Keyboard Command Sequence:**
```
adapg_type <advfront_quad/overlay_quad/
advfront_tria/delaunay_tria/patran_tetra>
```

**Other Buttons with the same description:**
- **ADVANCING FRONT TRIA**
- **DELAUNAY TRIA**
- **OVERLAY QUAD**
- **PATRAN TETRA**

**NAME**

**Command:** `adapt_name`

**Description:** This command sets or changes the name of the current adapt.

An adapt is a collection of parameters that controls the adaptive meshing of a set of elements during an analysis.

**Keyboard Command Sequence:**
```
adapt_name <adapt name>
Command:  adapt_param

Description:  This command sets a parameter value for the current adaptivity criterion. You must specify the parameter name and its value. The applicable parameter that may be set depends on the type of the current adapt.

Keyword Command Sequence:

   Adapt_param <Parameter Name> <Value>

Other Buttons with the same description:

   F2 through F5
   VALUE
   VALUE (ABS)
   VALUE (REL)
   XMAX
   XMIN
   YMAX
   YMIN
   ZMAX
   ZMIN

Command:  adapt_param xmin|ymin|zmin|xmax/|ymax|zmax

Description:  This command defines the box for the NODE WITHIN BOX criterion. The box is defined in global X, Y, and Z as:

   XMIN, YMIN, ZMIN to XMAX, YMAX, ZMAX.

Keyword Command Sequence:

   adapt_param <xmin|ymin|zmin|xmax/|ymax|zmax> <value>

Other Buttons with the same description:

   YMIN
   ZMIN
   XMAX
   YMAX
   XMAX
**MAX # LEVELS**

**Command:** adapt_param levels

**Description:** This command sets the number of levels an element is allowed to be subdivided. If set to zero the elements will not be subdivided.

**Keyboard Command Sequence:**
adapt_param <param_levels>

---

**VALUE (REL)**

**Command:** adapt_param

**Description:** These commands set the parameter values for the `uadap` criterion.

- **VALUE (REL)**: an element is subdivided or unrefined if `USER/max(USER) > REL` where `USER` is defined in the user subroutine and `max(USER)` is the largest value for any element.

- **VALUE (ABS)**: an element is subdivided or unrefined if `USER > ABS`.

**Keyboard Command Sequence:**
adapt_param value <param_value>
adapt_param val_abs <param_value>

Other Buttons with the same description:
VALUE (ABS)

---

**UNREFINE**

**Command:** adapt_option unrefine:<on|off>

**Description:** When this option is activated, elements are unrefined when all nodes of an element leaves the box.

**Keyboard Command Sequence:**
adapt_option unrefine:<on|off>
**VALUE**

Command: `adapt_param value`

Description: This command sets the value for the TEMPERATURE GRADIENT criterion. The value must be less than 1.

Keyboard Command Sequence:

```
adapt_param value <temperature>
```

**RESET**

Command: `adapt_reset`

Description: This command resets all values for the current adaptivity criterion.

Keyboard Command Sequence:

```
adapt_reset
```

**NODE WITHIN BOX**

Command: `adapt_type box`

Description: This command sets the criterion type to NODE WITHIN BOX.

The user specifies a box, which is aligned along the global coordinates. An element is subdivided when at least one of its nodes enters the box. When the unrefine option is activated in the submenu, the element will be unrefined when all its nodes leave the box.

Keyboard Command Sequence:

```
adapt_type box
```
Command: adapt_type_eq_strs_rel
adapt_type_eq_strs_abs
adapt_type_eq_strn_rel
adapt_type_eq_strn_abs
adapt_type_eq_pstrn_abs
adapt_type_eq_pstrn_rel
adapt_type_eq_cpstrn_rel
adapt_type_eq_cpstrn_abs

Description: These commands set the criterion type to one of the equivalent element quantity based criteria.

For the (REL) variants, an element is subdivided if

\[
\text{element value > VALUE} \times \text{maximum value}
\]

For the (ABS) variants, an element is subdivided if

\[
\text{element value > VALUE}
\]

VALUE is the value specified in the submenu to this command. The maximum value is the largest value in the whole model, for example the largest von Mises stress.

Keyboard Command Sequence:

adapt_type eq_strs_rel
adapt_type eq_strs_abs
adapt_type eq_strn_rel
adapt_type eq_strn_abs
adapt_type eq_pstrn_abs
adapt_type eq_pstrn_rel
adapt_type eq_cpstrn_rel
adapt_type eq_cpstrn_abs

Other Buttons with the same description:

- EQUIV. STRESS (ABS)
- EQUIV. STRAIN (REL)
- EQUIV. STRAIN (ABS)
- EQUIV. PLASTIC STRAIN (ABS)
- EQUIV. PLASTIC STRAIN (REL)
- EQUIV. CREEP STRAIN (REL)
- EQUIV. CREEP STRAIN (ABS)
**TEMPERATURE GRADIENT**

**Command:**  adapt_type max_grad  
**Description:**  This command sets the criterion type to TEMPERATURE GRADIENT.  

An element is subdivided if the temperature gradient in the element is greater than a fraction of the maximum gradient. The fraction is given as the value in the menu. It must be less than 1, and a typical value is 0.75.

**Keyboard Command Sequence:**  
adapt_type max_grad

**MEAN STRAIN ENERGY**

**Command:**  adapt_type mean_strn  
**Description:**  This command sets the criterion type to MEAN STRAIN ENERGY.  

An element is subdivided if

\[ \text{element strain energy} > \text{VALUE} \times \frac{\text{total strain energy}}{\text{numel}} \]

where VALUE is specified in the submenu and numel is the number of elements in the job.

**Keyboard Command Sequence:**  
adapt_type mean_strn

**NODE IN CONTACT**

**Command:**  adapt_type node_cont  
**Description:**  This command sets the criterion type to NODE IN CONTACT.  

An element is subdivided if at least one of its nodes comes into contact or is part of a contacted segment.

**Keyboard Command Sequence:**  
adapt_type node_cont
### USER SUB. UADAP

**Command:** `adapt_type usersub`

**Description:** This command sets the criterion type to USER SUBROUTINE. The adaptivity criterion is defined in the user subroutine `uadap` (see *MSC.Marc Volume D: User Subroutines and Special Routines* for a description).

Two values (REL and ABS) are specified for this criterion. The subroutine is called for each element and should return a value (USER). An element is subdivided if

\[
\frac{\text{USER}}{\max(\text{USER})} > \text{REL}
\]

or

\[
\text{USER} > \text{ABS}
\]

where max(USER) is the largest value of USER for any element. If REL or ABS is zero the check is not done.

It is also possible to unrefine elements with the user subroutine `uadap2`. This is activated with the UNREFINE button in the submenu. The subroutine is called for all active elements. It is not called for elements that have been subdivided into child elements since they are inactive. Similar to `uadap`, `uadap2` defines a value \(\text{USER}_2\). A subdivided element is unrefined if the criterion

\[
\frac{\text{USER}_2}{\max(\text{USER}_2)} > \text{REL}
\]

or

\[
\text{USER}_2 > \text{ABS}
\]

is satisfied for all its child elements. When an element is unrefined, its child elements are made inactive and the original element is activated again.

**Keyboard Command Sequence:**

```
adapt_type usersub
```

### UNREFINE (USER SUB. UADAP2)

**Command:** `adapt_option unrefine`

**Description:** This command turns on the unrefinement option for the user subroutine criterion.
The unrefinement is controlled by user subroutine **uadap2** (see *MSC.Marc Volume D: User Subroutines and Special Routines* for a description).

The subroutine is called for all active elements. It is not called for elements that have been subdivided into child elements since they are inactive. Similar to **uadap**, **uadap2** defines a value USER2. A subdivided element is unrefined if the criterion

\[
\frac{\text{USER2}}{\text{max(USER2)}} > \text{REL}
\]

or

\[
\text{USER2} > \text{ABS}
\]

is satisfied for all its child elements. When an element is unrefined, its child elements are made inactive and the original element is activated again.

**Keyboard Command Sequence:**

```
adapt_option unrefine:<on|off>
```

----

**ZIENKIEWICZ-ZHU STRESS**

**Command:**  
```
adapt_type_z_z_strs  
adapt_type_z_z_strn  
adapt_type_z_z_pstrn  
adapt_type_z_z_cpstrn
```

**Description:** This command sets the criterion type to one of the Zienkiewicz-Zhu based criteria.

The subdivision is based upon an element based error norm calculated using the difference between element results averaged at nodes and unaveraged results. The different criteria use different element results for the error norm. See *MSC.Marc Volume A: Theory and User Information*, Chapter 4 for details.

**Keyboard Command Sequence:**

```
adapt_type z_z_strs  
adapt_type z_z_strn  
adapt_type z_z_pstrn  
adapt_type z_z_cpstrn
```

Other Buttons with the same description:

- ZIENKIEWICZ-ZHU STRAIN ENERGY
- ZIENKIEWICZ-ZHU PLASTIC STRAIN
- ZIENKIEWICZ-ZHU CREEP STRAIN
MEAN STRAIN ENERGY

Command: adapt_type

Description: This command selects the type of the current adapt.

Currently supported adapt types include the following:

- mean_strn: Mean Strain Energy
- z_z_strs: Zienkiewicz-Zhu Stress
- z_z_strn: Zienkiewicz-Zhu Strain
- z_z_pstrn: Zienkiewicz-Zhu Plastic Strain
- z_z_cpstrn: Zienkiewicz-Zhu Creep Strain
- box: Nodes Within Box
- node_cont: Nodes in Contact
- max_grad: Maximum Solution Gradient
- eq_strs_rel: Equivalent Relative Stress
- eq_strs_abs: Equivalent Absolute Stress
- eq_strn_rel: Equivalent Relative Strain
- eq_strn_abs: Equivalent Absolute Strain
- eq_pstrn_rel: Equivalent Relative Plastic Strain
- eq_pstrn_abs: Equivalent Absolute Plastic Strain
- eq_cpstrn_rel: Equivalent Relative Creep Strain
- eq_cpstrn_abs: Equivalent Absolute Creep Strain
- usersub: User defined via user subroutine

Keyboard Command Sequence:

    adapt_type <adapt type>

Other Buttons with the same description:

- ZIENKIEWICZ-ZHU STRESS
- ZIENKIEWICZ-ZHU STRAIN ENERGY
- ZIENKIEWICZ-ZHU PLASTIC STRAIN
- ZIENKIEWICZ-ZHU CREEP STRAIN
- EQUIV. STRESS (REL)
- EQUIV. STRESS (ABS)
- EQUIV. STRAIN (REL)
- EQUIV. STRAIN (ABS)
- EQUIV. PLASTIC STRAIN (REL)
- EQUIV. PLASTIC STRAIN (ABS)
- EQUIV. CREEP STRAIN (REL)
- EQUIV. CREEP STRAIN (ABS)
- NODE WITHIN BOX
- NODE IN CONTACT
- TEMPERATURE GRADIENT
- USER SUB. UADAP
**Command:**  
**add_adapt_elements**

**Description:**  
This command adds elements to the current adapt. The elements in this list will be adaptively meshed during the analysis according to the parameters in the current adapt. This button is located in the LOCAL ADAPTIVITY menu.

**Keyboard Command Sequence:**
```
add_adapt_elements <element list> #
```

**Command:**  
**add_annotation**

**Description:**  
This command adds annotations to a view. Annotations are strings of text that is used to describe what is being displayed in a view. You must enter the annotation text. Everything you type up to a carriage return will be used as the annotation text. This button is located in the UTILITIES->MORE->ANNOTATIONS menu.

Next, you are prompted for the view where you wish the annotation to be placed and the location in the view. Views are specified by number, 1-4. View X and Y coordinates specified in view screen units (minimum = 0, maximum = 1) are measured from the upper-left corner of the view with X being the horizontal direction and Y being the vertical.

**Keyboard Command Sequence:**
```
add_annotation <text> <view> <X> <Y>
```

**Command:**  
**add_apply_cavities**

**Description:**  
This command adds cavities to the current boundary condition cavity pressure/mass load. Cavities are created in the Cavities menu located under the MESH GENERATION menu. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CAVITY PRESSURE/MASS LOAD menus.

**Keyboard Command Sequence:**
```
add_apply_cavities <cavity list> #
```
Command: **add_apply_curves**

Description: This command adds curves to the current boundary condition application. Element faces which are attached to the curves are associated with the boundary condition. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

Keyboard Command Sequence:

```
add_apply_curves <curve list> #
```

Command: **add_apply_edges**

Description: This command adds edges to the current boundary condition application. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

Keyboard Command Sequence:

```
add_apply_edges <edge list> #
```

Command: **add_apply_elements**

Description: This command adds elements to the current boundary condition application. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

Keyboard Command Sequence:

```
add_apply_elements <element list> #
```
Command: **add_apply_faces**

Description: This command adds faces to the current boundary condition application. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

**Keyboard Command Sequence:**

```
add_apply_faces <face list> #
```

---

Command: **add_apply_nodes**

Description: This command adds nodes to the current boundary condition application. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

**Keyboard Command Sequence:**

```
add_apply_nodes <node list> #
```

---

Command: **add_apply_points**

Description: This command adds points to the current boundary condition application. Nodes which are attached to the points are associated with the current boundary condition. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

**Keyboard Command Sequence:**

```
add_apply_points <point list> #
```
Command:  add_apply_surfaces
Description: This command adds surfaces to the current boundary condition application. Element faces which are attached to the surfaces are associated with the boundary condition. This button is located in the BOUNDARY CONDITIONS->MECHANICAL and other boundary conditions menus.

Keyboard Command Sequence:
add_apply_surfaces <surface list> #

Command:  add_cavity_curves
Description: This command adds curves to the current cavity. The curves are really added to the cavity definition only if the curves are attached to cavity element edges. It is located under the MESH GENERATION->CAVITIES (CURVES) menu.

Keyboard Command Sequence:
add_cavity_curves <curve list> #

Command:  add_cavity_edges
Description: This command adds element edges to the current cavity. It is located under the MESH GENERATION->CAVITIES (EDGES) menu.

Keyboard Command Sequence:
add_cavity_edges <edge list> #

Command:  add_cavity_faces
Description: This command adds element faces to the current cavity. It is located under the MESH GENERATION->CAVITIES (FACES) menu.
Keyboard Command Sequence:

```
add_cavity_faces <face list> #
```

Command: **add_cavity_surfaces**

Description: This command adds surfaces to the current cavity. It is located under the MESH GENERATION->CAVITIES (SURFACES) menu.

Keyboard Command Sequence:

```
add_cavity_surfaces <surface list> #
```

Command: **add_cbody_dc_edges**

Description: This command adds edges for the determination of discontinuities in the analytical description for deformable contacted bodies in a 3-D analysis. For 2-D or axisymmetric analysis, a list of nodes must be given to define the discontinuities (see `add_cbody_dc_nodes`). This button is located in the CONTACT->CONTACT BODIES->DEFORMABLE and RIGID w HEAT TRANSFER menus with ANALYTIC option on.

In the analytical formulation for deformable bodies, Coons surfaces are defined through neighboring boundary segments to allow a more accurate description of curved surfaces. With this discontinuity option, you can define edges that should not be smoothed by this formulation.

Keyboard Command Sequence:

```
add_cbody_dc_edges <edge list> #
```

Reference: *MSC.Volume C, Program Input*, Chapter 3: Model Definition Options, SPLINE option.

Command: **add_cbody_dc_nodes**

Description: This command adds nodes for the determination of discontinuities in the analytical description for deformable contacted bodies in a 2-D or
axisymmetric analysis. This button is located in the CONTACT->CONTACT BODIES->DEFORMABLE and RIGID w HEAT TRANSFER menus with ANALYTIC option on.

For 3-D analysis, a list of edges must be given to define the discontinuities (see add_cbody_dc_edges).

In the analytical formulation for deformable bodies, splines are defined through neighboring boundary nodes to allow a more accurate description of curved boundaries. With this discontinuity option, you can define corners that should not be smoothed by this formulation.

**Keyboard Command Sequence:**

```
add_cbody_dc_nodes <node list> #
```

**Reference:** *MSC.Volume C, Program Input*, Chapter 3: Model Definition Options, SPLINE option.

---

**Command:** **add_contact_body_curves**

**Description:** This command adds curves to the current contact body. This is applicable to rigid and symmetry contact bodies in a 2-D or axisymmetric analysis. For 3-D analysis, these bodies must be defined using surfaces. This button is located in the CONTACT->CONTACT BODIES menu.

**Keyboard Command Sequence:**

```
add_contact_body_curves <curve list> #
```

---

**Command:** **add_contact_body_elements**

**Description:** This command adds elements to the current contact body. The contact body may be either deformable, rigid with heat transfer, or acoustic. This button is located in the CONTACT->CONTACT BODIES menu.

**Keyboard Command Sequence:**

```
add_contact_body_elements <element list> #
```
Command: **add_contact_body_surfaces**

Description: This command adds surfaces to the current contact body. This is applicable to rigid and symmetry contact bodies in a 3-D analysis. For 2-D or axisymmetric analysis, these bodies must be defined using curves. This button is located in the CONTACT->CONTACT BODIES menu.

Keyboard Command Sequence:

```
add_contact_body_surfaces <surface list> #
```

Command: **add_crack_rigreg_nodes**

Description: This command adds nodes to the current rigid region of the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS/3-D CRACKS->MANUAL menu.

Keyboard Command Sequence:

```
add_crack_rigreg_nodes <rigid region> <node list> #
```

Command: **add_csect_elements**

Description: This command adds elements to the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:

```
add_csect_elements <element list> #
```
Command: **add_csect_nodes**  
**Description:** This command adds nodes to the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

**Keyboard Command Sequence:**
```
add_csect_nodes <node list> #
```

Command: **add_curves**  
**Description:** This command adds curves to the current model. The type of curve to be added depends on the current curve type (see set_curve_type). The data that you must enter depends on curve type. This button is located in the MESH GENERATION menu.

Required data by curve type:

- **line**: 2 line endpoints.
- **bezier**: List of control points.
- **cubic spline**: List of vertex points.
- **interpolate**: List of vertex points the curve should interpolate. A cubic NURBS curve is added.
- **nurb**: Number of nurb points, curve order, nurb points, homogeneous coordinates, and knot vector.
- **polyline**: List of vertex points.
- **tangent**: Curve endpoint and distance.
- **fillet**: Two curves and a radius. The head of the first curve is filleted with the tail of the second curve.
- **arc_craa**: Center point, radius, beginning angle, and final angle.
- **arc_cpp**: Center point and 2 endpoints.
- **arc_cpa**: Center point, initial point, and angle.
- **arc_ppp**: 3 points.
- **arc_tra**: Curve tangent endpoint, radius, and angle.
circle_cr  Center point and radius.
circle_cp  Center point and point.
composite  List of connecting curves.
sampled  Origin point, first point, and list of candidate points.

**Keyboard Command Sequence:**

```mcs
add_curves <data> ...
```

---

**Command:** add_elements

**Description:** This command adds elements to the current model. The type of element to be added depends on the current element class (see set_element_class). You must specify the nodes that define the topology of the element. The number of nodes that you must specify depends on the element class. This button is located in the MESH GENERATION menu.

The nodes are entered directly on the command line, by picking them with the mouse or by clicking on a grid point or previously defined point. In the latter two cases, new nodes are created automatically.

**Keyboard Command Sequence:**

```mcs
add_elements <nodes list>
```

---

**Command:** add_geometry_elements

**Description:** This command applies the current geometry property to elements. You must specify a list of elements. This button is located in the GEOMETRIC PROPERTIES->3D and other geometric properties menus.

**Keyboard Command Sequence:**

```mcs
add_geometry_elements <element list> #
```
Command: add_icond_curves
Description: This command adds curves to the current initial condition application. Element faces attached to the curves are associated with the initial condition. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:
add_icond_curves <curve list> #

Command: add_icond_edges
Description: This command adds edges to the current initial condition application. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:
add_icond_edges <edge list> #

Command: add_icond_elements
Description: This command adds elements to the current initial condition application. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:
add_icond_elements <element list> #

Command: add_icond_faces
Description: This command adds faces to the current initial condition application. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.
Keyboard Command Sequence:

```
add_icond_faces <face list> #
```

Command: **add_icond_nodes**

Description: This command adds nodes to the current initial condition application.
This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:

```
add_icond_nodes <node list> #
```

Command: **add_icond_points**

Description: This command adds points to the current initial condition application.
Nodes attached to the points are associated with the current initial condition. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:

```
add_icond_points <point list> #
```

Command: **add_icond_surfaces**

Description: This command adds surfaces to the current initial condition application.
Element faces attached to the surfaces are associated with the initial condition. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:

```
add_icond_surfaces <surface list> #
```
ELEMENT RESULTS

Command: **add_ideas_elem_var**

Description: This command selects element results variables to be written to the ideas results file during a Marc run. You must specify the variables by name.

Keyboard Command Sequence:

```
add_ideas_elem_var <variable name>
```

ELEMENT RESULTS AT NODES

Command: **add_ideas_elnod_var**

Description: This command selects element results variables reported at nodes to be written to the ideas results file during a Marc run. You must specify the variables by name.

Keyboard Command Sequence:

```
add_ideas_elnod_var <variable name>
```

Another Button with the same description:

```
add_ideas_elnod_var
```

NODAL RESULTS

Command: **add_ideas_node_var**

Description: This command selects nodal results variables to be written to the ideas results file during a Marc run. You must specify the variables by name.

Keyboard Command Sequence:

```
add_ideas_node_var <variable name>
```

ADD

Command: **add_insert_embedded**

Description: This command adds embedded elements or embedded nodes to current insert.

Keyboard Command Sequence:

```
add_insert_embedded <elements list> or <nodes list>
```
Command:  **add_insert_host_elements**  
Description:  This command adds host elements to current insert.

**Keyboard Command Sequence:**

```
add_insert_host_elements <elements list>
```

---

Command:  **add_job_applys**  
Description:  This command adds previously defined load applications to the current job. The specified loads are used as initial load conditions for the job. Load applications are specified by name. The text on the button will be the name of the initial condition. This button is located in the JOBS->MECHANICAL->INITIAL LOADS and other analysis class menus.

**Keyboard Command Sequence:**

```
add_job_applys <load apply names>
```

---

Command:  **add_job_careas**  
Description:  This command adds previously defined contact areas to the current job. The specified contact areas are used as initial contact areas for the job. Contact areas are specified by name. The text on the button will be the name of the contact area. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT and other analysis class menus.

**Keyboard Command Sequence:**

```
add_job_careas <contact area names>
```
Command: **add_job_exsegs**

Description: This command adds previously defined exclude segments to the current job. The specified exclude segments are used initially for the job. Exclude segments are specified by name. The text on the button will be the name of the exclude segment. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT and other analysis class menus.

Keyboard Command Sequence:

```
add_job_exsegs <exclude segment names>
```

Command: **add_job_iconds**

Description: This command adds previously defined initial conditions to the current job. The specified conditions are used as initial load conditions for the job. Initial conditions are specified by name. The text on the button will be the name of the initial condition. This button is located in the JOBS->MECHANICAL->INITIAL LOADS and other analysis class menus.

Keyboard Command Sequence:

```
add_job_iconds <initial condition names>
```

Command: **add_job_loadcases**

Description: This command adds previously defined loadcases to the current job. Loadcases are specified by name. The text on the button will be the name of the loadcase. This button is located in the JOBS->MECHANICAL->AVAILABLE and other analysis class menus.

Keyboard Command Sequence:

```
add_job_loadcases <loadcase ids>
```

Another Button with the same description:

```
add_job_loadcases
```
Command: **add_loadcase_adapgs**

Description: This command adds previously defined global remeshing criterion to the current loadcase. Global remeshing criteria are specified by name. The text on the button will be the name of the global remeshing item. This button is located in the LOADCASE->MECHANICAL->STATIC->GLOBAL REMESHING and other loadcase class menus.

**Keyboard Command Sequence:**

```
add_loadcase_adapgs <adapgs name>
```

Command: **add_loadcase_careas**

Description: This command activates contact areas in the current loadcase. Only activated contact areas are considered during analysis. You must specify by name, the contact area to activate. The text on the button will be the name of the contact area. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->CONTACT AREAS and other loadcase class menus.

**Keyboard Command Sequence:**

```
add_loadcase_careas <contact area names>
```

Command: **add_loadcase_cbodies**

Description: This command forces separation in the current loadcase of all nodes contacting this contact body. The release of contact forces can be done immediately at the start of the loadcase or gradually during the loadcase. This is controlled with the menus under FORCE REMOVAL: IMMEDIATE or GRADUAL. Removing the forces gradually can improve convergence of the solution.

Note that releasing a contact body does not prevent nodes from coming in contact with that body again during the loadcase. To prevent the latter, please disable contact with this contact body in the contact table.
To prevent the released contact bodies from contacting again, specify this in the contact table or make sure that the bodies are moved away from each other.

This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->CONTACT BODY RELEASES and other loadcase class menus. The name on the button is the name of the contact body.

**Keyboard Command Sequence:**

```
add_loadcase_cbodies <contact body names>
```

---

**Command:** add_loadcase_exsegs

**Description:** This command activates exclude segments in the current loadcase. Only activated exclude segments are considered during analysis. You must specify by name the exclude segments to activate. The text on the button will be the name of the exclude segment. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->EXCLUDE SEGMENTS and other loadcase class menus.

**Keyboard Command Sequence:**

```
add_loadcase_exsegs <exclude segment names>
```

---

**Command:** add_loadcase_loads

**Description:** This command activates boundary conditions (loads) in the current loadcase. Only activated boundary conditions are considered during analysis. You must specify by name the boundary condition application to activate. The text on the button will be the name of the boundary condition. This button is located in the LOADCASES->MECHANICAL->STATIC->LOADS and other loadcase class menus.

Boundary conditions defined before the creation of a loadcase are automatically activated in that loadcase.

**Keyboard Command Sequence:**

```
add_loadcase_loads <apply name>
```
Command:  add_marc_history_text

Description:  This command allows you to add a line of text that will be placed in the Marc data file at the end of the history definition section. This button is a text box and is located in the LOADCASES->MECHANICAL->ADDITIONAL INPUT FILE TEXT and other loadcase class menus.

The command input consists of the text string that will be written to the Marc data file exactly as it is typed.

Note that if you want the text to have preceding spaces, you will have to enclose the text string in quotation marks, such as “node,1,51,2,0.10035e-05,0.10000e-01”.

Keyboard Command Sequence:

add_marc_history_text <text>

Command:  add_marc_model_text

Description:  This command allows you to add a line of text that will be placed in the Marc data file at the end of the model definition section. This button is a text box and is located in the JOBS->MECHANICAL->ADDITIONAL INPUT FILE TEXT and other analysis class menus.

The command input consists of the text string that will be written to the Marc data file exactly as it is typed.

Note that if you want the text to have preceding spaces, you will have to enclose the text string in quotation marks, such as “node,1,51,2,0.10035e-05,0.10000e-01”.

Keyboard Command Sequence:

add_marc_model_text <text>
Command: \texttt{add_marc_param_text}  
**Description:** This command allows you to add a line of text that will be placed in the Marc data file at the end of the parameter definition section. This button is a text box and is located in the \texttt{JOBS->MECHANICAL->ADDITIONAL INPUT FILE TEXT->PARAMETERS SECTION} and other analysis class menus. 

The command input consists of the text string that will be written to the Marc data file exactly as it is typed.

Note that if you want the text to have preceding spaces, you will have to enclose the text string in quotation marks, such as 

```
"node,1,51,2,0.10035e-05,0.10000e-01"
```

**Keyboard Command Sequence:**

```
add_marc_param_text <text>
```

Command: \texttt{add_material_elements}  
**Description:** This command attaches the current material’s properties to elements. You must specify a list of elements. This button is located in the \texttt{MATERIAL PROPERTIES} menu.

**Keyboard Command Sequence:**

```
add_material_elements <element list> #
```

Command: \texttt{add_node_between}  
**Description:** This command adds a node at the midpoint of two given coordinates points.

**Keyboard Command Sequence:**

```
add_node_between <first x, y, z> <second x, y, z>
```
Command: **add_nodes**

**Description:** This command adds a node at the specified location. This button is located in the MESH GENERATION menu.

- The coordinates can be input either directly on the command line or by clicking on a grid point or a previously defined point or node.
- The coordinates are given in the currently active local coordinate system.

**Keyboard Command Sequence:**

```
add_node <x, y, and z coordinates>
```
Command: **add_normal_faces**

**Description:** This command selects element faces to be used to compute model normals. If no faces are selected, then all external faces will be used. Otherwise, only the faces selected will contribute to the computation. This button is located in the RESULTS->TOOLS menu.

This command is used to prevent normal averaging around sharp edges when post plotting normal or shear quantities. It is appropriate only in three-dimensional models.

**Keyboard Command Sequence:**

    add_normal_faces <element face list>

---

Command: **add_orient_elements**

**Description:** This command applies the current orientation to the specified elements. Materials attached to these elements will be oriented according to the type of the current orientation. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

**Keyboard Command Sequence:**

    add_orient_elements <elements list> #

---

Command: **add_point_between**

**Description:** This command adds a point at the midpoint of two given coordinates points.

**Keyboard Command Sequence:**

    add_point_between <first x, y, z> <second x, y, z>
Command: add_points
Description: This command adds a point at the specified location. This button is located in the MESH GENERATION menu.
The coordinates can be input either directly on the command line or by clicking on a grid point or a previously defined point or node.
The coordinates are given in the currently active local coordinate system.

Keyboard Command Sequence:
add_points <x, y, and z coordinates>

Command: add_post_tensor
Description: This command selects results tensors written to the results file during a Marc run. You must specify the tensors by name. This button is located in the JOBS->MECHANICAL->JOB RESULTS and other analysis class menus.

Keyboard Command Sequence:
add_post_tensor <tensor name>

Command: add_post_var (true command)
remove_post_var (false command)

Description: This command selects results variables written to the results file during a Marc run. You must specify the variables by name. If button is lights after pressing, this is equivalent to add_post_var. If button is lit, pressing the button causes it to turn off, corresponding to remove_post_var, and the variable is taken off the list. This button is located in the JOBS->MECHANICAL->JOB RESULTS and other analysis class menus.

Keyboard Command Sequence:
add_post_var <variable name>
add_rbe2_tied_nodes

Command: **add_rbe2_tied_nodes**
Description: This command adds tied nodes by list of node id’s. This button is located in the LINKS->RBE2’S menu.

The nodes are entered directly on the command line, by picking them with the mouse.

Keyboard Command Sequence:

```
add_rbe2_tied_nodes <node list> #
```

add_rbe3_ret_nodes

Command: **add_rbe3_ret_nodes**
Description: This command adds retained nodes by list of node id’s. This button is located in the LINKS->RBE3’S menu.

The nodes are entered directly on the command line, by picking them with the mouse.

Keyboard Command Sequence:

```
add_rbe3_ret_nodes <node list> #
```

ADD

Command: **add_solids**
Description: This command adds solids to the current model. The type of solid to be added depends on the current surface class (see set_solid_type). The required data varies depending on the solid type. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:

```
add_solids <type specific data>
```
Command: **add_surfaces**

**Description:** This command adds surfaces to the current model. The type of surface to be added depends on the current surface type (see set_surface_type). The data that you must enter depends on the surface type. This button is located in the MESH GENERATION menu.

Required data by surface type:

- **quad**: Four corner points.
- **bezier**: Number of points in \( u \) and \( v \), and a list of points.
- **driven**: Driven curve, driving curve.
- **nurb**: Number of nurb points in \( u \) and \( v \), curve order in \( u \) and \( v \), list of nurb points, homogeneous coordinates, and knot vector.
- **ruled**: Two curves.
- **sphere**: Center point and radius.
- **cylinder**: Axis endpoints and end radii.
- **swept**: Swept curve and sweeping curve.
- **interpolate**: Number of vertex points in \( u \) and \( v \), and a list of points the surface should interpolate. A bicubic NURBS surface is added.
- **coons**: Four bounding curves entered in clockwise or counterclockwise order forming a closed chain. The resulting surface will be twisted if the end point of one curve is not the start point of the next curve. Such curves can be reversed by the command `flip_curves`.
- **skin**: List of curves along a parametric direction. The surface interpolates the series of curves with tangent and curvature continuity. The resulting surface will be twisted if the direction of parameterization in the curves are different.
- **sampled**: Origin point, point in first direction, point in second direction, list of candidate points.

**Keyboard Command Sequence:**

```
add_surfaces <data> ...
```
**ADD**

**Command:** add_transform_nodes

**Description:** This command applies the current transform to the specified nodes. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->TRANSFORMS and other boundary conditions menus.

**Keyboard Command Sequence:**

```
add_transform_nodes <node list> #
```

**SURFACE QUAD/TRI MESH!**

**Command:** af_mapped_quad_trimesh

**Description:** This command automatically meshes the given list of surfaces with a combination of quadrilateral and triangular elements. This is done by meshing in the two-dimensional parametric space of each surface using the Advancing Front algorithm, and then mapping the resulting mesh onto the surface.

Quadrilateral elements will be preferred, but triangular elements are generated wherever quadrilateral elements have a distortion greater than the current maximum allowable quad distortion. The af_set_max_quad_distortion command should be called before this command to change the max quad distortion from the default value.

Each surface must contain closed trimming curves which bound the area to be meshed. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division or three divisions for closed curves.

Also see help: af_set_max_quad_distortion and apply_curve_divisions.

**Keyboard Command Sequence:**

```
af_mapped_quad_trimesh <surface list>
```
Command: af_mapped_quadmesh

Description: This command automatically meshes the given list of surfaces with quadrilateral elements, by meshing in the two-dimensional parametric space of each surface using the Advancing Front algorithm, and then mapping the resulting mesh onto the surface.

Each surface must contain closed trimming curves which bound the area to be meshed. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division or three divisions for closed curves.

In order to guarantee an all-quad mesh, all closed loops should contain an even number of divisions. Triangular elements may be included in the mesh if this condition is not met, or when the mesher would otherwise fail.

Also see help: apply_curve_divisions.

Keyboard Command Sequence:

af_mapped_quadmesh <surface list>

Command: af_mapped_trimesh

Description: This command automatically meshes the given list of surfaces with triangular elements, by meshing in the two-dimensional parametric space of each surface using the Advancing Front algorithm, and then mapping the resulting mesh onto the surface.

Each surface must contain closed trimming curves which bound the area to be meshed. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.
Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division or three divisions for closed curves.

Also see help: apply_curve_divisions.

Keyboard Command Sequence:

```
af_mapped_trimesh <surface list>
```

---

Command: `af_planar_quad_trimesh`

Description: This command automatically meshes the given list of curves with a combination of quadrilateral and triangular elements using the Advancing Front algorithm. Quadrilateral elements will be preferred, but triangular elements are generated wherever quadrilateral elements have a distortion greater than the current maximum allowable quad distortion. The `af_set_max_quad_distortion` command should be called before this command to change the max quad distortion from the default value.

The given curves must form at least one closed loop and lie in a unique plane. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division or three divisions for closed curves.

Also see help: `af_set_max_quad_distortion` and `apply_curve_divisions`.

Keyboard Command Sequence:

```
af_planar_quad_trimesh <curve list>
```
Command:  \texttt{af\_planar\_quadmesh}

Description:  This command automatically meshes the given list of curves with quadrilateral elements using the Advancing Front algorithm. The curves must form at least one closed loop and lie in a unique plane. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division, or three divisions for closed curves.

In order to guarantee an all-quad mesh, all closed loops should contain an even number of divisions. Triangular elements may be included in the mesh if this condition is not met, or when the mesher would otherwise fail.

Also see help: \texttt{apply\_curve\_divisions}.

Keyboard Command Sequence:

\begin{verbatim}
af_planar_quadmesh <curve list>
\end{verbatim}

Command:  \texttt{af\_planar\_trimesh}

Description:  This command automatically meshes the given list of curves with triangular elements using the Advancing Front algorithm. The curves must form at least one closed loop and lie in a unique plane. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have, by default, one division or three divisions for closed curves.
Also see help: apply_curve_divisions.

**Keyboard Command Sequence:**

```
af_planar_trimesh <curve list>
```

**Command:**  
**af_set_max_quad_distortion**

**Description:** Use this command to set the maximum allowed distortion for quadrilateral elements in a quad/tri mixed mesh to be created by the `af_planar_quad_trimesh` command. The parameter entered should be between 0.0 (no distortion allowed; results in an all-triangular mesh) and 1.0 (full distortion allowed; results in an all-quadrilateral mesh). Values entered outside of this range are reverted to the closest limit.

In the `af_planar_quad_trimesh` command, quadrilateral elements are preferred. For this reason, entering a low (< 0.5) value for this parameter is of little practical value. Doing so may cause the mesher to be very slow, and the resulting mesh will likely be largely comprised of triangular elements. To generate an all-triangular mesh, use one of the tri-mesh commands.

**Keyboard Command Sequence:**

```
af_set_max_quad_distortion <max quad distortion>
```

**Command:**  
**alias**

**Description:** Like the UNIX alias command, the MSC.Marc Mentat alias command associates a real command with a character string that represents the command and can be used in its place. You create an alias for a command (or even for an alias) by entering the alias command followed by the existing command or alias, followed in turn by the new alias name. If an alias has already been used, you are prompted to that effect. This button is a text box located in the UTILITIES->ALIASES menu.

**Keyboard Command Sequence:**

```
alias <alias name> <command string>
```
Command: **align_orient**

**Description:** This command sets the type of the current orientation to 3d_aniso and determines the user-defined vectors from the three specified points representing a cartesian coordinate system.

This command should only be used for orientations that are applied to 3-D elements.

**Keyboard Command Sequence:**
```
align_orient <X, Y, Z coordinates of origin>
<X, Y, Z coordinates of point on x-axis>
<X, Y, Z coordinates of point on in x-y plane>
```

---

Command: **align_shells**

**Description:** This command picks one element and makes all consecutively reached elements to have the same node orientation.

**Keyboard Command Sequence:**
```
align_shells < element >
```

---

Command: **align_transform**

**Description:** This command sets up an alignment for future transformations. The alignment you choose will define a vector along which transformations are directed for commands that use alignment.

You must enter an origin point, a point on the x-axis of the transformed system, and a point on the y-axis of the transformed system.

**Keyboard Command Sequence:**
```
align_transform <origin point> <x-axis point> <y-axis point>
```
Command: all_existing
all_selected
all_unselected
all_visible
all_invisible
all_outline
all_surface

Description: These pseudo-commands are shortcuts for providing entity lists to the command line. They may be used whenever a command expects a list of entities. Each pseudo-command generates a list of entities of the requested type and sends the list to the command requesting the list. The types of lists generated are as follows:

- **all_existing**: all entities of the requested type
- **all_selected**: all entities of the requested type that are currently selected
- **all_unselected**: all entities of the requested type that are not currently selected
- **all_visible**: all entities that are visible (see the visible_curves, _elements, _surfaces, and _selected commands)
- **all_invisible**: all entities that are visible (see the invisible_curves, _elements, _surfaces, and _selected commands)
- **all_outline**: all entities of the requested type on the outline of the model
- **all_surface**: all entities of the requested type on the surface of the model

Other Buttons with the same description:

INVIS.
OUTLINE
SELECT.
SURFACE
UNSEL.
VISIBLE
**INTERRUPT**

**Description:** This button interrupts the playing of animation files. Playing may be continued by pressing the RESUME button.

**RESUME**

**Description:** This button continues the playing of animation files. Playing may be interrupted with the INTERRUPT button.

**ANIMATION Menu:**

**Description:** This menu contains commands for creating and playing animation files. Animation files enable you to save a set of graphics display windows and play them back like a movie. This menu is located in the RESULTS-> MORE menu.

**Command:** *animation_index_all*

**Description:** This command toggles animation-play-all-indices. When ON, the animation_play command plays back all the animation files that start with the current animation base filename. When OFF, the animation_play command plays back only those animation files which have indices between the begin and end index values currently set for animation playback (see animation_index_begin and animation_index_end).

In other words, when animation_index_all is ON, all the animation files that have been created with the current animation base filename will be played back, and when animation_index_all is
OFF, only a range of those files will be played back. By default, `animation_index_all` is on.

**Keyboard Command Sequence:**

```
animation_index_all <on or off>
```

Another Button with the same description:

BEGIN TO END

---

**Command:** `animation_index_begin`  
**Description:** This command sets the beginning animation file playback index. This value is valid only if `animation_index_all` is OFF (see `animation_index_all`). When this value is valid, the `animation_play` command plays back a range of animation files that have been created with the current animation base filename, starting with this index.

**Keyboard Command Sequence:**

```
animation_index_begin <begin index>
```

---

**Command:** `animation_index_current`  
**Description:** This command sets the current animation file playback index. This value is valid only if an animation playback sequence has been temporarily interrupted (with a mouse-click or keystroke). When animation is resumed, it starts animating at this index.

**Keyboard Command Sequence:**

```
animation_index_current <current index>
```
Command: **animation_index_end**

Description: This command sets the end of the animation file playback index. This value is valid only if **animation_index_all** is OFF (see **animation_index_all**). When this value is valid, the **animation_play** command plays back a range of animation files that have been created with the current animation base filename, ending with this index.

**Keyboard Command Sequence:**

```
animation_index_end <end index>
```
Command:  **animation_name**

Description:  This command sets the animation base filename. The base filename is used by both the **animation_save** and the **animation_play** commands. The save or current index is appended to the base filename to create the actual filename for saving or playing back display lists. By default, **animation_name** is set to the base file name of the post file.

Keyboard Command Sequence:

```
animation_name <filename>
```

---

Command:  **animation_pause**

Description:  This command sets the number of seconds to pause between frames during animation playback. By default, **animation_pause** is zero.

Keyboard Command Sequence:

```
animation_pause <seconds>
```

Another Button with the same description:

```
PAUSE
```

---

Command:  **animation_play**

Description:  This command plays back display lists which have previously been saved with the **animation_save** command.

Keyboard Command Sequence:

```
animation_play
```
Command: **animation_repeat**

Description: This command toggles animation repeat mode. When OFF, the `animation_play` command plays, just once, the animation files that start with the current animation base filename. When ON, the `animation_play` command plays back those files over and over again. Either way, playback can be interrupted with a mouse-click or keystroke. By default, `animation_repeat` is OFF.

Keyboard Command Sequence:

```
animation_repeat <on or off>
```

Another Button with the same description:

SINGLE PLAY

---

Command: **animation_reverse**

Description: This command toggles animation reverse mode. When OFF, the `animation_play` command plays back, in ascending order, the animation files that start with the current animation base filename. When ON, the `animation_play` command plays back, in ascending order, those files. By default, `animation_reverse` is OFF.

Keyboard Command Sequence:

```
animation_reverse <on or off>
```

Another Button with the same description:

REVERSE
Command:  animation_save
Description: This command saves the current display lists to a disk file. The name of that file is created by appending the next animation file save index (see animation_index_save) onto the animation base filename (see animation_name). After saving the display lists, the animation file save index is automatically incremented. The saved lists can then be viewed sequentially by using the animation_play command.

Keyboard Command Sequence:
animation_save

Command:  animation_terminate
Description: This command terminates animation playback. Animation playback is terminated after the next frame is drawn. Animation playback is automatically terminated when animation repeat mode is OFF, and the playback finishes displaying the last saved frame.

Keyboard Command Sequence:
animation_terminate

Command:  append_procedure
Description: This command appends command lines to an existing procedure file. You must specify the name of the procedure file you wish to append to.

Keyboard Command Sequence:
append_procedure <file name>
Menu: BOUNDARY CONDITIONS
Description: The BOUNDARY CONDITIONS menu contains commands that apply boundary conditions to the mesh. Boundary conditions may be applied for the following analysis types: mechanical, heat transfer, acoustic, rigid/plastic, bearing, joule heating, electrostatic, magnetostatic, and electromagnetic. The types of boundary conditions that can be applied depend on the type of analysis. Example boundary conditions include fixed displacements, loads, pressures, temperatures, heat fluxes, velocities, voltages, currents, potentials, and charges.

Menu: ACOUSTIC
Description: The ACOUSTIC submenu within the BOUNDARY CONDITIONS menu contains commands for setting acoustic boundary condition types and application values. In an acoustic analysis, prescribed pressures can be defined and sources applied. Note that an unconstrained boundary represents a rigid reflecting surface. At least one pressure must be applied to at least one node to prevent a singular solution.

Note: If the eigenmodes of an unconstrained cavity are required, you must request the solution of the singular system.

Menu: ALL
Command: apply_all_layers
Description: This command activates all element layers for the current state variable application.
Keyboard Command Sequence: apply_all_layers
BEARING

Menu: BEARING

Description: The BEARING submenu within the BOUNDARY CONDITIONS menu contains commands for setting bearing boundary condition types and application values. In a hydrodynamic bearing analysis, the lubrication pressure can be defined and mass flow rates be applied. An unconstrained boundary represents no mass flow across that boundary. To ensure a nonsingular solution, a pressure must be defined for at least one point or the restrictor option be used.

APPLY CURVE DIVISIONS

Command: apply_curve_divisions

Description: This command will apply curve divisions to chosen curves, in accordance with the current specified curve division type and associated parameters, as well as the specified division restriction.

Also see help: curve_div_type and curve_div_rest

Keyboard Command Sequence:

apply_curve_divisions <curve list>

DISPLACEMENT X

Command: apply_dof
clear_apply_dof

Description: These commands add and clear the indicated degree of freedom from the current boundary condition application. The button text will be the name of the boundary condition, such as FLUX for Joule heating, PRESSURE for acoustic, etc.

Keyboard Command Sequence:

apply_dof <dof name>
clear_apply_dof <dof name>
**ELECTROMAGNETIC**

**Menu:** ELECTROMAGNETIC  
**Description:** The ELECTROMAGNETIC submenu within the BOUNDARY CONDITIONS menu contains commands for setting electromagnetic boundary condition types and application values. In electromagnetic analysis, the potential can be defined and charges and/or currents may be applied. In a harmonic analysis, both the magnitude and phase are required.

**ELECTROSTATIC**

**Menu:** ELECTROSTATIC  
**Description:** The ELECTROSTATIC submenu within the BOUNDARY CONDITIONS menu contains commands for setting electrostatic boundary condition types and application values. In an electrostatic analysis, the scalar potential may be defined and charges may be applied. It is necessary to define the scalar potential by at least one node to ensure a nonsingular system.

**JOULE**

**Menu:** JOULE  
**Description:** The JOULE submenu within the BOUNDARY CONDITIONS menu contains commands for setting joule boundary condition types and application values. In a Joule heating analysis, besides the conventional heat transfer boundary conditions, you can define voltages and apply currents. At least one node must have a constrained voltage to ensure that a nonsingular system occurs and a solution is obtainable.
LAYERS

Command: **apply_layers**

Description: This command activates element layers for the current state variable application. Element layers are specified as integers beginning at 1.

Keyboard Command Sequence:

```
apply_layers <layer numbers>
```

MAGNETOSTATIC

Menu: **MAGNETOSTATIC**

Description: The MAGNETOSTATIC submenu within the BOUNDARY CONDITIONS menu contains commands for setting magnetostatic boundary condition types and application values. In magnetostatic analysis, you can define the vector potential at nodes and apply currents. The potential should be applied at one point to ensure a nonsingular problem.

MECHANICAL

Menu: **MECHANICAL**

Description: The MECHANICAL submenu within the BOUNDARY CONDITIONS menu contains commands for setting mechanical boundary condition types and application values.

NAME

Command: **apply_name**

Description: This command sets or changes the name of the current application. A boundary condition application is a set of information that completely defines a boundary condition. Each application contains the type of boundary condition, the degrees of freedom and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected. Multiple applications may be defined and are stored in the list of currently defined applications.

Keyboard Command Sequence:

```
apply_name <application name>
```
Command: apply_option

Description: This command sets the value of a boundary condition option. You must specify the option to be set. Boundary condition options are specified as the name of the option and its value separated by a colon (e.g. bin_post_file:ieee, for post file format of IEEE).

The valid boundary condition options and their values are as follows:

- **bin_post_file** native, ieee
  - **NATIVE** (value `native`) specifies that the format of the post file is either native binary format, where or `IEEE` (value `ieee`) is for Cray only.

- **cavity_method** general, closed
  - The CLOSED CAVITY button (value `closed`) specifies that the cavity is closed and the gass mass is preserved. The pressure is automatically updated with volume changes. Otherwise the value is `general` and it allows for the use of the boundary condition option `dof_values` for ENTERED (entered), or user subroutine USER SUB. UCAV.

- **coriolis** on, off
  - This toggle specifies that the Coriolis effect should be used for the Centrifugal load.

- **dof_values** entered, post_file, process_cntrl, usersub
  - Specifies where the data is obtained.
    - If the option ENTERED (`entered`) is specified, then the data is entered manually.
    - If the option POST FILE (`post_file`) is used, the data is obtained from a previously generated post file.
    - The option SUPERPLASTICITY CONTROL (`process_cntrl`) specifies that superplasticity control should be used.
    - If the option USER SUB. (`usersub`) is specified, then the data is obtained from a user subroutine.
edge_load_mode  area, length  The FORCE / UNIT AREA button (area) indicates that the edge load is specified by unit area, the FORCE / UNIT LENGTH button (length) indicates that the load is by unit length.

harmonic_mode  magn_phase, real_imag  Specifies the input mode. Use MAGNITUDE & PHASE (magn_phase) for Magnitude and Phase values, or REAL & IMAGINARY (real_imag) for Real and Imaginary values.

Keyboard Command Sequence:
icond_option <icond option>:<value>

Other Buttons with the same description:
- CAVITY CLOSED
- ENTERED VALUES
- FORCE / UNIT AREA
- FORCE / UNIT LENGTH
- IEEE
- INCREMENT
- LAST
- POST FILE
- MAGNITUDE & PHASE
- REAL & IMAGINARY
- SUPERPLASTICITY CONTROL
- TIME
- USER SUB. UCAV

Command: apply_post_file
Description: This command specifies the name of the post file to be used for reading values for the current state variable application.

Keyboard Command Sequence:
apply_post_file <post file name>
**FIRST INCREMENT**

**Command:** \texttt{apply\_post\_increment}  
**Description:** This command specifies the first increment to be read from the post file to provide values for the state variable application.  
**Keyboard Command Sequence:**  
\texttt{apply\_post\_increment <increment number>}

Another Button with the same description:  
\texttt{INCREMENT}

**# STEPS**

**Command:** \texttt{apply\_post\_steps}  
**Description:** This command specifies the number of increments (steps) to be read from the post file to provide values for the state variable application.  
**Keyboard Command Sequence:**  
\texttt{apply\_post\_steps <number of increments>}

**RADIATION**

**Menu:** RADIATION  
**Description:** Radiation Boundary Conditions may be applied to element edges and faces. The emission of radiation may be toggled ON or OFF, and the temperature at infinity may be set. Shells may emit from either the top or bottom while continuum elements emit only from the top or outer direction. If neither side emits, then the elements only shadow other elements.  
The radiation boundary conditions are used in the viewfactor computation. Emissivities are set in the material properties for heat transfer.
TABLE

Command: apply_table

Description: This command selects a table to be applied to a boundary condition. You must specify a slot name and a previously defined table by name. Slot names are specified by a quantity name appended with a slot number (for example, \( x0 \)).

A table is a function of one variable, i.e. \( y = f(x) \), that is applied to the corresponding application value. The table is commonly a function of a variable such as time, frequency, temperature, position. These tables are used by the analysis program to define the spatial or temporal variation of a quantity.

Keyboard Command Sequence:

\[
\text{apply_table} \ <\text{quantity name}> <\text{slot number}> <\text{table name}>
\]

THERMAL

Menu: THERMAL

Description: The THERMAL submenu within the BOUNDARY CONDITIONS menu contains commands for setting thermal boundary condition types and application values. In heat transfer analyses, you can define a fixed temperature to a node and apply fluxes to the surface or volume. The film boundary conditions allow the definition of corrective boundary conditions. A prescribed temperature or film condition should be applied somewhere in the model to ensure a nonsingular system if a steady state analysis is performed.

EDGE CURRENT

Command: apply_type

Description: This command selects the boundary condition type of the current application. If the current application does not have any values set then this command simply changes the boundary condition type. If the current application has values set, then this command creates a new application in addition to setting the type. (This feature saves users from having to
manually creating new boundary condition applications using the new_apply command.)

**Mechanical boundary condition types:**

- fixed_displacement
- point_load
- edge_load
- face_load
- global_load
- gravity_load
- centrifugal_load
- edge_foundation
- face_foundation
- initial_displacement
- initial_velocity
- point_mass
- cavity_pressure_load
- cavity_mass_load

**Thermal boundary condition types:**

- fixed_temperature
- point_flux
- edge_flux
- face_flux
- volume_flux
- edge_film
- face_film
- initial_temperature
- plastic_heat_generation

**Acoustic boundary condition types:**

- fixed_pressure
- point_source
- edge_source
- face_source
- volume_source

**Rigid/Plastic boundary condition types:**

- fixed_velocity
- point_load
- edge_load
- face_load
- global_load
- gravity_load

**Bearing boundary condition types:**

- fixed_velocity
- restrictor
- point_mass_flow_rate
- edge_mass_flow_rate

**Joule heating boundary condition types:**

- fixed_temperature
- fixed_voltage
- point_flux
- edge_flux
- face_flux
- volume_flux
- point_current
- edge_current
- face_current
- volume_current

**Electrostatic boundary condition types:**

- fixed_potential
- point_charge
- edge_charge
- face_charge
- volume_charge

**Magnetostatic boundary condition types:**

- fixed_potential
- point_current
- edge_current
- face_current
- volume_current

**Electromagnetic boundary condition types:**

- fixed_potential
- point_current
- edge_current
- face_current
- volume_current
- point_charge
- edge_charge
- face_charge
- volume_charge
Keyboard Command Sequence:

apply_type <boundary condition type>

Other Buttons with the same description:
GLOBAL LOAD
GRAVITY LOAD
OK
PERMANENT MAGNETIZATION
VOLUME CHARGE
VOLUME CURRENT

Command: apply_value

Description: This command sets the value for a particular quantity in the current application. You must specify the quantity name and its value.

Keyboard Command Sequence:

apply_value <quantity or dof name> <value>

Other Buttons with the same description:
GRAVITY CONSTANT
INERTIA COEFFICIENT
INNER MASS DENSITY
INNER SURFACE ELEV
NEGATIVE
OUTER MASS DENSITY
OUTER SURFACE ELEV
POSITIVE
SEA BED ELEVATION
VX
VY
VZ
WAVE DIRECTION COS 1
WAVE DIRECTION COS 2
WAVE HEIGHT
WAVE PERIOD
WAVE PHASE
X CURRENT
X GRADIENT
X1
X2
Y CURRENT
Y GRADIENT
Y1
Y2
Z CURRENT
Z GRADIENT
Z1
Z2
**STATE VARIABLE ID**

Command: **apply_variable**  
Description: This command specifies the identifier of the state variable for the current state variable application. Temperature is always the first state variable at a point.  

**Keyboard Command Sequence:**  
apply_variable <state variable id>

**# FACETS**

Command: **arrow_facets**  
Description: This command allows you to set the number of circular facets used when pre- and postprocessing arrows are plotted in solid mode. Also, be sure to use the arrows_solid command to select solid plotting of arrows (wireframe plotting is the default). The default number of facets is eight. Also see arrows_solid and set_arrow_lines.

**Keyboard Command Sequence:**  
arrow_facets <# of facets>

**LENGTH**

Command: **arrow_length**  
Description: This command allows you to set the absolute length of preprocessing arrows. Be sure to use the set_auto_arrow command to turn off the auto-arrow-length mode which by default is ON. The default arrow length (when this mode is off) is 0.2. Also see set_auto_arrow and auto_arrow_length.

**Keyboard Command Sequence:**  
arrow_length <length>
Command: \texttt{arrows\_wireframe}  \texttt{arrows\_solid}

Description: These commands allow you to select the overall plot style of pre- and postprocessing arrows to be either wireframe or solid. Wireframe plotting is the default. If solid plotting is selected, you may also set the number of circular facets used to draw each arrow, and you may select whether or not the edges of the facets should be drawn.

Also see \texttt{arrow\_facets} and \texttt{set\_arrow\_lines}.

Keyboard Command Sequence:
\begin{verbatim}
arrows\_wireframe
arrows\_solid
\end{verbatim}

Another Button with the same description:
\begin{verbatim}
WIREFRAME
\end{verbatim}

Command: \texttt{boyce\_1993}

Description: Elastomer Free Energy Function

Number of coefficients: 1


Parameter:
N – The number of mers in a typical polymer chain Specified by the user.

Coefficient:
nkT – Determined by the fitter
n: Number of Polymer chains per unit volume
k: Boltzmann constant
T: Temperature

For this model, the calculation of the Bulk Modulus is not required by the MSC.Marc program. A volumetric test do not need to be supplied.
Subroutine: ASSOC

Description: The user subroutine ASSOC allows you to define the flow direction for plasticity when used in conjunction with the Generalized Plasticity option. The default is the associated flow law with the von Mises (J2) yield surface. ASSOC is used in conjunction with user subroutines SINCER, YIEL, and ZERO.

Menu: ATTACH

Description: This menu contains commands for associating a mesh with the underlying geometry. The following associations can be made:

- a node can be attached to a point
- an element edge can be attached to a curve
- an element face can be attached to a surface.

In addition to the commands in this menu, the automatic mesh generators in the AUTOMESH menu and the conversion operations from GEOMETRY to MESH in the CONVERT menu, automatically attach the mesh to the geometry.

Nodes which are attached to a point always have the same position as the point. Nodes of edges which are attached to a curve always lie on that curve and nodes of faces which are attached to a surface always lie on that surface. Note that this implies that the common node of two edges which are attached to different curves must lie on the intersection of the curves. Similarly, the common nodes of two faces which are attached to different surfaces must lie on the intersection of the two surfaces. The automatic mesh generators and the mesh generation commands that modify either the mesh or the geometry guarantee that this is always the case. For example, if one of the curves is moved or otherwise changed, the common node is repositioned automatically to the new point of intersection. If that point cannot be found, the operation is not permitted and an error message is issued.
Mesh entities which are attached to geometric entities inherit any initial or boundary conditions applied to the geometric entities, according to the following rules:

- A node which is attached to a point inherits any initial or boundary condition applied to that point.
- The nodes of an edge attached to a curve and the nodes of a face attached to a surface inherit any nodal initial or boundary conditions applied to the curve or surface.
- An edge attached to a curve inherits any edge-type initial or boundary condition applied to that curve.
- A face attached to a surface inherits any face-type initial or boundary condition applied to that surface.

**EDGES -> CURVE**

**Command:** attach_edges_curve

**Description:** This command moves the nodes of a list of element edges to a curve and establishes an association between the edges and the curve. The nodes are moved according to the current attach mode.

If an edge is attached to a curve, any initial or boundary condition applied to the curve will be inherited by the edge or by the nodes of the edge, depending on the type of condition. Moreover, if the curve is moved or otherwise changed, the nodes of the edge are moved along with the curve.

Attached element edges are drawn using color indices 208 (opaque) and 210 (translucent), instead of the normal edge color. These colors can be changed via the VISUALIZATION->COLORS menu. The default color is orange. The curve to which an edge is attached can be displayed using the set_edge_attach_labels command.

The association between the edge and the curve can be removed with the detach_edges command.

Also see: detach_edges and set_edge_attach_labels.

**Keyboard Command Sequence:**

```
attach_edges_curve <curve> <edge list> #
```
ELEMENTS -> CURVE

Command: **attach_elements_curve**

Description: This command attaches the first edge (edge 0) of a list of elements to a curve. Only the first edge of the line elements in the list will be attached to the curve. Elements of other classes (if any) are disregarded.

Also see: **attach_edges_curve**.

Keyboard Command Sequence:

```
attach_elements_curve <curve> <element list> #
```

ELEMENTS -> SURFACE

Command: **attach_elements_surface**

Description: This command attaches the first face (face 0) of a list of elements to a surface. Only the first face of the triangular and quadrilateral elements in the list will be attached to the surface. Elements of other classes (if any) are disregarded.

Also see: **attach_faces_surface**.

Keyboard Command Sequence:

```
attach_elements_surface <surface> <element list> #
```

FACES -> SURFACE

Command: **attach_faces_surface**

Description: This command moves the nodes of a list of element faces to a surface and establishes an association between the faces and the surface. The nodes are moved according to the current attach mode.

If a face is attached to a surface, any initial or boundary condition applied to the surface will be inherited by the face or by the nodes of the face, depending on the type of condition. Moreover, if the surface is moved or otherwise changed, the nodes of the face are moved along with the surface.

Attached element faces are drawn using color indices 209 (opaque) and 211 (translucent), instead of the normal face color. These colors can be
changed via the VISUALIZATION->COLORS menu. The default color is dark blue. The surface to which a face is attached can be displayed using the set_face_attach_labels command.

The association between the face and the surface can be removed with the detach_faces command.

Also see: detach_faces and set_face_attach_labels.

Keyboard Command Sequence:
attach_faces_surface <surface> <face list> #

Command: attach_mode_closest
Description: This command switches the attach mode to "closest". In this mode, the commands:
- attach_edges_curve
- attach_faces_surface
- attach_elements_curve
- attach_elements_surface

move the nodes of the edges or faces that must be attached to the point on the curve or surface closest to their original position.

Also see: attach_mode_directed, attach_edges_curve, attach_faces_surface, attach_elements_curve, and attach_elements_surface

Keyboard Command Sequence:
attach_mode_closest
Command: **attach_mode_directed**

**Description:** This command switches the attach mode to "directed". In this mode, the commands:

- `attach_edges_curve`
- `attach_faces_surface`
- `attach_elements_curve`
- `attach_elements_surface`

move the nodes of the edges or faces that must be attached in a particular direction to the curve or surface. The direction can be set by the `set_attach_direction` command.

Also see: `attach_mode_closest`, `set_attach_direction`, `attach_edges_curve`, `attach_faces_surface`, `attach_elements_curve`, and `attach_elements_surface`.

**Keyboard Command Sequence:**

```
attach_mode_directed
```
Command: **attach_nodes_curve**

Description: This command establishes an association between the nodes of a mesh and the underlying curves. Nodes and elements are not automatically associated with the curves and curves used to create them unless the mesh is generated with the AUTOMESH processor. Once they are attached, either manually or with AUTOMESH, they can only be separated with the detach_nodes command.

Keyboard Command Sequence:

```
attach_nodes_curve <curve list> 
```
**FACTOR**

**Command:**  auto-arrow_length

**Description:** This command allows you to set the length of preprocessing arrows, relative to the overall size of the model. The default auto-arrow-length mode should be on (but can be turned off with the set_auto_arrow command). The default arrow length fraction is 0.1.

Also see set_auto_arrow and arrow_length.

**Keyboard Command Sequence:**

auto_arrow_length <fraction>

---

**AUTOMESH**

**Menu:**  AUTOMESH

**Description:** This menu contains commands used to create a mesh from existing geometry. The geometry can be in the form of planar curves, trimmed surfaces, or surface elements (for generating solid elements).

---

**2D PLANAR MESHING**

**Menu:**  AUTOMESH 2D PLANAR

**Description:** This menu contains commands to create a mesh from existing 2-D geometry. The area to be meshed must be bounded by closed sets of curves.

---

**2D REBAR MESHING**

**Menu:**  AUTOMESH 2D REBAR

**Description:** This menu contains commands to create a mesh from existing 2-D mesh and existing curves. The curves are meshed between intersections with 2-D mesh.
SURFACE MESHING

Menu: AUTOMESH SURFACES

Description: This menu contains commands to create a mesh from existing surface geometry. The surfaces to be meshed must contain trimming curves.

MAKE AVI MOVIE

Command: avi_animation_make

Description: This command MAKE AVI MOVIE will generate an AVI movie file. It is only available on Windows NT.

This command shares many of the settings with the MPEG movie generation command. It will use the values displayed under INCREMENT SETTINGS, FIRST (movie_first_increment), LAST (movie_last_increment), STEP (movie_step_increment) and VIEW (movie_view).

The view will be specified in the VIEW display and must be the current view.

The AVI movie is generated in a 3 step process:

1. The animation files are automatically generated which is typically performed manually with the ANIMATION>CREATE>INCREMENTS button (the post_animate_increments command).

2. Once the animation files are created, you are prompted to select the compression type for the AVI file if the COMPRESSION DIALOG button (command movie_avi_compress) is on. If it is not on, then no compression is used, however, the file will be very large.

It is recommended that you use Microsoft Video 1 compression.

3. The animation is played and screen images are captured and written to the AVI file.

Note: This command will remove all animation display list files that begin with the BASE FILE name (*animation_name) before it starts, unless the command movie_gen_files GENERATE ANIMATION FILES is off.

You can interrupt the process of making the AVI file by pressing the Escape key, however, you will be returned to the animation play mode.
In this case, you will need to run the `show_model` command in the ANIMATION menu to display the model again.

**Note:** To generate a movie size equal to 640 x 480, you should use the startup option `-sz 856 577`. This provides the optimal image size for AVI movies.

**Keyboard Command Sequence:**

```plaintext
avi_animation_make
```

---

**Command:** `icond_type`

**Description:** This command defines control parameters for data transfer and mesh expansion from axisymmetric analysis to 3-D analysis. The button text will be the type of initial condition, such as VELOCITY, POINT MASS, etc.

**Keyboard Command Sequence:**

```plaintext
icond_type <boundary condition type>
```
Mentat Help Commands in B

**SCALE FACTOR**

**Command:** `beam_3d_scale`  
**Description:** This command sets the scale factor used when drawing 3-D general beams with cross-sections shown extended to three dimensions. This scale affects only the size of the beam cross-section as it is seen in three dimensions, and it does not affect the perceived length of the beam.  
Also see help: `draw_beams_3d` and `geometry_value`.  
**Keyboard Command Sequence:**  
`beam_3d_scale <scale factor>`
# ARC DIVISIONS

**Command:**  `beam_arc_divisions`

**Description:** This command sets the number of divisions which will be used to draw arcs and circular cross-sections when drawing 3-D general beams with the beam cross-sections shown extended to three dimensions. A higher number results in the beams appearing smoother, while a lower number will enable the beams to be drawn faster.

Also see help: `draw_beams_3d` and `geometry_value`.

**Keyboard Command Sequence:**

```
beam_arc_divisions <number of divisions>
```

## AXIS LENGTH FACTOR

**Command:**  `beam_axis_factor`

**Description:** This command sets the length factor for drawing X-Y axes on 3-D beams. The length of the axes drawn on each beam is the length of the beam multiplied by the length factor. The axes will only be drawn if the `draw_beam_axes` option is on.

Also see help: `draw_beam_axes`.

**Keyboard Command Sequence:**

```
beam_axis_factor <factor>
```
**AXIAL FORCE**

**Command:**
- post_beam_axial_force
- post_beam_moment
- post_beam_shear
- post_beam_torque
- post_beam_bimoment

**Description:** Beam Diagrams
These commands deal with the drawing of beam diagrams (shear, axial force, moment, etc.) on the beam elements of the model during postprocessing. These values are plotted using the direction of the beam element as the local z-axis. The local x-axis direction must have been specified for the element geometry and be available for postprocessing via post codes 261-263 (beam_orient in the JOB RESULTS menu).

Plotting of any of the beam diagram values is activated by the post_beam_shear, post_beam_moment, and commands mentioned above. Beam diagram values must be available for postprocessing via post codes 264-270 (bm_axi_for ... bm_bi_mom in the JOB RESULTS menu).

Also see help: post_off and draw_beam_axes.

**Keyboard Command Sequence:**
- post_beam_axial_force
- post_beam_moment
- post_beam_shear
- post_beam_torque
- post_beam_bimoment

Other Buttons with the same description:
- BEND. MOMENT
- BIMOMENT
- SHEAR FORCE
- TORS. MOMENT
Command:  
  \texttt{post\_beam\_draw\_x}  
  \texttt{post\_beam\_draw\_y}  
  \texttt{post\_beam\_filled}  
  \texttt{post\_beam\_factor}  

Description:  
Beam Diagrams Adjustment  
The values which are plotted in the local-xz plane (Vxz, Myy) and those  
which are plotted in the local-yz plane (Vyz, Mxx) may be independently  
toggled on and off by the \texttt{post\_beam\_draw\_x} and  
\texttt{post\_beam\_draw\_y} commands.  

The \texttt{post\_beam\_filled} command toggles filled/wireframe mode for  
the beam diagrams. Color contours representing the shear or moment  
values are drawn in the fill area or along the connecting line depending on  
whether filled or wireframe mode is used.  

The \texttt{post\_beam\_factor} command may be used to control the scale  
factor for drawing the values. First the maximum value/length ratio for all  
the beam elements is found, and then it is multiplied by this scale factor to  
determine the actual factor used in plotting the values.  

Also see: \texttt{post\_off} and \texttt{draw\_beam\_axes}.  

Keyboard Command Sequence:  
  \texttt{post\_beam\_draw\_x <on or off>}  
  \texttt{post\_beam\_draw\_y <on or off>}  
  \texttt{post\_beam\_filled <on or off>}  
  \texttt{post\_beam\_factor <factor>}  

Other Buttons with the same description:  
  LOCAL XZ (Vxz, Myy)  
  LOCAL YZ (Vyz, Mxx)  
  SCALE FACTOR
**BRANCH ID’s**

**Command:** `branch_labels`

**Description:** This command is used to specify whether or not beam section branches will be labeled.

**Keyboard Command Sequence:**

```
branch_labels <on or off>
```

**BREAK CURVES**

**Command:** `break_curves`

**Description:** This command splits at the endpoints of all given curves. This can be used to assure that nodes can be aligned during automatic mesh generation. Tolerance is used to indicate the distance at which two points should be considered to be equal. The tolerance value is computed by multiplying the relative tolerance value by the curve length. This command should be used before the `match_curves` command. This is available only in Mentat and not the SuperForm GUI.

**Keyboard Command Sequence:**

```
break_curves <curve_list>
```

**BREAK CURVES**

**Command:** `break_curves_af`

**Description:** This command splits at the endpoints of all given curves. This can be used to assure that nodes can be aligned during automatic mesh generation. The given tolerance indicates the distance at which two points should be considered to be equal. This command should be used before the `match_curves` command. This is available only in the SuperForm GUI and not in Mentat.

**Keyboard Command Sequence:**

```
break_curves_af <tolerance> <curve_list>
```
Mentat Help Commands in B MSC.Mentat Help Reference

Command: bsect_grid
  bsect_grid_rectangular
  bsect_grid_cylindrical
  bsect_grid_spherical
  bsect_grid_u_domain
  bsect_grid_v_domain
  bsect_grid_u_spacing
  bsect_grid_v_spacing
  bsect_grid_style_dots
  bsect_grid_style_lines
  bsect_grid_maxpoints

Description: These commands manipulate the grid used for creating beam sections. The grid is useful for designating coordinates for branch endpoints.

The bsect_grid command specifies whether or not the beam section grid is displayed.

The commands bsect_grid_rectangular and bsect_grid_cylindrical are used to select a rectangular or cylindrical grid.

The commands bsect_grid_u_domain and bsect_grid_v_domain are used to specify the U and V domains of the grid.

The commands bsect_grid_u_spacing and bsect_grid_v_spacing are used to specify the grid intervals in the respective directions.

The commands bsect_grid_style_dots and bsect_grid_style_lines specify the grid style.

The command bsect_grid_maxpoints specifies the upper limit on grid points (or grid line intersections) to be displayed.

Keyboard Command Sequence:

bsect_grid <on or off>
bsect_grid_rectangular
bsect_grid_cylindrical
bsect_grid_u_domain <min u value> <max u value>
bsect_grid_v_domain <min v value> <max v value>

bsect_grid_u_spacing <u spacing>
bsect_grid_v_spacing <v spacing>

bsect_grid_style_dots
bsect_grid_style_lines

bsect_grid_maxpoints <max # points>

Other Buttons with the same description:

CYLINDRICAL
DOTS
LINES
MAX POINTS
ON
RECTANGULAR
SET
U SPACING
V DOMAIN
V SPACING
Mentat Help Commands in C

**CALCULATE**

Command: **calculator**

Description: While you are working within MSC.Marc Mentat, you can invoke a calculator that will accept Fortran-like expressions and evaluate them for you. You can use parameters you have defined during your session in your expressions and you can store the results of expressions in variables that can be used elsewhere in your work within MSC.Marc Mentat.

Keyboard Command Sequence:

```
calculator
```
In general, standard structural elements are used to define the boundaries of cavities and no extra elements are needed. However, to model the boundaries of cavities in regions where standard finite elements are not present, e.g. along rigid boundaries, cavity surface elements (elements 171-174) can be used. These elements can also be glued to moving rigid surfaces. They are for volume calculation purposes only and do not contribute to the stiffness equations of the model. Care must be taken that element orientations must be consistent for all elements forming the cavity. This means that orientations of some elements might need to be flipped in order to get the cavity pressure pointing in the same direction.

The cavity is assumed to contain an ideal gas. The ideal gas constant is defined using some reference properties (pressure, temperature, and density) at an arbitrary state of the gas. The cavity pressure is updated according to the ideal gas relation. The ideal gas temperature is assumed to be the average temperature of the elements forming the cavity. In case of no thermal loading, the gas reference temperature is used instead. A user subroutine can be used to update the pressure in any other way.

Two types of cavity loadings are identified under the CAVITY PRESSURE LOAD and CAVITY MASS LOAD boundary condition menus:

- **Closed cavities**: The gas mass is preserved. The pressure is automatically updated with volume changes.
- **Open cavities**: The cavity can be loaded with an applied pressure and/or mass.

This menu contains commands for adding and removing cavity edges, faces, curves and surfaces. The curves and surfaces are actually added to the cavity definition only if they are attached to elements.

The menu also contains commands for defining the reference properties of the gas inside the cavity.

It is located under the MESH GENERATION menu.
**NAME**

Command: `cavity_name`

Description: This command sets or changes the name of the current cavity. It is located under the MESH GENERATION->CAVITIES menu.

Keyboard Command Sequence:
```
cavity_name <name>
```

**REF. PRESSURE**

Command: `cavity_param ref_pressure/ref_temperature/ref_density`

Description: This command is used to define the reference properties (pressure, absolute temperature, and density) of the gas inside the cavity. The reference gas properties should correspond to an arbitrary state of the gas. The references properties are used to calculate the gas constant according to the ideal gas relation. The gas constant is needed to update the cavity pressure when the cavity is closed. The reference density is also used to calculate the initial mass of the gas for permanently closed cavities. The reference parameters should be nonzero. It is located under the MESH GENERATION->CAVITIES (PARAMETERS) menu.

Keyboard Command Sequence:
```
cavity_param <ref_pressure/ref_temperature/ref_density> <value>
```

Other Buttons with the same description:
- REF. TEMPERATURE
- REF. DENSITY

**CONTROL NODE**

Command: `cbody_control_node`

Description: This command defines the control node for a load-controlled rigid body. This node is associated with the load-controlled rigid body. Forces and prescribed displacements can be applied to this node, which can be a free node not connected to the mesh or a node of an existing mesh.
For 2-D, this node has two translational degrees of freedom and for 3-D, it has three degrees of freedom. If rotations and/or moments are to be applied to the load-controlled rigid body, they are applied to the auxiliary node which can also be associated with the rigid body. That node has one rotational degree of freedom for 2-D and three for 3-D. Note that these degrees of freedom are referred to as X DISPLACEMENT, etc. in the boundary conditions menu. If no auxiliary node is given, the rotation of the rigid body is suppressed.

The coordinates of the control node are used as the center of rotation of the rigid body. Rotations applied to the auxiliary node are with respect to this point.

Keyboard Command Sequence:
```
cbody_control_node <node>
```

Command:  `cbody_control_node_rot`

Description: This command defines the auxiliary node for a load-controlled rigid body. This node is associated with the load-controlled rigid body. Moments and prescribed rotations can be applied to this node, which should not be part of the rest of the mesh.

This node has one rotational degree of freedom for 2-D and three for 3-D. Note that these degrees of freedom are referred to as X DISPLACEMENT, etc. in the boundary conditions menu. If no auxiliary node is given, the rotation of the rigid body is suppressed.

Rotations applied to the auxiliary node are with respect to the center of rotation of the rigid body, which is set to the coordinates of the control node.

The auxiliary node must be defined after the control node and it is not allowed to use the same node as control node and auxiliary node.

Keyboard Command Sequence:
```
cbody_control_node_rot <node>
```
**RESET**

Command: `cbody_reset`

Description: This command resets all contact body parameters and options to their default values.

Keyboard Command Sequence:

```
cbody_reset
```

---

**TABLE**

Command: `cbody_table`

Description: This command selects a table to be applied to a property of a contact body. You must specify a slot name and a previously defined table by name. Slot names are specified by a property name appended with a slot number (e.g. `x0`).

A table is a function of one variable, i.e. $y = f(x)$, that is applied to the corresponding contact body property value. The table is commonly a function of a variable such as time, frequency, temperature, or position. These tables are used by the analysis program to define the spatial or temporal variation of a quantity.

Keyboard Command Sequence:

```
cbody_table <property name><slot number><table name>
```

---

**CHANGE CLASS**

Menu: CHANGE CLASS

Description: This menu contains commands for changing the class of elements.

Note that when you change the class of an element, any boundary conditions that are specified on the element will be transferred to the new element(s). The attach information of the edges and faces and the set membership of the element will be transferred as well.
CURRENT DIRECTORY

Command: change_directory
Description: This command allows you to change the program’s current directory. The current directory is initially the directory from which MSC.Marc Mentat was started. Relative path and file names specified to MSC.Marc Mentat begin at the current directory.

Keyboard Command Sequence:
change_directory <directory path>

ELEMENTS

Command: change_elements
Description: This command redefines the class of existing elements. You must specify a list of elements.

This command is usually used to transform from lower order (linear) to higher order (quadratic) elements and vice versa. After performing such an operation, it is recommended that the sweep command be executed.

Keyboard Command Sequence:
change_elements <element list> #

ELEMENTS

Command: change_elements_class
Description: This command replaces existing elements with new elements of a given class. You must specify a list of elements.

The command is usually used to transform from lower-order (linear) to higher-order (quadratic) elements and vice versa. For those conversions that require the creation of new nodes (such as the conversion from lower order to higher order elements), the command creates unique midside nodes that are shared by coinciding edges and faces of neighboring elements. No subsequent sweep command is required.

Keyboard Command Sequence:
change_elements_class <element list> #
TO LINEAR ELEMENTS

Command: `change_elements_linear`

Description: This command replaces existing higher-order elements with new lower-order elements. You must specify a list of elements. The midside nodes of the original elements will be removed. No subsequent sweep command is required.

Keyboard Command Sequence:

```
change_elements_linear <element list> #
```

TO QUADRATIC ELEMENTS

Command: `change_elements_quadratic`

Description: This command replaces existing higher-order elements with new lower-order elements. You must specify a list of elements. The command creates unique midside nodes that are shared by coinciding edges and faces of neighboring elements. No subsequent sweep command is required.

Keyboard Command Sequence:

```
change_elements_quadratic <element list> #
```

ASPECT RATIO

Command: `check_aspect`

Description: This command checks every element in the mesh for aspect ratio. Aspect ratio is defined as the ratio of perimeter to the area of 2-D elements and total surface area to the volume in the case of 3-D elements, normalized with respect to corresponding elements of unit size. Equilateral triangles, squares, cubes etc., have a perfect aspect ratio of 1.0. Deviations from these perfect shapes result in higher aspect ratios. Upon completion of this command, all elements having aspect ratios higher than the set threshold + 1.0 are selected.

Keyboard Command Sequence:

```
check_aspect
```
CROSS ELEMENTS

Command: check_cross
Description: This command checks shell elements in the mesh for intersection. If one edge of an element goes through the interior area of another element, the two elements are considered to cross each other. If two elements stay co-planar, and one edge of an element cross with an edge of the other element, they are considered to cross each other. SWEEP-TOLERANCE is used to control how two elements are considered to be co-planar, where all nodes stay within two parallel planes whose distance is equal to SWEEP-TOLERANCE. A surface mesh model is not meshed if it contains cross elements.

Keyboard Command Sequence:
check_cross

CHECK CURVES

Command: check_curves
Description: This command checks the topology of the given list of curves. It will print out the number of segments, number of outer loops, number of closed loops, and the number of open loops defined. Also, it will print out the minimum and maximum lengths of the given curves. If curves are attached to surfaces, their lengths in surface parametric space are printed out too. The checking process is controlled by the relative tolerance. This command is available only in MSC.Marc Mentat and not the MSC.SuperForm GUI.

Keyboard Command Sequence:
check_curves <Curve List>

CHECK CURVES

Command: check_curves_af
Description: This command checks the topology of the given list of curves. It will print out the number of segments, number of outer loops, number of closed loops, and the number of open loops defined. Also, it will print out the minimum and maximum lengths of the given curves. If curves
are attached to surfaces, their lengths in surface parametric space are printed out too. This command is available only in the MSC.SuperForm GUI and not in MSC.Marc Mentat.

**Keyboard Command Sequence:**

```
check_curves_af <Curve List>
```

**Command:** check_distorted

**Description:** This command checks every element in the mesh for shape distortion. Elements with distortion value greater than user specified threshold value under **THRESHOLD** are identified as distorted. Upon completion of this command, all distorted elements are selected.

**Keyboard Command Sequence:**

```
check_distorted
```

Another Button with the same description:

**THRESHOLD**

**Menu:** CHECK

**Description:** This menu contains commands for checking the integrity of the model. There are commands for checking for distorted, upside-down, or inside-out elements, and commands for reversing the orientation of elements, curves, and surfaces.

Another Button with the same description:

**CHECK MESH**
**INSIDE OUT**

Command: `check_inside_out`

Description: This command checks every solid element for negative Jacobian. This usually results from an incorrect element connectivity caused by EXPAND or SYMMETRY commands. Upon completion of this command, all inside out elements are selected. Inside-out elements can be repaired with the `flip_elements` command.

Keyboard Command Sequence: 
```
check_inside_out
```

**CHECK**

Command: `check_job`

Description: This command checks the validity and completeness of the current job. Also, it will assign automatically the appropriate MSC.Marc element types for each element if possible.

Keyboard Command Sequence: 
```
check_job
```

**CHECK EDGES**

Command: `check_solid_edges`

Description: Traversed the selected edges and checked for problems. If there is a problem, it is printed to the message area. Here are the checks performed by the program:

- self-intersecting,
- is twisted,
- has too much oscillation,
- has regenerate edges, or
- is not G0, G1, or G2.

Keyboard Command Sequence: 
```
check_solid_edges <edge list> #
```
Command: check_solid_entities

Description: Traversed the selected entities and checked for problems. If there is a problem, it is printed to the message area. Here are the checks performed by the program.

Data Structure Checks:
- The parent has appropriate child-level entity; e.g. body has lump.
- Presence (non-NULL) and closure of backptr from child to parent; e.g. body slumps points to body.
- The coedge on spline surface has PCURVE.
- PCURVE indexing (0/+1/-+2) is appropriate.
- The PCURVE has non-NULL bs2_curve. The form of the PCURVE agrees with the form of the CURVE.

Topological Checks:
- LOOPS must be closed in both the next and prev directions.
- Coedges around a vertex form a circular double-linked list through alternate next/previous pointers and all share the same vertex.
- Coedges have partners whose partners are themselves.
- Partner coedges share (point to) the same edge and have opposite senses.

Geometric Checks:
- The vertex point lies in the face and on the edge curve.
- The face normal is consistent with coedge direction.
- The PCURVE start/end parameters match those of coedge.
- The PCURVE start/end location (wrt surf parameters) must equal coedge’s start/end location.
- Entities with geometry must have non-NULL geometry; e.g. FACE points to SURFACE and EDGE has CURVE, etc.

Keyboard Command Sequence:

check_solid_entities <entities list> #

Another Button with the same description:

CHECK LOG
CHECK FACES

Command: `check_solid_faces`

Description: Traversed the selected faces and checked for problems. If there is a problem, it is printed to the message area. Here are the checks performed by the program:
- self-intersecting,
- is twisted,
- has too much oscillation,
- has egenerate edges, or
- is not G0, G1, or G2.

Keyboard Command Sequence:
```
check_solid_faces <face list> #
```

CHECK SURFACES

Command: `check_surfaces`

Description: This command checks the topology of the curves trimming each surface in the given list. For each surface, it will print out the number of segments, number of outer loops, number of closed loops, and the number of open loops defined. Also, it will print out the minimum and maximum trimming curve lengths for the given surfaces. The checking process is controlled by the relative tolerance.

Keyboard Command Sequence:
```
check_surfaces <Surface List>
```

CHECK SURFACES

Command: `check_surfaces_af`

Description: This command checks the topology of the curves trimming each surface in the given list. For each surface, it will print out the number of segments, number of outer loops, number of closed loops, and the number of open loops defined. Also, it will print out the minimum and maximum trimming curve lengths for the given surfaces.

Keyboard Command Sequence:
```
check_surfaces_af <Surface List>
```
**UPSIDE DOWN**

**Command:** check_upside_down  
**Description:** This command checks all two-dimensional elements (including shells) for negative Jacobian. This usually results from an incorrect connectivity caused by EXPAND or SYMMETRY commands. Upon completion of this command, all upside-down elements are selected. Upside-down elements can be repaired with the flip_elements command.

**Keyboard Command Sequence:**
```
check_upside_down
```

**ZERO VOLUME**

**Command:** check_zero  
**Description:** This command checks every element in the mesh for zero volume. Upon completion of this command, all elements which have zero volume are selected.

**Keyboard Command Sequence:**
```
copy_insert
```

**CLEAN 2D CURVE LOOPS**

**Command:** clean_2d_loops  
This command removes curves whose lengths are smaller than the tolerance and merges points together when the distance between them is shorter than the tolerance. The tolerance value is computed by multiplying the relative tolerance value by the curve length.

Also see: set_relative_tol and clean_surface_loops.

**Keyboard Command Sequence:**
```
clean_2d_loops <curve list>
```
CLEAN 2D CURVE LOOPS

Command: clean_2d_loops_af

This command removes curves whose lengths are shorter than the minimum tolerance and merges points together when the distance between them is shorter than the minimum tolerance.

Also see: set_mesh_min_tol and clean_surface_loops.

Keyboard Command Sequence:
clean_2d_loops_af <curve list>

CLEAN

Command: clean_animation

Description: This command removes the temporary animation display lists which are created with the commands:

- animation_save
- post_animate_increments
- post_animate_mode
- post_animate_harmonic

or those that are created after you have performed the steps for MPEG MOVIE or AVI MOVIE.

The files that are removed are those named <base name>nnn, where <base name> is the name specified with the animation_name command (BASE FILE button). It defaults to animation. The nnn is a numbered sequence from 1 to 1000, unless LAST (movie_last_increment command) is larger, in which case it is used.

You would typically run this command after you have run movie_animation_make (MAKE MOVIE button) or avi_animation_make (MAKE AVI MOVIE button).

Also see help: clean_mpeg_animation.

Keyboard Command Sequence:
clean_animation
**CLEAN**

**Command:** clean_mpeg_animation

**Description:** This command removes the temporary animation display lists which are created when the command mpeg_animation_make (MAKE MPEG MOVIE button).

The files that are removed are those named <base name>nnn, where <base name> is the name specified with the animation_name command (BASE FILE button). It defaults to animation. The nnn is a numbered sequence from 1 to 1000, unless LAST (movie_last_increment command) is larger, in which case it is used.

The files named <base name>nnn.ppm are also removed.

You would typically run this command after you have successfully run the command mpeg_animation_make (MAKE MPEG MOVIE button) and have obtained a valid .mpeg file.

**Notes:** The MPEG movie is created in the background by running the mpeg_encode.exe program. It does NOT send any feedback to Mentat that it is done. You must verify that the mpeg_encode.exe program is no longer running to be sure that it has finished before running this command, otherwise you will have deleted the .ppm files that the MPEG encoder needs.

Use the Windows Task Manager on NT or use the ps command on UNIX to verify that the program has ended.

Also see help: clean_animation.

**Keyboard Command Sequence:**

```
clean_mpeg_animation
```

---

**CLEAN ENTITIES**

**Command:** clean_solid_entities

**Description:** Removes all unnecessary and/or redundant edges (faces and associated data) and vertices from the entity. An edge is not needed if the surface defining the two faces of the edge are the same geometrically.

**Keyboard Command Sequence:**

```
clean_solid_entities <entities list> #
```
**CLEAN SURFACE LOOPS**

**Command:** `clean_surface_loops`

**Description:** This command performs several functions to clean surface geometry for surface meshing. It hooks the trimming curves into loops on the base of the distances between the end points of curves. In addition, it also matches the neighboring trimming curves.

Four options are provided to control the cleaning process. They are Trim Bare Surfaces, Remove Free Curves, Break Curves, and Match Curves.

This process is implemented based on the tolerance. The tolerance parameter can be specified under OPTION menu. This tolerance parameter is used as a relative coefficient. The absolute tolerance is computed by multiplying this user specified value by the curve length.

Also see: `set_trim_surfs`, `set_rm_free_crvs`, `set_break_crvs`, `set_match_crvs`, and `set_relative_tol`.

**Keyboard Command Sequence:**

```
clean_surface_loops <surface list>
```

---

**CLEAN SURFACE LOOPS**

**Command:** `clean_surface_loops_af`

**Description:** This command performs several functions to clean surface geometry for surface meshing. It hooks the trimming curves into loops on the base of the distances between the end points of curves. In addition, it also matches the neighboring trimming curves.

Four options are provided to control the cleaning process. They are Trim Bare Surfaces, Remove Free Curves, Break Curves, and Match Curves.

This process is implemented based on the tolerance. The tolerance parameter can be specified under OPTION menu. This tolerance parameter is used as a relative coefficient. The absolute tolerance is computed by multiplying this user specified value by the curve length.
Also see: set_trim_surfs, set_rm_free_crvs, set_break_crvs, set_match_crvs, and set_relative_tol.

**Keyboard Command Sequence:**
```
clean_surface_loops_af <surface list>
```

---

**Command:** clear_annotations
**Description:** This command removes all annotations. This button is located in the UTILITIES->MORE menu.

**Keyboard Command Sequence:**
```
clear_annotations
```

---

**Command:** apply_dof  
clear_apply_dof
**Description:** These commands add and clear the specified degree of freedom from the current boundary condition application. The button text will be the name of the boundary condition.

**Keyboard Command Sequence:**
```
apply_dof <dof name>
clear_apply_dof <dof name>
```

---

**Command:** clear_csect_control_node
**Description:** This command removes the previously selected control node of the current cross-section. This button is located in the BOUNDARY CONDITION->MECHANICAL->CROSS SECTIONS menu.

**Keyboard Command Sequence:**
```
clear_csect_control_node
```
**Command:** clear_current_apply  
**Description:** This command clears all degrees of freedom, values, and table specifications from the current boundary condition. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->FIXED DISPLACEMENT and other boundary conditions menus.  
**Keyboard Command Sequence:**  
```
clear_current_apply
```
CLEAR

Command: clear_element_type
Description: This command clears the MSC.Marc element type for the specified elements. If a job is submitted to MSC.Marc with elements with no assigned type, MSC.Marc Mentat will assign an appropriate type based on element geometry and analysis type. MSC.Marc element types may be assigned to elements with the element_type command. This button is located in the JOBS->ELEMENT TYPES menu.

Keyboard Command Sequence:
   clear_element_type <element list> #

CLEAR GEOM

Command: clear_geometry
Description: This command deletes all existing points, curves, and surfaces. By using clear_geometry and clear_mesh together, the entire display can be cleared of its graphical components. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:
   clear_geometry

CLEAR

Command: clear_geometry_values
Description: This command clears all property values for the current geometry. This button is located in the JOBS->ELEMENT TYPES menu.

Keyboard Command Sequence:
   clear_geometry_values
Command: **icond_dof**  
**clear_icond_dof**

**Description:** These commands add and clear the specified degree of freedom from the current initial condition. The button text will be the name of the initial condition. This button is located in the INITIAL CONDITIONS->MECHANICAL->DISPLACEMENT and other initial conditions menus.

**Keyboard Command Sequence:**
```
icond_dof <dof name>
clear_icond_dof <dof name>
```

Command: **clear_icond_layers**

**Description:** This command clears element layers for the current state variable initial condition.

**Keyboard Command Sequence:**
```
clear_icond_layers
```

Command: **clear_ideas_elem_vars**

**Description:** This command clears the current job’s list of I-DEAS element results variables. This button is located in the JOBS->MECHANICAL->JOB RESULTS->I-DEAS other analysis class menus.

**Keyboard Command Sequence:**
```
clear_ideas_elem_vars
```
**INCREMENTS**

Command: `clear_job_buckle_increments`

Description: This command clears increments from the current job’s buckle analysis parameters.

**Keyboard Command Sequence:**
```
clear_job_buckle_increments
```

**CLEAR**

Command: `clear_job_careas`

Description: This command removes all contact areas from the current job. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT other analysis class menus.

**Keyboard Command Sequence:**
```
clear_job_careas
```

**CLEAR**

Command: `clear_job_ctable`

Description: This command removes the initial contact table from the current job. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->CONTACT TABLE and other analysis class menus.

**Keyboard Command Sequence:**
```
clear_job_ctable
```
Command: clear_job_exsegs
Description: This command removes all exclude segments from the current job. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT and other analysis class menus.

Keyboard Command Sequence:
   clear_job_exsegs

Command: clear_job_iconds
Description: This command clears all initial conditions from the current job. This button is located in the JOBS->MECHANICAL->INITIAL LOADS and other analysis class menus.

Keyboard Command Sequence:
   clear_job_iconds

Command: clear_job_loadcases
Description: This command removes all loadcases from the current job. This button is located in the JOBS->MECHANICAL and other analysis class menus.

Keyboard Command Sequence:
   clear_job_loadcases
Command: **clear_job_modal_increments**

Description: This command clears increments from the current job’s modal analysis parameters. This button is located in the JOBS->MECHANICAL->ANALYSIS OPTIONS->BUCKLE INCREMENTS menu.

Keyboard Command Sequence:

```
clear_job_modal_increments
```
Command: clear_loadcase_cbodies
Description: This command removes all contact body releases from the current loadcase. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT other loadcase type menus.

Keyboard Command Sequence:
    clear_loadcase_cbodies

Command: clear_loadcase_cstable
Description: This command clears the contact table from the current loadcase. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->CONTACT TABLE other loadcase type menus.

Keyboard Command Sequence:
    clear_loadcase_cstable

Command: clear_loadcase_exsegs
Description: This command removes all exclude segments from the current loadcase. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT other loadcase type menus.

Keyboard Command Sequence:
    clear_loadcase_exsegs
**Command:** clear_loadcase_loads  
**Description:** This command clears all boundary condition (loads) from the current loadcase. This button is located in the LOADCASES->MECHANICAL->STATIC->LOADS other loadcase class menus.

**Keyboard Command Sequence:**
```
clear_loadcase_loads
```

---

**Command:** clear_material_table  
**Description:** This command clears a table for a property in the current material.

The property that the table is to be removed from is designated by the material type and the property name is separated by a colon and appended with a table number.

For example, a table for Young’s modulus in anisotropic material is designated as isotropic:e0. Table numbers start at zero.

Note that only certain properties may have tables associated with them. Refer to Marc material model descriptions to determine which properties can have tables.

**Keyboard Command Sequence:**
```
clear_material_table <material type>: <property name> <table number>
```
**CLEAR**

**Command:**  `clear_mesh`

**Description:** This command deletes all nodes and element from the current model. It deletes the entire mesh, leaving the original geometry intact, so that the mesh can be regenerated. By using `CLEAR MESH` and `CLEAR GEOMETRY` together, the entire display can be cleared of its graphical components. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
clear_mesh
```

Another Button with the same description:

```
CLEAR MESH
```

**clear_post_tensors**

**Command:**  `clear_post_tensors`

**Description:** This command clears the current job’s list of results tensors. This command is only available in the SuperForm GUI.

**Keyboard Command Sequence:**

```
clear_post_tensors
```

**CLEAR**

**Command:**  `clear_post_vars`

**Description:** This command clears the current job’s list of results variables. This command is only available in the SuperForm GUI.

```
clear_post_vars
```
CLOSE

Command: close_procedure
Description: This command terminates the recording of commands in the procedure file and closes the file.
Keyboard Command Sequence:
   close_procedure

BACKGROUND

Command: color
Description: This command changes the color the program uses for a particular index. The indices range from zero to 95. Each color index is used when drawing a particular item on the screen or to a color plotter. For instance, the text in the dialogue area of the program uses color index 1. You must specify the color index to change and the new red, green, or blue values for the new color. These values range from zero to one, where zero means completely off, and one means completely on.

After using this command to change the color at an index, the lighting material properties of the index are automatically changed to look like plastic of the given color. Thus when lighting is on, or when rendering, items using the color index will look like they’re made of plastic.

Also see: color_hls and reset_colors.

Keyboard Command Sequence:
   color <color index> <red> <green> <blue>

Other Buttons with the same description:

0 through 95
ACTIVE WINDOW BORDERS
ANNOTATIONS
ARROW EDGES
BACKFACES
BOUNDARY CONDITIONS
CONTOUR LINE BACKGROUND
CONTOUR LOWER BOUND
CONTOUR UPPER BOUND
CURVES
Command: **color_ambient**

**Description:** This command changes the ambient lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The ambient color of an item interacts with the ambient light in a view when lighting is on, or when rendering. Ambient light is nondirectional, and contributes to the overall light in a view. The indices range from zero to 95. You must specify the color index to change and the new red, green, or blue values for the ambient color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_ambient <color index> <red> <green> <blue>
```
ATTEN. LENGTH

Command: color_atten_len
Description: This command changes the attenuation length lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The attenuation length of a solid item is used when rendering. White light traveling through a transparent solid becomes the attenuation color of the solid after traveling through it for a distance equal to the attenuation length of the solid. Thus, the thicker a solid is, the more it is colored by its attenuation color. Also, the smaller the attenuation length of the solid, the more colored it will be by its attenuation color. A solid is made transparent by setting its refractive color to 1, 1, 1. The indices range from zero to 95. You must specify the color index to change and the new attenuation length in view space.
Also see: color, color_attenuation, color_refractive, set_lighting, and render.

Keyboard Command Sequence:

    color_atten_len <color index> <length>

---

Command: color_attenuation
Description: This command changes the attenuation lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The attenuation color of a solid item is applied when rendering. White light traveling through a transparent solid becomes the attenuation color of the solid after traveling through it for a distance equal to the attenuation length of the solid. Thus, the thicker a solid is, the more it is colored by its attenuation color. A solid is made transparent by setting its refractive color to 1, 1, 1. The indices range from zero to 95. You must specify the color index to change and the new red, green, or blue values for the attenuation color. These values range from zero to one, where zero means completely off, and one means completely on.
Also see: color, atten_len, color_refractive, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_attenuation <color index> <red> <green> <blue>
```

---

**BRASS**

**Command:** color_brass

**Description:** This command makes items drawn with a given color index look like brass. The color and lighting material properties the program uses for the particular color index are set to create the appearance of brass. The full brass-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_brass <color index>
```

---

**BRONZE**

**Command:** color_bronze

**Description:** This command makes items drawn with a given color index look like bronze. The color and lighting material properties the program uses for the particular color index are set to create the appearance of bronze. The full bronze-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_bronze <color index>
```
CHROME

Command: color_chrome

Description: This command makes items drawn with a given color index look like chrome. The color and lighting material properties the program uses for the particular color index are set to create the appearance of chrome. The full chrome-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_chrome <color index>

COPPER

Command: color_copper

Description: This command makes items drawn with a given color index look like copper. The color and lighting material properties the program uses for the particular color index are set to create the appearance of copper. The full copper-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_copper <color index>
**DIAMOND**

**Command:** `color_diamond`  
**Description:** This command makes items drawn with a given color index look like diamond. The color and lighting material properties the program uses for the particular color index are set to create the appearance of diamond. The full diamond-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.  
Also see: `color`, `set_lighting`, and `render`.  

**Keyboard Command Sequence:**  
```  
color_diamond <color index>  
```

**DIAMONDIZE**

**Command:** `color_diamonize`  
**Description:** This command makes items drawn with a given color index look like a colored diamond. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of a colored diamond. The full diamond-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.  
Also see: `color`, `color_diamond`, `set_lighting`, and `render`.  

**Keyboard Command Sequence:**  
```  
color_diamonize <color index>  
```
**color_diffuse**

**Command:**  color_diffuse

**Description:** This command changes the diffuse lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The diffuse color of an item is its regular, dull, or matte color seen when lighting is on, or when rendering. This is the color of an item that becomes more intense as the angle of the light falling on the item approaches that of the items surface normal. The indices range from zero to 95. You must specify the color index to change and the new red, green, or blue values for the diffuse color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: color, color_specular, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_diffuse <color index> <red> <green> <blue>
```

**color_emission**

**Command:**  color_emission

**Description:** This command changes the emission lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The emission color of an item is the color of the light emitted by the item when lighting is on, or when rendering. The indices range from zero to 95. You must specify the color index to change and the new red, green, or blue values for the emission color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_emission <color index> <red> <green> <blue>
```
**FROSTED GLASS**

Command: **color_frosted_glass**

Description: This command makes items drawn with a given color index look like frosted glass. The color and lighting material properties the program uses for the particular color index are set to create the appearance of frosted glass. The full frosted glass-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **set_lighting**, and **render**.

Keyboard Command Sequence:

```
color_frosted_glass <color index>
```

---

**FROSTED GLASSIZE**

Command: **color_frosted_glassize**

Description: This command makes items drawn with a given color index look like colored frosted glass. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored frosted glass. The full frosted glass-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **color_frosted_glass**, **set_lighting**, and **render**.

Keyboard Command Sequence:

```
color_frosted_glassize <color index>
```
Command: color_ghost

Description: This command makes items drawn with a given color index look like ghost. The color and lighting material properties the program uses for the particular color index are set to create a translucent appearance that darkens with thickness, and which does not bend refracted light. The full ghost-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_ghost <color index>

Command: color_ghostize

Description: This command makes items drawn with a given color index look like colored ghost. The basic color already at a particular color index is used to set the lighting material properties for that index to create a colored translucent appearance that darkens with thickness, and which does not bend refracted light. The full ghost-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, color_ghost, set_lighting, and render.

Keyboard Command Sequence:

color_ghostize <color index>
Command: **color_glass**

**Description:** This command makes items drawn with a given color index look like glass. The color and lighting material properties the program uses for the particular color index are set to create the appearance of glass. The full glass-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **set_lighting**, and **render**.

**Keyboard Command Sequence:**

```
color_glass <color index>
```

---

Command: **color_glassize**

**Description:** This command makes items drawn with a given color index look like colored glass. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored glass. The full glass-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **color_glass**, **set_lighting**, and **render**.

**Keyboard Command Sequence:**

```
color_glassize <color index>
```
Command: **color_gold**

**Description:** This command makes items drawn with a given color index look like gold. The color and lighting material properties the program uses for the particular color index are set to create the appearance of gold. The full gold-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **set_lighting**, and **render**.

**Keyboard Command Sequence:**

```
color_gold <color index>
```

---

Command: **color_hls**

**Description:** This command changes the color the program uses for a particular index. The indices range from zero to 95. Each color index is used when drawing a particular item on the screen or to a color plotter. For instance, the text in the dialogue area of the program uses color index. You must specify the color index to change and the new hue, lightness, and saturation values for the new color.

The hue value ranges from zero to 360, where:

- 0 = blue
- 60 = magenta
- 120 = red
- 180 = yellow
- 240 = green
- 300 = cyan

The lightness component ranges from zero to one, where zero means no lightness, and one means complete lightness. The saturation component ranges from zero to one, where zero means no saturation (gray) and one means complete saturation (vibrant color).

After using this command to change the color at an index, the lighting material properties of the index are automatically changed to look like...
plastic of the given color. Thus when lighting is on, or when rendering, items using the color index will look like they’re made of plastic.

Also see: color and reset_colors.

**Keyboard Command Sequence:**

```
color_hls <color index> <hue> <lightness> <saturation>
```

### COLOR INDEX

**Command:** `color_index`

**Description:** This command sets the index of the current color table entry.

**Keyboard Command Sequence:**

```
color_index <color table index>
```

### INDEX

**Command:** `color_index_ref`

**Description:** This command changes the index of refraction lighting material property the program uses for a particular color index. Each color index is used when drawing a particular item on the screen or to a color plotter. The index of refraction of an item is used when rendering. Light traveling through a transparent item will be bent by an angle determined by the index of refraction of the item. An item is made transparent by setting its refractive color to something other than black (0, 0, 0). The indices range from zero to 95. You must specify the color index to change and the new index of refraction.

Also see: color, color_refractive, color_index_ref_spread, set_lighting, and render.

**Keyboard Command Sequence:**

```
color_index_ref <color index> <index of refraction>
```
Command: color_index_ref_spread

Description: This command changes the index of refraction spread lighting material property the program uses for a particular color index. Each color index is used when drawing a particular item on the screen or to a color plotter. The index of refraction spread of an item is used to modify the index of refraction of an item when rendering. Light traveling through a transparent item will be bent by an angle determined by the index of refraction of the item. If the spread is set to zero, then all light passing through the item will be bent by the same amount (coherently). If the spread is greater than zero, light passing through the item will be split into its separate red, green, and blue components, and each ray will be bent differently.

The red ray will be bent using an index of refraction which is one half of the spread less than the index of refraction of the item it’s passing through.

The green ray will be bent using the index of refraction of the item.

The blue ray will be bent using an index of refraction which is one half of the spread greater than the index of refraction of the item. An item is made transparent by setting its refractive color to something other than black (0, 0, 0). The indices range from zero to 95. You must specify the color index to change and the new index of refraction spread.

Also see: color, color_refractive, color_index_ref, set_lighting, and render.

Keyboard Command Sequence:

```
color_index_ref_spread <color index>
<index of refr. spread>
```
**Command:** color_initialize

**Description:** This command makes items drawn with a given color index look like black plaster. The color and lighting material properties the program uses for the particular color index are set to create a black opaque non-reflective non-shiny appearance. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

color_initialize <color index>

---

**Command:** color_metalize

**Description:** This command makes items drawn with a given color index look like colored metal. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored metal. The full metal-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

**Keyboard Command Sequence:**

color_metalize <color index>
**Command:**  
`color_mirror`

**Description:**  
This command makes items drawn with a given color index look like a mirror. The color and lighting material properties the program uses for the particular color index are set to create the appearance of a mirror. The full mirror-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: `color`, `set_lighting`, and `render`.

**Keyboard Command Sequence:**

```
color_mirror <color index>
```

---

**Command:**  
`color_mirrorize`

**Description:**  
This command makes items drawn with a given color index look like a colored mirror. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of a colored mirror. The full mirror-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: `color`, `color_mirror`, `set_lighting`, and `render`.

**Keyboard Command Sequence:**

```
color_mirrorize <color index>
```
PLASTERIZE

Command:  color_plasterize

Description: This command makes items drawn with a given color index look like colored plaster. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored plaster. The full plaster-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_plasterize <color index>

PLASTICIZE

Command:  color_plasticize

Description: This command makes items drawn with a given color index look like colored plastic. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored plastic. The full plastic-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_plasticize <color index>
**color_reflective**

**Command:**  
color_reflective

**Description:**  
This command changes the reflective lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The reflective color of an item determines how other items are reflected in the item when rendering. The indices range from zero to 95. You must specify the color index to change and the new red, green, and blue values for the reflective color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: color, color_specular, set_lighting, and render.

**Keyboard Command Sequence:**

color_reflective <color index> <red><green><blue>

**color_refractive**

**Command:**  
color_refractive

**Description:**  
This command changes the refractive lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The refractive color of an item is used to make the item transparent, or opaque when rendering. If the refractive color is not black (0, 0, 0), then the item is at least partially transparent, and white light striking the item will pass through it, and become colored by the refractive color of the item. The transmitted light will also be bent by an angle determined by the index of refraction of the item. The indices range from zero to 95. You must specify the color index to change and the new red, green, and blue values for the refractive color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: color, color_index_ref, color_attenuation, set_lighting, and render.

**Keyboard Command Sequence:**

color_refractive <color index> <red><green><blue>
Command:  **color_rubber**  

**Description:** This command makes items drawn with a given color index look like *rubber*. The color and lighting material properties the program uses for the particular color index are set to create the appearance of rubber. The full rubber-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **set_lighting**, and **render**.

**Keyboard Command Sequence:**

```
color_rubber <color index>
```

Command:  **color_rubberize**  

**Description:** This command makes items drawn with a given color index look like *colored rubber*. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored rubber. The full rubber-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: **color**, **color_rubber**, **set_lighting**, and **render**.

**Keyboard Command Sequence:**

```
color_rubberize <color index>
```
SHININESS

Command:  color_shininess

Description: This command changes the shininess lighting material property the program uses for a particular color index. Each color index is used when drawing a particular item on the screen or to a color plotter. The shininess of an item is used when lighting is on, or when rendering. Shininess can range from zero to one, and determines the size of the specular highlights which appear on an item. The smaller or more focused these highlights are, the shinier the item will appear to be. A shininess of zero turns off the specular highlights completely. A shininess slightly above zero creates large highlights, while a shininess near one creates small, focused highlights. The indices range from zero to 95. You must specify the color index to change and the new shininess.

Also see: color, color_specular, set_lighting, and render.

Keyboard Command Sequence:

color_shininess <color index> <shininess>

SHINY METALIZE

Command:  color_shiny_metalize

Description: This command makes items drawn with a given color index look like colored shiny metal. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored shiny metal. The full shiny metal-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

color_shiny_metalize <color index>
Command:  **color_silver**  

**Description:** This command makes items drawn with a given color index look like *silver*. The color and lighting material properties the program uses for the particular color index are set to create the appearance of silver. The full silver-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: *color, set_lighting, and render.*

**Keyboard Command Sequence:**

```
color_silver <color index>
```

---

Command:  **color_specular**  

**Description:** This command changes the specular lighting material property the program uses for a particular index. Each color index is used when drawing a particular item on the screen or to a color plotter. The specular color of an item is used to create the highlights on the item, if it’s shiny, when lighting is on, or when rendering. The indices range from zero to 95. You must specify the color index to change and the new red, green, and blue values for the specular color. These values range from zero to one, where zero means completely off, and one means completely on.

Also see: *color, color_shininess, color_diffuse, set_lighting, and render.*

**Keyboard Command Sequence:**

```
color_specular <color index> <red><green><blue>
```
STEEL

Command:  color_steel

Description:  This command makes items drawn with a given color index look like steel. The color and lighting material properties the program uses for the particular color index are set to create the appearance of steel. The full steel-like appearance of the item will be visible when lighting is on, or when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

    color_steel <color index>

TRANSLUCENCY

Command:  color_translucency

Description:  This command changes the translucency of all filled polygons drawn with a given color index. You must specify the index and the translucency value to be used. Translucency is specified as a number between zero and one. A translucency of zero means all filled polygons drawn with the color index will be completely opaque, whereas a translucency of one means they will be completely transparent. Intermediate values mean that white light striking the filled polygons will pass through them and become colored by their refractive color.

Translucency is actually the equivalent gray value for the refractive color at the given color index. When a new translucency is specified, the lightness of the refractive color is adjusted until its grayness matches the translucency value. The grayness of a color is defined to be the sum of 30% of the color’s red component, 59% of the color's green component, and 11% of the color’s blue component.

Also see: color_refractive and set_lightness.

Keyboard Command Sequence:

    color_translucency <color index>
    <translucency value>
Command: color_water
Description: This command makes items drawn with a given color index look like water. The color and lighting material properties the program uses for the particular color index are set to create the appearance of water. The full water-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, set_lighting, and render.

Keyboard Command Sequence:

    color_water <color index>

Command: color_waterize
Description: This command makes items drawn with a given color index look like colored water. The basic color already at a particular color index is used to set the lighting material properties for that index to create the appearance of colored water. The full water-like appearance of the item will be visible when rendering. A color index is used when drawing a particular item on the screen. You must specify the color index to change. The indices range from zero to 95.

Also see: color, color_water, set_lighting, and render.

Keyboard Command Sequence:

    color_waterize <color index>
Command: colormap

Description: This command sets the current color map for the program to be 1 through 8. There are eight predefined collections of colors that the program can be made to use for menus and ordinary geometry with this command. Each of these collections maps a red, green, and blue value to each color index. By default, color map 1 is used.

Keyboard Command Sequence:

\texttt{colormap <1, 2, 3, 4, 5, 6, 7, or 8>}

Other Buttons with the same description:

2 through 8
DEF.
GRAY
REVERSE

Menu: COLORS

Description: This menu contains commands for changing the colors used by the program and for saving and restoring the color map.

Command: composite_material

Description: This command adds material layers to the current composite material. You must specify the layer where the material is to be added and the name of the material name to add. The material to be added must already exist.

Keyboard Command Sequence:

\texttt{composite_material <layer number> <material name>
CONRAD

Subroutine: CONRAD

Description: The user subroutine CONRAD allows you to modify mass conrad rate, inlet temperature, and film coefficient in heat transfer analyses involving fluid channel elements. Both the inlet temperature and mass conrad rate can be dependent on time; the film coefficient can also be a function of streamline distance.

Command: contact_value
ac_bound_c1_inv

This command sets the inverse reactive boundary coefficient 1/C1. Using 1/C1 and 1/K1 (see cont_ac_bound_k1_inv), along the acoustic-structural interface non-reflecting boundary conditions are introduced using a spring-dashpot analogy, with K1, the spring; and C1, the dashpot parameter, both per unit area. This button is a text box and the button displays the value associated with this property.

Reference: MSC.Marc Volume A: Theory and User Information, Chapter 6

Keyboard Command Sequence:

```
contact_value
ac_bound_c1_inv <inv. react. bound. coef. 1/C1>
```
Command:  \texttt{contact\_value}
\texttt{ac\_bound\_k1\_inv}

This command sets the inverse reactive boundary coefficient $1/K_1$. Using $1/K_1$ and $1/C_1$ (see \texttt{cont\_ac\_bound\_c1\_inv}), along the acoustic-structural interface non-reflecting boundary conditions are introduced using a spring-dashpot analogy, with $K_1$, the spring; and $C_1$, the dashpot parameter, both per unit area. This button is a text box and the button displays the value associated with this property.

Reference:  \textit{MSC.Marc Volume A: Theory and User Information}, Chapter 6

Keyboard Command Sequence:

\begin{verbatim}
contact_value
ac_bound_k1_inv <inv. react. bound. coef. 1/K1>
\end{verbatim}

Menu:  \textbf{ACOUSTIC}

Description:  This menu contains commands for setting contact properties of an acoustic body.

An acoustic contact body is a set of elements that act as a body in an acoustic-solid analysis. The contact body must be a connected entity. In 2-D, up to 80 holes may be present in a contact body; in 3-D, there are no limitations regarding the number of holes. Nodes and elements may not occur in more than one contact body.

An acoustic contact body can contact other bodies but not itself and cannot be contacted by other bodies.
**CENTER OF ROTATION**

**Command:** `cbody_center_rot`  
**Description:** This command is used for defining the center of rotation of the current rigid body.

The center of rotation can only be defined directly for velocity and position controlled rigid bodies. For load controlled rigid bodies, it is set to the coordinates of the control node which is associated with the rigid body.

The center of rotation is the point around which a rotation of a rigid body is defined. In the postprocessing stage, it is the reference point for resultant moments on the rigid contact body.

**Keyboard Command Sequence:**

```
contact_value <cx/cy/cz> <value>
```
**CURVE DIVISIONS**

**Command:** `contact_value`  
`curve_div`

**Description:** This command is used for setting the number of divisions of a curve that acts as a rigid contact body.

For the analytical description, this value is only used for visualization of results if a post file style older than Marc version K7 is used.

For the discrete description, it defines the subdivision into straight segments. A default value is provided that is appropriate for most cases.

**Keyboard Command Sequence:**

```
contact_value  
curve_div <value>
```

---

**ANALYTICAL**

**Command:** `contact_option`  
`defo_desc:<analytical/discrete>`

**Description:** These commands are used for setting the type of boundary description to be used for a deformable body or a rigid body with heat transfer.

In the default discrete description, the boundary of the contacted body is described by the finite elements that the body is made up of. This can cause problems due to the fact that the normals of the body are not continuous for a curved boundary described with lower-order elements.

In the analytical description, local spline curves are defined through the boundary nodes of a two-dimensional contacted body. In three dimensions, local Coons surfaces are defined through neighboring surface segments. The nodes of the contacting body are now touching these analytical entities instead of the actual finite elements. These entities are updated as the body is deformed. Corners and edges that are not to be smoothed can be identified with the commands `add_cbody_dc_nodes` (2-D or axisymmetric analysis) and `add_cbody_dc_edges` (3-D analysis), respectively. These commands are available under ANALYTICAL DESC. DISCONTINUITY in the CONTACT BODIES menu.
**Keyboard Command Sequence:**

```
contact_option
defo_desc:<analytical/discrete>
```

Another Button with the same description:

**DISCRETE**

---

**Menu:** DEFORMABLE

**Description:** This menu contains commands for setting contact properties of a deformable body.

A deformable contact body is a set of elements that acts as a body in a contact analysis. The contact body must be a connected entity. In 2-D, up to 80 holes may be present in a contact body; in 3-D, there are no limitations regarding the number of holes. Nodes and elements may not occur in more than one contact body.

A deformable contact body can contact other bodies including itself and be contacted by other deformable bodies.

---

**Command:** contact_value film

**Description:** This command is used for setting the heat flux film coefficient corresponding to free convection for the current contact body. This coefficient is used whenever the boundary is free.

A film coefficient can also be applied as a boundary condition (BOUNDARY CONDITIONS -> THERMAL or JOULE). Avoid using this if a coupled analysis with contact is performed. Only use the definitions in the contact option in this case.

**Keyboard Command Sequence:**

```
contact_value film <film coefficient>
```
**FRICION COEFFICIENT**

**Command:** contact_value friction

**Description:** This command is used for setting the friction coefficient for the current contact body.

This friction coefficient is associated with the contact body. If a deformable body contacts a rigid body, the coefficient of the rigid body is used. If two deformable bodies contact, the average of the coefficients is used.

An alternative way to enter friction coefficients is by a contact table, which if used overrides the body specific value. Then the friction coefficient is explicitly specified for the contacting bodies.

The type of friction model to use is defined under CONTACT CONTROL in the JOBS menus.

**Keyboard Command Sequence:**

```
contact_value friction <friction coefficient>
```

**Command:** contact_value initvx/initvy/initvz/initvrot

**Description:** These commands are used for defining the initial velocity of the rigid body. The rotation is with respect to the center of rotation and the rotation axis. This button is a text box and the button displays the value associated with this property.

The rigid body is moved with this velocity until it contacts a deformable body before the analysis begins.

Note that the rotation velocities are given in radians per second. To enter a rotation velocity of 5 degrees/s, type `5*π/180` (π is a predefined variable).
Keyboard Command Sequence:

```
contact_value <initvx/initvy/initvz/initvrot> <value>
```

Other Buttons with the same description:

X
Y
Z

Command: **contact_option**

```
control:load
```

Description: This command is used to specify that the current rigid body is a general load-controlled rigid body.

A control node is associated with this body. Forces and prescribed displacements can be applied to this node. In addition, an auxiliary node can be associated with the rigid body. If no auxiliary node is given, the rotation of the rigid body is suppressed. If one is given, the rigid body can rotate. Moments and prescribed rotations can be applied to the auxiliary node. The control node has two translational degrees of freedom in 2-D and three in 3-D. The auxiliary node has one rotational degree of freedom in 2-D and three in 3-D. Note that the degrees of freedom are referred to as X DISPLACEMENT etc. in the boundary conditions menu. For instance, a moment around the z–axis is given as X FORCE for 2-D and Z FORCE for 3-D.

The initial velocity of the body is defined under the PARAMETERS menu. By giving an initial velocity to the rigid body, it will move in this direction until it contacts a deformable body before the analysis starts.

Keyboard Command Sequence:

```
contact_option control:position
```
Menu: PARAMETERS

Description: This menu contains commands for defining the initial velocity of a load-controlled rigid body.

Command: contact_option control:position

Description: This command is used to specify that the motion of the current rigid body is defined by specifying the position of its center of rotation as a function of time.

Note that it is the position of the center of rotation that is prescribed and not the displacement. The rotation is with respect to the original position (i.e. if a zero value is given for the angle position, the body will not rotate).

The prescribed position and initial velocity is defined under the PARAMETERS menu.

Keyboard Command Sequence:

contact_option control:position
**ANGLE (RAD)**

**Command:** `contact_value <bx/by/bz/brot>`

**Description:** These commands are used for defining the target position of the current rigid body.

Note that the position is defined with respect to the user defined center of rotation, which by default is in the origin.

**Keyboard Command Sequence:**

```
contact_value <bx/by/bz/brot> <value>
```

Other Buttons with the same description:

- X
- Y
- Z

**LINEAR MOTION**

**Command:**

```
contact_option pos_method:linear
contact_option pos_method:nonlinear
```

**Description:** These commands define the method to use for defining the position of a rigid body.

With NONLINEAR MOTION option, the position of the center of rotation of the rigid body is given as a function of time. Tables can be used to define variations in time of the components.

With LINEAR MOTION option, only the target position of the center of rotation is defined. The positions at intermediate stages are obtained by linear interpolation.

Note that the position is defined with respect to the user defined center of rotation, which by default is in the origin.

**Keyboard Command Sequence:**

```
contact_option pos_mentod:<linear/nonlinear>
```

Another Button with the same description:

- NONLINEAR MOTION
Menu: PARAMETERS

Description: This menu contains commands for defining the position and initial velocity of a rigid body.

Another Button with the same description:

PARAMETERS

Command: contact_value <px/py/pz/prot>

Description: These commands are used for defining the position of the current rigid body. Variation in time is given by means of tables.

Note that the position is defined with respect to the user defined center of rotation, which by default is in the origin.

Keyboard Command Sequence:

contact_value <px/py/pz/prot> <value>

Other Buttons with the same description:

X
Y
Z

Menu: RIGID

Description: This menu contains commands for setting contact properties of a rigid body.

A rigid contact body is a set of curves (2-D or axisymmetric analysis) or surfaces (3-D analysis) that acts as a body in a contact analysis. The rigid body can be fixed in space, be given a prescribed motion or have forces and moments applied to it.
Command: contact_option rigid_desc:<analytical/discrete>

Description: These commands are used for setting the type of description to be used for a rigid contact body.

By default, the lines and surfaces are described by analytical non-uniform rational B-splines (NURBS). This has the effect that the surfaces are smooth therefore, the surface normals are continuous. This formulation usually leads to more accurate results and better solution convergence than the discrete approach.

In the discrete formulation, the line or surface is treated like a piecewise linear entity, i.e. segments of straight lines or flat patches, respectively. The division of the line or surface can be controlled by the curve and surface divisions buttons. An appropriate value is automatically provided. If the contact body is a straight line or a flat surface, the difference between the two formulation is very small.

Avoid mixing analytical and discrete rigid bodies if a deformable body is touching more than one rigid body, as it may lead to a decreased accuracy in the determination of the intersection between the rigid bodies.

Keyboard Command Sequence:

contact_option rigid_disc:<analytical/discrete>

Another Button with the same description:

DISCRETE
Command: **clear_ideas_elnod_vars**

**Description:** This command clears the current job’s list of I-DEAS element-at-node results variables. This button is located in the JOBS->MECHANICAL->JOB RESULTS->I-DEAS other analysis class menus.

**Keyboard Command Sequence:**

```
clear_ideas_elnod_vars
```
Menu: RIGID w HEAT TRANSFER

Description: This menu contains commands for setting contact properties of a rigid body with heat transfer.

A rigid contact body with heat transfer is a set of heat transfer elements that acts as a rigid body in the stress part of a coupled thermal-mechanical contact analysis.

The contact body must be a connected entity. In 2-D, up to 80 holes may be present in a contact body; in 3-D, there are no limitations regarding the number of holes. Nodes and elements may not occur in more than one contact body.

The rigid body can be fixed in space or given a prescribed motion.

Command: `contact_value ax/ay/az`

Description: These commands are used for defining the rotation axis of the current rigid body.

Rotations of the rigid body is defined with respect to this axis.

**Keyboard Command Sequence:**

`contact_value <ax/ay/az> <value>`

Other Buttons with the same description:

Y
Z
Command:  contact_value surf_div_u  
            contact_value surf_div_v  

Description: These commands are used for setting the number of divisions of a surface  
               that acts as a rigid contact body.  
               For the analytical description these values are only used for visualization  
               of results if a post file style older than Marc version K7 is used.  
               For the discrete description, it defines the subdivision into straight  
               segments.  
               Default values are provided that are appropriate for most cases.  

Keyboard Command Sequence:  
            contact_value surf_div_u <value>  
            contact_value surf_div_v <value>  

Another Button with the same description:  
            SURFACE DIVISIONS V  

Menu:  SYMMETRY  

Description: This menu contains commands for setting contact properties of a rigid  
              symmetry body.  
              A rigid contact body is a set of curves (2-D or axisymmetric analysis) or  
              surfaces (3-D analysis) that acts as a body in a contact analysis. A  
              symmetry body is a special rigid body which is fixed in space and has  
              zero friction. It is usually a flat surface or straight line which acts as a  
              symmetry plane. A node cannot separate from a symmetry body. During  
              the heat transfer part of a coupled thermal-mechanical analysis, there is no  
              heat flux across a symmetry surface.  
              If a shell edge or beam end touches a symmetry body, the appropriate  
              rotations are also suppressed. This works also if the symmetry body is not  
              aligned with the global coordinate directions. In this case, a local  
              transformation of the rotations is performed.
ENVIRONMENT SINK TEMPERATURE

Command:  contact_value tsink
Description:  This command is used for setting the temperature of the environment of the current contact body.
This temperature is used in the analysis of the heat transfer between the surface of the body and the surrounding environment.

Keyboard Command Sequence:
  contact_value tsink
  <environment sink temperature>

TEMPERATURE

Command:  contact_value tsurf
Description:  This command is used for setting the temperature of the current rigid contact body.
This temperature is used in the analysis of the heat transfer between the rigid body and a deformable contact body. A rigid contact surface without heat transfer is considered to have a constant temperature (heat source or heat sink).

Keyboard Command Sequence:
  contact_value tsurf <temperature>

VELOCITY

Command:  contact_option control:velocity
Description:  This command is used to specify that the motion of the current rigid body is defined by giving it a velocity. If a rotational velocity is given, you need to specify the center of rotation and a rotational axis. The center of rotation is also used in the post processing phase as a reference point for resultant moment on the rigid body. The rotational axis cannot be changed during the analysis with this option. If that is desired, use
the load-controlled option instead (which can also be used to prescribe a motion of the body) or the user subroutine `UMOTION`.

A time scale is always defined in a contact analysis, also for a static analysis. This makes it possible to define a velocity in a static analysis as well as in a dynamic analysis.

The prescribed velocity is defined under the `PARAMETERS` menu.

**Keyboard Command Sequence:**

```
contact_option control:velocity
```

**PARAMETERS**

**Menu:** `PARAMETERS`

**Description:** This menu contains commands for defining the velocity of a rigid body.

**Command:** `contact_value vx/vy/vz/vrot`

**Description:** These commands are used to define the velocity of the rigid body. The rotation is with respect to the center of rotation and the rotation axis.

The velocity can be varied by using a table for each component. For a more general motion including a change of the rotation axis, use the load-controlled option instead (which can also be used to prescribe a motion of the body) or the user subroutine `UMOTION`.

Note that the rotation velocities are given in radians per second. To enter a rotation velocity of 5 degrees/s, type `5*pi/180` (pi is a predefined variable).

**Keyboard Command Sequence:**

```
contact_value <vx/vy/vz/vrot> <value>
```

Other Buttons with the same description:

Y
Z
Menu: CONTACT BODIES
Description: This menu contains commands for defining contact bodies for a contact analysis.

Command: contact_area_add_nodes
Description: This command adds nodes to the current contact area. The nodes should belong to the specified contact body of the contact area and lie on the outline or surface of that body. This button is located in the CONTACT->CONTACT AREAS menu.

Keyboard Command Sequence:
    contact_area_add_nodes <node list> #

Command: contact_area_body
Description: This command sets the contact body for the current contact area.

Keyboard Command Sequence:
    contact_area_body <contact body>

Command: contact_area_name
Description: This command sets or changes the name of the current contact area.

A contact area is a set of nodes that lie on a contact body that may potentially contact other contact bodies. Contact areas are used to reduce the computational cost of a contact analysis where it is known for which
nodes contact is possible. If no contact area is used, all exterior outline or surface nodes will be checked for contact.

**Keyboard Command Sequence:**

```plaintext
contact_area_name <contact area name>
```

**Command:** `contact_area_remove_nodes`

**Description:** This command removes nodes from the current contact area. This button is located in the CONTACT->CONTACT AREAS menu.

**Keyboard Command Sequence:**

```plaintext
contact_area_remove_nodes <node list> #
```

**Command:** `contact_area_reset`

**Description:** This command resets the current contact area. This button is located in the CONTACT->CONTACT AREAS menu.

**Keyboard Command Sequence:**

```plaintext
contact_area_reset
```

**Menu:** CONTACT AREAS

**Description:** This menu contains commands for defining contact areas.

A contact area is a set of nodes that lie on a contact body that may potentially contact other contact bodies. Contact areas are used to reduce the computational cost of a contact analysis where it is known for which nodes contact is possible. If no contact area is used, all exterior outline or surface nodes will be checked for contact.
Note that the contact area must be activated in the loadcase where they are to be used. This is defined in the CONTACT menu for the different loadcase types.

For the correct detection of initial contact (before the first loadcase), the contact area should also be activated in the current job. This is done in the INITIAL CONTACT menu in the CONTACT CONTROL menu for each analysis class (e.g. for an uncoupled mechanical analysis it is defined in: JOBS-> MECHANICAL-> CONTACT CONTROL-> INITIAL CONTACT).

Reference:  *MSC.Marc Volume C: Program Input*, Chapter 3

---

**CONTACT BODIES**

**Menu:** CONTACT BODIES

**Description:** This menu contains commands for defining contact bodies.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies, at least one of which must be a deformable body.

Always define deformable contact bodies before rigid.

Reference:  *MSC.Marc Volume A: Theory and User Information*, Chapter 8

---

**NAME**

**Command:** contact_body_name

**Description:** This command sets or changes the name of the current contact body.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies, at least one of which must be a deformable body.

**Keyboard Command Sequence:**

```
contact_body_name <contact body name>
```
EXCLUDE SEGMENTS

Menu: EXCLUDE SEGMENTS

Description: This menu contains commands for defining exclude segments.

An exclude segment is a set of edges and faces of a contact body that are guaranteed not to contact other contact bodies.

Exclude segments are used to reduce the computational cost of a contact analysis where it is known for which edges and faces contact is not possible. It is also a way to make sure that the proper part of a body is contacted. In some situations, a node can contact a part of a body which it is not supposed to contact. This can occur for instance when a node comes into contact close to a corner of another body and it slides along the wrong segment of the element at the corner. A better behavior can be obtained if the other segments are excluded.

Note that the exclude segments must be activated in the loadcase where they are to be used. This is defined in the CONTACT menu for the different loadcase types.

For the correct detection of initial contact (before the first loadcase), the exclude segments should also be activated in the current job. This is done in the INITIAL CONTACT menu in the CONTACT CONTROL menu for each analysis class. (For instance, for an uncoupled mechanical analysis, it is defined in JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT).

Reference: *MSC.Marc Volume C: Program Input*, Chapter 3
Command: **contact_option**

Description: This command sets the value of a contact body option. You must specify the option to be set. Job options are specified as the name of the option and its value separated by a colon (e.g. `symmetry:on`).

The valid contact body options and their values are as follows:

- `rigid_desc` analytical Rigid body description
- `symmetry` off, on Symmetry body

Keyboard Command Sequence:

```
contact_option <contact option>:<value>
```

---

Menu: **CONTACT PROPERTIES**

Description: This popup menu contains commands for setting contact body properties.

---

Command: **contact_rigid**

Contact deformation:

- `contact_deformable`

Description: These commands set the type of a contact body.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies, at least one of which must be a deformable body.

A deformable contact body is a collection of elements whose mechanical contact behavior is to be investigated. A rigid contact body (also called a rigid dye) is a collection of elements, curves, or surfaces that contact or press against deformable contact bodies causing deformation. The mechanical behavior of the rigid dye is not of interest and is assumed to be undeformable compared to the deformable contact body.
Keyboard Command Sequence:

contact_rigid
contact_deformable

Command: **contact_table_ac_bound_c1_inv**

This command sets the inverse reactive boundary coefficient 1/C1 for the current set of contact bodies. This button is a text box and the button displays the value associated with this property.

Using 1/C1 and 1/K1 (see **contact_table_ac_bound_k1_inv**), along the acoustic-structural interface non-reflecting boundary conditions are introduced using a spring-dashpot analogy with K1, the spring, and C1, the dashpot parameter, both per unit area.

Reference: *MSC.Marc Volume A: Theory and User Information*, Chapter 6

Keyboard Command Sequence:

```
contact_table_ac_bound_c1_inv
<inv. react. bound. coef. 1/C1>
```

Command: **contact_table_ac_bound_k1_inv**

This command sets the inverse reactive boundary coefficient 1/K1 for the current set of contact bodies. This button is a text box and the button displays the value associated with this property.

Using 1/K1 and 1/C1 (see **contact_table_ac_bound_c1_inv**), along the acoustic-structural interface non-reflecting boundary conditions are introduced using a spring-dashpot analogy with K1, the spring, and C1, the dashpot parameter, both per unit area.

Reference: *MSC.Marc Volume A: Theory and Information*, Chapter 6

Keyboard Command Sequence:

```
contact_table_ac_bound_k1_inv
<inv. react. bound. coef. 1/K1>
```
Command: `contact_table_deactivate`

Description: This command deactivates the contact table entry for the specified contact bodies in the current contact table. An inactive entry prevents Marc from checking for contact between the two bodies during a contact analysis. Contact bodies are specified by name. This button is only available in the SuperForm GUI.

Keyboard Command Sequence:
```
contact_table_deactivate <contact body name>
<contact body name>
```

Command: `contact_table_deactivate_all`

Description: This command deactivates all contact table entries for the current contact table. Inactive entries prevent Marc from checking for contact between contact bodies during a contact analysis. This button is only available in the SuperForm GUI.

Keyboard Command Sequence:
```
contact_table_deactivate_all
```

Command: `contact_table_deactivate_entry`

Description: These buttons are used for toggling the type of contact between the bodies in the current entry of the contact table. This button is only available in the SuperForm GUI.

- blank: The bodies will not contact each other.
- T: Contact will be detected between the bodies.
DISTANCE TOLERANCE

Command: contact_table_property
         <body1> <body2> dist_tol

Description: This command is used for setting the distance tolerance between the two current contact bodies.

This defines the size of the contact tolerance zone, i.e. the distance below which a node is considered touching a surface. The default value is 5% of the smallest element side in case of solid elements or 25% of the smallest beam or shell thickness. Note that the default values are based on all elements corresponding to any contact body.

The distance tolerance can also be input in the JOBS menus and is then valid for all contact bodies. If nonzero values are given in the contact table, they override the values given in the JOBS section.

Keyboard Command Sequence:

contact_table_property <contact body name>
  <contact body name>
  dist_tol <value>
**Contact Table Entry**

**Command:** `contact_table_entry`  
`<body1> <body2>`

**Description:** This command displays a menu in which for the current contact body combination various contact parameters can be defined.

**Keyboard Command Sequence:**
```
contact_table_entry <body number> <body number>
```

**Friction Coefficient**

**Command:** `contact_table_property`  
`<body1> <body2> friction`

**Description:** This command is used for setting the friction coefficient between the two current contact bodies.

Any nonzero value given here overrides any value given for the corresponding contact bodies.

The type of friction model to use is defined under CONTACT CONTROL in the JOBS menus.

**Keyboard Command Sequence:**
```
contact_table_property <contact body name>  
  <contact body name>  friction <value>
```
GLUE

Command:  `contact_table_glue`

Description: This command creates a glue entry for the specified contact bodies in the current contact table. A glue entry insures that Marc will enforce contact between the two bodies during a contact analysis. Contact bodies are specified by name. This button is only available in the Superform GUI.

Keyboard Command Sequence:
```
contact_table_glue <contact body name>
<contact body name>
```

GLUE ALL

Command:  `contact_table_glue_all`

Description: This command creates glue entries for all combinations of contact bodies in the current contact table. This button is only available in the SuperForm GUI.

The glue option ensures that Marc will detect contact between the bodies and additionally that the relative tangential motion is suppressed. The contacting body is *glued* to the contacted body. To avoid that the bodies separate, i.e. to really glue them together, use a very high separation force between the bodies.

Keyboard Command Sequence:
```
contact_table_glue_all
```
**CONTACT HEAT TRANSFER COEFFICIENT**

**Command:**  
`contact_table_property <body1> <body2> heat`

**Description:**  
This command is used for setting the contact heat transfer coefficient for the two current contact bodies. This coefficient is used whenever there is contact at the boundary of the contact bodies. Any nonzero value given here overrides any value given for the corresponding contact bodies.

**Keyboard Command Sequence:**

```
contact_table_property <contact body name>  
  <contact body name>  
  heat <value>
```

---

**INTERFERENCE CLOSURE**

**Command:**  
`contact_table_property <body1> <body2> closure`

**Description:**  
This command is used for setting the interference closure amount for the two current contact bodies.

A positive value of the interference closure implies that there is an overlap between the bodies; a negative value implies that a gap exists.

This option can for instance be used to simulate the effect of a shrink fit without using thermal strains. The bodies are modelled as just touching each other and the overlap of the shrink fit is input as a positive interference closure.

**Keyboard Command Sequence:**

```
contact_table_property <contact body name>  
  <contact body name>  
  closure <value>
```
Command: contact_table_name
Description: This command sets or changes the name of the current contact table.

Keyboard Command Sequence:

```
contact_table_name <contact table name>
```

Command: contact_table_option
Description: This command is used to set the contact type, the contact detection method, the behavior for glued contact, the coordinate changes upon initial contact, and the behavior at sharp corners, for a selected contact body pair.

The possible contact types are:

**NO CONTACT** *(contact_type: none):*
There will be no check for new contact between the contact bodies.

**TOUCHING** *(contact_type: touching):*
There will be a check on contact between the contact bodies. If a node is found to be in contact, the node is constrained in the direction normal to the contacted body.

**GLUE** *(contact_type: glue):*
There will be a check on contact between the contact bodies. If a node is found to be in contact, the node is constrained in the directions normal and tangential to the contacted body. This option might be useful to connect independently meshed parts of a structure.

The possible detection methods are:

**DEFAULT** *(detection: default):*
There will be a check on contact first for nodes of the first body with respect to the second body and then for nodes of the second body with respect to the first body. If single-sided contact is activated (this can be done in the JOBS menu), then there is only a check on contact for nodes of the first body with respect to the second body.
AUTOMATIC (detection:automatic):
The program figures out which of the two deformable contact bodies has
the smallest element edge at the outer boundary (or the smallest thickness
in case of shell elements). Then there will be only a check on contact for
nodes of this body with respect to the other body and not the other way
around.

FIRST->SECOND (detection:first_second):
This defines that there will only be a check on contact for nodes of the
first body with respect to the second body and not the other way around.

SECOND->FIRST (detection:second_first):
This defines that there will only be a check on contact for nodes of the
second body with respect to the first body and not the other way around.

DOUBLE-SIDED (detection:double_sided):
This overrules single-sided contact defined in the JOBS menu.

If the glue option active and a node is found to be in contact, but there is a
certain (small) distance between the node and the contacted body, forcing
the node on the contacted body might introduce locally high stresses. This
can be avoided by the option RETAIN GAPS/OVERLAPS
(retain_gaps:on or retain_gaps:off).

At initial contact (increment zero in an Marc analysis), nodes can be
found to be in contact, but there is a certain (small) distance between the
node and the contacted body. This gap or overlap can be removed by
changing the coordinates of the nodes in contact. In this way, stress-free
initial contact is obtained. The option to be used for this purpose is
PROJECT STRESS-FREE (project_stress_free:on or
project_stress_free:off).

If during the deformation process a node tends to slide off a deformable
body (so it passes a node (2-D) or edge (3) at a sharp corner), this local
loss of contact might deteriorate the overall solution. Using the option
DELAY SLIDE OFF (delay_slide_off:on or
delay_slide_off:off) this can be influenced, since activating
DELAY SLIDE OFF causes the program to tangentially extend the contacted
element edge (2-D) or face (3-D).

**Keyboard Command Sequence:**
```
contact_table_option <option:value>
```

Another Button with the same description:
```
PROJECT STRESS-FREE
```
Command: contact_table_option_all

Description: This command is used to easily set the contact type and the contact detection method between all contact body pairs defined in the current contact table.

The possible contact types are:

- NO CONTACT (contact_type:none):
  There will be no check for new contact between the contact bodies.

- TOUCHING (contact_type:touching):
  There will be a check on contact between the contact bodies. If a node is found to be in contact, the node is constrained in the direction normal to the contacted body.

- GLUE (contact_type:glue):
  There will be a check on contact for all the contact body pairs. If a node is found to be in contact, the node is constrained in the directions normal and tangential to the contacted body. This option might be useful to connect independently meshed parts of a structure.

The possible detection methods are:

- DEFAULT (detection:default):
  There will be a check on contact first for nodes of the first body with respect to the second body and then for nodes of the second body with respect to the first body. If single-sided contact is activated (this can be done in the JOBS menu), then there is only a check on contact for nodes of the first body with respect to the second body.

- AUTOMATIC (detection:automatic):
  The program figures out which of the two deformable contact bodies has the smallest element edge at the outer boundary (or the smallest thickness in case of shell elements). Then there will be only a check on contact for nodes of this body with respect to the other body and not the other way around.

- FIRST->SECOND (detection:first_second):
  This defines that there will only be a check on contact for nodes of the first body with respect to the second body and not the other way around.
SECOND->FIRST (detection:second_first):
This defines that there will only be a check on contact for nodes of the second body with respect to the first body and not the other way around.

DOUBLE-SIDED (detection:double_sided):
This overrules single-sided contact defined in the JOBS menu.

Keyboard Command Sequence:

  contact_table_option_all <option:value>

Other Buttons with the same description:

  AUTOMATIC
  DEFAULT
  DOUBLE-SIDED
  FIRST-
  GLUE
  SECOND-
  TOUCHING

Command:  contact_table_property

Description:  This command sets properties governing contact between two specified contact bodies. The contact bodies and the property are specified by name. This button is a text box and the button displays the value associated with this property.

The valid property names are as follows:

  dist_tol  Distance tolerance
  sep_for   Separation force
  friction  Friction coefficient
  closure   Interference closure amount, normal to contact surface
  heat      Heat transfer coefficient (coupled analyses only).

Keyboard Command Sequence:

  contact_table_property <contact body name> <contact body name> <property name> <property value>
ALLOW SEPARATION

Command: contact_table_property  
<body1> <body2> sep_for

Description: This command is used for setting the separation force between the two current contact bodies.

Separation occurs when the tensile force in the contact becomes larger than the separation force. This force is by default equal to the maximum residual force.

The separation force, as well as the option to use a separation stress instead, can also be input in the JOBS menus and is then valid for all contact bodies. If nonzero values are given in the contact table, they override the values given in the JOBS section.

Keyboard Command Sequence:

contact_table_property <contact body name>  
<contact body name>  
sep_for <value>

Other Buttons with the same description:

FORCE  
SEPARATION FORCE

TOUCH

Command: contact_table_touch

Description: This command creates a touch entry for the specified contact bodies in the current contact table. A touch entry insures that Marc will check for contact between the two bodies during a contact analysis. Contact bodies are specified by name. This button is only available in the SuperForm GUI.

Keyboard Command Sequence:

contact_table_touch <contact body name>  
<contact body name>
Command: contact_table_touch_all

Description: This command creates touch entries for all combinations of contact bodies in the current contact table, including self contact. This button is only available in the SuperForm GUI.

Touch entries insure that Marc will check for contact between bodies during a contact analysis.

Keyboard Command Sequence:

contact_table_touch_all

Menu: CONTACT TABLES

Description: This menu contains commands for defining contact tables.

A contact table is a set of entries which specifies the relationship between contact bodies in a contact analysis.

Contact tables can be used for different purposes, like:

• indicate which set of bodies may or may not touch each other, so that computational time can be saved;
• define different properties per set of contact bodies, like friction coefficient, error tolerance, separation force, and film coefficient;
• activate glued contact, which can be effectively used to couple separately meshed parts of a structure.

Note that the contact tables must be activated in the loadcase where they are to be used. This is defined in the CONTACT menu for the different loadcase types. Notice that if the user wants to deactivate existing contact between bodies, only selecting a different contact table in which contact between the relevant bodies is not allowed is not sufficient, you also have to release contact between those bodies.

For the correct detection of initial contact (before the first loadcase), the contact table should also be activated in the current job. This is done in the INITIAL CONTACT menu in the CONTACT CONTROL menu for each analysis class (e.g. for an uncoupled mechanical analysis, it is defined in...
JOBS-> MECHANICAL-> CONTACT CONTROL-> INITIAL CONTACT->
CONTACT TABLE).  
By default, if no contact table is used, every deformable body detects
possible contact with every other body including itself.

Reference:  *MSC.Marc Volume C: Program Input*, Chapter 3

**Command:**  `contact_value`

**Description:**  This command sets contact parameter values. You must specify the
parameter to be set by name and its value. This button is a text box and the
button displays the value associated with this property.

**Description:**

The available contact parameters are as follows:

- `cx, cy, cz` X, Y, and Z coordinates of the contact body center of
  rotation.
- `ax, ay, az` X, Y, and Z components of direction cosine relative to
  the center of rotation.
- `dx, dy, dz` X, Y, and Z components of velocity of the center of
  rotation.
- `rotation` Rotation angle of contact body (in radians).
- `friction` Friction coefficient.
- `film` Heat transfer (film) coefficient to environment.
- `tsink` Environment sink temperature.
- `contact` Contact heat (film) coefficient.
- `tsurf` Surface temperature. (Required for rigid surfaces only.)

**Keyboard Command Sequence:**

```
contact_value <parameter name> <parameter value>
```
Command: `contourmap`

Description: This command sets the current contour map for the program to be 1 through 8. There are eight predefined collections of colors that the program can be made to use for displaying contours with this command. Each of these collections maps a red, green, and blue value to each color index. By default, contour map 1 is used.

The command `PATRAN CONTOURMAP` will use a contour map that closely resembles the one used by Patran.

Keyboard Command Sequence:

`contourmap <1, 2, 3, 4, 5, 6, 7, or 8>`

Other Buttons with the same description:

2 through 8
DEF.
GRAY
PATRAN CONTOURMAP
REV. GRAY

Menu: `CONVERT`

Description: The `CONVERT` menu contains commands that convert existing geometrical entities, i.e. points, curves, and surfaces into their corresponding 2-D mesh elements.

The finite elements are attached to the specified geometric entities. If the geometry is changed, the finite element model is automatically modified.
**CURVES TO ELEMENTS**

**Command:** convert_curves

**Description:** This command converts curves into line elements. The newly created elements have their nodes attached to the curve.

CONVERT commands create 2-D meshes only. To expand 2-D meshes into 3-D, use EXPAND commands.

**Keyboard Command Sequence:**

convert_curves <curve list> #

**POINTS TO NODES**

**Command:** convert_points

**Description:** This command converts points into nodes. The finite element nodes are attached to the geometric points. If the geometric points are moved, the finite element node also moves.

**Keyboard Command Sequence:**

convert_points <point list> #

**SOLID FACES TO SURFACES**

**Command:** convert_solid_faces_surfaces

**Description:** This command converts faces in the solid model into Mentat NURBS surfaces with trim curves.

To convert Mentat surfaces into ACIS faces, use CONVERT TRIMMED SURFACES command.

**Note:** Occasionally when converting from pre-ACIS v5 models, there might be extra or missing trimming curves. When this happens, you can use unmap_curves and map_curves in conjunction with add_curves and remove_curves commands to edit the trimming curves.

If there are any inside out faces, this command will fail. You should first check the solid with the *check_solid_entities command and check the log file. If there are any errors, do not use this command on the solid.

**Keyboard Command Sequence:**

convert_solid_faces_surfaces <face list> #
**SURFACES TO ELEMENTS**

**Command:** `convert_surfaces`

**Description:** This command converts surfaces into quadrilateral elements. To expand 2-D elements into 3-D, use EXPAND commands.

**Keyboard Command Sequence:**
```
convert_surfaces <surface list> #
```

**TRIMMED SURFACES TO SOLID FACE**

**Command:** `convert_surfaces_solid_faces`

**Description:** This command converts Mentat NURBS surfaces with trim curves into ACIS faces. To convert ACIS faces into Mentat surfaces, use CONVERT SOLID FACES command.

**Keyboard Command Sequence:**
```
convert_surfaces_solid_faces <surface list> #
```

**SET**

**Menu:** SET (COORDINATE SYSTEM)

**Description:** This menu contains commands used to change the coordinate system. These commands allow the user to define the user coordinate system. This button is located in the MESH GENERATION menu.

**COPY**

**Command:** `copy_adapg`

**Description:** This command creates a new entry in the list of remeshing criteria by copying the current one. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

**Keyboard Command Sequence:**
```
copy_adapg
```
Command: **copy_adapt**
Description: This command creates a new adapt by copying the current one. This button is located in the MESH ADAPTIVITY-> LOCAL ADAPTIVITY menu.

**Keyboard Command Sequence:**
```
copy_adapt
```

Command: **copy_annotation**
Description: This command is used to replicate annotations. Annotations are strings of text that can be used to describe what is being displayed in a view. You must specify the annotation to copy, the view where you wish the new annotation to be placed, and the XY coordinates in that view. Views are specified by number, 1 - 4. This button is located in the UTILITIES-> MORE menu.

**Keyboard Command Sequence:**
```
copy_annotation <annotation number> <view> <X> <Y>
```

Command: **copy_apply**
Description: This command creates a new boundary condition application by copying the current one. This button is located in the BOUNDARY CONDITIONS menu.

**Keyboard Command Sequence:**
```
copy_apply
```
Command: **copy_autorange**

**Description:** This command sets the minimum and maximum values of the range limits to the values of the current increment. This is used when the RANGE is set to MANUAL (set_autorange off command).

Also see help: set_autorange and range.

**Keyboard Command Sequence:**

```
   copy_autorange
```

---

Command: **copy_cavity**

**Description:** This command creates a new cavity definition by copying the current one. It is located under the MESH GENERATION->CAVITIES menu.

**Keyboard Command Sequence:**

```
   copy_cavity
```

---

Command: **copy_contact_area**

**Description:** This command creates a new contact area by copying the current one. This button is located in the CONTACT->CONTACT AREAS menu.

**Keyboard Command Sequence:**

```
   copy_contact_area
```

---

Command: **copy_contact_body**

**Description:** This command creates a new contact body by copying the current one. This button is located in the CONTACT->CONTACT BODIES menu.

**Keyboard Command Sequence:**

```
   copy_contact_body
```
Command: **copy_contact_table**
Description: This command creates a new contact table by copying the current one. Upon copying, the new contact table becomes the current one. This button is located in the CONTACT->CONTACT TABLES menu.

**Keyboard Command Sequence:**
```plaintext
copy_contact_table
```

Command: **copy_crack**
Description: This command creates a new crack definition by copying the current one. This button is located in the FRACTURE MECHANICS->2D CRACKS and 3D->CRACKS menus.

Command: **copy_csect**
Description: This command creates a new cross-section definition by copying the current one. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

**Keyboard Command Sequence:**
```plaintext
copy_csect
```

Command: **copy_descon**
Description: This command creates a new design constraint by copying the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

**Keyboard Command Sequence:**
```plaintext
copy_descon
```
Command:  copy_desvar
Description:  This command creates a new design variable by copying the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

Keyboard Command Sequence:
  copy_desvar

Command:  copy_exseg
Description:  This command creates a new exclude segment by copying the current one. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

Keyboard Command Sequence:
  copy_exseg

Command:  copy_geometry
Description:  This command creates a new geometry by copying the current one. This button is located in the GEOMETRIC PROPERTIES menu.

Keyboard Command Sequence:
  copy_geometry

Command:  copy_icond
Description:  This command creates a new initial condition by copying the current one. This button is located in the INITIAL CONDITIONS menu.

Keyboard Command Sequence:
  copy_icond
Command:  copy_insert
Description:  This command creates a new insert by copying the current one.
Keyboard Command Sequence:
    copy_insert

Command:  copy_job
Description:  This command creates a new job by copying the current one. This button is located in the JOBS menu.
Keyboard Command Sequence:
    copy_job

Command:  copy_layered_materials
Description:  This command copies a block of layers and places them at the designated layer. You must specify the layers to be copied and the layer where the copied layers are to be placed. This button is located in the MATERIAL PROPERTIES->LAYERED MATERIAL->NEW COMPOSITE->DEFINE and NEW REBAR->DEFINE menus.
Keyboard Command Sequence:
    copy_layered_materials
    <from layer number> <to layer number>
    <destination layer number>
Command: copy_link
Description: This command creates a new link by copying the current one. This button is located in the LINKS->NODAL TIES, SERVO LINKS, and SPRINGS/DASHPOTS menus.

Keyboard Command Sequence:
    copy_link

Command: copy_loadcase
Description: This command creates a new loadcase by copying the current one. This button is located in the LOADCASES menu.

Keyboard Command Sequence:
    copy_loadcase

Command: copy_material
Description: This command creates a new material by copying the current one. This button is located in the MATERIAL PROPERTIES menu.

Keyboard Command Sequence:
    copy_material

Command: copy_orient
Description: This command creates a new orientation by copying the current one. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

Keyboard Command Sequence:
    copy_orient
Command:  copy_rbe2
Description:  This command creates a new rbe2 definition by copying the current one.
Keyboard Command Sequence:  
    copy_rbe2

Command:  copy_rbe3
Description:  This command creates a new rbe3 definition by copying the current one.
Keyboard Command Sequence:  
    copy_rbe3

Command:  copy_section
Description:  This command creates a new beam section by copying the current one.
This button is located in the GEOMETRIC PROPERTIES->MECHANICAL->BEAM SECTIONS menu.
Keyboard Command Sequence:  
    copy_section

Command:  copy_table
Description:  This command creates a new table by copying the current one. This button is located in the MATERIAL PROPERTIES->TABLES or any other link to the TABLES menu.
Keyboard Command Sequence:  
    copy_table
Command: **copy_to_clipboard**

**Description:** This command is only available on Windows NT. It is used to copy all the current tabular data from a PATH PLOT or a HISTORY PLOT to the Windows NT Clipboard, which can then be pasted into a Word or Excel document.

**Keyboard Command Sequence:**

```
copy_to_clipboard
```

Command: **copy_transform**

**Description:** This command creates a new transformation by copying the current one. This button is located in the BOUNDARY CONDITIONS->MECHANICAL menu.

**Keyboard Command Sequence:**

```
copy_transform
```

Command: **crack_name**

**Description:** This command sets or changes the name of the current crack.

**Keyboard Command Sequence:**

```
crack_name <name>
```
Command:  crack_option
          rigreg_method

Description: These commands are used for choosing the method for determining the
nodes in the so-called rigid region.

See help on the NEW button above for a description of the rigid region.

AUTOMATIC (TOPOLOGY SEARCH):
A number of rigid regions of increasing size is automatically determined
using the mesh topology (connectivity). The number of regions to define
is controlled by the RIGID REGIONS button.

2-D:
The first region consists of the nodes of all elements connected to the
crack tip. The next region consists of the nodes in the previous region plus
the nodes of all elements connected to a node in the previous region. If the
multiple tip nodes option is used, the first region will consist of all
elements connected to any node within the distance tolerance.

3-D:
The mesh around the crack front should consist of a regular mesh of
hexahedral elements, typically created by extruding a 2-D mesh along the
crack front. The rigid regions are created similarly to 2-D, but it is done
for each crack front node separately and the regions grow radially from
the crack front using only nodes on element faces. For a regular mesh, this
creates a disk of nodes perpendicular to the crack front. This will give
local values of the J-integral along the crack front.

AUTOMATIC (GEOMETRY SEARCH):
A number of rigid regions of variable size is determined by specifying a
radius. The number of regions to define is controlled by the RIGID
REGIONS button.

2-D:
The rigid region is defined as the nodes inside a circle of a given radius.

3-D:
The rigid region is defined as the nodes inside a cylinder of a given radius
and a relative length along the crack front. The relative length given is a
fraction of the distance between the current crack front node and the
previous and next node, respectively. The end points of the axis of the
cylinder are on the crack front at the positions defined by the relative length. This allows a similar type of rigid regions as for the topology based search to be defined for irregular meshes. The same radius and relative length are used for all nodes along the crack front.

**MANUAL:**
The nodes of the rigid region are explicitly given by the user. If a list of elements is available, use the SELECT NODES BY ELEMENTS option in the SELECT menu.

**Keyboard Command Sequence:**

```
crack_option rigreg_method:toplogy
```
```
crack_option rigreg_method:geometry
```
```
crack_option rigreg_method:direct
```

Other Buttons with the same description:

- AUTOMATIC (TOPOLOGY SEARCH)
- MANUAL

**Command:** `crack_option` 
**shift_method**

**Description:** These commands are used for choosing the method to define the shift direction. This is essentially a definition of the local crack tip system and the direction given is what is usually defined as the local x-direction. This should be perpendicular to the crack face near the crack tip.

**AUTOMATIC:**
The shift direction is determined automatically.

**2-D:**
The shift direction is determined using the element edges for elements connected to the crack tip. The element edge which is on the crack surface is used for defining the shift direction, which is from the edge node on the crack surface to the crack tip node.

**3-D:**
The shift direction is determined to be perpendicular to both the crack surface and the crack front. The shift direction is projected to the outer
surface of the model at the end points of the crack. Both the tangent to the
 crack front and the normal to the crack surface are determined locally
 along the crack front.

Note that the automatic option also works for a symmetric crack when the
 symmetry is applied with a rigid contact body.

Also note that the multiple tip nodes distance should be used also for the
 geometrically based rigid region method if multiple nodes are used at the
 crack tip. This allows the proper element faces to be used for determining
 the crack normal.

**MANUAL:**

The shift vector is explicitly given by the user.

**Keyboard Command Sequence:**

```
crack_option shift_method:automatic
```

Another Button with the same description:

**MANUAL**

---

**Command:** `crack_option symmetry`

**Description:** This command sets the symmetry flag. If this is set, the value of the
 J-integral will be multiplied by two to account for symmetry where only
 half of the crack is modelled.

This flag is automatically set if automatic determination of the rigid
 regions is used.

**Keyboard Command Sequence:**

```
crack_option symmetry:<on/off>
```
**LENGTH**

Command: `/crack_param geom_srch_cyl_l`

Description: This command is used to define the length of a cylinder which is used for defining nodes in the current rigid region. All nodes within the cylinder will be part of the rigid region. The length is defined as a fraction of the distance between the current crack front node and the previous and the next, respectively.

Using the default value of 0.5, the cylinder axis would start at the mid-point between the current crack front node and the previous node along the crack front and end at the mid-point between the current node and the next.

Keyboard Command Sequence:
```
crack_param geom_srch_cyl_l <value>
```

**RADIUS**

Command: `/crack_param geom_srch_r`

Description: This command is used to define the radius of the current rigid region. All nodes within a circle (cylinder in 3-D) with this radius will be part of the rigid region.

Keyboard Command Sequence:
```
crack_param geom_srch_r <value>
```
RIGID REGIONS

Command: crack_param
num_rig_regions

Description: This command is used to define the number of rigid regions to be defined for the current crack. For 3-D, this is the number of rigid regions to be defined for each crack front node in case of the automatic search options.

Keyboard Command Sequence:

    crack_param num_rig_regions <value>

Command: crack_param
shift_vector

Description: These commands are used for entering the shift vector. This is essentially a definition of the local crack tip system and the direction given is what is usually defined as the local x-direction. This should be perpendicular to the crack face near the crack tip. For 3-D, it should also be perpendicular to the crack front.

Keyboard Command Sequence:

    crack_param shift_x <value>
    crack_param shift_y <value>
    crack_param shift_z <value>

Other Buttons with the same description:

    Y
    Z
**DISTANCE**

**Command:** crack_param
tip_node_dist_tol

**Description:** This command is used to define the tolerance for multiple crack tip nodes. All nodes within this distance from the crack tip node are considered part of the crack tip. The first rigid region for the automatic topology search will consist of nodes connected to any of these nodes. These nodes are also used for automatically determining the shift vector.

This option is useful if a mesh with collapsed elements at the crack tip with separate nodes at the same location or defining a small notch is used for allowing a notch to be created at the crack tip. This type of mesh is typically used in elastoplastic analyses.

**Keyboard Command Sequence:**
```
crack_param tip_node_dist_tol <value>
```

**SET**

**Command:** set_crack_tip_node

**Description:** This command is used to define the current crack tip by selecting the crack tip node. The node number of this node is shown in the window next to the button.

**Keyboard Command Sequence:**
```
set_crack_tip_node <node>
```

**CLEAR**

**Command:** clear_crack_tip_node

**Description:** This command clears the current selection for the crack tip node.

**Keyboard Command Sequence:**
```
clear_crack_tip_node
```
Command: **set_crack_tip_node_path**

**Description:** This command is used to define the current crack front. Select the start and end points of the crack and an unsorted list of nodes defining the crack front will be created.

The number of nodes defined is shown in the window next to the button. These nodes are automatically defined as a node set:

```
crackname_nodes
```

where **crackname** is the name of the current crack.

**Keyboard Command Sequence:**

```
set_crack_tip_node_path <node list> #
```

---

Command: **clear_crack_tip_node_path**

**Description:** This command clears the current selection for the crack tip node path.

**Keyboard Command Sequence:**

```
clear_crack_tip_node_path
```

---

**Menu:** **2-D CRACKS / 3-D CRACKS**

**Description:** These menus contain commands used for defining cracks for a J-integral evaluation.

The concept of a crack is a crack tip for 2-D and a crack front in 3-D. For each crack, Marc evaluates the J-integral and prints the results to the output file.

For 2-D, the J-integral is defined as a line integral along a path surrounding the crack tip. In the method used by Marc, this integral is transformed into an area integral where the area to be integrated over is the one enclosed by the line integration path. Only a part of this area is...
actually used for the integration, usually the outermost \textit{ring} of elements. In this way, only results values at a distance from the crack tip are used and more accurate results can be obtained for the J-integral.

To identify the integration area, the user defines a list of nodes inside the elements that are used for the integration. These nodes define the RIGID REGION which the menus below refer to. Only elements that are connected to a node in the rigid region contribute to the J-integral value calculated. This does not include elements with all nodes in the rigid region. The term \textit{rigid region} comes from the fact that you can evaluate the J-integral by shifting the nodes in the rigid region and calculate the change in strain energy in the elements connected to a node in the rigid region.

For 3-D, the line integral becomes a surface integral over a surface surrounding the crack front (or a part of the crack front). The area integral becomes a volume integral and the rigid region is similarly defined as a list of nodes. By choosing the rigid region as a thin \textit{disk} of nodes perpendicular to the crack front, where the disk contains one node along the crack front, you can obtain local values of the J-integral along a crack front. The automatic search routines for the rigid regions are designed to create this type of rigid regions.

Shell elements are not supported in this release.

Another Button with the same description:

3-D CRACKS

---

**Command:** create\_procedure

**Description:** This command creates a new procedure file. All subsequent user input will be recorded in the given file. Procedures may include calls to other procedures (see exec\_procedure) which may in turn call other procedures.

**Keyboard Command Sequence:**

create\_procedure <file name>
Subroutine: CREDE

Description: The user subroutine CREDE allows you to input pre-specified state variables. The simplest option allows you to specify temperature increments throughout the mesh. Through the use of the STATE VARS parameter card, the number of state variables per point in the structure may be increased. For example, radiation fluxes (in reactor core problems) pore pressures (in soil mechanics) may be included. The program always assumes temperature is the first state variable given at a particular point or since the first state variable is used in conjunction with the tables of temperature dependence input specified in the TEMPERATURE EFFECTS option, and the first state variable is used to compute thermal strains. All state variables are available to all constitutive routines – see subroutines CRPLAW and VSWELL later in this section.

Subroutine CREDE will be called once per element in a loop over the elements when the THERMAL LOADS option is activated.

Subroutine: CRPLAW

Description: For creep analysis, the user subroutine CRPLAW allows you to specify the creep strain rate. CRPLAW will be called as required during the analysis.

For viscoplasticity analysis, the user subroutine CRPLAW may also be used for calculating the viscoplastic behavior. All the creep quantities are treated as viscoplastic strain quantities when the appropriate flag is set in the CREEP option.
Subroutine: CRPVIS

Description: The user subroutine CRPVIS allows you to include a general Kelvin model (in addition to the nonlinear Maxwell-type model allowed in the CREEP option).

Command: csect_name

Description: This command sets or changes the name of the current cross-section.

Keyboard Command Sequence:

```
csect_name <cross-section name>
```

Command: csect_param

```
normal_x/normal_y/normal_z
```

Description: These commands are used to define the cross-section normal vector.

Keyboard Command Sequence:

```
csect_param <normal_x/normal_y/normal_z> <value>
```

Other Buttons with the same description:

```
Y
Z
```

Subroutine: CUPFLX

Description: The user subroutine CUPFLX allows you to modify the default routine for the calculation of the internal heat generated due to inelastic energy dissipation. This routine is only used if a coupled thermal-mechanical analysis is being performed and a distributed FLUX type 101 is chosen.
**RESTRICTIONS**

**Description:** Curve Division Restrictions

These controls are used to establish an even/odd curve division restriction that will be used when the apply_curve_divisions command is invoked. This is useful for automatic quadrilateral meshing, as the total number of divisions on all curves forming a closed loop must be even to guarantee an all-quad mesh.

If the apply_curve_divisions command is invoked when there is an even/odd restriction established, the number of divisions formed on the chosen curves will be increased by one as necessary to conform with the restriction.

The even/odd restriction may be placed on the individual curves or on detected loops within the given curves. In the latter case, the restriction is only enforced on entire closed loops and individual curves may not necessarily meet the criteria. This is adequate for the purposes of 2-D quad meshing, but problems may arise in surface meshing when trying to match divisions for corresponding curves on adjacent surfaces.

Also see help: curve_div_type and apply_curve_divisions.

**Keyboard Command Sequence:**

```plaintext
set_curve_div_rest_off
set_curve_div_rest_evn
set_curve_div_rest_odd
set_curve_div_applyrest_cvs
set_curve_div_applyrest_lps
```

Other Buttons with the same description:

- NONE
- FORCE ODD DIV CURVES
Menu: TYPE

Description: Curve Division Type Controls

The following controls are used to establish the type and associated parameters for divisions along curves. Curve divisions are used as boundary element edges in automatic meshing. These controls are used when the apply_curve_divisions command is invoked. See the keyboard command sequences at the bottom of this document for the commands associated with setting these parameters.

Three curve division types are currently supported: fixed number of divisions, fixed average length, and curvature-dependent (variable). Only one of these types may be selected at any one time. They are described below along with their associated parameters.

Fixed Number of Divisions
The number of divisions along the curve will be fixed by the number of divisions parameter. All segments will have the same length along the curve.

Fixed Average Length
Each segment along the curve will be set as close as possible to the value of the average length parameter.

Curvature-Dependent
With this type, the length of each segment will be dependent on the curvature along the curve, and will be set in accordance with the following parameters:

- Min Length: The minimum length of generated segments.
- Max Length: The maximum length of generated segments.
- Tolerance: The maximum allowed deviation of the segment to the curve.
- Relative/Absolute: Specifies whether the tolerance is relative to the length of the curve or is an absolute distance.

Also see help: curve_div_rest and apply_curve_divisions.

Keyboard Command Sequence:

set_curve_div_type_fix_ndiv
set_curve_div_num <number of divisions>
set_curve_div_type_fix_avgl
set_curve_div_avgl <average length>
set_curve_div_type_variable
set_curve_div_minl <min segment length>
set_curve_div_maxl <max segment length>
set_curve_div_tol <tolerance>
set_curve_div_tol_rel
set_curve_div_tol_abs

Other Buttons with the same description:

FIXED # DIVISIONS
FIXED AVG_LENGTH
AVG LENGTH
CURVATURE DEPENDENT
MIN LENGTH
MAX LENGTH
TOLERANCE
REL

Menu: CURVE DIVISIONS
Description: This menu contains commands used to set division points on existing curves. The resulting curve segments are used as element edges in automatic meshing.

Command: curve_interpolated
Description: This command converts curves into interpolated curves. The original curve is sampled at the user defined subdivisions and an interpolated curve is passed through them. The resulting curve is a cubic NURBS curve with C1 and C2 continuity.

Keyboard Command Sequence:

curve_interpolated <curve list> #
**CURVES TO POLYLINES**

**Command:** curve_polylines

**Description:** This command converts curves into polylines.

**Keyboard Command Sequence:**

```
curve_polylines <curve list> #
```

---

**CURVE TYPE**

**Menu:** CURVE TYPE

**Description:** This menu contains commands for setting the type of geometric curves to be created with the `add_curves` command.

Available curve types are as follows:

**Miscellaneous types:**

- **LINE**
  A straight line between two points.

- **BEZIER**
  A bezier curve defined by a list of control points. Note that the curve will in general not pass through the control points.

- **CUBIC SPLINE**
  A cubic spline curve that interpolates a list of vertex points.

- **INTERPOLATE**
  A cubic NURBS curve that interpolates a list of vertex points.

- **NURB**
  A general NURBS curve given by a complete definition: the number of control points, the curve order, a list of control points, a list of homogeneous coordinates, and a knot vector.

- **POLYLINE**
  A piecewise straight line between given points.

- **COMPOSITE**
  A curve that combines a list of curves into one single curve. The curves must lie end to end to each other and can be given in any order.
TANGENT  A straight line that is tangent to an existing curve. Input is an end point of a curve and the length of the new curve.

FILLET  A fillet curve (radius) between two curves. Input is the two curves and the fillet radius. The curves are shortened or extended in order to create the fillet. Note that the orientation of the curves is important, the fillet starts at the head of the first curve and continues at the tail of the second curve. Thus, it may be necessary to flip the orientation of the curves.

SAMPLED  A straight line that connects a sequence of points. It is useful for creating curves from digitized data. Input is the origin point of the curve and the first point (to establish the direction of the curve) and then all the points that define the curve (including selecting the origin point and the first point again).

Arcs:

CENTER/RADIUS/ANGLE/ANGLE  An arc by giving the center, radius, and starting and ending angles.

CENTER/POINT/POINT  An arc by giving the center, starting point, and point defining the angle.

CENTER/POINT/ANGLE  An arc by giving the center, starting point, and angle.

POINT/POINT/POINT  An arc by giving the starting, ending, and intermediate points.

TANGENT/RADIUS/ANGLE  An arc by giving the ending point of another curve, the radius and the angle.

Circles:

CENTER/RADIUS  A circle by giving the center and the radius.

CENTER/POINT  A circle by giving the center and a point on the periphery.
Mentat Help Commands in D

CONTINUOUS DAMAGE

Command: \texttt{xcv\_model damage}

Description: This command is used if experimental data must be fitted using the Kachanov factor $K$, which is given by:

$$K(W) = d_{\infty} + \sum_{n=1}^{N} d_n \exp(-W/\lambda_n)$$

For continuous damage, $W$ represents the arc length of the effective strain energy density. In this case, the input data should be obtained by a constant strain amplitude test and given in terms of cycle number versus strain energy density. In addition, the free energy (corresponding to the undamaged state) at the fixed strain level is required.

For discontinuous damage, $W$ represents the maximum attained strain energy density. In this case, the input data should be obtained by an increasing strain amplitude test and given in terms of strain energy versus the ratio of the strain energy density and the strain energy density corresponding to the undamaged state.
The maximum number of terms (N) is 2.
The actual damage factor applied by Marc can never be larger than 1, so:

\[
\text{if } K(W) < 1 \text{ then } K = K(W), \text{ else } K = 1
\]

**Keyboard Command Sequence:**

xcv_model damage

Another Button with the same description:

DISCONTINUOUS DAMAGE

---

**Command:**  
activate_view  
deactivate_view

**Description:** These commands are used for multiple view management. Only active views are affected by view setting commands. The visibility of views is controlled by `show_view` and the `show_all_views` commands. Views may be visible but not active and vice versa. When the program is started, view 1 is the only active and visible view.

You must specify the view to be operated on. Views are specified by number, 1-4.

When a view is made visible, it is automatically activated and when a view is made invisible, it is automatically deactivated. The three commands `activate_all_views`, `activate_view`, and `deactivate_view` allow you to override this.

Also see: `activate_all_views` and `show_view`.

**Keyboard Command Sequence:**

activate_view <view number>  
deactivate_view <view number>
**Command:** define

**Description:** This command defines a new parameter or redefines an existing parameter. Parameters may be used as input to commands requiring data. Preceding the parameter with a dollar sign ($) forces the parameter to be expanded in the case where a string is being input. You must specify the parameter name and a value for it.

**Keyboard Command Sequence:**

```
define <name> <value>
```

**Menu:** SETTINGS (DEFORMED SHAPE)

**Description:** This menu contains commands for setting deformed shape plotting attributes such as auto/manual magnification.

**Command:** descon_algebraic  
descon_absolute

**Description:** These commands determine whether the constraint is placed on the algebraic or absolute value of the response quantity. These commands are only used for displacement, stress, and strain class design constraints.

**Keyboard Command Sequence:**

```
descon_algebraic  
descon_absolute
```
**Command:**  `descon_add_elements`
`descon_remove_elements`

**Description:** Specifies the elements that a stress or strain design constraint is to be applied to. This button is located in the DESIGN->DESIGN CONTRAINTS menu.

**Keyboard Command Sequence:**
```
descon_add_elements <element list>  #
descon_remove_elements <element list>  #
```

Another Button with the same description:
```
REM
```

---

**Command:**  `descon_add_nodes`
`descon_remove_nodes`

**Description:** Specifies the nodes that a nonrelative displacement design constraint is to be applied to. This button is located in the DESIGN->DESIGN CONTRAINTS menu.

**Keyboard Command Sequence:**
```
descon_add_nodes <node list>  #
descon_remove_nodes <node list>  #
```

Another Button with the same description:
```
REM
```
ALGEBRAIC

Command:  descon_algebraic  
descon_absolute

Description:  These commands determine whether the constraint is placed on the 
algebraic or absolute value of the response quantity. These commands are 
only used for displacement, stress, and strain class design constraints.

Keyboard Command Sequence:

    descon_algebraic  
descon_absolute

BOUND

Command:  descon_bound

Description:  Sets the bound for the design constraint response quantity value.

Keyboard Command Sequence:

    descon_bound <response quantity bound>

DISPLACEMENT

Command:  descon_class

Description:  Sets the class of the current design constraint. The allowable design 
constraint classes are as follows:

    displacement  
stress  
strain  
eigenvalue

Keyboard Command Sequence:

    descon_class <design constraint class>

Other Buttons with the same description:

    EIGENFREQ  
    STRAIN  
    STRESS
**CYCLES/TIME**

Command:  

```
    descon_cycles
descon_radians
```

Description: Specifies whether the bound given for an eigen-frequency design constraint is given in cycles per time or radians per time.

Keyboard Command Sequence:  

```
descon_cycles
descon_radians
```

**VECTOR DIRECTION**

Command:  

```
    descon_direction
```

Description: Sets the vector direction or plane normal for design constraint types that require them:

<table>
<thead>
<tr>
<th>vector</th>
<th>plane normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>displacement:dirtransl</td>
<td>stress:normal_stress</td>
</tr>
<tr>
<td>displacement:dirrotat</td>
<td>stress:plane_shear</td>
</tr>
<tr>
<td>displacement:dirreltransl</td>
<td></td>
</tr>
<tr>
<td>displacement:dirrelrotat</td>
<td></td>
</tr>
</tbody>
</table>

The program will calculate direction cosines from the vector provided by the user.

Keyboard Command Sequence:  

```
descon_direction <x-component> <y-component> <z-component>
```
Command: descon_less_than
descon_more_than

Description: Specifies whether the constraint response quantity value is to be less or more than the given bound.

Keyboard Command Sequence:

descon_less_than
descon_more_than

Command: descon_mode_1

Description: Specifies the mode that an eigenfrequency design constraint is to be applied to. If the eigenfrequency constraint is one related to the separation between two modes (FREQ DIFF), then this is the first one of the two modes. In this second case, both MODE buttons will have been highlighted.

Keyboard Command Sequence:

descon_mode_1 <mode number>

Command: descon_mode_2

Description: This button specifies the second of a pair of eigenfrequency modes for which the modal separation is constrained (FREQ DIFF type constraint).

Keyboard Command Sequence:

descon_mode_2 <mode number>
Command: descon_name
Description: This command sets or changes the name of the current design constraint.

A design constraint is an analysis response quantity that is constrained to be above or below a specified value, in order for the design to be acceptable.

Keyboard Command Sequence:

descon_name <design constraint name>

Command: descon_node_1
descon_node_2
Description: Specifies the nodes that a relative displacement design constraint is to be applied to.

Keyboard Command Sequence:

descon_node_1 <node id>
descon_node_2 <node id>

Another button with the same description:

NODE 2
Command: **descon_type**

Description: Sets the type of the current design constraint. The allowable design constraint types for each class are as follows:

<table>
<thead>
<tr>
<th>Class</th>
<th>Allowable Design Constraint Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>displacement</strong></td>
<td><strong>stress</strong></td>
</tr>
<tr>
<td>transl1</td>
<td>stress1</td>
</tr>
<tr>
<td>transl2</td>
<td>stress2</td>
</tr>
<tr>
<td>transl3</td>
<td>stress3</td>
</tr>
<tr>
<td>rotat1</td>
<td>stress4</td>
</tr>
<tr>
<td>rotat2</td>
<td>stress5</td>
</tr>
<tr>
<td>rotat3</td>
<td>stress6</td>
</tr>
<tr>
<td>retransl1</td>
<td>genstress1</td>
</tr>
<tr>
<td>resrotat</td>
<td>genstress2</td>
</tr>
<tr>
<td>dirtransl1</td>
<td>genstress3</td>
</tr>
<tr>
<td>dirrotat</td>
<td>genstress4</td>
</tr>
<tr>
<td>reltransl1</td>
<td>genstress5</td>
</tr>
<tr>
<td>reltransl2</td>
<td>genstress6</td>
</tr>
<tr>
<td>reltransl3</td>
<td>genstress7</td>
</tr>
<tr>
<td>relrotat1</td>
<td>genstress8</td>
</tr>
<tr>
<td>relrotat2</td>
<td>genstress9</td>
</tr>
<tr>
<td>relrotat3</td>
<td>vommises</td>
</tr>
<tr>
<td>dirretransl</td>
<td>principal1</td>
</tr>
<tr>
<td>dirresrotat</td>
<td>principal2</td>
</tr>
<tr>
<td></td>
<td>principal3</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Keyboard Command Sequence:

```
descon_type
<design constraint class>:<design constraint type>
```

Other Buttons with the same description:

- DIRECTED REL ROTATION
- DIRECTED REL TRANSLATION
- DIRECTED ROTATION
- DIRECTED TRANSLATION
- FIRST PRINCIPAL
- FREQUENCY DIFFERENCE
- GENERALIZED STRESS 1 through GENERALIZED STRESS 9
- MAJOR PRINCIPAL
- MAX SHEAR STRESS ON A PLANE
- MODAL FREQUENCY
- NORMAL STRESS ALONG A VECTOR
Menu: DESIGN

Description: By means of this button, the user enters the area where the design variables and design constraints can be specified for a design optimization or design sensitivity analysis job.

Design optimization refers to the process by which certain parameters (design variables) of a finite element model are modified in order to reach a feasible and desirable (improved) design. Design sensitivity, in the context of MSC.Marc, refers to the process whereby first order derivatives of response quantities with respect to each of the design variables are computed and output. Element contributions to each response quantity are also computed and are made available.

The design objective (objective function) is specified via the JOBS menu.
**DESIGN CONSTRAINTS**

**Menu:** DESIGN CONSTRAINTS

**Description:** This button allows access to the DESIGN CONSTRAINTS menu. Design constraints can be described by the user using the various tools offered in this menu.

A design constraint is an analysis response quantity that is constrained to be above or below a specified value, in order for the design to be acceptable.

---

**DESIGN VARIABLES**

**Menu:** DESIGN VARIABLES

**Description:** This button allows access to the DESIGN VARIABLES menu. Design variables can be described by the user using the various tools offered in this menu.

A design variable is the finite element model equivalent of a design parameter which can be varied in order to make changes in a given design.

---

**layer1**

**Command:** desvar_add_elements

**Description:** Adds elements to the current geometry class design variable. The specified element property (set with the desvar_type command) will vary for the selected elements during an optimization or sensitivity analysis. The button text will be the name of the layer. This button is located in the DESIGN->DESIGN VARIABLES->SELECT LAYERS menu.

**Keyboard Command Sequence:**

desvar_add_elements <element list> #
Command: desvar_add_layer
desvar_remove_layer

Description: These commands add and remove composite material layers from the current composite class design variable. The specified layer property (set with the design_type command) will vary for the selected layers during an optimization or sensitivity analysis.

Keyboard Command Sequence:

```
desvar_add_layer
    <composite material layer number>
desvar_remove_layer
    <composite material layer number>
```

Command: desvar_class

Description: This command sets the class of the current design variable. The allowable design variable classes are as follows:

```
material
composite
geometry
```

Keyboard Command Sequence:

```
desvar_class <design variable class>
```

Other Buttons with the same description:

```
GEOMETRY
MATERIAL
```
Command: **desvar_composite**

**Description:** This command sets the composite material for the current composite class design variable. This material will have one of its composite layer properties varied during an optimization or sensitivity analysis. Use the **desvar_type** command to specify which property to vary.

**Keyboard Command Sequence:**

```
   desvar_composite <composite material name>
```

---

Command: **desvar_linked**

**Description:** Specifies whether one or multiple independent design variables should be generated for the optimization or sensitivity analysis.

This command is only valid for composite and geometry class design variables. If the design variable is linked, then one design variable will be generated for the analysis which varies all specified composite layers or geometric element together. If unlinked, separate design variables will be generated for each composite layer or geometric element allowing each layer or element to vary independently.

**Keyboard Command Sequence:**

```
   desvar_linked <on or off>
```

Another Button with the same description:

**UNLINKED**
**LOWER BOUND**

**Command:** desvar_lower  
**Description:** This command sets the lower bound on the current design variable. Design variable bounds are specified as floating point numbers.  
**Keyboard Command Sequence:**

```
desvar_lower <lower bound>
```

---

**SEL MATERIAL**

**Command:** desvar_material  
**Description:** This command sets the material for the current material class design variable. This material has one of its properties varied during an optimization or sensitivity analysis. Use the desvar_type command to specify which property to vary.  
**Keyboard Command Sequence:**

```
desvar_material <material name>
```

---

**NAME**

**Command:** desvar_name  
**Description:** This command sets or changes the name of the current design variable. A design variable is the finite element model equivalent of a design parameter which can be varied in order to make changes in a given design.  
**Keyboard Command Sequence:**

```
desvar_name <design variable name>
```
**Command:**  **desvar_remove_elements**

**Description:** This command removes elements from the current geometry class design variable.

**Keyboard Command Sequence:**

```
desvar_remove_elements <element list> #
```

---

**Command:**  **N/A**

**Description:** This button takes the user to the pop-up menu which is used to select which layers of the composite material, chosen with the SEL COMPOSITE button that are associated with the current design variable of type COMPOSITE.

**Keyboard Command Sequence:**

```
N/A
```

---

**Command:**  **desvar_type**

**Description:** Sets the type of the current design variable. The allowable design variable types for each class are as follows:

<table>
<thead>
<tr>
<th><strong>material</strong></th>
<th><strong>composite</strong></th>
<th><strong>geometry</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>youngs_modulus</td>
<td>ply_angle</td>
<td>area</td>
</tr>
<tr>
<td>poissons_ratio</td>
<td>layer_thickness</td>
<td>ixx</td>
</tr>
<tr>
<td>mass_density</td>
<td></td>
<td>iyy</td>
</tr>
<tr>
<td>e11</td>
<td></td>
<td>beam_height</td>
</tr>
<tr>
<td>e22</td>
<td></td>
<td>beam_width</td>
</tr>
<tr>
<td>e33</td>
<td></td>
<td>pipe_radius</td>
</tr>
<tr>
<td>nu12</td>
<td></td>
<td>wall_thickness</td>
</tr>
<tr>
<td>nu23</td>
<td></td>
<td>thickness</td>
</tr>
<tr>
<td>nu31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Keyboard Command Sequence:**

```
  desvar_type <design variable class>:  
      <design variable type>
```

Other Buttons with the same description:

- BEAM HEIGHT
- BEAM WIDTH
- CONSTANT THICKNESS
- E11
- E22
- E33
- G12
- G23
- G31
- IXX
- IYY
- LAYER THICKNESS
- MASS DENSITY
- NU12
- NU23
- NU31
- PLY ANGLE
- RADIUS
- WALL THICKNESS

---

**Command:** `desvar_upper`

**Description:** This command sets the upper bound on the current design variable. Design variable bounds are specified as floating point numbers.

**Keyboard Command Sequence:**

```
  desvar_upper <upper bound>
```

---

**Command:** `detach_edges`

**Description:** This command removes the association between the edges in the edge list and the curve(s) to which they are attached. If an edge is detached from a curve, changes made to the curve no longer influence the position of the nodes of the edge, and initial or boundary conditions applied to the curve are no longer inherited by the edge.
Also see: attach_edges_curve.

**Keyboard Command Sequence:**

```
detach_edges <edge list> #
```

---

**Command:** `detach_elements`

**Description:** This command removes the association between the elements in the element list and the underlying geometry. The nodes of the elements will be detached from the points to which they are attached, the edges of the elements will be detached from their curves, and the element faces will be detached from their surfaces.

Also see: `detach_nodes`, `detach_edges`, and `detach_faces`.

**Keyboard Command Sequence:**

```
detach_elements <element list> #
```

---

**Command:** `detach_faces`

**Description:** This command removes the association between the faces in the face list and the surface(s) to which they are attached. If a face is detached from a surface, changes made to the surface no longer influence the position of the nodes of the face, and initial or boundary conditions applied to the surface are no longer inherited by the face.

Also see: `attach_faces_surface`.

**Keyboard Command Sequence:**

```
detach_faces <face list> #
```
Command: **detach_nodes**

Description: This command removes the association between the nodes in the node list and the point(s) to which they are attached. If an node is detached from a point, changes made to the point no longer influence the position of the node, and initial or boundary conditions applied to the point are no longer inherited by the node.

Also see: **attach_nodes_point**.

**Keyboard Command Sequence:**

```
detach_nodes <node list> #
```

---

Menu: **DEVICE**

Description: This menu contains commands used to set and/or change the appearance of the MSC.Marc Mentat menus.

---

Subroutine: **DIGEOM**

Description: The user subroutine **DIGEOM** allows you to enter the geometry directly in three-dimensional problems in which complicated rigid surfaces need to be entered.

Rigid surfaces are normally entered by means of several geometrical entities, which in turn are internally subdivided into 4-point patches. You may substitute one or more entities directly with the patches that will make up such entities.
LIST DIRECTORY

Command: directory
Description: This command displays a listing of the files contained in a directory. You must specify the directory path.
Keyboard Command Sequence:
   directory <path>

UNORIENT

Command: disorient_elements
Description: This command removes orientation for the specified elements.
Keyboard Command Sequence:
   disorient_elements <elements list> #

DISTANCE

Command: distance
Description: The distance utility calculates the distance between two given points. The command also prints the three components of the vector extending from the first point to the last as well as the angles between that vector and the three coordinate axes. You must specify two sets of XYZ coordinates.
Keyboard Command Sequence:
   distance <X Y Z coordinates> <X Y Z coordinates>
ADD ELEMENTS

Command:  **domain_add_elements**

Description:  This command adds elements to an element domain set.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the *MSC.Marc User’s Guide* or call MSC.Software Customer Support.

Keyboard Command Sequence:

```plaintext
domain_add_elements <domain name> <element list>
```

CREATE DOMAIN

Command:  **domain_create**

Description:  This command creates a new element domain set. Domain sets are stored as element sets in the database with names beginning with `domain`. The domain name specified by the user is appended to this prefix.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the *MSC.Marc User’s Guide* or call MSC.Software Customer Support.

Keyboard Command Sequence:

```plaintext
domain_create <domain name>
```
DELETE DOMAIN

Command:   `domain_delete`

Description:  This command deletes an element domain set.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the *MSC.Marc User’s Guide* or call MSC.Software Customer Support.

Keyboard Command Sequence:

```
domain_delete <domain name>
```

REM ELEMENTS

Command:   `domain_remove_elements`

Description:  This command removes elements from an element domain set.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the *MSC.Marc User’s Guide* or call MSC.Software Customer Support.

Keyboard Command Sequence:

```
domain_remove_elements <domain name>
<element list>
```
RESET DOMAIN

Command: domain_reset

Description: This command removes all elements from an element domain set. It does not delete the domain set.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the MSC.Marc User’s Guide or call MSC.Software Customer Support.

Keyboard Command Sequence:

domain_reset <domain name>

SELECT DOMAIN

Command: domain_select

Description: This command selects the elements in the specified domain set.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information see the MSC.Marc User’s Guide or call MSC.Software Customer Support.

Keyboard Command Sequence:

domain_select <domain name>
Command: **domains_clear**

**Description:** This command deletes all domain element sets.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the MSC.Marc User’s Guide or call MSC.Software Customer Support.

**Keyboard Command Sequence:**

```plaintext
domains_clear
```

---

Command: **domains_decompose**

**Description:** This command generates domain element sets. The approach is to use element to element connectivity information to minimize the surface area of each domain. The algorithm includes *annealing* and *island elimination*.

*Annealing* is done by moving elements from one domain to another if their connectivity to that domain is greater. This has the effect of smoothing the boundaries between domains.

*Island elimination* is performed by merging disconnected groups of domain elements with the closest neighboring domains. This is only done if the number of elements in the island is small enough not to cause an imbalance in the domain sizes.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.
For more information, see the MSC.Marc User’s Guide or call MSC.Software Customer Support.

**Keyboard Command Sequence:**

`domains_decompose <number of domains>`

---

**Command:** `domains_generate`

**Description:** This command generates domain element sets. Elements are grouped into sets based on their id. The user specifies the number of domains to create. `Renumber_elements_directed` may be used to renumber the elements to help control the grouping of elements within domains.

Domain decomposition is a special MSC.Marc feature that enables MSC.Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information, see the MSC.Marc User’s Guide or call MSC.Software Customer Support.

**Keyboard Command Sequence:**

`domains_generate <number of domains>`

---

**Command:** `draw`

**Description:** This command allows you to display changes made to your model. If the program is in manual draw mode (see `draw_manual` and `draw_automatic`), you must enter the draw command to cause model changes to be reflected on the screen. In automatic mode, the screen will be updated when there are changes.

**Keyboard Command Sequence:**

`draw`
**AUTOMATIC**

Command: `draw_automatic
draw_manual`

Description: These commands allow you to select the way in which changes are reflected in your model. If you select automatic, any changes to your model are shown in the graphics area immediately. If you select manual, the draw command must be used to show changes.

Keyboard Command Sequence:

- `draw_automatic
draw_manual`

Another button with the same description:

- MANUAL

**DRAW X-Y AXES**

Command: `draw_beam_axes`

Description: When this option is activated, elements which are associated with a 3-D general beam or elastic beam geometry, can have X-Y axes drawn at their centers to show the orientation of the beam cross section in the global coordinate system. The axes will not be visible if elements are being drawn in solid mode and `draw_beams_3d` is turned on.

This option is also available during postprocessing if the beam orientation vectors are available in the post file via post codes 261-263 (beam_orient in the JOB RESULTS menu).

Keyboard Command Sequence:

- `draw_beam_axes <on or off>
beam_axis_factor
draw_beams_3d
gameometry_value`
DRAW BEAM EDGES

Command:  draw_beam_edges

Description:  When this option is activated and 3-D general beams are being drawn with their cross section shown in three dimensions, the edges defined by the cross section will be drawn using the element edge color.

Keyboard Command Sequence:

    draw_beam_edges <on or off>
    draw_beams_3d

PLOT BEAMS IN 3D

Command:  draw_beams_3d

Description:  When this option is set, elements which have been associated with a 3-D general beam geometry will be drawn with their cross sections extended along the length of the beam in three dimensions.  This is intended to aid in the visualization of beams and their orientation within the model.

Beams may be drawn in solid or wireframe mode depending on the current element drawing mode. The scale factor used in drawing the beam cross sections, whether the cross section edges are drawn, and the number of divisions to use for drawing arc cross sections may be modified as well.

Keyboard Command Sequence:

    draw_beams_3d <on or off>
    beam_3d_scale
    draw_beam_edges
    beam_arc_divisions
    elements_wireframe
    elements_solid
Command: **draw_histplot_legend**

Description: This command toggles the drawing of the histplot legend.

Keyboard Command Sequence:

```
draw_histplot_legend <on or off>
```

---

Keyboard Command Sequence:

```
draw_interrupt_time <time, in seconds>
```

---

Command: **draw_legend**

Description: This command toggles the display of the legend (thermometer) when a post plotting option is turned on. It also toggles the display of the legend for path plots and history plots.

Keyboard Command Sequence:

```
draw_legend <yes or no>
```
**DRAW LIGHTS**

**Command:** `draw_local_lights`  
**Description:** This command toggles the display of symbols denoting local lights. In addition to this command, lighting must be on in a view for local lights to be drawn in that view.  
Also see: `set_lighting`.

**Keyboard Command Sequence:**
```
draw_local_lights <yes or no>
```

**LEGEND**

**Command:** `draw_pathplot_legend`  
**Description:** This command toggles the drawing of the pathplot legend.

**Keyboard Command Sequence:**
```
draw_pathplot_legend <on or off>
```

**LEGEND**

**Command:** `draw_xcurve_legend`  
**Description:** This command shows the correspondence between curve colors and response curves as provided by the experimental data fitting capability.

**TRI MESH!**

**Command:** `dt_planar_trimesh`  
**Description:** This command automatically meshes the given list of curves with triangular elements using the Delaunay algorithm. The curves must form at least one closed loop and lie in a unique plane. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.
Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have by default, one division or three divisions for closed curves.

Also see: apply_curve_divisions.

**Keyboard Command Sequence:**

```
dt_planar_trimesh <curve list>
```

Command:  **dt_surface_trimesh**  
**Description:** This command automatically meshes the given list of surfaces with triangular elements using the Delaunay algorithm for surfaces.

Each surface must contain closed trimming curves which bound the area to be meshed. Open loops are permitted, but any open loops not contained within a closed loop will be ignored. No loops may intersect.

Curve divisions should be applied to all curves before attempting to mesh. These divisions are used as element edges along the boundaries, and also determine the element size throughout the mesh. Curves without any divisions defined will have by default, one division or three divisions for closed curves.

Also see: apply_curve_divisions.

**Keyboard Command Sequence:**

```
dt_surface_trimesh <surface list>
```

Command:  **dt_tet_tri_mesh**  
**Description:** This command automatically meshes the given list of triangular elements with tetrahedral elements. The given triangles must completely enclose the volume to be meshed.

**Keyboard Command Sequence:**

```
dt_tet_tri_mesh <triangular element list>
```
Menu: DUPLICATE

Description: This menu contains commands used to copy elements and geometrical entities.

Command: duplicate_combined
set_duplicate_combined

Description: The duplicate_combined command duplicates a mixed list of items simultaneously. The list may contain items of the following types:

- cavities
- curves
- elements
- nodes
- points
- rbe2s
- rbe3s
- servos
- solids
- springs
- surfaces
- ties

Any attach relations that exist between the original mesh and geometry are duplicated for the copies of the mesh and the geometry.

The kind of items that are accepted by the command are controlled by the set_duplicate_combined command. This command activates or deactivates item types for a subsequent combined duplicate operation. Only active types are accepted by the duplicate_combined command and only items of these types are graphically pickable using the usual single pick, box pick, and polygon pick methods if the duplicate_combined command is executed. This allows, for example, to simultaneously duplicate elements and surfaces, but not curves. Wildcards like all_existing and all_selected can also be used with this command to indicate all existing or all selected items of the active types.

This button is located in the MESH GENERATION->DUPLICATE menu.

Also see:
- duplicate_nodes
- duplicate_elements
- duplicate_points
- duplicate_curves
- duplicate_surfaces
- duplicate_solids
- duplicate_ties
duplicate_servos
duplicate Springs
duplicate_rbe2s
duplicate_rbe3s
set_duplicate_point
set_duplicate_scale_factors
set_duplicate_rotations
set_duplicate_translations
set_duplicate_repetitions

Keyboard Command Sequence:

duplicate_combined <item list> #
set_duplicate_combined <item type> <on|off>

**Command:** duplicate_curves

**Description:** This command creates copies of curves. You must specify a list of curves to be duplicated. The control points associated with the curve are duplicated along the curve.

Also see: set_duplicate_point,
set_duplicate_scale_factors,
set_duplicate_rotations,
set_duplicate_translations, and
set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_curves <curve list> #

**Command:** duplicate_elements

**Description:** This command creates copies of elements. You must specify a list of elements to be duplicated.

Also see: set_duplicate_point,
set_duplicate_scale_factors,
set_duplicate_rotations,
set_duplicate_translations, and
set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_elements <element list> #

---

### DUPPLICATE LAYER

**Command:** `duplicate_layered_material`

**Description:** This command duplicates a material layer in a layered material. The specified layer can be duplicated into multiple layers. You must specify the layer to be duplicated and the layers to receive the duplicated layer. All layers between and including the specified layers will receive the duplicated layer.

**Keyboard Command Sequence:**

duplicate_layered_material
   <duplicate layer number>
   <from layer number> <to layer number>

---

### NODES

**Command:** `duplicate_nodes`

**Description:** This command creates copies of nodes. You must specify a list of nodes to be duplicated.

Also see: `set_duplicate_point`,
`set_duplicate_scale_factors`,
`set_duplicate_rotations`,
`set_duplicate_translations`, and
`set_duplicate_repetitions`.

**Keyboard Command Sequence:**

duplicate_nodes <nodes list> #
**POINTS**

**Command:**  duplicate_points

**Description:** This command creates copies of points. You must specify a list of points to be duplicated.

- Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_points <point list> #

---

**RBE2S**

**Command:**  duplicate_rbe2s

**Description:** This command creates copies of RBE2’s. Note that the nodes of the RBE2’s are duplicated as well. You must specify a list of RBE2’s to be duplicated.

- This button is located in the MESH GENERATION->DUPLICATE menu.

- Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_rbe2s <rbe2 list> #
**RBE3S**

**Command:**  **duplicate_rbe3s**

**Description:** This command creates copies of RBE3’s. Note that the nodes of the RBE3’s are duplicated as well. You must specify a list of RBE3’s to be duplicated.

This button is located in the MESH GENERATION->DUPLICATE menu.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_rbe3s <rbe3 list> #

---

**RESET**

**Command:**  **duplicate_reset**

**Description:** This command resets the values of the scale factors, rotations, translations, and repetitions used by the duplicate commands to their default value.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_reset
SERVOS

Command:  duplicate_servos
Description:  This command creates copies of servo links. Note that the nodes of the servo links are duplicated as well. You must specify a list of servo links to be duplicated.

This button is located in the MESH GENERATION->DUPLICATE menu.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

Keyboard Command Sequence:

duplicate_servos <servo list> #

SOLIDS

Command:  duplicate_solids
Description:  This command creates copies of solids. You must specify a list of solids to be duplicated.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

Keyboard Command Sequence:

duplicate_solids <solid list> #
**SPRINGS**

Command: **duplicate_springs**

Description: This command creates copies of springs. Note that the nodes of the springs are duplicated as well. You must specify a list of springs to be duplicated.

This button is located in the MESH GENERATION->DUPLICATE menu.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_springs <spring list> #

---

**SURFACES**

Command: **duplicate_surfaces**

Description: This command creates copies of surfaces. You must specify a list of surfaces to be duplicated.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_surfaces <surface list> #
**TIES**

**Command:** duplicate_ties  
**Description:** This command creates copies of nodal ties. Note that the nodes of the nodal ties are duplicated as well. You must specify a list of ties to be duplicated.

This button is located in the MESH GENERATION->DUPLICATE menu.

Also see: set_duplicate_point, set_duplicate_scale_factors, set_duplicate_rotations, set_duplicate_translations, and set_duplicate_repetitions.

**Keyboard Command Sequence:**

duplicate_ties <tie list> #

---

**DYN LIGHTING**

**Command:** dynamic_lighting_on  
**Command:** dynamic_lighting_off  
**Description:** These commands turn dynamic lighting on and off. When dynamic lighting is enabled and lighting is on, you can move the lights in a view directly using the mouse, rather than by repeatedly using the light_location command. Only the selected lights move.

With dynamic lighting on, holding the left mouse button <ML> down and moving the cursor in the graphics area causes translation of selected local lights. Holding the middle mouse button <MM> down and moving the cursor causes rotation of all selected lights. Holding the right mouse button <MR> down and moving the cursor causes scaling of selected local lights.

Also see: set_lighting, light_location, light_select, light_local, dynamic_spotlight_aiming_on, and set_draw_interrupt_on.

**Keyboard Command Sequence:**

dynamic_lighting_on  
dynamic_lighting_off
DYN. MODEL

Command:  
\begin{verbatim}
dynamic_model_on
dynamic_model_off
\end{verbatim}

Description:  These commands turn dynamic model on and off. When dynamic model is enabled you can move your model on the screen directly using the mouse, rather than by repeatedly using the translate \((\text{trans}_+)\) and rotate \((\text{rot}_+)\) commands.

With dynamic model on, holding the left mouse button \(<\text{ML}>\) down and moving the cursor in the graphics area translates the model, holding the middle mouse \(<\text{MM}>\) button down and moving the cursor rotates the model, and holding the right mouse button \(<\text{MR}>\) down and moving the cursor scales the model.

Keyboard Command Sequence:
\begin{verbatim}
dynamic_model_on
dynamic_model_off
\end{verbatim}

DYN SPOT AIM

Command:  
\begin{verbatim}
dynamic_spotlight_aiming_on
dynamic_spotlight_aiming_off
\end{verbatim}

Description:  These commands turn dynamic spotlight aiming on and off. When dynamic spotlight aiming is enabled and lighting is on, you can aim the spotlights in a view directly using the mouse, rather than by repeatedly using the \text{light_spot_direction} command. Only the selected spotlights change.

With dynamic spotlight aiming on, holding the middle mouse button \(<\text{MM}>\) down and moving the cursor causes rotation of the aim of all selected spotlights.

Also see: \text{set_lighting}, \text{light_spot_direction}, \text{light_select}, \text{set_light_spotlight}, and \text{dynamic_lighting_on}.

Keyboard Command Sequence:
\begin{verbatim}
dynamic_spotlight_aiming_on
dynamic_spotlight_aiming_off
\end{verbatim}
Command: **edge_area**

**Description:** This command determines the area enclosed by the specified edges. Edges are specified with an element and edge number, separated by a colon.

**Keyboard Command Sequence:**

```
edge_area <edge list> #
```
**EDGES TO CURVES**

**Command:** `edge_curves`

**Description:** This command converts element edges into curves (type line). You must specify a list of edges to convert. Edges are specified with an element and edge number, separated by a colon.

**Keyboard Command Sequence:**

```
edge_curves <edge list> #
```

**EDGE LENGTH**

**Command:** `edge_length`

**Description:** This command calculates the total length of a set of edges. The specified set of edges do not have to be connected; the length is calculated as the sum of the lengths of the individual edges.

**Keyboard Command Sequence:**

```
edge_length <edge list> #
```

Another Button with the same description:

```
OUTLINE EDGE LENGTH
```

**EDGES TO ELEMENTS**

**Command:** `edge_lines`

**Description:** This command converts element edges into line elements. You must specify a list of edges to convert. Edges are specified with an element and edge number, separated by a colon.

**Keyboard Command Sequence:**

```
edge_lines <edge list> #
```
Command: **edges_full**

Description: This command specifies that all edges in the visible portion of the model be displayed.

Keyboard Command Sequence:

```
edges_full
```

---

Command: **edges_outline**

Description: This command specifies that only edges on the outline of the visible portion of the model be displayed. Whenever the outline of the model has changed significantly, the related `find_solid_outline` command must also be issued to determine which edges are on the outline.

Also see: `find_solid_outline`.

Keyboard Command Sequence:

```
edges_outline
```

---

Command: **edges_surface**

Description: This command specifies that only edges on the surface of the visible portion of the model be displayed.

Keyboard Command Sequence:

```
edges_surface
```
EDIT

Command:  edit_adapg
Description:  This command selects an entry in the list of remeshing criteria to be edited and makes it the current one. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

Keyboard Command Sequence:

edit_adapg <crack name>

EDIT

Command:  edit_adapt
Description:  This command selects the adapt to be edited and makes it the current adapt. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

Keyboard Command Sequence:

edit_adapt <adapt name>

EDIT

Command:  edit_annotation
Description:  This command is used to change the text of an annotation. Annotations are strings of text used to describe what is being displayed in a view. You must specify the annotation to be edited and the new text for the annotation. Annotations are specified by number. This button is located in the UTILITIES->MORE menu.

Keyboard Command Sequence:

edit_annotation <annotation number> <text>
Command:  **edit_apply**  
**Description:** This command selects the application to be edited and makes it the current application. This button is located in the **BOUNDARY CONDITION** menu.

**Keyboard Command Sequence:**
```
edit_apply <application name>
```

Command:  **edit_cavity**  
**Description:** This command selects the cavity to be edited and makes it the current cavity. It is located under the **MESH GENERATION->CAVITIES** menu.

**Keyboard Command Sequence:**
```
edit_cavity <cavity name>
```

Command:  **edit_composite**  
**Description:** This command creates a new composite material if the current material is not of composite type, and makes it the current material.

**Keyboard Command Sequence:**
```
edit_composite
```

Command:  **edit_contact_area**  
**Description:** This command selects the contact area to be edited and makes it the current contact area. This button is located in the **CONTACT->CONTACT AREAS** menu.

**Keyboard Command Sequence:**
```
edit_contact_area <contact area name>
```
Command: **edit_contact_body**

Description: This command selects the contact body to be edited and makes it the current contact body. This button is located in the CONTACT->CONTACT BODIES menu.

**Keyboard Command Sequence:**

```
edit_contact_body <contact body name>
```

---

Command: **edit_contact_table**

Description: This command selects the contact table to be edited and makes it the current contact table. This button is located in the CONTACT->CONTACT TABLES menu.

**Keyboard Command Sequence:**

```
edit_contact_table <contact table name>
```

---

Command: **edit_crack**

Description: This command selects the crack to be edited and makes it the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS and 3-D CRACKS menus.

**Keyboard Command Sequence:**

```
edit_crack <crack name>
```
Command: edit_csect

Description: This command selects the cross-section to be edited and makes it the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:

edit_csect <cross-section name>

Command: edit_curves

Description: This command allows you to redefine a point used by a curve. The old points are detached from the curve but remain in the model. The sweep_points command is used to remove these points. Any surfaces or elements attached to the curve will be appropriately modified. You must specify the curve, a point on the curve, and a new point to replace the first point. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:

edit_curves <curve id> <existing curve point> <new curve point>

Command: edit_descon

Description: This command selects the design constraint to be edited and makes it the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

Keyboard Command Sequence:

edit_descon <design constraint name>
Command: **edit_desvar**

**Description:** This command selects the design variable to be edited and makes it the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

**Keyboard Command Sequence:**

```
edit_desvar <design variable name>
```

---

Command: **edit_elements**

**Description:** This command allows you to redefine the nodes used by an element. The old nodes are detached from the element but remain in the model. The `sweep_nodes` command is used to remove them from the model. You must specify the element, a node on the element, and a new node to replace the first node. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
edit_elements <element id> <existing element node> <new element node>
```

---

Command: **edit_exseg**

**Description:** This command selects the exclude segment to be edited and makes it the current one. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

```
edit_exseg <exclude segment name>
```
EDIT FILE

Command:  **edit_file**
Description:  This command opens an edit window for the specified file. The type of window (default: xterm) and the editor (default: vi) can be customized in the edit_window file in the Mentat tools directory. This button is located in the UTILITIES menu.

Keyboard Command Sequence:

```
edit_file <file name>
```

Another Button with the same description:

ANY FILE

EDIT

Command:  **edit_geometry**
Description:  This command selects the geometry property to be edited and makes it the current geometry. This button is located in the GEOMETRIC PROPERTIES menu.

Keyboard Command Sequence:

```
edit_geometry <geometry name>
```

EDIT

Command:  **edit_icond**
Description:  This command selects the initial condition to be edited and makes it the current initial condition. This button is located in the INITIAL CONDITIONS menu.

Keyboard Command Sequence:

```
edit_icond <application name>
```
Command:  edit_insert
Description:  This command selects the inserts to be edited and makes it the current insert.

Keyboard Command Sequence:
edit_insert <insert name>

Command:  edit_job
Description:  This command selects the job to be edited and makes it the current job.
This button is located in the JOBS menu.

Keyboard Command Sequence:
edit_job <job name>

Command:  edit_link
Description:  This command selects the link to be edited and makes it the current link.

Keyboard Command Sequence:
edit_link <link name>

Another Button with the same description:
OK
**Command:** edit_loadcase  
**Description:** This command selects the loadcase to be edited and makes it the current loadcase. This button is located in the LOADCASES menu.

**Keyboard Command Sequence:**

```
edit_loadcase <loadcase name>
```

**Command:** edit_marc_history_text  
**Description:** This command allows you to redefine the text line that will be placed in the Marc data file at the end of the history definition section.  
The command input consists of the line number that you want to replace and the text that will replace the existing text string.

**Note:** If you want the text to have preceding spaces you will have to enclose the text string in quotation marks, such as "node,1,51,2,0.10035e-05,0.10000e-01".

**Keyboard Command Sequence:**

```
edit_marc_history_text <line number> <text>
```

**Command:** edit_marc_model_text  
**Description:** This command allows you to redefine the text line that will be placed in the Marc data file at the end of the model definition section.  
The command input consists of the line number that you want to replace and the text that will replace the existing text string.

**Note:** If you want the text to have preceding spaces you will have to enclose the text string in quotation marks, such as "node,1,51,2,0.10035e-05,0.10000e-01".

**Keyboard Command Sequence:**

```
edit_marc_model_text <line number> <text>
```
**Command:** edit_marc_param_text  
**Description:** This command allows you to redefine the text line that will be placed in the Marc data file at the end of the parameter definition section. The command input consists of the line number that you want to replace and the text that will replace the existing text string. 

*Note:* If you want the text to have preceding spaces you will have to enclose the text string in quotation marks, such as "node,1,51,2,0.10035e-05,0.10000e-01".

**Keyboard Command Sequence:**

```
eid_t_marc_param_text <line number> <text>
```

---

**Command:** edit_material  
**Description:** This command selects the material to be edited and makes it the current material. This button is located in the MATERIAL PROPERTIES menu.

**Keyboard Command Sequence:**

```
eidt_material <material name>
```

---

**Command:** edit_nodes  
**Description:** This command allows you to relocate an existing node. You must specify the node and its new coordinates. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
eidt_nodes <node id> <new coordinates>
```
Command: **edit_orient**

**Description:** This command selects the orientation to be edited and makes it the current orientation. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

**Keyboard Command Sequence:**

```
edit_orient <orient name>
```

---

Command: **edit_points**

**Description:** This command allows you to relocate an existing point. You must specify the point and its new coordinates. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
edit_points <point id> <new coordinates>
```

---

Command: **edit_post_nodal_quantity**

**Description:** This command selects user defined results variables to be written to the results file during a Marc run. You must specify the variables by their label. The button to the left is the toggle button and is lit up after specifying the label, which is equivalent to the add_post_var command. To deactivate the user defined variable, press the button will to turn it off, corresponding to remove_post_var, and the variable is taken off the list. Note that you will only be allowed to change the user defined variables. These are named user<n> for ELEMENT SCALARS and User_Nodal_<n> for NODAL QUANTITIES, where <n> is a number from 1 to 30. If any other results variable is selected it will be ignored. This button is located in the JOBS->MECHANICAL->JOB RESULTS and other analysis class menus.

**Keyboard Command Sequence:**

```
edit_post_nodal_quantity <variable label>
```
Command:  **edit_post_var**  
Description:  This command selects user defined results variables to be written to the results file during a Marc run. You must specify the variables by their label. The button to the left is the toggle button and is lit up after specifying the label, which is equivalent to the **add_post_var** command. To deactivate the user defined variable, press the button will to turn it off, corresponding to **remove_post_var**, and the variable will be taken off the list. Note that you will only be allowed to change the user defined variables. These are named **user<n>** for **ELEMENT SCALARS** and **User_Nodal_<n>** for **NODAL QUANTITIES**, where **<n>** is a number from 1 to 30. If any other results variable is selected it will be ignored. This button is located in the **JOBS->MECHANICAL->JOB RESULTS** and other analysis class menus.  

Keyboard Command Sequence:

\[
\text{edit_post_var} \ <\text{variable label}> 
\]

Command:  **edit_rbe2**  
Description:  This command selects the rbe2 to be edited and makes it the current rbe2. This button is located in the **LINKS->RBE2'S** menu.  

Keyboard Command Sequence:

\[
\text{edit_rbe2} \ <\text{name}> 
\]

Command:  **edit_rbe3**  
Description:  This command selects the rbe3 to be edited and makes it the current rbe3. This button is located in the **LINKS->RBE3'S** menu.  

Keyboard Command Sequence:

\[
\text{edit_rbe3} \ <\text{name}> 
\]
Command: edit_section
Description: This command selects the beam section to be edited and makes it the current beam section. This button is located in the GEOMETRIC PROPERTIES->3D->BEAM SECTIONS menu.

Keyboard Command Sequence:
edit_section <beam section name>

Command: edit_surfaces
Description: This command allows you to redefine a point used by a surface. The old points are detached from the surface but remain in the model. The sweep_points command is used to remove these points. Any elements attached to the surface are appropriately modified. You must specify the surface, a point on the surface, and a new point to replace the first point. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:
edit_surfaces <surface id> <existing surface point> <new surface point>

Command: edit_table
Description: This command selects the table to be edited and makes it the current table. This button is located in the MATERIAL PROPERTIES->TABLES menu.

Keyboard Command Sequence:
edit_table <table name>
Command: **edit_transform**

**Description:** This command selects the transform to be edited and makes it the current transform. This button is located in the BOUNDARY CONDITIONS->MECHANICAL menu.

**Keyboard Command Sequence:**

```
edit_transform <transform name>
```
**ELEMENT MASS**

**Command:**  
`element_mass`

**Description:**  
This command calculates the total mass of a set of elements. The specified set of elements do not have to be connected; the mass is calculated as the sum of the masses of the individual elements. The elements must have material properties assigned to them so that the appropriate mass densities can be used. Only volume elements are currently supported.

**Keyboard Command Sequence:**  
```
   element_mass <element list> #
```

---

**Command:**  
`element_type`

**Description:**  
This command permits the user to set the Marc element type for the specified elements. At creation time, an element's type is initialized to zero. The command modifies only those elements whose geometry matches that of the specified type. You must specify the element type and a list of elements. Note that to use element types 15 and 16, you can use the user subroutine `ufxord`. Mentat will not calculate the derivatives required for these element types.

**Keyboard Command Sequence:**  
```
   element_type <type> <element list> #
```

**Other Buttons with the same description:**
- 10
- 101 through 109
- 11
- 110 through 119
- 12
- 120 through 139
- 14
- 140 through 149
- 15
- 150 through 157
- 16
- 18
- 19
- 2
- 20 through 23
**ELEMENT TYPES**

**Menu:** ELEMENT TYPES

**Description:** This menu contains menus that contain commands for setting the Marc element type for specified elements.

**ELEMENT VOLUME**

**Command:** `element_volume`

**Description:** This command calculates the total volume of a set of elements. The specified set of elements do not have to be connected; the volume is calculated as the sum of the volumes of the individual elements.

**Keyboard Command Sequence:**

```
  element_volume <element list> #
```
Button: **SOLID** *(elements_solid, elements_wireframe)*

**Description:** This button allows toggling between solid and wireframe display of elements.

If the toggle is ON, element faces are displayed with solid color.

If the toggle is OFF, elements are displayed in wireframe mode. In this mode, element faces are represented by crossed diagonal lines. In addition, an arrow is located on the first edge pointing from the first node to the second. This is used to show the orientation of the element.

**Keyboard Command Sequence:**

```
elements_solid
elements_wireframe
```

Another Button with the same description:

**SOLID**

---

**Subroutine: ELEVAR**

**Description:** The user subroutine **ELEVAR** makes element (integration point) quantities available at each increment so that you save them in any form convenient for your postprocessing. **ELEVAR** is used in conjunction with the UDUMP option.

---

**Subroutine: ELEVEC**

**Description:** The user subroutine **ELEVEC** makes element (integration point) quantities available at each harmonic subincrement so that you save them in any form convenient for your postprocessing. **ELEVEC** is used in conjunction with the UDUMP subroutine.
**EXECUTE**

Command: **exec_procedure**

Description: This command executes a previously created procedure file. The procedure file may contain commands to execute other procedures.

**Keyboard Command Sequence:**

```
exec_procedure <file name>
```

Another Button with the same description:

MESH

---

**EXECUTE 1**

Command: **execute_job**

Description: This command submits a job to the Marc analysis program. It first checks for the existence of an input deck with a name of the form `model_job.dat` and then starts the job. The job submission files `submit1`, `submit2`, and `submit3`, located in the tools directory, are tailored by the user. You must specify the job submission number (1, 2, or 3) which specifies which job submission file to use.

This command is used instead of SUBMIT when the input deck has been modified.

**Keyboard Command Sequence:**

```
execute_job <1, 2, or 3>
```

Other Buttons with the same description:

EXECUTE 2
EXECUTE 3
EXIT MESSAGE

Description: This popup menu displays the exit message of the current Marc job.

PLOT OPTIONS

Description: This menu allows the user to adjust the parameters to the curves supplied by the experimental data fitting capability.

You can e.g. set (manually or automatically) the bounds of the x- and y-axis, switch on or off plotting of predicted modes, and modify the colors of the curves.

TABLES

Description: Load and process experimental data tables.

Upon having read the tables, the user associates the table with an experiment.

Input files are expected to be read as RAW

Input files are either x-y, or x-y-z column tabular.

- Field separators are blank spaces.
- The z-field is read but cannot be displayed.

Other notes:

a. Do not imply high levels of accuracy. i.e.

0.009000000000012  1034.454
b. Do not load tables with more than 200 entries. Typically 50 points per experiment is sufficient. The total number of points which will be treated by the fitting capability is 200. i.e. Should the input tables for the experiments contain the following number of entries:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIAXIAL</td>
<td>89</td>
</tr>
<tr>
<td>BIAXIAL</td>
<td>87</td>
</tr>
<tr>
<td>PLANAR SHEAR</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>288</td>
</tr>
</tbody>
</table>

then only the first (112-88) 24 points of the PLANAR SHEAR test will be used when fitting together with the USE ALL DATA button.

c. Good practice includes incorporating many data points near the undeformed state and near regions of high curvature.

---

Menu: **EXPAND**

**Description:** This menu contains commands that expand the dimensionality of elements and geometric entities.

---

**Command:** `expand_apply_transform`

**Description:** This command turns on/off flag for generation of apply transformations during the `expand_axito3d` command.

If there are any Point Loads or Prescribed Nodal Displacements in y- or z-direction, this button should be activated.

**Keyboard Command Sequence:**

```
set_expand_apply_transform <on/off>
```
EXPAND MODEL

Command:  expand_axito3d
Description: This menu contains the commands for expanding axisymmetric mesh to 3-D for AXITO3D analysis. It works for both equal and nonequal spaced expansion. This process also converts the boundary conditions on 2-D elements to the new ones on the new 3-D elements.

It also allow users to specify if the load table curves are shifted or not during the expanding process. The load table curves are shifted based on the user input time value or increment number.

EXPAND

Command:  expand_combined
set_expand_combined

Description: The expand_combined command expands a mixed list of items simultaneously. The list may contain items of the following types:

nodes  elements  points  curves

Any attach relations that exist between the original mesh and geometry are expanded as well. For example, if an edge is attached to a curve then the faces that result from expansion of the edge will be attached to the surfaces that result from expansion of the curve.

The kind of items that are accepted by the command are controlled by the set_expand_combined command. This command activates or deactivates item types for a subsequent combined expand operation. Only active types are accepted by the expand_combined command and only items of these types are graphically pickable using the usual single pick, box pick, and polygon pick methods if the expand_combined command is executed. This allows, for example, to simultaneously expand nodes and points, but not elements. Wildcards like all_existing and all_selected can also be used with this command to indicate all existing or all selected items of the active types.

This button is located in the MESH GENERATION->EXPAND menu.

Also see: expand_nodes, expand_elements, expand_points, and expand_curves.
**Keyboard Command Sequence:**

```
expand_combined <item list> #
set_expand_combined <item type> <on|off>
```

---

**CURVES**

**Command:** `expand_curves`

**Description:** This command expands curves into surfaces about a given point, with the specified scale factors, rotations, and translations. For each repetition of the expansion, a ruled surface is created between consecutive curve positions. You must specify a list of curves.

**Keyboard Command Sequence:**

```
expand_curves <curve list> #
```

---

**ELEMENTS**

**Command:** `expand_elements`

**Description:** This command expands elements about a given point with the specified scale factors, rotations, and translations. For each repetition of the expansion, a new element is created for each of the specified elements. The dimension of the new elements is one greater than the elements from which they were expanded. In other words, line elements are expanded into quadrilateral elements, and quadrilateral elements are expanded into brick elements. The following is a list of the expansion classes:

<table>
<thead>
<tr>
<th>Original</th>
<th>Expanded</th>
</tr>
</thead>
<tbody>
<tr>
<td>line2</td>
<td>quad4</td>
</tr>
<tr>
<td>line3</td>
<td>quad8</td>
</tr>
<tr>
<td>tria3</td>
<td>penta6</td>
</tr>
<tr>
<td>tria6</td>
<td>penta15</td>
</tr>
<tr>
<td>quad4</td>
<td>hex8</td>
</tr>
<tr>
<td>quad6</td>
<td>hex12</td>
</tr>
<tr>
<td>quad8</td>
<td>hex20</td>
</tr>
<tr>
<td>quad9</td>
<td>hex27</td>
</tr>
</tbody>
</table>

You must specify a list of elements. It is recommended that you do a `sweep_all` before running this command.

**Keyboard Command Sequence:**

```
expand_elements <element list> #
```
ELEMENTS ALONG CURVE

Command:  `expand_elements_curve`
Description:  This command expands elements along a given curve. Translations and rotations are dictated by the shape of the given curve.

Keyboard Command Sequence:
```
expand_element_curve <curve> <element list> #
```

LINE ELEMENTS

Command:  `expand_lines`
Description:  This command expands one-dimensional elements into two-dimensional elements in a direction normal to their edges. The resulting two-dimensional elements will have the specified thickness and will be offset by the specified offset in the direction normal to the edges of the original elements. The element expansion classes are:

<table>
<thead>
<tr>
<th>Original</th>
<th>Expanded</th>
</tr>
</thead>
<tbody>
<tr>
<td>line2</td>
<td>quad4</td>
</tr>
<tr>
<td>line3</td>
<td>quad8</td>
</tr>
</tbody>
</table>

You must specify a list of elements.

Keyboard Command Sequence:
```
expand_lines <element list> #
```

NODES

Command:  `expand_nodes`
Description:  This command expands nodes into two-noded line elements. The expansion takes place about a point with a given scale factor, rotation, and translation. For each repetition of the expansion, a two-noded line element is created for each of the specified nodes. You must specify a list of nodes.

Keyboard Command Sequence:
```
expand_nodes <node list> #
```
**NON-EQUISPACED EXPAND**

Command: **expand_non_equispace**

**Description:** This menu contains the commands for expanding elements from 2-D to 3-D. It works for both equal and non-equal spaced expansion. This process expands mesh elements only.

**Keyboard Command Sequence:**

```
expand_non_equispace
```

**POINTS**

Command: **expand_points**

**Description:** This command expands points into lines. The expansion takes place about a point with a given scale factor, rotation, and translation. For each repetition of the expansion, a line is created for each of the specified points. You must specify a list of points.

**Keyboard Command Sequence:**

```
expand_points <point list> #
```

**REMOVE**

Command: **expand_remove**

**Description:** This command specifies that the elements specified in an expand_elements command will be removed from the model upon completion of the command.

**Keyboard Command Sequence:**

```
expand_remove
```
**RESET**

Command:  **expand_reset**  
Description:  This command resets the expansion parameters to their original values.

Keyboard Command Sequence:  
```
expand_reset
```  

**SAVE**

Command:  **expand_save**  
Description:  This command specifies the elements in an `expand_elements` command to be saved upon completion of the command.

Keyboard Command Sequence:  
```
expand_save
```  

**SHELL ELEMENTS**

Command:  **expand_shells**  
Description:  This command expands two-dimensional elements into three-dimensional elements in a direction normal to their faces. The resulting solid elements have the specified thickness and are offset by the specified offset in the direction normal to the faces of the original elements. The element expansion classes are:

<table>
<thead>
<tr>
<th>Original</th>
<th>Expanded</th>
</tr>
</thead>
<tbody>
<tr>
<td>tria3</td>
<td>penta6</td>
</tr>
<tr>
<td>tria6</td>
<td>penta15</td>
</tr>
<tr>
<td>quad4</td>
<td>hex8</td>
</tr>
<tr>
<td>quad6</td>
<td>hex12</td>
</tr>
<tr>
<td>quad8</td>
<td>hex20</td>
</tr>
<tr>
<td>quad9</td>
<td>hex27</td>
</tr>
</tbody>
</table>

You must specify a list of elements.

Keyboard Command Sequence:  
```
expand_shells <element list> #
```
**SHIFT**

**Command:**   expand_shift

**Description:**  This command specifies the elements in an expand_elements command to be shifted to the final position upon completion of the command.

**Keyboard Command Sequence:**

    expand_shift

---

**AXISYMMETRIC MODEL TO 3D**

**Menu:**   AXISYMMETRIC MODEL TO 3D

**Description:**  This menu contains the commands for expanding axisymmetric mesh to 3-D for AXITO3D analysis. It works for both equal and nonequal spaced expansion. This process also converts the boundary conditions on 2-D elements to the new ones on the new 3-D elements.

   It also allow users to specify if the load table curves are shifted or not during the expanding process. The load table curves are shifted based on the user input time value or increment number.

---

**EXPERIMENTAL DATA FIT**

**Menu:**   EXPERIMENTAL DATA FITTING

**Description:**  Obtain the material properties from user supplied experimental data.

   On the next menu, the user reads in the x, y(z) data as raw data, to a table. The user then associates with each designation of an experiment, the corresponding table.
**EXPORT**

**Command:** `export`

**Description:** Here are the different data types that can be produced by Mentat.

- ACIS
- FIDAP
- IGES
- NASTRAN
- STL
- VRML

**Keyboard Command Sequence:** (for everything except VRML)

```bash
export <Data type> <File name>
```

**Keyboard Command Sequence:** (for VRML)

```bash
export_vrml <view number> <File name>
```

**ACIS**

**Command:** `export acis`

**Description:** This command exports an ACIS geometry file. You must specify the name of the file. There are two options:

- TEXT/BINARY
- ACIS VERSION
  
  - 1.6 : Mentat 2.3.1 used this version
  - 2.1 : Mentat 3.1 used this version
  - 3.0 : Mentat 3.2 used this version

**Keyboard Command Sequence:**

```bash
export acis <ACIS file name>
```

Other Buttons with the same description:

- VERSION 1.6
- VERSION 2.1
- VERSION 3.0
- VERSION 4.0
- VERSION 5.0
- acis_screen2
**DXF**

**Command:** `export dxfout`  
**Description:** This command exports a DXF geometry file. You must specify the name of the file.  
The current version is AutoCAD 2000.  
**Keyboard Command Sequence:**  
```
export dxfout <DXF file name>
```

**FIDAP**

**Command:** `export fidap`  
**Description:** This command exports a FIDAP geometry file. You must specify the name of the file.  
**Keyboard Command Sequence:**  
```
export fidap <FIDAP file name>
```

**IGES**

**Command:** `export iges`  
**Description:** This command exports an IGES geometry file. You must specify the name of the file.  
The current version is 5.2.  
**Keyboard Command Sequence:**  
```
export iges <IGES file name>
```
**NASTRAN**

Command: **export nastran**

Description: This command exports a NASTRAN bulk data file. You must specify the name of the file.

Keyboard Command Sequence:
```
export nastran <NASTRAN bulk data file name>
```

**STL**

Command: **export stl**

Description: This command exports an ASCII Stereolithography Interface Format (STL) file. You must specify the name of the file.

The current version is October 1989.

Keyboard Command Sequence:
```
export stl <STL ASCII file name>
```

**VDAFS**

Command: **export vdaout**

Description: This command exports an VDAFS geometry file. You must specify the name of the file.

The current version is VDA-FS Revision 2.0.

Keyboard Command Sequence:
```
export vdaout <VDAFS file name>
```
Command: **export_vrml**

Description: This command exports MSC.Marc Mentat’s graphics window to an ASCII VRML (the Virtual Reality Modeling Language) file. You must specify the view number and the name of the file.

This VRML writer uses VRML97 (a.ka. ISO VRML and VRML 2.0) format to author the 3-D virtual worlds. It supports all graphics entities (e.g. line, poly_fill, etc), lighting, symbols, text, materials, color, transformation, background info, etc.

**TEXT SIZE**
This variable specifies the height (in object space units) of the glyphs rendered and determines spacing of adjacent lines of text.

**SYMBOL SIZE**
This variable specifies various dimensions of the symbols, for example:

- NODE -> box (h=size*1.5, d=size*1.5, w=size1.5
  plus (aka. point) -> sphere (radius = size)
  etc.

VIEW 1
VIEW 2
VIEW 3
VIEW 4

Buttons for setting the output view number.

The current version is VRML97.

**Keyboard Command Sequence:**

```
export_vrml <view number> <VRML ASCII file name>
```

Other Buttons with the same description:

- VIEW 2
- VIEW 3
- VIEW 4
- VRML
Command: **exseg_add_edges**

**Description:** This command adds edges to the current exclude segment. The edges should belong to the exclude segment’s specified contact body and lie on that body’s outline or surface. This button is located in the CONTACT-> EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**
```
exseg_add_edges <edge list> #
```

Command: **exseg_add_faces**

**Description:** This command adds faces to the current exclude segment. The faces should belong to the exclude segment’s specified contact body and lie on that body’s outline or surface. This button is located in the CONTACT-> EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**
```
exseg_add_faces <face list> #
```

Command: **exseg_body**

**Description:** This command sets the contact body for the current exclude segment.

**Keyboard Command Sequence:**
```
exseg_body <contact body>
```
Command: **exseg_name**

**Description:** This command sets or changes the name of the current exclude segment.

**Keyboard Command Sequence:**

```
exseg_name <exclude segment name>
```

Command: **exseg_remove_edges**

**Description:** This command removes edges from the current exclude segment. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

```
exseg_remove_edges <edge list> #
```

Command: **exseg_remove_faces**

**Description:** This command removes faces from the current exclude segment. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

```
exseg_remove_faces <face list> #
```

Command: **exseg_reset**

**Description:** This command resets the current exclude segment. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

```
exseg_reset
```
Command: **extime**

**Description:** This command displays how much time (in seconds) has elapsed since the current session of MSC.Marc Mentat began running. This may be useful for timing an entire procedure within MSC.Marc Mentat.

Also see: extime_diff and extime_set.

**Keyboard Command Sequence:**

```
extime
```

---

Command: **extime_diff**

**Description:** This command displays how much time (in seconds) has elapsed since this command was last called, or since the most recent use of the extime_set command. After displaying the elapsed time, extime_diff then resets this timer to zero just as extime_set does. These commands may be useful for timing a particular operation or group of other commands within MSC.Marc Mentat. The following is an example of this from a MSC.Marc Mentat procedure file.

```
*message Regen, wireframe, surface edges and faces, nodes:
*extime_set
*regenerate
*extime_diff
```

Here, the message command displays what is about to be timed, then the call to extime_set resets the timer to zero, then a graphics regeneration is performed, and finally the extime_diff command displays how long the regeneration took.

Also see: extime, extime_set, and message.

**Keyboard Command Sequence:**

```
extime_diff
```
Command: **extime_set**

Description: This command resets an internal timer to zero, so that a subsequent call to the `extime_diff` command will report how much time (in seconds) has elapsed since the call to `extime_set`. The `extime_set` command does not display any information when called. These commands may be useful for timing a particular operation or group of commands within MSC.Marc Mentat. The following is an example of this from a MSC.Marc Mentat procedure file.

```
*message Regen, wireframe, surface edges and faces, nodes:
*extime_set
*regenerate
*extime_diff
```

Here, the `message` command displays what is about to be timed, then the call to `extime_set` resets the timer to zero, then a graphics regeneration is performed, and finally the `extime_diff` command displays how long the regeneration took.

Also see: `extime`, `extime_diff`, and `message`.

**Keyboard Command Sequence:**

```
extime_set
```
Mentat Help Commands in F

FACE AREA

Command: face_area

Description: This command calculates the sum of the areas of a set of faces. The specified set of faces do not have to be connected; the area is calculated as the sum of the areas of the individual faces. Faces are specified with an element number and a face number, separated by a colon.

Keyboard Command Sequence:

    face_area <face list> #
**FACES TO ELEMENTS**

**Command:** `face_quads`

**Description:** This command converts element faces into quad elements. You must specify a list of faces. Faces are specified with an element number and a face number, separated by a colon.

**Keyboard Command Sequence:**

```
face_quads <face list> #
```
Command:  faces_full
Description:  This command specifies that all faces in the visible portion of the model will be drawn.
Keyboard Command Sequence:  
faces_full

Command:  faces_surface
Description:  This command specifies that only the element faces on the surface of the visible portion of the model will be drawn.
Keyboard Command Sequence:  
faces_surface

Menu:  DIVISIONS
Description:  Curve and Surface Plotting Controls
Curves and surfaces are represented graphically by linear approximations, referred to as facets.
The breakup is performed by recursively subdividing the curve or surface until a specified deviation tolerance distance is satisfied. This tolerance may be specified in absolute or relative terms.
The breakup of curves and surface into facets is controlled by the following settings:
Relative/Absolute:
Specifies whether the tolerance is in terms of an absolute distance, or is in terms of the distance relative to the length of the curve or area of the surface.
**Tolerance:**
The maximum allowed deviation in absolute or relative distance of the facet to the curve or surface.

**Max Depth:**
The maximum depth of recursion allowed when subdividing the facets. Subdivision will continue until the tolerance is satisfied, or the maximum recursion depth is reached.

**Min Depth:**
This specifies the minimum depth of recursion, and when set to nonzero values forces the curve or surface to be represented by a minimum number of facets, even though the tolerance may have been satisfied.

It is recommended that relative tolerances be used for the general case. Absolute tolerance may be used to advantage to minimize the drawing of small details in a model whose dimension is known.

For convenience, three predefined settings: low, medium, and high are provided. The low setting is designed to minimize drawing time, while the high setting is designed to provide an extremely accurate representation of the geometry. The default setting is medium.

---

**Menu:** FILE

**Description:** This menu contains commands that control input and output from Mentat. You can open and save the model database, read and write Marc files, and write PostScript graphics files. From the FILE I/O menu, you can also import model information from other CAD systems.
Command:  fill_automatic
Description:  This command specifies that filling of areas to be done automatically. You might not wish to have the program fill areas automatically for better program performance. Filling can be made manual with the fill_manual command.

Keyboard Command Sequence:
fill_automatic

ManuaL

Command:  fill_manual
Description:  This command specifies that filling of areas to be done manually. You might wish to perform filling manually for better program performance. Filling will be made automatic with the fill_automatic command.

Keyboard Command Sequence:
fill_manual

FILL

Command:  fill_view
Description:  This command changes the zoom factor in all active views so that the entire model is visible. View rotations remain unchanged.

Keyboard Command Sequence:
fill_view
Subroutine: FILM
Description: The user subroutine FILM allows you to include nonuniform film coefficients and sink temperatures for the calculation of convection or radiation boundary conditions.

FILM is called at each time step for each integration point on each element surface given in the FILMS model definition set, and allows you to modify the film coefficient and sink temperature.

Command: find_solid_outline
Description: This command determines which edges of solid elements are to be considered outline edges. This is done by computing the dihedral angle between the neighboring faces of the edge. When this angle is beyond 60 degrees, the edge is considered to be on the outline. Plot settings must be on EDGES OUTLINE and the draw command must be issued to see the results of this computation.

Also see: edges_outline.

Keyboard Command Sequence:

find_solid_outline

Command: flip_curves
Description: This command reverses the orientation of curves. You must provide a list of curves.

Keyboard Command Sequence:

flip_curves <curve list> #
FLIP ELEMENTS

Command:  `flip_elements`
Description: This command reverses the orientation of the surface normals of elements. You must provide a list of elements.

Keyboard Command Sequence:
```plaintext
flip_elements <element list> #
```

FLIP SURFACES

Command:  `flip_surfaces`
Description: This command reverses the direction of surface normals. You must provide a list of surfaces.

Keyboard Command Sequence:
```plaintext
flip_surfaces <surface list> #
```

FLOW

Subroutine:  `FLOW`
Description: The user subroutine `FLOW` allows you to modify mass flow rate, inlet temperature, and film coefficient in heat transfer analyses involving fluid channel elements. Both the inlet temperature and mass flow rate can be dependent on time; the film coefficient can also be a function of streamline distance.
**Command:** flowlines_full  
**Description:** This command specifies that all flowline edges in the visible portion of the model be displayed.

Also see: flowlines_off, flowlines_outline, flowlines_surface, and flowlines_quad_edge.

**Keyboard Command Sequence:**

flowlines_full

---

**Command:** flowlines_off  
**Description:** This command specifies that no flowlines be displayed.

Also see: flowlines_full, flowlines_outline, flowlines_surface, and flowlines_quad_edge.

**Keyboard Command Sequence:**

flowlines_off

---

**Command:** flowlines_outline  
**Description:** This command specifies that only flowline edges on the outline of the visible portion of the model be displayed.

Also see: flowlines_full, flowlines_off, flowlines_surface, and flowlines_quad_edge.

**Keyboard Command Sequence:**

flowlines_outline
**Command:** `flowlines_quad_edge`  
**Description:** This command specifies which edges of quad element flowlines may be displayed. By default, all four edges are displayed. You must specify an edge (0, 1, 2, or 3) and whether or not it should be displayed.

Also see: `flowlines_full`, `flowlines_off`, `flowlines_outline`, and `flowlines_surface`.

**Keyboard Command Sequence:**

`flowlines_quad_edge <edge> <on or off>`

Other Buttons with the same description:

QUAD EDGE 1  
QUAD EDGE 2  
QUAD EDGE 3

---

**Command:** `flowlines_surface`  
**Description:** This command specifies that only flowline edges on the surface of the visible portion of the model be displayed.

Also see: `flowlines_full`, `flowlines_off`, `flowlines_outline`, and `flowlines_quad_edge`.

**Keyboard Command Sequence:**

`flowlines_surface`

---

**Subroutine:** `FLUX`  
**Description:** The user subroutine `FLUX` allows you to specify surface or body fluxes as functions of time, temperature, or position for heat transfer analysis. The use of this subroutine is flagged by the appropriate flux type in the DIST FLUXES or FLUXES input option where the type chosen depends on element type.
Command: **xcv_model foam**

Description: This command is used if experimental data must be fitted using the foam strain energy function $\bar{W}$, which is given by:

$$
\bar{W} = \sum_{n=1}^{N} \left( \frac{\mu_n}{\alpha_n} \left( \lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3 \right) + \frac{\mu_n}{\beta_n} \left( 1 - J^{\beta_n} \right) \right)
$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are the principal stretch ratios, $J$ is the determinant of the deformation gradient, $N$ is the number of terms and $\mu_n$, $\alpha_n$ and $\beta_n$ are the material parameters to be determined.

The maximum number of terms is 10, but it is recommended to use no more terms than necessary to get a sufficiently good fit.

This model should be used for highly compressible elastic materials. Except for the simple shear test, volumetric information must be available.

In order to perform a plausible extrapolation for the foam model, dilatational information is needed beyond the data set. This is achieved using linear extrapolation based on the two start and/or end points of the measured data. This linear extrapolation may restrict the validity of the response outside the range of the measured data.

For dual mode plotting (except for simple shear), dilatational information is needed for the foam model. For a volumetric test, this readily follows from the strain, but for uniaxial, biaxial, and planar shear tests this must be calculated. This calculation is based on the requirement that the stress in perpendicular direction must be zero. If the fitted coefficients do not fulfil this requirement, zero stresses are returned for such a dual mode.

Keyboard Command Sequence:

```
xcv_model foam
```
Subroutine:  FORCDF

Description:  The user subroutine FORCDF is used to input frequency dependent point loads or displacement histories. (Use the user subroutine FORCEM for distributed loads.)

FORCDF is flagged by introducing a model definition set (FORCDF) listing the node numbers for which this routine is called. Then, at each harmonic increment of the analysis, for each of the nodes on the list, the subroutine is called. Displacement and load arrays are available for stress analysis.

For dynamic analysis, velocity and acceleration are available.

For nodes without kinematic boundary conditions, you may define increments of point loads (thus overwriting any point load input in the same nodes in the point loads block).

For nodes with kinematic boundary conditions (i.e., listed in the FIXED DISP or DISP CHANGE blocks) you may define increments of harmonic displacement.
FORCDT

Subroutine: FORCDT

Description: The user subroutine FORCDT is used to input time dependent point loads or displacement histories. (Use the user subroutine FORCEM for distributed loads.)

FORCDT is flagged by introducing a model definition set (FORCDT) listing the node numbers for which this routine is called. Then, at each increment of the analysis, for each of the nodes on the list, the subroutine is called. Displacement and load arrays are available for stress analysis. For dynamic analysis, velocity, and acceleration are available.

For nodes without kinematic boundary conditions, you may define increments of point loads (thus overwriting any point load input in the same nodes in the point loads block).

For nodes with kinematic boundary conditions (i.e., listed in the FIXED DISP or DISP CHANGE blocks) you may define increments of displacement.

FORCEM

Subroutine: FORCEM

Description: The user subroutine FORCEM is used to input nonuniform distributed loads. This routine may be used to specify the load magnitude as a function of position and/or time.

FORCEM is called during the calculation of the equivalent nodal loads, at each integration point needed to calculate the loads specified in the DIST LOADS. If data cards are read by the user’s FORCEM, they should therefore appear after the corresponding END OPTION or CONTINUE card in the input deck.
Command: material_option

Description: This input option accepts the polynomial functions describe the FLD curve. There are 9 parameters (C1, C2, C3, C4, C5 and D1, D2, D3, D4) are necessary to form the curve. For details, please refer to MSC.Marc Volume A: Theory and User Information and MSC.Marc Volume C: Program Input.

Keyboard Command Sequence:

material_option <forming_limit:fitted>

Command: material_option

Description: This input option accepts the predicted FLD curve. The predicted functions are generated based on the theories about local necking (Hill) and diffuse (Swift) necking. Both theories assume that the material obeys the power-law strain hardening.

The input data:

The strain hardening exponent: n
The thickness coefficient: tc
= 3.59 if thickness is defined by [Inch],
= 1.41 if thickness is defined by [cm],
= 0.141 if thickness is defined by [mm],
and so on.

For details, please refer to MSC.Marc Volume A: Theory and User Information and MSC.Marc Volume C: Program Input.

Keyboard Command Sequence:

material_option <forming_limit:predicted>
TABLE

Command:  **material_option**

Description:  The TABLE function in MSC.Marc allows users to define any curves through TABLE model definition option. The independent variable is set as minor principal strain, the variable is the FLD values. For example, if user has FLD points of the material, it is possible to define the FLD as piecewise linear curve. Details refer to TABLE.

Forming limit parameter is the scale factor set for the Table, which is default as 1.0.

For details, please refer to *MSC.Marc Volume A: Theory and User Information* and *MSC.Marc Volume C: Program Input*.

Keyboard Command Sequence:

```
material_option <forming_limit:table>
```

FULL WINDOW

Command:  **full_window**

Description:  This command causes the current graphics window to occupy the entire area of the MSC.Marc Mentat parent window. The full window can be disabled pressing the escape (Esc) key or by entering the `full_window_exit` command. This command is useful when running demo procedure files or when using screen capture programs for producing hardcopy images of the graphics window.

Keyboard Command Sequence:

```
full_window
```

full_window_exit

Command:  **full_window_exit**

Description:  This command restores the default window layout if in full window mode.

Keyboard Command Sequence:

```
full_window_exit
```
Mentat Help Commands in G

GAPT

Subroutine: GAPT
Description: The user subroutine GAPT allows you to redefine gap temperature (TGAP) based on the nodal temperatures T1 and T2 in conjunction with the CONRAD GAP option. If the gap temperature (TGAP) is greater than or equal to the gap closure temperature (TCLOSE), then the gap is closed, otherwise the gap is open.

GAPU

Subroutine: GAPU
Description: The user subroutine GAPU allows you to input or modify the direction and closure distance of gap element type 12 and 97 based on the current position of the end nodes of the element. This makes it possible to model contact sliding along curved surfaces which may occur in the analysis of metal forming problems. Note that although the gap direction and closing distance can be changed,
**GAPU** does not allow for finite sliding of two meshes with respect to each other, since the load transfer path is unchanged. In addition, **GAPU** allows for specification of a nonlinear relationship between the normal force and the maximum friction force instead of the regular linear Coulomb relation.

**Note:** If this option is used to change the direction of the gap, then friction should NOT be included.

**GAPU** also allows you to specify certain tolerances to control gap closure and friction iterations. Note that this last feature will generally not be used.

---

**MATERIAL**

**Command:**  `set_gasket_membrane_mat`

**Description:** This command sets the material to be used for the membrane (or in-plane) behavior of the current gasket material. The membrane behavior of a gasket is elastic.

The material must be an **ISOTROPIC** material and only the thermo-elastic properties (Young’s modulus, Poisson's ratio, and the thermal expansion coefficients) of that material are being used.

**Keyboard Command Sequence:**

```
set_gasket_membrane_mat <material_name>
```

**Reference:** *MSC.Volume A: Theory and User Information*, Chapter 7: Section on Gaskets

*MSC.Volume C: Program Input*, Chapter 3: Model Definition Options, GASKET option.
**YIELD PRESSURE**

**Command:**

```plaintext
material_value gasket:yield_pressure
material_value gasket:tensile_modulus
material_value gasket:initial_gap
material_table gasket:load0
material_table gasket:unload_10
material_table gasket:unload_20
material_table gasket:unload_30
material_table gasket:unload_40
material_table gasket:unload_50
material_table gasket:unload_60
material_table gasket:unload_70
material_table gasket:unload_80
material_table gasket:unload_90
material_table gasket:unload_100
```

**Description:**

These commands set the properties in thickness direction of the gasket for the current gasket material. The behavior in the thickness direction of the gasket is either elastic or elastic-plastic.

The **YIELD PRESSURE** (`material_value gasket:yield_pressure`) is the pressure above which plastic deformation develops. If the yield pressure is not reached by the loading path (see below), the gasket will be fully elastic.

The **TENSILE MODULUS** (`material_value gasket:tensile_modulus`) is the elastic stiffness of the gasket when it is loaded in tension. The modulus is defined as pressure per unit closure distance.

The **INITIAL GAP** (`material_value gasket:initial_gap`) is the amount of closure the gasket elements must reached before building up pressure. It basically shifts the loading and unloading paths along the closure axis in the positive direction. This is useful if the thickness of (a part of) the actual gasket is smaller than the thickness of the elements used to model the gasket.

The **LOADING PATH** (`material_value gasket:load0`) is the relation between the pressure on the gasket and the closure of the gasket when the gasket is being loaded in compression. It must be a table of type `gasket_closure` and must start at zero pressure and zero closure.
The UNLOADING PATHS (material_value gasket:unload_10, material_value gasket:unload_20, etc) are relations between the pressure on the gasket and the closure of the gasket when the gasket unloads elastically, after having developed a certain amount of plastic deformation. Up to 10 unloading paths may be specified. If unloading occurs at plastic closures for which no path has been specified, the path will be constructed by interpolation from the two nearest paths. The unloading paths must be tables of type gasket_closure and start at zero pressure. The closure at zero pressure is the plastic closure for that unloading path and must be positive. If no unloading paths are defined, the gasket is fully elastic.

Keyboard Command Sequence:

```
mystery_value gasket:yield_pressure <value>
mystery_value gasket:tensile_modulus <value>
mystery_value gasket:initial_gap <value>
mystery_table gasket:load0 <table_name>
mystery_table gasket:unload_10 <table_name>
mystery_table gasket:unload_20 <table_name>
mystery_table gasket:unload_30 <table_name>
mystery_table gasket:unload_40 <table_name>
mystery_table gasket:unload_50 <table_name>
mystery_table gasket:unload_60 <table_name>
mystery_table gasket:unload_70 <table_name>
mystery_table gasket:unload_80 <table_name>
mystery_table gasket:unload_90 <table_name>
mystery_table gasket:unload_100 <table_name>
```

Reference: MSC.Volume A: Theory and User Information, Chapter 7: Section on Gaskets
MSC.Volume C: Program Input, Chapter 3: Model Definition Options, GASKET option.

Other Buttons with the same description:

INITIAL GAP
TABLE
TABLE 1
TABLE 10
TABLE 2
TABLE 3
TABLE 4
TABLE 5
TABLE 6
TABLE 7
TABLE 8
TABLE 9
TENSILE MODULUS
**MODULUS**

Command: `material_value gasket:tr_shear_modulus`

Description: This command sets the transverse shear modulus for the current gasket material. The transverse shear behavior is elastic.

Keyboard Command Sequence:

```
material_value gasket:tr_shear_modulus <value>
```

Reference: *MSC.Volume A: Theory and User Information*, Chapter 7: Section on Gaskets

*MSC.Volume C: Program Input*, Chapter 3: Model Definition Options, GASKET option.

**MESH CURVES**

Command: `generate_2d_rebar`

Description: This command automatically meshes the given list of curves with line rebar elements between intersections with existing 2-D mesh.

The 2-D elements are optionally placed into insert set as host elements, while the line elements are placed into insert set as embedded elements.

Also see help: `set_insert_create`

Keyboard Command Sequence:

```
generate_2d_rebar <curve list> <element list>
```
Command: **generate_csect_control_node**

**Description:** This command creates a control node and assigns it to the current cross-section. The position vector of the control node is given by the average position vectors of the cross-section nodes. In this way, defining boundary conditions on the control nodes and performing postprocessing might be easier if there are several cross-sections present in the model, since the position vector of each control node is related to the location of the corresponding cross-section.

Alternatively, if the user has defined a separate node in the model, the command **set_csect_control_node** can be used to assign this node to the current cross-section.

**Keyboard Command Sequence:**

```
generate_csect_control_node
```
Command: gent
Description: Elastomer Free Energy Function

Number of coefficients 2


Coefficient: E, I_m – Determined by the fitter

E: Small strain tensile modulus
I_m: maximum value for the first invariant of deformation (I1)

Warning: This phenomenological model is designed to exhibit finite extensibility of polymer chains and forces the stresses and tangent to asymptote to infinity as I1 approaches I_m. It is conceivable that during the solution stage, Marc evaluates a set of trial displacements such that I1 > I_m. In which case, the convergence ratios can be expected to oscillate, or even worse, the solution may not converge at all. Therefore, if I_m < 4, it is recommended that in a Marc job, users apply loads in small increments.

For this model, the calculation of the Bulk Modulus is not required by the Marc program. A volumetric test need not be supplied.

Menu: GEOMETRIC PROPERTIES
Description: This menu contains commands for defining sets of geometric properties and applying them to elements in the model. The types of properties that can be defined are truss area, beam properties, and shell thickness.
Menu: GEOMETRY DISTANCE

Description: This menu contains commands for computing the distance of nodes from geometric entities such as curves and surfaces. The results of this computation are made available as scalar and vector plots. Controls are available to control the accuracy and cost of the computation.

Command: geomdist_surfaces <surface list> <node list>
geomdist_curves <curve list> <node list>

For each given node, every given curve, or surface is searched to find the closest point. This search is done in two steps. First a coarse search is performed on a precomputed regular grid of points to find the closest point. Then a binary search is done to tolerance around that location. Tests are done to exclude points outside of trimming curves, and points which are not normal to the surface.

set_geomdist_grid_div <ndiv>
This command is used to set the size of the precomputation grid. For complex surfaces, it may be necessary to increase this number to avoid ambiguous results.

set_geomdist_tolerance <tol>
Set the tolerance to be used during the binary search used to find the closest point. A lower tolerance will improve the accuracy of the results.

set_geomdist_check_trim <on/off>
When searching surfaces having trimming curves, this command will control whether a surface will be excluded when the closest point is outside of the trimming loops. Default on.

set_geomdist_trim_div <ndiv>
When determining whether a point is inside the trimming loops, it is necessary to break each trimming curve into segments. This command controls the number of segments generated by each curve. A larger number of segments will increase the accuracy of this computation.

set_geomdist_trim_tolerance <tol>
This controls the detection of points which are outside of all trimming loops. Trimming loops are actually scaled down very slightly to allow
points which are on the boundary of the underlying surface to be outside of the trimming loops.

`set_geomdist_check_normal <on/off>`

This command controls normal checking once the closest point on the curve or surface has been found. This is done by comparing the vector between the node and the closest point, and the curve or surface normal. Default on.

`set_geomdist_normal_tolerance <tol>`

This command sets the tolerance used for normal checking.

Other Buttons with the same description:

- CHECK NORMAL
- CHECK TRIMMING CURVES
- CURVE DISTANCES
- GRID DIV.
- NORMAL TOL.
- RESET
- SURFACE DISTANCES
- TOLERANCE
- TRIM DIV.
- TRIM TOL.

---

**Command:** `geometry_name`

**Description:** This command sets or changes the name of the current set of geometric properties.

An element geometry is a collection of the geometric properties to be applied to a set of elements. Geometric properties include but are not limited to shell thickness, beam and truss areas, and beam moments of inertia.

**Keyboard Command Sequence:**

`geometry_name <geometry name>`
**Command:**  geometry_option

**Description:**  This command sets the value of a geometry option. You must specify the option to be set. Geometry options are specified as the name of the option and its value separated by a colon (e.g. shape:curved).

The valid geometry options and their values are as follows:

- **addprops**: off, on  
  Addition properties specified.
- **section**: circular, general  
  Cross section type.
- **shape**: curved, straight  
  Elbow shape.
- **shellthick**: constant, variable  
  Shell thickness variation.
- **flat**: off, on  
  Element technology.
- **cdilatation**: off, on  
  Constant dilitation.
- **assumedstrn**: off, on  
  Assumed strain.
- **beamheight**: constant, variable  
  Beam height variation.
- **gaptype**: fixeddir, truedist  
  Gap/Friction link type (fixed direction or true distance).
- **distance**: minimum, maximum  
  True distance type.
- **gapcondition**: open, closed  
  Gap condition at increment 0.
- **ctemperature**: off, on  
  Constant temperature.

**Keyboard Command Sequence:**

```
geometry_option <geometry option>::<value>
```

**Other Buttons with the same description:**

ASSUMED STRAIN
CIRCULAR
CLOSED
CONSTANT
CONSTANT DILATATION
**geometry_reset**

**Command:**  `geometry_reset`

**Description:** This command resets all geometry values and options for the current geometry to their default values.

**Keyboard Command Sequence:**

`geometry_reset`

**BEAM SECTION**

**Command:**  `geometry_section`

**Description:** This command assigns a beam section to the current geometry. Beam section may be assigned to 3-D general beams only.

**Keyboard Command Sequence:**

`geometry_section <beam section name>`
Command:  geometry_type

Description:  This command selects the type of the current geometry property. You must specify the type of geometry. The valid geometry types are as follows:

heat_three_truss
heat_three_shell
heat_three_solid
heat_three_comp_cont
heat_axisym_shell
heat_axisym_solid
heat_axisym_comp_cont
heat_planar
heat_planar_comp_cont
magn_three_solid
mech_three_truss
mech_three_cable
mech_three_beam_ela
mech_three_beam_gen
mech_three_elbow
mech_three_membrane
mech_three_shear
mech_three_shell
mech_three_solid
mech_three_rebar
mech_three_membr_rebar
mech_three_comp_cont
mech_axisym_shell
mech_axisym_solid
mech_axisym_rebar
mech_axisym_comp_cont
mech_gap
mech_planar_beam_str
mech_planar_beam_cur
mech_planar_pstress
mech_planar_pstrain
Keyboard Command Sequence:

```
geometry_type <geometry type>
```

Other Buttons with the same description:

- CABLE
- ELASTIC BEAM
- GENERAL BEAM
- ELBOW
- MEMBRANE
- SHEAR PANEL
- SHELL
- SOLID
- SOLID COMPOSITE
- 2-D STRAIGHT BEAM
- 2-D CURVED BEAM
- PLANЕ STRAIN
- PLANЕ STRAIN COMPOSITE/GASKET
- GAP/FRICTION LINK
- 3-D SOLID

Command: **geometry_value**

Description: This command sets geometric property values. You must specify the name of the property to be set and a value. Valid geometry property names are as follows:

- **area**: Truss or beam area.
- **momentx**, **momenty**: Beam moments of inertia.
- **orientx**, **orienty**, **orientz**: X, Y, and Z components of the beam orientation vector.
- **thickness**: Shell thickness.
- **dilation**: Shell dilation option. (0 = off, 1 = on)
- **assumed_strain**: Shell assumed strain option. (0 = off, 1 = on)
Keyboard Command Sequence:

```
geometry_value <property name> <value>
```

Other Buttons with the same description:

- AREA
- AT 1ST CORNER NODE
- AT 2ND CORNER NODE
- AT 3RD CORNER NODE
- AT 4TH CORNER NODE
- BENDING RADIUS
- CENTER POINT X
- CENTER POINT Y
- CENTER POINT Z
- CLOSURE DISTANCE
- CONTACT RADIUS
- DISTANCE
- EFF. T-SHEAR AREA Ax
- EFF. T-SHEAR AREA Ay
- EMISSIVITY
- FILM COEFFICIENT
- FRICTION COEFFICIENT
- FRICTION DIRECTION X
- FRICTION DIRECTION Y
- FRICTION DIRECTION Z
- GAP DIRECTION X
- GAP DIRECTION Y
- GAP DIRECTION Z
- HEIGHT
- INITIAL STRESS
- Ixx
- Iyy
- LENGTH
- MOMENTUM RATIO 1ST NODE
- MOMENTUM RATIO 2ND NODE
- PENALTY
- RADIUS
- STIFFNESS IN FRICTION DIRECTION
- STIFFNESS IN GAP DIRECTION
- TORSIONAL STIFFNESS FACTOR
- TRANSITION THICKNESS
- VECTOR DEF. LOCAL X-AXIS X
- VECTOR DEF. LOCAL X-AXIS Y
- VECTOR DEF. LOCAL X-AXIS Z
- WIDTH
- X
- Y
- Z
GLOBAL REMESHING

Menu: GLOBAL REMESHING

Description: This menu contains commands for defining the parameters used to control the global remeshing.

Global remeshing allows different parts of the model to be remeshed. A new mesh is automatically created, relevant solution quantities are transferred to the new mesh and the analysis continues.

This feature is currently only supported for contact bodies and only boundary conditions applied with contact bodies are allowed on the remeshed body. The remeshing is performed separately for each contact body.

Only 2-D problems can use remeshing in this release.

U DOMAIN

Command: grid_u_domain
grid_v_domain
grid_w_domain
grid_u_spacing
grid_v_spacing
grid_w_spacing
grid_u_position
grid_v_position
grid_w_position
grid_fix_u
grid_fix_v
grid_fix_w
grid_style_dots
grid_style_lines
grid_maxpoints
Description: These commands manipulate the grid representing the user’s coordinate system. In Mentat, grids are always shown in two dimensions. Therefore, for any coordinate system, the grid is displayed with one variable fixed with the other two variables. The commands `grid_fix_u`, `grid_fix_v`, and `grid_fix_w` specify which variable is to be fixed.

The following table shows the definitions of u, v, and w for the different coordinate systems:

<table>
<thead>
<tr>
<th>Coord. Sys. Type</th>
<th>u</th>
<th>v</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>Radius</td>
<td>Theta</td>
<td>Z</td>
</tr>
<tr>
<td>Spherical</td>
<td>Radius</td>
<td>Theta</td>
<td>Phi</td>
</tr>
</tbody>
</table>

The commands `grid_u_domain`, `grid_v_domain`, and `grid_w_domain` are used to specify the domains of the free variables in the grid.

The commands `grid_u_spacing`, `grid_v_spacing`, and `grid_w_spacing` are used to specify the grid intervals for the respective free variables.

The commands `grid_u_position`, `grid_v_position`, and `grid_w_position` are used to specify the values of the respective fixed variables.

The commands `grid_style_dots` and `grid_style_lines` specify the grid style.

The command `grid_maxpoints` specify the upper limit on grid points (or grid line intersections) to be displayed.

Also see: `set_grid`, `origin`, `origin_x`, `origin_y`, `origin_z`, `system_align`, `system_reset`, `system_translate`, `system_rotate`.

Keyboard Command Sequence:

```
grid_u_domain <min u value> <max u value>
grid_v_domain <min v value> <max v value>
grid_w_domain <min w value> <max w value>
grid_u_spacing <u spacing>
grid_v_spacing <v spacing>
grid_w_spacing <w spacing>
```
grid_u_position <u position>
grid_v_position <v position>
grid_w_position <w position>

grid_fix_u
grid_fix_v
grid_fix_w

grid_style_dots
grid_style_lines

grid_maxpoints <max # points>

Other Buttons with the same description:
DOTS
FIX U
FIX V
FIX W
LINES
MAX POINTS
U
U SPACING
V
V DOMAIN
V SPACING
W
W DOMAIN
W SPACING
Mentat Help Commands in H

Command: help
Description: This command displays the help text for the specified command.
Keyboard Command Sequence:
    help <command>
Menu: **MSC.Marc Mentat Help**

Description: The HELP button located in the far right of the STATIC menu area provides a menu with links to the MSC.Marc online documentation. The submenu RUN A DEMO PROBLEM contains links for several demonstration problems.
Command:  **help_pdf**

Description:  This command toggles between a text-based online help system and a system via PDF files. The latter requires that a PDF-file reader (such as the Acrobat reader) is installed on the system. The online help system can also be selected at startup via the `-ha` command line option. In that case, `-ha true` selects the PDF-based system and `-ha false` selects the text-based system. The default is to use the PDF files.

The button is located in the HELP menu which can be opened from the HELP button in the far right of the STATIC menu area.

Keyboard Command Sequence:

```
help_pdf <on/off>
table_unfilled
```

Command Line Option:

```
-ha true/false
```

Other Buttons with the same description:

PDF

---

Command:  **hexer_detect_edges**

Description:  This command is used to automatically detect geometric edges in a list of triangular and quadrilateral elements which enclose a volume. The elements in the input list must be oriented with their tops facing outward, and there must not be any free edges or holes existing in the list.
Detected edges are used by HexMesh to help define geometry. The resulting hexahedral mesh will not have any element faces which cross these edges. Detected element edges are left selected by hexer_detect_edges, and then the list of selected edges may be modified before using HexMesh. Proper selection of these edges is important so that the mesh will accurately represent the geometry of the input surface mesh.

Also see help: hexer_hexmesh and hexmesh_controls.

**Keyboard Command Sequence:**

```
hexer_detect_edges <element list>
```

---

**Command:**  **hexer_hexmesh**

**Description:** This command runs the automatic hexahedral mesh generator, HexMesh. For input, a list of triangular and quadrilateral elements which enclose a volume is required. The elements must be oriented with their tops facing outward, and there must not be any free edges or holes existing in the list.

If any element edges are selected, then these edges will be used by the mesher to help define geometry. The resulting hexahedral mesh will not have any element faces which cross these edges. Proper selection of these edges is important so that the mesh will accurately represent the geometry of the input surface mesh.

If no element edges are selected, then the mesher will automatically detect edges according to the current edge sensitivity parameter as part of the meshing process.

Also see help: hexmesh_controls, hexer_detect_edges.

**Keyboard Command Sequence:**

```
hexer_hexmesh <element list>
```
**ELEMENT SIZE**

**Command:**  `hexmesh_controls`

**Description:** The following parameters relate to HexMesh:

- **ELEMENT SIZE (XYZ)**: The desired size of the hexahedral elements being generated in the x, y, and z-directions.

- **EDGE SENSITIVITY**: Used during edge detection to determine when the shared edge between two input elements represents a real edge. A value of one means that all element edges between elements which do not lie in the same plane will be treated as real edges. A value of zero means that no real edges will be found.

- **GAP**: Nonzero values for this parameter will affect the size of the gap which is initially left between the inner (kernel) hexahedral elements and the surface during mesh generation. Negative values will result in a smaller gap (even penetration) and positive values will result in a wider gap. The value of this parameter must be between -1 and 1.

- **SHAKES**: The number of times to *shake* the mesh, where shaking is a method of global mesh enhancement.

- **RUNS**: The maximum number of times the mesher is allowed to run. The mesher reruns with a smaller element size if it fails.

- **COARSENING LEVELS**: To reduce the number of generated elements, larger elements may be produced in the interior of the mesh. Tying equations are used to maintain compatibility of neighboring elements. A value of zero indicates that no coarsening will occur, while a value of two indicates that the elements in the interior can be up to four times larger on each side than elements on the surface. Acceptable values are 0, 1, and 2.
LOCAL ENHANCE  Setting this parameter causes the mesher to perform some additional mesh enhancement within localized areas.

COONS PATCHES  Turning on this option allows a smoother representation of the input surface to be used, resulting in a better approximation of the input geometry than by just using quadrilaterals and triangles. This approach is however more costly in CPU time.

ALLOW WEDGES  When this option is on, the creation of wedge elements is allowed if an edge crosses the diagonal of a face of the hexahedral element.

Also see help: hexer_detect_edges, hexer_hexmesh.

Keyboard Command Sequence:

set_hexer_element_size <size_x> <size_y> <size_z>
set_hexer_edge_sensitivity <sensitivity>
set_hexer_gap <gap>
set_hexer_shakes <number of shakes>
set_hexer_runs <number of runs>
set_hexer_coarsening_levels <number of levels>
set_hexer_local_enhance <y or n>
set_hexer_coons_patches <y or n>
set_hexer_allow_wedges <y or n>

Other Buttons with the same description:

0
1
2
ALLOW WEDGES
COONS PATCHES
EDGE SENS.
GAP
RUNS
SHAKES
**ADD 1-NODE CURVE**

**Command:** history_add  

**Description:** This command adds a single curve based on a single node to a history plot. You must already have collected history plotting data (see set_history_nodes and history_collect). You must specify the node for which you want the plot, x-axis and y-axis variables.  

**Keyboard Command Sequence:**  

```
history_add <node> <X-variable> <Y-variable>
```

**ADD 2-NODE CRV**

**Command:** history_add_2nodes  

**Description:** This command adds a single curve based on two nodes to a history plot. You must already have collected history plotting data (see set_history_nodes and history_collect). You must specify the x-axis node, x-axis variable, y-axis node, and y-axis variable.

**Keyboard Command Sequence:**  

```
history_add_2nodes <X-node> <X-variable> <Y-node> <Y-variable>
```

**ADD NODE**

**Command:** history_add_node  

**Description:** This command adds a family of curves based on a single node to a history plot. You must already have collected history plotting data (see set_history_nodes and history_collect). One curve for every result variable in the results file will be added. You must specify the node for which you want the plot and x-axis variable.  

**Keyboard Command Sequence:**  

```
history_add_node <node> <X-variable>
```
ADD GLOBAL CRV

Command: **history_add_nodeless**

Description: This command adds a single history curve based on data not associated with any node, such as die forces and optimization data, to a history plot. You must already have collected history plotting data (see `history_collect`). You must specify the x-axis and y-axis variables.

Keyboard Command Sequence:

```
history_add_nodeless <X-variable> <Y-variable>
```

ADD VARIABLE

Command: **history_add_var**

Description: This command adds a family of curves each based on a different single node to a history plot. You must already have collected history plotting data (see `set_history_nodes` and `history_collect`). One curve for every node in the model will be added. You must specify the x-axis and y-axis variables.

Keyboard Command Sequence:

```
history_add_var <X-variable> <Y-variable>
```

CLEAR CURVES

Command: **history_clear**

Description: This command clears all curves from the history plot.

Keyboard Command Sequence:

```
history_clear
```
Command: **history_collect**

Description: This command collects from the current results file from which history plots can be made. Data may be collected from every increment in the file or from regularly spaced increments. You must provide the first and last increment numbers and an increment step size. Increment numbers are in the form **Inc:Sub**. That is, an increment number followed by a colon and the subincrement number. If the colon and subincrement number are left off, then the subincrement is assumed to be 0.

Note that you can collect either from **Inc1:Sub** to **Inc2:Sub** or from **Inc:Sub1** to **Inc:Sub2**, so incremental and subincremental data cannot be collected together.

Keyboard Command Sequence:

```
history_collect <first increment> <last increment> <step size>
```

Another Button with the same description:

**COLLECT GLOBAL DATA**

Command: **history_filled**

Description: This command causes areas under history curves to be unfilled. To fill plots with color, toggle this button ON to use the **history_filled** command.

Keyboard Command Sequence:

```
history_filled
```
**Command:**  
**history_fit**

**Description:** This command alters the limits of the history plots so that all values fit onto the screen.

**Keyboard Command Sequence:**

```
history_fit
```

---

**Menu:**  
**HISTORY PLOT PRESETS**

**Description:** The commands in this menu add curves with predefined X- and Y-variable to a history plot.

You do not need to have the collected history plotting data.

Commands for which the X- and/or Y-variable represent contact body data, add a curve for each contact body in the model.

**Keyboard Command Sequence:**

```
history_add_body_force_time
history_add_body_force_body_pos
history_add_body_force_body_angle
history_add_body_force_press_angle (only AutoForge)
```

Other Buttons with the same description:

**BODY FORCE vs. BODY POSITION**
**BODY FORCE vs. TIME**
**PRESETS**
**REMOVE CURVE**

**Command:** history_remove  
**Description:** This command removes curves from the history plot. You must specify the curve to be removed.  
**Keyboard Command Sequence:**  

```
history_remove <history curve number>
```

**CONVERT**

**Command:** history_table  
**Description:** This command creates a table from the values in the specified history curve. History curves are specified by number.  
**Keyboard Command Sequence:**  

```
history_table <history curve number>
```

**FILLED**

**Command:** history_unfilled  
**Description:** This command causes areas under history curves to be unfilled. To fill plots with color, toggle this button ON to use the history_filled command.  
**Keyboard Command Sequence:**  

```
history_unfilled
```
Command: **history_write**

**Description:** This command writes the current history curves to a file. You must specify the name of the file to receive the data.

**Keyboard Command Sequence:**

```
history_write <file name>
```
Command: hitchcock_out

Description: This command moves the camera away from the lookat point, and zooms the camera in. First, if the projection is currently orthographic, it is changed to perspective. Then, the camera is moved so that its distance from the lookat point is multiplied by the current zoom increment. Finally, the camera zoom is multiplied by the current zoom increment. This has the effect of decreasing the amount of perspective in the view. After using this command, objects behind the lookat point may appear larger than before, objects at the lookat point will look the same, and objects in front of the lookat point may appear smaller.

This command acts on all the currently active views.

Also see: hitchcock_in, view_perspective, view_viewpoint, zoom_increment, view_zoom_factor, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

hitchcock_out

Menu: HOLES IN PLATE (Python)

Description: This button will run the demonstration problem for Python. It is the example shown in the MSC.Marc Python Tutorial: Chapter 4, describing a parametric design example in which the number of holes to place in a plate are determined by the Python script.

Selecting this button brings up a dialog box that contains 4 text boxes. The variables assigned to these boxes represent the width, height, hole diameter, and hole spacing values.

To run this example, you must set your current directory to be examples/python/tutorial/c04.
Subroutine: **HOOKLW**

**Description:** The user subroutine **HOOKLW** is an alternative mechanism to user subroutine **ANELAS**. In this routine, you supply the elastic stress-strain law. This law is given in terms of the coordinate system defined in the **ORIENTATION** option. You should insure that the stress-strain law is symmetric. Note that this routine will be called for each integration point of those elements that have anisotropic properties. You may define either the stress-strain relation or the compliance strain-stress relation. The returned value of argument IMOD must be set accordingly.

Subroutine: **HOOKVI**

**Description:** The user subroutine **HOOKVI** allows you to modify the time dependent properties of an orthotropic material specified through the **VISCELORTH** Model Definition option. **HOOKVI** is automatically called for every material defined in that option.

Subroutine: **HYPELA**

**Description:** The user subroutine **HYPELA** allows you to define elastic constants as functions of elastic strain and state variables.
Mentat Help Commands in I

### Commands Begin with I

**Command:** icond_all_layers
**Description:** This command activates all element layers for the current state variable initial condition.

**Keyboard Command Sequence:**
```
icond_all_layers
```

**Command:** icond_dof, clear_icond_dof
**Description:** These commands add and clear the specified degree of freedom from the current initial condition.

**Keyboard Command Sequence:**
```
icond_dof <dof name>
clear_icond_dof <dof name>
```
Command: **icond_layers**

**Description:** This command activates element layers for the current state variable initial condition. Element layers are specified as integers beginning at 1.

**Keyboard Command Sequence:**

```
icond_layers <layer numbers>
```

Command: **icond_name**

**Description:** This command sets or changes the name of the current initial condition.

A initial condition is a boundary condition that is applied to the mesh at the beginning of the analysis. Each initial condition contains the degrees of freedom and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected.

**Keyboard Command Sequence:**

```
icond_name <initial condition name>
```

Command: **icond_option**

**Description:** This command sets the value of an initial condition option. You must specify the option to be set. Initial condition options are specified as the name of the option and its value separated by a colon (e.g. `bin_post_file:ieee`, for post file format of IEEE).

The valid initial condition options and their values are as follows:

- `bin_post_file` native, ieee: Specifies that the format of the post file is either native binary format or IEEE (Cray only).
- `post_file_inc` number, last: The option `last` specifies that the last increment on the post file is to be used for the data, otherwise the increment number is specified.
post_file_pos time, inc

The option time specifies that a time mark on the post file is where the data will be obtained, otherwise an increment is used.

dof_values entered, post_file, usersub

Specifies where the data is obtained. If the option entered is specified, then the data is entered manually.
If the option post_file is used, the data is obtained from a previously generated post file.
If the option usersub is specified, then the data is obtained from a user subroutine.

Keyboard Command Sequence:
icond_option <icond option>:<value>

Other Buttons with the same description:
BY NUMBER
ENTERED VALUES
IEEE
INCREMENT
LAST
POST FILE
TIME
USER SUB.

INCREMENT

Command: icond_param_value post_file_inc | post_file_time |
state_var_id | exp_repetitions >

Description: This command sets the initial condition values for the following:

post_file_inc Increment at which the initial condition is read.
post_file_time Time to read the post file.
state_var_id State variable id for state variable being set.
exp_repetitions Number of repetitions in the theta direction.

These settings are available in the INITIAL CONDITIONS->MECHANICAL->STATE VARIABLE and also AXISYMMETRIC TO 3D initial condition menus.

Keyboard Command Sequence:
icond_param_value post_file_inc <value>
icond_param_value post_file_time <value>
icond_param_value state_var_id <value>
icond_param_value exp_repetitions <value>

Other Buttons with the same description:
SET
STATE VARIABLE ID
EXPAND REPETITIONS

Command: **icond_post_file**
Description: This command specifies the name of the post file to be used for reading values for the current state variable initial condition.

**Keyboard Command Sequence:**
icond_post_file <post file name>

Command: **icond_post_increment**
Description: This command specifies the first increment to be read from the post file to provide values for the state variable initial condition.

**Keyboard Command Sequence:**
icond_post_increment <increment number>

Another Button with the same description:
SET
Command: **icond_type**

**Description:** This command selects the type of the current initial condition. If the current initial condition does not have any values set, then this command simply changes the initial condition type. If the current initial condition has values set, then this command creates a new initial condition in addition to setting the type. (This feature saves users from having to manually create new initial conditions using the new_icond command.) The text on the button will be the initial condition type.

Initial condition types:

- displacement
- plastic_strain
- porosity
- relative_density
- temperature
- velocity
- pore_pressure
- void_ratio
- nodal_temperature
- preconsolidation
- state_variable
- point_mass
- stress
- porosity
- void_ratio
- nodal_temperature
- preconsolidation
- state_variable
- point_mass
- stress

**Keyboard Command Sequence:**

```
icond_type <boundary condition type>
```

---

Command: **icond_value**

**Description:** This command sets the value for a particular quantity in the current initial condition. You must specify the quantity name and its value.

**Keyboard Command Sequence:**

```
icond_value <quantity or dof name> <value>
```
STATE VARIABLE ID

Command: **icond_variable**

Description: This command specifies the identifier of the state variable for the current state variable initial condition. Temperature is always the first state variable at a point.

Keyboard Command Sequence:

```
icond_variable <state variable id>
```

ALL

Command: **ideas Elem All Layers**

Description: This command sets all layers for a selected I-DEAS element variable. Values for this variable are written at all layers. The number of layers per shell element is set with the command `job_param layers`.

Keyboard Command Sequence:

```
ideas Elem All Layers
```

DEF

Command: **ideas Elem Default Layer**

Description: This command sets the default layer for a selected I-DEAS element variable. Values for this variable are written at the default layer only.

Keyboard Command Sequence:

```
ideas Elem Default Layer
```
Command: ideas_elem_layers
Description: This command sets the layers for a selected I-DEAS element variable. Values for this variable are written at the specified layers only.

Keyboard Command Sequence:
ideas_elem_layers <layers>

Command: ideas_elem_outer_layers
Description: This command sets the outer and middle layers for a selected I-DEAS element variable. Values for this variable are written at the outer and middle layers only.

Keyboard Command Sequence:
ideas_elem_outer_layers

Command: ideas_elnod_all_layers
Description: This command sets all layers for a selected I-DEAS element-at-node variable. Values for this variable are written at all layers. The number of layers per shell element is set with the command job_param layers.

Keyboard Command Sequence:
ideas_elnod_all_layers
Command: ideas_elnod_default_layer
Description: This command sets the default layer for a selected I-DEAS element-at-node variable. Values for this variable are written at the default layer only.

Keyboard Command Sequence:

    ideas_elnod_default_layer

Command: ideas_elnod_layers
Description: This command sets the layers for a selected I-DEAS element-at-node variable. Values for this variable are written at the specified layers only.

Keyboard Command Sequence:

    ideas_elnod_layers <layers>

Command: ideas_elnod_outer_layers
Description: This command sets the outer and middle layers for a selected I-DEAS element-at-node variable. Values for this variable are written at the outer and middle layers only.

Keyboard Command Sequence:

    ideas_elnod_outer_layers
Command: **identify_adapgs**  
Description: This command turns on the identification of entries in the list of global remeshing criteria.

Keyboard Command Sequence:  
```
identify_adapgs
```

Another Button with the same description:  
```
ID GLOBAL REMESHING CRITERIA
```

Command: **identify_adapts**  
Description: This command turns on the identification of adaptivity criteria.

Keyboard Command Sequence:  
```
identify_adapts
```

Another Button with the same description:  
```
LCL ADPT CRIT
```

Command: **identify_applys**  
Description: This command turns on the identification of boundary condition applications.

Keyboard Command Sequence:  
```
identify_applys
```

Other Buttons with the same description:  
```
BOUNDARY CONDS  
ID BOUNDARY CONDS
```
Command: **identify_backfaces**

**Description:** This command turns on the identification of back-facing polygons. Front/inside denotes the inside of the rigid body. The contacting body should be on the outside part of the rigid surface, by default colored brown.

**Keyboard Command Sequence:**

```plaintext
identify_backfaces
```

Another Button with the same description:

```plaintext
ID BACKFACES
```

---

Command: **identify_classes**

**Description:** This command turns on the identification of classes of polygons.

**Keyboard Command Sequence:**

```plaintext
identify_classes
```

Another Button with the same description:

```plaintext
ID CLASSES
```

---

Command: **identify_contact**

**Description:** This command turns on the identification of contact bodies.

**Keyboard Command Sequence:**

```plaintext
identify_contact
```

Another Button with the same description:

```plaintext
ID CONTACT
```
Command: **identify_domains**

Description: This command turns on the identification of domain decomposition element domains.

Keyboard Command Sequence:

```
identify_domains
```

Another Button with the same description:

ID DOMAINS

---

Command: **identify_geometries**

Description: This command turns on the identification types of geometry contained in the model.

Keyboard Command Sequence:

```
identify_geometries
```

Another Button with the same description:

ID GEOMETRIES

---

Command: **identify_iconds**

Description: This command turns on the identification of initial boundary conditions.

Keyboard Command Sequence:

```
identify_iconds
```

Other Buttons with the same description:

INIT CONDS
INITIAL CONDS
Command: **identify_inserts**
**Description:** This command turns on the identification of inserts.
**Keyboard Command Sequence:**
```
identify_inserts
```

Command: **identify_materials**
**Description:** This command turns on the identification of materials.
**Keyboard Command Sequence:**
```
identify_materials
```
Another Button with the same description:
```
MATERIALS
```

Command: **identify_none**
**Description:** This command turns off the identification feature.
**Keyboard Command Sequence:**
```
identify_none
```
Command: **identify_sets**

**Description:** This command turns on the identification of set identifiers of all sets in the model.

**Keyboard Command Sequence:**

```
identify_sets
```

Another Button with the same description:

```
SETS
```

Command: **identify_types**

**Description:** This command turns on the identification of Marc element types in the model.

**Keyboard Command Sequence:**

```
identify_types
```

Another Button with the same description:

```
TYPES
```

Command: **image_display**

**Description:** This command brings up a new window displaying an image file. The image is displayed at its normal size.

To make the image window go away, move the mouse to the image window and press q or the ESC key.

You must specify the image file name.

Also see: **image_save_rgb**, **render**, and **image_slide**.
Keyboard Command Sequence:
```
image_display <file name>
```

Another Button with the same description:
```
IRIS RGB
```

---

**Command:** `image_ps_height`

**Description:** This command allows you to specify the height of the page in inches for subsequent PostScript plotting of image files.

- You must specify the page height.
- Also see: `image_save_ps`, `render`, `image_ps_print`, and `image_ps_width`.

Keyboard Command Sequence:
```
image_ps_height <height>
```

---

**Command:** `image_ps_print`

**Description:** This command sends a PostScript file containing a copy of a specified image file to a specified printer. Each printer may be configured by editing the appropriate file in the tools directory.

- You must specify the printer number and image file name.
- Also see: `image_save_ps`, `render`, and `image_ps_write`.

Keyboard Command Sequence:
```
image_ps_print <printer number> <file name>
```

Other Buttons with the same description:
```
PRINT 2
PRINT 3
```
**PAGE WIDTH**

Command:  `image_ps_width`

Description: This command allows you to specify the width of the page in inches for subsequent PostScript plotting of image files.

You must specify the page width.

Also see: `image_save_ps`, `render`, `image_ps_print`, and `image_ps_height`.

**Keyboard Command Sequence:**

```
image_ps_width <width>
```

**image_ps_write**

Command:  `image_ps_write`

Description: This command converts a specified image file into a PostScript file.

You must specify the image file name.

Also see: `image_save_ps`, `render`, and `image_ps_print`.

**Keyboard Command Sequence:**

```
image_ps_write <file name> <yes/no>
```

**X ORIGIN**

Command:  `image_ps_xorg`

Description: This command allows you to specify the X origin of the image in inches for subsequent PostScript plotting of image files.

You must specify the image X origin.

See also `image_save_ps`, `render`, `image_ps_print`, and `image_ps_yorg`.

**Keyboard Command Sequence:**

```
image_ps_xorg <X origin>
```
Y ORIGIN

Command:  **image_ps_yorg**

Description:  This command allows you to specify the Y origin of the image in inches for subsequent PostScript plotting of image files.

You must specify the image Y origin.

Also see:  `image_save_ps`, `render`, `image_ps_print`, and `image_ps_xorg`.

**Keyboard Command Sequence:**

```
image_ps_yorg <Y origin>
```
Command:  **image_save_bmp**  
Description:  This command captures the appearance of the graphics window into an MS-Windows BMP image file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture, and the image file name.

Also see:  *image_save* and *set_buffered*.

**Keyboard Command Sequence:**

```
image_save_bmp <view> <file name> <yes or no>
```

Other Buttons with the same description:

2
3
4

---

Command:  **image_save_gif**  
Description:  This command captures the appearance of the graphics window into a GIF image file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.
You must specify the number of the view to capture and the image file name.

Also see: `image_save` and `set_buffered`.

**Keyboard Command Sequence:**

```
image_save_gif <view> <file name> <yes or no>
```

Other Buttons with the same description:

2
3
4

**Command:** `image_save_jpeg`

**Description:** This command captures the appearance of the graphics window into a JPEG image file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture and the image file name.

Also see: `image_save`, `set_buffered`, `jpeg_quality`, and `jpeg_smoothing`.

**Keyboard Command Sequence:**

```
image_save_jpeg <view> <file name> <yes or no>
```

Another Button with the same description:

2
3
4
Command:  image_save_ppm

Description:  This command captures the appearance of the graphics window into a PPM image file. This is like taking a snapshot of the graphics area. The PPM image file format is used by the MPEG encoder to generate an MPEG animation file.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture, and the image file name.

Also see: image_save, set_buffered.

Keyboard Command Sequence:

image_save_ppm <view> <file name> <yes or no>

Other Buttons with the same description:

2
3
4
Command: **image_save_ps**

**Description:** This command captures the appearance of the graphics window into a Postscript file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture and the image file name.

Also see: image_save and set_buffered.

**Keyboard Command Sequence:**

```
image_save_ps <view> <file name> <yes or no>
```

Another Button with the same description:

2
3
4
Command:  `image_save_rgb`

Description: This command captures the appearance of the graphics window into an IRIS RGB format file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture and the image file name.

Also see: `image_display`, `image_slide`, `set_buffered`, and `render`.

Keyboard Command Sequence:

```
image_save_rgb <view> <file name> <yes or no>
```

Other Buttons with the same description:

2
3
4
Command:  image_save_tiff

Description:  This command captures the appearance of the graphics window into a TIFF image file. This is like taking a snapshot of the graphics area.

Make sure the graphics window is not obscured by another window while the snapshot is taken.

On some systems, the quality of the saved image may be improved by switching the program into single-buffered mode before taking the snapshot.

You must specify the number of the view to capture and the image file name.

Also see:  image_save and set_buffered.

Keyboard Command Sequence:

image_save_tiff <view> <file name> <yes or no>

Other Buttons with the same description:

2
3
4
**Command:**  image_slide

**Description:** This command brings up a new window displaying an image file. The image is displayed at a size that will fill your screen.

To remove the image window, move the mouse to the image window, and press q or the ESC key.

You must specify the image file name.

Also see: image_save, render, and image_display.

**Keyboard Command Sequence:**

```
image_slide <file name>
```

Another Button with the same description:

SLIDE

---

**Subroutine:** IMPD

**Description:** The user subroutine IMPD makes the displacements, coordinates, reaction forces, velocities, and accelerations available at each increment so that you may save them in any form convenient for your postprocessing. During harmonic subincrements, IMPD allows you to obtain the complex displacements and reactions. In heat transfer (or Joule heating) analysis, this subroutine allows you to obtain nodal temperatures, fluxes, and voltages for your postprocessing. This routine is used in conjunction with the UDUMP option.
Command: `import`

Description: Here are the different data types that can be translated into Mentat.

```
MARC   I-DEAS   PATRAN
ACIS   IGES    VDAFS
DXF    NASTRAN
```

Keyboard Command Sequence:

```
import <Data type> <File name>
```

Command: `import acis`

Description: This command imports an ACIS file. You must specify the name of the file.

- `FORMAT`
  - When this is set, it reads in a Binary file (SAB). The default reads in a ASCII file (SAT).

- `CHECK STATUS`
  - On default, the reader will not check for any errors in the imported file. If this option is turn on, entities in the file are checked for geometric, topological, and data structure errors. Furthermore, error messages are written to a file.

Keyboard Command Sequence:

```
import acis <ACIS file name>
```

Note: ACIS v3.0 and earlier version are supported.

Other Buttons with the same description:

```
READ CHECK LOG
acis_screen
```
**Command:** import cmold

**Description:** This command imports a C-MOLD finite element model. It reads data from four C-MOLD files:

- the parameter file (extension .par or .PAR)
- the finite element mesh file (extension .fem or .FEM)
- the material properties file (extension .mtl or .MTL)
- the results file of the C-MOLD stress analysis (extension .ppt or .PPT)

These files should reside in the same directory. You must specify the name of one of these files. The names of the others are automatically derived from it.

Part of the data is imported directly into Mentat. The other data (most notably, the residual stresses, the elastic and thermal properties, and material orientations, which are all layer and element dependent) is written to a Marc post file that are viewed directly from the RESULTS menu. This post file data will be read at the start of a Marc job. This requires the user subroutine `cmold2marc.f` in the Mentat bin directory to be used.

There is one switch (or option) that can be controlled within Mentat. Here is the description of that switch.

**REPORT**
This allows you to create a summary report of the input. It lists the C-MOLD data sets that have been read from the four files and the data that is actually extracted from them.

The following data is extracted from the C-MOLD files:

Parameter file (.par or .PAR):

<table>
<thead>
<tr>
<th>Data Set</th>
<th>T-CODE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRMT</td>
<td>100</td>
<td>number of layers across the full-gap thickness</td>
</tr>
<tr>
<td></td>
<td>620</td>
<td></td>
</tr>
<tr>
<td>TITL</td>
<td>---</td>
<td>title of the model (currently not used)</td>
</tr>
</tbody>
</table>
Finite element mesh file (.fem or .FEM):

<table>
<thead>
<tr>
<th>Data Set</th>
<th>T-CODE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPRO</td>
<td>30100</td>
<td>thickness of triangular elements</td>
</tr>
<tr>
<td>NODE</td>
<td>---</td>
<td>coordinates of the nodes</td>
</tr>
<tr>
<td>QUAD</td>
<td>---</td>
<td>connectivity for quadrilateral element</td>
</tr>
<tr>
<td>TITL</td>
<td>---</td>
<td>title of the model (currently not used)</td>
</tr>
<tr>
<td>TRI</td>
<td>---</td>
<td>connectivity for triangular element</td>
</tr>
</tbody>
</table>

Material properties file (.mtl or .MTL):

<table>
<thead>
<tr>
<th>Data Set</th>
<th>T-CODE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTRL</td>
<td>1600</td>
<td>isotropic material properties</td>
</tr>
<tr>
<td></td>
<td>1602</td>
<td>orthotropic material properties</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>isotropic thermal expansion coefficient</td>
</tr>
<tr>
<td></td>
<td>1702</td>
<td>orthotropic thermal expansion coefficients</td>
</tr>
<tr>
<td>TITL</td>
<td>---</td>
<td>title of the model (currently not used)</td>
</tr>
</tbody>
</table>

Results file (.ppt or .PPT):

<table>
<thead>
<tr>
<th>Data Set</th>
<th>T-CODE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELDT</td>
<td>---</td>
<td>layer-based residual stresses and material properties for fiber-filled analyses</td>
</tr>
<tr>
<td>TITL</td>
<td>---</td>
<td>title of the model (currently not used)</td>
</tr>
<tr>
<td>TSDT</td>
<td>---</td>
<td>layer-based residual stresses for unfilled analyses; material properties are taken from Material properties file</td>
</tr>
</tbody>
</table>

Keyboard Command Sequence:

import cmold <C-MOLD file name>

Notes: 1) There are more command line options. In order to use all options, you have to run the C-MOLD translator in standalone mode.

2) The current version of the interface supports C-MOLD versions 98.7 to 99.1.
Command: **import dxf**  

Description: This command imports an ASCII AutoCAD DXF file. You must specify the name of the file. There are four switches (or options) which you can control within Mentat. Here are the descriptions of the switches.

- **COLOR(S)**  
  This allows a particular color to be translated into Mentat. By default, ALL colors are translated.

- **LEVEL(S)**  
  This allows a particular level to be translated into Mentat. By default, ALL levels are translated.

- **TOLERANCE**  
  This allows user to specify a tolerance for the input model. By default, tolerance is obtained from the input file.

- **REPORT**  
  This allows user to create a summary report of the input file. Here are some of the goodies included in the report: DXF version number, DXF entity summary, warnings messages regarding bad entities in the DXF file, Mentat entity summary, etc.

**Keyboard Command Sequence:**

```
import dxf <AutoCAD file name>
```

**Notes:**  
1) There are more command line options. In order to use all options, you will need to run the DXF translator in the standalone mode.  
2) AutoCAD r13c4’s DXF is supported.

Other Buttons with the same description:

- COLORS(S)  
- DXF  
- LAYER(S)  
- NO REPORT  
- TOLERANCE
Command: `import ideas <filename>`

Description: This command imports information from Ideas-produced universal file. You must specify the name of the file. Messages generated during the import operation provide information about which data have been imported and about any errors that occur.

The element types (shapes) currently supported are:

Dataset 780

<table>
<thead>
<tr>
<th>mentat element class</th>
<th>ideas desc id</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE2</td>
<td>11 21 22 23 122 171</td>
</tr>
<tr>
<td>LINE3</td>
<td>24 172</td>
</tr>
<tr>
<td>QUAD4</td>
<td>44 54 64 71 84 94 151 152</td>
</tr>
<tr>
<td>QUAD8</td>
<td>45 55 75 82 85 95</td>
</tr>
<tr>
<td>TRIA3</td>
<td>41 42 51 61 74 81 91</td>
</tr>
<tr>
<td>TRIA6</td>
<td>52 72 92</td>
</tr>
<tr>
<td>TETRA4</td>
<td>111</td>
</tr>
<tr>
<td>TETRA10</td>
<td>118</td>
</tr>
<tr>
<td>PENTA6</td>
<td>101 112</td>
</tr>
<tr>
<td>PENTA15</td>
<td>113</td>
</tr>
<tr>
<td>HEX8</td>
<td>115</td>
</tr>
<tr>
<td>HEX20</td>
<td>116</td>
</tr>
</tbody>
</table>

The apply types currently supported are:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>restraint type</th>
<th>1</th>
<th>nodal displacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>755</td>
<td>type</td>
<td>2</td>
<td>nodal temperature</td>
</tr>
<tr>
<td>782</td>
<td>load type</td>
<td>1</td>
<td>nodal force</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>element face pressure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>element face heat flux</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>element edge pressure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>element edge heat flux</td>
</tr>
</tbody>
</table>
The material types currently supported are:

<table>
<thead>
<tr>
<th>Dataset 773</th>
<th>material type</th>
<th>1 isotropic indices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 modulus of elasticity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 poisson’s ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 mass density</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 coefficient thermal expansion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 thermal conductivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 specific heat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 yield stress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18 emissivity</td>
</tr>
<tr>
<td>Dataset 788 or 772</td>
<td>physical prop.</td>
<td></td>
</tr>
<tr>
<td>only thickness</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Command: **import iges**

**Description:** This command imports an ASCII IGES geometry file. You must specify the name of the file. There are six switches (or options) which you can control within Mentat. Here are the descriptions of the switches.

**VALIDATE**
This turns on the validation of IGES entities. All explicitly defined semantics in the IGES specification are checked in addition to many implied restrictions. In most cases, a fixup can be performed which will correct invalid entities so that they may be processed. By default, validation is not done. This assumes the input file is good, and it speeds up the translation time.

**REAL SP CRV**
This allows the real space curves (or model space curves) from the IGES files to be included in the Mentat model. By default, ONLY parameter curves are translated into Mentat.

**COLOR(S)**
This allows a particular color to be translated into Mentat. By default, ALL colors are translated.

**LEVEL(S)**
This allows a particular level to be translated into Mentat. By default, ALL levels are translated.

**TOLERANCE**
This allows user to specify a tolerance for the input model. By default, tolerance is obtained from the input file.

**REPORT**
This allows user to create a summary report of the input file. Here are some of the goodies included in the report: IGES version number, IGES entity summary, validation result (if validation is turn on), Mentat entity summary, etc.

**Keyboard Command Sequence:**

```
import iges <IGES file name>
```

**Notes:**
1) There are more command line options. In order to use all options, you will need to run the IGES translator in the standalone mode.
2) IGES 5.2 and earlier version are supported.
Other Buttons with the same description:

- COLOR(S)
- IGES
- LEVEL(S)
- NO REPORT
- REAL SP CRV
- TOLERANCE
- VALIDATE

Command: **import intergraph**

**Description:** This command imports an INTERGRAPH neutral file. You must specify the name of the file. Messages generated during the import operation provide information about which data have been imported and about any errors that occur.

The element types (shapes) currently imported are:

- Line elements: LINE2, LINE3
- Tri elements: TRIA3, TRIA6
- Quad elements: QUAD4, QUAD8
- Tetra elements: TETRA4
- Penta elements: PENTA6, PENTA15
- Hex elements: HEX8, HEX20

Keep in mind that any loads or boundary conditions you create in INTERGRAPH must be created so that each set of conditions pertains to the same loads or other properties. For example, a set that contains forces applied to several nodes must use the SAME forces and components for ALL the nodes in the set. Also, you cannot build sets that combine loads and displacements or other types of data.

**Keyboard Command Sequence:**

```plaintext
import intergraph <INGERGRAPH neutral file name>
```
**MARC INPUT**

**Command:** `import marc_read`  
**Description:** This command reads a Marc input file into Mentat.  

**Note:** On the console where the program was started, there should be some messages regarding the import process. In general, “exit number” 3025 means the .dat file has been imported successfully.

**Keyboard Command Sequence:**

```
import marc_read <MARC file name>
```

---

**NASTRAN**

**Command:** `import nastran`  
**Description:** This command reads a NASTRAN neutral file. You must specify the NASTRAN file name.

**Keyboard Command Sequence:**

```
import nastran <NASTRAN file name>
```

---

**PATRAN**

**Command:** `import patran`  
**Description:** This command imports a Patran neutral file. You must specify the name of the file. Messages generated during the import operation provide information about which data has been imported and about any errors that occur.

**Packet Type 01: Node Data**  
This translator doesn’t read Coordinate Frame CID.

**Packet Type 02: Element Data**  
The element types (shapes) currently imported are:  
Line elements: LINE2, LINE3
Tri elements: TRIA3, TRIA6
Quad elements: QUAD4, QUAD8
Tetra elements: TETRA4
Penta elements: PENTA6, PENTA15
Hex elements: HEX8, HEX20

Packet Type 03: Material properties

ISOTROPIC
    youngs_modulus
    poissons_ratio
    thermal_expansion
    yield_stress
    specific_heat
    mass_density
    emissivity
    conductivity

ORTHOTROPIC
    youngs_modulus11
    youngs_modulus22
    youngs_modulus33
    poissons_ratio12
    poissons_ratio23
    poissons_ratio31
    shear_modulus12
    shear_modulus23
    shear_modulus31
    mass_density
    thermal_exp11
    thermal_exp22
    thermal_exp33
    conductivity11
    conductivity22
    conductivity23
    specific_heat

Packet Type 04: Element Properties

Material ID is only available.

Packet Type 06: Distributed Loads

This is unsupported in case GFLAG=1.
Packet Type 07: Node Forces
   Coordinate frame ID CID is unsupported.

Packet Type 08: Node Displacements
   Coordinate frame ID CID is unsupported.

Packet Type 10: Node Temperatures

Packet Type 15: Nodal Heat Source

Packet Type 16: Distributed Heat Source
   This is unsupported in case NFLAG=1.

Packet Type 17: Convection Coefficients
   This is unsupported in case NFLAG=1.

Keyboard Command Sequence:
   import patran (PATRAN neutral file name)

Command:  import stl
Description:  This command imports an ASCII or BINARY Stereolithography Interface Format (STL) file into Mentat.
   Current version is October 1989.

Keyboard Command Sequence:
   import stl (STL file name)
**Command:** import vda

**Description:** This command imports an ASCII VDAFS file. You must specify the name of the file. There are two switches (or options) which one can control within Mentat. Here are the descriptions of the switches.

**TOLERANCE**
This allows user to specify a tolerance for the input model. By default, tolerance is obtained from the input file.

**REPORT**
This allows user to create a summary report of the input file. Here are some of the goodies included in the report: VDA version number, VDA entity summary, validation result (if validation is turn on), Mentat entity summary, etc.

**Keyboard Command Sequence:**

```plaintext
import vda <VDAFS file name>
```

**Notes:**
1) There are more command line options. In order to use all options, you will need to run the VDA translator in the standalone mode.
2) VDAFS 2.0 and earlier version are supported.

Other Buttons with the same description:

- NO REPORT
- REPORT FILE
- TOLERANCE
INCREMENT HUE

Command: increment_hue

Description: This command interpolates the color hues used to show variations in increments for contour plotting and other graphical incremental indicators. You must specify the starting and ending color indices and the hue increment.

The two color indices will be used as the starting and ending values of a color interpolation that uses the specified hue increment.

Keyboard Command Sequence:

increment_hue <starting index> <ending index> <hue increment>

INITIAL CONDITIONS

Menu: INITIAL CONDITIONS

Description: The INITIAL CONDITIONS menu contains commands that apply initial conditions to the mesh. Initial conditions may be applied for mechanical and heat transfer analysis types. The types of boundary conditions that can be applied depend on the type of analysis.

INITPL

Subroutine: INITPL

Description: The user subroutine INITPL is called in a loop over all the elements in the mesh when the INITIAL PLASTIC option appears in the model definition series with a 2 in column 10 of the second data block series of that option. It is often necessary to enter the amount of previously accumulated plastic strain. This initial value is only used in the work (strain) hardening calculation.
INITSV

Subroutine: INITSV

Description: The user subroutine INITSV allows you to define initial values of state variables. This routine will be called in a loop over all the elements in the mesh when the INITIAL STATE option appears in the model definition series with a 2 in column 10 of the second data block series of that option.

NAME

Command: insert_name

Description: This command sets or changes the name of the current insert.

Keyboard Command Sequence:

    insert_name <insert name>

INTCRD

Subroutine: INTCRD

Description: The user subroutine INTCRD makes the integration point coordinates for the stiffness matrix available at each increment. You may save them in any form convenient for your postprocessing.
HUE-LIGHTNESS-SATURATION CUBE

Command:  interpolate_hls
Description: This command interpolates color values using hue, level, and saturation. The gray values and lighting properties for each color index are also interpolated. You must specify the first and last color indices between which colors are to be interpolated.

Keyboard Command Sequence:

    interpolate_hls <first color index> <last color index>

RED-GREEN-BLUE CUBE

Command:  interpolate_rgb
Description: This command interpolates color values using red, green, and blue values. The gray values and lighting properties for each color index are also interpolated. You must specify the first and last color indices between which colors are to be interpolated.

Keyboard Command Sequence:

    interpolate_rgb <first color index> <last color index>

INTERSECT

Menu:  INTERSECT
Description: This menu contains commands that intersect curves and surfaces.
**CURVE/CURVE**

**Command:** `intersect_curves`

**Description:** This command finds all the intersections of the given list of curves. This command also breaks the curves at the intersection points.

Notice that trim curves can be intersected only with other trim curves of the same parent surface.

**Keyboard Command Sequence:**

```
intersect_curves <list of curves>
```

---

**CURVE/SURFACE**

**Command:** `intersect_curves_surface`

**Description:** This command finds all the intersections of the given list of curves against a surface. This command also breaks the curves at the intersection points.

**Keyboard Command Sequence:**

```
intersect_curves_surface <surface, list of curves>
```
**SURFACE/SURFACE**

Command: **intersect_surface**

Description: This command generates the curve(s) of intersection of two surfaces. The surfaces are subdivided till they are flat enough to approximate the intersections by plane plane intersection. The resulting line segments are then stringed to get the intersection curves. For stringing line segments, the algorithm uses the `sweep_tolerance` for checking equality of 3-D points. If this tolerance is very small, the algorithm will return disconnected curve segments. This tolerance can be set using `set_sweep_tolerance` command.

For each intersection, in addition to the real space curve, the two parameter space curves are also returned as trim curves. The user can switch on or off any of these three curves. Notice that, if the trim curves from a closed loop in the parametric space, it affects the topology of the surface.

**Keyboard Command Sequence:**

```
intersect_surface <surface 1> <surface 2>
```
CURVES

Command: `invisible_curves`
Description: This command makes the specified curves invisible.
Keyboard Command Sequence:
```
invisible_curves <curve list> #
```

ELEMENTS

Command: `invisible_elements`
Description: This command makes the specified elements invisible.
Keyboard Command Sequence:
```
invisible_elements <element list> #
```

MAKE INVISIBLE

Command: `invisible_selected`
Description: This command causes the currently selected entities to be made invisible. Selected entities can be made visible with the `visible_selected` command.
Keyboard Command Sequence:
```
invisible_selected
```
**SOLIDS**

**Command:** `invisible_solids`

**Description:** This command makes the specified solids invisible. Solids can be made visible with the `visible_solids` command.

**Keyboard Command Sequence:**

```
invisible_solids <solid list> #
```

---

**SURFACES**

**Command:** `invisible_surfaces`

**Description:** This command makes the specified surfaces invisible.

**Keyboard Command Sequence:**

```
invisible_surfaces <surface list> #
```
Mentat Help Commands in J

**REM**

Command:  **job_activate_elements**

Description: This command activates elements that have been deactivated with the **job_deactivate_elements** command.

Keyboard Command Sequence:

```
job_activate_elements <element list> #
```

**MAX. # ELEMENTS**

Command:  **job_param max_el**

**job_param max_nod**

Description: These commands set upper bounds to the number of elements and nodes that are allowed to be created during remeshing or adaptivity (element splitting).

MAX. # ELEMENTS  upper bound for the number of elements
**MAX. # NODES**

upper bound for the number of nodes

The values are used for allocating memory for the analysis. The options STOP AT LIMIT and ALLOW MEMORY GROWTH are used to control if the analysis should stop when the maximum number is exceeded or if more memory should be allocated.

**Reference:** *MSC.Marc Volume C: Program Input*, Chapter 2: parameter ADAPTIVE

**Keyboard Command Sequence:**

```
job_param max_el <value>
```

Another Button with the same description:

**MAX. # NODES**

**MAX. # CONTACT SEGMENTS/BODY**

**Command:**

```
job_param max_cont_seg
job_param max_cont_nod
```

**Description:** These commands set upper bounds to the number of segments and nodes on the boundary of any contact body that are allowed to be created during remeshing or adaptivity (element splitting).

**MAX. # CONTACT SEGMENTS / BODY**

upper bound for the number of segments

**MAX. # CONTACT NODES / BODY**

upper bound for the number of nodes

The values are used for allocating memory for the analysis. Larger values need to be specified for the remeshed bodies. Note that if zero values are specified, the memory allocation is based upon the initial mesh. If a mesh with more elements is to be created, larger values must be specified.

**Reference:** *MSC.Marc Volume C: Program Input*, Chapter 3: Model Definition CONTACT, 2nd data block, 2nd and 3rd parameters

**Keyboard Command Sequence:**

```
job_param max_cont_seg <value>
job_param max_cont_nod <value>
```
**Command:**  `job_option rmsh_analysis`  

**Description:** This command determines the way the mesher is started during remeshing.

**STOP AND RESTART**
The main analysis is stopped and the mesher is run. When the mesher is done the main analysis restarts, reads in the mesh information file from the mesher and continues.

**WAIT**
The main analysis just halts while the mesher runs. On Windows, a system call is issued from the main analysis to run the mesher. On UNIX systems, a separate script is run that controls the running of the mesher.

The advantage with **STOP AND RESTART** is that the memory used by the main analysis is released so that the mesher can use more memory if needed. With the **WAIT** option, the main analysis and the mesher occupy memory at the same time. A disadvantage with **STOP AND RESTART** is that a potentially large restart file is written very incrementally.

**Keyboard Command Sequence:**

```
job_option
job_option rmsh_analysis:<wait/stop_rstrt>
```

Another Button with the same description:

**WAIT**
ALLOW MEMORY GROWTH

Command: job_option rmsh_limit

Description: This command defines whether the analysis should stop if the maximum number of elements or nodes is reached during remeshing.

STOP AT LIMIT stop the analysis if the maximum number of elements or nodes is reached
ALLOW MEMORY GROWTH allocate more memory if possible to be able to continue the analysis

Keyboard Command Sequence:

job_option rmsh_limit:<stop/grow>

Another Button with the same description:

STOP AT LIMIT

PROCEED (PREVIOUS MESH)

Command: job_option adap_proceed

Description: This command defines whether the analysis should stop if the maximum number of elements or nodes is reached during adaptivity (element splitting).

STOP AT LIMIT stop the analysis if the maximum number of elements or nodes is reached
PROCEED (PREVIOUS MESH) proceed the analysis using the previous mesh

Keyboard Command Sequence:

job_option adap_proceed:<on/off>

Another Button with the same description:

STOP AT LIMIT
**ANALYSIS OPTIONS**

**Menu:** ANALYSIS OPTIONS

**Description:** This popup menu contains commands for setting analysis options for the current job.

---

**INCREMENTS**

**Command:** job_buckle_increments

**Description:** This command specifies which increments a buckle analysis will be performed.

**Keyboard Command Sequence:**

```
job_buckle_increments <increment numbers>
```

---

**CLEAR**

**Command:** clear_job_buckle_increments

**Description:** This command clears the increments specified for buckle analysis.

**Keyboard Command Sequence:**

```
clear_job_buckle_increments
```

---

**OFF**

**Command:** job_option buckle_inc

**Description:** This command specifies if a buckling analysis with the BUCKLE INCREMENT option is to be performed.

This option allows you to specify at which increments a buckling analysis is performed. It is used as either a replacement to the BUCKLE loadcase option or in conjunction with it. This option allows you to extract modes within a loadcase with multiple increments. The increments are specified with the job_buckle_increments command under the BUCKLE
The choice of buckle solution method is made with the buttons above. The specification of the number of modes to extract is done under JOBS->MECHANICAL->JOB PARAMETERS. The eigenvectors, stresses, and the reactions are automatically set to be printed out to the post file.

The option in Marc to perform a perturbation of the geometry by the buckling mode is not yet supported by Mentat and must be entered into the input file by the user.

Reference:  
MSC.Marc Volume C: Program Input, Chapter 3: Model Definition CONTACT  
MSC.Marc Volume A: Theory and User Information, Chapter 5

Keyboard Command Sequence:  
job_option buckle_inc:<on/off>

Another Button with the same description:  
ON
**INVERSE POWER SWEEP**

**Command:**  `job_option buckle_method`

**Description:** This command specifies the method to be used to extract the buckling modes.

The inverse power sweep method is the default. The Lanczos method is usually more efficient, in particular for large models and when several modes are to be extracted. It also handles multiple eigenvalues better.

The specification of the number of buckling modes to extract is done under JOBS->MECHANICAL->JOB PARAMETERS. The eigenvectors, stresses and the reactions are automatically set to be printed out to the post file.

The option in Marc to perform a perturbation of the geometry by the buckling mode is not yet supported by Mentat and must be entered into the input file by the user.

**Reference:**  *MSC.Marc Volume A: Theory and User Information*, Chapter 5

**Keyboard Command Sequence:**

```
job_option buckle_method:<power/lanczos>
```

Another Button with the same description:

LANCZOS

---

**# BUCKLE MODES**

**Command:**  `job_param buckle_nbmodes`

`job_param buckle_npbmodes`

**Description:** These commands specify the number of buckling modes to be estimated.

The first command, `job_param buckle_nbmodes`, # BUCKLE MODES is used to specify the number of buckling modes to extract.

The second command, `job_param buckle_npbmodes`, # POS. BUCKLE MODES refers to the number of buckling modes with positive eigenvalues that is to be extracted.

In many buckling problems, collapse modes corresponding to loads of opposite magnitude to those of interest exist. With this option, you can
ensure that the modes of interest are obtained. The program stops the modal search when all these modes have been formed, or when the total number of modes requested have been formed, whichever occurs first. Thus, make sure to use a larger total number of modes than positive modes. If this parameter is set to zero, the number of buckling modes specified by # BUCKLE MODES is used, regardless of sign.

Reference: MSC.Marc Volume C: Program Input, Chapter 2: parameter BUCKLE

Keyboard Command Sequence:

```
job_param buckle_nbmodes
  <total number of modes to extract>
job_param buckle_npbmodes
  <number of modes with positive eigenvalues to extract>
```

Another Button with the same description:

```
# POS. BUCKLE MODES
```

---

Command: `job_param bi_maxit`

```
job_param bi_tol
```

Description: These commands specify parameters for the inverse power sweep method if it is used in the current job.

The power sweep terminates when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance.

If the number of iteration required is larger than the maximum specified, the analysis stops. The default number is 40.

Reference: MSC.Marc Volume C: Program Input, Chapter 4

Keyboard Command Sequence:

```
job_param bi_maxit
job_param bi_tol
```

Another Button with the same description:

```
TOLERANCE
```
Command:  job_class

Description:  This command sets the class of the current job to the specified class.

The valid job classes are as follows:

- mechanical
- heat
- coupled
- joule Heating
- joule mechanical
- electrostatic
- piezo Electric
- acoustic
- acoustic solid
- magnetostatic
- electromagnetic
- fluid
- fluid thermal
- fluid solid
- fluid thermal solid
- bearing

Keyboard Command Sequence:

- job_class

Other Buttons with the same description:

- ACOUSTIC
- ACOUSTIC-SOLID
- BEARING
- COUPLED
- ELECTROMAGNETIC
- ELECTROSTATIC
- FLUID
- FLUID-SOLID
- FLUID-THERMAL
- FLUID-THERMAL-SOLID
- HEAT TRANSFER
- JOULE HEATING
- JOULE MECHANICAL
- MAGNETOSTATIC
Command:  
job_clear_host_file

Description:  Clears the current job’s host file.

Keyboard Command Sequence:

job_clear_host_file

Command:  
job_clear_scratch_directory

Description:  Clears the current job’s scratch directory.

Keyboard Command Sequence:

job_clear_scratch_directory

Command:  
job_clear_usersub_file

Description:  Clears the current job’s user subroutine file.

Keyboard Command Sequence:

job_clear_usersub_file

Menu:  CONTACT PARAMETERS

Description:  This popup menu contains commands for setting contact job parameters for the current job.
DISTANCE TOLERANCE BIAS

Command: `job_param disttolbias`

Description: Sets the contact bias factor for the current job.

This factor takes values between -1 and 1 and controls the position of the contact tolerance zone. By default, the bias factor is zero and the contact tolerance zone is symmetric around the contact entity. With a positive bias factor, the contact tolerance zone is moved into the body and thus providing a more accurate contact description.

Reference: *MSC.Marc Volume C: Program Input*, Chapter 3: Model Definition CONTACT

*MSC.Marc Volume A: Theory and User Information*, Chapter 8

Keyboard Command Sequence:

```
job_param disttolbias <distance tolerance bias>
```

ALLOWED

Command: `job_option chatter`

Description: This command is used to specify whether a node that comes into contact in the current increment is allowed to separate again during the current increment.

Setting this to `suppressed` prevents chattering.

Reference: *MSC.Marc Volume C: Program Input*, Chapter 3: Model Definition CONTACT

Keyboard Command Sequence:

```
job_option chatter:<allowed/suppressed>
```

Another Button with the same description:

SUPPRESSED
DISTANCE TOLERANCE

Command:  job_param disttol
Description:  Sets the contact distance tolerance for the current job.

This defines the size of the contact tolerance zone, i.e. the distance below which a node is considered touching a surface. The default value is 5% of the smallest element side in case of solid elements or 50% of the smallest beam or shell thickness. Note that the default values are based on all elements in the model.

Reference:  MSC.Marc Volume C: Program Input, Chapter 3: Model Definition CONTACT
MSC.Marc Volume A: Theory and User Information, Chapter 8

Keyboard Command Sequence:
    job_param disttol <distance tolerance>

DOUBLE-SIDED

Command:  job_option style:double
Description:  This command specifies that a double-sided contact treatment should be used.

A lower-numbered contact body is first checked for contact with a higher-numbered contact body and then the other way around.

Reference:  MSC.Marc Volume C: Program Input, Chapter 3: Model Definition CONTACT

Keyboard Command Sequence:
    job_option style:double
FORCE

**Command:**  
`job_option separation`

**Description:**  
This command specifies if the criterion used to decide whether or not a node in contact should separate is force-based or stress-based:

**Force-based separation:**  
Separation occurs when the tensile contact normal force on the node in contact becomes larger than the separation threshold force. This threshold force can be set by the user with the option `job_param sepforce` or by using a contact table. In the latter case, the separation threshold can be set per pair of contact bodies.

The separation threshold force is by default equal to the maximum residual force component.

**Stress-based separation:**  
Separation occurs when the tensile contact normal stress on the node in contact becomes larger than the separation threshold stress. With stress-based separation, the following is available:

- Define relative or absolute checking:
  
  If absolute testing is used, then the user can enter directly the separation threshold stress. If relative checking is used, then the user has to enter a factor and the separation threshold stress is defined by this factor times the maximum compressive contact normal stress. The option to set the method of checking is `job_option sep_strs_measure`.

- Define the method to derive the contact normal stress:
  
  The contact normal stress can be obtained by either dividing the nodal force by the corresponding equivalent area or by extrapolating and averaging the integration point stress values. The second method cannot be used for shell elements and MSC.Marc will automatically switch to the first if necessary. Defining the method is done with the option `job_option sep_strs_deriv`.

The threshold value to be entered by the user, either a factor or a stress, is done via the option `job_param sepstress` or `job_param rel_sepstress`, or via a contact table. In the latter case, the separation threshold can be set per pair of contact bodies.
By default, separation occurs if the tensile contact normal stress becomes larger than 10 percent of the maximum compressive contact normal stress.

Stress-based separation has the advantage over force-based separation that it eliminates the influence of the element size. For true quadratic contact, only stress-based separation can be used, where the nodal stresses are determined by extrapolating and averaging integration point stress values.

Reference:  
*MSC.Marc Volume C: Program Input*, Chapter 3: Model Definition CONTACT  
*MSC.Marc Volume A: Theory and User Information*, Chapter 8

Keyboard Command Sequence:

```
job_option separation:<force or stress>
```

Another Button with the same description:

```
STRESS
```

Command:  
```
job_option frictype
```

Description:  
This command is used to specify the type of friction that is used in the current job.

- **NONE**: friction is not to be taken into account.
- **STICK-SLIP**: a true stick-slip model which models the transition from stick to slip accurately. The model uses the coulomb model of friction. The behavior can be controlled by three parameters below.
- **SHEAR**: a cohesive model of friction in which the friction stress is based upon the coefficient of friction and the equivalent von Mises stress in the material.
- **COULOMB**: an adhesive model of friction in which the friction stress is based upon the coefficient of friction and the normal stress at the surface.
- **SHEAR FOR ROLLING**: enhanced shear model.
- **COULOMB FOR ROLLING**: enhanced coulomb model.
The main difference between the models "FOR ROLLING" and without rolling is that the "FOR ROLLING" models use an improved estimate of the friction condition at the first cycle of an increment. The "FOR ROLLING" models are therefore recommended over the ones without "rolling", in particular for cases where there is a large difference between the absolute sliding velocity and the relative sliding velocity between two contacting bodies.

The extension "FOR ROLLING" has historical significance in that the improvements to the model were done initially for ring rolling simulations. However, the enhanced model shows improved accuracy and convergence also for other types of applications, like sheet forming and elastomer analysis.

For all models except the stick-slip model a parameter "relative sliding velocity" must be specified. It is typically set to 1%-10% of the relative sliding velocity between contacting bodies. Note that it has the unit of length over time and that the default value of 1.0 is in general inappropriate.

For the coulomb and coulomb for rolling models there is the choice between having the friction based on nodal stress or nodal force. An exception is for shell elements where nodal force is always used. The stick-slip model always uses nodal force and the shear model nodal stress.


Keyboard Command Sequence:

```
job_option frictype:<none/coul_stick_slip/shear/
coulomb/shear_roll/coulomb_roll>
```

Other Buttons with the same description:

- COULOMB FOR ROLLING
- NONE
- SHEAR
- SHEAR FOR ROLLING
- STICK-SLIP

Command: job_option fricmeth
**Description:** This command is used to define if the friction model is based on nodal force or nodal stress.

See the help menu on the different friction types above for applicability.

**Reference:** *MSC.Marc Volume A: Program Input*, Chapter 8.

**Keyboard Command Sequence:**

```
job_option fricmeth:<nstress/nforce>
```

Another Button with the same description:

**NODAL STRESS**

---

**Command:** job_param vsliding

**Description:** This command is used to enter the relative sliding velocity that is used by all friction models except the stick-slip model.

This parameter controls the transition from stick to slip in these models. It is typically set to 1%-10% of the relative sliding velocity between contacting bodies.

Note that it has the unit of length over time and the default value of 1.0 is in general inappropriate. A value too small leads to convergence problems in the solution and a value too large gives inaccurate friction description.


**Keyboard Command Sequence:**

```
job_param vsliding <relative sliding velocity>
```

---

**Command:**

```
job_param stick_slip_trans
job_param fric_coeff_mult
job_param fric_force_tol
```
**Description:** These commands are used to define the properties of the stick-slip models. Usually, the default values are appropriate. See the manual reference for details.

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 8

**Keyboard Command Sequence:**

```
job_param stick_slip_trans <new value>
job_param fric_coeff_mult <new value>
job_param fric_force_tol <new value>
```

Other Buttons with the same description:

- FRICTION FORCE TOLERANCE
- SLIP TO STICK TRANS. REGION

---

**Command:** `job_option quadr_cont_seg:genuine`

**Description:** If the model contains a deformable or heat transfer rigid body with quadratic elements, then this command activates a boundary description based on the complete quadratic geometry. This implies that the mid-side nodes are fully taken into account: they can come into contact, and the patch they correspond to is described using a complete quadratic field.

**Keyboard Command Sequence:**

```
job_option quadr_cont_seg:genuine
```

---

**Menu:** INITIAL CONTACT

**Description:** This menu contains commands for activating initial contact conditions: contact table, contact areas, and exclude segments.

---

**Command:** `job_option quadr_cont_seg:linearized`
**Description:** If the model contains a deformable or heat transfer rigid body with quadratic elements, then this command activates a boundary description based on a linearized geometry. This implies that the mid-side nodes are not treated independently: they cannot come into contact, and the geometry of the patch they correspond to is described using the corner nodes only. In a contact stress analysis this may result in strange stress concentrations in the contact area.

**Keyboard Command Sequence:**

```
job_option quadr_cont_seg:linearized
```

**Command:**  
job_param maxsep

**Description:** This command is used to define the maximum number of separations that can occur during an increment.

Each time a node is found to separate, the increment is recycled. This parameter limits the number of times that will occur. The default number is 9,999.

**Reference:**  
MSC.Marc Volume C: Program Input, Chapter 3: Model Definition CONTACT

**Keyboard Command Sequence:**

```
job_param maxsep <max. number of separations>
```

**Command:**  
job_param rel_sepstress

**Description:** This command is used to specify the relative separation stress to be used instead of the default value.

Also see help: job_option separation.

**Keyboard Command Sequence:**

...
**Command:** `job_option separation`  
**Description:** This command specifies if the criterion used to decide whether or not a node in contact should separate is force-based or stress-based:

- **Force-based separation:**  
  Separation occurs when the tensile contact normal force on the node in contact becomes larger than the separation threshold force. This threshold force can be set by the user with the option `job_param sepforce` or by using a contact table. In the latter case, the separation threshold can be set per pair of contact bodies.

  By default, the separation threshold force is equal to the maximum residual force component.

- **Stress-based separation:**  
  Separation occurs when the tensile contact normal stress on the node in contact becomes larger than the separation threshold stress. With stress-based separation, the following is available:

  - Define relative or absolute checking:
    
    If absolute testing is used, then the user can enter directly the separation threshold stress. If relative checking is used, then the user has to enter a factor and the separation threshold stress is defined by this factor times the maximum compressive contact normal stress. The option to set the method of checking is `job_option sep_strs_measure`. 

```plaintext
job_param rel_sepstress
<relative separation stress>
```
– Define the method to derive the contact normal stress:

The contact normal stress can be obtained by either dividing the nodal force by the corresponding equivalent area or by extrapolating and averaging the integration point stress values. The second method cannot be used for shell elements and Marc will automatically switch to the first if necessary. Defining the method is done with the option job_option sep_strs_deriv.

The threshold value to be entered by the user, either a factor or a stress, is done via the option job_param sepstress or job_param rel_sepstress, or via a contact table. In the latter case, the separation threshold can be set per pair of contact bodies.

By default, separation occurs if the tensile contact normal stress becomes larger than 10 percent of the maximum compressive contact normal stress.

Stress-based separation has the advantage over force-based separation that it eliminates the influence of the element size. For true quadratic contact, only stress-based separation can be used, where the nodal stresses are determined by extrapolating and averaging integration point stress values.

Reference:  
MSC.Marc Volume C: Program Input, Chapter 3: Model Definition Option, CONTACT.  
MSC.Marc Volume A: Theory and User Information, Chapter 8.

Keyboard Command Sequence:  
job_option separation:<force/stress>

Other Buttons with the same description:  
STRESS

Command:  
job_param sepforce

Description:  
This command is used to specify the separation force to be used instead of the default value.

Also see help: job_option separation.

Keyboard Command Sequence:
job_param sepforce <separation force>
Command: job_option sepinc

Description: This command is used to specify when separation should actually occur. When a separation is detected, the bodies are separated either in the current increment or in the next. The latter option may lead to decreased computational cost but lower the accuracy of the solution.

Reference: MSC.Marc Volume C: Program Input, Chapter 3: Model Definition, CONTACT

Keyboard Command Sequence:

```
job_option sepinc:<current/next>
```

Another Button with the same description:

NEXT

Command: job_param sepstress
job_param rel_sepstress

Description: This command is used to specify the separation stress (sepstress) or relative separation stress (rel_sepstress) to be used instead of the default value. The type that is used is dependent on whether the job_option sepstrs_measure is specified as absolute or relative (the button ABSOLUTE/RELATIVE located adjacent to this button.

Also see help: job_option separation and job_option sep_strs_measure.

Reference: MSC.Marc Volume C: Program Input, Chapter 3: Model Definition CONTACT
MSC.Marc Volume A: Theory and User Information, Chapter 8.

Keyboard Command Sequence:

```
job_param sepstress <separation stress>
job_param rel_sepstress
  <relative separation stress>
```
**FORCE/AREA**

Command:  `job_option sep_strs_deriv`  

Description: For stress-based separation, this command is used to specify the way in which the contact normal stress is derived; this can be either done by dividing the contact normal force by the equivalent area or by extrapolating and averaging integration point stress values.

Also see help: `job_option separation`.

Keyboard Command Sequence:

```
job_option sep_strs_deriv:<force_area/extrapolation>
```

Another Button with the same description:

EXTRAPOLATION

---

**ABSOLUTE**

Command:  `job_option sep_strs_measure`  

Description: For stress-based separation, this command is used to specify that separation has to be based on either absolute or relative checking.

Also see help: `job_option separation`.

Keyboard Command Sequence:

```
job_option sep_strs_measure:<absolute/relative>
```

Another Button with the same description:

RELATIVE
**Command:**  job_option checkshell

**Description:**  This command is used to specify the contact treatment for shell elements.

- **CHECK TOP & BOTTOM SURFACE**  contact is detected with either side of the surface and the shell thickness is always taken into account.

- **CHECK TOP SURFACE**  contact is only detected with the top surface of the shell. The top surface is in the positive direction of the shell normal.

- **CHECK BOTTOM SURFACE**  contact is only detected with the bottom surface of the shell. The bottom surface is in the negative direction of the shell normal.

The choice of taking the shell thickness into account for the latter two options is made with the command job_option ignore: on/off (menu IGNORE THICKNESS).

The orientation of the elements is visualized with the command identify_backfaces under:

CONTACT->CONTACT BODIES->ID BACKFACES
(turn off with identify_none).

By default, the top surface is plotted in blue. Shell normals of connected elements are aligned with the command align_shells under:

MESH GENERATION->CHECK->ALIGN SHELLS.

**Reference:**  *MSC.Marc Volume A, Theory and User Information*, Chapter 8.

**Keyboard Command Sequence:**

job_option checkshell:<top/bottom/both>

**Other Buttons**  with the same description:

CHECK TOP SURFACE
CHECK BOTTOM SURFACE
Command:  job_option style:single
Description: This command specifies that a single-sided contact treatment should be used.

A lower-numbered contact body is only checked for contact with a higher-numbered contact body. Compared with the double-sided version, this leads to decreased cost for cases where one knows that a one-way treatment is appropriate.

Reference:  MSC.Marc Volume C: Program Input, Chapter 3: Model Definition, CONTACT

Keyboard Command Sequence:

job_option style:single

END OF INCREMENT

Command:  job_option split
Description: This command defines the method used to avoid that nodes might penetrate into a contact body. The possible options are:

END OF INCREMENT:
At the end of each increment, a check is performed on nodes not being in contact at the beginning of the increment. If such a node has penetrated into a contact body beyond the contact tolerance zone, the increment is split in two parts. The first part is a linear scaling of the solution found so far, where the scaling factor is based on the node with the largest relative amount of penetration. The second part will finish the increment with one or more additional nodes in contact. If necessary, this process will be repeated.

PER ITERATION:
After the incremental solution in the Newton-Raphson iteration process has been found, a check is performed if this iterative displacement vector will cause nodes to come into contact or to penetrate beyond the contact tolerance. If the latter is the case, the iterative displacement vector will be scaled such that no penetration will occur. Although this checking per iteration is more expensive than checking per increment, it is in general more reliable and accurate.
NONE:
The penetration of a node is ignored in the current increment. Although this leads to decreased computational costs, it also leads to decreased accuracy and should normally not be used.

Reference:  *MSC.Marc Volume A: Theory and User Information*, Chapter 8

Keyboard Command Sequence:

```
job_option split:<allowed/iterative/suppressed>
```

Other Buttons with the same description:

- OFF
- PER ITERATION

**Command:**  `job_contact_table`

**Description:** This command is used to specify the contact table to be used initially for this job.

Other contact tables can be activated in each loadcase.

**Keyboard Command Sequence:**

```
job_contact_table <contact table name>
```

**Command:**  `job_option ignore`

**Description:** This command is used to specify if thickness is to be taken into account for shell elements.

It is only applicable when contact is detected with the top or bottom surface.

Reference:  *MSC.Marc Volume A: Theory and User Information*, Chapter 8

**Keyboard Command Sequence:**

```
job_option ignore:<on/off>
```
CONVECTIVE TERMS

Command:  job_option convective

Description: This command is used to activate or deactivate the option of using a formulation with convection in a heat transfer analysis.

The general convection-diffusion equation is used in this case and its solver using the SUPG method.

This formulation requires that the velocity field is known, either by prescribing it or by using a thermomechanical coupled analysis with the rigid plastic flow option.


Keyboard Command Sequence:

```
job_option sep_strs_deriv:<force_area/extrapolation>
```
CYCLIC SYMMETRY

Menu: CYCLIC SYMMETRY
Description: This popup menu automatically contains commands for taking cyclic symmetry conditions into account in the current job.

# REPETITIONS

Menu: CYCLIC SYMMETRY PARAMETERS
Description: This menu contains commands for automatically taking cyclic symmetry conditions into account in the current job. These conditions allow rotationally periodic structures to be analyzed by modeling only one segment of the complete structure. The user has to specify the axis around which the structure is rotationally periodic and the number of times the segment fits in the complete structure (i.e. the number of periods in the full 360 degree range). MSC.Marc will automatically detect the nodes that lie on the boundaries of the segment and create the proper constraint equations for these nodes. The meshes at these boundaries do not have to match.

If contact is used in the model, the elements along the cyclic symmetry boundaries must belong to the same contact body.

CYCLIC SYMMETRY
When switched on, cyclic symmetry conditions are taken into account in the current job.
AXIS OF ROTATION
The axis of rotation is defined by specifying its DIRECTION and a POINT on the axis.

# REPETITIONS
The number of times the segment fits in the complete structure.

AUTOMATIC SUPPRESSION OF ROTATIONAL RIGID BODY MOTION
When switched on, Marc will automatically suppress the rotational rigid body motion of the segment around the axis of rotation. For two-dimensional analyses, this is sufficient to eliminate all rigid body motions. For three-dimensional analyses, boundary conditions still have to be created to suppress the rigid body motion in the direction of the axis of rotation.
Keyboard Command Sequence:

```
job_option cyc_symm_rot: on|off
job_param cycs_rot_axis_dir_x <x_component_of_direction>
job_param cycs_rot_axis_dir_y <y_component_of_direction>
job_param cycs_rot_axis_dir_z <z_component_of_direction>

job_param cycs_rot_axis_point_x <x_component_of_point>
job_param cycs_rot_axis_point_y <y_component_of_point>
job_param cycs_rot_axis_point_z <z_component_of_point>

job_param cycs_rot_nrepetitions <number_of_repetitions>

job_param cycs_rot_rigb_supp: automatic|off
```

Reference:  
- *MSC.Marc Volume A: Theory and User Information*, Chapter 9: Section Kinematic Constraints  
- *MSC.Marc Volume C: Program Input*, Chapter 3: Model Definition Option  

CYCLIC SYMMETRY  

Other Buttons with the same description:  
- AUTOMATIC SUPPRESSION  
- CYCLIC SYMMETRY  
- X  
- Y  
- Z

---

**Command:**  
```
job_damping
```

**Description:**  
This command is used to specify the damping coefficients for each mode in a dynamic analysis.

**Keyboard Command Sequence:**

```
job_damping <mode number> <damping coefficient>
```

Other Buttons with the same description:

- MODE 2
  - :
  - :
  - MODE 30
Command:  `job_dcom_hostname <name>`  

Description:  This command specified the DCOM server to be used for Marc. It is only available for Windows.

If the DCOM option is on, then this command is enabled to allow a server hostname to be specified to run the job remotely. It is independent of the parallel (USE DDM) option, and does not support multiple domains.

To use this option, you MUST ensure that:

- The machine that is specified as the host must be enabled as a Marc DCOM server.
- Your datafiles are on shared directories.
- Your username/password is the same as for the host.

Keyboard Command Sequence:

```
job_dcom_hostname <name>
```

---

Command:  `job_deactivate_elements`  

Description:  This command deactivates elements for the current job. Deactivated elements do not participate in the analysis.

Keyboard Command Sequence:

```
job_deactivate_elements <element list> 
```

#
Command: job_option decomposition
Description: This command choose type of domain decomposition algorithm.

- **metis_best**: performs metis element based and node based decomposition and picks the better solution.
- **metis_element_based**: performs metis element based decomposition.
- **metis_node_based**: performs metis node based decomposition.
- **geometric**: performs decomposition based on coordinates (previously decompose) algorithm tries to anneal domains.
- **simple**: performs decomposition based on coordinates of element centroids (previously generate)

**Keyboard Command Sequence:**

```
job_option decomposition:<metis_best/<metis_element_based/<metis_node_based/<geometric/<simple>
```

Other Buttons with the same description:

- METIS ELEMENT BASED
- METIS NODE BASED
- GEOMETRIC
- SIMPLE
**DESIGN**

**Command:** N/A

**Description:** This button brings up the popup menu for:

a) setting either design SENSITIVITY or design OPTIMIZATION as the requested procedure.

b) defining the DESIGN OBJECTIVE

c) selecting the design variables (from the set already defined) which apply to this particular job and

d) determining the optional switches (2 each) for the selected procedure type.

**MAX ACTIVE SET SIZE**

**Command:** job_param des_act_cons

**Description:** This optional button allows the user to specify the maximum size of the active set of constraints during design optimization. The desired number is then entered in the dialogue area. If this button is not used, the default value of 100 is taken. The use of this button properly prevents a very high number of constraints (say a hundred-thousand) being dealt within the active set. All constraints are still evaluated, and the entries in the active set change dynamically.

**COST**

**Command:** job_option des_sen_object:cost

**Description:** This button appears when SENSITIVITY or OPTIMIZATION has been pressed. By pressing this button, the user specifies the objective function to be the total cost of the structure represented by the finite element model. In this case, it is necessary that the material data contain the unit cost of the materials (per unit volume or per unit mass as appropriate).
Command: `job_param des_crit_cons`

Description: This button lights up only when the SORT CONSTRAINTS button has been pressed ON. It allows the user to specify in the dialog area, the number of most critical constraints for which sensitivity analysis is to be performed, thereby, overriding the default of 100. See also the help button for SORT CONSTRAINTS.

Command: `job_option des_sen_object:mass`

Description: This button appears when SENSITIVITY or OPTIMIZATION has been pressed. By pressing this button, the user specifies the objective function to be the total mass of the structure as represented by the finite element model. In this case, it is necessary that the material data contain the unit mass of the materials.

Command: `job_param des_opt_cycles`

Description: This optional button allows the user to change the maximum allowable number of design optimization cycles up or down from its default value of 20. By setting a reasonable number of cycles for a substantially large problem, the time spent is reduced with possibly a sacrifice in how much better the design can be improved. The number of cycles is entered in the dialog area after the button is pressed.
Command:  job_option design:off
Description:  If design SENSITIVITY or design OPTIMIZATION was previously selected, pressing this OFF button will turn off the previous selection.

Command:  job_option design:optimization
Description:  This button selects DESIGN OPTIMIZATION as the job type. When this button is pressed, the appropriate selections for DESIGN OBJECTIVE will appear along with the MAX ACTIVE SET SIZE and MAX CYCLES options (switches) which may be used with the DESIGN OPTIMIZATION parameter.

Command:  job_option design:sensitivity
Description:  This button selects DESIGN SENSITIVITY analysis as the job type. When this button is pressed, the appropriate selections for DESIGN OBJECTIVE will appear along with the SORT CONSTRAINTS and CRITICAL options (switches) which may be used with the DESIGN SENSITIVITY parameter.

Command:  job_option des_sen_object:volume
Description:  This button appears when SENSITIVITY or OPTIMIZATION has been pressed. By pressing this button, the user specifies the objective function to be the total volume of the structure as represented by the finite element model.
Command:  job_option des_sen_object:none
Description:  This button appears only when SENSITIVITY has been pressed. The user has the option of obtaining sensitivity information regarding the objective function by entering the type of objective function. Pressing this OFF button will result in no objective function being selected and the sensitivity analysis being limited to design constraints only.

Command:  job_add_desvar)
job_remove_desvar)
Description:  This command, located in the JOBS->MECHANICAL->DESIGN menu, will toggle between the ON and OFF phases of the button. ON corresponds to the related design variable being defined for the current job. OFF corresponds to the related design variable not being associated with the current job.

Keyboard Command Sequence:

job_add_desvar
job_remove_desvar

Command:  clear_job_desvars
Description:  Pressing this button clears any design variables which may have been assigned to the current job. The buttons for any assigned variables (the relevant buttons turned on) will be turned off as a result.
EDIT

Command: job_edit_host_file
Description: This command opens an edit window for the selected host file for the current job.

Keyboard Command Sequence:
job_edit_host_file

EDIT INPUT

Command: job_edit_input
Description: This command opens an edit window for the input file for the current job. The command determines the name of the file using the current model and job names.

Note that this command does not write the input file so this command must either be issued after the job is run or use the WRITE INPUT command first.

Keyboard Command Sequence:
job_edit_input

LOG FILE

Command: job_edit_log_file
Description: This command opens an edit window for the log file for the current job, and determines the name of the file using the current model and job names.

Keyboard Command Sequence:
job_edit_log_file
Command: job_edit_output
Description: This command opens an edit window for the output file for the current job, and determines the name of the file using the current model and job names.

Keyboard Command Sequence:
job_edit_output

Command: job_edit_usersub_file
Description: This command opens an edit window for the selected user subroutine file for the current job.

Keyboard Command Sequence:
job_edit_usersub_file

Command: job_option network_hosts
Description: This option is used for defining if the machines connected in the network run are compatible or not. This is only relevant if user subroutines are used. If a user subroutine is used and the connected machines are not compatible (i.e. if any remote host cannot run the executable created on the parent host), the user subroutine is compiled on each connected machine and the new executable is given a unique name. This is handled by the Marc run script automatically.

The connection of incompatible machines is only possible on machines using the UNIX operative system in this release. A restriction is that Compaq/DEC machines cannot be connected to other UNIX machines.
Keyboard Command Sequence:

```
job_option network_hosts:
    <compatible/incompatible>
```

Another Button with the same description:

```
INCOMPATIBLE
```

Command:  `job_option copy_input_file`

```
job_option copy_input_file
job_option copy_post_file
```

Description: These options are used to automatically copy input files to the remote hosts if necessary before the job starts and to copy the post files from the remote hosts back to the parent host after the job is finished.

COPY INPUT FILE  if on, copy the input files to the remote hosts
COPY POST FILE   if on, copy the post files back to the parent host

Keyboard Command Sequence:

```
job_option copy_input_file <on/off>
job_option copy_post_file <on/off>
```

Another Button with the same description:

```
COPY POST FILE
```

Command:  `job_host_file`

Description: This command specifies the name of the host file for a parallel job over a network of computers. The host file contains information on which machines to use in the run, how many processes to use on each and where the working and installation directories are. The user is responsible for editing this file and providing the necessary information.

No particular name or file extension is used for the host file. Avoid the name `jobname.host`, where `jobname` is the name of the current job, since this file is created automatically when the job is run. (If the
model name is `model1` and the current job name is `job1`, the name to avoid is `model1_job1.host`).

For each host (computer) used in the analysis, there is one line in the host file. Lines starting with `#` and blank lines are ignored. Each line must start at column 1. The general format is

```
host1 n1
host2 n2 workdir2 marcdir2
host3 n3 workdir3 marcdir3
```

This defines the three hosts `host1`, `host2`, and `host3` to be part of the analysis. `host1` uses `n1` processors, `host2` `n2` and `host3` `n3` processors. The sum `n1+n2+n3` must be equal to the number of domains used. `host1` is the machine on which the job is started. `workdir2` is the working directory for `host2`. This is where the input files for this host must be and where the results files are created. `marcdir2` is the Marc installation directory that `host2` is using. If `workdir` and `marcdir` are not specified for a host, it is assumed to be the same as for the parent host (`host1`). If `marcdir` is specified, `workdir` also needs to be specified.

Suppose `n1=n2=n2=2`. The domains 1 and 2 are run on `host1`, 3 and 4 on `host2`, and 5 and 6 on `host3`.

The directories given are as seen from that host. Suppose the working directory on `host1` is called `/users/smith` (Windows: `C:\users\smith`). Suppose that directory is also to be used on `host2` and can be accessed as `/nfs/host1/users/smith` (Windows: `\host1\users\smith`) from `host2`. On Windows, "cusers" is the UNC sharename for the directory `C:\users` (the name given in the sharing setting). Similarly for the install dir suppose Marc is installed in `/users/marc/marc2` (C:\users\marc\marc2) on `host1`. The host file would be, with one process on each host:

```
host1 1
host2 1 /nfs/host1/users/smith /nfs/host1/users/marc/marc2
Windows: host2 1 \\host1\users\smith \\host1\users\marc\marc2
```

If local directories are used on the remote host (`host2`), this can be specified as:

```
host1 1
host2 1 /users/jones /users/marcinstall/marc2
Windows: host2 1 f:\users\jones c:\marc\marc2
```

In this case, the input file for domain 2 needs to be transferred over to `/users/jones` on `host2` and the results file needs to be transferred back for postprocessing of the whole model. This is done automatically if the
COPY DAT FILE and COPY POST FILE options are activated. By default, all files are copied if necessary.

If user subroutines are used, the user subroutine is compiled and made available on all hosts automatically. If different types of UNIX machines are connected, the user subroutine is compiled on each host and the new executable is given a unique name if the working directory is shared.

**Keyboard Command Sequence:**

```
job_host_file <file name>
```
**MANUAL**

**Command:** `job_option inactive_elements`

**Menu:** JOBS

**Description:** This option is used to specify the type of input of inactive elements.

If manual is selected, user can manually select inactive elements using Add/Remove.

If import is selected, user can specify the name of the file that contains the cutter path information.

Two types of the cutter path data are supported here:

- **APT source:** Cutter path data output by CATIA. This file must have extension `.apt`.
- **CL file:** Cutter path data output by APT compilers. This file must have extension `.ccl`.

**Notes:**
1. The cutter path file must reside in the job directory.
2. If no extension for the file name is given, MSC.Marc program will add the extensions `.apt` or `.ccl`, automatically.

**Keyboard Command Sequence:**

```
job_option inactive_elements:<manual/import>
```

Another Button with the same description:

IMPORT

**INITIAL LOADS**

**Menu:** INITIAL LOADS

**Description:** This popup menu contains commands for selecting initial boundary condition applications (loads) for the current job.
ASSUMED STRAIN

Command:  job_option assumedstrn

Description: This option causes the assumed strain formulation to be used by the elements in the model that support it.

The bending behavior can be improved by using the assumed strain formulation for elements type 3 (plane stress), type 11 (plane strain), and type 7 (3-D brick). This procedure replaces the standard bi- or trilinear interpolation functions with an enriched group that is able to represent pure bending behavior. This formulation results in improved accuracy for isotropic behavior, but it should be noted that the computational costs increase.

Alternatively, this formulation can be applied to specified elements through the commands under the GEOMETRIC PROPERTIES menu.

Keyboard Command Sequence:

job_option assumedstrn:<on/off>

CONSTANT DILATATION

Command:  job_option cdilatation

Description: This option causes the constant dilatation option to be used by the elements in the model that support it.

When performing nearly incompressible analysis with displacement based elements, the conventional isoparametric interpolation methods result in poor behavior for lower order elements. This results in overly stiff behavior when using element type 7 (3-D brick), type 10 (axisymmetric), type 11 (plane strain), type 19 (generalized plane strain), or type 20 (axisymmetric with twist). When this option is included, all elements of these types are modified to use the constant dilatation formulation (also known as the mean dilatation or B-bar formulation). This is recommended for elastic-plastic analysis and creep analysis because of the potentially near incompressibility behavior.

Alternatively, this formulation can be applied to specified elements through the commands under the GEOMETRIC PROPERTIES menu.

Keyboard Command Sequence:

job_option cdilatation:<on/off>
**LINEAR ELASTIC ANALYSIS**

**Command:**  `job_option elastic`

**Description:** This command is used for specifying if a linear elastic analysis with several load cases is to be performed.

If this option is turned on, all loadcases are treated independently. All loads must then be given as total values as opposed to incremental values when a nonlinear analysis is performed. The assembled and decomposed stiffness matrices are used in all load cases, so make sure to use a direct solver.

The decomposed stiffness matrix is only written once to the restart file, following increment 0. When using the RESTART option with this feature, always restart at increment 0.

When adaptivity is used in conjunction with this option, only the initial loads are considered. These loads are then reanalyzed until the error criteria are satisfied. Initial loads are defined in INITIAL LOADS in the JOBS menus.

The memory storage used in the analysis can be reduced with this option by using one of the parameters as described below.

The alternatives for the storage are as follows:

- **FULL** no reduction in storage
- **REDUCED (1)** creep strains, plastic strains, incremental strains, plastic strain rates and incremental stresses are not stored
- **REDUCED (2)** in addition to REDUCED (1), strain energies, thermal strains and elastic strains are not stored

If you request these items on the post file and they are not stored, the information will be incorrect.

**Keyboard Command Sequence:**

```
job_option elastic:<off/on/nostore1/nostore2>
```
**Command:**  
`job_option elasticity`

**Description:** This command is used for defining the formulation to use for large elastic strains.

The choice between the total and updated Lagrangian formulation is relevant to the Mooney-Rivlin and Ogden material models only. The default for these models is to use the total Lagrangian formulation. Note that if the elements are plane strain, generalized plane strain, axisymmetric, axisymmetric with twist or three-dimensional solids, the Herrmann incompressible elements must be used. For the foam model, conventional elements are used.

The updated Lagrangian formulation for large elastic strains is available for the Mooney-Rivlin and Ogden models, including the rubber damage models. In the updated formulation, element quantities are evaluated with respect to a reference system that is updated by the current displacements. Conventional elements are used, and the material constants entered are the same as for the total Lagrangian formulation. Note that this formulation is not yet available for plane stress, membrane and shell elements.

Advantages of using the updated formulation:

- The rubber materials can be used with explicit dynamics, which only uses the updated Lagrangian formulation.
- Since Herrmann elements are not used, the solution of the equation system is more stable when the iterative solver is used.

**Reference:**  

**Keyboard Command Sequence:**  
```
job_option elasticity:<s_strn/l_strn_tot
/l_strn_upd>
```
Command:  job_option follow

Description:  This command is used for specifying that the effect of follower forces is to be taken into account in a nonlinear analysis.

This option should be used with the LARGE DISPLACEMENT option.

It causes distributed loads to be based on the current geometry as opposed to the default of having the loads based on the undeformed geometry.

In a coupled thermal-stress analysis, the fluxes are based upon the current geometry.

The follower force contributes to a stiffness effect on the tangential stiffness matrix. This effect can optionally be taken into account. It only affects the convergence of the equilibrium iteration, but not the final result. This may result in a nonsymmetric stiffness matrix, in which case you can choose to use an unsymmetric solver or use a symmetric solver. In the latter case, the matrix is made symmetric.

The options for this command are:

NO FOLLOWER FORCE  (default)
FOLLOWER FORCE  no follower force stiffness
FOLLOWER FORCE/STIFFNESS  also include follower force stiffness
FOLLOWER FORCE (BEGIN INC)  base the follower forces on the displacements at the beginning of the increment, as opposed to the last iteration

Whenever follower forces are used, the distributed magnitude given in user subroutine **FORCEM** must be the total magnitude to be reached after the current increment and not the incremental magnitude. If the loads are specified directly, incremental values are given as usual.

Keyboard Command Sequence:

```
job_option follow:<off/on/stiffness/begin_inc>
```
**Command:**  `job_option large`

**Description:** This command is used to specify that large displacements are to be taken into account in the current job.

This has the effect that the analysis is nonlinear. It signals the program to calculate the geometric stiffness matrix and the initial stress stiffness matrix.

When used together with a formulation for small strains and the UPDATED LAGRANGE PROCEDURE is not set, the total Lagrangian method is used. The program uses second Piola-Kirchhoff stress and Green-Lagrange Strain. Output are these measures as well as engineering stresses and strains, and Cauchy stresses and corresponding strains.

Large displacements are automatically taken into account when the MOONEY, OGDEN, or FOAM models are used.

This parameter should be used when performing a linearized buckling analysis with a nonlinear preloading. If the loading before buckling (in increment 0) is linear, this parameter need not be set.

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapters 5 and 7.

**Keyboard Command Sequence:**

```
job_option large:<on/off>
```
Command:  

**job_option lump**

Description:  This command flags the use of a lumped (diagonal) mass matrix for dynamics or specific heat matrix for heat transfer.

The default is to use a consistent mass matrix except for the fast central difference method.

The use of a lumped mass matrix is not recommended for second-order elements (8-node quadrilateral or 20-node brick elements) or for shell type elements.

Keyboard Command Sequence:

```
job_option lump:<on/off>
```

Other Buttons with the same description:

- LUMPED MASS
- LUMPED MASS
- CAPACITY

---

Command:  

**job_option plasticity**

Description:  This command is used to choose the formulations to use for plasticity.

The three options of this commands are:

1. small strain – mean normal
2. large strain – additive decomposition
3. large strain – multiplicative decomposition

The choices are between:

- small or large strains  which formulation to use
- additive or multiplicative  the decomposition of the deformation into elastic and plastic parts

With the small strain procedure, the workhardening data should be given in terms of engineering stress and strain, while in the large strain formulations it should be given in terms of Cauchy stress and logarithmic strain (*true* stress and strain). The large strain formulations automatically
invoke the updated Lagrangian formulation and the constant dilatation option where applicable. Avoid using three-noded triangular elements and four-noded tetrahedral elements for the large strain formulations.

The additive variant is a hypoelastic, rate-based formulation where the strain rate is decomposed into a sum of elastic and plastic terms. The Jaumann rate of Cauchy stress is used with this formulation. In the event that the elastic strains become large, this formulation does not accurately model the material response. This is also the case when very large shearing takes place in the material.

The multiplicative variant is a hyperelastic based formulation with a multiplicative decomposition of the deformation gradient. This gives a more sound treatment of elastic deformations, which is important if for instance spring-back effects are modelled. This formulation allows larger increments of strain to be used, with greater accuracy and better convergence. Currently, the material must be isotropic with this option.


**Keyboard Command Sequence:**

```
job_option plasticity:<s_strn_mn/s_strn_rr/
  l_strn_mn_add/l_strn_rr_add/l_strn_rr_mlt>
```

---

**Scale to First Yield**

**Command:** job_option scale

**Description:** This option causes scaling of a linear elastic solution to reach the yield stress in the highest stressed element. Scaling takes place for small displacement elastic-plastic analysis where element properties do not depend on temperature. All aspects of the initial solution are scaled, including displacements, strains, stresses, temperature changes, and loads. Subsequent incrementation is then based on the scaled solution.

**Keyboard Command Sequence:**

```
job_option scale:<on/off>
```
TRANSVERSE SHEAR

Command:  job_option tshear:<on/off>

Description:  This command flags the use of a special formulation for the transverse shear for thick beams and shells.

The default distribution of transverse shear strain through the thickness for thick shell element types 22, 75, and 140, and for thick beam type 45, is a constant shear-strain distribution. With this option, a more parabolic beam-like distribution derived from a strength-of-material approach is used. The distribution is exact for beam 45 but is only approximate for the shells since it is based on the assumption that the stresses in perpendicular directions are independent of each other.

For composite models, this formulation allows the calculation of interlaminar shears for these elements. The interlaminar shears are available for postprocessing (post variables il_shear_1 and il_shear_2).


Keyboard Command Sequence:

job_option tshear:<on/off>

UPDATED LAGRANGE PROCEDURE

Command:  job_option update

Description:  This command flags the use of the updated Lagrange procedure for elements for which such a formulation can be applied.

In this formulation, element quantities are evaluated with respect to a reference system that is updated by the current displacements. The program uses and prints Cauchy (true) stresses and the corresponding strain measure, the true strain.

For small strains, this procedure is typically used for beam and shell structures where the rotations are large. It should be used in conjunction with the LARGE DISPLACEMENT option. It is strongly recommended to use these options for beams and shells that undergo large rotations (but
small incremental rotations). An exception is when shell element 49 is used as it can model large incremental rotations accurately.

The updated Lagrange procedure is typically used in large strain plasticity analysis. For large strain applications (elastic or plastic), use the commands under ELASTICITY PROCEDURE and PLASTICITY PROCEDURE below instead.


**Keyboard Command Sequence:**

```
job_option update:<on/off>
```

### INCREMENTS

**Command:** *job_modal_increments*

**Description:** This command specifies for which increments a modal analysis will be performed.

**Keyboard Command Sequence:**

```
job_modal_increments <increment numbers>
```

### NAME

**Command:** *job_name*

**Description:** This command sets or changes the name of the current job.

**Keyboard Command Sequence:**

```
job_name <name>
```
Command: job_option

Description: This command sets the value of a job option. You must specify the option to be set. Job options are specified as the name of the option and its value separated by a colon (e.g. large:on, for large displacement on).

The valid job options and their values are as follows:

- **large**: off, on
  - Large displacement analysis.
- **update**: off, on
  - Updated Lagrangian.
- **finite**: off, on
  - Finite strain plasticity.
- **follow**: off, on
  - Distributed loading based on current geometry.
- **modal**: power, lanczos
  - Modal analysis solution methods (inverse power or Lanczos).
- **lump**: off, on
  - Lumped mass or heat capacity matrix.
- **dynamic**: superposit, newmark, si_st_houbolt, houbolt, central, fastcentral
  - Dynamic operator (modal superposition, Newmark beta, Single Step Houbolt, Houbolt, central difference or fast central difference). For an implicit dynamic contact analysis, the Single Step Houbolt method is recommended.
- **post**: ascii, binary
  - Type of results file.
- **dimen**: three, two, pstrain, pstress, axisym
  - Analysis dimensionality (three-dimensional, two-dimensional, plane strain, plane stress, or axisymmetric).
- **reduced**: off, on
  - Reduced integration.
- **elsto**: off, on
  - Out-of-core element storage.
- **storage**: all, centroid
  - State storage (all or at centroid only).
<table>
<thead>
<tr>
<th>Command</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>solver</td>
<td>dir_profile, it_ebe, it_sparse, nonsym, hardware, dir_sparse</td>
<td>Type of solver (direct profile, iterative element-by-element, iterative sparse matrix, nonsymmetric, hardware specific, or direct sparse matrix).</td>
</tr>
<tr>
<td>restart</td>
<td>off, on, last</td>
<td>Write restart information (on, off, or last converged increment only).</td>
</tr>
<tr>
<td>access</td>
<td>write, read, both</td>
<td>Type of restart file.</td>
</tr>
<tr>
<td>noprint</td>
<td>off, on</td>
<td>Suppress elemental and nodal output.</td>
</tr>
<tr>
<td>summary</td>
<td>off, on</td>
<td>Print increment summary.</td>
</tr>
<tr>
<td>version</td>
<td>k5, k4</td>
<td>Marc version output file type.</td>
</tr>
<tr>
<td>elastic</td>
<td>off, on, nostore1, nostore2</td>
<td>Elastic analysis with multiple loads. The value nostore1 specifies that creep, swelling, plastic and incremental strains, plastic strain rates, and incremental strains should not be stored in the elastic analysis. The value nostore2 specifies that in addition to the above, strain energies, thermal strains, and elastic strains should not be stored. Do not specify nostore1 or nostore2 if a corresponding item is to be written to the post file.</td>
</tr>
<tr>
<td>distribution</td>
<td>linear, quadratic</td>
<td>Temperature distribution for shells.</td>
</tr>
<tr>
<td>interference</td>
<td>off, on</td>
<td>Interference checking.</td>
</tr>
<tr>
<td>split</td>
<td>allowed, suppressed</td>
<td>Increment splitting.</td>
</tr>
<tr>
<td>style</td>
<td>double, single</td>
<td>Deformable-Deformable contact checking style.</td>
</tr>
<tr>
<td>sepinc</td>
<td>current, next</td>
<td>Separation increment.</td>
</tr>
<tr>
<td>Variable</td>
<td>Values</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>chatter</td>
<td>allowed, suppressed</td>
<td>Contact chattering.</td>
</tr>
<tr>
<td>frictype</td>
<td>none, shear, coulomb, shear_roll, coulomb_roll</td>
<td>Contact friction type.</td>
</tr>
<tr>
<td>fricmeth</td>
<td>nstress, nforce</td>
<td>Friction method.</td>
</tr>
<tr>
<td>motion</td>
<td>off, on</td>
<td>Call user subroutine \texttt{MOTION}.</td>
</tr>
<tr>
<td>sepfor</td>
<td>off, on</td>
<td>Call user subroutine \texttt{SEPFOR}.</td>
</tr>
<tr>
<td>ufric</td>
<td>off, on</td>
<td>Call user subroutine \texttt{UFRIC}.</td>
</tr>
<tr>
<td>uhtcoe</td>
<td>off, on</td>
<td>Call user subroutine \texttt{UHTCOE}.</td>
</tr>
<tr>
<td>uhtcon</td>
<td>off, on</td>
<td>Call user subroutine \texttt{UHTCON}.</td>
</tr>
<tr>
<td>ignore</td>
<td>off, on</td>
<td>Ignore shell thickness.</td>
</tr>
<tr>
<td>cdamping</td>
<td>off, on</td>
<td>Complex damping.</td>
</tr>
<tr>
<td>inertia</td>
<td>off, on</td>
<td>Inertia effects.</td>
</tr>
<tr>
<td>creepetype</td>
<td>expl_maxw, viscopl, nassoc, impl_maxw</td>
<td>Creep type (explicit Maxwell, explicit viscoplastic, nonassociative flow rule, implicit Maxwell).</td>
</tr>
<tr>
<td>crpvis</td>
<td>off, on</td>
<td>Call user subroutine \texttt{CRPVIS}.</td>
</tr>
<tr>
<td>rpflow</td>
<td>euler, upd_lagrange</td>
<td>Rigid-plastic procedure.</td>
</tr>
<tr>
<td>convective</td>
<td>off, on</td>
<td>Convective terms.</td>
</tr>
<tr>
<td>tshear</td>
<td>off, on</td>
<td>Transverse shear.</td>
</tr>
<tr>
<td>ebe_prec</td>
<td>none, inc_chol</td>
<td>Element-by-element solver precondition (none or incomplete Choleski).</td>
</tr>
<tr>
<td>sparse_prec</td>
<td>diag, scaled_diag, inc_chol</td>
<td>Sparse solver preconditioner (diagonal, scaled diagonal, or incomplete Choleski).</td>
</tr>
<tr>
<td>soil_calc</td>
<td>no_fluid, steady_state, transient</td>
<td>Soil calculations.</td>
</tr>
</tbody>
</table>
nonpos, off, on
soil_stress, off, on
elem_mats, off, on
tying_mats, off, on
contact_info, off, on
rezone_info, off, on
conr_flu_info, off, on
inc_disp_info, off, on
lat_heat_info, off, on
orient_info, off, on
tshear_info, off, on
load_sol_vecs, off, on
adap_info, off, on
scale, off, on
check, off, on
modal_inc, off, on
mi_sturm, off, on
mi_restart, no, write, read, both
mi_autoshift, off, on
buckle_inc, off, on
adap_attach, off, on
band_opt, on, off
parallel, on, off
pvm_to on, off

Allow nonpositive definite solution.
Perform stress analysis.
Print element matrices.
Print tying matrices.
Print contact analysis information.
Print rezoning information.
Print conrad gap and fluid change information.
Print contact status information.
Print latent heat information.
Print orientation information.
Print transverse shear.
Print load and solution vectors.
Print adaptive meshing information.
Scale to first yield.
Check sizes.
Perform modal analysis at specified increments.
Perform Sturm sequence checking during modal analysis.
Modal analysis restart file.
Modal analysis auto shift.
Perform buckle analysis at specified increments.
Attached nodes.
Bandwidth optimization.
Run Marc Analyses in parallel using domain decomposition and pvm.
Copy input files to the pvm host. If off, a link will be used.
Keyboard Command Sequence:

```
job_option <job option>[:value>
```

Other Buttons with the same description:

- 2-D
- 3-D
- ADVANCED OPTIONS
- ALL BODIES
- ALL POINTS
- ANALYTICAL DESCRIPTION MODEL  F
- ATOM SIZE
- ATTACHED NODES
- AUTO SHIFT
- AUTOMATIC
- AXISYMMETRIC
- CARTESIAN GRID
- CENTRAL DIFFERENCE
- CENTROID
- COMPLEX DAMPING
- CONRAD GAP CONTACT FLUID CH.
- CONTACT
- CONVECTIVE TERMS
- COPY TO CURRENT
- CUSTOM
- DEFAULT
- DIRECT PROFILE
- ECHO OF CONNECTIVITY
- ECHO OF COORDINATES
- ECHO OF INPUT FILE
- ELEC PROPS STRONG FUNC OF TEMP
- ELEMENT MATRICES
- EQUIV. STRESS EULERIAN TOTAL EQUIV. P
- EULERIAN
- EXPLICIT
- FAST CENTRAL DIFFERENCE
- FULL ELEMENT GEOMETRIC DISTORTION NODE PRINT
- GEOMETRIC DISTORTION
- IMMEDIATE
- IMPD (NODES)
- IMPLICIT
- IMPLICIT
- INCREM. DISP. (LOCAL)
- INERTIA EFFECTS
- INITIAL MESH
- INVERSE POWER SWEEP

```pvm_back on, off```  
Copy output files from the pvm host. If off, a link will be used.

```pvm_debug off, level1, level2, level3```  
Level of pvm debug output.
JOULE HEATING
LANCZOS
LAST CONVERGED LAST CONVERGED ONLY PERIODIC
LAST CONVERGED ONLY
LATENT HEAT
LINEAR
LOAD LOCAL ADAPTIVITY SOLUTION VECTORS
LOCAL ADAPTIVITY
MANUAL (.dat)
MANUAL (.feb)
MANUAL (.t18)
MODAL INCREMENTS
MODAL SUPERPOSITION
MODIFIED
MULTIFRONTAL SPARSE
MULTIPLE INCREMENTS FILE
NEWMARK
NO COPY
NO FLUID
OFF
ON
ORIENTATION
PERFORM STRESS ANALYSIS
PLANAR
PLANE STRAIN
PLANE STRESS
QUADRATIC
READ
READ REBARnVERIFICATION WRITE
REBARnVERIFICATION
REZONING
SINGLE BODY
SINGLE INCREMENT FILE
SINGLE INCREMENT FILE(S)
SINGLE-STEP HOUBOLT
STEADY STATE
STRESS DISCONTINUITY
STURM SEQUENCE
SUMMARY
THROUGH POINT
TRANSIENT
TRANSVERSE SHEAR
TYING MATRICES
UNMODIFIED
UPDATED LAGRANGIAN
WRITE
# DYNAMIC MODES

**Command:** job_param  
**Description:** This command sets the value of a job parameter. You must specify the parameter to be set and its value.

Valid job parameters are as follows:

- **post**
  - Results file increment frequency.

- **memory**
  - Number of words of memory to allocate in the Marc job.

- **layers**
  - Number of shell layers to output.

- **restart**
  - Restart increment frequency.

- **start**
  - Restart starting increment.

- **vsliding**
  - Contact job - sliding velocity.

- **disttol**
  - Contact job - distance tolerance.

- **instrain**
  - Initial strain rate.

- **cutoff**
  - Cutoff strain rate.

- **sepforce**
  - Contact job - separation force.

- **nmodes**
  - Number of dynamic modes.

- **disttolbias**
  - Contact job - distance tolerance bias.

- **maxsep**
  - Contact job - maximum number of separations per increment.

- **nbmodes**
  - Maximum number of buckling modes.

- **npbModes**
  - Maximum number of positive buckling modes.

- **max_groups**
  - Maximum number of groups.

- **ebe_maxit**
  - Element-by-element solver maximum number of iterations.

- **ebe_stress_tol**
  - Element-by-element solver stress analysis conjugate gradient convergence tolerance.

- **ebe_heats_tol**
  - Element-by-element solver heat analysis conjugate gradient convergence tolerance.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>Element-by-element solver condition number cutoff.</td>
</tr>
<tr>
<td>sparse_maxit</td>
<td>Sparse solver maximum number of iterations.</td>
</tr>
<tr>
<td>sparse_tol</td>
<td>Sparse solver convergence tolerance.</td>
</tr>
<tr>
<td>max_el</td>
<td>Adaptive meshing maximum number of elements.</td>
</tr>
<tr>
<td>max_nod</td>
<td>Adaptive meshing maximum number of nodes.</td>
</tr>
<tr>
<td>max_fixed_dof</td>
<td>Adaptive meshing maximum fixed degrees-of-freedom.</td>
</tr>
<tr>
<td>max_cont_seg</td>
<td>Adaptive meshing maximum contact segments.</td>
</tr>
<tr>
<td>max_cont_nod</td>
<td>Adaptive meshing maximum contact nodes.</td>
</tr>
<tr>
<td>conversion</td>
<td>Heat generation conversion factor.</td>
</tr>
<tr>
<td>mi_low</td>
<td>Modal analysis Lanczos method lowest frequency.</td>
</tr>
<tr>
<td>mi_high</td>
<td>Modal analysis Lanczos method highest frequency.</td>
</tr>
<tr>
<td>mi_nmodes</td>
<td>Modal analysis Lanczos method number of modes.</td>
</tr>
<tr>
<td>mi_maxit</td>
<td>Modal analysis power sweep method maximum number of iterations.</td>
</tr>
<tr>
<td>mi_tol</td>
<td>Modal analysis power sweep method tolerance.</td>
</tr>
<tr>
<td>mi_initshift</td>
<td>Modal analysis power sweep method initial shift.</td>
</tr>
<tr>
<td>mi_maxfreq</td>
<td>Modal analysis power sweep method maximum frequency.</td>
</tr>
<tr>
<td>mi_shiftmodes</td>
<td>Modal analysis power sweep method number of modes per shift.</td>
</tr>
<tr>
<td>mi_shiftpar</td>
<td>Modal analysis power sweep method shift parameter.</td>
</tr>
<tr>
<td>bi_maxit</td>
<td>Buckle analysis maximum number of iterations.</td>
</tr>
<tr>
<td>bi_tol</td>
<td>Buckle analysis tolerance.</td>
</tr>
<tr>
<td>ncpus</td>
<td>Number of cpus that Marc should utilize (on machines with multiple cpus only).</td>
</tr>
</tbody>
</table>
Keyboard Command Sequence:

`job_param <job parameter name> <value>`

Other Buttons with the same description:

# MODES
# MODES PER SHIFT
# RECYCLES
# STEPS
# USER DEFINED VECTORS (USER S % OF TOTAL LOAD
2-D CONTACT LIMIT ANGLE
3-D CONTACT LIMIT ANGLE
BOUNDARY CONDITION PENALTY MUL
CONVERSION FACTOR
DYNAMIC CONTACT PROJECTION FAC
END TIME
FLUID INCOMPRESSIBILITY PENALT
FREQUENCY
HIGHEST FREQUENCY
INCREMENT FREQUENCY
INCREMENTAL STRAIN PREDICTION
INITIAL SHIFT
LOWEST FREQUENCY
MAX # ITERATIONS
MAX STEP SIZE
MEAN STRESS SUBTRACTION FACTOR
NEWMARK-BETA BETA PARAMETER
NEWMARK-BETA GAMMA PARAMETER
OUT-OF-CORE THRESHOLD
REASSEMBLY INTERVAL
RESTART INCREMENT
RIGID PLASTIC CUTOFF STRAIN RA
RIGID PLASTIC INCOMPRESSIBILITY
RIGID PLASTIC INIT. STRAIN RATE
SHELL ELEMENT DRILLING MODE FACT
SHIFT PARAMETER
SINGLE-STEP HOUBOLT GAMMA PARAM
SINGLE-STEP HOUBOLT GAMMA1 PARA
STEFAN-BOLTZMANN CONSTANT
TIME STEP (DYNAMIC ANALYSIS)
TOLERANCE
UNIVERSAL GAS CONSTANT
X
Y
Z
**BANDWIDTH OPTIMIZATION**

**Command:** `job_option band_opt`  
**Description:** This command activates the nodal bandwidth optimization for the current model.  
This is important if a direct solver is used in order to minimize the solution time and the amount of storage.  
For the iterative solvers, this parameter has no effect.  
For the profile solver, the default method is the Sloan algorithm. If the Cuthill-McKee algorithm is to be used, the Marc input file must be modified.  
For the sparse direct solver, the minimum degree algorithm is used.  

**Keyboard Command Sequence:**  
`job_option band_opt::<on/off>`

**ALL POINTS**

**Command:** `job_option storage`  
**Description:** This command determines the way the calculation and storage of stresses and strains (or temperature for heat transfer) is handled.  
**ALL POINTS** Stresses and strains (or temperatures) are calculated and stored in all integration points of the elements (default).  
**CENTROID** Stresses and strains (or temperatures) are calculated and stored in the centroid of each element only. This can be used in a linear analysis in order to reduce storage requirements and computational cost. It should not be used in a nonlinear analysis.  

**Keyboard Command Sequence:**  
`job_option storage::<all/centroid>`

Another Button with the same description:  
`CENTROID`
OUT-OF-CORE ELEMENT STORAGE

Command:  job_option elsto
Description: This command flags the ELSTO option in Marc.
This has the effect that element quantities are stored out-of-core. This reduces the amount of main memory required in the analysis, but may increase the amount of I/O.
Keyboard Command Sequence:

job_option elsto:<on/off>

# SHELL/BEAM LAYERS

Command:  job_param layers
Description: This command specifies the number of layers (through thickness integration points) that is used by shell and beam-in-a-plane elements.
This number must be odd.
For linear material behavior, one point is sufficient.
For most nonlinear problems, 7 points are sufficient to describe the response accurately. For extremely nonlinear response, such as elastic-plastic dynamic problems, 11 points might be needed.
Keyboard Command Sequence:

job_param layers <value>
**MEMORY ALLOCATION**

**Command:** `job_param memory`

**Description:** This command is used for setting the amount of memory that MSC.Marc is to be use. MSC.Marc parameter `sizing`.

In the MSC.Marc release 7.1 and later, dynamic memory allocation is used. In this case, the value given here is the initial allocation. Additional memory is then requested when needed. If set to higher than what is needed, this amount is used in the analysis. If a zero value is given, a default value is used for the initial memory allocation (defined in the file `include` in the `tools` directory of the MSC.Marc distribution). In some cases, it is advisable to set a correct memory allocation, for instance, when amounts close to the limit of the computer is needed.

In earlier versions of MSC.Marc, the amount of memory needed in the analysis must be defined with this command.

**Reference:** *MSC.Marc Volume: Program Input: Appendix B, page B-1*

**Keyboard Command Sequence:**

```
job_param memory <value>
```

**Menu:** **SOLVER**

**Description:** This menu contains commands for choosing the solver type for the current job.

There are two types of solvers available in MSC.Marc: `direct` and `iterative` solvers.

The `direct solvers` use a strategy similar to classical Gaussian elimination but optimized for the special structure of the matrices in the finite element method.

The `iterative solver` is based on a preconditioned conjugate gradient method. The solution of the equation system is obtained in an iterative way. The number of iterations required for convergence is in general not known a priori. This type of solver can be very efficient in solving equation systems, both with respect to time and storage. These methods are not efficient if the equation system is ill-conditioned. This often occur in problems involving beams and shells due to the large difference in
membrane and bending stiffness. The performance of iterative solvers is often improved if preconditioners are used.

There is also the choice of storage method. The profile storage is advantageous if few zeros are present within the bandwidth. If the matrix is sparse, with many zeros within the bandwidth, the sparse storage system is more efficient.

For certain platforms (HP, SGI, and Sun), hardware optimized solvers are available. These are usually more efficient than the standard solvers. They can also be used in a parallel analysis.

The iterative solver is usually not appropriate for the following situations:

- elastic analysis with multiple load vectors
- explicit creep analysis with multiple load vectors
- complex harmonic analysis
- substructures
- beam and shell analysis
- eigenvalue analysis
- use of gap or Herrmann elements

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11

Another Button with the same description:

**SOLVER**

**CONDITION NR CUTOFF**

**Command:** `job_param condition`

**Description:** This command is used for setting a condition number cut-off. If the calculated condition number of the equation system is less than this value, the analysis is stopped.

**Note:** This command is available in SuperForm only.

**Keyboard Command Sequence:**

`job_param condition <value>`
HEAT ANALYSIS TOL.

Command:  \texttt{job_param ebe\_heat\_tol}

Description: This command is used for setting the tolerance on conjugate gradient convergence for heat transfer analysis.

The solution is converged when the maximum residual force over the maximum reaction force is less than the specified tolerance.

\textbf{Note:} This command is available in SuperForm only.

Keyboard Command Sequence:

\texttt{job_param ebe\_heat\_tol <value>}

\begin{center}
\textbf{MAX \# GROUPS}
\end{center}

Command:  \texttt{job_param max\_groups}

Description: This command is used for setting the maximum number of groups used by the EBE iterative solver.

\textbf{Note:} This command is available in SuperForm only.

Keyboard Command Sequence:

\texttt{job_param max\_groups <value>}

\begin{center}
\textbf{MAX \# ITERATIONS}
\end{center}

Command:  \texttt{job_param ebe\_maxit}

Description: This command is used for setting the maximum number of conjugate gradient iterations.

\textbf{Note:} This command is available in SuperForm only.

Keyboard Command Sequence:

\texttt{job_param ebe\_maxit <value>
Command:  **job_option ebe_prec**

Description: This command is used for choosing a preconditioner for the EBE iterative solver.

A preconditioner is used for improving the convergence behavior of the iterative solver. Both sides of the equation is premultiplied with a preconditioning matrix.

- NONE: no preconditioner is used
- INCOMPLETE CHOLESKI: the use of this method usually improves convergence

**Note:** This command is available in SuperForm only.

Keyboard Command Sequence:

```
job_option ebe_prec:<none/chol>
```

---

Command:  **job_param ebe_stress_tol**

Description: This command is used for setting the tolerance on conjugate gradient convergence for stress analysis.

The solution is converged when the maximum residual force over the maximum reaction force is less than the specified tolerance.

**Note:** This command is available in SuperForm only.

Keyboard Command Sequence:

```
job_param ebe_stress_tol <value>
```
OUT-OF-CORE THRESHOLD

Command:  job_param hardw_sparse_thresh

Description: This command is used for setting a memory threshold for when the solver goes out-of-core.

This option is only available for the SGI hardware sparse solver. If the memory needed for the solver is larger than the specified limit, the solver will run out-of-core, that is, data needed for the solution is stored on the hard disk instead of being kept in memory. If zero is entered the limit is given by the maximum amount of memory allowed as defined by the variable MAXSIZE in the include file in the tools subdirectory of the MSC.Marc installation directory.

Keyboard Command Sequence:

job_param hardw_sparse_thresh <value>

DIRECT PROFILE

Command:  job_option solver

Description: This command selects the solution method for linear equation systems.

The following solvers are available:

DIRECT PROFILE

DIRECT SPARSE  the default method.

ITERATIVE SPARSE  uses a sparse storage method. Requires more input.

HARDWARE SPARSE  available for HP, SGI, and Sun.

MULTIFRONTAL SPARSE  a direct sparse solver

Solution of nonsymmetric systems can be performed with the MULTIFRONTAL SPARSE and the DIRECT PROFILE solver.

Out-of-core capabilities are available for all the solvers except the ITERATIVE SPARSE and HARDWARE SPARSE for HP and Sun. For a parallel run using domain decomposition, only MULTIFRONTAL SPARSE and HARDWARE SPARSE for SGI support of out-of-core solution.
Systems with complex arithmetics is supported by MULTIFRONTAL SPARSE and DIRECT PROFILE, both for symmetric and nonsymmetric systems.

**Reference:**  *MSC.Marc Volume A: Theory and User Information*, Chapter 11

**Keyboard Command Sequence:**

```
job_option solver:<dir_profile/dir_sparse/it_ebe/
  it_sparse/hardware/sparse/nonsym>
```

Other Buttons with the same description:

- DIRECT SPARSE
- HARDWARE SPARSE
- ITERATIVE SPARSE
- MULTIFRONTAL SPARSE
- NONSYMMETRIC SOLUTION

**Command:**  `job_param sparse_maxit`

**Description:**  This command is used for setting the maximum number of conjugate gradient iterations. The job will stop if this maximum is reached.

**Keyboard Command Sequence:**

```
job_param sparse_maxit <value>
```
**DIAGONAL**

**Command:**  
\texttt{job_option sparse_prec}

**Description:**  
This command is used for choosing a preconditioner for the sparse iterative solver.

A preconditioner is used for improving the convergence behavior of the iterative solver. Both sides of the equation are premultiplied with a preconditioning matrix.

- **DIAGONAL:** use the diagonal of the stiffness matrix.
- **SCALED DIAGONAL:** scale the diagonal terms appropriately in order to improve convergence.
- **INCOMPLETE CHOLESKI:** this is usually the method of choice.

**Keyboard Command Sequence:**

\texttt{job_option sparse_prec:<diag/scaled_diag/inc_chol>}

Other Buttons with the same description:

- **INCOMPLETE CHOLESKI**
- **SCALED DIAGONAL**

**TOLERANCE**

**Command:**  
\texttt{job_param sparse_tol}

**Description:**  
This command is used for setting the tolerance on conjugate gradient convergence for the sparse iterative solver.

The solution is converged when the maximum residual force over the maximum reaction force is less than the specified tolerance.

**Keyboard Command Sequence:**

\texttt{job_param sparse_tol <value>
**USER DATA MEMORY ALLOCATION**

**Command:** `job_param usdata_alloc`

**Description:** This option invokes the call to MSC.Marc user subroutine `USDATA` for the initialization of user data.

In MSC.Marc, these data are stored in a common block `USCADM` that can be used in other subroutines. The value given in this option defines the amount of data used in this common block in `real*4` words. This is needed for the memory allocation.

**Keyboard Command Sequence:**

```
job_param usdata_alloc <value>
```

---

**job_param_version**

**Command:** `job_option version`

**Description:** This command toggles the release of MSC.Marc which is to be used in the current job.

**Keyboard Command Sequence:**

```
job_option version:<k7/k6/k5/k4>
```

---

**JOB PARAMETERS**

**Menu:** PARAMETERS

**Description:** This popup menu contains commands for setting job parameters for the current job.
**Mentat Help Commands in J**

**job_print_contact**

**Command:** `job_option contact_info`

**Description:** This command turns on additional print of information about the contact to the output file.

Information is given on when a node comes into contact, what rigid body or segment is contacted, when separation occurs, when a node contacts a sharp corner, the displacements in the local system, and the force in the local coordinate system. For large problems, this can result in a significant amount of output.

**Keyboard Command Sequence:**

```
job_option contact_info:<on/off>
```

**RESET**

**Command:** `job_reset`

**Description:** This command resets all job parameters and options to their default values.

**Keyboard Command Sequence:**

```
job_reset
```

**RESTART FILE**

**Command:** `job_restart_file`

**Description:** This command specifies the name of the job restart file.

**Keyboard Command Sequence:**

```
job_restart_file <file name>
```
**JOB RESULTS**

 Menu: JOB RESULTS
 Description: This popup menu contains commands for selecting output parameters for the current job.

**RUN**

 Menu: RUN
 Description: This popup menu contains commands for controlling and running the current MSC.Marc job.

**CHECK**

 Command: `job_option check`
 Description: This command causes the MSC.Marc `stop` command to be written to the input file. When MSC.Marc is run, it will stop after memory allocation. Useful for finding out the amount of memory needed for the analysis.

 Keyboard Command Sequence:

 `job_option check:<on/off>`

 Another Button with the same description:

 CHECK SIZES
**Command:**  `job_option user_source`

**Description:** This command defines how the user subroutine file defined for this job is to be treated.

- **COMPILE / NO SAVE**
  The user subroutine file is compiled but the new executable file is not saved.

- **COMPILE AND SAVE**
  The user subroutine file is compiled and the new executable file is saved. If the user subroutine file is called `file.f`, the executable will be called `file.marc`.

- **RUN SAVED EXECUTABLE**
  Instead of compiling the user subroutine file, the previously saved executable is used.

**Keyboard Command Sequence:**

```
job_option user_source:<compile_nosave/
compile_save/run_saved>
```

Another Button with the same description:

IMPORT
**Command:** job_option dcom:<on/off>

**Description:** This command toggles the DCOM server option in MSC.Marc.

- It is only available for Windows.
- If this option is on, the job_dcom_hostname command is enable to allow a server name to be specified in the adjacent job_dcom_hostname option.
- To use this option, you MUST ensure that:
  - The machine that is specified as the host must be enabled as a MSC.Marc DCOM server.
  - Your datafiles are on shared directories.
  - Your username/password is the same as for the host.

**Keyboard Command Sequence:**

   job_option dcom:<on/off>

---

**Menu:** SETTINGS

**Description:** This menu contains settings and definitions for the host file used in a network parallel job.
Command: job_option parallel_setup

Description: This command sets the parallel execution type to SINGLE MACHINE or NETWORK. The number of processors to use in an analysis is equal to the number of domains created for the model.

- SINGLE MACHINE: The parallel analysis is performed on a single multiprocessor computer.
- NETWORK: The parallel analysis is performed on a network of separate computers. The machines to be used and other settings are defined under the SETTINGS menu.

Keyboard Command Sequence:

job_option parallel_setup:<single/network>

Another Button with the same description:

SINGLE MACHINE

Command: job_option parallel:<on/off>

Description: This command toggles the parallel option in MSC.Marc.

If this option is on, a parallel job is performed when one of the SUBMIT commands are used. Input files for a parallel job are written with the WRITE INPUT command.

The parallel option in MSC.Marc uses DDM (Domain Decomposition Method). The model is decomposed into domains of elements and each domain is solved on one processor (CPU). The communication of data between the CPUs is made using MPI (Message Passing Interface). The domains are created using the options under JOBS-> DOMAIN DECOMPOSITION.

The analysis can be performed on a single multiprocessor computer or on a network of separate computers.

Many of the features of MSC.Marc are supported in parallel mode. Specifically, the contact option including thermomechanical coupling is supported.
The list of unsupported features are:

- acoustics
- adaptive
- auto therm creep
- bearing
- buckling
- beam-to-beam contact
- design sensitivity and optimization
- electromagnetics
- explicit dynamics
- fluid and its coupled analysis
- gap elements
- harmonic
- hydrodynamics
- j-integral
- out-of-core solver
- out-of-core element storage (elsto)
- radiation
- response spectrum
- rezoning/remeshing

**Keyboard Command Sequence:**

```
job_option parallel:<on/off>
```

**Command:** `job_option inp_file_prec`

**Description:** This command is used to set the option for using extended precision in the input file.

- All integers are written using 10 digits instead of 5 with the normal option.
- All floating point numbers are written using 20 or 30 positions instead of 10 or 15 with the normal option.
- All character strings are written using 20 positions instead of 10 with the normal option.
- This allows models with more than 99,999 elements which is the limit with the normal option.

**Keyboard Command Sequence:**

```
job_option inp_file_prec:<normal/extended>
```
**SELECTED USER SUBS**

**Menu:** SEL. USER SUBS

**Description:** This menu contains a list of all user subroutines that are flagged to be used in the current model.

These subroutines are expected to be present in the user subroutine file selected by the USER SUB FILE button above (keyboard command `job_usersub_file <file name>`).

---

**SCRATCH DIRECTORY**

**Command:** `job_scratch_dir`

**Description:** This command allows you to set the directory that MSC.Marc will use to store scratch files. This defaults to directory in which the `.dat` file is located. This is used with the `-sdir` option to the `run_marc` command.

**Keyboard Command Sequence:**

`job_scratch_dir <directory path>`

---

**Command:** `job_scratch_directory`

**Description:** This command allows you to set the directory that MSC.Marc will use to store scratch files. This defaults to directory in which the `.dat` file is located. This is used with the `-sdir` option to the `run_marc` command.

**Keyboard Command Sequence:**

`job_scratch_directory <directory path>`
SORT CONSTRAINTS

Command:  
- `job_option des_sort_con:on` (turns on the button)
- `job_option des_sort_con:off` (turns off the button)

Description:  Pressing this button on specifies that the prescribed constraints will be sorted in order of criticalness before sensitivity analysis is performed. In this case, unless the # CRITICAL button is also pressed, the default number of constrained response quantities for which sensitivity analysis will be performed is set to the most critical 100 by default. If the # CRITICAL button is then pressed, the user can now change this default up or down.

STEADY STATE ROLLING

Menu:  STEADY STATE ROLLING

Description:  This popup menu contains commands for setting rotation and cornering axes parameters for steady state rolling.

Menu:  STEADY STATE ROLLING PARAMETERS

Description:  This menu contains commands for setting steady state rolling parameters.

- **AXIS OF ROTATION**
  The axis of rotation is defined by specifying its DIRECTION and a POINT on the axis.

- **AXIS OF CORNERING**
  The axis of cornering is defined by specifying its DIRECTION and a POINT on the axis.

Keyboard Command Sequence:

- `job_param ss_roll_rot_axis_dir_x <x_component_of_direction>`
- `job_param ss_roll_rot_axis_dir_y <y_component_of_direction>`
- `job_param ss_roll_rot_axis_dir_z <z_component_of_direction>`

- `job_param ss_roll_rot_axis_point_x <x_component_of_point>`
- `job_param ss_roll_rot_axis_point_y <y_component_of_point>`
- `job_param ss_roll_rot_axis_point_z <z_component_of_point>`
job_param ss_roll_corn_axis_dir_x
<x_component_of_direction>
job_param ss_roll_corn_axis_dir_y
<y_component_of_direction>
job_param ss_roll_corn_axis_dir_z
<z_component_of_direction>

job_param ss_roll_corn_axis_point_x <x_component_of_point>
job_param ss_roll_corn_axis_point_y <y_component_of_point>
job_param ss_roll_corn_axis_point_z <z_component_of_point>

Other Buttons with the same description:
Y
Z

Command: job_submit_reset
Description: This command resets the submit status of the current job to Not Submitted. This command is useful if a MSC.Marc job has been terminated before completion and MSC.Marc Mentat is not able to detect the fact that the job is no longer running.

Keyboard Command Sequence:
job_submit_reset

Command: job_title
Description: This command specifies an optional title to be written into the MSC.Marc input file.

Keyboard Command Sequence:
job_title <job title>

Another Button with the same description:
TITLE
**USER SUBROUTINE FILE**

**Command:** job_usersub_file  
**Description:** This command specifies the name of the file containing the source code for the user subroutines needed for the current job. The user is responsible for editing this file and providing the source code. Templates for most user subroutines can be found in the user subdirectory in the MSC.Marc directory. The .f file name extension may be omitted.

**Keyboard Command Sequence:**

```
job_usersub_file <file name>
```

**CONTACT LOCATION MODEL FILES**

**Command:** job_option write_contact_bm_bm  
job_param bm_bm_files_freq  
**Description:** The command "job_option write_contact_bm_bm" is used to make the location of the contact points in a beam-to-beam contact analysis available in formatted MSC.Marc Mentat model files, written by MSC.Marc.

The files contain line segments that connect the contact points on contacting beam elements. They can be merged with the current postprocessing model during postprocessing for visualization purposes. Since the location of these points may change from increment to increment, the contact location can be made available for different increments.

The frequency of writing this information is user-defined and can be set by the command "job_param bm_bm_files_freq". The files are named jobname_bbc_inc.mfd, in which inc is the increment number.

Also see: post_merge_bbc.

**Keyboard Command Sequence:**

```
job_option write_contact_bm_bm_files:<on/off>
job_param bm_bm_files_freq <value>
```
WRITE INPUT

Command:  job_write_input
Description: This command writes the MSC.Marc input file without submitting the job. This command should be used when the input file needs to be inspected or modified before submission. The name of the file produced is determined from the current model and job names. After the input file has been created and/or modified, it may be submitted with the execute_job command.

Keyboard Command Sequence:
  job_write_input

ANALYTICAL DESCRIPTION MODEL FILES

Command:  job_write_spline_files
Description: This option is used to get the spline description of analytical deformable contact bodies available in formatted MSC.Marc Mentat model files, written by MSC.Marc.

For a 2-D analysis, the spline description is represented by NURBS curves, for a 3-D analysis by NURBS surfaces. Moreover, the geometric entities representing the boundaries of contact bodies are stored in sets. Since the position and shape of a deformable contact body may change from increment to increment, the spline representation can be made available for different increments.

The frequency of writing this information is user-defined and can be set by the command "job_param spline_files_freq". The files are named jobname_spline_inc.mfd, in which inc is the increment number.

Keyboard Command Sequence:
  job_write_spline_files:<on/off>
  job_param spline_files_freq <value>
Menu: JOBS
Description: The JOBS menu contains commands that define, run, and monitor MSC.Marc jobs.

**JPEG QUALITY**

Command: `jpeg_quality`
Description: This command will set the JPEG Quality factor. This value has a range of 1 through 100. The higher the value, the higher the color quality of the image, but it will result in a larger file size.

Keyboard Command Sequence:
```
jpeg_quality <value>
```

**JPEG SMOOTHING**

Command: `jpeg_smoothing`
Description: This command will set the JPEG Smoothness factor. This value has a range of 1 through 30. The higher the value, the more color blending and edge smoothing with adjacent pixels will take place, but you may see a loss of brightness. It only has a slight effect on file size.

Keyboard Command Sequence:
```
jpeg_smoothing <value>
```
Mentat Help Commands in K

Command: kill_job
Description: This command terminates the currently running job.

   The word STOP is written to the file jobname.cnt. Marc reads this file and exits with exit number 3020.

Keyboard Command Sequence:

   kill_job
Mentat Help Commands in L

**LABEL CONTOURS**

**Command:** `label_contours`

**Description:** This command sets whether or not contour line plots will be labeled.

Also see help: `post_contour_lines`

**Keyboard Command Sequence:**

```
label_contours <yes/no>
```

**BOUNDARY CONDITION PENALTY MULTIPLIER**

**Command:** `loadcase_value bcond_pen_mp`

**Description:** This command is used to enter the multiplier used to calculate the penalty factor to impose boundary conditions.

The penalty factor is this multiplier times the maximum diagonal value of the operator matrix. The default multiplier is 1.0e9.
This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

**Keyboard Command Sequence:**

```plaintext
loadcase_value bcond_pen_mp <value>
```

---

**NEWMARK-BETA BETA PARAMETER**

**Command:** `loadcase_value beta`

**Description:** This command is used to enter the beta parameter used in transient dynamic analysis using the Newmark-Beta procedure. The default is 0.25.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

**Keyboard Command Sequence:**

```plaintext
loadcase_value beta <value>
```

---

**OFF**

**Command:** `loadcase_option approach_sync`

**Description:** This command is used to specify the way rigid bodies are moved during a body approach loadcase. If the synchronize approach is turned off (default), then the rigid bodies are moved until they all touch a deformable body, whereas, if it is turned on the bodies are moved until one is touching a deformable body.

**Keyboard Command Sequence:**

```plaintext
loadcase_option approach_sync:<on/off>
```

Another Button with the same description:

ON
**LOADCASE TIME**

**Command:** `loadcase_value time`

**Description:** This command is used to set the time period of the motion of the rigid bodies during the body approach loadcase.

This time period is only used for output purposes and to coordinate with tables and user subroutines. An exception is for coupled analysis where a transient period is performed.

**Reference:** *MSC.Marc Volume C, Program Input*, Chapter 4

**Keyboard Command Sequence:**

```
loadcase_value time <time period>
```

**BODY APPROACH**

**Menu:** BODY APPROACH

**Description:** This menu contains commands for defining the body approach as a loadcase. This allows you to move rigid bodies so that they make contact with deformable bodies. In the case of multistage forging, you usually have a time period where the first set of bodies are released, followed by a new time period where the second set of bodies are brought into contact.

This option is used in conjunction with the CONTACT TABLE option to determine which bodies are now applicable, and the MOTION CHANGE option which prescribes the velocity of the new bodies. A time step is associated with this period, but it is only used for output purposes and to coordinate with tables and user subroutines except in coupled analysis where a transient period is performed. Marc moves each rigid body until either all of the bodies are in contact or the first. This behavior depends on if the SYNCHRONIZE option is used or not.

**Reference:**
- *MSC.Marc Volume C: Program Input*, Chapter 4: History Definition
- APPROACH
- *MSC.Marc Volume A: Theory and User Information*, Chapter 8
**Menu:** BUCKLE

**Description:** This menu contains commands for defining a buckling analysis as a loadcase.

By choosing this option, a buckling analysis will be performed in the current loadcase. A buckling analysis is usually performed after a preloading. The estimated buckling load is then given as a scaling of the preload. If the analysis is nonlinear prior to the buckle analysis, the formulation for large deformations should be used.

The only settings that are done in this menu are parameters for the inverse power sweep method. All other settings are done in the JOBS menus. The choice between the inverse power sweep method and the Lanczos method is done under JOBS->MECHANICAL->ANALYSIS OPTIONS and the specification of the number of buckling modes to be extracted is done under JOBS->MECHANICAL->JOB PARAMETERS. The eigenvectors, stresses and the reactions are automatically set to be printed out to the post file.

The option in Marc to perform a perturbation of the geometry by the buckling mode is not yet supported by Mentat and must be entered into the input file by the user.

A related command is the BUCKLE INCREMENT entered under JOBS->MECHANICAL->ANALYSIS OPTIONS which allows a buckling analysis to be performed after given increments in a loadcase. This can be used instead of the current option or in connection with it.

**Reference:**
- *MSC.Marc Volume C: Program Input*, Chapter 2 and Chapter 4
- *MSC.Marc Volume A: Theory and User Information*, Chapter 5

**Command:** loadcase_value buckle_tol
loadcase_value maxit

**Description:** These commands specify parameters for the inverse power sweep method if it is used in the current loadcase.
The power sweep terminates when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance.

If the number of iteration required is larger than the maximum specified, the analysis stops. The default number is 40.

Reference:  *MSC.Marc Volume C, Program Input*, Chapter 4

**Keyboard Command Sequence:**

```
loadcase_value buckle_tol
  <tolerance for the inverse power sweep method>
loadcase_value maxit
  <maximum number of iterations>
```

Another Button with the same description:

TOLERANCE

**Menu: CONVERGENCE TESTING**

**Description:** This popup menu contains commands for setting convergence testing parameters for the current loadcase.

Other Buttons with the same description:

CONVERGENCE TESTING FLUID REGI
CONVERGENCE TESTING HEAT TRANS
CONVERGENCE TESTING SOLID REGI

**TOTAL CREEP TIME**

**Command:** loadcase_value creeptime

**Description:** This command is used to enter the total creep time to be covered during this AUTO THERM CREEP option. This value can be changed at restart by using the RESTART option. The default is 0.

This option is available in the LOADCASES->MECHANICAL->CREEP-> (TEMPERATURE, CREEP STRAIN/STRESS)->PARAMETERS menu.

**Keyboard Command Sequence:**

```
loadcase_value creeptime <value>
```
**CUTOFF STRESS TOLERANCE**

**Command:** `loadcase_value cutoff_stress`

**Description:** This command is used to enter the tolerance on low stress point cut-off.

Points with a stress lower than this ratio relative to the maximum stress in the structure are not used in the creep tolerance checking. The default is 0.05.

This option is available in the LOADCASES->MECHANICAL->CREEP-(CREEP STRAIN/STRESS)->PARAMETERS menu.

**Keyboard Command Sequence:**
```
loadcase_value cutoff_stress <value>
```

**DESIGN CONSTRAINTS**

**Menu:** DESIGN CONSTRAINTS

**Description:** Pressing this button will bring up the pop-up menu for associating previously specified design constraints (see the button DESIGN in the Main Menu) with the current loadcase being defined.

**MAX PRESSURE CHANGE ALLOWED**

**Command:** `loadcase_value dp_allowed`

**Description:** This command specifies the maximum nodal pressure change allowed. It is used to control the automatic time step scheme for heat transfer.

This option is available in the LOADCASES->THERMAL-DIFFUSION->TRANSIENT->CONVERGENCE TESTING menu.

**Keyboard Command Sequence:**
```
loadcase_value dp_allowed <value>
```
### MAX PRESSURE CHANGE BEFORE REASSEMBLY

**Command:** `loadcase_value dp_assemble`  
**Description:** This command specifies the maximum nodal pressure change allowed before properties are re-evaluated and matrices reassembled.  
The default value is 100 degrees.  
This option is available in the `LOADCASES->THERMAL-DIFFUSION->TRANSIENT->CONVERGENCE TESTING` menu.

**Keyboard Command Sequence:**

```
loadcase_value dp_assemble <value>
```

### MAX ERROR IN PRESSURE ESTIMATE

**Command:** `loadcase_value dp_error`  
**Description:** This command specifies the maximum error in pressure estimate used for property evaluation. This control provides a recycling capability to improve accuracy in highly nonlinear heat transfer problems (for example, latent heat, radiation boundary conditions).  
The default value is 0, which bypasses this test. It should be set to the maximum pressure error which is considered acceptable.  
This option is available in the `LOADCASES->THERMAL-DIFFUSION->STEADY STATE->CONVERGENCE TESTING` and `TRANSIENT` menus.

**Keyboard Command Sequence:**

```
loadcase_value dp_error <value>
```
**MAX TEMPERATURE CHANGE ALLOWED**

Command: `loadcase_value dt_allowed`

**Description:** This command specifies the maximum nodal temperature change allowed. It is used to control the automatic time step scheme for heat transfer.

This option is available in the LOADCASES->HEAT TRANSFER->TRANSIENT->CONVERGENCE TESTING menu.

**Keyboard Command Sequence:**

```
loadcase_value dt_allowed <value>
```

---

**MAX TEMPERATURE CHANGE BEFORE REASSEMBLY**

Command: `loadcase_value dt_assemble`

**Description:** This command specifies the maximum nodal temperature change allowed before properties are re-evaluated and matrices reassembled. The default value is 100 degrees.

This option is available in the LOADCASES->HEAT TRANSFER->TRANSIENT->CONVERGENCE TESTING menu.

**Keyboard Command Sequence:**

```
loadcase_value dt_assemble <value>
```

---

**MAX ERROR IN TEMPERATURE ESTIMATE**

Command: `loadcase_value dt_error`

**Description:** This command specifies the maximum error in temperature estimate used for property evaluation. This control provides a recycling capability to improve accuracy in highly nonlinear heat transfer problems (for example, latent heat, radiation boundary conditions).

The default value is 0, which bypasses this test. It should be set to the maximum temperature error which is considered acceptable.

This option is available in the LOADCASES->HEAT TRANSFER->STEADY STATE->CONVERGENCE TESTING and TRANSIENT menus.

**Keyboard Command Sequence:**

```
loadcase_value dt_error <value>
```
**DYNAMIC CONTACT PROJECTION FACTOR**

**Command:** `loadcase_value dyn_contact_pro`  
**Description:** This command is used to enter the surface projection factor for the Single Step Houbolt procedure. The default is 0.0.  
This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.  

**Keyboard Command Sequence:**  
`loadcase_value dyn_contact_pro <value>`

---

**FLUID INCOMPRESSIBILITY PENALTY**

**Command:** `loadcase_value fluid_incomp_pen`  
**Description:** This command is used to enter the penalty factor used to satisfy incompressibility in fluid analysis when displacement elements are used. The default is 1.0e6.  
This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.  

**Keyboard Command Sequence:**  
`loadcase_value fluid_incomp_pen <value>`

---

**MAXIMUM ABSOLUTE VELOCITY**

**Command:** `loadcase_value fluid_maxvelo`  
**Description:** This command specifies the convergence tolerance when absolute residual testing on velocity is used. If set to zero, no checking on residual velocity takes place.  
This option is available in the LOADCASES->FLUID->STEADY STATE->CONVERGENCE TESTING and other FLUID loadcase class menus.  

**Reference:**  
*Volume A: Theory and User Information, Chapter 11.*  

**Keyboard Command Sequence:**  
`loadcase_value fluid_maxvelo <value>`
MINIMUM VELOCITY CUTOFF

Command: `loadcase_value fluid_minvelo`

Description: This command specifies a cutoff value for the velocity when relative testing on velocity is used. If the velocity is less than this value, checking will be bypassed, or if the RELATIVE/ABSOLUTE option is used, absolute testing will be used.

This option is available in the LOADCASES->FLUID->STEADY STATE->CONVERGENCE TESTING and other FLUID loadcase class menus.


Keyboard Command Sequence:
```
loadcase_value fluid_minvelo <value>
```
HIGHEST FREQUENCY

Command:  \texttt{loadcase\_value high}

Description:  This command is used to enter the highest frequency of mode to be extracted in cycles per time if the job option LANCZOS is selected.

For a DYNAMIC MODAL analysis, this value is used provided that the option RANGE (\texttt{loadcase\_option frequency:range}) is selected. When the option RANGE is not used, the value set with the \# MODES button (\texttt{loadcase\_value nmode}) is used. This can be changed upon restart.

This option is available in the LOADCASES$\rightarrow$MECHANICAL$\rightarrow$DYNAMIC MODAL, DYNAMIC HARMONIC and SPECTRUM RESPONSE menus.

Also see the help for the \texttt{loadcase\_option frequency} (the NUMBER or RANGE buttons).

Keyboard Command Sequence:
\[
\texttt{loadcase\_value high <value>}
\]

INACTIVE ELEMENTS

Command:  \texttt{loadcase\_option inactive\_elements}

Menu:  LOADCASES

Description:  This option is used to specify the type of input of inactive elements.
If manual is selected, the user can manually select inactive elements using Add/Remove.
If import is selected, user can specify the name of the file that contains the cutter path information.

Two types of the cutter path data are supported here:

APT source:  Cutter path data output by CATIA. This file must have extension .apt.

CL file:  Cutter path data output by APT compilers. This file must have extension .ccl.

Notes:  1. The cutter path file must reside in the job directory.
   2. If no extension for the file name is given, MSC.Marc program will add the extensions: .apt or .ccl, automatically.

Keyboard Command Sequence:
\[
\texttt{loadcase\_option inactive\_elements:<manual/import>}
\]
# INCS BETWEEN REASSEMBLY

**Command:** `loadcase_value inc_assemble`  
**Description:** This command is used to enter the number of increments between stiffness matrix updates. This option is used to prevent unnecessary updating of the stiffness matrix during large displacement analysis.  
This option is available in the LOADCASES->MECHANICAL->CREEP->(CREEP STRAIN/STRESS)->PARAMETERS and other ADAPTIVE STEPPING menus.  
**Keyboard Command Sequence:**  
`loadcase_value inc_assemble <value>`

## INCREMENTAL STRAIN PREDICTION MULTIPLIER

**Command:** `loadcase_value inc_strn_pred_mp`  
**Description:** This command is used to enter the incremental strain prediction multiplier. The default is 1.0.  
This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.  
**Keyboard Command Sequence:**  
`loadcase_value inc_strn_pred_mp <value>`

## INITIAL SHIFT

**Command:** `loadcase_value initshift`  
**Description:** This command is used to enter the initial shift in cycles per time. The power shift is likely to start converging to the eigenvalue closest to this value. The default is 0.  
This option is available in the LOADCASES->MECHANICAL->DYNAMIC MODAL menu.  
**Keyboard Command Sequence:**  
`loadcase_value initshift <value>`
INITIAL TIME STEP

Command:  `loadcase_value initime`

Description: This command is used to enter the suggested time increment for the analysis.

When the automatic creep control is in use, MSC.Marc iterates for the appropriate increment size to satisfy the tolerances placed on stress and strain increments. The default is 1.0.

This option is available in the \texttt{LOADCASES->MECHANICAL->CREEP->(CREEP STRAIN/STRESS)->PARAMETERS} menu.

Keyboard Command Sequence:

```
loadcase_value initime <value>
```

2-D CONTACT LIMIT ANGLE

Command:  `loadcase_value limit_angle_2d`

Description: This command is used to enter the angle at which a node separates from a convex corner or becomes stuck in a concave corner for two-dimensional contact. The default is 8.625 degrees.

This option is available in the \texttt{LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES} and other loadcase type menus.

Keyboard Command Sequence:

```
loadcase_value limit_angle_2d <value>
```

3-D CONTACT LIMIT ANGLE

Command:  `loadcase_value limit_angle_3d`

Description: This command is used to enter the angle at which a node separates from a convex corner or becomes stuck in a concave corner for three-dimensional contact. The default is 20.0 degrees.

This option is available in the \texttt{LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES} and other loadcase type menus.

Keyboard Command Sequence:

```
loadcase_value limit_angle_3d <value>
```
Menu: LOADS

Description: This popup menu contains commands for selecting loads and boundary condition applications for the current loadcase.

LOWEST FREQUENCY

Command: loadcase_value low

Description: This command is used to enter the lowest frequency of mode to be extracted in cycles per time. This is also the initial shift point. This cannot be changed upon restart. The default is 0.

This option is available in the LOADCASES->MECHANICAL->DYNAMIC MODAL, DYNAMIC HARMONIC, and SPECTRUM RESPONSE menus.

Keyboard Command Sequence:
loadcase_value low <value>

MAXIMUM ITERATIVE DISPLACEMENT COMPONENT

Command: loadcase_value max_it_disp_comp

Description: This command is used to enter the maximum iterative displacement component. The default is 1.0e30.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
loadcase_value max_it_disp_comp <value>
**MAXIMUM # INCREMENTS**

**Command:** `loadcase_value maxinc`  
**Description:** This command is used to enter the maximum number of time increments to be allowed during this part of the analysis. The default is 50 increments.  
This option is available in the LOADCASES->MECHANICAL->STATIC->(ARC LENGTH)->PARAMETERS and other loadcase type menus.  

**Keyboard Command Sequence:**  
`loadcase_value maxinc <value>`

**MAXIMUM # INTERATIONS**

**Command:** `loadcase_value maxit`  
**Description:** This command is used to enter the value for the maximum number of iterations allowed to modify the time step during an increment. The default is 5.  
This option is available in the LOADCASES->MECHANICAL->BUCKLE and LOADCASES->MECHANICAL->STATIC->(CREEP STRAIN/STRESS)->PARAMETERS menus.  

**Keyboard Command Sequence:**  
`loadcase_value maxit <value>`

**MAX # SUBINCREMENTs**

**Command:** `loadcase_value maxsubinc`  
**Description:** This command is used to enter the maximum number of subincrements to be allowed during this part of the creep analysis. The default is 50.  
This option is available in the LOADCASES->MECHANICAL->CREEP->(TEMPERATURE CREEP STRAIN/STRESS)->PARAMETERS menu.  

**Keyboard Command Sequence:**  
`loadcase_value maxsubinc <value>`
MEAN STRESS SUBTRACTION FACTOR

Command: \texttt{loadcase\_value mean\_strs\_subtr}

Description: This command is used to enter the fraction of the hydrostatic pressure that is subtracted from the stress tensor in the initial stress calculation. The default is 1.0.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
\texttt{loadcase\_value mean\_strs\_subtr <value>}

# FREQUENCIES

Command: \texttt{loadcase\_value nfreq}

Description: This command is used to enter the maximum number of frequencies to be extracted in a dynamic harmonic analysis. The default is 1.

This option is available in the LOADCASES->MECHANICAL->DYNAMIC menu.

Keyboard Command Sequence:
\texttt{loadcase\_value nfreq <value>}

# MODES

Command: \texttt{loadcase\_value nmodes}

Description: This command is used to enter the number of modes to be extracted in cycles per time provided that the option \texttt{NUMBER} (\texttt{loadcase\_option frequency:number}) is used.

When the option \texttt{NUMBER} is not used, the value set with the HIGHEST FREQUENCY button (\texttt{loadcase\_value high}) is used. This can be changed upon restart. The default is 10.

This option is available in the LOADCASES->MECHANICAL->DYNAMIC MODAL, DYNAMIC HARMONIC, and SPECTRUM RESPONSE menus.

Also see the help for the \texttt{loadcase\_option frequency} (the \texttt{NUMBER} or \texttt{RANGE} buttons).
Keyboard Command Sequence:

.loadcase_value nmodes <value>

**Menu:** BUCKLE

**Description:** This menu contains commands for defining a buckling analysis as a loadcase.

By choosing this option, a buckling analysis will be performed in the current loadcase. A buckling analysis is usually performed after a pre-loading. The estimated buckling load is then given as a scaling of the preload. If the analysis is nonlinear prior to the buckling analysis, the formulation for large deformations should be used.

The only settings that are done in this menu are parameters for the inverse power sweep method. All other settings are done in the JOBS menus. The choice between the inverse power sweep method and the Lanczos method is done under

JOBS->PIEZO-ELECTRIC->ANALYSIS OPTIONS

and the specification of the number of buckling modes to be extracted is done under

JOBS->PIEZO-ELECTRIC->JOB PARAMETERS.

The eigenvectors, stresses and the reactions are automatically set to be printed out to the post file.

The option in MSC.Marc to perform a perturbation of the geometry by the buckling mode is not yet supported by MSC.Marc Mentat and must be entered into the input file by the user.

A related command is the BUCKLE INCREMENT entered under JOBS->PIEZO-ELECTRIC->ANALYSIS OPTIONS which allows a buckling analysis to be performed after given increments in a loadcase. This can be used instead of the current option or in connection with it.

MECHANICAL and ELECTROSTATIC boundary conditions can be selected.

Make sure that for piezo electric elements the MECHANICAL, ELECTROSTATIC and PIEZO-ELECTRIC material properties are set.

**Reference:**

*MSC.Marc Volume A: Theory and User Information*, Chapter 5

*MSC.Marc Volume C: Program Input*, Chapter 2 & 4
**CREEP**

**Menu:** CREEP  
**Description:** This button sets the current loadcase to be a piezo-electric creep analysis.  
The menu contains commands for defining active loads, solution controls, convergence testing parameters, active contact definitions, numerical parameter settings, and details of the load stepping procedure.  
**MECHANICAL** and **ELECTROSTATIC** boundary conditions can be selected.  
Make sure that for piezo electric elements the **MECHANICAL**, **ELECTROSTATIC**, and **PIEZO-ELECTRIC** material properties are set.

**DYNAMIC HARMONIC**

**Menu:** DYNAMIC HARMONIC  
**Description:** This button sets the current loadcase to be a piezo-electric dynamic harmonic analysis.  
The menu contains commands to set the harmonic loads and the number of frequencies with the range.  
**MECHANICAL** and **ELECTROSTATIC** boundary conditions can be selected.  
Make sure that for piezo electric elements, the **MECHANICAL**, **ELECTROSTATIC** and **PIEZO-ELECTRIC** material properties are set.

**DYNAMIC MODAL**

**Menu:** DYNAMIC MODAL  
**Description:** This button sets the current loadcase to be a piezo-electric dynamic modal analysis.  
The menu contains commands for the lanczos method and for the power sweep method.  
**MECHANICAL** and **ELECTROSTATIC** boundary conditions will be used if they are set as initial boundary conditions in JOBS.  
Make sure that for piezo electric elements the **MECHANICAL**, **ELECTROSTATIC**, and **PIEZO-ELECTRIC** material properties are set.
DYNAMIC TRANSIENT

Menu: DYNAMIC TRANSIENT

Description: This button sets the current loadcase to be a piezo-electric dynamic transient analysis.

The menu contains commands for defining active loads, solution controls, convergence testing parameters, active contact definitions, numerical parameter settings, and details of the load stepping procedure.

MECHANICAL and ELECTROSTATIC boundary conditions can be selected.

Make sure that for piezo electric elements the MECHANICAL, ELECTROSTATIC and PIEZO-ELECTRIC material properties are set.

STATIC

Menu: STATIC

Description: This button sets the current loadcase to be a piezo-electric static analysis.

The menu contains commands for defining active loads, solution controls, convergence testing parameters, active contact definitions, numerical parameter settings, and details of the load stepping procedure.

MECHANICAL and ELECTROSTATIC boundary conditions can be selected.

Make sure that for piezo electric elements the MECHANICAL, ELECTROSTATIC and PIEZO-ELECTRIC material properties are set.
**HIGHEST FREQUENCY**

Command: `loadcase_value pshigh`

Description: This command is used to enter the maximum frequency to be extracted in cycles per time if the job option INVERSE POWER SWEEP is selected. If this value is zero, the number of modes requested on the DYNAMIC parameter is extracted. If this value is nonzero, the extraction ends when this frequency is exceeded or when the number of modes requested on the DYNAMIC parameter is reached, whichever occurs first. The default is 0.

This option is available in the LOADCASES->MECHANICAL->DYNAMIC MODAL menu.

Keyboard Command Sequence:
```
loadcase_value pshigh <value>
```

**RIGID PLASTIC CUTOFF STRAIN RATE**

Command: `loadcase_value rp_ctff_strn_rt`

Description: This command is used to enter the cutoff strain rate for rigid plastic analysis. The default is 1.0e-12.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
```
loadcase_value rp_ctff_strn_rt <value>
```
RIGID PLASTIC INCOMPRESSIBILITY PENALTY

Command:  loadcase_value rp_incomp_pen

Description: This command is used to enter the penalty factor used to satisfy the incompressibility in rigid plastic analysis for plane strain, axisymmetric, or solid analysis when displacement elements are used. The default is 100.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
loadcase_value rp_incomp_pen <value>

RIGID PLASTIC INIT. STRAIN RATE

Command:  loadcase_value rp_init_strn_rt

Description: This command is used to enter the initial strain rate for rigid plastic analysis. The default is 1.0e-4.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
loadcase_value rp_init_strn_rt <value>

SHELL ELEMENT DRILLING MODE FACTOR

Command:  loadcase_value shell_drill

Description: This command is used to enter the factor used to calculate the drilling mode for shell elements type 22, 75, 138, 139, and 140. The default is 0.0001.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:
loadcase_value shell_drill <value>
# MODES PER SHIFT

**Command:** `loadcase_value shiftmodes`  
**Description:** This command is used to enter the number of modes extracted per shift. This data field is active only if **AUTO SHIFT** is selected. If **AUTO SHIFT** is not required, it is set equal to or greater than the number of modes requested on the **DYNAMIC** parameter. The default is 5.

This option is available in the **LOADCASES->MECHANICAL->DYNAMIC MODAL** menu.

Also see the help for the `loadcase_option autoshift` (**AUTO SHIFT**).

**Keyboard Command Sequence:**
```
loadcase_value shiftmodes <value>
```

## SHIFT PARAMETER

**Command:** `loadcase_value shiftpar`  
**Description:** This command is used to enter the **AUTO SHIFT** parameter. MSC.Marc determines the new shift point (in frequency squared) as the highest frequency square plus this times the difference between the highest and the next highest distinct frequency squared. The default is 1.0.

This option is available in the **LOADCASES->MECHANICAL->DYNAMIC MODAL** menu.

Also see the help for the `loadcase_option autoshift` (**AUTO SHIFT**).

**Keyboard Command Sequence:**
```
loadcase_value shiftpar <value>
```

## SINGLE-STEP HOUBOLT GAMMA PARAMETER

**Command:** `loadcase_value si_st_hou_gamma`  
**Description:** This command is used to enter the gamma parameter used in transient dynamic analysis using the Single Step Houbolt procedure. The default is -.50.

This option is available in the **LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES** and other loadcase type menus.
Keyboard Command Sequence:

```
loadcase_value si_st_hou_gamma <value>
```

**SINGLE-STEP HOUBOLT GAMMA1 PARAMETER**

**Command:** `loadcase_value si_st_hou_gamma1`

**Description:** This command is used to enter the gamma1 parameter used in transient dynamic analysis using the Single Step Houbolt procedure. The default is 1.50.

This option is available in the LOADCASES->MECHANICAL->STATIC->NUMERICAL PREFERENCES and other loadcase type menus.

Keyboard Command Sequence:

```
loadcase_value si_st_hou_gamma1 <value>
```

**SOLUTION CONTROL**

**Menu:** SOLUTION CONTROL

**Description:** This popup menu contains commands for setting solution control parameters for the current loadcase.

**STEADY STATE ROLLING**

**Menu:** STEADY STATE ROLLING

**Description:** This popup menu contains commands for setting steady state rolling parameters for the current loadcase.

**STEADY STATE ROLLING**

**Menu:** STEADY STATE ROLLING

**Description:** This button sets the current loadcase to be a steady state rolling analysis. The menu contains commands for defining active loads, solution controls, convergence testing parameters, active contact definitions, steady state rolling parameters, numerical parameter settings and details of the load stepping procedure.
**VAR ID**

**Command:** loadcase_value state_var_id

**Description:** This command specifies the state variable identifier to be used in the analysis.

This option is available in the LOADCASES->HEAT TRANSFER->TRANSIENT-(MULTICRITERIA)->PARAMETERS->USER-DEFINED CRITERIA, and also under the JOULE HEATING and JOULE-MECHANICAL loadcase class menus.

**Keyboard Command Sequence:**

```
loadcase_value state_var_id <value>
```

**STATIC**

**Menu:** STATIC

**Description:** This button sets the current loadcase to be a mechanical static analysis.

The menu contains commands for defining active loads, solution controls, convergence testing parameters, active contact definitions, active gap geometries, numerical parameter settings, active design constraints, and details of the load stepping procedure.

**STRAIN CHANGE TOLERANCE**

**Command:** loadcase_value strain_change

**Description:** This command is used to enter the appropriate creep strain value.

If RELATIVE testing is selected, enter the tolerance on the creep strain increment relative to the elastic strain. The default is 0.50. Note that a high value is likely to cause stability problems.

If ABSOLUTE testing is selected, enter the maximum creep strain increment allowed. The default is 0.1.

This option is available in the LOADCASES->MECHANICAL->CREEP->(CREEP STRAIN/STRESS)->PARAMETERS menu.

**Keyboard Command Sequence:**

```
loadcase_value strain_change <value>
```
**STRESS CHANGE TOLERANCE**

**Command:** `loadcase_value stress_change`  
**Description:** This command is used to enter the appropriate creep stress value.  
If *RELATIVE* testing is selected, enter the tolerance on the stress change per stress during creep. The default is 0.10.  
If *ABSOLUTE* testing is selected, enter the maximum stress increment allowed. The default is 100.  
This control is included primarily for accuracy purposes. The default value is adequate for creep laws of the type $\epsilon = A * \sigma^n$, where $3 < n < 6$; for higher values, it should be decreased.  
This option is available in the `LOADCASES->MECHANICAL->CREEP->(CREEP STRAIN/STRESS)->PARAMETERS` menu.  

**Keyboard Command Sequence:**  
`loadcase_value stress_change <value>`

**TOLERANCE**

**Command:** `loadcase_value tolerance`  
**Description:** This command is used to enter the convergence tolerance. The power sweep terminates when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance. The default is $1.0e^{-5}$.  
This option is available in the `LOADCASES->MECHANICAL->DYNAMIC MODAL` menu.  

**Keyboard Command Sequence:**  
`loadcase_value tolerance <value>`
**X DISPLACEMENT WEIGHT FACTOR**

**Command:** `loadcase_value weight_dx <dy,dz,rx,ry,rz>`

**Description:** These commands are used to enter the weighting factor for the first (X DISP), second (Y DISP), third (Z DISP), fourth (X ROT), fifth (Y ROT) and sixth (Z ROT) degrees of freedom for a spectrum response analysis. This option is available in the LOADCASES->MECHANICAL->SPECTRUM RESPONSE menu.

**Keyboard Command Sequence:**

```
loadcase_value weight_dx <value>
loadcase_value weight_dy <value>
loadcase_value weight_dz <value>
loadcase_value weight_rx <value>
loadcase_value weight_ry <value>
loadcase_value weight_rz <value>
```

Other Buttons with the same description:

- Y DISPLACEMENT WEIGHT FACTOR
- Z DISPLACEMENT WEIGHT FACTOR
- X ROTATION WEIGHT FACTOR
- Y ROTATION WEIGHT FACTOR
- Z ROTATION WEIGHT FACTOR

**EXPONENTIAL**

**Command:**
- `legend_format_aut`
- `legend_format_exp`
- `legend_format_flt`
- `legend_format_int`

**Description:** This button toggles the type of format to display for the legend numerics. Available formats are:

- AUTOMATIC (command `legend_format_aut`)
  The format will be the default and the exponent will be displayed.

- EXPONENTIAL (command `legend_format_exp`)
  The mantissa precision to the right of the decimal may be specified using the PRECISION (`legend_prec`) command. There will be one digit to the left of the decimal point and the exponent will be displayed.
FLOATING (command `legend_format_flt`)
The exponent will not be displayed. The PRECISION (`legend_prec`) command is used to adjust the number of digits displayed.

INTEGER (command `legend_format_int`)
The values displayed will be integer values. The values will be rounded to the nearest integer.

**Keyboard Command Sequence:**
```
legend_format_aut
legend_format_exp
legend_format_flt
legend_format_int
```

### PRECISION

**Command:** `legend_prec`

**Description:** This command specifies the number of digits to the right of the decimal point for the following LEGEND FORMAT commands:

- EXPONENTIAL (`legend_format_exp`)
- FLOATING (`legend_format_flt`)

The value specified must be in the range of 0 – 10.

**Keyboard Command Sequence:**
```
legend_prec <value>
```

### # LEVELS

**Command:** `levels`

**Description:** This command sets the number of contour levels displayed in contour plots. The maximum is 30. The default is 10.

**Keyboard Command Sequence:**
```
levels <number of levels>
```
Command: **light_ambient**

**Description:** This command changes the color of the light’s ambient component in a view. The ambient light is nondirectional, and contributes to the overall light in a view, whenever this light is turned on in that view. You must specify the view and the number of the light whose ambient color you wish to change. There are eight lights, numbered 1 through 8. You must also specify the new red, green, and blue values for the ambient light. These values range from zero to one, where zero means completely OFF, and one means completely ON. Lighting and the given light must be ON in the view in order to see the effect.

Also see help: `set_lighting`
- `set_light, light_color`
- `light_ambient`

**Keyboard Command Sequence:**
```
lighting_ambient <view> <red> <green> <blue>
```
LOCAL

Command:  
  light_local  
  light_infinite

Description:  These commands make a light local or infinite in a view. A local light is an omnidirectional light source, with an absolute location in view space. An infinite light represents a light a great distance away (like the sun), and shines only in the direction away from its location. Only a local light can be a spotlight, in which case, it is no longer omnidirectional. You must specify the view and the number of the light to make local or infinite. There are eight lights, numbered 1 through 8. Lighting and the given light must be ON in the view in order to see the effect.

Also see help:  set_lighting  
  set_light  
  set_light_spotlight  
  light_location

Keyboard Command Sequence:  
  light_local <view> <light>  
  light_infinite <view> <light>

Another Button with the same description:  
  INFINITE

LOCATION DIRECTION

Command:  light_location

Description:  This command specifies a light’s location in a view. Infinite lights use this location as a direction vector pointing toward the light in view space, and local lights use this location as their absolute position in view space. You must specify the view and the number of the light whose location you wish to set. There are eight lights, numbered 1 through 8. Lighting and the given light must be ON in the view in order to see the effect.

Also see help:  set_lighting  
  set_light  
  dynamic_lighting_on

Keyboard Command Sequence:  
  light_location <view> <light> <x> <y> <z>
**RADIUS**

**Command:**  *light_radius*

**Description:** This command changes the radius of a local light in a view. You must specify the view and the number of the light whose radius you wish to change. There are eight lights, numbered 1 through 8. You must also specify the new radius for the light. The radius must be greater than or equal to zero, and is measured in view space.

Also see help: 
- `set_lighting`
- `set_light`
- `light_local`

**Keyboard Command Sequence:**

```
light_radius <view <light> <radius>
```
**SPOTLIGHT RADIUS ANGLES**

**Command:** light_spot_angles

**Description:** This command specifies a spotlight’s outer and inner conic radius angles in a view. The outer radius angle can range from 0 through 90 degrees, and specifies the outermost cone radius angle for a spotlight. The inner radius angle can range from 0 up to the outer radius angle, and specifies the cone radius angle for a spotlight at which the light intensity begins its falloff. You must specify the view and the number of the spotlight whose angles you wish to change. There are eight lights, numbered 1 through 8. Lighting and the given light must be ON, and it must be a spotlight in the view in order to see the effect.

```
Also see help: set_lighting
               set_light
               set_light_spotlight
               light_spot_direction.
```

**Keyboard Command Sequence:**

```
light_spot_angles <view> <light> <outer_angle> <inner_angle>
```

**SPOTLIGHT DIRECTION**

**Command:** light_spot_direction

**Description:** This command specifies a spotlight’s direction in a view. A spotlight will shine light in a conic shape facing this direction in view space. You must specify the view and the number of the spotlight whose direction you wish to change. There are eight lights, numbered 1 through 8. Lighting and the given light must be ON, and it must be a spotlight in the view in order to see the effect.

```
Also see help: set_lighting
               set_light
               set_light_spotlight
               dynamic_spotlight_aiming_on
               light_spot_angles.
```

**Keyboard Command Sequence:**

```
light_spot_direction <view> <light> <x> <y> <z>
```
AMBIENT LIGHT COLOR

Command: `lighting_ambient`

Description: This command changes the color of the ambient light in a view. The ambient light is nondirectional, and contributes to the overall light in a view, whenever lighting is turned ON in that view. You must specify the view and the new red, green, and blue values for the ambient light. These values range from zero to one, where zero means completely OFF, and one means completely ON.

Also see help: `set_lighting`
`light_ambient`

Keyboard Command Sequence:
```
lighting_ambient <view> <red> <green> <blue>
```

ATTENUATION FACTORS

Command: `lighting_attenuation`

Description: This command changes the amount of lighting attenuation in a view. Lighting attenuation is relevant when lighting is ON and `set_lighting_attenuation` is ON. Lighting attenuation affects only the local lights in a view. When attenuation is ON, the strength of a local light may diminish as its distance from geometry increases. You must specify the view and the new constant (k0), linear (k1), and distance-squared (k2) attenuation factors. The formula for attenuation is:

\[ \text{attenuation factor} = \frac{1}{(k0 + k1 \times \text{dist} + k2 \times \text{dist} \times \text{dist})} \]

where dist is the distance between the geometry and a local light source in view space. The values 1.0, 0.0, 0.0 for k0, k1, and k2 effectively turns OFF attenuation.

Also see help: `set_lighting`
`set_lighting_attenuation`
`light_local`

Keyboard Command Sequence:
```
lighting_attenuation <view> <k0> <k1> <k2>
```
**LINEAR CONTACT WITH REDUCED STORAGE**

**Command:**  
`job_option linear_contact_rs`

**Description:**  
This command defines whether the analysis should use the linear contact with reduced storage option to reduce memory and CPU usage for problems that have stable contact conditions. For example, an engine block to head gasket to cylinder head contact analysis would see significant benefits if using this option.

This capability will attempt to reduce memory usage by performing all contact calculations first before any other memory allocation. When the contact portion has been completed, the contact surface data and some contact vectors are deleted and the stiffness matrix allocated for.

Only contact data necessary to solve subsequent systems are saved. For three-dimensional models with a large amount of contact surface data, this means that significant memory savings can be realized, allowing for the stiffness matrix to be retained in-core.

If the element storage is transferred to the disk by using the OUT-OF-CORE ELEMENT STORAGE option (elsto), most of the memory available can be used to store the stiffness matrix.

Since the contact data saved are mostly limited to the constraints and the state vectors, all subsequent contact behavior is linearized. At the beginning of the analysis, contact between bodies is established. Contacting nodes are subsequently allowed to slide along a plane tangential and attached to the initial point of contact.

**RESTRICTIONS:**
- 3-D models only
- Friction and coupled behavior are not supported

This button is located under the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT menu.

**Keyboard Command Sequence:**

```
job_option linear_contact_rs:<on/off>
```
**link_class**

**Command:**  
*link_class*

**Description:**  
This command sets the class of the current link. You must provide the link class. The valid classes of links are nodal ties (tie), servo links (servo), and springs/dashpots (spring).

**Keyboard Command Sequence:**  

\[ \text{link_class} \text{ <servo, spring, or tie>} \]

**link_class_set**

**Menu:**  
SET

**Description:**  
This menu contains commands for setting properties of the current link.

**CREATE PATHS**

**Command:**  
*link_multi_servo_create_paths*

**Description:**  
This command toggles the flag that controls whether or not the *link_multi_servo_n_to_n* command will create node paths from the nodes input by the user.

**Keyboard Command Sequence:**  

\[ \text{link_multi_servo_create_paths} \text{ on or off} \]

**ADD SERVOS**

**Command:**  
*link_multi_servo_n_to_1*

**Description:**  
This command creates multiple servo links from the specified tied nodes (a list of N nodes) to the previously specified retained nodes.

**Keyboard Command Sequence:**  

\[ \text{link_multi_servo_n_to_1} \text{ <list of N tied nodes> #} \]
ADD SERVOS

Command:  link_multi_servo_n_to_n
Description:  This command creates multiple servo links from the specified tied nodes (a node path of N nodes) to the retained nodes specified by the node paths of N nodes each. The number of retained node paths corresponds to the number of retained terms previously specified.

Keyboard Command Sequence:
```
link_multi_servo_n_to_n <path of N tied nodes> #
<path of N retained nodes> #
.
.
.<path of N retained nodes> #
```

# TERMS

Command:  link_multi_servo_nterms
Description:  This command sets the number of terms for servo links to be created with the link_multi_servo_n_to_1 and link_multi_servo_n_to_1 commands.

Keyboard Command Sequence:
```
link_multi_servo_nterms <# terms>
```

COEF.

Command:  link_multi_servo_rcoef
Description:  This command sets the retained coefficients for the servo links to be created with the link_multi_servo_n_to_1 and link_multi_servo_n_to_1 commands. You must specify the term number.

Keyboard Command Sequence:
```
link_multi_servo_rcoef <term #> <coefficient>
```
Command: **link_multi_servo_rdof**

Description: This command sets the retained degree-of-freedom for the servo links to be created with the `link_multi_servo_n_to_1` and `link_multi_servo_n_to_1` commands. You must specify the term number.

**Keyboard Command Sequence:**

```plaintext
link_multi_servo_rdof <term #> <dof>
```

Command: **link_multi_servo_reset**

Description: This command resets the multi-servo link tied degree-of-freedom and retained nodes, degrees-of-freedom, and coefficients.

**Keyboard Command Sequence:**

```plaintext
link_multi_servo_reset
```

Command: **link_multi_servo_rnode**

Description: This command sets the retained nodes for the servo links to be created with the `link_multi_servo_n_to_1` command. You must specify the term number.

**Keyboard Command Sequence:**

```plaintext
link_multi_servo_rnode <term #> <retained node>
```
Command: link_multi_servo_tdof
Description: This command sets the tied degree-of-freedom for servo links to be created with the link_multi_servo_n_to_1 and link_multi_servo_n_to_1 commands.

Keyboard Command Sequence:
link_multi_servo_tdof <tied dof>

Command: link_multi_spring_damping
Description: This command sets the damping coefficient for the springs/dashpots to be created with the link_multi_spring_n_to_1 and link_multi_spring_n_to_n commands.

For a linear dashpot, this value is used directly.

For a nonlinear dashpot (i.e. when a table is provided), this property value will scale the values in the table if the GRADIENT option (set_link_multi_gradient) is not specified.

Keyboard Command Sequence:
link_multi_spring_damping <damping coefficient>

Command: link_multi_spring_dof
Description: This command sets the degree-of-freedom at the ends of the springs/dashpots to be created with the link_multi_spring_n_to_1 and link_multi_spring_n_to_n commands. You must specify the slot (end) with a 0 or 1.

Keyboard Command Sequence:
link_multi_spring_dof <0 or 1> <degree-of-freedom>
Command:  **link_multi_spring_enode**
Description:  This command sets the end node for the springs/dashpots to be created with the **link_multi_spring_n_to_1** command.

**Keyboard Command Sequence:**
```
link_multi_spring_enode <node>
```

---

ADD SPRINGS

Command:  **link_multi_spring_n_to_1**
Description:  This command creates multiple spring/dashpots from the specified begin nodes (a list of N nodes) to the previously specified end node.

**Keyboard Command Sequence:**
```
link_multi_spring_n_to_1
  <list of N begin nodes> #
```

---

ADD SPRINGS

Command:  **link_multi_spring_n_to_n**
Description:  This command creates multiple spring/dashpots from the specified begin nodes (a node path of N nodes) to the end nodes (another node path of N nodes).

**Keyboard Command Sequence:**
```
link_multi_spring_n_to_n
  <path of N begin nodes> #
  <path of N end nodes> #
```

---

RESET

Command:  **link_multi_spring_reset**
Description:  This command resets the multi-spring/dashpot stiffness, damping coefficient, user subroutine setting, degrees-of-freedom, and end node.

**Keyboard Command Sequence:**
```
link_multi_spring_reset
```
STIFFNESS

Command:  link_multi_spring_stiffness
Description:  This command sets the stiffness for the springs/dashpots to be created with the link_multi_spring_n_to_1 and link_multi_spring_n_to_n commands.

For a linear spring, this value is used directly.

For a nonlinear spring (i.e. when a table is provided), this property value will scale the values in the table if the GRADIENT option (set_link_multi_gradient) is not specified.

Keyboard Command Sequence:
link_multi_spring_stiffness <stiffness>

USER SUBROUTINE USPRNG

Command:  link_multi_spring_usersub
Description:  This command specifies whether or not user subroutine USPRNG will be called during a MSC.Marc run for the springs/dashpots to be created with the link_multi_spring_n_to_1 and link_multi_spring_n_to_n commands.

Keyboard Command Sequence:
link_multi_spring_usersub <on or off>

CREATE PATHS

Command:  link_multi_tie_create_paths
Description:  This command toggles the flag that controls whether or not the link_multi_tie_n_to_n command will create node paths from the nodes input by the user.

Keyboard Command Sequence:
link_multi_tie_create_paths <on or off>
Command: **link_multi_tie_n_to_1**

**Description:** This command creates multiple nodal ties from the specified tied nodes (a list of \( N \) nodes) to the one (1) retained node.

**Keyboard Command Sequence:**

```
link_multi_tie_n_to_1 <list of N tied nodes> #
```

---

Command: **link_multi_tie_n_to_n**

**Description:** This command creates multiple nodal ties from the specified tied nodes (a node path of \( N \) nodes) to the retained nodes specified by the node paths of \( N \) nodes. The number of retained node paths corresponds to the number of retained nodes for the tie type.

**Keyboard Command Sequence:**

```
link_multi_tie_n_to_n <path of N tied nodes> #
<path of N retained nodes> #
.
.
.<path of N retained nodes> #
```

---

Command: **link_multi_tie_nrnodes**

**Description:** This command sets the number of retained nodes for the user-defined ties to be created with the **link_multi_tie_n_to_1** and **link_multi_tie_n_to_n** commands.

**Keyboard Command Sequence:**

```
link_multi_tie_nrnodes <# retained nodes>
```
Command: **link_multi_tie_reset**

**Description:** This command resets the multi-nodal tie type and retained node.

**Keyboard Command Sequence:**

```
link_multi_tie_reset
```

---

Command: **link_multi_tie_rnode**

**Description:** This command sets the retained nodes for nodal tie to be created with the `link_multi_tie_n_to_1` and `link_multi_tie_n_to_n` commands. You must specify the number of the retained node (slot) you wish to set and the retained node.

**Keyboard Command Sequence:**

```
link_multi_tie_rnode <slot> <retained node>
```

**Other Buttons with the same description:**

```
NODE 2 through NODE 8
```

---

Command: **link_multi_tie_type**

**Description:** This command sets the Marc type of the links to be created with the `link_multi_tie_n_to_1` command. Tie types are specified as integers. See the *MSC.Marc Volume C, Program Input* manual for valid tie types.

**Keyboard Command Sequence:**

```
link_multi_tie_type <tie type>
```

---

Command: **link_name**

**Description:** This command sets or changes the name of the current link.

**Keyboard Command Sequence:**

```
link_name <link name>
```
Command: `link_tie_nrnodes`
Description: This command sets the number of retained nodes for the current user-defined tie.

Keyboard Command Sequence:
```
link_tie_nrnodes <# retained nodes>
```

Command: `link_value`
Description: This command sets the coefficient values for the current servo link. This command has been superseded by the `spring_param` command and is for backward compatibility only.

Valid link properties are as follows:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>coefficient_1, coefficient_10</td>
<td>Servo link retained node coefficients.</td>
</tr>
<tr>
<td>stiffness</td>
<td>Spring stiffness coefficient.</td>
</tr>
<tr>
<td>damping</td>
<td>Dashpot damping coefficient.</td>
</tr>
</tbody>
</table>

Keyboard Command Sequence:
```
link_value <property name> <value>
```

Other Buttons with the same description:
- DAMPING COEF.
- INITIAL FORCE
- STIFFNESS

Menu: LINKS
Description: This menu contains commands for defining links (multi-point constraints). The types of links that can be created are nodal ties, servo links, and springs/dashpots.
list_aliases

Command:  list_aliases
Description: This command lists all currently defined aliases. To see the value of a single alias use the exam_alias command.

Also see help: alias.

Keyboard Command Sequence:
  list_aliases

list_parameters

Command:  list_parameters
Description: This command lists all currently defined parameters. To see the value of a single parameter use the exam_parameter command.

Also see help: define.

Keyboard Command Sequence:
  list_parameters

load_camera

Command:  load_camera
Description: This command restores the camera position and settings in a view to the position and settings that were previously saved in a view file. You must specify the view whose camera is to be changed. Views are specified by number, 1 to 4. You must also specify the name of the view file.

Also see help: save_view
            load_view
            load_trans
and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
  load_camera <view> <filename>
**Command:** load_colors

**Description:** This command reads a color map previously saved to a file and restores the color table to those values. To save a color map to a file use the `save_colors` command. You must specify the name of the color map file.

**Keyboard Command Sequence:**

```plaintext
load_colors <file name>
```

**Command:** load_lighting

**Description:** This command reads a lighting model previously saved to a file and restores it into a view. To save a lighting model to a file use the `save_lighting` command. You must specify the view and the name of the file.

Also see help: `set_lighting`

```plaintext
save_lighting
reset_lighting
```

**Keyboard Command Sequence:**

```plaintext
load_lighting <view> <file name>
```

**Command:** load_procedure

**Description:** This command loads a previously created procedure file without executing it.

**Keyboard Command Sequence:**

```plaintext
load_procedure <file name>
```
**LOAD MODEL**

**Command:** load_trans

**Description:** This command restores the model viewing transformations in a view to the viewing transformations that were previously saved in a view file. You must specify the view whose transformations are to be changed. Views are specified by number, 1 to 4. You must also specify the name of the view file.

Also see help: save_view

load_view

load_camera

and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

```
load_trans <view> <filename>
```

**LOAD VIEW**

**Command:** load_view

**Description:** This command restores a view that was previously saved in a view file. The view’s camera position and settings, and model viewing transformations are restored from the file. You must specify the view to be changed. Views are specified by number, 1 to 4. You must also specify the name of the view file.

Also see help: save_view

load_camera

load_trans

and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

```
load_view <view> <filename>
```
**LOADCASES**

**Menu:** LOADCASE

**Description:** The LOADCASE menu contains commands that specify the parameters needed to perform a MSC.Marc analysis. The operations that may be performed include load selection, specification of load distribution, selection of output types, and other MSC.Marc parameters.

**REM**

**Command:** loadcase_activate_elements

**Description:** This command activates elements that have been deactivated with the loadcase_deactivate_elements command.

**Keyboard Command Sequence:**

loadcase_activate_elements <element list> #

**Command:** loadcase_add_descon

(To associate constraint with loadcase)

loadcase_remove_descon

(To dissociate constraint from loadcase)

**Description:** If button is pressed when it is not lit up, it will light up and the constraint will be associated with the current loadcase. If button was already lit up, then pressing it will turn it off and the constraint will no longer be associated with the current loadcase. Alternatively, you can give one of the commands loadcase_add_descon or loadcase_remove_descon, as appropriate, for the same results.
INITIAL FRACTION

Command: loadcase_value initfraction

Description: This command sets the initial load step size by specifying the fraction of the total load in this loadcase.

Keyboard Command Sequence:

    loadcase_value initfraction <value>

MAXIMUM FRACTION

Command: loadcase_value maxfraction

Description: This command sets the maximum fraction of the total load in this loadcase that can be applied in each increment.

If this value is set too small in an analysis with linear pre-buckling behavior, a large load step will be obtained if its size is not limited. This can cause convergence problems in the solution.

Keyboard Command Sequence:

    loadcase_value maxfraction <value>

MAX RATIO ARC LENGTH/INITIAL ARC LENGTH

Command: loadcase_value maxmultiplier

Description: This command sets the maximum ratio of the current load step to the initial load step that is allowed.

This is used to limit the maximum load step size. A typical case when this is important is when a linear part of the solution is followed by a sharp nonlinearity. Problems may occur if the load step is allowed to be too large.

Keyboard Command Sequence:

    loadcase_value maxmultiplier <value>
Command: \texttt{loadcase\_option arclength\_meth}

Description: This command is used to select the type of arc-length method to use.

In an arc-length method, the size of the load increment is limited by an extra constraint. This constraint is defined such that the \textit{length} of the load increment is equal to an arc-length. The definition of this constraint is what differs between the different methods. With this formulation, the prescribed load is allowed to both increase and decrease so that unstable solution paths can be followed. The solution to the constraint equation gives the new solution increment to apply.

CRISFIELD
This method uses a quadratic constraint equation for the load increment. This may cause problems if no real roots can be found. This can typically happen if follower forces are present and the symmetric solver is used or the force stiffness is not included (see help on follower forces in the JOBS menus). In general, nonlinearities causing sudden changes in residual forces can cause problems, in particular contact changes.

RIKS-RAMM
A linearized constraint equation is used in this method. This makes it more robust than Crisfield’s method but may give slower convergence. The Riks-Ramm method can also have problems if sudden changes occur in a contact analysis.

MODIFIED RIKS-RAMM
This is the default method. The Riks-Ramm has been modified in order to handle contact problems more efficiently.

CRISFIELD/MODIFIED RIKS-RAMM
Uses Crisfield’s quadratic equation with a switch to the modified Riks-Ramm method if no real roots are found.

It is difficult to give general recommendations on which method to use, it is very problem dependent. In contact analysis, the modified Riks-Ramm or Crisfield/modified Riks-Ramm method is usually appropriate. If no sudden changes in residual forces occur, Crisfield’s method is often found to be efficient.
Some frequent problems:

- Desired number of recycles set too high. This leads to a very large arc length and consequently very large load step.
- Maximum load step specified but no limitation on arc length. Gives a very large load step in initially linear problem.
- Maximum number of recycles set equal to desired number of recycles. Analysis will stop if the nonlinearity of the problem becomes weaker.
- Solution of nonpositive definite system not set. The analysis stops at a snap-through point.
- Convergence tolerances set too high. May lead to no real root with Crisfield’s method or sudden load reversal.
- Tying type 80 (rigid link) present in the model. Avoid using this tying type with the arc-length methods.


Keyboard Command Sequence:

<crisf/riks/mod_riks/crisf_mod_riks>

Other Buttons with the same description:

CRISFIELD/MODIFIED RIKS-RAMM
MODIFIED RIKS-RAMM
RIKS-RAMM

Command: loadcase_value minmultiplier
Description: This command sets the minimum ratio of the current load step to the initial load step that is allowed.

This is used to limit the minimum length of the load step to avoid small increments.

Keyboard Command Sequence:

loadcase_value minmultiplier <value>
**PARAMETERS**

**Menu:** PARAMETERS

**Description:** This menu contains commands for setting parameters for the arc-length method.

---

**loadcase_autoload_set**

**Command:** `loadcase_option loadcase_autoload_set`

**Description:** This command defines the set of nodes or elements for which the current criterion is to be applied.

*Note:* The nodal values have to be applied to node sets and the element values to element sets.

**Keyboard Command Sequence:**

```
loadcase_option loadcase_autoload_set
<criterion number>
```

---

**CLEAR**

**Command:** `clear_loadcase_descons`

**Description:** Pressing this button or giving the equivalent command will clear all constraints which have been associated with this loadcase. In other words, these constraints will now no longer be associated with the loadcase under consideration.

---

**loadcase_contact**

**Menu:** CONTACT

**Description:** This popup menu contains commands for setting contact properties for the current loadcase.
**AUTO SWITCH**

**Command:** `loadcase_option error_auto_switch`

**Description:** This command is used to switch on or off the feature to identify the extreme cases where the absolute value of either reaction force or displacement is close to the cut-off numbers of computer accuracy. These small values may induce unnecessary failures in detecting the converged solution if the convergence criterion is not set properly. Such cases are often seen when the analysis might includes:

- Rigid body motion : reaction forces close to zero
- Springback : small displacement increment
- Free thermal expansion : reaction forces close to zero
- Constraint thermal expansion: displacement increment is zero

ON allow to use this feature
OFF do not use this feature

Note if any kind of absolute value testing has been set as convergence tolerance, this feature will be deactivated.

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11

**Keyboard Command Sequence:**

```
loadcase_option error_auto_switch:<on/off>
```
With relative residual testing, the check is made if the maximum residual force divided by the maximum reaction force is smaller than the tolerance (corresponding for moments).

With relative displacement testing, the check is made if the largest iterative displacement correction divided by the maximum displacement increment is smaller than the tolerance (corresponding for rotations).

With strain energy testing, the change in total strain energy is used. The check is made if the strain energy of the iterative correction divided by the strain energy increment is smaller than the tolerance.

The relative residual force testing is the default and is usually appropriate. In cases where there are no reaction forces, either absolute testing, displacement testing, or the mixed RELATIVE/ABSOLUTE testing should be used. In cases where the reaction forces become very large, for instance in certain axisymmetric analyses and contact problems, either displacement testing or a smaller tolerance should be used.

**Note:** The relative displacement and strain energy testing always leads to at least one iteration, regardless of the accuracy of the solution.

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11

**Keyboard Command Sequence:**

<force_only/residuals/disp_only/
displacements/strain>

Other Buttons with the same description:

DISPLACEMENTS
INCLUDE MOMENTS
INCLUDE ROTATIONS
STRAIN ENERGY
Command:  `loadcase_option converge`

Description:  This command is used to specify the criterion to use for convergence testing on residual forces and/or displacements.

- **RESIDUAL FORCE or DISPLACEMENTS**
  Use both residuals and displacements, convergence will be obtained if one of them converged.

- **RESIDUAL FORCE and DISPLACEMENTS**
  Use both residuals and displacements, convergence will be obtained if both converged simultaneously.

With **RESIDUAL FORCE and/or DISPLACEMENTS**, both residual forces (or moments) and displacement (or rotations) testing are available.

In the residual part, either residual force and/or residual moment can be used. Both relative and absolute values can be selected. Default is set for relative residual force only.

In the displacement part, either displacement and/or rotation can be used. Both relative and absolute values can be selected. Default is set for relative displacement only.

**Note:** The relative displacement always leads to at least one iteration, regardless of the accuracy of the solution.


For more details see help text for buttons of:

- RESIDUAL FORCE
- DISPLACEMENT
- STRAIN ENERGY
- RESIDUAL FORCE & MOMENT
- DISPLACEMENT & ROTATION

**Keyboard Command Sequence:**

```
loadcase_option converge:
<force_or_disp/force_and_disp>
```

Another Button with the same description:

- RESIDUALS OR DISPLACEMENTS
MAXIMUM ABSOLUTE DISPLACEMENT

Command: `loadcase_value maxdisp`

Description: This command specifies the convergence tolerance when absolute testing on displacements is used.

If set to zero, no checking on displacements takes place.


Keyboard Command Sequence:

```
loadcase_value maxdisp <value>
```

MINIMUM DISPLACEMENT CUTOFF

Command: `loadcase_value mindisp`

Description: This command specifies a cutoff value for the displacement increment if relative testing is used. If the displacement increment is less than this value, checking will be bypassed, or, if the RELATIVE/ABSOLUTE option is used, absolute testing will be used.


Keyboard Command Sequence:

```
loadcase_value mindisp <value>
```

RELATIVE DISPLACEMENT TOLERANCE

Command: `loadcase_value displacement`

Description: This command specifies the convergence tolerance when relative testing on displacements is used.


Keyboard Command Sequence:

```
loadcase_value displacement <value>
```
**RELATIVE ENERGY TOLERANCE**

Command: `loadcase_value strain`

Description: This command specifies the convergence tolerance when relative testing on strain energy is used.


Keyboard Command Sequence:

```
loadcase_value strain <value>
```

**MAXIMUM ABSOLUTE RESIDUAL FORCE**

Command: `loadcase_value maxforce`

Description: This command specifies the convergence tolerance when absolute residual testing on forces is used.

If set to zero, no checking on residual forces takes place.


Keyboard Command Sequence:

```
loadcase_value maxforce <value>
```

**MINIMUM REACTION FORCE CUTOFF**

Command: `loadcase_value minforce`

Description: This command specifies a cutoff value for the reaction force if relative testing is used. If the reaction force is less than this value, checking will be bypassed, or if the RELATIVE/ABSOLUTE option is used, absolute testing will be used.


Keyboard Command Sequence:

```
loadcase_value minforce <value>
```
**RELATIVE FORCE TOLERANCE**

**Command:** loadcase_value force  
**Description:** This command specifies the convergence tolerance when relative residual testing on forces is used.  
**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11.  
**Keyboard Command Sequence:**  
loadcase_value force <value>

**MAXIMUM ABSOLUTE RESIDUAL MOMENT**

**Command:** loadcase_value maxmoment  
**Description:** This command specifies the convergence tolerance when absolute residual testing on moments is used.  
If set to zero, no checking on residual moments takes place.  
**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11.  
**Keyboard Command Sequence:**  
loadcase_value maxmoment <value>

**MINIMUM REACTION MOMENT CUTOFF**

**Command:** loadcase_value minmoment  
**Description:** This command specifies a cutoff value for the reaction moment if relative testing is used. If the reaction moment is less than this value, checking will be bypassed, or if the RELATIVE/ABSOLUTE option is used, absolute testing will be used.  
**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11.  
**Keyboard Command Sequence:**  
loadcase_value minmoment <value>
**RELATIVE MOMENT TOLERANCE**

**Command:**  `loadcase_value moment`

**Description:** This command specifies the convergence tolerance when relative residual testing on moments is used.

**Reference:** *MSC.Marc Volume A: Theory and User Information*, Chapter 11.

**Keyboard Command Sequence:**

```
loadcase_value moment <value>
```
Keyboard Command Sequence:

\texttt{loadcase\_option error:<>relative/absolute/relabs>}

Another Button with the same description:

RELATIVE

Command: \texttt{loadcase\_value maxrotation}

Description: This command specifies the convergence tolerance when absolute testing on rotations is used.

\hspace{1em} If set to zero, no checking on rotations takes place.


Keyboard Command Sequence:

\texttt{loadcase\_value maxrotation <value>}

Command: \texttt{loadcase\_value minrotation}

Description: This command specifies a cutoff value for the rotation increment if relative testing is used. If the rotation increment is less than this value, checking will be bypassed, or if the RELATIVE/ABSOLUTE option is used, absolute testing will be used.


Keyboard Command Sequence:

\texttt{loadcase\_value minrotation <value>
RELATIVE ROTATION TOLERANCE

Command: loadcase_value rotation
Description: This command specifies the convergence tolerance when relative testing on rotations is used.
Reference: MSC.Marc Volume A, Theory and User Information, Chapter 11.
Keyboard Command Sequence:
   loadcase_value rotation <value>

CORNERING VELOCITY

Command: loadcase_value
Menu: LOADCASE STEADY STATE ROLLING PARAMETERS
Description: This value specifies cornering velocity magnitude.
Keyboard Command Sequence:
   loadcase_value cornering_velocity <value>

CONTACT TABLE

Command: loadcase_cstable
Description: This command specifies the contact table to be used during the current loadcase.
Keyboard Command Sequence:
   loadcase_cstable <contact table name>
AUTOMATIC TIME STEP CUT BACK

Command:  **loadcase_option time_cut**

Description: This command specifies the option of automatically reducing the time step and start over from the beginning of the current increment.

This will be done when one of the following things happens during an increment and the analysis would otherwise stop if this option is not activated:

- convergence could not be obtained
- one or more elements go inside out
- a node slides off the end of a rigid body in a contact analysis

Each time the step is cut back the time step is divided by two. The cut-back is repeated if necessary until a converged solution is obtained or the maximum number of times this is allowed to be done is reached, as specified with the # CUT BACKS ALLOWED option.

With the multi-criteria adaptive stepping procedure (auto step), a cut-back is also done when a user specified desired number of recycles is exceeded or a user specified physical criterion is violated. This option should always be turned on when used with the multi-criteria adaptive scheme, otherwise the time step is not allowed to decrease.

A backup of the data needed to start the increment from the beginning again is kept in memory (or on disk if the OUT-OF-CORE ELEMENT STORAGE (elsto) option is used or if memory could not be allocated).

In a fixed time stepping procedure, subincrements are performed until an increment is completed. Thus, the user specified number of increments and step sizes are preserved.

In an adaptive procedure, the time step is modified when a cut back occurs.

**Keyboard Command Sequence:**

```
loadcase_option time_cut:<on/off>
```
# CUT BACKS ALLOWED

**Command:** `loadcase_value ntime_cuts`

**Description:** This command specifies the maximum number of times the time step is allowed to be cut back in each increment. The program will stop if more cut backs are attempted during an increment.

Note that for Adaptive Stepping (`Multi_Criteria`), this specifies the maximum number of times the time step can be reduced to satisfy any physical criteria that may be defined by the user or automatically generated by the program.

**Keyboard Command Sequence:**

```
loadcase_value ntime_cuts <value>
```

---

**CUTTER FILE**

**Menu:** CUTTER FILE

**Description:** This popup menu enables to specify the name of the cutter file.

---

**Command:** `loadcase_cutter_file`

**Description:** This entry box enables to specify the name of the cutter file.

**Keyboard Command Sequence:**

```
loadcase_cutter_file <file_name>
```

---

**ADD**

**Command:** `loadcase_deactivate_elements`

**Description:** This command deactivates elements for the current loadcase. Deactivated elements do not participate in the analysis.

**Keyboard Command Sequence:**

```
loadcase_deactivate_elements <element list> #
```
**DESIRED # RECYCLES/INCREMENT**

**Command:** `loadcase_value desired`

**Description:** This command sets the desired number of recycles (iterations) for the current loadcase.

If the number of recycles in the previous increment is larger than the desired number, the time step is scaled down in an attempt to arrive at the desired number of recycles.

**Keyboard Command Sequence:**

```
loadcase_value desired <value>
```

**INITIAL FRACTION**

**Command:** `loadcase_value initfraction`

**Description:** This command sets the initial fraction of total load in the current loadcase that is applied initially.

This load increment then can be modified based on the convergence behavior during the current and previous increments.

**Keyboard Command Sequence:**

```
loadcase_value initfraction <value>
```

**MAXIMUM FRACTION**

**Command:** `loadcase_value maxfraction`

**Description:** This command sets the maximum fraction of total load in the current loadcase that can be applied in each increment.

**Keyboard Command Sequence:**

```
loadcase_value maxfraction <value>
```
MAX LOAD STEP MULTIPLIER

Command: `loadcase_value maxmultiplier`
Description: This command sets the maximum load step multiplier. This value is used when the time step is increased. The time step of the next increment is increased when the quotient of convergence ratio and convergence tolerance in cycle one is less than 0.9.

Keyboard Command Sequence:
```
loadcase_value maxmultiplier <value>
```

MIN LOAD STEP MULTIPLIER

Command: `loadcase_value minmultiplier`
Description: This command sets the minimum load step multiplier. This value is used when the time step is decreased. The time step of the next increment is decreased when the number of recycles is larger than the desired number of recycles or if the quotient of convergence ratio and convergence tolerance in cycle one is greater than 2.

Keyboard Command Sequence:
```
loadcase_value minmultiplier <value>
```

loadcase_equil_mintime

Command: `loadcase_value mintstep`
Description: This command sets the minimum allowed time step in the current loadcase.

Keyboard Command Sequence:
```
loadcase_value mintstep <value>
```
**Parameters Menu**

**Description:** This menu contains commands for setting parameters for the equilibrium based load stepping method.

### # Steps

**Command:** `loadcase_value nsteps`  
**Description:** This command is used to define the number of steps (increments) to use in the current loadcase if a fixed time stepping procedure is used.

**Keyboard Command Sequence:**  
`loadcase_value nsteps <value>`

### Gradual Friction

**Command:** `loadcase_option`  
**Description:** If the gradual friction is turned on, friction coefficient increases gradually from 0 to the final value within the loadcase, based on the time increment. In the case only solution at the end of the loadcase is physically meaningful. This may enhance stability of the analysis, especially for the transition periods from standstill to rolling, and from brake to traction.

**Keyboard Command Sequence:**  
`loadcase_option gradual_friction <on/off>`

### Loadcase Steady State Rolling

**Command:** `loadcase_value`  
**Description:** This Value specifies ground velocity_x magnitude.

**Keyboard Command Sequence:**  
`loadcase_value ground_velocity_x <value>`
LOADCASE STEADY STATE ROLLING

Command: `loadcase_value`

Description: This Value specifies ground velocity_y magnitude.

Keyboard Command Sequence:
```
loadcase_value ground_velocity_y <value>
```

LOADCASE STEADY STATE ROLLING

Command: `loadcase_value`

Description: This Value specifies ground velocity_z magnitude.

Keyboard Command Sequence:
```
loadcase_value ground_velocity_z <value>
```

loadcase_load_phase

Command: `loadcase_load_phase`

Description: This command is used to specify the phase to be applied to a load in a harmonic analysis. You must specify the name of the load application that the phase is applied to and the phase in radians.

Keyboard Command Sequence:
```
loadcase_load_phase <application name> <phase>
```

MAX # ADJUSTMENTS

Command: `loadcase_value max_num_adjustments`

Description: This command sets the maximum number of adjustments.

Keyboard Command Sequence:
```
loadcase_value max_num_adjustments <value>
```
Command: loadcase_value maxinc
Description: This command sets the maximum number of increments that is allowed in the current loadcase.

Keyboard Command Sequence:
loadcase_value maxinc <value>

Another Button with the same description:
MAX # INCREMENTS IN LOADCASE

Command: loadcase_option automatic_crit
Description: This command sets the option for automatic physical criteria.

If the option is active, the analysis will augment the user’s physical criteria if appropriate. Automatic criteria are only added if there are no competing user criteria.

At present, 4 physical criteria are added automatically by the solver for mechanical / coupled analysis:
– Total strain increment for any large displacement analysis < 50%
– Plastic strain increment for large strain, large displacement analysis < 10%
– For explicit creep analysis,
  • normalized creep strain increment = 0.5
  • normalized stress increment = 0.5
– Stress increment associated with temperature changes < 50% of total stress

Keyboard Command Sequence:
loadcase_option automatic_crit:<on,off>
Command:  loadcase_option  
multi_crit_<n>

Description: These commands are used to select the criteria to use for controlling the time step. These buttons are available under the UNDER DEFINED CRITERIA menu associated with the PARAMETERS button for MULTI-CRITERIA.

The time step is reduced during the equilibrium iterations if the increment in any of the selected quantity is larger than the user defined allowed increment. Note that <n> is a value from 1 to 4.

If the TREAT CRITERIA AS TARGETS option is used, the time step for the next increment will be increased to try to reach the value of the least critical criterion.

The factors with which the time step can be changed are limited by the minimum and maximum step ratios.

The limits and ranges for which the limits are valid are defined under the respective PARAMETER menu of each criterion.

Keyboard Command Sequence:

loadcase_option  
multi_crit_<n>:<strain/pl_strain etc.>

Other Buttons with the same description:

CREEP STRAIN INCREMENT
DISPLACEMENT INCREMENT
NORMALIZED CREEP STRAIN INCREMENT
PLASTIC STRAIN INCREMENT
ROTATION INCREMENT
STRAIN ENERGY INCREMENT
STRESS INCREMENT
TEMPERATURE INCREMENT
Mentat Help Commands in L

**Command:** `loadcase_value mc_dely1_<n>`

**Description:** This group of buttons are used for setting the maximum allowed increment of the current criterion.

The group of buttons have the text 1,2,3,4, and are located in the USER-DEFINED CRITERIA/STRAIN INCREMENT/PARAMETER menu. Note that `<n>` is a value from 1 to 4.

Other groups of buttons with the same description:

- PLASTIC STRAIN ALLOWED
- CREEP STRAIN ALLOWED
- STRESS INCR. ALLOWED
- NORMALIZED STRESS ALLOWED
- STRAIN ENERGY ALLOWED
- STATE VARIABLE ALLOWED
- DISPLACEMENT ALLOWED
- ROTATION INCR. ALLOWED
- TEMPERATURE ALLOWED

**Menu:** PARAMETER

**Description:** This menu contains commands for defining the maximum value of the increment for each criterion.

Both the limits and the ranges for which the limits are valid is defined.

**Command:** `loadcase_value mc_xmax1_<n>`

**Description:** This group of buttons are used for setting the lower bound for which the limits for the current criterion is to be applied.

Thus, it is possible to have different tolerances for different values of the criterion used.
The group of buttons have the text 1,2,3,4, and are located in the PARAMETER menu under USER-DEFINED CRITERIA. Note that <n> is a value from 1 to 4.

Other groups of buttons with the same description:

- PLASTIC STRAIN RANGE LOWER BOUND
- CREEP STRAIN RANGE LOWER BOUND
- STRESS RANGE LOWER BOUND
- NORMALIZED STRESS RANGE LOWER BOUND
- STRAIN ENERGY RANGE LOWER BOUND
- STATE VARIABLE RANGE LOWER BOUND
- DISPLACEMENT RANGE LOWER BOUND
- ROTATION RANGE LOWER BOUND
- TEMPERATURE RANGE LOWER BOUND

**Command:** loadcase_value mc_xmax2_<n>

**Description:** This group of buttons are used for setting the lower bound for which the limits for the current criterion is to be applied.

Thus, it is possible to have different tolerances for different values of the criterion used.

The group of buttons have the text 1,2,3,4, and are located in the PARAMETER menu under USER-DEFINED CRITERIA. Note that <n> is a value from 1 to 4.

Other groups of buttons with the same description:

- PLASTIC STRAIN RANGE LOWER BOUND
- CREEP STRAIN RANGE LOWER BOUND
- STRESS RANGE LOWER BOUND
- NORMALIZED STRESS RANGE LOWER BOUND
- STRAIN ENERGY RANGE LOWER BOUND
- STATE VARIABLE RANGE LOWER BOUND
- DISPLACEMENT RANGE LOWER BOUND
- ROTATION RANGE LOWER BOUND
- TEMPERATURE RANGE LOWER BOUND
Command:  `loadcase_value desired`

Description: This command sets the desired number of recycles per increment in the current loadcase.

If the number of recycles in the previous increment is not equal to the desired number of recycles, the time step is modified.

The maximum allowed change in time step between increments is defined by the buttons `MINIMUM STEP RATIO` and `MAXIMUM STEP RATIO`.

Keyboard Command Sequence:

```
loadcase_value desired <value>
```
Command: **loadcase_value inittime**

Description: This command sets the initial time step for the current loadcase.

A value of zero will allow MSC.Marc to default to 1% of the time period for this loadcase.

Keyboard Command Sequence:

```
loadcase_value inittime <value>
```

---

Command: **loadcase_value initfraction**

Description: This command defines the initial time step for the current loadcase. The initial time step is calculated as the factor given with this option times the total time for the current loadcase.

Keyboard Command Sequence:

```
loadcase_value initfraction <value>
```

---

Command: **loadcase_option multi_criteria_mode**

Description: This command defines the way the time step changes within an increment and for the next increment are treated depending on the characteristics of the physical criteria.

**TREAT CRITERIA AS LIMITS:**
The time step is reduced in an increment if a physical criterion is not satisfied, i.e. the criteria simply act as limits to the time step of the current increment.

**TREAT CRITERIA AS TARGETS:**
In addition to reducing the time step within an increment if necessary, the time step is increased for the next increment in an attempt to reach the value of the least critical criterion, i.e. the criteria act as targets to the time step of the current and next increment.
The TARGETS option has another important use: in dynamic problems with the implicit Newmark-Beta or SSH operators, this option can be set if one does not want the auto step algorithm to check the time step for integration errors. This may be useful for high-noise/contact dynamic problems.

**Keyboard Command Sequence:**

```
loadcase_option
multi_criteria_mode:<limit/target>
```

---

**MAXIMUM FRACTION OF LOADCASE**

**Command:** `loadcase_value maxfraction`

**Description:**
This command defines the maximum time step for the current loadcase. The maximum time step is calculated as the factor given with this option times the total time for the current loadcase.

The maximum time step puts a limit on the size of the time step. It can be used to avoid very large time steps.

**Keyboard Command Sequence:**

```
loadcase_value maxfraction <value>
```

---

**MAXIMUM # STEPS**

**Command:** `loadcase_value maxnsteps`

**Description:**
This command sets the maximum number of steps (increments) that is allowed during this loadcase.

If set to zero, a value of 10 times the inverse of the initial fraction of loadcase time will be used.

**Keyboard Command Sequence:**

```
loadcase_value maxnsteps <value>
```
MAXIMUM STEP RATIO

Command: `loadcase_value max_step_ratio`

Description: This command sets the largest factor with which the time step is allowed to increase in the next increment due to a physical criterion. This factor is only significant when the TREAT CRITERIA AS TARGETS option is used.

Keyboard Command Sequence:

`loadcase_value max_step_ratio <value>`

MINIMUM FRACTION OF LOADCASE TIME

Command: `loadcase_value minfraction`

Description: This command defines the minimum time step for the current loadcase. The minimum time step is calculated as the factor given with this option times the total time for the current loadcase.

If the time step is decreased to below the minimum time step the program will stop, unless the quasi-static inertial damping option is used.

Keyboard Command Sequence:

`loadcase_value minfraction <value>`
MINIMUM STEP RATIO

Command: `loadcase_value min_step_ratio`

Description: This command sets the largest factor with which the time step is allowed to decrease within an increment due to a physical criterion.

Keyboard Command Sequence:
```
loadcase_value min_step_ratio <value>
```

loadcase_multi_mintime

Command: `loadcase_value mintstep`

Description: This command sets the minimum allowed time step. Defaults to total time divided by the maximum number of steps.

Keyboard Command Sequence:
```
loadcase_value mintstep <value>
```

PARAMETERS

Menu: PARAMETERS

Description: This menu contains commands for setting parameters for the multi-criteria load stepping method.

PROCEED WHEN NOT SATISFIED

Command: `loadcase_option proceed_crit`

Description: This command sets the option for continuation of the analysis when the physical criteria (either user-defined or program-defined) are not satisfied. When this option is active, if the physical criteria are not satisfied even after the minimum time step has been reached or the maximum number of cutbacks for satisfying the physical criteria has been reached, then the analysis continues.

Keyboard Command Sequence:
```
loadcase_option proceed_crit:<on,off>
```
**TIME STEP SCALE FACTOR**

**Command:**  `loadcase_value scale_step`

**Description:** This command defines the scale factor that is used for scaling the time step based upon the desired number of recycles.

**Description:** If the desired number of recycles is exceeded in an increment, the time step is decreased using a cutback and the increment is repeated. The cutback factor should be less than or equal to (1.0/scale factor). As the number of cutbacks increase in an increment, an automatic algorithm is used to reduce the time step more aggressively such that the minimum allowed time step is reached within a reasonable number of cutbacks.

If the number of recycles needed in an increment is less than the desired number, the time step size for the next increment is multiplied by this factor.

**Keyboard Command Sequence:**

`loadcase_value scale_step <value>`

---

**USER-DEFINED CRITERIA**

**Command:**  `loadcase_option multi_crit_n`

(n = criterion no. from 1 to 12)

**Description:** This button enables access to user defined physical criteria. Both element based criteria (stresses, strains, etc.) and nodal based criteria (displacements, rotations, etc.) can be specified. The criteria can be specified for the entire set of finite element entities or for specific sets of entities.

These criteria can be augmented by automatic physical criteria that the program adds at run-time.

**Keyboard Command Sequence:**

`loadcase_option multi_crit_n: <on/off>`
**NAME**

**Command:**  `loadcase_name`

**Description:** This command sets or changes the name of the current loadcase.

A loadcase is a collection of boundary conditions, analysis control parameters, and options that are to be used in a portion of a MSC.Marc analysis. Multiple loadcases may be defined and are stored in the list of currently defined loadcases.

**Keyboard Command Sequence:**

```
loadcase_name <loadcase name>
```

---

**RESIDUAL FORCE**

**Command:**  `loadcase_option`

**Description:** This command specifies an option value for the current loadcase. You must specify the option name and its value separated by a colon (e.g. `converge:displacements`).

The available loadcase options and their values are as follows:

- **stepping**
  - fixed,
  - arclength,
  - time Mech,
  - time heat,
  - thermal,
  - equilibrium
  - Stepping procedure (fixed, arc length [auto increment], time [auto time or auto creep], time [transient], thermal [auto therm or auto therm creep], equilibrium [auto time]).

- **creeptest**
  - relative,
  - Auto creep type of testing, absolute

- **nonpos**
  - off, on
  - If on, forces the solution of non-positive definite system of equations.

- **sturm**
  - off, on
  - If on, perform Sturm sequence check on extracted eigenvalues.

- **vectors**
  - no,
  - write,
  - read,
  - both
  - In modal analyses, controls the writing of Lanczos vectors to the restart file.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>error</strong></td>
<td>relative, absolute</td>
</tr>
<tr>
<td><strong>converge</strong></td>
<td>residuals, displacements, strain</td>
</tr>
<tr>
<td><strong>proceed</strong></td>
<td>off, on</td>
</tr>
<tr>
<td><strong>procedure</strong></td>
<td>fullnr, modifiednr, straincorr</td>
</tr>
<tr>
<td><strong>initstress</strong></td>
<td>full, deviatoric, none, begininc</td>
</tr>
<tr>
<td><strong>finish</strong></td>
<td>no, below, exceed</td>
</tr>
<tr>
<td><strong>autoshift</strong></td>
<td>off, on</td>
</tr>
<tr>
<td><strong>damping_comp</strong></td>
<td>off, on</td>
</tr>
<tr>
<td><strong>stiffness_comp</strong></td>
<td>off, on</td>
</tr>
<tr>
<td><strong>frequency</strong></td>
<td>range, number</td>
</tr>
<tr>
<td><strong>freq_div</strong></td>
<td>linear, log</td>
</tr>
</tbody>
</table>

**Keyboard Command Sequence:**

```
loadcase_option <option name>:<value>
```

Other Buttons with the same description:

- ABSOLUTE
- AUTO SHIFT
BEGIN INCREMENT STRESS
CONSTANT
DAMPING
DEVIATORIC STRESS
DIRECT SUBSTITUTION
EQUILIBRIUM
FULL
FULL NEWTON-RAPHSON
MODIFIED NEWTON-RAPHSON
N-R WITH STRAIN CORRECTION
NON-POSITIVE DEFINITE
NONE
NUMBER
OFF
ON
PROCEED WHEN NOT CONVERGED
RAMP
RANGE
RELATIVE
STIFFNESS
STURM SEQUENCE
TENSILE STRESS
VELOCITY

**ASSEMBLY EACH ITERATION**

**Command:** `loadcase_option operator_assembly`

**Description:** This command will enable the option to assemble the stiffness matrix for each iteration when it is turned on (option iteration). When off, it uses option automatic.

Note that this switches off the Modified Newton-Raphson option in the ITERATIVE PROCEDURE menu when this option is turned on (iteration).

This button is available in the LOADCASES->MECHANICAL->STATIC->SOLUTION CONTROL and other loadcase class menu.

**Keyboard Command Sequence:**

```
loadcase_option operator_assembly:
   <iteration|automatic>
```
Command: loadcase_option post
Description: This command will allow for altering the increment interval at which data is written to the post file.

UNCHANGED: The post data for this loadcase will be written at the same interval as the previous loadcase.

INCREMENT FREQUENCY: The post data will be written to the post file at the interval specified using the SET button.

NONE: No post data will be written for this loadcase.

This option is available in the LOADCASES->MECHANICAL->STATIC->LOADCASE RESULTS menu.

Keyboard Command Sequence:
loadcase_option post:<unchanged|freq|none>

Other Buttons with the same description:
INCREMENT FREQUENCY
SET
NONE

Command: loadcase_option post_file_method
Description: This command sets the option for post file output frequency.

RESULTS AT FIXED TIME INTERVALS

If this option is active, results will be written to the post file at equally spaced intervals in time. Specify the number of states with the # POST FILE STATES button. The time step will be temporarily adjusted to exactly reach the times for post output.

If the option is not active, the increment based output frequency defined in the JOBS menu is used.
Keyboard Command Sequence:

```
loadcase_option
post_file_method:<time/increment>
```

Command: **loadcase_value num_post**

**Description:** This command sets the number of states that are written to the post file if the option RESULTS AT FIXED TIME INTERVALS is active.

Keyboard Command Sequence:

```
loadcase_value num_post <value>
```

Command: **loadcase_option contact_release**

**Description:** This command defines the way the nodal forces are removed in the contact body release option.

- **IMMEDIATE** — the forces are removed at the start of the increment.
- **GRADUAL** — the forces are removed gradually.

Keyboard Command Sequence:

```
loadcase_option contact_release:<immediate/gradual>
```

Another Button with the same description:

**IMMEDIATE**
Command:  **loadcase_reset**

**Description:** This command resets loadcase parameters and options to their default values.

**Keyboard Command Sequence:**

```
loadcase_reset
```

---

**BEGIN INCREMENT STRESS**

Command:  **loadcase_option initstress**

**Description:** This option controls the contribution of the initial stress matrix (geometric stiffness matrix) to the tangent stiffness matrix.

By default, the (stress dependent) initial stress matrix gives a full contribution to the tangent stiffness matrix. In certain cases, it can be advantageous not to have a full contribution. This influences the convergence behavior of the solution, but not the accuracy of the converged value.

The options are:

- **FULL**
  - Full contribution of initial stress matrix.
- **NONE**
  - No contribution of initial stress matrix.
- **DEViatoric STRESS**
  - Only deviatoric stresses contribute to the initial stress matrix. This is often useful in rubber analysis, when the volumetric stresses are large. The contribution of volumetric stresses can be varied linearly with use of a factor $\ell_t$. For $\ell_t$ equal to one, it gives no contribution of volumetric stresses (default), and $\ell_t$ equal to zero gives full contribution. The factor $\ell_t$ can be set in the NUMERICAL PREFERENCES menu in the different loadcase menus. It is called MEAN STRESS SUBTRACTION FACTOR for INIT. STRESS.
BEGIN INCREMENT STRESS  The stresses at the beginning of the increment are used for the initial stress matrix and not the stresses at the last iteration.

TENSILE STRESS  Only tensile stresses contribute to the initial stress matrix.

Reference:  MSC.Marc Volume C: Program Input, Chapter 3.

Keyboard Command Sequence:

loadcase_option initstress:<none/full/deviatoric/begininc>

Other Buttons with the same description:

DEVIA TORIC STRESS
FULL
NONE
TENSILE STRESS

Command:  loadcase_option procedure

Description:  This command is used to choose the solution procedure to use in the equilibrium iterations in a nonlinear static analysis.

The FULL NEWTON-RAPHSON method is the default procedure. The tangent stiffness matrix is reformulated for each iteration (recycle).

In the MODIFIED NEWTON-RAPHSON method, the tangent stiffness matrix is formulated only once per increment. This leads to lower cost, but possibly slower convergence.

NEWTON-RAPHSON WITH STRAIN CORRECTION is a variant of the full Newton-Raphson method. It uses a special formulation for how the strains enter the tangent stiffness matrix and load vector and is appropriate for shell and beam problems in which rotations are large, but membrane stresses are small. The use of this method can lead to a more accurate solution for a given load step size.

The secant method is also available in Marc, but not through Mentat. To use this method, edit the MSC.Marc input file. See MSC.Marc Volume C, Chapter 3: page 3-143.
The full Newton-Raphson method can be expensive for large, three-dimensional problems when a direct solver is used. The modified Newton-Raphson method is effective for large, mildly nonlinear problems. When an iterative solver is used, simple back substitution is not possible which makes this method ineffective. In such cases, the full Newton-Raphson method should be used instead.


Keyboard Command Sequence:

```
loadcase_option procedure:<fullnr/modifiednr/
straincorr>
```

Other Buttons with the same description:

- MODIFIED NEWTON-RAPHSON
- N-R WITH STRAIN CORRECTION

Command:  `loadcase_value maxjobinc`

Description: This command is used to set the maximum number of increments that are allowed during this job. The job stops if this number is reached. The default number used by MSC.Marc Mentat is 99999.

This is usually used to stop the run when restart is being used. This option is ignored if the ELASTIC option is being used (set in the JOBS menu). Then all loadcases are analyzed.

**Note:** The number of increments can be very large in a dynamic analysis, in particular with the explicit operator.

Keyboard Command Sequence:

```
loadcase_value maxjobinc <value>
```
Command: `loadcase_value maxrec`

Description: This command sets the maximum number of recycles (iterations) that is allowed during the equilibrium iteration in an increment of a nonlinear analysis. The job stops if this number of recycles is reached.

The default in MSC.Marc Mentat is 10. This number must sometimes be increased for problems which show slow convergence, in particular a rigid-plastic flow analysis. If a large number of recycles are needed, it is usually better to use smaller load increments or use automatic load stepping.

Keyboard Command Sequence:

```
loadcase_value maxrec <value>
```

Command: `loadcase_value minrec`

Description: This command sets the minimum number of recycles (iterations) that is performed during the equilibrium iteration in an increment of a nonlinear analysis.

The default is 0. This is used to force a given number of recycles to take place in all increments.

Keyboard Command Sequence:

```
loadcase_value minrec <value>
```
**NON-POSITIVE DEFINITE**

**Command:** `loadcase_option nonpos`  
**Description:** If this option is turned on, the solution of a nonpositive definite system is forced.  
It is only recommended to use this option when automatic load stepping with the arc-length methods is used in post-buckling analysis. It should not be used in a static analysis to remove a rigid body motion.  

**Keyboard Command Sequence:**  
`loadcase_option nonpos:<on/off>`

**PROCEED WHEN NOT CONVERGED**

**Command:** `loadcase_option proceed`  
**Description:** This option is used to specify that the program should proceed with the next increment even if convergence is not obtained within the current increment.  
If the maximum number of recycles is reached without convergence, a warning is given and the analysis is continued.  
This option is in general not recommended.  

**Keyboard Command Sequence:**  
`loadcase_option proceed:<on/off>`

**LOADCASE STEADY STATE ROLLING**

**Command:** `loadcase_value`  
**Description:** This value specifies spinning velocity magnitude.  

**Keyboard Command Sequence:**  
`loadcase_value spinning_velocity <value>`
LOADCASE STEADY STATE ROLLING

Command: `loadcase_option`
Description: This option is used to set type of input of steady state rolling parameters. If `spin_velocity` is selected, spinning velocity controls the analysis. If torque is selected, torque controls the analysis.

Keyboard Command Sequence:
```
loadcase_option ss_rolling_input:<spin_velocity/torque>
```

---

LOADCASE STEADY STATE ROLLING

Command: `loadcase_value`
Description: This command sets the tolerance. It is considered converged if the ratio of torque change at current iteration over the maximum torque is less than the tolerance.

Keyboard Command Sequence:
```
loadcase_value ss_rolling_tolerance <value>
```

---

ARC LENGTH

Command: `loadcase_option stepping:arclength`
Description: This command is used to specify that the arc-length based load stepping scheme is to be used.

This procedure uses a so-called `arc-length` method for controlling the load step. It is capable of handling unstable deformation such as snap-through and collapse analysis.

In an arc-length method, the size of the load increment is limited by an extra constraint. This constraint is defined such that the `length` of the load increment is equal to an arc-length. The definition of this arc-length is what differs between the different methods. With this formulation, the prescribed load is allowed to both increase and decrease so that unstable
solution paths can be followed. The solution to the constraint equation gives the new solution increment to apply.

The arc-length (and thus the step size) is initially set as a fraction of the total load during the loadcase. It can change in size based on the number of recycles needed in a previous increment compared with a user-defined desired number of recycles. The size of the load step can also be limited in order to avoid problems when the degree of nonlinearity in the problem changes.

Corresponds to MSC.Marc command auto increment.

Keyboard Command Sequence:

```
loadcase_option stepping:arclength
```

---

**QUASI-STATIC INERTIAL DAMPING**

Command: **loadcase_stepping_damp_it**

Description: This toggle when turned on will provide inertial damping for static contact problems. Otherwise, no inertial damping is used which is the default.

The damping becomes active only if the minimum time step is reached and is turned off again if the time step is increased above the minimum time step.

Keyboard Command Sequence:

```
loadcase_stepping_damp_it
loadcase_stepping_damp_it:off
```
**EQUILIBRIUM**

**Command:** `loadcase_option stepping:equilibrium`

**Description:** This command is used to specify that the auto time procedure is to be used.

In this procedure, the load step is determined based upon the convergence behavior in the previous and current increments.

The more general MULTI-CRITERIA procedure is recommended instead of this one.

Corresponds to MSC.Marc command `autotime`.

**Keyboard Command Sequence:**

```
loadcase_option stepping:equilibrium
```
**MULTI-CRITERIA**

Command: `loadcase_option stepping:multicriteria`

Description: This command is used to specify that the general adaptive multi-criteria load stepping scheme is to be used.

The load step can be adjusted based upon the number of recycles needed to reach equilibrium. There is also an option to control the time step based upon a user specified allowed increment of stress, strain, or other criteria.

Corresponds to MSC.Marc command `autostep`.

Keyboard Command Sequence:

```
loadcase_option stepping:multicriteria
```

**TEMPERATURE**

Command: `loadcase_option stepping:thermal`

Description: This command is used to specify that the temperature driven load stepping scheme is to be used.

This procedure is used in a thermally loaded mechanical analysis where the thermal load is obtain from a previous thermal analysis. The temperatures are read from the post file of the previous analysis.

Corresponds to MSC.Marc command `autotherm`.

Keyboard Command Sequence:

```
loadcase_option stepping:thermal
```
**TEMPERATURE**

Command:   **loadcase_option stepping:time_heat**

Description:  This command is used to specify that the temperature based load stepping scheme is to be used.

This procedure is used in a thermal or coupled analysis where the temperature is used to control the time step.

The target temperature increment is specified as MAX TEMPERATURE CHANGE ALLOWED in the CONVERGENCE TESTING menu.

Corresponds to MSC.Marc command transient.

The more general MULTI-CRITERIA option covers this option and also allows more control over the way the time step is changed.

Keyboard Command Sequence:

   loadcase_option stepping:time_heat

---

**loadcase_temp_finish**

Command:   **loadcase_option finish**

Description:  This option is used to stop a transient analysis involving a thermal part if all nodal temperatures are below or exceed the specified finish temperature.

Keyboard Command Sequence:

   loadcase_option finish:<no/below/exceed>

---

**FINISH TEMPERATURE**

Command:   **loadcase_value temp_finish**

Description:  This command sets the temperature value used for the finish check option.

Keyboard Command Sequence:

   loadcase_value temp_finish <value>
# INCS BETWEEN REASSEMBLY

**Command:** loadcase_value inc_assemble  
**Description:** This command sets the reassembly interval for element matrices. This influences the speed of convergence, but not the time step.  
**Keyboard Command Sequence:** 

```
loadcase_value inc_assemble <value>
```

---

MAX TEMPERATURE CHANGE

**Command:** loadcase_value temp_change  
**Description:** This command sets the maximum temperature change that is allowed in each increment. If the applied temperature is larger than this value, the time increment is subdivided assuming a linear change in temperature during the increment. The choice of the maximum temperature change should be based on accuracy of nonlinear effects like plasticity, creep, or geometry changes.  
**Keyboard Command Sequence:** 

```
loadcase_value temp_change <value>
```

---

MAXIMUM TIME STEP

**Command:** loadcase_value maxtime  
**Description:** This command sets the maximum time step that is allowed. The temperature increment is scaled such that the time step is smaller than this value.  
**Keyboard Command Sequence:** 

```
loadcase_value maxtime <value>
```
**PARAMETERS**

**Menu:** PARAMETERS

**Description:** This menu contains commands for setting parameters for the *auto therm* load stepping method.

---

**OK**

**Command:** `loadcase_title`

**Description:** This command sets the title for the current loadcase. The title will appear in the MSC.Marc output file and in the results plots.

**Keyboard Command Sequence:**

`loadcase_title <title>`

---

**TORQUE**

**Command:** `loadcase_value`

**Description:** This *value* specifies torque magnitude.

**Keyboard Command Sequence:**

`loadcase_value <value>`

---

**TOTAL LOADCASE TIME**

**Command:** `loadcase_value time`

**Description:** This command is used to enter the total time period associated with the current loadcase.

If temperatures are read from a post file, this value should be the same as the total thermal time period.

**Keyboard Command Sequence:**

`loadcase_value time <value>`
Command: **loadcase_type**

**Description:** This command sets the type of the current loadcase to the specified type. The type of loadcase that may be used depends on the type of analysis desired.

The valid loadcase types are given below:

<table>
<thead>
<tr>
<th>Analysis Class</th>
<th>Valid Loadcase Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>mechanical</td>
<td>static, buckle, creep, dynamic_modal, dynamic_transient, dynamic_harmonic, spectrum, rpflow, contact_approach</td>
</tr>
<tr>
<td>heat</td>
<td>steady_state, transient_heat</td>
</tr>
<tr>
<td>coupled</td>
<td>coupled, coupled_creep, coupled_dyn_transient, coupled_rpflow, coupled_cont_approach</td>
</tr>
<tr>
<td>joule</td>
<td>joule_steady_state, joule_transient</td>
</tr>
<tr>
<td>fluid</td>
<td>fluid_steady_state, fluid_transient</td>
</tr>
<tr>
<td>fluid_thermal</td>
<td>fluidtherm_steady, fluidtherm_transient</td>
</tr>
<tr>
<td>fluid_solid</td>
<td>fluidsolid_steady, fluidsolid_transient</td>
</tr>
<tr>
<td>fluid_therm_solid</td>
<td>fluidthermsolid_steady, fluidthermsolid_transient</td>
</tr>
<tr>
<td>acoustic</td>
<td>acoustic_modal, acoustic_transient</td>
</tr>
<tr>
<td>bearing</td>
<td>bearing</td>
</tr>
<tr>
<td>electrostatic</td>
<td>electrostatic</td>
</tr>
<tr>
<td>magnetostatic</td>
<td>magnetostatic</td>
</tr>
<tr>
<td>electromagnetic</td>
<td>electromagnetic_harmonic, electromagnetic_transient</td>
</tr>
</tbody>
</table>

**Keyboard Command Sequence:**

`loadcase_type <loadcase type>`

Other Buttons with the same description:

- BODY APPROACH
- DYNAMIC HARMONIC
- DYNAMIC MODAL
- DYNAMIC TRANSIENT
HARMONIC
MODAL
QUASI-STATIC
SPECTRUM RESPONSE
STEADY STATE
TRANSIENT

# FREQUENCIES

Command: **loadcase_value**

Description: This command sets loadcase parameter values. You must specify the parameter to be set and its value.

Valid loadcase parameters are as follows:

- abscurrent,
- angle1,
- angle4,
- beta,
- cooling_time,
- cutoff_stress,
- displacement,
- dt_error,
- fluid_incomp_pen,
- fluid_minforce,
- force,
- inc_assemble,
- initshift,
- limit_angle_3d,
- maxdisp,
- maxinc,
- maxmoment,
- maxrec,
- maxtime,
- mc_dely1_3,
- mc_dely1_7,
- mc_dely1_10,
- mc_dely2_3,
- mc_dely2_7,
- mc_dely2_10,
- mc_dely3_3,
- adapg_frequency,
- angle2,
- angle5,
- buckle_tol,
- crank_length,
- cycle_time,
- dt_allowed,
- eccentricity,
- fluid_maxforce,
- fluid_minvelo,
- gamma,
- inc_strn_pred_mp,
- initime,
- low,
- maxforce,
- maxit,
- maxmultiplier,
- maxrotation,
- mc_dely1_1,
- mc_dely1_4,
- mc_dely1_8,
- mc_dely2_1,
- mc_dely2_4,
- mc_dely2_8,
- mc_dely3_1,
- mc_dely3_4,
- angle0,
- angle3,
- bcond_pen_mp,
- contribution,
- creptime,
- desired,
- dt_assemble,
- fluid_force,
- fluid_maxvelo,
- fluid_velo,
- high,
- initfraction,
- limit_angle_2d,
- max_step_ratio,
- maxfraction,
- maxjobinc,
- maxnsteps,
- maxsubinc,
- mc_dely1_2,
- mc_dely1_5,
- mc_dely1_9,
- mc_dely2_2,
- mc_dely2_5,
- mc_dely2_9,
- mc_dely3_2,
- mc_dely3_5,
Keyboard Command Sequence:

```
loadcase_value <parameter label> <value>
```

Other Buttons with the same description:

- # INCREMENTS
- # INCS BETWEEN REASSEMBLY
- # MODES
- # MODES PER SHIFT
- 2-D CONTACT LIMIT ANGLE
- 3-D CONTACT LIMIT ANGLE
ABSOLUTE
BOUNDARY CONDITION PENALTY MUL
CUTOFF STRESS TOLERANCE
DESIRED # RECYCLES
DYNAMIC CONTACT PROJECTION FAC
FLUID INCOMPRESSIBILITY PENALT
FRACTION OF NODES IN CONTACT
HIGHEST FREQUENCY
INCREMENTAL STRAIN PREDICTION
INITIAL FRACTION
INITIAL SHIFT
INITIAL TIME STEP
LOWEST FREQUENCY
MAX # INCREMENTS
MAX # INCREMENTS IN JOB
MAX # ITERATIONS
MAX # RECYCLES
MAX # SUBINCREMENTS
MAX ERROR IN TEMPERATURE ESTIM
MAX LOAD STEP MULTIPLIER
MAX TEMPERATURE CHANGE
MAX TEMPERATURE CHANGE ALLOWED
MAX TEMPERATURE CHANGE BEFORE
MAX TIME STEP MULTIPLIER
MAXIMUM
MAXIMUM ABSOLUTE RESIDUAL FORC
MAXIMUM ABSOLUTE VELOCITY
MAXIMUM FRACTION
MAXIMUM TIME STEP
MEAN STRESS SUBTRACTION FACTOR
MIN # RECYCLES
MIN LOAD STEP MULTIPLIER
MIN TIME STEP
MIN TIME STEP MULTIPLIER
MINIMUM
MINIMUM REACTION FORCE CUTOFF
MINIMUM VELOCITY CUTOFF
NEWMARK-BETA BETA PARAMETER
NEWMARK-BETA GAMMA PARAMETER
PRE-STRESS
RELATIVE
RELATIVE FORCE TOLERANCE
RELATIVE VELOCITY TOLERANCE
RIGID PLASTIC CUTOFF STRAIN RA
RIGID PLASTIC INCOMPRESSIBILIT
RIGID PLASTIC INIT. STRAIN RAT
SHELL ELEMENT DRILLING MODE FA
SHIFT PARAMETER
SINGLE-STEP HOUBOLT GAMMA PARA
SINGLE-STEP HOUBOLT GAMMA1 PAR
STRAIN CHANGE TOLERANCE
STRAIN SAMPLING CUTOFF
STRESS CHANGE TOLERANCE
TARGET STRAIN RATE
TOLERANCE
TOTAL CREEP TIME
X DISPLACEMENT WEIGHT FACTOR
LOCAL ADAPTIVITY

Menu: LOCAL ADAPTIVITY

Description: The adaptive mesh generation capability increases the number of elements and nodes to improve the accuracy of the solution. The capability is applicable for both linear elastic analysis and for nonlinear analysis. The capability can be used for lower-order elements, 3-node triangular, 4-node quadrilateral, 4-node tetrahedral, and 8-node hexahedral elements.

This menu defines the types of adaptivity and related parameters.

Currently supported adapt types include the following:

- mean_strn: Mean Strain Energy
- z_z_strs: Zienkiewicz-Zhu Stress
- z_z_strn: Zienkiewicz-Zhu Strain
- z_z_pstrn: Zienkiewicz-Zhu Plastic Strain
- z_z_cpstrn: Zienkiewicz-Zhu Creep Strain
- box: Nodes Within Box
- node_cont: Nodes in Contact
- max_grad: Maximum Solution Gradient
- eq_strs_rel: Equivalent Relative Stress
- eq_strs_abs: Equivalent Absolute Stress
- eq_strn_rel: Equivalent Relative Strain
- eq_strn_abs: Equivalent Absolute Strain
- eq_pstrn_rel: Equivalent Relative Plastic Strain
- eq_pstrn_abs: Equivalent Absolute Plastic Strain
- eq_cpstrn_rel: Equivalent Relative Creep Strain
- eq_cpstrn_abs: Equivalent Absolute Creep Strain
- usersub: User Defined via User Subroutine UADAP
Mentat Help Commands in M

**MAIN Menu:**

**Description:** This menu is the top menu in MSC.Marc Mentat.

**MAP TRIMMING CURVES**

**Command:** map_curves

**Description:** This command is the reverse operation to unmap_curves. It maps a list of curves onto a chosen surface.

**Keyboard Command Sequence:**

map_curves <surface> <curve_list>
MAPPING MESH CURVE DIVISIONS

Command:  map_msh_crv_divisions

Description:  This command modifies the placement of curve divisions on surface trimming curves under specific conditions such that when the surface is meshed, a mapped mesh with regularly-shaped elements results.

Two simple situations are considered: a rectangular-shaped surface and a ring-shaped surface. When one of these shapes is identified, an attempt is made to match the curve divisions on facing sides or loops. However, this operation only modifies curve divisions on any given curve by an even number, such that any even/odd restrictions are not affected.

Also see help: apply_curve_divisions

Keyboard Command Sequence:

map_msh_crv_divisions <curve list>

MATERIAL PROPERTIES

Menu:  MATERIAL PROPERTIES

Description:  The MATERIAL PROPERTIES menu contains commands that specify material property values to be used in the mesh. Material properties may be specified for the following analysis types: mechanical, heat transfer, acoustic, bearing, Joule heating, electrostatic, magnetostatic, and electromagnetic.
MATCH CURVE DIVISIONS

Command:  match_curves

Description:  This command assures that the number of mesh divisions on coincident curves are equal and the divisions are aligned. Tolerance is used to determine when two curves are actually coincident. The tolerance is computed by multiplying the relative tolerance value by the curve length.

Keyboard Command Sequence:

    match_curves <curve_list>

MATCH CURVE DIVISIONS

Command:  match_curves_af

Description:  This command assures that the number of mesh divisions on coincident curves are equal and the divisions are aligned. The given tolerance is used to determine when two curves are actually coincident.

Keyboard Command Sequence:

    match_curves_af <tolerance> <curve_list>
Material Name: QSt 32–3
Material Number: 1_0303__

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 500.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **D 5-1 B**  
Material Number: **1_0312_b**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe  
Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: D 5-1 U
Material Number: 1_0312_u

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper,
Imtiaz Saeed Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **C 15**  
Material Number: **1_0401_n**  

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: C 15  
Material Number: 1_0401_w  

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1000.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 8.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute  Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: C 22
Material Number: 1_0402

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>600.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>0.2 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: C 35
Material Number: 1_0501_a

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain (1/s)</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 100.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  
C 35
Material Number: 1_0501_n

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>700.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Not kemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: C 45  
Material Number: 1_0503__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute  
Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: C 60
Material Number: 1_0601_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>700.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Imitaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblatt (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  Ck 15  
Material Number:  1_1141__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data
For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 15
Material Number: 1_1141_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 15
Material Number: 1_1141_n

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 15
Material Number: 1_1141_w

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 35  
Material Number: 1_1181__  

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>1080.0 - 1330.0</td>
<td>0.0 - 2.0</td>
<td>0.1 - 36.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute  Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 45
Material Number: 1_1191__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: Ck 45
Material Number: 1_1191_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Duesseldorf, 1992
**Material Name:** Ck 45  
**Material Number:** 1_1191_w  
Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td>1.6 - 40.0</td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **Cf 53**  
Material Number: **1_1213**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 800.0</td>
<td>0.0 - 1.0</td>
<td>40.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **Ck 67**  
Material Number: **1_1231**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1000.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.  
Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien  

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  **X 54 NiCrMoW 4**  
Material Number:  **1_2765**  

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1250.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>20.0 - 400.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
*Fliesskurvenatlas metallischer Werkstoffe* Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 100 Cr 6
Material Number: 1_3505_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  

\textbf{X 10 Cr 13} 

Material Number: \texttt{1_4006__} 

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N] 

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 90.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\textbf{Note:}  If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. 

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

\textbf{Reference for this material data} 

For yield stress data: 

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed 
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, 
Muenchen, Wien 

For all other material data: 

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher 
Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X10Cr13 + 0.5 Mo
Material Number: 1_4006_b

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1150.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 20 Cr 13  
Material Number: 1_4021__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 40 Cr 13
Material Number: 1_4034_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Not kemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  X 22 CrNi 17  
Material Number:  1_4057___

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1250.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 800.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **X 15 CrMo 13**  
Material Number: **1_4119**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>750.0 - 1000.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X12 CrNi18 8
Material Number: 1.4300

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>900.0 - 1200.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 800.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 5 CrNi 18 9
Material Number: 1_4301__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 800.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td>0.2 - 63.0</td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 8 CrNi 12 12
Material Number: 1_4307

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>900.0 - 61200.0</td>
<td>0.0 - 1.0</td>
<td>0.2 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 800.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Flieskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **X 5 CrNiMo 18 10**  
Material Number: **1_4401_1**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>900.0 - 1200.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 1000.0</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.  
Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X5CrNiMo18_10
Material Number: 1_4401_2

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1250.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 5 CrNiMo 18 10
Material Number: 1_4401_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>0.2 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 10 CrNiTi 18 9  
Material Number: 1_4541__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 10 CrNiMoTi 18 10
Material Number: 1_4571__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 63.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblatter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 20 CrMoV 12 1
Material Number: 1_4922__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1150.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>20.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,
München, Wien

For all other material data:

Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Düsseldorf, 1992
Material Name: X 6 CrNiWNb 16 16
Material Number: 1_4945__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **X 12 CrNiWTi 16 13**
Material Number: **1_4962__**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,
München, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Düsseldorf, 1992
Material Name: X 12 CrCoNi 21 20  
Material Number: 1_4971__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher
Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: X 40 CoCrNi 20 20
Material Number: 1_4977_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1100.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 1000.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 15 CrNi 6
Material Number: 1_5919__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **X 2 NiCoMo 18 9 5**  
Material Number: **1_6358**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>950.0 - 1250.0</td>
<td>0.0 - 1.0</td>
<td>1.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td>-100.0 - 400.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,  
München, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 15 Cr 3
Material Number: 1_7015_a

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Property</th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
**Material Name:** 15Cr3

**Material Number:** 1_7015_b

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 500.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 8.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 15 Cr 3
Material Number: 1_7015_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 15 Cr 3
Material Number: 1_7015_w

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  **41 Cr 4**  
Material Number:  **1_7035**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: **41 Cr 4**  
Material Number: **1_7035_g**

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 500.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 41 Cr 4  
Material Number: 1_7035_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>200.0 - 600.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 41 Cr 4  
Material Number: 1_7035_n  

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>8.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 41 Cr 4  
Material Number: 1_7035_w

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 16 MnCr 5  
Material Number: 1_7131__  

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1100.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.  
Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 20 MnCr 5
Material Number: 1_7147_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 800.0</td>
<td>0.0 - 1.0</td>
<td>40.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblätter (SEW) des Vereins Deutscher Eisenhüttenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 42 CrMo 4
Material Number: 1_7225_

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 42 CrMo 4  
Material Number: 1_7225_1  
Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>40.0 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:
Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag, Muenchen, Wien

For all other material data:
Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 42 CrMo 4  
Material Number: 1_7225_h

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>600.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name:  **42 CrMo 4**  
Material Number: **1_7225_w**

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 16 CrMo 4
Material Number: 1_7242__

Units: Length [mm], Mass [Mg], Time [s], Temperature [°C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>800.0 - 1200.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value.

Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fließkurvenatlas metallischer Werkstoffe Hanser Verlag,
München, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
**Material Name:** 16 CrMo 4  
**Material Number:** 1_7242_w

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th>Property</th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 600.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

**Reference for this material data**

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Imtiaz Saeed  
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,  
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
Material Name: 20 MoCr 4
Material Number: 1_7321__

Units: Length [mm], Mass [Mg], Time [s], Temperature [C], Force [N]

<table>
<thead>
<tr>
<th></th>
<th>Temperature Range (degree)</th>
<th>Log. Strain</th>
<th>Strain Rate (1/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow stress</td>
<td>20.0 - 500.0</td>
<td>0.0 - 1.0</td>
<td>1.6 - 40.0</td>
</tr>
<tr>
<td>Elastic and Thermal Properties</td>
<td>-100.0 - 600.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If a calculated value is outside the defined range, no extrapolation of the material data is done. In this case, the program uses the smallest or largest defined value. Only in the case of a temperature dependent elastic modulus, a linear extrapolation will be performed (see material tables).

Reference for this material data

For yield stress data:

Eckart Doege, Heinz Meyer-Notkemper, Intiaz Saeed
Fliesskurvenatlas metallischer Werkstoffe Hanser Verlag,
Muenchen, Wien

For all other material data:

Stahl-Eisen-Werkstoffblaetter (SEW) des Vereins Deutscher Eisenhuettenleute Verlag Stahleisen mbh, Duesseldorf, 1992
**Command:** `material_name`  
**Description:** This command sets or changes the name of the current material.  
A material is a collection of information that describes a material to be applied to elements in the model. Materials contain the material type, material properties, tables or functions that apply to those properties, and options that will be applied to the specified set of elements during analysis. Multiple materials may be defined and are stored in the list of currently defined materials.  

**Keyboard Command Sequence:**

```
material_name <material name>
```

---

**Command:** `material_option`  
**Description:** This command sets an option for the current material. Options vary depending on the material type.  
The option to be set is designated by the material type, the option name, and the desired value separated by colons.  
For example, modal damping method for a dynamic material is designated as `dynamic:method:modal`. The value or choice for an option is always specified as a label.  

**Keyboard Command Sequence:**

```
material_option <material type>:<option name>:<choice>
```

Other Buttons with the same description:
- `ADDITIVE DECOMPOSITION`
- `ANISOTROPIC`
- `BINGHAM`
- `BUYUKOZTURK CONCRETE`
- `CAM CLAY`
- `CARREAU`
- `COMBINED`
- `CONSTANT`
COWPER-SYMonds
DATABASE VALUES
DEF. GRADIENT DEF. GRADIENT DEF. GRADIENT, ROTATION TENSOR
STRETCH RATIOS ROTATION TENso
DEF. GRADIENT DEF. GRADIENT, ROTATION TENSOR STRETCH RATIOS
DEF. GRADIENT, ROTATION TENSOR
ENTERED VALUES
FULL ALPHA RESET ORNL
GENERALIZED PLASTICITY
GENERALIZED POWER LAW
HILL
HOFFMAN
ISOTROPIC
JOHNSON-COOK
KINEMATIC
KUMAR
LINEAR
LINEAR MOHR-COULomb
MAXIMUM STRAIN
MAXIMUM STRESS
MULTIPLICATIVE DECOMPOSITION
NARAYANASwAMY
NEWTONIAN
NONE
NONE
OAK RIDGE NATIONAL LAB
ORTHOTROPIC
PARABOLIC MOHR-COULomb
PIECEWISE LINEAR
PIECEWISE-LINEAR
PLASTIC STRAIN CONTROL
POWER LAW
POWER SERIES
PROGRESSIVE FAILURE
RATE POWER LAW
REVERSED PLASTICITY ORNL
RIGID-PLASTIC
STRAIN DEPENDENT
STRESS CONTROL
TSAI-WU
USER SUB.
USER SUB. ANEXP
USER SUB. ANKOND
USER SUB. CRPLAW
USER SUB. HYPELA
USER SUB. HYPELA2
USER SUB. TRSFAC
USER SUB. UBEAM
USER SUB. UCRACK
USER SUB. UEPS
USER SUB. UFAIL
USER SUB. UGRAIN
USER SUB. UMU
USER SUB. UNEWTN
USER SUB. UPHI
USER SUB. UVOIDN
USER SUBROUTINE UELDAM
Command: `material_read`

Description: This command reads a material model file and imports the materials into the current model.

Keyboard Command Sequence:

```
material_read <material model name>
```

Another Button with the same description:

READ OTHER MATERIAL

Command: `material_remove_type`

Description: Removes the specified type or subtype from the current material. You must specify the material type or subtype (see `material_type`).

Keyboard Command Sequence:

```
material_remove_type <type>
material_remove_type <type>:<mechanical type option>
material_remove_type <subtype>
material_remove_type <subtype>:<subtype option>
```
**Command:** material_reset

**Description:** This command resets the values of the specified material subtype to their initial default values. You must specify the material type or subtype (see material_type).

**Keyboard Command Sequence:**

```
material_reset <type>
material_reset <type>::<mechanical type option>
material_reset <subtype>
material_reset <subtype>::<subtype option>
```

---

**Command:** material_table

**Description:** This command adds a table for a property in the current material. A table is a function that makes the property value dependent on another variable during analysis.

The property to receive the table is designated by the material type and the property name separated by a colon, and an appended table number. For example, a table for Young’s modulus in anisotropic material is designated as isotropic:e0. Table numbers start at zero.

Note that only certain properties may have tables associated with them. Refer to Marc material model descriptions to determine which properties can have tables.

The table specified for the property must have already been created.

**Keyboard Command Sequence:**

```
material_table <material type>::<property name>
     <table number> <table name>
```
Command: **material_type**

Description: This command is used to set the type of the current material and to add subtypes to the current material.

To set the type of the current material, the material type must be provided as the argument. The current material types are mechanical, heat, joule, acoustic, bearing, electrostatic, and magnetostatic. If the type is mechanical, then the desired mechanical option must be given, appended with a colon as in `mechanical:isotropic`.

To add a subtype, the subtype must be provided. The current subtypes are `dynamic`, `plasticity`, `thermal_expansion`, `rate` (for rate effects), `thermo_rheologic` (for thermo-rheologically simple) and `damage`. Some subtypes require an additional option specification (also appended after a colon). The subtypes requiring additional options are as follows:

**thermal_expansion:**
- `iso_thermal` (isotropic thermal expansion)
- `ortho_thermal` (orthotropic thermal expansion)

**rate:**
- `iso_visco` (linear isotropic viscoelastic)
- `ortho_visco` (linear orthotropic viscoelastic)
- `strain_visco` (large strain viscoelastic)
- `creep`
- `phi` (phi functions)

**damage:**
- `cracking`
- `damage`
- `failure`

**forming limit:**

To define the Forming Limit Diagram (FLD) curve, this type of property can be applied to shell/membrane elements. There are three methods to define the FLD curves as following:

- `fitted`: (fitted polynomial formulation of FLD curve)
- `predicted`: (predicted FLD curve based on the material properties)
table: (table form definition for FLD curve)

Reference: MSC.Marc Volume A: Theory and Information.

Keyboard Command Sequence:

```plaintext
material_type <type>
material_type <type>:<mechanical type option>
material_type <subtype>
material_type <subtype>:<subtype option>
```

Other Buttons with the same description:

- ACOUSTIC
- ANISOTROPIC
- ARRUDA-BOYCE
- BEARING
- CRACKING
- CREEP
- DAMAGE
- DAMPING
- ELASTIC-PLASTIC
- ELECTROMAGNETIC
- ELECTROSTATIC
- FAILURE
- FLUID
- FOAM
- GENT
- GRAIN SIZE
- HEAT TRANSFER
- HYPOELASTIC
- JOULE HEATING
- LATENT HEAT
- MAGNETOSTATIC
- MOONEY
- OGDEN
- OK
- ORTHOTROPIC
- PHI
- PLASTICITY
- POWDER
- RUBBER
- SOIL
- THERMAL EXP.
- THERMO-RHEOLOGICALLY SIMPLE
- VISCOELASTIC
**YOUNG'S MODULUS**

Command: **material_value**

**Description:** This command sets a property value for the current material. The value entered should be expressed in units consistent with those used to describe the mesh and with other material property values.

The property to be set is designated by the material type and the property name separated by a colon. For example, Young’s modulus for anisotropic material is designated as `isotropic:youngs_modulus`.

**Keyboard Command Sequence:**

```
material_value <material type>:<property name> <value>
```

Other Buttons with the same description:

- 1 through 9
- # DEVIATORIC TERMS
- # DILATATIONAL TERMS
- # LATENT HEATS
- # TERMS
- # VOLUMETRIC TERMS
- 10
- 10TH CYCLE YIELD STRESS
- 11 through 16
- 1ST YIELD SURFACE MULTIPLIER
- 21 through 26
- 2ND YIELD SURFACE MULTIPLIER
- 31 through 36
- 41 through 46
- 51 through 56
- 61 through 66
- A1 through A4
- ABS. TEMPERATURE SHIFT
- ACTIVATION ENERGY
- AIR PERMEABILITY
- ALPHA
- ALPHA11
- ALPHA22
- ALPHA33
- ANGLE
- BACK STRESS
- BETA
- BULK MODULUS
- C0
- C01
- C1
- C10
- C11
- C2
C20
C3
C30
C4
C5
CAV. PRESSURE
CHAIN LENGTH
COEFFICIENT
COEFFICIENT A
COEFFICIENT B
COEFFICIENT B0
COEFFICIENT B1
COEFFICIENT B2
COEFFICIENT B3
COEFFICIENT B4
COEFFICIENT B5
COEFFICIENT B6
COEFFICIENT C
COEFFICIENT N
COMPRESSION RATIO
CONDUCTIVITY
CONDUCTIVITY1
CONDUCTIVITY11
CONDUCTIVITY12
CONDUCTIVITY13
CONDUCTIVITY2
CONDUCTIVITY22
CONDUCTIVITY23
CONDUCTIVITY3
CONDUCTIVITY33
CRITICAL STATE CURVE SLOPE
CRITICAL STRESS
CRITICAL VOID VOLUME FRACTION
CRUSHING STRAIN
CUTOFF STRAIN RATE
DYNAMIC VISCOSITY
E11 through E33
EMISSIVITY
EXPONENT
EXPONENT M
EXPONENT N
FAILURE INDEX
FAILURE VOID VOLUME FRACTION
FLUID BULK MODULUS
FLUID DRAG
FRACTION PARAMETER
G12
G23
G31
GAMMA
GLOBAL LAYER ID
H-B RELATION
H-B RELATION1
H-B RELATION11
H-B RELATION22
H-B RELATION33
INITIAL GRAIN SIZE
INITIAL STRAIN RATE
INITIAL VOID VOLUME FRACTION
INITIAL YIELD STRESS
INTERACTIVE STRENGTH TENSOR XY
INTERACTIVE STRENGTH TENSOR YZ
INTERACTIVE STRENGTH TENSOR ZX
INV PERMEABILITY
INV PERMEABILITY11
INV PERMEABILITY22
INV PERMEABILITY33
INVERSE EXPONENT P
K
LIQUID COEF
MASS DENSITY
MASS MATRIX MULTIPLIER
MAX COMPRESSIVE STRAIN X
MAX COMPRESSIVE STRAIN Y
MAX COMPRESSIVE STRAIN Z
MAX COMPRESSIVE STRESS X
MAX COMPRESSIVE STRESS Y
MAX COMPRESSIVE STRESS Z
MAX SHEAR STRAIN XY
MAX SHEAR STRAIN YZ
MAX SHEAR STRAIN ZX
MAX SHEAR STRAIN XY
MAX SHEAR STRAIN YZ
MAX SHEAR STRAIN ZX
MAX TENSILE STRAIN X
MAX TENSILE STRAIN Y
MAX TENSILE STRAIN Z
MAX TENSILE STRESS X
MAX TENSILE STRESS Y
MAX TENSILE STRESS Z
MAX. 1ST INVARIANT
MELT TEMPERATURE
MULTIPLIER
N
NKT
NU12
NU23
NU31
PERMEABILITY
PERMEABILITY11
PERMEABILITY22
PERMEABILITY33
PERMITTIVITY
PERMITTIVITY11
PERMITTIVITY22
PERMITTIVITY33
POISSON RATIO
POISSON'S RATIO
REBAR AREA
RECOMPRESSION RATIO
REF PLANE
REF. TEMPERATURE
REL. LAYER POSITION
RESISTIVITY
RESISTIVITY11
RESISTIVITY22
RESISTIVITY33
ROOM TEMPERATURE
SCALAR FACTOR @ INF
SHEAR RETENTION
SOFTWARE MODULUS
SOLID COEF
SPECIFIC HEAT
STANDARD DEVIATION
STIFFNESS MATRIX MULTIPLIER
STRUCT. RELAX. REF. TEMP.
TENSILE MODULUS
THERMAL EXP. COEF.
THERMAL EXPANSION
THICKNESS
VISCOSITY
VISCOSITY @ INF.
VOLUME FRACTION FOR VOID NUCLE
VOLUMETRIC EXP.
X
Y
YIELD STRENGTH
YIELD STRESS
YRDIR1
YRDIR2
YRDIR3
YRSR1
YRSR2
YRSR3
Z

---

**Command:** `max_curve_divisions`

**Description:** This command assures that the number of edge divisions on the list of given curves is no greater than the given number.

**Keyboard Command Sequence:**

```
max_curve_divisions <number of divisions>
<curve list>
```
**MEMORY**

**Command:** memory

**Description:** This command prints the current MSC.Marc Mentat memory usage on standard output.

**Keyboard Command Sequence:**

memory

---

**MENU EXECUTE**

**Command:** menu_execute

**Description:** This command specifies whether or not menu commands entered through the dialog area will be executed.

**Keyboard Command Sequence:**

menu_execute <on or off>

---

**MENU RECORD**

**Command:** menu_record

**Description:** This command specifies whether or not menu commands are recorded in the dialog area.

**Keyboard Command Sequence:**

menu_record <on or off>
### MERGE DUPLICATE BOUND CONDS

**Command:** `merge_applies`

**Description:** This command merges loads and boundary conditions which have identical data.

**Keyboard Command Sequence:**

```
merge_applies
```

### MERGE

**Command:** `merge_models`

**Description:** This command merges a MSC.Marc Mentat model database with the current model. The current model is preserved. To replace the current model with a new model, use the `open_model` command.

**Keyboard Command Sequence:**

```
merge_model <model name>
```

### TRANSITION

**Command:** `set_mesh_transition`

**Description:** This command sets mesh transition parameter. Its default value is 1. When the value is bigger than 1, the element size at the central area will be larger. When the value is smaller than 1, the element size at the central area will be smaller.

**Keyboard Command Sequence:**

```
set_mesh_transition <value>
```
Menu: **MESH GENERATION**

Description: This menu contains mesh generation commands which are used to create and/or edit geometry and element meshes. Functionality includes adding, removing, and editing nodes, elements, points, curves, and surfaces of new or existing meshes.

In addition to these basic functions, the MESH GENERATION menu contains several submenus that contain commands for transforming the mesh.

**message**

Command: **message**

Description: This command displays any text which follows it. This may be useful for commenting the output generated from a MSC.Marc Mentat procedure file.

**Keyboard Command Sequence:**

```plaintext
message Put your message here....
```

**min_curve_divisions**

Command: **min_curve_divisions**

Description: This command assures that the number of edge divisions on the list of given curves is no less than the given number.

**Keyboard Command Sequence:**

```plaintext
min_curve_divisions <number of divisions> <curve list>
```
**DESCRIPTION**

Menu: **MODEL DESCRIPTION**

Description: This menu allows the user to set the description for the current model. The text field is used for entering the description. Either a return typed into the text field or clicking on the OK button will cause the `model_description` command to be executed and close the popup menu. Clicking the RESET button resets the text field to the current model description. Clicking the CANCEL button closes the popup menu without executing the command.

Keyboard Command Sequence:

```
model_description <model description>
```

Other Buttons with the same description:

CANCEL
OK

---

**MONITOR**

Command: **monitor_job**

Description: This command updates the status of a job on a periodic basis in order to provide a means of watching the progress of an analysis without the interaction of the user. The command is terminated by any input by the user.

Keyboard Command Sequence:

```
monitor_job
```
**MOONEY(2)**

**Command:** `xcv_model mooney2`  
**Description:** This command is used if experimental data must be fitted using the two term Mooney-Rivlin strain energy function $W$, which is given by:

$$W = C_{10}*(I_1 - 3) + C_{01}*(I_2 - 3)$$

where $I_1$ and $I_2$ are the first and second invariant of the right Cauchy-Green strain tensor and $C_{10}$ and $C_{01}$ are the material parameters to be determined.

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (`material_type mechanical:mooney`). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**

`xcv_model mooney2`

**MOONEY(3)**

**Command:** `xcv_model mooney3`  
**Description:** This command is used if experimental data must be fitted using the three term Mooney-Rivlin strain energy function $W$, which is given by:

$$W = C_{10}*(I_1 - 3) + C_{01}*(I_2 - 3) + C_{11}*(I_1 - 3)*(I_2 - 3)$$

where $I_1$ and $I_2$ are the first and second invariant of the right Cauchy-Green strain tensor and $C_{10}$, $C_{01}$, and $C_{11}$ are the material parameters to be determined.

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (`material_type mechanical:mooney`). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**

`xcv_model mooney3`
**Subroutine: MOTION**

**Description:** The user subroutine MOTION allows you to define nonuniform rigid surface motions, in conjunction with the option CONTACT. Its call is triggered by the model definition option UMOTION.

Subroutine MOTION is called during the calculations at the beginning of each time increment, and you return surface velocities for that increment. Imposed displacement increments at nodal points in contact with rigid surfaces are obtained from the velocity multiplied by the time increment. The surface path becomes an explicit forward integration of velocities. Therefore, caution should be used when there are abrupt changes in surface path direction or abrupt changes in velocity, by making time increments as small as necessary.

If, at the start of the analysis, a surface is placed apart from the body to be deformed, subroutine MOTION will also be used in the approaching phase, in time increments of 0.1 seconds.

**Menu: MOVE**

**Description:** This menu contains commands for relocating geometry and mesh entities.
Command: **move_annotation**

Description: This command moves annotations within views and between views. You must specify the annotation to move, the view to receive the annotation, and its XY coordinates within that view. Annotations and views are specified by number.

Keyboard Command Sequence:

```
move_annotation <annotation> <view> <X> <Y>
```
To set the parameters that control the relocation, use the following commands:

```
set_move_point
set_move_scale_factor
set_move_rotations
set_move_translation
set_move_formulas
```

Also see:

```
move_nodes
move_elements
move_points
move_curves
move_surfaces
move_solids
move_ties
move_servos
move Springs
move_rbe2s
move_rbe3s
```

**Keyboard Command Sequence:**

```
move_combined <item list> #
set_move_combined <item type> <on|off>
```

---

**Command:** move_curves

**Description:** This command moves the specified curves. To set the parameters that control the relocation, use the following commands:

```
set_move_point
set_move_scale_factors
set_move_rotations
set_move_translation
set_move_formulas
```

**Keyboard Command Sequence:**

```
move_curves <curve list> #
```
**ELEMENTS**

**Command:** move_elements  
**Description:** This command moves the specified elements. To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factors
- set_move_rotations
- set_move_translation
- set_move_formulas

**Keyboard Command Sequence:**

```
move_elements <element list> #
```

**MODEL**

**Command:** move_model  
**Description:** This command moves the entire model (i.e. mesh, geometry, and links). To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factor
- set_move_rotations
- set_move_translation
- set_move_formulas

This button is located in the MESH GENERATION->MOVE menu.

**Keyboard Command Sequence:**

```
move_model
```
**NODES**

**Command:**  move_nodes

**Description:** This command moves the specified nodes. To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factors
- set_move_rotations
- set_move_translation
- set_move_formulas

**Keyboard Command Sequence:**

```
move_nodes <node list> #
```

---

**POINTS**

**Command:**  move_points

**Description:** This command moves the specified points. To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factors
- set_move_rotations
- set_move_translation
- set_move_formulas

**Keyboard Command Sequence:**

```
move_points <point list> #
```
**RBE2S**

**Command:** `move_rbe2s`

**Description:** This command moves the specified RBE2’s. To set the parameters that control the relocation, use the following commands:

- `set_move_point`
- `set_move_scale_factor`
- `set_move_rotations`
- `set_move_translation`
- `set_move_formulas`

This button is located in the MESH GENERATION->MOVE menu.

**Keyboard Command Sequence:**

```
move_rbe2s <rbe2 list> #
```
**RESET**

**Command:**  move_reset

**Description:** This command resets the move parameters to their default values.

**Keyboard Command Sequence:**

move_reset

---

**SERVOS**

**Command:**  move_servos

**Description:** This command moves the specified servo. To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factor
- set_move_rotations
- set_move_translation
- set_move_formulas

This button is located in the MESH GENERATION->MOVE menu.

**Keyboard Command Sequence:**

move_servos <servo list> #

---

**SPRINGS**

**Command:**  move_springs

**Description:** This command moves the specified springs. To set the parameters that control the relocation, use the following commands:

- set_move_point
- set_move_scale_factor
- set_move_rotations
- set_move_translation
- set_move_formulas

This button is located in the MESH GENERATION->MOVE menu.

**Keyboard Command Sequence:**

move_springs <spring list> #
**SURFACES**

Command: **move_surfaces**

Description: This command moves the specified surfaces. To set the parameters that control the relocation, use the following commands:

- `set_move_point`
- `set_move_scale_factors`
- `set_move_rotations`
- `set_move_translation`
- `set_move_formulas`

Keyboard Command Sequence:

```
move_surfaces <surface list> #
```

**TIES**

Command: **move_ties**

Description: This command moves the specified ties. To set the parameters that control the relocation, use the following commands:

- `set_move_point`
- `set_move_scale_factor`
- `set_move_rotations`
- `set_move_translation`
- `set_move_formulas`

This button is located in the MESH GENERATION->MOVE menu.

Keyboard Command Sequence:

```
move_ties <tie list> #
```
MAKE MOVIE

Command:  movie_animation_make

Description:  This command makes a movie from existing animation files. The current animation base filename is used along with the current animation begin/end index settings to determine which animation files to playback and capture. Those animation files are then played in forward order, and the appearance of each one in a selected view is captured into an image file. Each created image file will have the same name as its corresponding animation file, followed by .rgb.

Make sure the graphics window is not obscured by another window while the movie is made.

On some systems, the quality of the movie may be improved by switching the program into single-buffered mode before making the movie.

You must specify the number of the view to make the movie from.

Also see: movie_animation_play, animation_name, animation_index_all, animation_index_begin, image_save, and set_buffered.

Keyboard Command Sequence:

  movie_animation_make <view>
**Command:**  movie_animation_play

**Description:** This command brings up a new window displaying a movie created from animation files. The current animation base filename is used along with the current animation begin/end index settings to determine which image files to playback as a movie. Thus, an image file must exist for each indicated animation file.

After all the frames have been loaded, and the movie is running, these keystrokes in the movie window do the following:

- **+** Increase playback speed by shortening pause time
- **-** Decrease playback speed by lengthening pause time
- **F,f** Forward
- **R,r** Reverse
- **L,l** Loop
- **S,s** Swing
- **P,p** Pause
- **C,c** Continue
- **Q,q,<ESC>** Exit

Also see **movie_animation_make**.

**Keyboard Command Sequence:**

movie_animation_play
**COMPRESSION DIAGLOG**

**Command:** `movie_avi_compress`

**Description:** This command toggles whether or not to display the AVI compression methods dialog box when making an AVI animation. The default uses no compression and does not display the dialog box. The compression options displayed are those that are installed on your system, and the dialog box is a Windows NT based menu.

Note that displaying the dialog box will make the process interactive and not able to automatically run from a procedure file since the dialog box is a Windows NT dialog box.

Also see: `avi_animation_make` (MAKE AVI MOVIE).

**Keyboard Command Sequence:**

```
movie_avi_compress <on/off>
```

**DELAY**

**Command:** `movie_delay`

**Description:** The DELAY command sets the delay value for an MPEG animation by duplicating frames in the animation file. Each frame appears DELAY+1 times in the animation. This is useful for some MPEG players that attempt to play the animation in (LAST-FIRST)/30fps time (in seconds). The default value is 5.

Also see: `mpeg_animation_make` (MAKE MPEG MOVIE) and `avi_animation_make` (MAKE AVI MOVIE).

**Keyboard Command Sequence:**

```
movie_delay <value>
```
Command: **movie_first_increment**

**Description:** This command sets the first increment in the post file to be used in generating an MPEG or AVI animation. The default value is zero, which is the model increment in the post file.

Note that this value is the increment found in the post file, not necessarily the increment number displayed by MSC.Marc Mentat. That number will typically be one less than the post file increment. For example, if you wish to end with the increment that displays as 10, then you would set the LAST INCREMENT to be 11.

Also see: mpeg_animation_make (MAKE MPEG MOVIE) and avi_animation_make (MAKE AVI MOVIE).

**Keyboard Command Sequence:**

```
movie_first_increment <value>
```

---

Command: **movie_gen_files**

**Description:** This command enables or disables the automatic generation of the animation files which are typically created using the command ANIMATION>CREATE>INCREMENTS button (the post_animate_increments command).

In some cases, it may be more useful to generate the animation files individually or incrementally, and then generate the MPEG or AVI movie. If that is the case, then turn off this option.

Also see: mpeg_animation_make (MAKE MPEG MOVIE) and avi_animation_make (MAKE AVI MOVIE).

**Keyboard Command Sequence:**

```
movie_gen_files <on/off>
```
Command: **movie_last.increment**

Description: This command sets the last increment to be used in generating an MPEG or AVI animation. The default value is the last increment found on the post file.

Note that this value is the increment found in the post file, not necessarily the increment number displayed by MSC.Marc Mentat. That number will typically be one less than the post file increment. For example, if you wish to end with the increment that is displayed as 10, then you would set the LAST_INCREMENT to be 11.

Also see: mpeg_animation_make (MAKE MPEG MOVIE) and avi_animation_make (MAKE AVI MOVIE).

Keyboard Command Sequence:

```
movie_last.increment <value>
```

Command: **movie_step_increment**

Description: This command sets the value for the increment step size to be used when generating an MPEG or AVI animation. The default value is one.

Also see: mpeg_animation_make (MAKE MPEG MOVIE) and avi_animation_make (MAKE AVI MOVIE).

Keyboard Command Sequence:

```
movie_step_increment <value>
```
**movie_view**

**Description:** This command sets the view to use when generating an MPEG or AVI animation. The values must be in the range of 1 through 4, and must be the current view when generating the movie.

Also see: mpeg_animation_make (MAKE MPEG MOVIE) and avi_animation_make (MAKE AVI MOVIE).

**Keyboard Command Sequence:**

```
movie_view <value>
```

**mpeg_animation_make**

**Description:** This command (MAKE MPEG MOVIE) will generate an MPEG movie file.

It will use the values displayed under INCREMENT SETTINGS, FIRST (movie_first_increment), LAST (movie_last_increment), STEP (movie_step_increment), VIEW (movie_view), and DELAY (animation_pause).

The view will be specified in the VIEW display and must be the current view.

The MPEG movie is generated in a 4 step process:

1. The animation files are automatically generated which is typically performed manually with the ANIMATION->CREATE->INCREMENTS button (the post_animate_increments command). Also see the option GENERATE ANIMATION FILES (movie_gen_files command).

2. Once the animation files are created, the animation is played and screen images are captured into a sequence of .ppm files. These file types are required for the MPEG encoder.

3. After the .ppm files have been created, a file named mpeg_params is created in the current working directory which contains a list of the image files and other parameters to be used in generating the MPEG movie file.
4. The MPEG encoder (mpeg_encode.exe in the bin directory) is run in the background. This will generate a file named <base file>.mpeg, where <base file> is the name specified with the BASE FILE button (the animation_name command).

Note: The MPEG encoder does not provide any feedback to MSC.Marc Mentat to indicate that it has completed. You must verify that the mpeg_encode.exe program is no longer running before you attempt to use the resulting .mpeg file.

Use the Windows Task Manager on NT or use the ps command on UNIX to verify that the program has ended.

The mpeg_params file may be modified and rerun using the syntax:

    mpeg_encode -realquiet mpeg_params

Note that this command will remove all animation display list files and .ppm files that begin with the BASE FILE name (*animation_name) before it starts. The GENERATE ANIMATION FILES option (movie_gen_files <on/off> command) can be used to enable or disable the automatic generation of the animation files.

You can interrupt the process of making the MPEG file by pressing the Escape key, however, you will be returned to the animation play mode. In this case, you will need to run the show_model command in the ANIMATION menu to display the model again.

Note: To generate a movie size equal to 640 x 480, you should use the startup option -sz 856 577.

Keyboard Command Sequence:

    mpeg_animation_make
Mentat Help Commands in N

**NASSOC**

**Subroutine:** NASSOC

**Description:** The user subroutine NASSOC allows you to calculate a strain increment using a flow rule differing from the normality rule of plasticity, which is the default used by the program.

**NEO-HOOKEAN**

**Command:** xcv_model neo_hookean

**Description:** This command is used if experimental data must be fitted using the Neo-Hookean strain energy function \( W \), which is given by:

\[
W = C10 \cdot (I1 - 3)
\]

where \( I1 \) is the first invariant of the right Cauchy-Green strain tensor and \( C10 \) is the material parameter to be determined.
For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (material_type mechanical: mooney). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**

```
xcv_model neo_hookean
```

---

**Command:** `new_adapg`

**Description:** This command creates a new entry in the list of global remeshing criteria and makes it the current entry. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

An entry in the list of global remeshing criteria is a collection of parameters that control remeshing of a contact body during an analysis.

Global remeshing allows different parts of the model to be remeshed. A new mesh is automatically created, relevant solution quantities are transferred to the new mesh and the analysis continues.

This feature is currently only supported for contact bodies and only boundary conditions applied with contact bodies are allowed on the remeshed body. The remeshing is performed separately for each contact body.

Only 2-D problems can use remeshing in this release.

**Keyboard Command Sequence:**

```
new_adapg
```

---

**Command:** `new_adapt`

**Description:** This command creates a new entry in the list of adaptats and makes it the current adapt. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

An adapt is a collection of parameters that controls the adaptive meshing of a set of elements during an analysis.

**Keyboard Command Sequence:**

```
new_adapt
```
Command: new_apply
Description: This command creates a new entry in the list of applications and makes it the current application. This button is located in the BOUNDARY CONDITIONS menu.

A boundary condition application is a set of information that completely defines a boundary condition. Each application contains the type of boundary condition, the degrees of freedom, and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected. Multiple applications may be defined and are stored in the list of currently defined applications.

Keyboard Command Sequence:
new_apply

Command: new_cavity
Description: This command creates a new entry in the list of cavities and makes it the current entry.

A cavity entry is defined by a set of edges, faces, curves, or surfaces. The dimension can be 2-D or 3-D. It is located under the MESH GENERATION->CAVITIES menu.

Keyboard Command Sequence:
new_cavity
NEW COMPOSITE

**Command:** new_composite

**Description:** This command creates a new composite material and makes it the current material.

**Keyboard Command Sequence:**

new_composite

NEW

**Command:** new_contact_area

**Description:** This command creates a new entry in the list of contact areas and makes it the current contact area. This button is located in the CONTACT->CONTACT AREAS menu.

A contact area is a set of nodes that lie on a contact body that may potentially contact other contact bodies. Contact areas are used to reduce the computational cost of a contact analysis where it is known for which nodes contact is possible. If no contact area is used, all exterior outline or surface nodes will be checked for contact.

**Note:** The contact area must be activated in the loadcase where they are to be used. This is defined in the CONTACT menu for the different loadcase types.

For the correct detection of initial contact (before the first loadcase), the contact area should also be activated in the current job. This is done in the INITIAL CONTACT menu in the CONTACT CONTROL menu for each analysis class. (For instance for an uncoupled mechanical analysis it is defined in JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT.)

**Keyboard Command Sequence:**

new_contact_area
**Command:**  `new_contact_body`  
**Description:** This command creates a new entry in the list of contact bodies and makes it the current contact body. This button is located in the CONTACT->CONTACT BODIES menu.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies and at least one of which must be a deformable body.

**Keyboard Command Sequence:**

```
new_contact_body
```  

---  

**Command:**  `new_contact_table`  
**Description:** This command creates a new entry in the list of contact tables and makes it the current contact table. This button is located in the CONTACT->CONTACT TABLES menu.

**Keyboard Command Sequence:**

```
new_contact_table
```  

---  

**Command:**  `new_crack`  
**Description:** This command creates a new entry in the list of cracks and makes it the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS and 3-D CRACKS menus.

The concept of a crack is a crack tip for 2-D and a crack front in 3-D. For each crack, Marc evaluates the J-integral and prints the results to the output file.

For 2-D, the J-integral is defined as a line integral along a path surrounding the crack tip. In the method used by Marc, this integral is
transformed into an area integral where the area to be integrated over is the one enclosed by the line integration path. Only a part of this area is actually used for the integration, usually the outermost ring of elements. In this way, only results values at a distance from the crack tip are used and more accurate results can be obtained for the J-integral.

To identify the integration area, you can define a list of nodes inside the elements that are used for the integration. These nodes define the RIGID REGION which the menus below refer to. Only elements that are connected to a node in the rigid region contribute to the J-integral value calculated. This does not include elements with all nodes in the rigid region. The term rigid region comes from the fact that you can evaluate the J-integral by shifting the nodes in the rigid region and calculate the change in strain energy in the elements connected to a node in the rigid region.

For 3-D, the line integral becomes a surface integral over a surface surrounding the crack front (or a part of the crack front). The area integral becomes a volume integral and the rigid region is similarly defined as a list of nodes. By choosing the rigid region as a thin disk of nodes perpendicular to the crack front, where the disk contains one node along the crack front, you can obtain local values of the J-integral along a crack front. The automatic search routines for the rigid regions are designed to create this type of rigid regions.

Shell elements are not supported in this release.

**Keyboard Command Sequence:**

```
new_crack
```

**Command:** new_csect

**Description:** This command creates a new entry in the list of cross-sections and makes it the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

A cross-section can be used to e.g. model prestressed bolts in a structure and is generally defined by:

- the nodes in the cross-section
The elements sharing nodes in the cross-section and lying on the side of the cross-section corresponding to the opposite normal direction
- the normal vector to the cross-section
- a control node

The control node has one degree of freedom, which can be seen as the shortening of the cross-section elements. The dual force is the force in the cross-section. The control node should generally not be connected to any element in the model.

**Keyboard Command Sequence:**

```
new_csect
```

**Command:** `new_descon`
**Description:** This command creates a new entry in the list of design constraints and makes it the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

A design constraint is an analysis response quantity that is constrained to be above or below a specified value.

**Keyboard Command Sequence:**

```
new_descon
```

**Command:** `new_desvar`
**Description:** This command creates a new entry in the list of design variables and makes it the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

A design variable is a quantity that can vary during an optimization or sensitivity analysis.

**Keyboard Command Sequence:**

```
new_desvar
```
Command:  **new_exseg**

Description:  This command creates a new entry in the list of exclude segments and makes it the current one. This button is located in the CONTACT-> EXCLUDE SEGMENTS menu.

An exclude segment is a set of edges and faces of a contact body that are guaranteed not to contact other contact bodies.

Exclude segments are used to reduce the computational cost of a contact analysis where it is known for which edges and faces contact is not possible. It is also a way to make sure that the proper part of a body is contacted. In some situations, a node can contact a part of a body which it is not supposed to contact. This can occur for instance when a node comes into contact close to a corner of another body and it slides along the wrong segment of the element at the corner. A better behavior can be obtained if the other segments are excluded.

Note that the exclude segments must be activated in the loadcase where they are to be used. This is defined in the CONTACT menu for the different loadcase types.

For the correct detection of initial contact (before the first loadcase), the exclude segments should also be activated in the current job. This is done in the INITIAL CONTACT menu in the CONTACT CONTROL menu for each analysis class. (For instance, for an uncoupled mechanical analysis, it is defined in JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT.)

**Keyboard Command Sequence:**

```
new_exseg
```
Command: new_gasket
Description: This command creates a new gasket material and makes it the current material. This button is located in the MATERIAL PROPERTIES->LAYERED MATERIALS menu.

Keyboard Command Sequence:
new_gasket

Command: new_geometry
Description: This command creates a new entry in the list of geometries and makes it the current geometry. This button is located in the GEOMETRIC PROPERTIES menu.

An element geometry is a collection of the geometric properties to be applied to a set of elements. Geometric properties include but are not limited to shell thickness, beam and truss areas, and beam moments of inertia.

Keyboard Command Sequence:
new_geometry
Command: new_icond
Description: This command creates a new entry in the list of initial condition and makes it the current initial condition. This button is located in the INITIAL CONDITIONS menu.

A initial condition is a boundary condition that is applied to the mesh at the beginning of the analysis. Each initial condition contains the degrees of freedom and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected.

Keyboard Command Sequence:
new_icond

Command: new_insert
Description: This command creates a new entry in the list of inserts and makes it the current insert.

Keyboard Command Sequence:
new_insert
Command:  new_job

Description:  This command creates a new entry in the list of jobs and makes it the current job. This button is located in the JOBS menu.

A job is a collection of information that is needed to submit and run Marc on the current model. A job contains the analysis class, analysis options, loadcases, and results specifications required for a Marc run. Multiple jobs may be defined and are stored in the list of currently defined jobs.

Keyboard Command Sequence:

    new_job

Command:  new_link

Description:  This command creates a new entry in the list of links and makes it the current links. This button is located in the LINKS->NODAL TIES, SERVO LINKS, and SPRINGS/DASHPOTS menus.

Keyboard Command Sequence:

    new_link
Command: **new_loadcase**

**Description:**
This command creates a new entry in the list of loadcases and makes it the current loadcase. This button is located in the LOADCASES menu. A loadcase is a collection of boundary conditions, analysis control parameters, and options that are to be used in a portion of a Marc analysis. Multiple loadcases may be defined and are stored in the list of currently defined loadcases.

**Keyboard Command Sequence:**

```
new_loadcase
```

---

Command: **new_material**

**Description:**
This command creates a new entry in the list of materials and makes it the current material. This button is located in the MATERIAL PROPERTIES menu.

A material is a collection of information that describes a material to be applied to elements in the model. Materials contain the material type, material properties, tables or functions that apply to those properties, and options that will be applied to the specified set of elements during analysis. Multiple materials may be defined and are stored in the list of currently defined materials.

**Keyboard Command Sequence:**

```
new_material
```
**Command:** new_model

**Description:** This command discards the current model and initializes a new model. This button is located in the FILES menu.

*Caution:* This command completely removes your current model (you may want to save the model before invoking this command).

**Keyboard Command Sequence:**

new_model

---

**Command:** new_orient

**Description:** This command creates a new entry in the list of orientations and makes it the current orientation. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

An orientation is the specification of directions and angles that define the orientation of materials.

**Keyboard Command Sequence:**

new_orient

---

**Command:** new_rbe2

**Description:** This command creates a new rbe2 and makes it the current rbe2. This button is located in the LINKS->RBE2'S menu.

**Keyboard Command Sequence:**

new_rbe2
Command: **new_rbe3**

**Description:** This command creates a new rbe3 and makes it the current rbe3. This button is located in the LINKS->RBE3'S menu.

**Keyboard Command Sequence:**

```
new_rbe3
```

---

Command: **new_rebar**

**Description:** This command creates a new rebar material and makes it the current material. This button is located in the MATERIAL PROPERTIES->LAYERED MATERIALS menu.

**Keyboard Command Sequence:**

```
new_rebar
```

---

Command: **new_section**

**Description:** This command creates a new entry in the list of beam sections and makes it the current section. This button is located in the GEOMETRIC PROPERTIES->MECHANICAL->BEAM SECTIONS menu.

A beam section is a set of dimensions and properties defining a beam cross-section. Sections may be referenced by geometric properties of type general beam.

**Keyboard Command Sequence:**

```
new_section
```
Command: **new_table**

**Description:** This command creates a new entry in the list of tables and makes it the current table. This button is located in the MATERIAL PROPERTIES->TABLES or any other link to the TABLES menu.

A table is a function of one variable, i.e. \( y = f(x) \), that may be applied to values specified by the user. Possible values include degree of freedom values in boundary condition applications and material property values. A table is defined by a series of two-dimensional data points. Multiple tables may be defined and are stored in the list of currently defined tables.

**Keyboard Command Sequence:**

```
new_table
```

---

Command: **new_transform**

**Description:** This command creates a new entry in the list of transforms and makes it the current transform. This button is located in the BOUNDARY CONDITIONS->MECHANICAL menu.

Transformations are used to convert nodal quantities from one coordinate system to another. Transformations are most commonly used to transform the orientation of point loads or fixed displacements, usually from global to local coordinates. Transformations contain the type of transformation, the data required by that transformation type, and a list of nodes the transformation applies to.

**Keyboard Command Sequence:**

```
new_transform
```
Subroutine: NEWSV
Description: The user subroutine NEWSV allows you to define the new values of any state variable for the end of the current step. This subroutine will be called in a loop over all the elements in the mesh when requested through the CHANGE STATE option.

Command: next_adapg
Description: This command selects the next entry in the list of remeshing criteria and makes it the current one. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

Keyboard Command Sequence: next_adapg

Command: next_adapt
Description: This command selects the next entry in the list of adapts and makes it the current adapt. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

An adapt is a collection of parameters that controls the adaptive meshing of a set of elements during an analysis.

Keyboard Command Sequence: next_adapt
Command: **next_apply**

Description: This command selects the next entry in the list of applications and makes it the current application. This button is located in the BOUNDARY CONDITIONS menu.

A boundary condition application is a set of information that completely defines a boundary condition. Each application contains the type of boundary condition, the degrees of freedom and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected. Multiple applications may be defined and are stored in the list of currently defined applications.

Keyboard Command Sequence:

```
next_apply
```

Command: **next_cavity**

Description: This command selects the next entry in the list of cavities and makes it the current cavity. It is located under the MESH GENERATION->CAVITIES menu.

Keyboard Command Sequence:

```
next_cavity
```

Command: **next_contact_area**

Description: This command selects the next entry in the list of contact areas and makes it the current contact area. This button is located in the CONTACT->CONTACT AREAS menu.

A contact area is a set of nodes that lie on a contact body that may potentially contact other contact bodies. Contact areas are used to reduce the computational cost of a contact analysis where it is known
for which nodes contact is possible. If no contact area is used, all exterior outline or surface nodes will be checked for contact.

**Keyboard Command Sequence:**

`next_contact_area`

**Command:** `next_contact_body`

**Description:** This command selects the next entry in the list of contact bodies and makes it the current contact body. This button is located in the CONTACT->CONTACT BODIES menu.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies, at least one of which must be a deformable body.

**Keyboard Command Sequence:**

`next_contact_body`

**Command:** `next_contact_table`

**Description:** This command selects the next entry in the list of contact tables and makes it the current contact table. This button is located in the CONTACT->CONTACT TABLES menu.

**Keyboard Command Sequence:**

`next_contact_table`
Command: next_crack
Description: This command selects the next entry in the list of cracks and makes it the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS and 3-D CRACKS menus.

Keyboard Command Sequence:
next_crack

Command: next_csect
Description: This command selects the next entry in the list of cross-sections and makes it the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:
next_csect

Command: next_descon
Description: This command selects the next entry in the list of design constraints and makes it the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

A design constraint is an analysis response quantity that is constrained to be above or below a specified value.

Keyboard Command Sequence:
next_descon
Command:  **next_desvar**

**Description:** This command selects the next entry in the list of design variables and makes it the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

A design variable is a quantity that can vary during an optimization or sensitivity analysis.

**Keyboard Command Sequence:**

```
next_desvar
```

---

Command:  **next_exseg**

**Description:** This command selects the next entry in the list of exclude segments and makes it the current one. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

```
next_exseg
```

---

Command:  **next_geometry**

**Description:** This command selects the next entry in the list of geometries and makes it the current geometry. This button is located in the GEOMETRIC PROPERTIES menu.

An element geometry is a collection of the geometric properties to be applied to a set of elements. Geometric properties include but are not limited to shell thickness, beam and truss areas, and beam moments of inertia.

**Keyboard Command Sequence:**

```
next_geometry
```
Command: **next_i cond**

**Description:** This command selects the next entry in the list of initial conditions and makes it the current initial condition. This button is located in the INITIAL CONDITIONS menu.

A initial condition is a boundary condition that is applied to the mesh at the beginning of the analysis. Each initial condition contains the degrees of freedom and their corresponding applied values, optional tables or functions that apply to the values, and the geometric and FEM entities affected.

**Keyboard Command Sequence:**

`next_i cond`

---

Command: **next_insert**

**Description:** This command selects the next entry in the list of inserts and makes it the current insert.

**Keyboard Command Sequence:**

`next_insert`
Command: next_job
Description: This command selects the next entry in the list of jobs and makes it the current job. This button is located in the JOBS menu.

A job is a collection of information that is needed to submit and run Marc on the current model. A job contains the analysis class, analysis options, loadcases, and results specifications required for a Marc run. Multiple jobs may be defined and are stored in the list of currently defined jobs.

Keyboard Command Sequence:
next_job

Command: next_link
Description: This command selects the next entry in the list of links and makes it the current link. This button is located in the LINKS->NODAL TIES, SERVO LINKS, and SPRINGS/DASHPOTS menus.

Keyboard Command Sequence:
next_link

Command: next_loadcase
Description: This command selects the next entry in the list of loadcases and makes it the current loadcase. This button is located in the LOADCASES menu.

A loadcase is a collection of boundary conditions, analysis control parameters, and options that are to be used in a portion of a Marc analysis. Multiple loadcases may be defined and are stored in the list of currently defined loadcases.

Keyboard Command Sequence:
next_loadcase
Command: next_material
Description: This command selects the next entry in the list of materials and makes it the current material. This button is located in the MATERIAL PROPERTIES menu.

A material is a collection of information that describes a material to be applied to elements in the model. Materials contain the material type, material properties, tables or functions that apply to those properties, and options that will be applied to the specified set of elements during analysis. Multiple materials may be defined and are stored in the list of currently defined materials.

Keyboard Command Sequence:

next_material

Command: next_orient
Description: This command selects the next entry in the list of orientations and makes it the current orientation. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

An orientation is the specification of directions and angles that define the orientation of materials.

Keyboard Command Sequence:

next_orient
Command: **next_rbe2**

**Description:** This command selects the next entry in the list of rbe2’s and makes it the current one. This button is located in the LINKS->RBE2’S menu.

**Keyboard Command Sequence:**

```
next_rbe2
```

Command: **next_rbe3**

**Description:** This command selects the next entry in the list of rbe3’s and makes it the current one. This button is located in the LINKS->RBE3’S menu.

**Keyboard Command Sequence:**

```
next_rbe3
```

Command: **next_section**

**Description:** This command selects the next entry in the list of beam sections and makes it the current section. This button is located in the GEOMETRIC PROPERTIES->MECHANICAL->BEAM SECTIONS menu.

A beam section is a set of dimensions and properties defining a beam cross-section. Sections may be referenced by geometric properties of type general beam.

**Keyboard Command Sequence:**

```
next_section
```
Command: **next_table**

**Description:** This command selects the next entry in the list of tables and makes it the current table. This button is located in the MATERIAL PROPERTIES->TABLES or any other link to the TABLES menu.

A table is a function of one variable, i.e. \( y = f(x) \), that may be applied to values specified by the user. Possible values include degree of freedom values in boundary condition applications and material property values. A table is defined by a series of two-dimensional data points. Multiple tables may be defined and are stored in the list of currently defined tables.

**Keyboard Command Sequence:**

```plaintext
next_table
```

---

Command: **next_transform**

**Description:** This command selects the next entry in the list of transforms and makes it the current transform. This button is located in the BOUNDARY CONDITIONS->MECHANICAL menu.

Transformations are used to convert nodal quantities from one coordinate system to another. Transformations are most commonly used to transform the orientation of point loads or fixed displacements, usually from global to local coordinates. Transformations contain the type of transformation, the data required by that transformation type, and a list of nodes the transformation applies to.

**Keyboard Command Sequence:**

```plaintext
next_transform
```
Command:  **numeric_format_aut**  

**Description:** This button toggles the type of format to display for NUMERICS. Available formats are:

- **AUTOMATIC** (command `numeric_format_aut`)
  - The format will be the default and the exponent will be displayed.

- **EXPONENTIAL** (command `numeric_format_exp`)
  - The mantissa precision to the right of the decimal may be specified using the `PRECISION (numeric_prec)` command. There will be one digit to the left of the decimal point and the exponent will be displayed.

- **FLOATING** (command `numeric_format_flt`)
  - The exponent will not be displayed. The `PRECISION (numeric_prec)` command is used to adjust the number of digits displayed.

- **INTEGER** (command `numeric_format_int`)
  - The values displayed will be integer values. The values will be rounded to the nearest integer.

This button is located in the **RESULTS->SCALAR PLOT->SETTINGS->NUMERICS** menu.

**Keyboard Command Sequence:**

- `numeric_format_aut`
- `numeric_format_exp`
- `numeric_format_flt`
- `numeric_format_int`
Command: numeric_prec

Description: This command specifies the number of digits to the right of the decimal point for the following FORMAT commands:

- EXPONENTIAL (numerics_format_exp)
- FLOATING (numerics_format_flt)

The value specified must be in the range of 0 - 10.

This button is located in the RESULTS->SCALAR PLOT->SETTINGS->NUMERICS menu.

Keyboard Command Sequence:

numeric_prec <value>
Mentat Help Commands in O

Command: **xcv_model ogden**

**Description:** This command is used if experimental data must be fitted using the Ogden strain energy function \( W \), which is given by:

\[
W = \sum_{n=1}^{N} \left( \frac{\mu_n}{\alpha_n} \right) \left( J^\left(\frac{-\alpha_n}{3}\right) \right) \left( \lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3 \right) + 4.5K \left( J^{(1/3)} - 1 \right)^2
\]

where \( \lambda_1, \lambda_2, \) and \( \lambda_3 \) are the principal stretch ratios, \( J \) is the determinant of the deformation gradient, \( N \) is the number of terms and \( \mu_n, \alpha_n, \) and \( K \) are the material parameters to be determined.

The maximum number of terms is 10, but it is recommended to use no more terms than necessary to get a sufficiently good fit.

This model can be used for incompressible as well as for slightly compressible elastic materials. Compressibility is included based on a constant bulk modulus. In case of compressibility, volumetric information
is needed, preferably using a volumetric test, but volumetric data can also be included for uniaxial, biaxial, and planar shear tests.

In order to perform a plausible extrapolation for the compressible Ogden model, dilatational information is needed beyond the data set. This is achieved using linear extrapolation based on the two start and/or end points of the measured data. This linear extrapolation may restrict the validity of the response outside the range of the measured data.

For dual mode plotting (except for simple shear), dilatational information is needed for the compressible Ogden model. For a volumetric test, this readily follows from the strain, but for uniaxial, biaxial, and planar shear tests, this must be calculated. This calculation is based on the requirement that the stress in perpendicular direction must be zero. If the fitted coefficients do not fulfil this requirement, zero stresses are returned for such a dual mode.

**Keyboard Command Sequence:**
```
xcv_model ogden
```

---

**Command:** **open_model**

**Description:** This command reads a Mentat database file and imports the model. You must specify the model name.

*Caution:* Opening a model file while you are currently editing a model will cause your current model to be discarded. Hence, you should save your existing model before executing this command.

**Keyboard Command Sequence:**
```
open_model <model name>
```

---

**Command:** **opt_plot_filled**

**Description:** This command causes areas under opt_plot curves be filled with color. To unfilled plots, use the opt_plot_unfilled command.

**Keyboard Command Sequence:**
```
opt_plot_filled
```
**FIT**

Command: `opt_plot_fit`

Description: This command alters the limits of the response gradient’s result plot so that all values fit onto the screen.

Keyboard Command Sequence:

```
opt_plot_fit
```

**PLOT TITLE**

Command: `set_opt_plot_title`

Description: This command sets the title of a plot.

Keyboard Command Sequence:

```
set_opt_plot_title <title>
```

**X-AXIS NAME**

Command: `set_opt_plot_xname`

Description: This command sets the name of X-axis.

Keyboard Command Sequence:

```
set_opt_plot_xname <name>
```

**Y-AXIS NAME**

Command: `set_opt_plot_yname`

Description: This command sets the name of Y-axis.

Keyboard Command Sequence:

```
set_opt_plot_yname <name>
```
Subroutine: ORIENT

Description: The user subroutine ORIENT allows you to supply a preferred orientation so that ANELAS, HOOKLW, ANPLAS, and ANKOND may supply anisotropic material constants in this orientation.

Command: orient_cylindrical

Description: This command creates a separate orientation of type 3d_aniso for each of the specified elements and determines the user-defined vectors from the two specified points representing the axis of a cylindrical coordinate system.

The specified elements should be 3-D elements.

Keyboard Command Sequence:

```
orient_cylindrical
   <X, Y, Z coordinates of first axis point>
   <X, Y, Z coordinates of second axis point>
   <3-D element list> #
```
Command: **orient_local**

Description: This command creates a separate orientation of type `3d_aniso` for each of the specified elements and determines the user-defined vectors from the two specified points representing the axis of a local coordinate system. The resulting orientations are initialized to be aligned with the local coordinate system of each element, and then rotated about the local X, Y, and Z axes by the given angles.

The specified elements should be 3-D elements.

Keyboard Command Sequence:

```
orient_local <X, Y, Z rotations>
<3-D element list> #
```

Command: **orient_name**

Description: This command sets or changes the name of the current orientation. An orientation is the specification of directions and angles that define the orientation of materials.

Keyboard Command Sequence:

```
orient_name <orientation name>
```

Command: **orient_reset**

Description: This command resets the type, angle, and user-defined vectors of the current orientation to their default values.

Keyboard Command Sequence:

```
orient_reset
```
Command: orient_rotate

Description: This command rotates the current orientation by the specified rotations and should only be used for orientations of type 3d_aniso.

Keyboard Command Sequence:
orient_rotate <X, Y, Z axis rotation in degrees>

Command: orient_type

Description: This command sets the type of the current orientation. Orientation type determines the orientation directions and the type of data required from the user. The valid orientation types are as follows:

- edge12: Edge types, for 2-D continuum and shell elements. The direction vector along the specified edge is projected onto the surface tangent plane (x-y plane if continuum, V1-V2 plane if shell) at each integration point. The first preferred direction is given by a rotation about the surface normal (z axis if continuum, V3 axis if shell) equal to the orientation angle. The third preferred direction is given by the surface normal, and the second preferred direction is given by the cross product of the third and first directions.

- edge23
- edge34
- edge41

- xy_plane: Global intersecting plane types, for 2-D elements. The specified global coordinate plane is intersected with the surface tangent plane. The first preferred direction is given by a rotation about the surface normal from this intersection equal to the orientation angle. The third preferred direction is given by the surface normal, and the second preferred direction is given by the cross product of the third and first directions.
xu_plane  User-defined intersecting plane types, for 2-D elements. For types xu_, yu_, and zu_plane, the plane determined by the coordinate direction and a user-defined vector is intersected with the surface tangent plane. For type uu_plane, the plane determined by 2 user-defined vectors is intersected with the surface tangent plane. The first preferred direction is given by a rotation about the surface normal from this intersection equal to the orientation angle. The third preferred direction is given by the surface normal, and the second preferred direction is given by the cross product of the third and first directions.

yu_plane  
zu_plane  
uu_plane  

3d_aniso  3-D type for 3-D elements. The first and second preferred directions are as given by the user. The third preferred direction is given by the cross product of the first and second directions.

usersub  UORIENT user subroutine type for all types of elements. Marc will call UORIENT to obtain the transformation matrix between global coordinates (for continuum elements) or local coordinates (for beams, plates, or shells).

Keyboard Command Sequence:

orient_type  <orient type name>

Other Buttons with the same description:

EDGE12  
EDGE23  
EDGE31  
EDGE34  
EDGE41  
USUB ORIENT  
UU PLANE  
XU PLANE  
XY PLANE  
YU PLANE  
YZ PLANE  
ZU PLANE  
ZX PLANE
Command:  **origin**

**Description:** This command sets the XYZ coordinates of the origin of the coordinate system to the specified values. To change the X, Y, or the Z coordinates only, use the **origin_x**, **origin_y**, or the **origin_z** command, respectively.

**Keyboard Command Sequence:**

```
origin <X coordinate> <Y coordinate> <Z coordinate>
```

---

Command:  **origin_x**

**Description:** This command sets the X coordinate of the origin of the coordinate system to the specified values. To change all three X, Y, and Z coordinates use the **origin** command. To change just the Y or Z coordinate, use the **origin_y** or **origin_z** command, respectively.

**Keyboard Command Sequence:**

```
origin_x <X coordinate>
```

---

Command:  **origin_y**

**Description:** This command sets the Y coordinate of the origin of the coordinate system to the specified values. To change all three X, Y, and Z coordinates use the **origin** command. To change just the X or Z coordinate, use the **origin_x** or **origin_z** command, respectively.

**Keyboard Command Sequence:**

```
origin_y <Y coordinate>
```
Command:  origin_z
Description:  This command sets the Z coordinate of the origin of the coordinate system to the specified values. To change all three X, Y, and Z coordinates use the origin command. To change just the X or Y coordinate, use the origin_x or origin_y command, respectively.

Keyboard Command Sequence:

origin_z <Z coordinate>

Menu:  OVERLAY
Description:  The OVERLAY mesh generation processor contains menus which generate meshes over curves in the plane and curves trimming surfaces (formerly in AUTOMESH).

Command:  overlay_mesh
Description:  This command generates a mesh on a two-dimensional boundary geometry. The area to be meshed is defined by a list of curves which lie in the XY plane of the local coordinate system, and enclose a nonzero area in that plane. The area may be multiply connected, that is, it may have holes in it.

Keyboard Command Sequence:

overlay_mesh <curve list> #
Mentat Help Commands in P

### ADD CURVE

**Command:** `pathplot_add`

**Description:** This command adds a single curve to a path plot. You must specify the variables for the x and y axes. To specify the nodes to plot values for, use the `set_pathplot_path` command.

**Keyboard Command Sequence:**

`pathplot_add <x variable> <y variable>`

### ADD VARIABLE

**Command:** `pathplot_add_var`

**Description:** This command adds a family of curves of all variables in the results file for the current node path. You must specify the x-axis variable. To specify the nodes to plot values for, use the `set_pathplot_path` command.

**Keyboard Command Sequence:**

`pathplot_add_var <x variable>`
CLEAR CURVES

Command: `pathplot_clear`
Description: This command removes all of the path plot curves.
Keyboard Command Sequence:
```
pathplot_clear
```

FILLED

Command: `pathplot_filled`
Description: This command causes areas under path plot curves to be filled with color. Toggle this button ON to use the `pathplot_filled` command, or OFF to use the `pathplot_unfilled` command.
Keyboard Command Sequence:
```
pathplot_filled
```

FIT

Command: `pathplot_fit`
Description: This command alters the limits and scales of path plots so that all values fit onto the screen.
Keyboard Command Sequence:
```
pathplot_fit
```

REMOVE CURVE

Command: `pathplot_remove`
Description: This command removes the specified path plot curves. Curves are specified by number.
Keyboard Command Sequence:
```
pathplot_remove <curve number>
```
**CONVERT**

**Command:** `pathplot_table`

**Description:** This command converts a path plot curve into a table. You must specify the number of the table to be converted.

**Keyboard Command Sequence:**

```
pathplot_table <curve number>
```

**FILLED**

**Command:** `pathplot_unfilled`

**Description:** This command causes areas under path plot curves to be unfilled. Toggle this button ON to use the `pathplot_filled` command, or OFF to use the `pathplot_unfilled` command.

**Keyboard Command Sequence:**

```
pathplot_unfilled
```

**SAVE**

**Command:** `pathplot_write`

**Description:** This command writes a path plot to a file. You must specify the name of the file.

**Keyboard Command Sequence:**

```
pathplot_write <file name>
```

**pause**

**Command:** `pause`

**Description:** The command causes the program to pause for the specified number of seconds. This is useful when executing procedure files.

**Keyboard Command Sequence:**

```
pause <seconds>
```
Command: **pick_aperture**

**Description:** This command sets the pick aperture. The pick aperture is the area of an imaginary square that surrounds the graphics cursor, within which you can pick graphical entities shown in MSC.Marc Mentat’s graphics area. You must specify the size of the pick aperture in pixels.

**Keyboard Command Sequence:**

```
pick_aperture <aperture>
```

---

Command: **pick_inside_complete**

**Description:** This command specifies that graphical entities are selectable only if they are completely inside the pick box. Use the `pick_inside_partial` command to specify that selectable entities need only be partially inside the pick box.

**Keyboard Command Sequence:**

```
pick_inside_complete
```

---

Command: **pick_inside_partial**

**Description:** This command specifies that graphical entities are selectable if they are even only partially inside the pick box. Use the `pick_inside_complete` command to specify that selectable entities must be completely inside the pick box.

**Keyboard Command Sequence:**

```
pick_inside_partial
```
Command:  **PIEZO_ELECTRIC**  *(piezo_electric)*

Description:  This command sets the piezo-electric coupling matrix. This matrix can be either stress based or strain based. Stress based means that it is the matrix \( e \) occurring in the following equation:

\[
\text{stress} = L \times \text{strain} - e \times \text{Electric\_field\_intensity}
\]

And strain based means that it is the \( d \) matrix occurring in the following equation:

\[
\text{strain} = \text{INV}(L) \times \text{stress} + d \times \text{Electric\_field\_intensity}
\]

The dielectric properties of piezoelectric material can be set in the electrostatic material section. These properties also depend on whether they are stress based or strain based, and are set with the same flag.

Keyboard Command Sequence:

```
piezo_electric:matrix:stress_based
piezo_electric:matrix:strain_based
```

Another Button with the same description:

**STRAIN-BASED**

---

Command:  **PIEZO_ELECTRIC**  *(piezo_electric)*

Description:  This command is used to set the piezo-electric coupling matrix. A piezo-electric material requires the mechanical, electrostatic, and piezo-electric properties to be set. The *mechanical properties* can be isotropic, orthotropic, or anisotropic. The *electrostatic properties* can be either isotropic or orthotropic, and are either stress based or strain based. The piezo-electric properties are either stress based or strain based.

Keyboard Command Sequence:

```
material\_type piezo\_electric
```
STRESS-BASED COUPLING MATRIX $E_{[i,j,k]}$

Command: PIEZO_ELECTRIC (piezo_electric)

Description: This command sets the values for piezo-electric coupling matrix. The $i$, $j$ components are related to the mechanical part, and the $k$ component is related to the electrical part.

Keyboard Command Sequence:

```
material_value piezo_electric:eijk
```

STRAIN-BASED COUPLING MATRIX $D_{[i,j,k]}$

Command: PIEZO_ELECTRIC (piezo_electric)

Description: This command sets the values for piezo-electric coupling matrix. The $i$, $j$ components are related to the mechanical part, and the $k$ component is related to the electrical part.

Keyboard Command Sequence:

```
material_value piezo_electric:dijk
```

COMPUTE

Command: material_value

Description: This command will set a property value for Barlat’s (1991) Yield Function. The exponent $m$ is mainly associated with the crystal structure. A higher $m$ value has the effect of decreasing a radius of curvature of the rounded vertices of the yield surface, so that this criterion can present the small radius of curvature of the yield surface near the uniaxial and balanced biaxial tension ranges. The yield surface becomes convex, when $m$ is greater than 1.

Recommended values of the exponent for this criterion are $m=8$ for FCC (aluminum) materials and $m=6$ for BCC (steel) materials. For von Mises yield function, $m=2$ and Tresca yield function is $m=1$ or infinite.

In order to use this yield function, the experimentally measured $Y_0, Y_{45}, Y_{90}$ and $Y_b$ are required. $Y_0, Y_{45}, Y_{90}$ and $Y_b$ represent the yield stresses at 0, 45, 90 and biaxial directions, respectively. Usually $Y_b$ is measured from bulge test. If $Y_b$ is not available, usually take $Y_b/Y_0 = 1.0$. 
If \( Y_{45}/Y_0 = Y_{90}/Y_0 = Y_{b}/Y_0 = 1 \) with \( m = 2 \), this criterion becomes von-Mises yield function.

If \( Y_{45}/Y_0 = Y_{90}/Y_0 = Y_{b}/Y_0 = 1 \) with \( m = 1 \), this criterion becomes Tresca yield function.

**Keyboard Command Sequence:**

```plaintext
material_value plasticity:barlat_m <value>
maintenance_value plasticity:barlat_y45 <value>
maintenance_value plasticity:barlat_y90 <value>
maintenance_value plasticity:barlat_yb <value>
```

Other Buttons with the same description:

- M
- C1
- C2
- C3
- C6
- Y90/Y0
- Yb/Y0
- Y45/Y0

**Command:** `material_value`

**Description:** This command will set a property value for Hill's (1948) Yield Function. The definition of anisotropic coefficients for Hill's (1948) yield function is shown in *MSC.Marc Volume A: Theory and User Information*, Chapter 7. If all anisotropic constants are 1, it becomes von Mises yield function.

**Keyboard Command Sequence:**

```plaintext
material_value plasticity:yrdir1 <value>
maintenance_value plasticity:yrdir2 <value>
maintenance_value plasticity:yrdir3 <value>
maintenance_value plasticity:yrshr1 <value>
maintenance_value plasticity:yrshr2 <value>
maintenance_value plasticity:yrshr3 <value>
```

Other Buttons with the same description:

- r0
- r45
- r90
- Y45/Y0
- Y90/Y0
- YRDIR1
- YRDIR2
Command: **material_value**

Description: This command will set the fraction factor of kinematic hardening in combined hardening rule.

- If $f = 0$, it means isotropic hardening.
- If $f = 1$, it means pure-kinematic hardening.
- If $0 < f < 1$, it means combined hardening.

The fraction $f$ can be used with Hill's and Barlat's yield criterion only. Otherwise, combined hardening takes the last slope of stress-strain curve automatically.

Keyboard Command Sequence:

```
material_value plasticity:kinematic_fraction
<value>
```
**HILL'S (1948) CRITERION**

Command: `material_option`

Description: This command selects Hill's (1948) Yield Function for the plasticity material. The quadratic yield function by Hill has long been the most popular choice to present planar anisotropy. The criterion is being successfully applied for finite element modeling of anisotropic sheets, particularly for steel. Because the function cannot describe anomalous behavior of aluminum alloy, please don't use the criterion for aluminum. For aluminum alloy, Barlat's yield function is recommended.

**Keyboard Command Sequence:**

```
material_option plasticity:yield:hill
```

---

**PLAY AVI**

Command: `play_avi`

Description: This command will play the AVI movie created with the `avi_animation_make (MAKE AVI MOVIE)` command.

Also see: `avi_animation_make (MAKE AVI MOVIE)`.

**Keyboard Command Sequence:**

```
play_avi
```

---

**PLAY MPEG**

Command: `play_mpeg`

Description: This command will play the MPEG movie created with the `mpeg_animation_make (MAKE MPEG MOVIE)` command.

Windows NT

It runs the `mpeg_window.bat` script located in the bin directory which defaults to using the application associated with the `.mpeg` extension for Windows NT. The script contains a counter and will loop and sleep for a maximum time period of 5 minutes. The counter limit variable MAXCNT can be modified for a shorter or longer limit.
Unix

In runs the script `mpeg_window` which is in the bin directory, which must be modified to use the application on your system that will play the MPEG movie.

Also see: `mpeg_animation_make` (MAKE MPEG MOVIE).

**Keyboard Command Sequence:**

```
play_mpeg
```

**Menu:** PLOT

**Description:** This menu contains commands for controlling the display of model entities.

**Command:** plot_reset

**Description:** This command resets all plotting parameters to their default values.

**Keyboard Command Sequence:**

```
plot_reset
```

**Subroutine:** PLOTV

**Description:** The user subroutine PLOTV is used in conjunction with either code 19 or a negative code in either the CONTOURS option of the mesh display plotting to allow contouring of user defined variables, or with the POST option. The latter case allows you to define a variable to be written to the POST file.
HARMONICS

Command: post_animate_harmonic
Description: This command saves a series of animation files representing a single harmonic shape. This is done by incrementing the phase angle used to view the harmonic until one full cycle is completed. You must specify the number of frames for the cycle.

Keyboard Command Sequence:
post_animate_harmonic <number of frames>

INCREMENTS

Command: post_animate_increments
Description: This command collects and saves a series of increments contained in a post file for the purpose of animating them at a later time. The first increment saved is the current increment. You must specify the number of increments to save and the increment step size.

Keyboard Command Sequence:
post_animate_increments <number of increments> <step size>

MODE

Command: post_animate_mode
Description: This command saves a series of animation files representing a single mode shape. This is done by applying a sinusoidal magnification factor to the current displacements. You must specify the number of frames to save.

Keyboard Command Sequence:
post_animate_mode <number of frames>
**BEAM CONTOURS**

**Command:**  `post_beam_contour`

**Description:** This command sets the results plotting style to contoured beams and element edges. Each contour represents element values that are within a certain range.

**Keyboard Command Sequence:**

```
post_beam_contour
```

---

**BEAM VALUES**

**Command:**  `post_beam_value`

**Description:** This command sets the results plotting style to beam integration point values. The colors of the symbols and letters from the alphabet indicate the amplitude of the results quantity at the integration points.

**Keyboard Command Sequence:**

```
post_beam_value
```

---

**CLOSE**

**Command:**  `post_close`

**Description:** This command closes the current results file and turns off all post plotting. The model existing prior to opening the post file is restored.

**Keyboard Command Sequence:**

```
post_close
```
CONTOUR BANDS

Command: post_contour_bands

Description: This command sets the results plotting style to contour bands. Contour bands are areas of interpolated nodal values that are within a certain range.

Keyboard Command Sequence:

post_contour_bands

CONTOUR CENT

Command: post_contour_centroid

Description: This command sets the results plotting style to centroid contours. Centroid contours are created from element centroidal values.

Keyboard Command Sequence:

post_contour_centroid

CONTOUR LINES

Command: post_contour_lines

Description: This command sets the results plotting style to contour lines. Contour lines show lines along paths of equal interpolated nodal values.

Keyboard Command Sequence:

post_contour_lines

CUTTING PLANES

Command: post_cutting_planes

Description: This command sets the results plotting style to cutting planes. This mode shows bands of equal interpolated nodal values on intersecting planes.

Keyboard Command Sequence:

post_cutting_planes
**ADD**

**Command:** post_elements

**Description:** This command indicates which elements should be post plotted when a post plotting option is turned on. This button is located in the RESULTS->MORE menu.

**Keyboard Command Sequence:**

post_elements <element list>

---

**AVERAge**

**Menu:** EXTRAPOLATION

**Description:** These commands control the manner in which element integration point data is extrapolated to the nodes of an element. In addition they control the inter-element averaging of the nodal data after it has been extrapolated.

- post_extrap_linear
  - Extrapolate by averaging the integration points to the centroid of the element and then doing a linear extrapolation from the centroid through the integration point to the node.

- post_extrap_translate
  - Actually do not extrapolate, but rather copy the data at each integration point to its corresponding node. In those cases where there are fewer integration points than nodes, some averaging of neighboring integration points is done.

- post_extrap_average
  - The average of all the integration points is computed and assigned to the nodes. Therefore, all nodes have an equal value assigned to them.

post_nodal_averaging <on/off>

This command toggles the averaging of element values at the nodes. Averaging is done to assure that contour lines are continuous. When it is turned off, each element is independently contoured and contour lines usually will appear discontinuous.

Other Buttons with the same description:

EXTRAPOLATION
LINEAR
HISTORY PLOT

Menu: HISTORY PLOT
Description: This menu contains commands for creating two-dimensional history plots. History plots are generally used to show the variation of results values with analysis increments.

ISOLATE ELEMENTS

Command: post_isolate
Description: This command prevents the averaging of element values across element edges. This is usually used when there is a discontinuity in a mesh.

Keyboard Command Sequence:
```
post_isolate <element list> #
```

ISO-SURFACES

Command: post_isosurfaces
Description: This command sets the results plotting style to iso-surfaces. Iso-surfaces are surfaces with equal interpolated nodal values.

Keyboard Command Sequence:
```
post_isosurfaces
```

post_list

Command: post_list
Description: This command lists the results quantities available for the current load increment or subincrement. Different types of analyses will have different types of results quantities available for display. Use the post_value command to set the results value you wish to see plotted.

Keyboard Command Sequence:
```
post_list
```
BEAM TO BEAM CONTACT

Command: **post_merge_bbc**

Description: This command automatically merges the formatted MSC.Marc Mentat model file that contains the beam-to-beam contact location points for the current increment with the current post-processing model, if such a file exists.

The command can be used in a procedure file that is loaded with the "post_procedure_file" command, to automatically merge the appropriate file when a new increment is read from the post file. Note that in such a procedure file, the "clear_geometry" command should be called prior to this command to remove any points or curves that were merged from a previous increment.

Also see: `job_option write_contact_bm_bm`,
`job_param bm_bm_files_freq`,
`set_post_procedure`, and
`post_procedure_file`

Keyboard Command Sequence:

`post_merge_bbc`

ANALYTICAL DESCRIPTION

Command: **post_merge_spline**

Description: This command automatically merges the formatted MSC.Marc Mentat model file that contains the spline description of analytical deformable contact bodies for the current increment with the current post-processing model, if such a file exists.

The command can be used in a procedure file that is loaded with the "post_procedure_file" command, to automatically merge the appropriate file when a new increment is read from the post file. Note that in such a procedure file, the "clear_geometry" command should be called prior to this command to remove any points, curves or surfaces that were merged from a previous increment.

Also see: `job_write_spline_files`,
`job_param spline_files_freq`,

set_post_procedure, and
post_procedure_file

**Keyboard Command Sequence:**

post_merge_spline

---

**MONITOR**

**Command:** post_monitor

**Description:** This command specifies that results will be continuously read from a results file that is being produced by a currently running job. As results become available, they will automatically be read from the current open post file and displayed. The command can be interrupted by pressing the escape (Esc) key.

**Keyboard Command Sequence:**

post_monitor

---

**NEXT**

**Command:** post_next

**Description:** This command reads the results of the next load increment or subincrement into memory so they can be displayed. This button is located in the RESULTS menu.

No action is taken if the current increment is the last increment in the file.

**Keyboard Command Sequence:**

post_next
ADD

Command: post_nodes
Description: This command indicates which nodes should be post plotted when a post plotting option is turned ON (for Vectors, Symbols and Numerics). Normally this is controlled by the post_elements command, however, this command allows individual nodes to be controlled. By default, all nodes have been ADDed. Therefore, to view specific nodes, use the REM button to remove the nodes that are not to be displayed. This button is located in the RESULTS->MORE menu.

Note: Once a node has been removed, it cannot be selected. Using the ADD button will require the node numbers to be typed in.

Keyboard Command Sequence:
post_nodes <node list>

NUMERICS

Command: post_numerics
Description: This command sets the results plotting style to numerics. Numerical values are shown at nodes.

Keyboard Command Sequence:
post_numerics

OFF

Command: post_off
Description: This command turns off the plotting of results values. This button is located in the RESULTS menu.

Keyboard Command Sequence:
post_off
Command: post_open
Description: This command opens a file containing Marc results and makes it available for use in postprocessing. This button is located in the RESULTS menu.
Marc results are either ASCII or binary files produced by Marc that contain results for every increment and subincrement in an analysis.

Keyboard Command Sequence:
post_open <file name>

Command: post_open_default
Description: This command opens the post file associated with the current job. This button is located in the RESULTS menu.

Keyboard Command Sequence:
post_open_default

Menu: RESPONSE GRAD/DESIGN VAR
Description: Pressing this button brings up the menu for plotting response gradients or element contributions to response (as obtained from sensitivity analysis), or the values of design variables at various cycles of design optimization. With this menu, such information can be plotted as bar charts or line graphs.
For contour plots of response gradients or element contributions, the contour bands (and similar) capabilities of MSC.Marc Mentat can be used.
**PATH PLOT**

**Menu:** PATH PLOT

**Description:** This menu contains commands for creating two-dimensional path plots. Path plots are generally used to show the variation of results values with position in the model.

**IMAGINARY PART**

**Command:** `post_phase`

**Description:** Sets the phase angle for plotting harmonic results. The angle should be given in degrees and is used when computing the displacements of nodes.

**Keyboard Command Sequence:**

```
post_phase <angle>
```

**Other Buttons with the same description:**

- PHASE ANGLE PHI
- REAL PART

**PREV**

**Command:** `post_prev`

**Description:** This command reads the results of the previous load increment or subincrement into memory so they can be displayed. This button is located in the RESULTS menu.

No action is taken if the current increment is the first increment in the file.

**Keyboard Command Sequence:**

```
post_prev
```
**MIN PRINC VAL**

**Command:**  post_principal_1  

**Description:** This command turns on results plotting of the first principal current tensor quantity vector for each element. At the centroid of each post-plotted element, only the first principal current tensor quantity vector will be displayed. The first principal current tensor quantity vector represents the principal current tensor quantity with the lowest eigenvalue. A small star is drawn at the end of each vector in the direction it is pointing.

If the vector represents compression, it is displayed in blue and points toward the element centroid.

If the vector represents tension, it is displayed in red and points away from the element centroid.

**Keyboard Command Sequence:**

`post_principal_1`

---

**INT PRINC VAL**

**Command:**  post_principal_2  

**Description:** This command turns on results plotting of the second principal current tensor quantity vector for each element. At the centroid of each post-plotted element, only the second principal current tensor quantity vector will be displayed. The second principal current tensor quantity vector represents the principal current tensor quantity with the median eigenvalue. A small star is drawn at the end of each vector in the direction it is pointing.

If the vector represents compression, it is displayed in blue and points toward the element centroid.

If the vector represents tension, it is displayed in red and points away from the element centroid.

**Keyboard Command Sequence:**

`post_principal_2`
Command: **post_principal_3**

**Description:** This command turns on results plotting of the third principal current tensor quantity vector for each element. At the centroid of each post-plotted element, only the third principal current tensor quantity vector will be displayed. The third principal current tensor quantity vector represents the principal current tensor quantity with the highest eigenvalue. A small star is drawn at the end of each vector in the direction it is pointing.

If the vector represents compression, it is displayed in blue and points toward the element centroid.

If the vector represents tension, it is displayed in red and points away from the element centroid.

**Keyboard Command Sequence:**

```
post_principal_3
```

Command: **post_principal_all**

**Description:** This command turns on results plotting of all three principal current tensor quantity vectors for each element. At the centroid of each post-plotted element, all three principal current tensor quantity vectors will be displayed. The three vectors represent the first, second, and third principal vectors of the current tensor quantity, and are in order of increasing eigenvalue. The first principal current tensor quantity vector is displayed in red, the second is green, and the third is yellow. A small star is drawn at the end of each vector in the direction it is pointing.

If the vector represents compression, it points toward the element centroid.

If the vector represents tension, it points away from the element centroid.

**Keyboard Command Sequence:**

```
post_principal_all
```
Command: **post_principal_major**

**Description:** This command turns on results plotting of the largest principal current tensor quantity vector, in absolute magnitude (length), for each element. At the centroid of each post-plotted element, only the major principal current tensor quantity vector will be displayed. A small star is drawn at the end of each vector in the direction it is pointing.

If the vector represents compression, it is displayed in blue and points toward the element centroid.

If the vector represents tension, it is displayed in red and points away from the element centroid.

**Keyboard Command Sequence:**

```
post_principal_major
```
Store the above contents in a file and specify it with this command. Then run the command *set_post_procedure on which is located just above and to the right of this command to active it.

Whenever a new increment is read from the post file, the contents of the procedure file is executed, such as for the post_next, post_previous, post_skip_to, post_monitor, or post_rewind commands.

Since one of the primary uses of this is to create a 3-D model from a 2-D analysis, any mesh that is created from the previous increment must be cleared when the next increment is read.

**Keyboard Command Sequence:**

```
post_procedure_file <file name>
```
Command: **post_scan**

Description: This command displays the currently available increments and subincrements available on the post file.

Keyboard Command Sequence:

```
post_scan
```

Another Button with the same description:

UPDATE

---

Command: **post_show_nodes**

Description: This command prints the current deformations, scalar, and vector plotting quantities at the given node.

Keyboard Command Sequence:

```
post_show_nodes <node>
```

---

Command: **post_skip**

Description: This command skips the specified number of load increments or subincrements in the current results file and reads the next increment.

If the specified number of increments to skip is greater than the number of increments remaining in the file, the file is positioned to the last increment.

Keyboard Command Sequence:

```
post_skip <number of increments>
```
SCAN

Command: `post_skip_to`
Description: This command skips to the specified increment or subincrement in a post file. (Subincrements are specified by appending a colon and the subincrement number immediately after the increment number, as in 10:4.)

Keyboard Command Sequence:
```
post_skip_to <increment number>
  <subincrement number>
```

Another Button with the same description:
SKIP TO INC

LAST

Command: `post_skip_to_last`
Description: This command skips to the last increment in a post file. If the post file is still being written, it skips to the last increment that is currently in the post file.

Keyboard Command Sequence:
```
post_skip_to_last
```

SYMBOLS

Command: `post_symbols`
Description: This command sets the results plotting style to symbols. The colors of the symbols indicate the amplitude of the results quantity at the nodes.

Keyboard Command Sequence:
```
post_symbols
```
Command: \texttt{post\_tensor}

Description: This command sets the results tensor plotting quantity.

Keyboard Command Sequence:

\texttt{post\_vector <tensor quantity>}

---

Command: \texttt{post\_tensor\_all\_layers}

Description: This command sets all layers for a selected post tensor. Values for this tensor will be written at all layers. The number of layers per shell element is set with the command \texttt{job\_param layers}.

Keyboard Command Sequence:

\texttt{post\_tensor\_all\_layers}

---

Command: \texttt{post\_tensor\_default\_layer}

Description: This command sets the default layer for a selected post tensor. Values for this tensor will be written at the default layer only.

Keyboard Command Sequence:

\texttt{post\_tensor\_default\_layer}
**post_tensor_layers**

**Command:** *post_tensor_layers*

**Description:** This command sets the layers for a selected post tensor. Values for this tensor are written at the specified layers only.

**Keyboard Command Sequence:**

```
post_tensor_layers <layers>
```

**post_tensor_outer_layers**

**Command:** *post_tensor_outer_layers*

**Description:** This command sets the outer and middle layers for a selected post tensor. Values for this tensor will be written at the outer and middle layers only.

**Keyboard Command Sequence:**

```
post_tensor_outer_layers
```

**RESULTS COORDINATE SYSTEM**

**Menu:**

**Description:** These commands control the coordinate system used to decompose tensors and vectors into scalar components for scalar plotting. Normally tensors and vectors are decomposed in a rectangular coordinate system aligned with the global axes. Alternatively, the user may desire to decompose the tensors and vectors in another coordinate system, such as a cylindrical one, not aligned with the global axes, and having a different origin.

```
tensor_system_active <on/off>
```

Activate or inactivate application of the transformation. By default, the transformation is inactive.
tensor_system_rectangular
tensor_system_cylindrical
tensor_system_spherical

Set the coordinate system to be either rectangular (default), cylindrical, or spherical. In **cylindrical**, the first direction becomes the direction radial to the local z-axis, and the second direction becomes the circumferential direction. In **spherical**, the first direction becomes the direction radial to the local origin, and the second direction is tangential and normal to the first direction. In all cases, the third direction (**rectangular**) is perpendicular to the first two.

tensor_system_origin <x y z>
tensor_system_origin_x <x>
tensor_system_origin_y <y>
tensor_system_origin_z <z>

Set the origin of the local coordinate system to be used in the transformation. This affects the cylindrical and spherical types only.

tensor_system_align <p1 p2 p3>

Set the orientation of the local coordinate system giving three points: the origin, a point on the x-axis, and a point on the xy plane.

tensor_system_translate <x y z>

Translate the origin of the local coordinate system.

tensor_system_rotate <x y z>

Rotate the local coordinate system about its origin in the order x, y, and z.

tensor_system_reset

Reset the origin, orientation, and type of the coordinate system to the default settings.

tensor_system_axes <on/off>
tensor_system_axes_length <l>
tensor_system_axes_wireframe
tensor_system_axes_solid
tensor_system_axes_facets <n>
tensor_system_axes_lines <on/off>

These commands control the appearance of the local coordinate system, including its size and manner of display.
Other Buttons with the same description:

- # FACETS
- ALIGN
- AXES LENGTH
- CYLINDRICAL
- DRAW AXES
- EDGES
- LOAD
- RECTANGULAR
- RESET
- RESULTS COORDINATE SYSTEM
- ROTATE
- SAVE
- SET ORIGIN
- SOLID
- SPHERICAL
- TRANSLATE
- WIREFRAME
- X
- Y
- Z

**Command:** post_title

**Description:** This command places the specified title at the bottom of a post plot.

**Keyboard Command Sequence:**

```
post_title <title>
```

**Command:** post_tracks

**Description:** This command switches on the display of particle tracks. The position of a particle as a function of time is visualized by means of a curve. The particles must have been identified during preprocessing by means of a node set. This button is located in the RESULTS->PARTICLE TRACKING menu.

**Keyboard Command Sequence:**

```
post_tracks
```
Command: **post_tracks_marker**

Description: This command toggles the display of markers on particle tracks. The position of a particle as a function of time is visualized by means of a curve. The marker indicates the position of a particle at the current increment. This button is located in the **RESULTS->PARTICLE TRACKING** menu.

Also see: **post_tracks**.

**Keyboard Command Sequence:**

```
post_tracks_marker
```

Command: **post_tracks_off**

Description: This command switches off the display of particle tracks. This button is located in the **RESULTS->PARTICLE TRACKING** menu.

Also see: **post_tracks**.

**Keyboard Command Sequence:**

```
post_tracks_off
```

Command: **post_tracks_strain**

Description: This command switches on the display of particle tracks. The position of a particle as a function of time is visualized by means of a curve. The color of the curve indicates the value of the total equivalent plastic strain of a particle as a function of time. The particles must be identified during preprocessing by means of a node set. This button is located in the **RESULTS->PARTICLE TRACKING** menu.

**Keyboard Command Sequence:**

```
post_tracks_strain
```
**Command:** post_tracks_stress

**Description:** This command switches on the display of particle tracks. The position of a particle as a function of time is visualized by means of a curve. The color of the curve indicates the value of the equivalent stress of a particle as a function of time. The particles must be identified during preprocessing by means of a node set. This button is located in the RESULTS->PARTICLE TRACKING menu.

**Keyboard Command Sequence:**

```
post_tracks_stress
```

---

**Command:** post_tracks_value

**Description:** This command sets the results quantity to be displayed with particle tracking. The available quantities are those that are specified in the track file .trk. The quantity can be specified as a number or by name. This command will show the track of the particle and the value of the particle using a contour line.

**Note:** The post_tracks and post_tracks_off commands will display only the track of the particles.

**Keyboard Command Sequence:**

```
post_tracks_value <quantity>
```
Command: **post_value**

Description: This command sets the results quantity to be displayed. The available quantities can be obtained with the command `post_list`. The quantity can be specified as a number or by name. This button is located in the RESULTS menu.

Keyboard Command Sequence:

```
post_value <quantity>
```

---

Command: **post_var_all_layers**

Description: This command sets all layers for a selected post variable. Values for this variable will be written at all layers. The number of layers per shell element is set with the command `job_param layers`.

Keyboard Command Sequence:

```
post_var_all_layers
```

---

Command: **post_var_default_layer**

Description: This command sets the default layer for a selected post variable. Values for this variable will be written at the default layer only.

Keyboard Command Sequence:

```
post_var_default_layer
```
Command: **post_var_layers**

Description: This command sets the layers for a selected post variable. Values for this variable will be written at the specified layers only.

Keyboard Command Sequence:

```
post_var_layers <layers>
```

---

Command: **post_var_outer_layers**

Description: This command sets the outer and middle layers for a selected post variable. Values for this variable will be written at the outer and middle layers only.

Keyboard Command Sequence:

```
post_var_outer_layers
```

---

Command: **post_vect_threshold**

Description: Sets the threshold for vector plotting. If the ratio of the vector length to the maximum vector length in the plot is smaller than the threshold, the vector will not be drawn.

Keyboard Command Sequence:

```
post_vect_threshold <threshold>
```
Command: post_vector
Description: This command sets the results vector plotting quantity.

Keyboard Command Sequence:
    post_vector <vector quantity>

Command: post_vectors
Description: This command sets the results plotting style to vectors. Vectors representing nodal results values are shown at the nodes.

Keyboard Command Sequence:
    post_vectors

Command: postscript_color
Description: This command sends a color PostScript file representing the current graphics image to the specified printer. Each printer may be configured by editing the appropriate file in the tools directory.

Keyboard Command Sequence:
    postscript_color <plotter number>

Other Buttons with the same description:
    COLOR PRINT 2
    COLOR PRINT 3
**GRAY PRINT 1**

Command:  **postscript_gray**  
**Description:** This command sends a grayscale postscript file representing the current graphics image to the specified printer. Each printer may be configured by editing the appropriate file in the tools directory.  
**Keyboard Command Sequence:**  
   postscript_gray <plotter number>
Other Buttons with the same description:  
   GRAY PRINT 2  
   GRAY PRINT 3

**LANDSCAPE**

Command:  **postscript_landscape**  
**Description:** This command rotates the image 90 degrees in subsequent PostScript files. Thus, the current page height is used for the image’s width, and this often results in a larger printed image. This will be the effect when the graphics window width is greater than its height, and when the current page height is greater than its width.  
Also see:  postscript_portrait. 
**Keyboard Command Sequence:**  
   postscript_landscape

**PORTRAIT**

Command:  **postscript_portrait**  
**Description:** This command aligns the image with the page in subsequent PostScript files (no rotation of the image). Thus, the current page width is used for the image’s width.  
Also see:  postscript_landscape.  
**Keyboard Command Sequence:**  
   postscript_portrait
**RASTER**

Command:  
```
postscript_raster
postscript_vector
```

Description: These commands select either the raster or vector method of producing PostScript files. The (default) raster method draws the Mentat geometry to a virtual graphics window of arbitrary size (resolution). With this method, the final PostScript file contains a raster representation of the drawn image. The vector method simply writes raw geometric primitives such as lines and text to the output file.

Also see: `postscript_thin_lines`.

**Keyboard Command Sequence:**
```
postscript_raster
postscript_vector
```

**THIN LINES**

Command:  
```
postscript_thin_lines
```

Description: This command turns the thin line option for raster PostScript output on or off. This feature (which is OFF by default) makes all the drawn lines have a width of one dot or pixel. This can be desirable for high resolution images that have many lines (such as a mesh with many thousands of elements). When this option is off, a thicker line width is used, which compensates for varying resolutions.

Also see: `postscript_raster`.

**Keyboard Command Sequence:**
```
postscript_thin_lines <on or off>
```
Command: `postscript_raster`  
`postscript_vector`

Description: These commands select either the raster or vector method of producing PostScript files. The (default) raster method draws the Mentat geometry to a virtual graphics window of arbitrary size (resolution). With this method, the final PostScript file contains a raster representation of the drawn image. The vector method simply writes raw geometric primitives such as lines and text to the output file.

Also see: `postscript_thin_lines`.

Keyboard Command Sequence:
```
postscript_raster  
postscript_vector
```

Menu: PREVIOUS

Description: This button brings back the previously continued menu.

Command: `previous_adapg`

Description: This command selects the previous entry in the list of remeshing criteria and makes it the current one. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

Keyboard Command Sequence:
```
previous_adapg
```
Command: previous_adapt

Description: This command selects the previous entry in the list of adapts and makes it the current adapt. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

An adapt is a collection of parameters that controls the adaptive meshing of a set of elements during an analysis.

Keyboard Command Sequence:

previous_adapt

Command: previous_apply

Description: This command selects the previous entry in the list of applications and makes it the current application. This button is located in the BOUNDARY CONDITIONS menu.

A boundary condition application is a set of information that completely defines a boundary condition.

Each application contains:

– the type of boundary condition,
– the degrees of freedom and their corresponding applied values,
– optional tables or functions that apply to the values,
– and the geometric and FEM entities affected.

Multiple applications may be defined and are stored in the list of currently defined applications.

Keyboard Command Sequence:

previous_apply
Command: previous_contact_area
Description: This command selects the previous entry in the list of contact areas and makes it the current contact area. This button is located in the CONTACT->CONTACT AREAS menu.

A contact area is a set of nodes that lie on a contact body that may potentially contact other contact bodies. Contact areas are used to reduce the computational cost of a contact analysis where it is known for which nodes contact is possible. If no contact area is used, all exterior outline or surface nodes will be checked for contact.

Keyboard Command Sequence:

previous_contact_area

Command: previous_cavity
Description: This command selects the previous entry in the list of cavities and makes it the current cavity.

Keyboard Command Sequence:

previous_cavity

Command: previous_contact_body
Description: This command selects the previous entry in the list of contact bodies and makes it the current contact body. This button is located in the CONTACT->CONTACT BODIES menu.

A contact body is a set of curves, surfaces, or elements that act as a body in a contact analysis. In a contact analysis, there must be at least two contact bodies, at least one of which must be a deformable body.

Keyboard Command Sequence:

previous_contact_body
Command: previous_contact_table
Description: This command selects the previous entry in the list of contact tables and makes it the current contact table. This button is located in the CONTACT->CONTACT TABLES menu.

Keyboard Command Sequence:
previous_contact_table

---

Command: previous_crack
Description: This command selects the previous entry in the list of cracks and makes it the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS and 3-D CRACKS menus.

Keyboard Command Sequence:
previous_crack

---

Command: previous_csect
Description: This command selects the previous entry in the list of cross-sections and makes it the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:
previous_csect
Command:  previous_descon

Description:  This command selects the previous entry in the list of design constraints and makes it the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

A design constraint is an analysis response quantity that is constrained to be above or below a specified value.

Keyboard Command Sequence:

previous_descon

Command:  previous_desvar

Description:  This command selects the previous entry in the list of design variables and makes it the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

A design variable is a quantity that can vary during an optimization or sensitivity analysis.

Keyboard Command Sequence:

previous_desvar

Command:  previous_exseg

Description:  This command selects the previous entry in the list of exclude segments and makes it the current one. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

Keyboard Command Sequence:

previous_exseg
Command: **previous_geometry**

Description: This command selects the previous entry in the list of geometries and makes it the current geometry. This button is located in the GEOMETRIC PROPERTIES menu.

An element geometry is a collection of the geometric properties to be applied to a set of elements. Geometric properties include but are not limited to shell thickness, beam and truss areas, and beam moments of inertia.

Keyboard Command Sequence:

```
previous_geometry
```

Command: **previous_icond**

Description: This command selects the previous entry in the list of initial conditions and makes it the current initial condition. This button is located in the INITIAL CONDITIONS menu.

A initial condition is a boundary condition that is applied to the mesh at the beginning of the analysis. Each initial condition contains:

- the degrees of freedom and their corresponding applied values,
- optional tables or functions that apply to the values,
- and the geometric and FEM entities affected.

Keyboard Command Sequence:

```
previous_icond
```

Command: **previous_insert**

Description: This command selects the previous entry in the list of inserts and makes it the current inserts.

Keyboard Command Sequence:

```
previous_insert
```
Command:  **previous_job**  
**Description:** This command selects the previous entry in the list of jobs and makes it the current job. This button is located in the JOBS menu.

A job is a collection of information that is needed to submit and run Marc on the current model. A job contains the analysis class, analysis options, loadcases, and results specifications required for a Marc run. Multiple jobs may be defined and are stored in the list of currently defined jobs.

**Keyboard Command Sequence:**  
previous_job

Command:  **previous_link**  
**Description:** This command selects the previous entry in the list of links and makes it the current links. This button is located in the LINKS->NODAL TIES, SERVO LINKS, and SPRINGS/DASHPOTS menus.

**Keyboard Command Sequence:**  
previous_link

Command:  **previous_loadcase**  
**Description:** This command selects the previous entry in the list of loadcases and makes it the current loadcase. This button is located in the LOADCASES menu.

A loadcase is a collection of boundary conditions, analysis control parameters, and options that are to be used in a portion of a Marc analysis. Multiple loadcases may be defined and are stored in the list of currently defined loadcases.

**Keyboard Command Sequence:**  
previous_loadcase
Command: `previous_material`

Description: This command selects the previous entry in the list of materials and makes it the current material. This button is located in the MATERIAL PROPERTIES menu.

A material is a collection of information that describes a material to be applied to elements in the model. Materials contain the material type, material properties, tables or functions that apply to those properties, and options that will be applied to the specified set of elements during analysis. Multiple materials may be defined and are stored in the list of currently defined materials.

Keyboard Command Sequence:

`previous_material`

---

Command: `previous_orient`

Description: This command selects the previous entry in the list of orientations and makes it the current orientation. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

An orientation is the specification of directions and angles that define the orientation of materials.

Keyboard Command Sequence:

`previous_orient`
Command:  previous_rbe2  
Description:  This command selects the previous entry in the list of and makes it the current rbe2. This button is located in the LINKS->RBE2'S menu.

Keyboard Command Sequence:  
  previous_rbe2

Command:  previous_rbe3  
Description:  This command selects the previous entry in the list of and makes it the current rbe3. This button is located in the LINKS->RBE3'S menu.

Keyboard Command Sequence:  
  previous_rbe3

Command:  previous_section  
Description:  This command selects the previous entry in the list of beam sections and makes it the current section. This button is located in the GEOMETRIC PROPERTIES->MECHANICAL->BEAM SECTIONS menu. 

A beam section is a set of dimensions and properties defining a beam cross-section. Sections may be referenced by geometric properties of type general beam.

Keyboard Command Sequence:  
  previous_section
Command: previous_table
Description: This command selects the previous entry in the list of tables and makes it the current table. This button is located in the MATERIAL PROPERTIES->TABLES or any other link to the TABLES menu.

A table is a function of one variable, i.e. \( y = f(x) \), that may be applied to values specified by the user. Possible values include degree of freedom values in boundary condition applications and material property values. A table is defined by a series of two-dimensional data points. Multiple tables may be defined and are stored in the list of currently defined tables.

Keyboard Command Sequence:

previous_table

Command: previous_transform
Description: This command selects the previous entry in the list of transformations and makes it the current transformation. This button is located in the BOUNDARY CONDITIONS->MECHANICAL menu.

Transformations are used to convert nodal quantities from one coordinate system to another. Transformations are most commonly used to transform the orientation of point loads or fixed displacements, usually from global to local coordinates. Transformations contain the type of transformation, the data required by that transformation type, and a list of nodes the transformation applies to.

Keyboard Command Sequence:

previous_transform
Command: principal_factor
Description: This command changes the scale factor used when displaying principal current tensor quantity vectors. The lengths of the displayed vectors are directly proportional to this scale factor. The default value is 1.0.

Keyboard Command Sequence:
principal_factor <factor>

Command: principal_uniform
Description: This command turns uniform scaling of principal current tensor quantity vectors on or off. When principal current tensor quantity vectors are displayed, they may have a display scale which is uniform across the entire model, or a display scale that varies with the size of each element. The default is uniform scaling.

Keyboard Command Sequence:
principal_uniform <on or off>

Another Button with the same description:
UNIFORM

Command: pt_reset
Description: This command resets feature edge angle and feature vertex angle to their respective default values.
Also see help: pt_set_feature_edge_ang and pt_set_feature_vert_ang

Keyboard Command Sequence:
pt_reset
COARSENING FACTOR

Command: **pt_set_coarsen_factor**

**Description:** This command sets coarsening factor for Patran tetrahedral mesher. This factor can be used to gradually enlarge the tetrahedral element size from the surface to the interior region. It will help reduce the total number of elements in the mesh. The default is 1.5.

Also see help: **pt_tet_mesh**.

**Keyboard Command Sequence:**

```
pt_set_coarsen_factor <value>
```

FEATURE EDGE ANGLE

Command: **pt_set_feature_edge_ang**

**Description:** This command sets feature edge angle for Patran surface mesh-on-mesh (MOM) mesher. If the angle between normal vectors of two neighboring surfaces of an edge is larger than this angle (feature edge angle), the edge will be considered as a soft feature edge. A soft edge will be kept after the remeshing but new nodes can be placed on the edge.

Also see help: **pt_surface_trimesh**.

**Keyboard Command Sequence:**

```
pt_set_feature_edge_ang <value>
```
FEATURE VERTEX ANGLE

Command: pt_set_feature_vert_ang

Description: This command sets the feature vertex angle for Patran surface mesh-on-mesh (MOM) mesher. If two feature edges join at a point and the angle between the two feature edge vectors pointing outward of the point is smaller than this angle (feature vertex angle), the point will be considered as a hard point. A hard point will be kept as an element node after remeshing.

Also see help: pt_surface_trimesh and pt_set_feature_edge_ang.

Keyboard Command Sequence:

pt_set_feature_vert_ang <value>

ELEMENT SIZE

Command: pt_set_tet_element_size

Description: This command sets target element size for Patran tetrahedral mesher. If the target element size is set to 0, three times the average size of the input elements will be used. This number is used with coarsening factor to reduce the total number of elements in the mesh.

Also see help: pt_tet_mesh and pt_set_coarsen_factor.

Keyboard Command Sequence:

pt_set_tet_element_size <value>
**SURFACE TRI REMESH!**

**Command:** `pt_surface_trimesh`  
**Description:** This command uses Patran mesh-on-mesh (MOM) mesher to remesh the given list of surface triangle elements or geometry defined by triangles such as that in stl format. The size of the new triangle elements are defined by the user `pt_set_element_size`. It is recommended that the outline nodes in the input elements be swept before remeshing with this command.

Also see help: `pt_set_element_size`, `pt_set_feature_edge_ang`, and `pt_set_feature_vert_ang`.

**Keyboard Command Sequence:**

```
pt_set_surface_trimesh <triangle element list>
```

**SOLID TET REMESH!**

**Command:** `pt_tet_mesh`  
**Description:** This command uses Patran mesher to mesh the volume defined by the given list of triangle elements with tetrahedral elements. The input triangles must completely enclose the volume to be meshed. The coarsening factor can be used to gradually enlarge the tetrahedral element size from the surface to the interior region.

Also see help: `pt_set_coarsen_factor`.

**Keyboard Command Sequence:**

```
pt_tet_mesh <triangle element list>
```
**Command:**  
**py_call_arguments**

**Description:**  This command sets the argument list for a Python script. It is only available for Python scripts run as a separate process (Run As Separate Process button is on). The argument list is a character string limited to 80 characters. The data is passed to the main() routine in the Python script, and the number must match those expected by the main routine. This button is a text box and is located in the UTILITIES->PYTHON->RUN menu.

**Keyboard Command Sequence:**

```
py_call_arguments <argument list>
```

**Example:**

```
*py_call_arguments 10 20
```

In the Python script:

```python
from py_mentat import *
import sys
def main(x,y):
    print "x is ",x, " and y is ",y
if __name__ == '__main__':
    py_connect("", 40007)
    x = sys.argv[1]
    y = sys.argv[2]
    main(x,y)
    py_disconnect()
```
**py_call_module**

**Command:** py_call_module

**Description:** This command calls a named module in a Python script. When the script is run, Mentat will initiate a socket connection if the RUN AS SEPARATE PROCESS button is on. It uses the current port number setting, which is set with the py_port command. Mentat will wait until it has established a connection with the separate Python process, however, it will still be active (commands may be issued and buttons may be pressed). The Python script must complete the connection with a call to the PyMentat method py_connect. When the Python script completes, it should call the PyMentat method py_disconnect.

If the RUN AS SEPARATE PROCESS button is not on, then the script will be run as an embedded script, meaning that Mentat will not be active until the script runs to completion (similar to executing a Mentat procedure file). No socket connection is initiated.

**Keyboard Command Sequence:**

```
py_call_module <python script name> <module name>
```
**INITIATE CONNECTION**

**Command:**  \texttt{py\_connect}

**Description:** This command will initiate a connection to an external Python process using the port number specified with the \texttt{py\_port} command. When the command is issued, it will print the port number to which it is attempting to make the connection. If the connection has been initiated successfully, the “Wait connect” message will appear in the Status window. This button is located in the UTILITIES->PYTHON menu.

A Python script will subsequently call the PyMentat method \texttt{py\_connect()} to complete the connection, such as:

\begin{verbatim}
py_connect(“myhost.mydomain.com”,40007)
\end{verbatim}

**Keyboard Command Sequence:**

\texttt{py\_connect}

**CLOSE CONNECTION**

**Command:**  \texttt{py\_disconnect}

**Description:** This command will terminate a waiting connection to an external Python process using the port number specified with the \texttt{py\_port} command. This button is located in the UTILITIES->PYTHON menu.

This command may be issued from the command prompt, however, the PyMentat method \texttt{py\_disconnect()} should be used instead in the Python script.

**Keyboard Command Sequence:**

\texttt{py\_disconnect}

**PyMentat Usage:**

\begin{verbatim}
py\_disconnect()
\end{verbatim}
Command: **py_echo**

Description: This command will enable or disable the echoing of Mentat commands in the dialogue window when a Python script is executed. This button is located in the UTILITIES->PYTHON menu.

Keyboard Command Sequence:

```
py_echo <on or off>
```

PyMentat Example Usage:

```
py_send("*py_echo off")
```

Command: **py_file_run**

Description: This command will execute a Python script. There must be a module (method/function) named "main" defined in the script. That module will be invoked. This button is located in the UTILITIES->PYTHON menu.

When the script is run, Mentat will initiate a socket connection if the RUN AS SEPARATE PROCESS button is on. It will use the current port number setting, which can be set with the py_port command. Mentat will wait until it has established a connection with the separate Python process, however, it will still be active (commands may be issued and buttons may be pressed). The Python script must complete the connection with a call to the PyMentat method py_connect. When the Python script completes, it should call the PyMentat method py_disconnect(). You should not exit Mentat until the Python script has completed, as this will cause the port to be remain "in use".

If the RUN AS SEPARATE PROCESS button is not on, then the script will be run as an embedded script, meaning that Mentat will not be active until the script runs to completion (similar to executing a Mentat procedure file). No socket connection is initiated.

Keyboard Command Sequence:

```
py_file_run <python script name>
```
### py_get_float

**Command:** py_get_float

**Description:** This command is issued by a Python script using the PyMentat module to obtain a floating point value from Mentat.

This command is not used from the command prompt.

The argument can be the name of a float variable, a formula which yields a float value, or a Mentat float function. No spaces are allowed in the formula.

**Keyboard Command Sequence:** None

**PyMentat Usage:**

```python
py_get_float(<formula>)
```

**Example:**

```python
py_send("*define Width 20")
delta = py_get_float("nnodes()/Width")
```

### py_get_int

**Command:** py_get_int

**Description:** This command is issued by a Python script using the PyMentat module to obtain an integer value from Mentat.

This command is not used from the command prompt.

The argument can be the name of an integer variable, a formula which yields an integer value, or a Mentat integer function. No spaces are allowed in the formula.

**Keyboard Command Sequence:** None

**PyMentat Usage:**

```python
py_get_int(<formula>)
```

**Example:**

```python
py_send("*define Blocks 20")
n = py_get_int("Width*nnodes()")
```
**py_get_string**

**Command:** `py_string_int`

**Description:** This command is issued by a Python script using the PyMentat module to obtain a string value from Mentat.

This command is not used from the command prompt.

The argument can only be one of the Mentat functions `set_name(id)` or `set_type(id)`, such as `set_name(2)`. No spaces are allowed in the argument.

**Keyboard Command Sequence:**

None

**PyMentat Usage:**

```python
py_get_string(<set_name(id) | set_type(id)>)
```

**Example:** This example prints the names and id’s of all sets:

```python
for i in range(1, p_get_int("nsets()")
    id = py_get_int("set_id(%d)" %i )
    sn = py_get_string("set_name(%d)" % id)
    print "Set ",i, " has id ",id," and name ",sn
```

**PORT**

**Command:** `py_port <value>`

**Description:** This command will set the socket port number used with the `py_connect` or `py_loop_connect` commands. After setting the port number, the `py_connect` or the `py_loop_connect` command is issued. This button is located in the UTILITIES->PYTHON menu.

**Keyboard Command Sequence:**

`py_port <value>`

**Example:** `py_port 40007`
**py_prompt**

**Command:** py_prompt  
**Description:** This command is issued by a Python script using the PyMentat module py_prompt() to display the message at the command prompt. It is typically used inside of a Python script to send a status message to the user. This command is not used from the command prompt.

**Keyboard Command Sequence:**  
py_prompt <string>

**PyMentat Usage:**

```
str = "Creating Nodes"
py_send("*py_prompt " + str)
```

**RESET**

**Command:** py_reset  
**Description:** This command will terminate the Python interpreter. It should be called when a Python error occurs.

**Keyboard Command Sequence:**

py_reset

**RUN AS SEPARATE PROCESS**

**Command:** py_separate_process  
**Description:** This command will enable or disable running a Python script in a separate process. It allows a Python script to run outside of Mentat and thus allowing the graphics to be updated as the script proceeds. This button is located in the UTILITIES->PYTHON->RUN menu.

**Note:** This option may not be turned on for the IBM platforms.

**Keyboard Command Sequence:**

py_separate_process <on or off>
Mentat Help Commands in Q

**Command:** quit

**Description:** This command terminates a Mentat session.

**Keyboard Command Sequence:**

```
quit <'y'or'\n'>
```
Mentat Help Commands in R

**LIMITS**

**Command:** range

**Description:** This command sets the highest and lowest values displayed in contour plots. This range is used if automatic range calculation is turned off. The range is set to the current increment’s min and max values if the COPY LIMITS button (copy_autorange command) is selected.

Also see help: set_autorange and copy_autorange.

**Keyboard Command Sequence:**

```
range <lowest value> <highest value>
```

**NAME**

**Command:** rbe2_name

**Description:** This command sets the name for the current rbe2.

**Keyboard Command Sequence:**

```
rbe2_name <name>
```
Command: **rbe3_ret_node**

Description: This command sets the id of the retained (reference) node for the current rbe2.

Keyboard Command Sequence:

```
 rbe2_ret_node <node_id>
```

---

Command: **rbe2_tied_dof**

Description: This command sets the dof of the retained (reference) node for the current rbe2.

Keyboard Command Sequence:

```
 rbe2_tied_dof <dof_id>
```

---

Command: **rbe3_ret_coef**

Description: This command sets the coef of the retained node for the current rbe3.

Keyboard Command Sequence:

```
 rbe3_ret_coef <slot_id> <coef>
```

---

Command: **rbe3_ret_dof**

Description: This command sets the dof of the retained node for the current rbe3.

Keyboard Command Sequence:

```
 rbe3_ret_dof <dof_id>
```
**NODE**

**Command:** `rbe3_ret_node`  
**Description:** This command sets the node id of the retained node for the current rbe3.  
**Keyboard Command Sequence:**
```
rbe3_ret_node <slot_id> <node_id>
```

**DOF**

**Command:** `rbe3_tied_dof`  
**Description:** This command sets the dof of the retained (reference) node for the current rbe3.  
**Keyboard Command Sequence:**
```
rbe3_tied_dof <dof_id>
```

**NODE**

**Command:** `rbe3_tied_node`  
**Description:** This command sets the id of the tied (reference) node for the current rbe3.  
**Keyboard Command Sequence:**
```
rbe3_tied_node <node_id>
```

**read_marc**

**Command:** `read_marc`  
**Description:** This command reads an existing Marc file. If you are currently editing a model, entities in the Marc file will be merged with the existing model. It is left to the user to see that names and numbers remain unique. You must specify the name of the Marc file.  
**Keyboard Command Sequence:**
```
read_marc <Marc file name>
```
Subroutine: **REBAR**

**Description:** The user subroutine **REBAR** allows the input of the position, size, and orientation of the single strain rebar elements (23, 46, 47, 48).

**Command:** **redraw**

**Description:** This command redraws the current model. Only the information contained in the display list is redrawn, not any information that may require regeneration of the model to be seen (see the **REGENERATE** command).

**Keyboard Command Sequence:**

```
redraw
```

**Command:** **refine_node**

**Description:** This command subdivides elements about a node. The subdivision is such that only edges and faces adjacent to the node are modified, which preserves mesh continuity.

**Keyboard Command Sequence:**

```
refine_node <node> <element list> #
```

**Command:** **regenerate**

**Description:** This command instructs the program to traverse its database of geometric and mesh entities, in order to regenerate the graphics display lists with those entities, using the current plotting settings. The new display lists are then drawn, bringing the graphical appearance of the program up-to-date.

**Keyboard Command Sequence:**

```
regenerate
```
Menu: RELAX
Description: This menu contains commands for modifying the mesh by repositioning nodes. Nodes are moved to reduce the amount of or distortion in the mesh.

Command: relax_nodes
Description: This command smoothes the mesh by adjusting the coordinates of the specified nodes. After completion, the coordinate of each node moved is the average of its neighbors’ coordinates.

Keyboard Command Sequence:
relax_nodes <node list> #

Command: relax_none_fixed
Description: This command specifies that all nodes given to the relax_nodes command will be relaxed.

Keyboard Command Sequence:
relax_none_fixed

Command: relax_outline_fixed
Description: This command specifies that all nodes on the outline of the mesh will be excluded from the list given to the relax_nodes command and will therefore be unchanged.

Keyboard Command Sequence:
relax_outline_fixed
Command: **relax_reset**
Description: This command resets the relaxation parameters to their default settings.

**Keyboard Command Sequence:**

```
relax_reset
```

---

**SURFACE**

Command: **relax_surface_fixed**
Description: This command specifies that all nodes on the surface of the mesh will be excluded from the list given to the `relax_nodes` command and will therefore remain unchanged.

**Keyboard Command Sequence:**

```
relax_surface_fixed
```

---

**relax_tolerance**

Command: **relax_tolerance**
Description: This command sets the relaxation tolerance. The tolerance is used to terminate the relaxation iteration process and specifies the greatest node movement allowed (sum of the squares) before termination of the process. A limit of ten iterations is imposed.

**Keyboard Command Sequence:**

```
relax_tolerance <tolerance>
```

---

**MESH ADAPTIVITY**

Menu: **MESH ADAPTIVITY**
Description: This menu contains the commands for specifying the criteria for global remeshing and local adaptivity.

Global remeshing allows different parts of the model to be remeshed. A new mesh is automatically created, relevant solution quantities are transferred to the new mesh and the analysis continues.
Local adaptivity provides the capability to increase the number of elements and nodes to improve the accuracy of the solution. The capability is applicable for both linear elastic analysis and for nonlinear analysis. The capability can be used for lower-order elements, 3-node triangular, 4-node quadrilateral, 4-node tetrahedral, and 8-node hexahedral elements.

Command: **remove_adapt_elements**

Description: This command removes elements from the current adapt. The elements in the list will no longer be adaptively meshed during the analysis according to the parameters in this adapt. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

Keyboard Command Sequence:

```
remove_adapt_elements <element list> #
```

Command: **remove_all_applys**

Description: This command deletes all boundary conditions.

Keyboard Command Sequence:

```
remove_all_applys
```

Command: **remove_all_cavities**

Description: This command removes all cavities from the model. It is located under the MESH GENERATION->CAVITIES (CURVES) menu.

Keyboard Command Sequence:

```
remove_all_cavities
```
REMOVE ALL INITIAL CONDS

Command: remove_all_iconds
Description: This command deletes all initial conditions.
Keyboard Command Sequence:
remove_all_iconds

REMOVE ALL RBE2’S

Command: remove_all_rbe2s
Description: This command removes all existing RBE2’s.
Keyboard Command Sequence:
remove_all_rbe2s

REMOVE ALL RBE3’S

Command: remove_all_rbe3s
Description: This command removes all existing RBE3’s.
Keyboard Command Sequence:
remove_all_rbe3s

REM ALL SERVOS

Command: remove_all_servos
Description: This command deletes all servo links.
Keyboard Command Sequence:
remove_all_servos
**REMOVE ALL SETS**

Command:  `remove_all_sets`

Description: This command removes all sets from the model.

Keyboard Command Sequence:

```
remove_all_sets
```

**REM ALL SPRINGS**

Command:  `remove_all_springs`

Description: This command deletes all servo links.

Keyboard Command Sequence:

```
remove_all_springs
```

**REM ALL TIES**

Command:  `remove_all_ties`

Description: This command deletes all ties.

Keyboard Command Sequence:

```
remove_all_ties
```

**REMOVE**

Command:  `remove_annotation`

Description: This command is used to remove annotations from a view. Annotations are strings of text that can be used to describe what is being displayed in a view. You must specify a list of annotations to remove. This button is located in the UTILITIES->MORE menu.

Keyboard Command Sequence:

```
remove_annotation <annotation list> #
```
Command: **remove_apply_cavities**

Description: This command removes cavities from the current boundary condition cavity pressure/mass load. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CAVITY PRESSURE/MASS LOAD menus.

**Keyboard Command Sequence:**

```plaintext
remove_apply_cavities <cavity list> #
```

---

Command: **remove_apply_curves**

Description: This command removes curves from the current boundary condition application. You must specify a list of curves. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->FIXED DISPLACEMENT and other boundary condition class menus.

**Keyboard Command Sequence:**

```plaintext
remove_apply_curves <curve list> #
```

---

Command: **remove_apply_edges**

Description: This command removes edges from the current boundary condition application. You must specify a list of edges. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->EDGE LOAD and other boundary condition class menus.

**Keyboard Command Sequence:**

```plaintext
remove_apply_edges <edge list> #
```
Command: **remove_apply_elements**

**Description:** This command removes elements from the current boundary condition application. You must specify a list of elements. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->GLOBAL LOAD and other boundary condition class menus.

**Keyboard Command Sequence:**

`remove_apply_elements <element list> #`  

Command: **remove_apply_faces**

**Description:** This command removes faces from the current boundary condition application. You must specify a list of faces. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->FACE LOAD and other boundary condition class menus.

**Keyboard Command Sequence:**

`remove_apply_faces <face list> #`  

Command: **remove_apply_nodes**

**Description:** This command removes nodes from the current boundary condition application. You must specify a list of nodes. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->FIXED DISPLACEMENT and other boundary condition class menus.

**Keyboard Command Sequence:**

`remove_apply_nodes <node list> #`  

Command: **remove_apply_points**

**Description:** This command removes points from the current boundary condition application. You must specify a list of points. This button is located in the
BOUNDARY CONDITIONS->MECHANICAL->FIXED DISPLACEMENT and other boundary condition class menus.

**Keyboard Command Sequence:**

```
remove_apply_points <point list> #
```

---

**Command:** remove_apply_surfaces  
**Description:** This command removes surfaces from the current boundary condition application. You must specify a list of surfaces. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->FIXED DISPLACEMENT and other boundary condition class menus.

**Keyboard Command Sequence:**

```
remove_apply_surfaces <surface list> #
```

---

**Command:** remove_cavity_curves  
**Description:** This command removes curves from the current cavity. It is located under the MESH GENERATION->CAVITIES (CURVES) menu.

**Keyboard Command Sequence:**

```
remove_cavity_curves <curve list> #
```

---

**Command:** remove_cavity_edges  
**Description:** This command removes element edges from the current cavity. It is located under the MESH GENERATION->CAVITIES (EDGES) menu.

**Keyboard Command Sequence:**

```
remove_cavity_edges <edge list> #
```
**Command:** remove_cavity_faces

**Description:** This command removes element faces from the current cavity. It is located under the MESH GENERATION->CAVITIES (FACES) menu.

**Keyboard Command Sequence:**
```
remove_cavity_faces <face list> #
```

**Command:** remove_cavity_surfaces

**Description:** This command removes surfaces from the current cavity. It is located under the MESH GENERATION->CAVITIES (SURFACES) menu.

**Keyboard Command Sequence:**
```
remove_cavity_surfaces <surface list> #
```

**Command:** remove_cbody_dc_edges

**Description:** This command removes edges from the list of edges defining discontinuities in the analytical formulation for deformable bodies. You must specify a list of edges. This button is located in the CONTACT->CONTACT BODIES->DEFORMABLE and RIGID w HEAT TRANSFER menus with ANALYTIC option on.

**Keyboard Command Sequence:**
```
remove_cbody_dc_edges <edge list> #
```

**Command:** remove_cbody_dc_nodes

**Description:** This command removes nodes from the list of nodes defining discontinuities in the analytical formulation for deformable bodies. You must specify a list of nodes. This button is located in the CONTACT->CONTACT BODIES->DEFORMABLE and RIGID w HEAT TRANSFER menus with ANALYTIC option on.
**Keyboard Command Sequence:**

```
remove_cbody_dc_nodes <node list> #
```

---

**Command:** `remove_contact_body_curves`

**Description:** This command removes curves from the current contact body. You must specify a list of curves. This button is located in the CONTACT->CONTACT BODIES->RIGID and SYMMETRY menus.

**Keyboard Command Sequence:**

```
remove_contact_body_curves <curve list> #
```

---

**Command:** `remove_contact_body_elem`

**Description:** This command removes elements from the current contact body. You must specify a list of elements. This button is located in the CONTACT->CONTACT BODIES->DEFORMABLE and other contact body type menus.

**Keyboard Command Sequence:**

```
remove_contact_body_elem <element list> #
```

---

**Command:** `remove_contact_body_surf`

**Description:** This command removes surfaces from the current contact body. You must specify a list of surfaces. This button is located in the CONTACT->CONTACT BODIES->RIGID and SYMMETRY menus.

**Keyboard Command Sequence:**

```
remove_contact_body_surf <surface list> #
```
Command: remove_crack_rigreg_nodes
Description: This command removes nodes from the current rigid region of the current crack. This button is located in the FRACTURE MECHANICS->2-D CRACKS/3-D CRACKS->MANUAL menu.

Keyboard Command Sequence:
remove_crack_rigreg_nodes <rigid region> <node list> #

Command: remove_csect_elements
Description: This command removes elements from the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:
remove_csect_elements <element list> #

Command: remove_csect_nodes
Description: This command removes nodes from the current cross-section. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:
remove_csect_nodes <node list> #
Command: remove_current_adapg
Description: This command removes the current entry of remeshing criteria. Upon removal, the previous entry becomes the current one. If there is no previous entry, the next one will become the current one. This button is located in the MESH ADAPTIVITY->GLOBAL REMESHING menu.

Keyboard Command Sequence:
remove_current_adapg

Command: remove_current_adapt
Description: This command removes the current adapt. Upon removal, the previous adapt becomes the current one. If there is no previous adapt, the next one will become the current one. This button is located in the MESH ADAPTIVITY->LOCAL ADAPTIVITY menu.

Keyboard Command Sequence:
remove_current_adapt

Command: remove_current_apply
Description: This command removes the current boundary condition application. Upon removal, the previous application becomes the current one. If there is no previous application, the next one will become the current one. This button is located in the BOUNDARY CONDITIONS menu.

Keyboard Command Sequence:
remove_current_apply
Command: remove_current_cavity
Description: This command removes the current cavity. Upon removal, the previous cavity becomes the current one. If there is no previous cavity, the next one will become the current one. It is located under the MESH GENERATION->CAVITIES menu.

Keyboard Command Sequence:
remove_current_cavity

Command: remove_current_contact_area
Description: This command removes the current contact area. Upon removal, the previous contact area becomes the current one. If there is no previous contact area, the next one becomes the current one. This button is located in the CONTACT->CONTACT AREAS menu.

Keyboard Command Sequence:
remove_current_contact_area

Command: remove_current_contact_body
Description: This command removes the current contact body. Upon removal, the previous contact body becomes the current one. If there is no previous contact body, the next one becomes the current one. This button is located in the CONTACT->CONTACT BODIES menu.

Keyboard Command Sequence:
remove_current_contact_body
Command:  remove_current_contact_table
Description:  This command removes the current contact table. Upon removal, the previous contact table becomes the current one. If there is no previous contact table, the next one becomes the current one. This button is located in the CONTACT->CONTACT TABLES menu.

Keyboard Command Sequence:  
remove_current_contact_table

Command:  remove_current_crack
Description:  This command removes the current crack. Upon removal, the previous crack becomes the current one. If there is no previous crack, the next one will become the current one.

Keyboard Command Sequence:  
remove_current_crack

Command:  remove_current_csect
Description:  This command removes the current cross-section. Upon removal, the previous cross-section becomes the current one. If there is no previous cross-section, the next one will become the current one. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->CROSS SECTIONS menu.

Keyboard Command Sequence:  
remove_current_csect
Command: **remove_current_descon**
Description: This command removes the current design constraint. Upon removal, the previous design constraint becomes the current one. If there is no previous design constraint, the next one will become the current one. This button is located in the DESIGN->DESIGN CONSTRAINTS menu.

**Keyboard Command Sequence:**

remove_current_descon

Command: **remove_current_desvar**
Description: This command removes the current design variable. Upon removal, the previous design variable becomes the current one. If there is no previous design variable, the next one will become the current one. This button is located in the DESIGN->DESIGN VARIABLES menu.

**Keyboard Command Sequence:**

remove_current_desvar

Command: **remove_current_exseg**
Description: This command removes the current exclude segment. Upon removal, the previous exclude segment becomes the current one. If there is no previous exclude segment, the next one becomes the current one. This button is located in the CONTACT->EXCLUDE SEGMENTS menu.

**Keyboard Command Sequence:**

remove_current_exseg
Command: remove_current_geometry
Description: This command removes the current geometry property. Upon removal, the previous geometry becomes the current one. If there is no previous geometry property, the next one will become the current one. This button is located in the GEOMETRIC PROPERTIES menu.

Keyboard Command Sequence:
remove_current_geometry

Command: remove_current_icond
Description: This command removes the current initial condition. Upon removal, the previous initial condition becomes the current one. If there is no previous initial condition, the next one will become the current one. This button is located in the INITIAL CONDITIONS menu.

Keyboard Command Sequence:
remove_current_icond

Command: remove_current_insert
Description: This command removes the current insert. Upon removal, the previous insert becomes the current one. If there is no previous insert, the next one will become the current one.

Keyboard Command Sequence:
remove_current_insert
Command: remove_current_job
Description: This command removes the current job. Upon removal, the previous job becomes the current one. If there is no previous job, the next one will become the current one. This button is located in the JOBS menu.

Keyboard Command Sequence:
remove_current_job

Command: remove_current_link
Description: This command removes the current link. Upon removal, the previous link becomes the current one. If there is no previous boundary condition, the next one will become the current one. This button is located in the LINKS->NODAL TIES, SERVO LINKS, and SPRINGS/DASHPOTS menus.

Keyboard Command Sequence:
remove_current_link

Command: remove_current_loadcase
Description: This command removes the current loadcase. Upon removal, the previous loadcase becomes the current one. If there is no previous loadcase, the next one will become the current one. This button is located in the LOADCASES menu.

Keyboard Command Sequence:
remove_current_loadcase
Command: **remove_current_material**

Description: This command removes the current material. Upon removal, the previous material becomes the current one. If there is no previous material, the next one will become the current one. This button is located in the MATERIAL PROPERTIES menu.

Keyboard Command Sequence:

```
remove_current_material
```
Command: **remove_current_section**

Description: This command removes the current beam section. Upon removal, the previous section becomes the current one. If there is no previous section, the next one will become the current one. This button is located in the GEOMETRIC PROPERTIES->MECHANICAL->BEAM SECTIONS menu.

Keyboard Command Sequence:

```
remove_current_section
```
Command: **remove_curves**

Description: This command removes curves from the model. You must specify a list of curves. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:

```
remove_curves <curve list> #
```

Command: **remove_edge_sets**

Description: This command removes all edge sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

Keyboard Command Sequence:

```
remove_edge_sets
```

Command: **remove_element_sets**

Description: This command removes all element sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

Keyboard Command Sequence:

```
remove_element_sets
```

Command: **remove_elements**

Description: This command removes elements from the model. You must specify a list of elements. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:

```
remove_elements <element list> #
```
**REMOVE EMPTY SETS**

**Command:**  remove_empty_sets

**Description:** This command removes all empty sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

**Keyboard Command Sequence:**

remove_empty_sets

**FACE**

**Command:**  remove_face_sets

**Description:** This command removes all face sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

**Keyboard Command Sequence:**

remove_face_sets

**REM**

**Command:**  remove_geometry_elements

**Description:** This command removes elements from the current geometry property. You must specify a list of elements. This button is located in the GEOMETRIC PROPERTIES->3D and other geometric properties menus.

**Keyboard Command Sequence:**

remove_geometry_elements <element list> #
Command: **remove_icond_curves**

**Description:** This command removes curves from the current initial condition. You must specify a list of curves. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

**Keyboard Command Sequence:**

```
remove_icond_curves <curve list> #
```

---

Command: **remove_icond_edges**

**Description:** This command removes edges from the current initial condition. You must specify a list of edges. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

**Keyboard Command Sequence:**

```
remove_icond_edges <edge list> #
```

---

Command: **remove_icond_elements**

**Description:** This command removes elements from the current initial condition. You must specify a list of elements. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

**Keyboard Command Sequence:**

```
remove_icond_elements <element list> #
```
Command: **remove_icond_faces**

*Description:* This command removes faces from the current initial condition. You must specify a list of faces. This button is located in the **INITIAL CONDITIONS->MECHANICAL** and other initial conditions menus.

*Keyboard Command Sequence:*

```
remove_icond_faces <face list> #
```

Command: **remove_icond_nodes**

*Description:* This command removes nodes from the current initial condition. You must specify a list of nodes. This button is located in the **INITIAL CONDITIONS->MECHANICAL** and other initial conditions menus.

*Keyboard Command Sequence:*

```
remove_icond_nodes <node list> #
```

Command: **remove_icond_points**

*Description:* This command removes points from the current initial condition. You must specify a list of points. This button is located in the **INITIAL CONDITIONS->MECHANICAL** and other initial conditions menus.

*Keyboard Command Sequence:*

```
remove_icond_points <point list> #
```
Command:  `remove_icond_surfaces`

Description:  This command removes surfaces from the current initial condition. You must specify a list of surfaces. This button is located in the INITIAL CONDITIONS->MECHANICAL and other initial conditions menus.

Keyboard Command Sequence:

```
remove_icond_surfaces <surface list> #
```

Command:  `remove_ideas_elem_var`

Description:  This command removes results ideas variables from the list of requested variables for the current job. You must specify a variable by name. The button text will be the name of the results variable. This button is located in the JOBS->MECHANICAL->JOB RESULTS->I-DEAS and other analysis class menus.

Keyboard Command Sequence:

```
remove_ideas_elem_var <variable name>
```

Command:  `remove_ideas_elnod_var`

Description:  This command removes results ideas variables from the list of requested variables for the current job. You must specify a variable by name. The button text will be the name of the results variable. This button is located in the JOBS->MECHANICAL->JOB RESULTS->I-DEAS and other analysis class menus.

Keyboard Command Sequence:

```
remove_ideas_elnod_var <variable name>
```
Command:  **remove_ideas_node_var**

**Description:** This command removes results ideas variables from the list of requested variables for the current job. You must specify a variable by name. The button text will be the name of the results variable. This button is located in the JOBS->MECHANICAL->JOB RESULTS->I-DEAS and other analysis class menus.

**Keyboard Command Sequence:**

```
remove_ideas_node_var <variable name>
```

---

Command:  **remove_insert_embedded**

**Description:** This command removes embedded elements or embedded nodes from current insert.

**Keyboard Command Sequence:**

```
remove_insert_embedded <elements list> or <nodes list>
```

---

Command:  **remove_insert_host_elements**

**Description:** This command removes host elements from current insert.

**Keyboard Command Sequence:**

```
remove_insert_host_elements <elements list>
```
Command: **remove_job_applys**

**Description:** This command removes initial load applications from the current job. Load applications are specified by name. The button text will be the name of the initial condition. This button is located in the JOBS->MECHANICAL->INITIAL LOADS and other analysis class menus.

**Keyboard Command Sequence:**

```plaintext
remove_job_applys <load apply name>
```

Command: **remove_job_careas**

**Description:** This command removes contact areas from the current job. The specified contact areas will no longer be used as initial contact areas for the job. Contact areas are specified by name. The button text will be the name of the contact area. This button is located in the JOBS->MECHANICAL->CONTACT CONTROL->INITIAL CONTACT and other analysis class menus.

**Keyboard Command Sequence:**

```plaintext
remove_job_careas <contact area names>
```

Command: **remove_job_iconds**

**Description:** This command removes initial conditions from the current job. The specified conditions will no longer be used as initial load conditions for the job. Initial conditions are specified by name. The text on the button will be the name of the initial condition. This button is located in the JOBS->MECHANICAL->INITIAL LOADS and other analysis class menus.

**Keyboard Command Sequence:**

```plaintext
remove_job_iconds <initial condition names>
```
Command: remove_job_loadcases
Description: This command removes loadcases from the current job. Loadcases are specified by name. The button text will be the name of the loadcase. This button is located in the JOBS->MECHANICAL->AVAILABLE and other analysis class menus.

Keyboard Command Sequence:

```
remove_job_loadcases <loadcase name>
```

Command: remove_loadcase_careas
Description: This command deactivates contact areas in the current loadcase. Only activated contact areas are considered during analysis. Contact areas are specified by name. The button text will be the name of the contact area. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->CONTACT AREAS and other loadcase class menus.

Keyboard Command Sequence:

```
remove_loadcase_careas <contact area names>
```

Command: remove_loadcase_cbodies
Description: This command allows adhesion between the specified contact bodies and all other contact bodies for the current loadcase. Releasing a contact body (see add_loadcase_cbodies) during an analysis forces a separation between that body and any other body touching it. The button text will be the name of the contact area. This button is located in the LOADCASES->MECHANICAL->STATIC->CONTACT->CONTACT AREAS and other loadcase class menus.

Keyboard Command Sequence:

```
remove_loadcase_cbodies <contact body names>
```
Command: **remove_loadcase_loads**

Description: This command deactivates boundary conditions applications (loads) in the current loadcase. Deactivated applications are ignored during analysis. You must specify a boundary condition application name. The button text will be the name of the load. This button is located in the LOADCASES->MECHANICAL->STATIC->LOADS and other loadcase class menus.

Keyboard Command Sequence:

```
remove_loadcase_loads <application name>
```

---

Command: **remove_marc_history_text**

Description: This command allows removing a line that will be placed in the Marc data file at the end of the history definition section. The command input consists of the line number that you want to remove.

Keyboard Command Sequence:

```
remove_marc_history_text <line number>
```

---

Command: **remove_marc_model_text**

Description: This command allows removing a line that will be placed in the Marc data file at the end of the model definition section. The command input consists of the line number that you want to remove.

Keyboard Command Sequence:

```
remove_marc_model_text <line number>
```
Command: **remove_marc_param_text**

Description: This command allows removing a line that will be placed in the Marc data file at the end of the parameter definition section.

The command input consists of the line number that you want to remove.

**Keyboard Command Sequence:**

`remove_marc_param_text <line number>`

Command: **remove_material_elements**

Description: This command removes elements from the current material. This has the effect of removing the material from the elements. You must specify a list of elements. This button is located in the **MATERIAL PROPERTIES** menu.

**Keyboard Command Sequence:**

`remove_material_elements <element list> #`

Command: **remove_node_sets**

Description: This command removes all node sets from the model. This button is located in the **MESH GENERATION->SELECT->REMOVE SETS** or any other link to the **SELECT** menu.

**Keyboard Command Sequence:**

`remove_node_sets`
Command: **remove_nodes**

**Description:** This command removes nodes from the model. You must specify a list of nodes. Only the nodes which are not used by any element will be removed. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
remove_nodes <node list> #
```

---

Command: **remove_normal_edges**

**Description:** This command is used to specify which element edges contribute to the computation of node normals used in post plotting. This button is located in the RESULTS->TOOLS menu.

**Keyboard Command Sequence:**

```
remove_normal_edges <element edge list>
```

---

Command: **remove_normal_faces**

**Description:** This command is used to specify which element faces contribute to the computation of node normals used in post plotting. This button is located in the RESULTS->TOOLS menu.

**Keyboard Command Sequence:**

```
remove_normal_faces <element face list>
```
**Command:** remove_orient

**Description:** This command removes the current orientation. Upon removal, the previous orientation becomes the current one. If there is no previous orientation, the next one will become the current one. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

**Keyboard Command Sequence:**

```
remove_orient
```

---

**Command:** remove_orient_elements

**Description:** This command removes the current orientation from the specified elements. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

**Keyboard Command Sequence:**

```
remove_orient_elements <elements list> #
```

---

**Command:** remove_point_sets

**Description:** This command removes all point sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

**Keyboard Command Sequence:**

```
remove_point_sets
```
Command: **remove_points**

**Description:** This command removes points from the model. You must specify a list of points. Only those points not used by any curve or surface will be removed. This button is located in the MESH GENERATION menu.

**Keyboard Command Sequence:**

```
remove_points <point list> #
```

---

**Stress**

Command: **remove_post_tensor**

**Description:** This command removes results tensors from the list of requested tensors for the current job. You must specify a tensor name. The button text will be the name of the element tensor. This button is located in the JOBS->MECHANICAL->JOB RESULTS and other analysis class menus.

**Keyboard Command Sequence:**

```
remove_post_tensor <tensor name>
```

---

**Equivalent Von Mises**

Command: **remove_post_var**

**Description:** This command removes results variables from the list of requested variables for the current job. You must specify a variable name. The button text will be the name of the element variable. This button is located in the JOBS->MECHANICAL->JOB RESULTS and other analysis class menus.

**Keyboard Command Sequence:**

```
remove_post_var <variable name>
```
**Commands**

**remove_rbe2_tied_nodes**

**Description:** This command removes tied nodes by list of node id’s. This button is located in the LINKS->RBE2’S menu.

The nodes are entered directly on the command line, by picking them with the mouse.

**Keyboard Command Sequence:**

```
remove_rbe2_tied_nodes <node list> #
```

**remove_rbe3_ret_nodes**

**Description:** This command removes retained nodes by list of node id’s. This button is located in the LINKS->RBE3’S menu.

The nodes are entered directly on the command line, by picking them with the mouse.

**Keyboard Command Sequence:**

```
remove_rbe3_ret_nodes <node list> #
```

**remove_set_entries**

**Description:** This command removes entries from the specified set. The type of entries specified in the list must correspond to the type of set; e.g. a list of nodes should be given for a node set. This command is used for types of sets for which there is a `store` command (as in `store_nodes`, `store_elements`, etc). This button is located in the MESH GENERATION->SELECT or any other link to the SELECT menu.

**Keyboard Command Sequence:**

```
remove_set_entries <set name> <entry list> #
```
**REMOVE ONE SET**

**Command:** `remove_sets`

**Description:** This command removes stored sets. You must specify a list of sets. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

**Keyboard Command Sequence:**

```
remove_sets <set list> #
```

**SOLID EDGE**

**Command:** `remove_solid_edge_sets`

**Description:** This command removes all solid edge sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

```
remove_solid_edge_sets
```

**SOLID FACE**

**Command:** `remove_solid_face_sets`

**Description:** This command removes all solid face sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

**Keyboard Command Sequence:**

```
remove_solid_face_sets
```
Command: `remove_solid_sets`
Description: This command removes all solid sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

Keyboard Command Sequence:
```
remove_solid_sets
```

Command: `remove_solid_vertex_sets`
Description: This command removes all solid vertex sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

Keyboard Command Sequence:
```
remove_solid_vertex_sets
```

Command: `remove_solids`
Description: This command removes solids from the model. You must specify a list of solids. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:
```
remove_solids <solid list> #
```

Command: `remove_surface_sets`
Description: This command removes all surface sets from the model. This button is located in the MESH GENERATION->SELECT->REMOVE SETS or any other link to the SELECT menu.

Keyboard Command Sequence:
```
remove_surface_sets
```
Command: remove_surfaces

Description: This command removes surfaces from the model. You must specify a list of surfaces. This button is located in the MESH GENERATION menu.

Keyboard Command Sequence:

```
remove_surfaces <surface list> #
```

Command: remove_transform

Description: This command removes the current node transform. All nodes which were transformed by the transformation will no longer be transformed. After the transform is removed, the previous transform will become the current transform. If there is no previous transform, the next one will become the current one. A transform may not be deleted if nodes are still being transformed by it. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->TRANSFORMS and other boundary conditions menus.

Keyboard Command Sequence:

```
remove_transform
```
**Command:** remove_transform_nodes

**Description:** This command removes the specified nodes from the current transformation. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->TRANSFORMS and other boundary conditions menus.

**Keyboard Command Sequence:**

```
remove_transform_nodes <node list> #
```

**Command:** remove_unused_nodes

**Description:** This command removes from the model all nodes which are not being used by any element. This button is located in the MESH GENERATION->SWEEP menu.

**Keyboard Command Sequence:**

```
remove_unused_nodes
```

**Command:** remove_unused_orients

**Description:** This command removes all orientations that have not been applied to elements from the list of orientations. This button is located in the MATERIAL PROPERTIES->ORIENTATIONS menu.

**Keyboard Command Sequence:**

```
remove_unused_orients
```
Command: **remove_unused_points**

Description: This command removes from the model all points which are not being used by any curve or surface. This button is located in the MESH GENERATION->SWEEP menu.

Keyboard Command Sequence:

remove_unused_points

---

Command: **remove_unused_transforms**

Description: This command removes from the model all transforms which are not being used by any node. This button is located in the BOUNDARY CONDITIONS->MECHANICAL->TRANSFORMS and other boundary conditions menus.

Keyboard Command Sequence:

remove_unused_transforms

---

Command: **render**

Description: This command starts a rendering job in the background, which creates a photorealistic image of the 3-D geometry in a view. Mentat first writes out a render file with information for the render program. This file contains the current render settings, view settings, lighting settings, colormap information, and 3-D geometry. Then an external process is started, which reads in this file, and writes out the completed image file. When the image is complete, a new window displaying the image appears.

To remove the image window, move the mouse to the image window, and press "q" or the ESC key.
You must specify which view to render (1, 2, 3, or 4). A base file name must also be specified. Then, file name .ren will be the render file, and file name .rgb will be the image file.

Also see: image_display, image_slide, and image_save.

**Keyboard Command Sequence:**

```plaintext
render <view> <file name> <yes/no>
```

---

**Command:** `render_antialias_depth`

**Description:** This command allows you to specify the antialias depth for subsequent rendered images. Antialiasing is a technique which attempts to minimize the jaggedness and other unwanted artifacts that sometimes appear in raster images.

The Mentat renderer uses adaptive supersampling as its antialiasing technique. The exact color is computed at the four corners and center of each pixel. If any of these five colors differs from the average of the five by more than the antialias color tolerance, the pixel is divided into four sub-pixels, and the process is repeated for each sub-pixel, until the antialias depth has been reached. Thus the greater the depth is, the more times a pixel may be subdivided. A higher depth may make an image look better, while making it take longer to create.

You must specify the depth (one or greater).

Also see: render, render_antialias, and render_antialias_tol.

**Keyboard Command Sequence:**

```plaintext
render_antialias_depth <depth>
```
**TOLERANCE**

**Command:** `render_antialias_tol`

**Description:** This command allows you to specify the antialias color tolerance for subsequent rendered images. Antialiasing is a technique which attempts to minimize the jaggedness and other unwanted artifacts that sometimes appear in raster images.

The Mentat renderer uses adaptive supersampling as its antialiasing technique. The exact color is computed at the four corners and center of each pixel. If any of these five colors differs from the average of the five by more than the antialias color tolerance, the pixel is divided into four sub-pixels, and the process is repeated for each sub-pixel, until the antialias depth has been reached. Thus the smaller the tolerance is, the more likely a pixel will be subdivided. A lower tolerance may make an image look better, while making it take longer to create.

You must specify the color tolerance (0.0 to 1.0). A color tolerance of 0.0 forces all pixels to be subdivided up to the antialias depth.

Also see: `render`, `render_antialias`, and `render_antialias_depth`.

**Keyboard Command Sequence:**

```
render_antialias_tol <tolerance>
```

**COLOR**

**Command:** `render_atmosphere`

**Description:** This command sets a colored atmosphere for subsequent rendered images. Thus, atmospheric depth cueing will be present. This means that the further away an object is, the more it will be obscured by the atmosphere color.

You must specify the red, green, and blue components of the color. These values range from zero to one, where zero means completely OFF, and one means completely ON. Setting the atmosphere color to black (0, 0, 0) turns off the atmosphere effect.

Also see: `render`, `render_clear`, `render_fog`, `render_haze`, `render_smog`, `render_atmosphere_len`, and `render_atmosphere_index_ref`. 
Keyboard Command Sequence:

`render_atmosphere <red> <green> <blue>`

**INDEX OF REFRACTION**

Command: `render_atmosphere_index_ref`

Description: This command allows you to specify the atmosphere index of refraction for subsequent rendered images. Whenever the atmosphere color is not black (0, 0, 0), atmospheric depth cueing will be present in rendered images. This means that the further away an object is, the more it will be obscured by the atmosphere color. Light traveling through the atmosphere will be bent by an angle determined by its index of refraction.

You must specify the index of refraction.

Also see: `render_atmosphere` and `render_atmosphere_len`.

Keyboard Command Sequence:

`render_atmosphere_index_ref <index of refraction>`

**LENGTH**

Command: `render_atmosphere_len`

Description: This command allows you to specify the half-atmosphere length for subsequent rendered images. Whenever the atmosphere color is not black (0, 0, 0), atmospheric depth cueing will be present in rendered images. This means that the further away an object is, the more it will be obscured by the atmosphere color. The half-atmosphere length is the distance that light must travel in view space, for its color to change halfway to the atmosphere color. Thus the smaller this number is, the thicker the atmosphere will appear.

You must specify the half-atmosphere length in view space.

Also see: `render_atmosphere` and `render_atmosphere_index_ref`.

Keyboard Command Sequence:

`render_atmosphere_len <length>`
Command:  **render_awesome**

Description:  This command sets a variety of rendering parameters so that subsequent rendered images will be produced with *awesome* quality. This mode is appropriate for important images, where the high quality is worth waiting for. In awesome mode highlights, shadows and antialiasing are turned on. The reflection depth is set to eight, and the pixel size is set to one.

Also see:  `render`, `render_quick`, `render_standard`, `set_render_highlights`, `set_render_shadows`, `set_render_antialias`, `render_depth`, and `render_pixel_size`.

**Keyboard Command Sequence:**

`render_awesome`

---

Command:  **render_black_sky**

Description:  This command sets a black sky background for subsequent rendered images. The render zenith, horizon, and ground colors are set to black.

Also see:  `render`, `render_blue_sky`, `render_sunset`, `render_overcast`, `render_zenith`, `render_horizon`, and `render_ground`.

**Keyboard Command Sequence:**

`render_black_sky`

---

Command:  **render_blue_sky**

Description:  This command sets a blue sky background for subsequent rendered images. The render zenith, horizon, and ground colors are set to shades of blue.

Also see:  `render`, `render_black_sky`, `render_sunset`, `render_overcast`, `render_zenith`, `render_horizon`, and `render_ground`.

**Keyboard Command Sequence:**

`render_blue_sky`
CLEAR

**Command:**  render_clear

**Description:** This command clears the atmosphere for subsequent rendered images. Thus, no atmospheric depth cueing will be present.

Also see: render, render_fog, render_haze, render_smog, and render_atmosphere.

**Keyboard Command Sequence:**

render_clear

REFLECTIONS

**Command:**  render_depth

**Description:** This command allows you to specify the maximum number of reflections and refractions that will be followed along each ray, starting at the camera, for subsequent rendered images. During rendering, a light ray is traced backwards from the camera through each image pixel to the scene. Each time the ray hits an object in the scene, it can reflect off the object, and it can refract through the object (depending on the lighting material properties of the object). This command lets you specify how many times a light ray can be reflected or refracted before the rendering program terminates it. When this limit has been reached, the rendering program simply computes the color of the last object hit, by considering how all the lights in the scene illuminate the object at that point.

A high maximum number of reflections improves the quality of an image, at the expense of time needed to create the image.

You must specify the number of reflections.

Also see: render and render_standard.

**Keyboard Command Sequence:**

render_depth <# of reflections>
Command:  `render_fog`

Description: This command sets a gray atmosphere for subsequent rendered images. Thus, atmospheric depth cueing will be present. This means that the further away an object is, the more it will be obscured by fog.

Also see: `render`, `render_clear`, `render_haze`, `render_smog`, and `render_atmosphere`.

Keyboard Command Sequence:

```
render_fog
```

Command:  `render_ground`

Description: This command sets the ground color for subsequent rendered images. The ground color appears at infinity, below the horizon, which is 90 degrees away from the overhead up-vector direction.

You must specify the red, green, and blue components of the color. These values range from zero to one, where zero means completely OFF, and one means completely ON.

Also see: `render`, `render_black_sky`, `render_blue_sky`, `render_sunset`, `render_overcast`, `render_zenith`, and `render_horizon`.

Keyboard Command Sequence:

```
render_ground <red> <green> <blue>
```
HAZE

Command: render_haze
Description: This command sets a blue atmosphere for subsequent rendered images. Thus, atmospheric depth cueing will be present. This means that the further away an object is, the more it will be obscured by haze.

Also see: render, render_clear, render_fog, render_smog, and render_atmosphere.

Keyboard Command Sequence:
render_haze

HEIGHT

Command: render_height
Description: This command allows you to specify the pixel height of subsequent rendered images.

You must specify the image height.

Also see: render and render_width.

Keyboard Command Sequence:
render_height <height>

HORIZON COLOR

Command: render_horizon
Description: This command sets the horizon color for subsequent rendered images. The horizon color appears 90 degrees away from the overhead up-vector direction, at infinity. This color fades into the zenith color in rendered images.

You must specify the red, green, and blue components of the color. These values range from zero to one, where zero means completely OFF, and one means completely ON.
Also see: render, render_black_sky, render_blue_sky, render_sunset, render_overcast, render_zenith, and render_ground.

**Keyboard Command Sequence:**

```
render_horizon <red> <green> <blue>
```

**LINE RADIUS**

Command: `render_line_radius`

Description: This command allows you to specify the radius of lines in view space for subsequent rendered images. All 3-D lines in a view are rendered as cylinders. This command lets you set the radius of those cylinders. Setting the radius to zero turns OFF the rendering of lines completely.

You must specify the line radius in view space.

Also see: `render` and `render_symbol_radius`.

**Keyboard Command Sequence:**

```
render_line_radius <radius>
```

**OVERCAST SKY**

Command: `render_overcast`

Description: This command sets an overcast sky background for subsequent rendered images. The render zenith, horizon, and ground colors are set to shades of gray.

Also see: render, render_black_sky, render_blue_sky, render_sunset, render_zenith, render_horizon, and render_ground.

**Keyboard Command Sequence:**

```
render_overcast
```
**PIXEL SIZE**

**Command:** render_pixel_size

**Description:** This command allows you to specify the pixel size of subsequent rendered images. In standard mode, the pixel size is one, which means that a separate color is computed for every pixel in an image. If the pixel size is set to a higher number, an image can be created much faster, but with a lower resolution. For example, a pixel size of four means that the image resolution is now 16 times lower than in standard mode, and that such an image may be created in as little as 1/16 of the standard time.

You must specify the pixel size.

Also see: render and render_standard.

**Keyboard Command Sequence:**

```
render_pixel_size <size>
```

---

**QUICK**

**Command:** render_quick

**Description:** This command sets a variety of rendering parameters so that subsequent rendered images will be produced quickly. This can be helpful for getting a rough idea of how an image will turn out. In quick mode: highlights, shadows, and antialiasing are turned off. The reflection depth is set to one, and the pixel size is set to four.

With the reflection depth set to one, transparent materials such as glass and reflective materials such as chrome won’t look right. A higher depth is required for them to look correct.

Also see: render, render_standard, render_awesome, set_render_highlights, set_render_shadows, set_render_antialias, render_depth, and render_pixel_size.

**Keyboard Command Sequence:**

```
render_quick
```
Command: **render_smog**

**Description:** This command sets a brownish atmosphere for subsequent rendered images. Thus, atmospheric depth cueing will be present. This means that the further away an object is, the more it will be obscured by smog.

Also see: `render`, `render_clear`, `render_fog`, `render_haze`, and `render_atmosphere`.

**Keyboard Command Sequence:**

```
render_smog
```
### SUNSET SKY

**Command:** `render_sunset`  
**Description:** This command sets a sunset sky background for subsequent rendered images. The render zenith, horizon, and ground colors are set to sunset shades.  
Also see: `render`, `render_black_sky`, `render_blue_sky`, `render_overcast`, `render_zenith`, `render_horizon`, and `render_ground`.

**Keyboard Command Sequence:**  
`render_sunset`  

### SYMBOL RADIUS

**Command:** `render_symbol_radius`  
**Description:** This command allows you to specify the radius of symbols in view space for subsequent rendered images. All 3-D symbols (nodes, points, etc.) in a view are rendered as spheres. This command lets you set the radius of those spheres. Setting the radius to zero turns off the rendering of symbols completely.  
You must specify the symbol radius in view space.  
Also see: `render` and `render_line_radius`.

**Keyboard Command Sequence:**  
`render_symbol_radius <radius>`
**WIDTH**

Command:   `render_width`
Description:  This command allows you to specify the pixel width of subsequent rendered images.

You must specify the image width.

Also see: `render` and `render_height`.

Keyboard Command Sequence:

```plaintext
render_width <width>
```

**ZENITH COLOR**

Command:   `render_zenith`
Description:  This command sets the zenith color for subsequent rendered images. The zenith color appears overhead at infinity in the direction of the view up-vector. This color fades into the horizon color in rendered images.

You must specify the red, green, and blue components of the color. These values range from zero to one, where zero means completely OFF, and one means completely ON.

Also see: `render`, `render_black_sky`, `render_blue_sky`, `render_sunset`, `render_overcast`, `render_horizon`, and `render_ground`.

Keyboard Command Sequence:

```plaintext
render_zenith <red> <green> <blue>
```

**RENUMBER**

Menu:   `RENUMBER`
Description:  This menu contains commands that renumber geometric and mesh entities.
Command: renumber_all

Description: This command renumbers all entities in the model starting at the start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_all

Another Button with the same description:
RENUMBER ALL

Command: renumber_curves

Description: This command renumbers curves starting at the specified start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_curves

Command: renumber_elements

Description: This command renumbers elements starting at the specified start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_elements
**ELEMS DIRECTED**

**Command:** `renumber_elements_directed`

**Description:** This command renumbers elements in the specified direction starting at the start id (see `set_renumber_start`) and incrementing by the specified renumber increment (see `set_renumber_increment`).

**Keyboard Command Sequence:**

```
renumber_elements <direction vector>
```

**NODES**

**Command:** `renumber_nodes`

**Description:** This command renumbers nodes starting at the specified start id (see `set_renumber_start`) and incrementing by the specified increment (see `set_renumber_increment`).

**Keyboard Command Sequence:**

```
renumber_nodes
```

**NODES DIRECTED**

**Command:** `renumber_nodes_directed`

**Description:** This command renumbers nodes in the specified direction starting at the start id (see `set_renumber_start`) and incrementing by the specified renumber increment (see `set_renumber_increment`).

**Keyboard Command Sequence:**

```
renumber_nodes_directed <direction vector>
```
POINTS

Command: renumber_points
Description: This command renumbers points starting at the specified start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_points

SOLIDS

Command: renumber_solids
Description: This command renumbers solids starting at the specified start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_solids

SURFACES

Command: renumber_surfaces
Description: This command renumbers surfaces starting at the specified start id (see set_renumber_start) and incrementing by the specified increment (see set_renumber_increment).

Keyboard Command Sequence:
renumber_surfaces
Menu: REPAIR GEOMETRY
Description: This menu contains commands used to repair curves and surfaces for meshing.

Other Buttons with the same description:
    MANUAL
    OPTIONS

Command: repair_model
Description: This command resets internal consistency data in the model. This command is only to be used after the command check_model indicates a corrupted data structure.

Keyboard Command Sequence:
    repair_model

Command: reset
Description: This command resets all program variables to their default values.

Keyboard Command Sequence:
    reset
Command:  
**reset_all_views**

Description:  This command resets all views to their original (program startup) state. The camera position and settings, and model viewing transformations are all reset. This command does not alter translation, rotation, scale, or zoom increments.

This command acts on all views.

Also see:  reset_camera, reset_trans, load_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:  
```
reset_all_views
```

---

Command:  
**reset_camera**

Description:  This command restores the camera position and settings to their original (program startup) values.

This command acts on all the currently active views.

Also see:  reset_view, reset_trans, load_camera, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:  
```
reset_camera
```

---

Command:  
**reset_colors**

Description:  This command returns the program to its original color appearance. All colors are reset to their original values.

Keyboard Command Sequence:  
```
reset_colors
```
Command:  reset_font
Description: This command instructs the program to use its original text font. The
-font command line switch is used to specify the original font for
the program.

Keyboard Command Sequence:
reset_font

Command:  reset_lighting
Description: This command resets lighting in a view. The positions and colors and
other settings of all the lights, and the overall lighting model are reset to
the original values. You must specify the view.

Keyboard Command Sequence:
reset_lighting <view>

Command:  reset_trans
Description: This command resets the model viewing transformations to their original
(program startup) state.
This command acts on all the currently active views.
Also see: reset_view, reset_camera, load_trans, and the
topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
reset_trans
**RESET VIEW**

**Command:**  reset_view  

**Description:**  This command resets the view to its original (program startup) state. The view’s camera position and settings, and model viewing transformations are all reset. This command does not alter translation, rotation, scale, or zoom increments.

This command acts on all the currently active views.

Also see: reset_camera, reset_trans, load_view, and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

    reset_view

---

**RESTORE**

**Command:**  restore_model  

**Description:**  This command restores the model to the condition at the last save (see save_model). All changes to the model subsequent to saving it are deleted. To restore the model to its state before the last command, use the undo command.

**Keyboard Command Sequence:**

    restore_model

---

**RESULTS**

**Menu:**  RESULTS  

**Description:**  The RESULTS menu contains commands that control the viewing of Marc results. There are commands for selecting and manipulating Marc results files, selecting the results quantity for viewing, and for setting the viewing style and associated parameters.
Command: **retained_dof**

Description: This command sets the retained nodes’ degrees of freedom for servo link class links. You must specify the number of the retained node (slot) you wish to set and the retained degree of freedom.

Keyboard Command Sequence:
```
retained_dof <slot> <retained dof>
```

Command: **retained_node**

Description: This command sets the retained nodes for nodal tie and servo link class links. You must specify the number of the retained node (slot) you wish to set and the retained node.

Keyboard Command Sequence:
```
retained_node <slot> <retained node>
```

Other Buttons with the same description:

   NODE 1 through NODE 8

Menu: **RETURN**

Description: This button returns the program to the previous menu.
Menu: REVOLVE

Description: This menu contains commands for creating surfaces of revolution from curves.

For these commands to work correctly observe the following:

1. The local coordinate system’s y-axis (010) is used to revolve the curves around.
2. The curves should lie in the z=0 plane.
3. All the curves to be revolved must lie entirely to the right of the local coordinate systems y-axis (no portion of any curve should have negative x).

Command: revolve_angles

Description: This command sets the angles used by the revolve_curves command when creating surfaces of revolution. You must specify the starting and ending angles in degrees.

For correct surfaces of revolution, observe the following:

1. The local coordinate system’s y-axis (010) is used to revolve the curves around.
2. The curves should lie in the z=0 plane.
3. All the curves to be revolved must lie entirely to the right of the local coordinate system’s y-axis (no portion of any curve should have negative x).

Keyboard Command Sequence:

revolve_angles <start angle> <end angle>
Command:  **revolve_curves**

**Description:** This command creates a surface of revolution by revolving curves. Each curve is rotated in a positive direction about the local coordinate system’s y-axis from a specified start angle to a specified end angle (see `revolve_angles`). You must specify a list of curves.

For correct surfaces of revolution, observe the following:

1. The local coordinate systems y-axis (010) is used to revolve the curves around.
2. The curves should lie in the z=0 plane.
3. All the curves to be revolved must lie entirely to the right of the local coordinate systems y-axis (no portion of any curve should have negative x).

**Keyboard Command Sequence:**

```
revolve_curves <curve list> #
```

---

Command:  **revolve_reset**

**Description:** This command resets the angles used by the `revolve_curves` command when creating surfaces of revolution to their initial values.

For correct surfaces of revolution, observe the following:

1. The local coordinate system’s y-axis (010) is used to revolve the curves around.
2. The curves should lie in the z=0 plane.
3. All the curves to be revolved must lie entirely to the right of the local coordinate system’s y-axis (no portion of any curve should have negative x).

**Keyboard Command Sequence:**

```
revolve_reset
```
**REMOVE FREE CURVES**

Command:  `rm_free_curves`

Description: This command is used to remove curves that are not attached to any surfaces.

Keyboard Command Sequence:

`rm_free_curves`

---

**ALL+**

Command:  `rot_camera_cspace_c_for`

Description: This command rotates the camera about the current rotation/scale center point, along axes parallel to the camera space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

`rot_camera_cspace_c_for`
Command: rot_camera_cspace_c_rev
Description: This command rotates the camera about the current rotation/scale center point, along axes parallel to the camera space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

    rot_camera_cspace_c_rev

Command: rot_camera_cspace_x_for
Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space X-axis, in the positive sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

    rot_camera_cspace_x_for
Command: **rot_camera_cspace_x_rev**

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space X-axis, in the negative sense, by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_camera_cspace_x_rev
```

---

Command: **rot_camera_cspace_y_for**

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space Y-axis, in the positive sense, by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_camera_cspace_y_for
```
Command: **rot_camera_cspace_y_rev**

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space Y-command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_camera_cspace_y_rev
```

Command: **rot_camera_cspace_z_for**

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space Z-axis, in the positive sense, by an angle set with the rot_camera_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_camera_cspace_z_for
```
Z-

Command:  \texttt{rot\_camera\_cspace\_z\_rev}

Description:  This command rotates the camera about the current rotation/scale center point, along an axis parallel to the camera space Z-axis, in the negative sense, by an angle set with the \texttt{rot\_camera\_increment} command.  

This command acts on all the currently active views.

Also see:  \texttt{rot\_scale\_center\_lookat}, \texttt{rot\_camera\_increment}, \texttt{activate\_view}, and the topic \texttt{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{rot\_camera\_cspace\_z\_rev}

---

\textbf{ROTATE}

Command:  \texttt{rot\_camera\_increment}

Description:  This command allows you to set the rotation increment angle that is used by subsequent camera rotation commands such as \texttt{rot\_camera\_cspace\_x\_for}.  

See the topic \texttt{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{rot\_camera\_increment <angle>}

---
Command:  
**rot_camera_mspace_c_for**

Description:  This command rotates the camera about the current rotation/scale center point, along axes parallel to the model space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_camera_mspace_c_for
```

---

Command:  
**rot_camera_mspace_c_rev**

Description:  This command rotates the camera about the current rotation/scale center point, along axes parallel to the model space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_camera_mspace_c_rev
```
Command: rot_camera_mspace_x_for

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space X-axis, in the positive sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_camera_mspace_x_for

Command: rot_camera_mspace_x_rev

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space X-axis, in the negative sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_camera_mspace_x_rev
Command:  

**rot_camera_mspace_y_for**

**Description:** This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space Y-axis, in the positive sense, by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_camera_mspace_y_for
```

Command:  

**rot_camera_mspace_y_rev**

**Description:** This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space Y-axis, in the negative sense, by an angle set with the `rot_camera_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_camera_mspace_y_rev
```
Command: rot_camera_mspace_z_for

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space Z-axis, in the positive sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_camera_mspace_z_for

Command: rot_camera_mspace_z_rev

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the model space Z-axis, in the negative sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_camera_mspace_z_rev
**ALL+**

**Command:** rot_camera_vspace_c_for

**Description:** This command rotates the camera about the current rotation/scale center point, along axes parallel to the view space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

rot_camera_vspace_c_for

**ALL-**

**Command:** rot_camera_vspace_c_rev

**Description:** This command rotates the camera about the current rotation/scale center point, along axes parallel to the view space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

rot_camera_vspace_c_rev
X+

Command: rot_camera_vspace_x_for
Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space X-axis, in the positive sense, by an angle set with the rot_camera_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
rot_camera_vspace_x_for

X-

Command: rot_camera_vspace_x_rev
Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space X-axis, in the negative sense, by an angle set with the rot_camera_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
rot_camera_vspace_x_rev
Command: rot_camera_vspace_y_for

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space Y-axis, in the positive sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

    rot_camera_vspace_y_for

---

Command: rot_camera_vspace_y_rev

Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space Y-axis, in the negative sense, by an angle set with the rot_camera_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

    rot_camera_vspace_y_rev
Command: rot_camera_vspace_z_for
Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space Z-axis, in the positive sense, by an angle set with the rot_camera_increment command. This command acts on all the currently active views.
Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
rot_camera_vspace_z_for

Command: rot_camera_vspace_z_rev
Description: This command rotates the camera about the current rotation/scale center point, along an axis parallel to the view space Z-axis, in the negative sense, by an angle set with the rot_camera_increment command. This command acts on all the currently active views.
Also see: rot_scale_center_lookat, rot_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
rot_camera_vspace_z_rev
Command: **rot_model_cspace_c_for**

**Description:** This command rotates the model about the current rotation/scale center point, along axes parallel to the camera space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_cspace_c_for
```

---

Command: **rot_model_cspace_c_rev**

**Description:** This command rotates the model about the current rotation/scale center point, along axes parallel to the camera space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_cspace_c_rev
```
Command:  rot_model_cspace_x_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space X-axis, in the positive sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_cspace_x_for

Another Button with the same description:

X+

---

Command:  rot_model_cspace_x_rev

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space X-axis, in the negative sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_cspace_x_rev

Another Button with the same description:

X-
Command:  **rot_model_cspace_y_for**

Description:  This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space Y-axis, in the positive sense, by an angle set with the `rot_model_increment` command. This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_cspace_y_for
```

Another Button with the same description:

`Y+`

---

Command:  **rot_model_cspace_y_rev**

Description:  This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space Y-axis, in the negative sense, by an angle set with the `rot_model_increment` command. This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_cspace_y_rev
```

Another Button with the same description:

`Y-`
Command:  `rot_model_cspace_z_for`

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space Z-axis, in the positive sense, by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_model_cspace_z_for
```

Another Button with the same description:

`Z+`

Command:  `rot_model_cspace_z_rev`

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the camera space Z-axis, in the negative sense, by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_model_cspace_z_rev
```

Another Button with the same description:

`Z-`
**Mentat Help Commands in R**

**R-810**

---

**ROTATE**

**Command:** `rot_model_increment`

**Description:** This command allows you to set the rotation increment angle that will be used by subsequent viewing model rotation commands such as `rot_model_cspace_x_for`.

See the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_increment <angle>
```

---

**ALL+**

**Command:** `rot_model_mspace_c_for`

**Description:** This command rotates the model about the current rotation/scale center point, along axes parallel to the model space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_mspace_c_for
```
Command: rot_model_mspace_c_rev

Description: This command rotates the model about the current rotation/scale center point, along axes parallel to the model space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_c_rev

Command: rot_model_mspace_x_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space X-axis, in the positive sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_x_for
Command: rot_model_mspace_x_rev

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space X-axis, in the negative sense, by an angle set with the rot_model_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_x_rev

Command: rot_model_mspace_y_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space Y-axis, in the positive sense, by an angle set with the rot_model_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_y_for
Command: rot_model_mspace_y_rev

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space Y-axis, in the negative sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

See also rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_y_rev

Command: rot_model_mspace_z_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space Z-axis, in the positive sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_z_for
Command: rot_model_mspace_z_rev

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the model space Z-axis, in the negative sense, by an angle set with the rot_model_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_mspace_z_rev

Command: rot_model_vspace_c_for

Description: This command rotates the model about the current rotation/scale center point, along axes parallel to the view space X, Y, then Z-axes, each in the positive sense, and each by an angle set with the rot_model_increment command. This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_vspace_c_for
Command: rot_model_vspace_c_rev

Description: This command rotates the model about the current rotation/scale center point, along axes parallel to the view space Z, Y, then X-axes, each in the negative sense, and each by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_vspace_c_rev

---

Command: rot_model_vspace_x_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the view space X-axis, in the positive sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

rot_model_vspace_x_for
Command:  `rot_model_vspace_x_rev`

Description:  This command rotates the model about the current rotation/scale center point, along an axis parallel to the view space X-axis, in the negative sense, by an angle set with the `rot_model_increment` command.

This command acts on all the currently active views.

Also see: `rot_scale_center_lookat`, `rot_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
rot_model_vspace_x_rev
```
Command: rot_model_vspace_y_rev

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the view space Y-axis, in the negative sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

   rot_model_vspace_y_rev

Command: rot_model_vspace_z_for

Description: This command rotates the model about the current rotation/scale center point, along an axis parallel to the view space Z-axis, in the positive sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

   rot_model_vspace_z_for
**Command:** rot_model_vspace_z_rev

**Description:** This command rotates the model about the current rotation/scale center point, along an axis parallel to the view space Z-axis, in the negative sense, by an angle set with the rot_model_increment command.

This command acts on all the currently active views.

Also see: rot_scale_center_lookat, rot_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_model_vspace_z_rev
```

---

**Command:** rot_scale_center

**Description:** This command sets the current rotation/scale center point location. Thus, after issuing this command, rotations of the model or camera (using commands like rot_model_cspace_x_for) will take place in the camera space.

Also see: rot_scale_center_lookat, rot_scale_center_morigin, rot_scale_center_viewpoint, rot_scale_center_vorigin, and rot_scale_center_fixed.

**Keyboard Command Sequence:**

```
rot_scale_center <x> <y> <z>
```
**CAMERA SPACE**

**Command:** `rot_scale_center_cspace`

**Description:** This command sets the current rotation/scale center point to be in camera space. Thus, after issuing this command, rotations of the model or camera (using commands like `rot_model_cspace_x_for`) will take place in the camera space.

Also see: `rot_scale_center_fixed` and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

`rot_scale_center_cspace`

---

**LOOKAT**

**Command:** `rot_scale_center_lookat`

**Description:** This command sets the current rotation/scale center point to be the lookat point. Thus, after issuing this command, rotations of the model or camera (using commands like `rot_model_cspace_x_for`) will take place about the lookat point. Subsequent scaling of the model (using commands like `scale_model_up`) will also be about this point.

Also see: `view_lookat`, `rot_scale_center_viewpoint`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

`rot_scale_center_lookat`
**MODEL ORIGIN**

**Command:** `rot_scale_center_morigin`

**Description:** This command sets the current rotation/scale center point to be the model space origin. Thus, after issuing this command, rotations of the model or camera (using commands like `rot_model_cspace_x_for`) will take place about the model space origin. Subsequent scaling of the model (using commands like `scale_model_up`) will also be about this point.

Also see: `rot_scale_center_vorigin` and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_scale_center_morigin
```

---

**MODEL SPACE**

**Command:** `rot_scale_center_mspace`

**Description:** This command sets the current rotation/scale center point to be in model space. Thus, after issuing this command, rotations of the model or camera (using commands like `rot_model_cspace_x_for`) will take place in the model space.

Also see: `rot_scale_center_vorigin` and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
rot_scale_center_mspace
```
**VIEWPOINT**

**Command:** rot_scale_center_viewpoint  
**Description:** This command sets the current rotation/scale center point to be the viewpoint. Thus, after issuing this command, rotations of the model or camera (using commands like rot_model_cspace_x_for) will take place about the viewpoint. Subsequent scaling of the model (using commands like scale_model_up) will also be about this point.  
Also see: view_viewpoint, rot_scale_center_morigin, and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

rot_scale_center_viewpoint

**VIEW ORIGIN**

**Command:** rot_scale_center_vorigin  
**Description:** This command sets the current rotation/scale center point to be the view space origin. Thus, after issuing this command, rotations of the model or camera (using commands like rot_model_cspace_x_for) will take place about the view space origin. Subsequent scaling of the model (using commands like scale_model_up) will also be about this point.  
Also see: rot_scale_center_fixed and the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

rot_scale_center_vorigin
Command:  \texttt{rot\_scale\_center\_vspace}

Description:  This command sets the current rotation/scale center point to be in view space. Thus, after issuing this command, rotations of the model or camera (using commands like \texttt{rot\_model\_cspace\_x\_for}) will take place in the view space.

Also see: \texttt{rot\_scale\_center\_fixed} and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{rot\_scale\_center\_vspace}

---

Menu:  \texttt{rubber\_fitting}

Description:  Fit elastomeric model to data set.
Mentat Help Commands in S

SAMPLE ELEMENT

Command: sample
Description: This command samples nodal values over the specified element, giving both local coordinates of the sample and the value at those coordinates.

Keyboard Command Sequence:

    sample <element> <XYZ coordinates>

sample_volume

Command: sample_volume
Description: This command samples the elements of an entire model placing the results in the specified file. The samples are located over the bounding cube with the number of divisions specified.

Keyboard Command Sequence:

    sample_volume <file name> <divisions>
Command: save_as_model
Description: This command saves a model under the specified name. All subsequent save_model commands will use the name given in this command. The format for the model file is affected by the style set with the set_save_as_version command. This button is located next to the SAVE AS button.

Keyboard Command Sequence:
    save_as_model <name>

Command: save_colors
Description: This command saves the current color table to a file. You must specify the name of the file in which to save the color table.

Keyboard Command Sequence:
    save_colors <file name> <yes/no>

Command: save_lighting
Description: This command saves the lighting model in a view to a file. You must specify the view and the name of the file.
    Also see: load_lighting.

Keyboard Command Sequence:
    save_lighting <view> <file name> <yes or no>
Command: **save_model**

Description: This command saves the model under its current name. By default, the current name is set to `model#x`, where `x` is a digit. To save the model under a different name, use the `save_as_model` command. The format of the model file is always the default style and is not affected by the `set_save_as_version` command.

Keyboard Command Sequence:

```
save_model
```

Other Buttons with the same description:

SAVE
SAVE & EXIT
SAVE MODEL

Command: **save_view**

Description: This command saves a view’s settings into a view file. The view’s camera position and settings and model viewing transformations are stored in the file. You must specify the view to be saved. Views are specified by number, 1 to 4. You must also specify the name of the view file.

Also see: `load_camera`, `load_trans`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
save_view <view> <filename>
```
Command: scale_model_down
Description: This command scales all active views using the current scale factor set by the scale_model_increment command. Scaling down divides the views’ scale factors by the scale increment.

Also see: scale_model_increment.

Keyboard Command Sequence:
scale_model_down

Command: scale_model_increment
Description: This command allows you to set the scale factor that is used by subsequent scale_model_up and scale_model_down commands. See the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
scale_model_increment <factor>

Command: scale_model_up
Description: This command scales all active views using the current scale factor set by the scale_model_increment command. Scaling up multiplies the views’ scale factors by the scale increment.

Also see: scale_model_increment.

Keyboard Command Sequence:
scale_model_up
SECOND ORDER INVARIANT

Command:  \texttt{xcv\_model second\_order\_invariant}

Description:  This command is used if experimental data must be fitted using the second order invariant strain energy function $W$, which is given by:

$$W = C_{10}*(I_1 - 3) + C_{01}*(I_2 - 3) + C_{11}*(I_1 - 3)*(I_2 - 3) + C_{20}*(I_1 - 3)^2$$

where $I_1$ and $I_2$ are the first and second invariant of the right Cauchy-Green strain tensor and $C_{10}$, $C_{01}$, $C_{11}$, and $C_{20}$ are the material parameters to be determined.

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (\texttt{material\_type mechanical: mooney}). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

Keyboard Command Sequence:

\begin{verbatim}
xcv_model second_order_invariant
\end{verbatim}

ADD ARC

Command:  \texttt{section\_add\_arc}

Description:  This command adds an arc branch to the current beam section.

Note that branches must be ordered \textit{heat-to-toe} in such a way that the entire cross-section is traversed. This means that zero thickness sections may be necessary to complete the traversal.

Keyboard Command Sequence:

\begin{verbatim}
section_add_arc <center point X, Y> <radius> 
<beginning thickness> <ending thickness> 
<number of stress divisions (must be even)>
\end{verbatim}
Command: `section_add_line`

Description: This command adds a line branch to the current beam section. Note that branches must be ordered *heat-to-toe* in such a way that the entire cross-section is traversed. This means that zero thickness sections may be necessary to complete the traversal.

Keyboard Command Sequence:

```
section_add_line <first point X, Y>
    <second point X, Y> <beginning thickness>
    <ending thickness>
    <number of stress divisions (must be even)>
```
Command: **section_edit_branch**

Description: This command is used to edit a branch by respecifying its end thicknesses and the number of stress divisions.

Keyboard Command Sequence:

```
section_edit_branch <branch number>
<beginning thickness> <ending thickness>
<number of stress divisions (must be even)>
```

Command: **section_filled**

Description: This command specifies whether beam sections should be drawn filled or unfilled.

Keyboard Command Sequence:

```
section_filled <on or off>
```

Command: **section_name**

Description: This command sets or changes the name of the current beam section.

A beam section is a set of dimensions and properties defining a beam cross-section. Sections may be referenced by geometric properties of type general beam.

Keyboard Command Sequence:

```
section_name <beam section name>
```
Command: **section_remove_branch**

**Description:** This command removes branches from the current beam section. Note that branches must be ordered *heat-to-toe* in such a way that the entire cross-section is traversed. This means that zero thickness sections may be necessary to complete the traversal.

**Keyboard Command Sequence:**

```
section_remove_branch <branch numbers>
```
Command: select_by_any
Description: This command specifies that the select by commands operate in ANY IN LIST mode. The influence of the select by mode is explained in the help texts on the select by commands. This mode is the default mode.
Also see: select_by_all.

Keyboard Command Sequence:
select_by_any

Command: select_clear
Description: This command clears the current selection. Using this command does not change the selection method or mode, nor does it erase any previously defined sets.

Keyboard Command Sequence:
select_clear

Command: select_contact_body
Description: This command adds all elements, nodes, surfaces, curves, and points that belong to the specified contact body to the respective lists of selected entities.

Keyboard Command Sequence:
select_contact_body <contact body name>
Command: **select_curves**

Description: This command adds curves to the list of selected curves. Once curves are selected, they are used to define a set with the `store_curves` command.

The method for specifying curves depends on the select method being used (see `select_method_single`, `select_method_path`, `select_method_box`, `select_method_user_box`, `select_method_plane`, `select_method_flood`, `select_method_point_dist`, `select_method_curve_dist`, `select_method_surface_dist`).

Keyboard Command Sequence:

```
select_curves
```

Command: **select_curves_cbody**

Description: This command adds all curves that belong to the specified contact body to the list of selected curves. Once curves are selected, they are used to define a set with the `store_curves` command.

Keyboard Command Sequence:

```
select_curves_cbody <contact body name>
```
**NODES**

**Command:** `select_curves_nodes`

**Description:** This command adds curves to the list of selected curves, based on a list of nodes given by the user. Which curves are added depends on the value of the `select by` mode:

- **ANY IN LIST:** curves to which *any* node in the list is attached
- **ALL IN LIST:** curves to which *all* nodes in the list are attached

Once curves are selected, they are used to define a set with the `store_curves` command.

**Keyboard Command Sequence:**

```
select_curves_nodes <node list> #
```

---

**POINTS**

**Command:** `select_curves_points`

**Description:** This command adds curves to the list of selected curves, based on a list of points given by the user. Which curves are added depends on the value of the `select by` mode:

- **ANY IN LIST:** curves for which *any* point is included in the list
- **ALL IN LIST:** curves for which *all* points are included in the list

Once curves are selected, they are used to define a set with the `store_curves` command.

**Keyboard Command Sequence:**

```
select_curves_points <point list> #
```
Command: **select_edges**

Description: This command adds edges to the list of selected edges. Once edges are selected, they can be used to define a set with the `store_edges` command.

The method for specifying edges depends on the select method being used (see `select_method_single`, `select_method_path`, `select_method_box`, `select_method_user_box`, `select_method_plane`, `select_method_flood`, `select_method_point_dist`, `select_method_curve_dist`, `select_method_surface_dist`).

Keyboard Command Sequence:

```
select_edges
```

---

Command: **select_edges_elements**

Description: This command adds edges to the list of selected edges, based on a list of elements given by the user. Edges that belong to any element in the list, will be added. The command is not influenced by the value of the `select by` mode. Once edges are selected, they are used to define a set with the `store_edges` command.

Keyboard Command Sequence:

```
select_edges_elements <element list> #
```
**FACES**

**Command:** `select_edges_faces`  
**Description:** This command adds edges to the list of selected edges, based on a list of faces given by the user. Which edges are added depends on the value of the `select by` mode:

- **ANY IN LIST:** edges that belong to *any* face in the list  
- **ALL IN LIST:** edges that belong to *all* faces in the list; a maximum of two faces may be given.

Once edges are selected, they are used to define a set with the `store_edges` command.

**Keyboard Command Sequence:**

```
select_edges_faces <face list> #
```
**ELEMENTS**

**Command:** select_elements

**Description:** This command adds elements to the list of selected elements. Once elements are selected, they are used to define a set with the `store_elements` command.

The method for specifying elements depends on the select method being used (see `select_method_single`, `select_method_path`, `select_method_box`, `select_method_user_box`, `select_method_plane`, `select_method_flood`, `select_method_point_dist`, `select_method_curve_dist`, `select_method_surface_dist`).

**Keyboard Command Sequence:**

select_elements

**CONT BODY**

**Command:** select_elements_cbody

**Description:** This command adds all elements that belong to the specified contact body to the list of selected elements. Once elements are selected, they are used to define a set with the `store_elements` command.

**Keyboard Command Sequence:**

select_elements_cbody <contact body name>

**OK**

**Command:** select_elements_class

**Description:** This command adds all elements in the model of the specified class to the list of selected elements. Once elements are selected, they are used to define a set with the `store_elements` command.

**Keyboard Command Sequence:**

select_elements_class <element class>
FACES

Command: select_elements_faces
Description: This command adds elements to the list of selected elements, based on a list of faces given by the user. Which elements are added depends on the value of the select by mode:

ANY IN LIST: elements for which any face is included in the list
ALL IN LIST: elements for which all faces are included in the list

Once elements are selected, they are used to define a set with the store_elements command.

Keyboard Command Sequence:
select_elements_faces <face list> #

GEOMETRY

Command: select_elements_geometry
Description: This command adds all elements in the model with the specified geometric property to the list of selected elements. Once elements are selected, they are used to define a set with the store_elements command.

Keyboard Command Sequence:
select_elements_geometry <geometry name>

MATERIAL

Command: select_elements_material
Description: This command adds all elements in the model with the specified material to the list of selected elements. Once elements are selected, they are used to define a set with the store_elements command.

Keyboard Command Sequence:
select_elements_material <material name>
Command: **select_elements_nodes**

Description: This command adds elements to the list of selected elements, based on a list of nodes given by the user. Which elements are added depends on the value of the *select by* mode:

- **ANY IN LIST**: elements for which *any* node is included in the list
- **ALL IN LIST**: elements for which *all* nodes are included in the list

Once elements are selected, they are used to define a set with the `store_elements` command.

**Keyboard Command Sequence:**

```
select_elements_nodes <node list> #
```
Command: **select_faces**

Description: This command adds faces to the list of selected faces. Once faces are selected, they are used to define a set with the **store_faces** command.

The method for specifying faces depends on the select method being used (see **select_method_single**, **select_method_path**, **select_method_box**, **select_method_user_box**, **select_method_plane**, **select_method_flood**, **select_method_point_dist**, **select_method_curve_dist**, **select_method_surface_dist**).

Keyboard Command Sequence:

```
select_faces
```

Command: **select_faces_edges**

Description: This command adds faces to the list of selected faces, based on a list of edges given by the user. Which faces are added depends on the value of the **select by** mode:

- **ANY IN LIST**: faces for which *any* edge is included in the list
- **ALL IN LIST**: faces for which *all* edges are included in the list

Once faces are selected, they are used to define a set with the **store_faces** command.

Keyboard Command Sequence:

```
select_faces_edges <edge list> #
```
Command: **select_faces_elements**

**Description:** This command adds faces to the list of selected faces, based on a list of elements given by the user. Faces that belong to any element in the list, are added. The command is not influenced by the value of the `select by` mode. Once faces are selected, they are used to define a set with the `store_faces` command.

**Keyboard Command Sequence:**

```
select_faces_elements <element list> #
```

---

Command: **select_faces_nodes**

**Description:** This command adds faces to the list of selected faces, based on a list of nodes given by the user. Which faces are added depends on the value of the `select by` mode:

- ANY IN LIST: faces for which *any* node is included in the list
- ALL IN LIST: faces for which *all* nodes are included in the list

Once faces are selected, they are used to define a set with the `store_faces` command.

**Keyboard Command Sequence:**

```
select_faces_nodes <node list> #
```

---

Command: **select_filter_none**

**Description:** This command specifies that no filters will be used by the select commands. You may specify filtering with the `select_filter_outline` and `select_filter_surface` commands.

**Keyboard Command Sequence:**

```
select_filter_none
```
Command: select_filter_outline
Description: This command specifies that outline filtering are used by the select command. With outline filtering, only those entities on the outline of the model are selected by the select commands.

Keyboard Command Sequence:

select_filter_outline

Command: select_filter_surface
Description: This command specifies that surface filtering are used by the select command. With surface filtering, only those entities on the surface of the model are selected by the select commands.

Keyboard Command Sequence:

select_filter_surface

Command: select_method_association
Description: This command specifies that select commands use the association method. This method allows the user to select items and other items in the set that those items belong to will also become selected.

Keyboard Command Sequence:

select_method_association
Command: **select_method_box**

Description: This command specifies that select commands use the box method. This method allows the user to select items within a specified region in space. All items that fall within the specified box (global coordinate system) become selected.

**Keyboard Command Sequence:**

```plaintext
select_method_box
```

Command: **select_method_curve_dist**

Description: This command specifies that select commands use the curve distance method. This method allows the user to select items that fall within the SELECT DISTANCE of a curve. The SELECT DISTANCE can be specified using the `*set_select_distance` command. Only items that fall entirely within the specified distance will be selected.

This button is located under the UTILS->SELECT->METHOD menu.

**Keyboard Command Sequence:**

```plaintext
select_method_curve_dist
```

Command: **select_method_flood**

Description: This command specifies that select commands use the flood method. To use this method, first define a boundary of selected nodes and then activate the flood method. The method then works by selecting all entities of the desired type (points, curves, surfaces, nodes, or elements) that are connected to the specified node and are within the boundary of previously selected nodes.

**Keyboard Command Sequence:**

```plaintext
select_method_flood
```
Command: \texttt{select\_method\_path}

Description: This command specifies that select commands use the path method. This method allows the user to select connected nodes or points along a path from the first entity specified to the last. You may specify only the beginning and end of the path, or choose entities at various locations along the path.

Keyboard Command Sequence:

\texttt{select\_method\_path}

Command: \texttt{select\_method\_plane}

Description: This command specifies that select commands use the plane method. This method allows the user to select nodes or elements in a specified plane by choosing an element face. All nodes or elements which are in the plane specified by the centroid of the face and the face normal at the centroid will be selected.

Keyboard Command Sequence:

\texttt{select\_method\_plane}

Command: \texttt{select\_method\_point\_dist}

Description: This command specifies that select commands use the point distance method. This method allows the user to select items that fall within the SELECT DISTANCE of a point in space. The SELECT DISTANCE can be specified using the \texttt{*set\_select\_distance} command. Only items that fall entirely within the specified distance will be selected.

This button is located under the UTILS->SELECT->METHOD menu.

Keyboard Command Sequence:

\texttt{select\_method\_point\_dist}
Command: **select_method_single**

Description: This command specifies that select commands use the single method. This method allows the user to select entities singly by specifying their ids. This method is the default method.

**Keyboard Command Sequence:**

```
select_method_single
```
If the user coordinate system is SPHERICAL, the box is defined by specifying the ranges in the R-, PHI-, and THETA-direction of the coordinate system. The angles that specify the range in the THETA direction must be between -90 and +90 degrees.

This button is located under the UTILS->SELECT->METHOD menu.

**Keyboard Command Sequence:**

```
select_method_user_box
```

---

**Command:** `select_mode_and`

**Description:** This command specifies that select commands operate in \textit{and} mode. In this mode, additionally selected entities are \textit{and}-ed with those already in the selected list, thus adding them to the list. This mode is the default mode.

Also see: `select_mode_except`, `select_mode_invert`, and `select_mode_intersect`.

**Keyboard Command Sequence:**

```
select_mode_and
```

---

**Command:** `select_mode_except`

**Description:** This command specifies that select commands operate in \textit{except} mode. In this mode, additionally selected entities are removed from the list of selected entities.

Also see: `select_mode_and`, `select_mode_invert`, and `select_mode_intersect`.

**Keyboard Command Sequence:**

```
select_mode_except
```
**INTERSECT**

**Command:** select_mode_intersect

**Description:** This command specifies that select commands operate in intersect mode. In this mode, additionally selected entities are intersected with the current list of selected entities to form the new list.

Also see: select_mode_and, select_mode_except, and select_mode_invert.

**Keyboard Command Sequence:**

    select_mode_intersect

**INVERT**

**Command:** select_mode_invert

**Description:** This command specifies that select commands operate in invert mode. In this mode, additionally selected entities are added to the list of selected entities if they do NOT belong to it and are removed if they do already belong to it.

Also see: select_mode_and, select_mode_except, and select_mode_intersect.

**Keyboard Command Sequence:**

    select_mode_invert

**NODES**

**Command:** select_nodes

**Description:** This command adds nodes to the list of selected nodes. Once nodes are selected, they are used to define a set with the store_nodes command.

The method for specifying nodes depends on the select method being used (see select_method_single, select_method_path, select_method_box, select_method_user_box, select_method_plane, select_method_flood, select_method_point_dist,
select_method_curve_dist,
select_method_surface_dist).

**Keyboard Command Sequence:**

```
select_nodes
```

---

**Command:** select_nodes_curves

**Description:** This command adds nodes to the list of selected nodes, based on a list of curves given by the user. Nodes that are attached to any curve in the list, will be added. The command is not influenced by the value of the select by mode. Once nodes are selected, they are used to define a set with the store_nodes command.

**Keyboard Command Sequence:**

```
select_nodes_curves <curve list> #
```

---

**Command:** select_nodes_edges

**Description:** This command adds nodes to the list of selected nodes, based on a list of edges given by the user. Which nodes are added depends on the value of the select by mode:

- ANY IN LIST: nodes that belong to *any* edge in the list
- ALL IN LIST: nodes that belong to *all* edges in the list

Once nodes are selected, they are used to define a set with the store_nodes command.

**Keyboard Command Sequence:**

```
select_nodes_edges <edge list> #
```
Command: \texttt{select\_nodes\_elements}

Description: This command adds nodes to the list of selected nodes, based on a list of elements given by the user. Which nodes are added depends on the value of the \textit{select by} mode:

- \texttt{ANY IN LIST}: nodes that belong to \textit{any} element in the list
- \texttt{ALL IN LIST}: nodes that belong to \textit{all} elements in the list

Once nodes are selected, they are used to define a set with the \texttt{store\_nodes} command.

Keyboard Command Sequence:
\begin{verbatim}
select\_nodes\_elements <element list> #
\end{verbatim}

Command: \texttt{select\_nodes\_faces}

Description: This command adds nodes to the list of selected nodes, based on a list of faces given by the user. Which nodes are added depends on the value of the \textit{select by} mode:

- \texttt{ANY IN LIST}: nodes that belong to \textit{any} face in the list
- \texttt{ALL IN LIST}: nodes that belong to \textit{all} faces in the list

Once nodes are selected, they are used to define a set with the \texttt{store\_nodes} command.

Keyboard Command Sequence:
\begin{verbatim}
select\_nodes\_faces <face list> #
\end{verbatim}

Command: \texttt{select\_nodes\_points}

Description: This command adds nodes to the list of selected nodes, based on a list of points given by the user. Nodes that are attached to any point in the list, are added. The command is not influenced by the value of the \textit{select by} mode.
Once nodes are selected, they are used to define a set with the `store_nodes` command.

**Keyboard Command Sequence:**

```
select_nodes_points <point list> #
```

---

**SURFACES**

**Command:** `select_nodes_surfaces`

**Description:** This command adds nodes to the list of selected nodes, based on a list of surfaces given by the user. Which nodes are added depends on the value of the `select by` mode:

- **ANY IN LIST:** nodes that are attached to *any* surface in the list
- **ALL IN LIST:** nodes that are attached to *all* surfaces in the list; a maximum of two surfaces may be given.

Once nodes are selected, they are used to define a set with the `store_nodes` command.

**Keyboard Command Sequence:**

```
select_nodes_surfaces <surface list> #
```

---

**TRANSFORM**

**Command:** `select_nodes_transform`

**Description:** This command adds all nodes in the model with the specified transformation to the list of selected nodes.

Once nodes are selected, they are used to define a set with the `store_nodes` command.

**Keyboard Command Sequence:**

```
select_nodes_transform <transform name>
```
Command: `select_points`

Description: This command adds points to the list of selected points. Once points are selected, they are used to define a set with the `store_points` command.

The method for specifying points depends on the select method being used (see `select_method_single`, `select_method_path`, `select_method_box`, `select_method_user_box`, `select_method_plane`, `select_method_flood`, `select_method_point_dist`, `select_method_curve_dist`, `select_method_surface_dist`).

Keyboard Command Sequence:

```
select_points
```

---

Example:

```
select_points
```

---

Command: `select_points_curves`

Description: This command adds points to the list of selected points, based on a list of curves given by the user. Which points are added depends on the value of the `select by` mode:

- **ANY IN LIST**: points that belong to *any* curve in the list
- **ALL IN LIST**: points that belong to *all* curves in the list

Once points are selected, they are used to define a set with the `store_points` command.

Keyboard Command Sequence:

```
select_points_curves <curve list> #
```
**NODES**

**Command:** `select_points_nodes`  
**Description:** This command adds points to the list of selected points, based on a list of nodes given by the user. Which points are added depends on the value of the `select by` mode:

- **ANY IN LIST:** points to which *any* node in the list is attached
- **ALL IN LIST:** points to which *all* nodes in the list are attached

Once points are selected, they are used to define a set with the `store_points` command.  
**Keyboard Command Sequence:**
```
select_points_nodes <node list> #
```

**SURFACES (DFN)**

**Command:** `select_points_surfaces`  
**Description:** This command adds points to the list of selected points, based on a list of surfaces given by the user. Which points are added depends on the value of the `select by` mode:

- **ANY IN LIST:** defining points that belong to *any* surface in the list
- **ALL IN LIST:** defining points that belong to *all* surfaces in the list

Once points are selected, they are used to define a set with the `store_points` command.  
**Keyboard Command Sequence:**
```
select_points_surfaces <surface list> #
```

**RESET**

**Command:** `select_reset`  
**Description:** This command resets select parameters, method, mode, filter, and `select by` mode to their default values. This command does not alter the lists of currently selected entities.  
**Keyboard Command Sequence:**
```
select_reset
```
**SELECT BY SCALAR VALUE**

**Menu:**  
**SELECT BY SCALAR VALUE**

**Description:** This menu provides tools for adding nodes or elements to the list of selected nodes or elements based upon the values of the scalar quantity that is currently being post processed. Only nodes or elements that are currently visible and that are not removed from post plotting using the `unpost_nodes` or `unpost_elements` commands are considered. The commands always consider the extrapolated values of the scalar quantity at the nodes of the elements based on the current extrapolation scheme, even when selecting elements.

If some elements have been isolated using the `post_isolated` command (i.e. averaging of element values across element edges is switched off between the set of isolated elements and the set of non-isolated elements) the commands will consider the isolated values and the nodes of those elements.

**SELECT BY EXTREMES**  
These tools add all nodes or elements to the list of selected nodes or elements in which the scalar quantity assumes its minimum or maximum value.

**SELECT NODES IN RANGE**  
**SELECT ELEMENTS IN RANGE**

These commands add nodes or elements to the list of selected nodes or elements in which the scalar quantity assumes values in a given RANGE.

**Keyboard Command Sequence:**

- `select_elements_scalar_min`
- `select_elements_scalar_max`

- `select_nodes_scalar_min`
- `select_nodes_scalar_max`

- `select_scalar_range <lower_bound> <upper_bound>`
- `select_nodes_scalar_range`
- `select_elements_scalar_range`

**Related Commands:**

- `post_extrap_linear/post_extrap_translate/
  post_extrap_average`
post_elements/unpost_elements
post_nodes/unpost_nodes
post_isolate

Other Buttons with the same description:
MINIMUM
RANGE
SELECT ELEMENTS IN RANGE
SELECT NODES IN RANGE

Command: select_sets
Description: This command selects a previously defined set of entities (see store_curves, store_edges, store_elements, store_faces, store_nodes, store_points, and store_surfaces) and applies them to the current list of selected entities with the current select method, mode, and filter. You must specify the name of a previously defined set.

Keyboard Command Sequence:
select_set <set name>

Command: select_surfaces
Description: This command adds surfaces to the list of selected surfaces. Once surfaces are selected, they are used to define a set with the store_surfaces command.

The method for specifying surfaces depends on the select method being used (see select_method_single, select_method_path, select_method_box, select_method_user_box, select_method_plane, select_method_flood, select_method_point_dist, select_method_curve_dist, select_method_surface_dist).

Keyboard Command Sequence:
select_surfaces
**CONT BODY**

**Command:** `select_surfaces_cbody`

**Description:** This command adds all surfaces that belong to the specified contact body to the list of selected surfaces. Once surfaces are selected, they are used to define a set with the `store_surfaces` command.

**Keyboard Command Sequence:**

```
select_surfaces_cbody <contact body name>
```

---

**NODES**

**Command:** `select_surfaces_nodes`

**Description:** This command adds surfaces to the list of selected surfaces, based on a list of nodes given by the user. Which surfaces are added depends on the value of the `select by` mode:

- **ANY IN LIST**: surfaces to which *any* node in the list is attached
- **ALL IN LIST**: surfaces to which *all* nodes in the list are attached

Once surfaces are selected, they are used to define a set with the `store_surfaces` command.

**Keyboard Command Sequence:**

```
select_surfaces_nodes <node list> #
```

---

**POINTS (DFN)**

**Command:** `select_surfaces_points`

**Description:** This command adds surfaces to the list of selected surfaces, based on a list of points given by the user. Which surfaces are added depends on the value of the `select by` mode:

- **ANY IN LIST**: surfaces for which *any* defining point is included in the list
- **ALL IN LIST**: surfaces for which *all* defining points are included in the list
Once surfaces are selected, they are used to define a set with the `store_surfaces` command.

**Keyboard Command Sequence:**

```
select_surfaces_points <point list> #
```

---

**CURVES**

**Command:** `select_surfaces_trim_curves`

**Description:** This command adds surfaces to the list of selected surfaces, based on a list of curves given by the user. Which surfaces are added depends on the value of the `select by` mode:

- **ANY IN LIST:** surfaces for which *any* trimming curve is included in the list
- **ALL IN LIST:** surfaces for which *all* trimming curves are included in the list

Once surfaces are selected, they are used to define a set with the `store_surfaces` command.

**Keyboard Command Sequence:**

```
select_surfaces_trim_curves <curve list> #
```

---

**POINTS (TRIM)**

**Command:** `select_surfaces_trim_points`

**Description:** This command adds surfaces to the list of selected surfaces, based on a list of points given by the user. Which surfaces are added depends on the value of the `select by` mode:

- **ANY IN LIST:** surfaces for which *any* trimming point is included in the list
- **ALL IN LIST:** surfaces for which *all* trimming points are included in the list

Once surfaces are selected, they are used to define a set with the `store_surfaces` command.

**Keyboard Command Sequence:**

```
select_surfaces_trim_points <point list> #
```
SURFACES

Command: select_trim_curves_surfaces
Description: This command adds curves to the list of selected curves, based on a list of surfaces given by the user. The trimming curves of any surface in the list are added. The command is not influenced by the value of the select by mode.

Once curves are selected, they are used to define a set with the store_curves command.

Keyboard Command Sequence:
select_trim_curves_surfaces <surface list> #

SURFACES (TRIM)

Command: select_trim_points_surfaces
Description: This command adds points to the list of selected points, based on a list of surfaces given by the user. The trimming points of any surface in the list are added. The command is not influenced by the value of the select by mode.

Once points are selected, they are used to define a set with the store_points command.

Keyboard Command Sequence:
select_trim_points_surfaces <surface list> #

SEPFOR

Subroutine: SEPFOR
Description: The user subroutine SEPFOR allows you to define the separation force in conjunction with the model definition option CONTACT. The separation forces, FNORM and FTANG, are either calculated by the program or entered through the CONTACT option, and are then passed into this subroutine. You decide whether this magnitude at the current increment is good enough to determine whether separation will occur.

FNORM is the normal reaction force above which a node in contact will separate from a surface. Any compressive or negative value indicates
real contact, while a positive reaction force indicates a tendency to separate. The default is taken as the maximum value of the residual force in the structure for the current increment. This value can be reset in the input deck. Defining a small value may result in an increased number of iterations. Defining a large value will eliminate the possibility of separation.

**set_acis_formatted**

Command: **set_acis_formatted**

Description: This command sets the format of acis files read and written with the read_acis and write_acis commands. Formatted acis files are ascii text files and are editable and portable across machine architectures. Unformatted (binary) database files are not always architecture independent but are smaller and are processed more quickly.

Keyboard Command Sequence:

```
set_acis_formatted <on or off>
```

**set_annotations**

Command: **set_annotations**

Description: This command toggles the display of annotations. When ON, all annotations are displayed. When OFF, no annotations are displayed. Annotations are strings of text that are used to describe what is being displayed in a view.

Keyboard Command Sequence:

```
set_annotations <on or off>
```

Another Button with the same description:

DISPLAY ANNOTATIONS
**CORIOLIS**

Command:  `set_apply_coriolis`
Description: This command specifies whether or not Coriolis type loading will be applied for the current centrifugal boundary condition.

Keyboard Command Sequence:

```
set_apply_coriolis <on or off>
```

**ENTERED VALUES**

Command:  `set_apply_entered`
Description: This command specifies that values for the current boundary condition are being entered by the user (as opposed to being provided through a user subroutine; see `set_apply_usersub`).

Keyboard Command Sequence:

```
set_apply_entered
```

**BOUND CONDS**

Command:  `set_apply_labels`
Description: This command toggles the display of boundary condition labels.

Keyboard Command Sequence:

```
set_apply_labels <on or off>
```

**POST FILE**

Command:  `set_apply_post_tape`
Description: This command specifies that values for the current state variable boundary condition will be determined at run time by reading a previously generated post file.

Keyboard Command Sequence:

```
set_apply_post_tape
```
**USER SUB.**

**Command:** set_apply_usersub

**Description:** This command specifies that Marc should call the appropriate user subroutine for this boundary condition during the analysis. If the user subroutine has been invoked, there is no need to enter values for the application since they will be ignored by Marc in lieu of the values provided by the subroutine.

If the user subroutine is used, the FORTRAN source code file must exist in your current working directory, so that it can be compiled and linked in with Marc when you submit the job (see `job_usersub_file`).

**Keyboard Command Sequence:**

```
set_apply_usersub <on or off>
```

**BOUND CONDS**

**Command:** set_applys

**Description:** This command toggles the display of boundary conditions.

**Keyboard Command Sequence:**

```
set_applys <on or off>
```

**HEADS**

**Command:** set_arrow_heads

**Description:** This command toggles the plotting of head of the arrows when pre- and postprocessing are plotted. By default, heads are on. You will need to do a `DRAW`, `REDRAW`, or `REGEN` to display the change.

Also see: `arrows_solid` and `arrow_wireframe`.

**Keyboard Command Sequence:**

```
set_arrow_heads <on or off>
```
Command: **set_arrow_lines**

Description: This command toggles the plotting of facet edges when pre- and postprocessing arrows are plotted in solid mode. Be sure to also use the `arrows_solid` command to select solid plotting of arrows (wireframe plotting is the default). By default, facet edges are on. You may wish to turn them off if you have turned lighting on. You will need to do a `DRAW`, `REDRAW`, or `REGEN` to display the change.

Also see: `arrows_solid` and `arrow_facets`.

Keyboard Command Sequence:

```
set_arrow_lines <on or off>
```

---

Command: **set_attach_direction**

Description: This command sets the attach direction to be used in directed mode, while attaching edges to curves and faces to surfaces. You must specify the components of a vector defining the attach direction.

Also see: `attach_mode_directed`  
`attach_edges_curve`  
`attach_faces_surface`  
`attach_elements_curve`  
`attach_elements_surface`

Keyboard Command Sequence:

```
set_attach_direction <XYZ components of direction>
```
DISTANCE

Command: **set_attach_distance**

Description: This command sets the distance limit for attaching nodes to points, edges to curves, and faces to surfaces. Items that fall (either totally or partially) beyond the specified distance will not be attached if the attach limit has been activated.

Also see: set_attach_limit
attach_nodes_point
attach_edges_curve
attach_faces_surface
attach_elements_curve
attach_elements_surface

Keyboard Command Sequence:

```
set_attach_distance <distance>
```

DIVISIONS

Command: **set_attach_divisions**

Description: This command sets the number of curve (surface) divisions used by the attach commands for determining node locations, when moving nodes to curves (surfaces). The higher the specified divisions the more accurately the nodes or elements will be attached.

Also see: attach_edges_curve
attach_faces_surface
attach_elements_curve
attach_elements_surface

Keyboard Command Sequence:

```
set_attach_divisions <divisions>
```
Command: `set_attach_limit`

**Description:** This command toggles the attach limit mode. If this mode is activated, attaching of nodes, edges and faces will be limited to those items that fall entirely within the distance limit specified by the `set_attach_distance` command (DISTANCE button).

Also see: `set_attach_distance`
- `attach_nodes_point`
- `attach_edges_curve`
- `attach_faces_surface`
- `attach_elements_curve`
- `attach_elements_surface`

**Keyboard Command Sequence:**
```
set_attach_limit <on or off>
```

Command: `set_attach_tolerance`

**Description:** This command sets the tolerance used by the attach commands for determining node locations when moving nodes to curves and surfaces. The smaller the specified tolerance the more accurately the nodes will be attached.

Also see: `attach_edges_curve`
- `attach_faces_surface`
- `attach_elements_curve`
- `attach_elements_surface`

**Keyboard Command Sequence:**
```
set_attach_tolerance <tolerance>
```
Command: set_auto_arrow
Description: This command toggles the automatic scaling of preprocessing arrows. By default, the mode is ON making the arrow length a fraction of the overall model size. This fraction may be changed from the default value of 0.1 with the auto_arrow_length command. When the mode is OFF, this fraction is ignored and the arrow length is then an absolute value, independent of the model size. This length may also be changed from its default value of 0.2 with the arrow_length command.

Also see: auto_arrow_length and arrow_length.

Keyboard Command Sequence:

set_auto_arrow <on or off>

Another Button with the same description:
MANUAL

Command: set_automag
Description: This command toggles the automatic deformation scaling for results viewing. By default, the mode is OFF and the magnification is 1.0 (actual displacements). When the mode is ON, the user specified magnification is ignored and the deformation scaling is set such that displacements are easily viewed on the screen.

Keyboard Command Sequence:

set_automag <on or off>

Another Button with the same description:
MANUAL
Command: set_autorange
Description: This command toggles automatic plotting range for contour plots. The default is on which means that the global maximum and minimum will be used. When the option is off, the user specified scale will be used, which is set by using the SET LIMITS button (range command). By using the COPY LIMITS button (copy_autorange command), the range will be set to the current increment’s min and max values.
Also see help: copy_autorange and range.

Keyboard Command Sequence:
set_autorange <on or off>

Another Button with the same description:
MANUAL

Command: set_autovec
Description: This command toggles automatic vector scaling during post plotting. When on, the vectors are scaled to be 5 percent of the size of the model. When off, the vectors are scaled by the magnification factor set by the set_vecmag command.

Keyboard Command Sequence:
set_autovec <on or off>

Another Button with the same description:
MANUAL

Command: set_break_crvs
Description: This command toggles the break_curves parameter which tells that the command clean_surface_loops will break curves or not.

Keyboard Command Sequence:
set_break_crvs <on or off>
**Command:**  set_buffered

**Description:** This command toggles graphics double-buffering.

- When double-buffering is ON, graphical changes to the program are made to an out-of-sight area, and then copied to the screen upon completion. Thus, only finished images are seen.

- When double-buffering is OFF, all graphical changes to the program are seen as they occur.

Double-buffering makes the program look smoother, but on some systems, it makes the program run slower, requires more memory, or displays its images with too much halftoning, which degrades image quality. By default, double-buffering is ON.

**Keyboard Command Sequence:**

```
set_buffered <on or off>
```

Another Button with the same description:

DIRECT

---

**Command:**  cavity_option

**Description:** This command toggles the dimension of cavity between two-dimensional and three-dimensional. It is located under the MESH GENERATION->CAVITIES (DIMENSION) menu.

**Keyboard Command Sequence:**

```
cavity_option <dimension> <2D or 3D>
```

Another Button with the same description:

3-D
Command: `set_change_class`

Description: This command sets the element class to which elements will be changed by the `change_class` command. Valid classes, grouped by family are as follows:

- **Lines:** line2, line3
- **Triangles:** tria3, tria6
- **Quads:** quad4, quad6, quad8, quad9
- **Tetras:** tetra4, tetra10
- **Hexahedrals:** hex8, hex12, hex20, hex27
- **Pentas:** penta6, penta15

Keyboard Command Sequence:

```
set_change_class <class>
```

Other Buttons with the same description:

- HEX (12)
- HEX (20)
- HEX (27)
- HEX (8)
- LINE (2)
- LINE (3)
- PENTA (15)
- PENTA (6)
- QUAD (6)
- QUAD (8)
- QUAD (9)
- TETRA (10)
- TETRA (4)
- TRIA (3)
- TRIA (6)

Command: `set_change_mode`

Description: This command specifies a special mode of operation for the `change Elements Class` command:

- When changing quad4 elements into tria3 elements, only two tria3 elements will be generated for every quad4 rather than the normal four. The converse operation changes one tria3 element into three quad4 elements, instead of into a single collapsed element.
– When changing tetra4 and penta6 elements into hex8 elements, the tetra4 elements are replaced by four hex8 elements and the penta6 elements are replaced by three hex8 elements, instead of by collapsed hex8 elements.

**Keyboard Command Sequence:**
```
set_change_mode <on or off>
```

**BIAS FACTORS**

**Command:** `set_convert_bias_factors`

**Description:** This command sets the bias factors in the U and V- directions used by convert commands. Bias factors are numbers between -1 and 1. Positive bias factors skew new entity creation locations in the positive direction, i.e. more entities are created in the positive direction. Negative bias factors skew new entity creation locations in the negative direction. Biasing is parabolic. Zero factors produce evenly distributed entities.

Singularities may occur if the magnitude of the bias factor is greater than one third.

**Keyboard Command Sequence:**
```
set_convert_bias_factors <U-direction factor> <V-direction factor>
```

**DIVISIONS**

**Command:** `set_convert_divisions`

**Description:** This command sets the number of divisions used by convert commands, i.e. the number of entities created from each specified entity.

**Keyboard Command Sequence:**
```
set_convert_divisions <number in U-direction> <number in V-direction>
```
**Command:** set_csect_control_node

**Description:** This command sets the control node of the current cross-section. The control node has one degree of freedom, which is seen as the shortening of the cross-section elements in the normal direction of the cross-section. Alternatively, the option generate_csect_control_node can be used to create a control node and assign it to the current cross-section.

**Keyboard Command Sequence:**

```
set_csect_control_node <node>
```

---

**Command:** set_curve_direction

**Description:** This command toggles the drawing of an arrowhead on each curve, which points in the direction the curve is defined in. Thus, for a given curve, the arrowhead points in the direction its curve is traversed when that curve is evaluated in an increasing direction in parametric space.

**Keyboard Command Sequence:**

```
set_curve_direction <on or off>
```

---

**Command:** set_curve_divisions

**Description:** This command sets the number of subdivisions to be used when plotting curves.

**Keyboard Command Sequence:**

```
set_curve_divisions <divisions>
```
Command: **set_curve_labels**

Description: This command toggles the labeling of curves with their ids.

Keyboard Command Sequence:

```
set_curve_labels <on or off>
```
FILLET

A fillet curve (radius) between two curves. Input is the two curves and the fillet radius. The curves are shortened or extended in order to create the fillet. Note that the orientation of the curves is important, the fillet starts at the head of the first curve and continues at the tail of the second curve. Thus, it may be necessary to flip the orientation of the curves.

SAMPLED

A straight line that connects a sequence of points. It is useful for creating curves from digitized data. Input is the origin point of the curve and the first point (to establish the direction of the curve) and then all the points that define the curve (including selecting the origin point and the first point again).

Arcs:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTER/RADIUS/ANGLE/ANGLE</td>
<td>An arc by giving the center, radius, and starting/ending angles.</td>
</tr>
<tr>
<td>CENTER/POINT/POINT</td>
<td>An arc by giving the center, starting point, and point defining the angle.</td>
</tr>
<tr>
<td>CENTER/POINT/ANGLE</td>
<td>An arc by giving the center, starting point, and angle.</td>
</tr>
<tr>
<td>POINT/POINT/POINT</td>
<td>An arc by giving the starting, ending, and intermediate points.</td>
</tr>
<tr>
<td>TANGENT/RADIUS/ANGLE</td>
<td>An arc by giving the ending point of another curve, the radius, and the angle.</td>
</tr>
</tbody>
</table>

Circles:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTER/RADIUS</td>
<td>A circle by giving the center and the radius.</td>
</tr>
<tr>
<td>CENTER/POINT</td>
<td>A circle by giving the center and a point on the periphery.</td>
</tr>
</tbody>
</table>
Other Buttons with the same description:

BEZIER  
COMPOSITE  
CUBIC SPLINE  
FILLET  
INTERPOLATE  
NURB  
POLY LINE  
SAMPLED  
TANGENT

**CENTER/POINT**

Command:  `set_curve_type circle_cp`

Curve type:  Circle: CENTER/POINT

Description:  A circle defined by the center and a point on the periphery.

The coordinates are given in the currently active local coordinate system.

Keyboard Command Sequence:

```
set_curve_type <type>
```

**CENTER/POINT/ANGLE**

Command:  `set_curve_type arc_cpa`

Curve type:  Arc: CENTER/POINT/ANGLE

Description:  An arc defined by its center, starting point, and swept angle.

The arc starts at the point and sweeps through the given angle in counterclockwise direction (in the current coordinate system).

The coordinates are given in the currently active local coordinate system.

Keyboard Command Sequence:

```
set_curve_type arc_cpa
```
### CENTER/POINT/POINT

**Command:** `set_curve_type arc_cpp`  
**Curve type:** Arc: CENTER/POINT/POINT  
**Description:** An arc defined by its center, and starting/ending points by giving their coordinates.  
The arc starts at the first point and the angle is defined by the second point. The arc sweeps counterclockwise.  
The coordinates are given in the currently active local coordinate system.  

**Keyboard Command Sequence:**  
```
set_curve_type arc_cpp
```  

### CENTER/RADIUS

**Command:** `set_curve_type arc_cr`  
**Curve type:** Circle: CENTER/RADIUS  
**Description:** A circle defined by the center and radius.  
The coordinates are given in the currently active local coordinate system.  

**Keyboard Command Sequence:**  
```
set_curve_type circle_cr
```  

### CENTER/RADIUS/ANGLE/ANGLE

**Command:** `set_curve_type arc_craa`  
**Curve type:** Arc: CENTER/RADIUS/ANGLE/ANGLE  
**Description:** An arc defined by its center, radius, and starting/ending angles.  
The coordinates are given in the currently active local coordinate system.  
The angles are given counterclockwise starting from the local x-axis.  

**Keyboard Command Sequence:**  
```
set_curve_type arc_craa
```
Command: `set_curve_type arc_ppp`
Curve type: Arc: POINT/POINT/POINT
Description: An arc defined by an initial point, ending point, and an intermediate point. The arc will pass through all three points. The coordinates of the points are given in the currently active local coordinate system.

Keyboard Command Sequence:

```
set_curve_type arc_ppp
```

---

Command: `set_curve_type arc_tra`
Curve type: Arc: TANGENT/RADIUS/ANGLE
Description: An arc defined by a tangent, a radius, and a swept angle. Input for the tangent is the end point of a previously defined curve for which the tangent is used. The arc will start tangentially to the other curve and sweep through the given angle. A positive angle will give a counterclockwise direction of the arc, in the currently active local system.

Keyboard Command Sequence:

```
set_curve_type arc_tra
```
BEZIER

Command: `set_curve_type bezier`
Curve type: BEZIER
Description: A Bezier curve defined by a list of control points.

Note that this type of curve generally does not pass through all the control points.

Keyboard Command Sequence:
```
set_curve_type bezier
```

COMPOSITE

Command: `set_curve_type composite`
Curve type: COMPOSITE
Description: A curve that combines a list of curves into one single curve.

The curves to join must lie end-to-end but can be input in any order. The old curves are removed and replaced by the composite curve.

Keyboard Command Sequence:
```
set_curve_type composite
```
**FILLET**

**Command:** set_curve_type fillet  
**Curve type:** FILLET  
**Description:** A fillet curve (radius) between two existing curves.  
Input is the two curves and the fillet radius.  
The curves are shortened or extended in order to create the fillet. Note that the orientation of the curves is important, the fillet starts at the head of the first curve and continues at the tail of the second curve. Thus it may be necessary to flip the orientation of the curves, which can be done with the command *flip_curves <list of curves>* available in the menu system under MESH GENERATION->CHECK->FLIP CURVES.  
This will result in a number of new curves: the fillet itself and the new segments if the curves were extended. To obtain one single curve, use the COMPOSITE curve type.  
If the curves have points in common with other curves, new points are created.  

**Keyboard Command Sequence:**  
```
set_curve_type fillet
```

**INTERPOLATE**

**Command:** set_curve_type inter  
**Curve type:** INTERPOLATE  
**Description:** A cubic NURBS curve that interpolates a list of points.  
The curve type CUBIC SPLINE is similar, but uses a cubic spline to interpolate the points.  

**Keyboard Command Sequence:**  
```
set_curve_type interpolate
```
**LINE**

Command: `set_curve_type line`
Curve type: LINE
Description: A straight line between two points.
The line is defined by giving the end points.
Keyboard Command Sequence:

```
set_curve_type line
```

**NURB**

Command: `set_curve_type nurb`
Curve type: NURB
Description: A general NURBS curve.
The whole definition of the NURBS curve is given (NURBS – Nonuniform Rational B-Spline):
- the number of control points (npoints)
- the curve order (order)
- a list of control points (npoints)
- a list of homogeneous coordinates (npoints)
- a knot vector (npoints+order)

Keyboard Command Sequence:

```
set_curve_type nurb
```
POLY LINE

Command: set_curve_type polyline

Curve type: POLY LINE

Description: A piecewise linear curve.

The curve is defined by a list of points and consists of straight line segment between these points.

Keyboard Command Sequence:

set_curve_type polyline

SAMPLED

Command: set_curve_type sampled

Curve type: SAMPLED

Description: A straight line that connects a sequence of points.

It is useful for creating curves from digitized data.

Input is the origin point of the curve and the first point (to establish the direction of the curve) and then all the points that define the curve (including selecting the origin point and the first point again).

Keyboard Command Sequence:

set_curve_type tangent
### CUBIC SPLINE

**Command:** `set_curve_type spline`  
**Curve type:** CUBIC SPLINE  
**Description:** A cubic spline that interpolates a list of points. The curve type INTERPOLATE is similar, but uses a more general NURBS curve to interpolate the points.

**Keyboard Command Sequence:**  
```
set_curve_type spline
```

### TANGENT

**Command:** `set_curve_type tangent`  
**Curve type:** TANGENT  
**Description:** A straight line that is tangent to an existing curve. Input is an end point of an existing curve and the length of the tangent curve. If more than one curve has this point as an end point, the curve with the lowest number will be chosen.

**Keyboard Command Sequence:**  
```
set_curve_type tangent
```

### CURVES

**Command:** `set_curves`  
**Description:** This command toggles the drawing of curves.

**Keyboard Command Sequence:**  
```
set_curves <on or off>
```
**NORMAL**

**Command:** `set_cut_normal`

**Description:** This command specifies the normal of the cutting planes used for post plotting. The components of the vector specified need not be normalized.

**Keyboard Command Sequence:**

```
set_cut_normal <XYZ components>
```

---

**# PLANES**

**Command:** `set_cut_planes`

**Description:** This command specifies the number of cutting planes to be drawn.

**Keyboard Command Sequence:**

```
set_cut_planes <number of cutting planes>
```

---

**POINT**

**Command:** `set_cut_point`

**Description:** This command specifies the point through which the first cutting plane will pass when post plotting. You must specify the point’s coordinates.

**Keyboard Command Sequence:**

```
set_cut_point <XYZ coordinates>
```

---

**SPACING**

**Command:** `set_cut_spacing`

**Description:** This command specifies the distance between consecutive cutting planes when post plotting.

**Keyboard Command Sequence:**

```
set_cut_spacing <distance>
```
Command: **set_decomposition_type**

**Description:** This button chooses the type of domain decomposition algorithm.

- **metis_best**
  - Performs Metis element-based and node-based decomposition, and picks the better solution.

- **metis_element_based**
  - Performs Metis element-based decomposition.

- **metis_node_based**
  - Performs Metis node-based decomposition.

- **geometric**
  - Performs decomposition based on coordinates (previously `domains_decompose`); the algorithm tries to anneal domains.

- **simple**
  - Performs decomposition based on ids of elements (previously `domains_generate`). The command `renumber_elements_directed` may be used to renumber the elements to help control the grouping of elements within domains.

Domain decomposition is a special Marc feature that enables Marc analyses to be run in parallel on certain machines or networks. The user must divide the model into sets of elements called domains. The analysis is performed by iteratively solving systems of equations for each domain on separate CPUs and integrating the results when the solution converges.

For more information see the *MSC.Marc User’s Guide* or call MSC Customer Support.

**Keyboard Command Sequence:**

```
set_decomposition_type <metis_best/
    metis_element_based/
    metis_node_based/
    geometric/
    simple>
```

Other Buttons with the same description:

- **METIS ELEMENT BASED**
- **METIS NODE BASED**
- **GEOMETRIC**
- **SIMPLE**
Command: `set_defmag`

Description: This command specifies the magnification of deformations when post plotting. This value is used when in manual deformation mode only (see `set_automag`).

Keyboard Command Sequence:

```
set_defmag <magnification factor>
```

Command: `set_deformed`

Description: This command specifies the mode of display of the deformed model when post plotting. The `set_autmag` and `set_defmag` commands control the degree of exaggeration of the actual deformations. You must specify whether the mode should be deformed shape only (on), deformed and undeformed shape (both), or none (off).

Keyboard Command Sequence:

```
set_deformed <on, off, or both>
```

Other Buttons with the same description:

- `DEF ONLY`
- `DEF & ORIG`
Interruptible Drawing

Command:  

- set_draw_interrupt_on  
- set_draw_interrupt_off

Description:  These commands specify whether or not interruptible drawing is enabled. The advantage of having interruptible drawing disabled is that the model is always displayed completely drawn. However, interruptible drawing of the model makes certain Mentat features like dynamic model more user-responsive.

When interruptible drawing is ON, if the model is currently being drawn, the drawing may be interrupted at any time by new user input that would change the way the model is drawn. Thus, the user is not forced to wait for an unwanted draw of the model to complete before a more desirable draw of the model may begin.

Interruptible drawing makes it easier to correctly position a large model, using the dynamic model feature. The draw-time responsiveness may be adjusted with the draw_interrupt_time command.

By default, interruptible drawing is ON.

Also see: dynamic_model_on and draw_interrupt_time.

Keyboard Command Sequence:

- set_draw_interrupt_on  
- set_draw_interrupt_off

Point

Command:  

- set_duplicate_point

Description:  This command sets the point about which objects will be rotated and scaled during duplication. You must specify the coordinates of the point.

Keyboard Command Sequence:

- set_duplicate_point <XYZ coordinates>
REPETITIONS

Command: set_duplicate_repetitions
Description: This command sets the number of times objects will be duplicated.
Keyboard Command Sequence:
set_duplicate_repetitions <number of repetitions>

ROTATIONS

Command: set_duplicate_rotations
Description: This command sets the amount of object rotation about the X, Y, and Z-axes during duplication. Each repetition of the duplication will rotate objects about the X, Y, and Z-axes, in that order. You must specify the angles in degrees.
Keyboard Command Sequence:
set_duplicate_rotations <X-axis rotation> <Y-axis rotation> <Z-axis rotation>

SCALE FACTORS

Command: set_duplicate_scale_factors
Description: This command sets the scale factors in the X, Y, and Z-directions for duplication of objects.
Keyboard Command Sequence:
set_duplicate_scale_factors <X-direction factor> <Y-direction factor> <Z-direction factor>
Command: **set_duplicate_trans_from_to**

**Description:** This command sets the translation vector for duplication of objects as the displacement vector that arises when travelling from a location (FromX,FromY,FromZ) to another location (ToX,ToY,ToZ), as follows:

\[
\begin{align*}
TX &= ToX - FromX \\
TY &= ToY - FromY \\
TZ &= ToZ - FromZ
\end{align*}
\]

Both locations can be entered by clicking on grid points, nodes or points. This button is located in the MESH GENERATION->DUPLICATE menu.

**Keyboard Command Sequence:**

```
set_duplicate_trans_from_to
<FromX> <FromY> <FromZ>
<ToX>   <ToY>   <ToZ>
```

---

Command: **set_duplicate_translations**

**Description:** This command sets the translations in the X, Y, and Z-directions for duplication of objects.

**Keyboard Command Sequence:**

```
set_duplicate_translations <X-direction translation> <Y-direction translation> <Z-direction translation>
```
**Command:** set_edge_attach_labels

**Description:** This command toggles the inclusion of attach information in the element edge labels. If switched on, the label of an attached edge includes the curve number to which the edge is attached, separated by an @-sign from the edge number.

Also see: attach_edges_curve
           detach_edges

**Keyboard Command Sequence:**

```
set_edge_attach_labels <on or off>
```

**Command:** set_edge_labels

**Description:** This command toggles the display of element edge labels. Edge labels are of the format element:edge.

**Keyboard Command Sequence:**

```
set_edge_labels <on or off>
```

**Command:** set_edges

**Description:** This command toggles the display of element edges.

**Keyboard Command Sequence:**

```
set_edges <on or off>
```

**Command:** set_element_class

**Description:** This command sets the element creation class (see add_elements). Available classes are as follows:

- line2, line3
- tria3, tria6
quad4, quad6, quad8, quad9
    tetra4, tetra10
    penta6, penta15
    hex8, hex12, hex20, hex27

**Keyboard Command Sequence:**

    set_element_class <class>

Other Buttons with the same description:

    HEX (12)
    HEX (20)
    HEX (27)
    HEX (8)
    LINE (2)
    LINE (3)
    PENTA (15)
    PENTA (6)
    QUAD (6)
    QUAD (8)
    QUAD (9)
    TETRA (10)
    TETRA (4)
    TRIA (3)
    TRIA (6)

---

**Command:**  set_element_labels

**Description:**  This command toggles the drawing of element labels. Labeling is only available in wireframe mode (see elements_solid, elements_wireframe).

**Keyboard Command Sequence:**

    set_element_labels <on or off>

---

**Command:**  set_element_size

**Description:**  This command specifies the relative size at which elements should be displayed. If less than 100% is specified, elements will be displayed with a size less than their actual size. This feature allows you to shrink displayed elements so that other model features will be visible.

**Keyboard Command Sequence:**

    set_element_size <percent size>
Command: set_elements
Description: This command toggles the display of elements.
Keyboard Command Sequence:

```
set_elements <on or off>
```

Another Button with the same description:

ELEMENTS

Command: set_expand_axito3d_upd_load
Description: With this command, the user can specify the time or the increment number at which the 3-D job starts. The load table curves are shifted based on the user input value if UPDATE LOAD TABLES button is on.

After the table curve is shifted at the origin, the time value will be zero, the apply value is the one at the user specified time if the apply is in type of load, or it will be zero if the apply is in type of displacement.

Keyboard Command Sequence:

```
set_expand_axito3d_upd_load <on/off>
```

Other Buttons with the same description:

TIME
UPDATE LOAD TABLES

Command: set_expand_point
Description: This command sets the point about which objects will be rotated and scaled during expansion. You must specify the coordinates of the point.

Keyboard Command Sequence:

```
set_expand_point <XYZ coordinates>
```
REPETITIONS

Command: set_expand_repetitions
Description: This command sets the number of times objects will be expanded.
Keyboard Command Sequence:

set_expand_repetitions <number of repetitions>

ROTATIONS

Command: set_expand_rotations
Description: This command sets the amount of object rotation about the X, Y, and Z-axes during expansion. Each repetition of the expansion will rotate objects about the X, Y, and Z-axes, in that order. You must specify the rotations in degrees.
Keyboard Command Sequence:

set_expand_rotations <X-axis rotation> <Y-axis rotation> <Z-axis rotation>

SCALE FACTORS

Command: set_expand_scale_factors
Description: This command sets the scale factors in the X, Y, and Z-directions for expansion of objects.
Keyboard Command Sequence:

set_expand_scale_factors <X-direction factor> <Y-direction factor> <Z-direction factor>
**FROM / TO**

**Command:** `set_expand_trans_from_to`

**Description:** This command sets the translation vector for expansion of objects as the displacement vector that arises when travelling from a location (FromX,FromY,FromZ) to another location (ToX,ToY,ToZ), as follows:

\[
\begin{align*}
TX &= ToX - FromX \\
TY &= ToY - FromY \\
TZ &= ToZ - FromZ
\end{align*}
\]

Both locations can be entered by clicking on grid points, nodes or points. This button is located in the MESH GENERATION->EXPAND menu.

**Keyboard Command Sequence:**

```
set_expand_trans_from_to
<FromX> <FromY> <FromZ>
<ToX> <ToY> <ToZ>
```

**TRANSLATIONS**

**Command:** `set_expand_translations`

**Description:** This command sets the translations in the X, Y, and Z-directions for expansion of objects.

**Keyboard Command Sequence:**

```
set_expand_translations <X-direction translation> <Y-direction translation> <Z-direction translation>
```
Command: **set_face_attach_labels**

**Description:** This command toggles the inclusion of attach information in the element face labels. If switched on, the label of an attached face includes the surface number to which the face is attached, separated by an @-sign from the face number.

Also see: attach_faces_surface
detach_faces

**Keyboard Command Sequence:**

```
set_face_attach_labels <on or off>
```

Command: **set_face_labels**

**Description:** This command toggles the display of element face labels. Face labels are of the format `element:face`.

**Keyboard Command Sequence:**

```
set_face_labels <on or off>
```

Command: **set_faces**

**Description:** This command toggles the display of element faces.

**Keyboard Command Sequence:**

```
set_faces <on or off>
```

Command: **set_font**

**Description:** This command selects the text font the program will use. This font is used for drawing text in the dialogue, menu, status, help, and graphics areas. To reset the font to the default, use the `reset_font` command.

**Keyboard Command Sequence:**

```
set_font <font name>
```
**GRID**

Command: **set_grid**

Description: This command toggles the drawing of a grid representing the local coordinate system. Grids enable the specification of coordinates by picking gridpoints.

Keyboard Command Sequence:

```
set_grid <on or off>
```

**set_grid_size**

Command: **set_grid_size**

Description: This command sets the size of the grid, in user local coordinates u and v. The grid will range from \(-u\_size\) to \(+u\_size\) in the U-direction, and from \(-v\_size\) to \(+v\_size\) in the V-direction, where \(u\_size\) and \(v\_size\) are the sizes you specify.

Keyboard Command Sequence:

```
set_grid_size <u size> <v size>
```

**set_grid_spacing**

Command: **set_grid_spacing**

Description: This command sets the spacing between gridpoints. The specified values will be the spacings in the U and V-directions.

Keyboard Command Sequence:

```
set_grid_spacing <u spacing> <v spacing>
```
**SHOW IDS**

**Command:**  set_history_increment_id

**Description:** This command sets step size between history point labels to the specified value. If the step size is 0, no history point labels will be displayed. If step size is 1, all labels will be displayed. If step size is 3, every third label will be displayed, etc.

**Keyboard Command Sequence:**

```
set_history_increment_id <step size>
```

**SET NODES**

**set_history_nodes**

**Command:**  set_history_nodes

**Description:** This command specifies a list of nodes for use in history plots. You must specify a list of nodes.

**Keyboard Command Sequence:**

```
set_history_nodes <node list> #
```

**set_history_sort**

**Command:**  set_history_sort

**Description:** This command instructs the program whether or not it should sort the curves in your history plot. Sorting is useful in filled mode, so that the curves will be arranged such that larger curves do not obscure smaller curves. By default, history curve sorting is ON.

**Keyboard Command Sequence:**

```
set_history_sort <on or off>
```
**XMAX**

**Command:**  `set_history_xmax`  
**Description:** This command specifies the maximum X value to be shown in the history plot. By default, this value is one.  
**Keyboard Command Sequence:**  
```
set_history_xmax <max x>
```

**XMIN**

**Command:**  `set_history_xmin`  
**Description:** This command specifies the minimum X value to be shown in the history plot. By default, this value is zero.  
**Keyboard Command Sequence:**  
```
set_history_xmin <min x>
```

**XSTEP**

**Command:**  `set_history_xstep`  
**Description:** This command specifies the number of steps in X to be used in the history plot. By default, the number of steps is ten.  
**Keyboard Command Sequence:**  
```
set_history_xstep <number of steps>
```

**YMAX**

**Command:**  `set_history_ymax`  
**Description:** This command specifies the maximum Y value to be shown in the history plot. By default, this value is one.  
**Keyboard Command Sequence:**  
```
set_history_ymax <max y>
```
Command: set_history_ymin
Description: This command specifies the minimum Y value to be shown in the history plot. By default, this value is zero.

Keyboard Command Sequence:
set_history_ymin <min y>

Command: set_history_ystep
Description: This command specifies the number of steps in Y to be used in the history plot. By default, the number of steps is ten.

Keyboard Command Sequence:
set_history_ystep <number of steps>

Command: set_hue
Description: This command sets a range of hues between the colors of two color indices. You must specify the starting index, the ending index, and the hue value to be applied to the range. Hue is specified as a number between zero and one.

Keyboard Command Sequence:
set_hue <start index> <end index> <hue value>
**ENTERED VALUES**

**Command:**  `set_icond_entered`

**Description:** This command specifies that values for the current initial condition are being entered by the user (as opposed to being provided through a user subroutine; see `set_icond_usersub`).

**Keyboard Command Sequence:**

```
set_icond_entered
```

**INIT CONDS**

**Command:**  `set_icond_labels`

**Description:** This command toggles the display of initial condition labels.

**Keyboard Command Sequence:**

```
set_icond_labels <on or off>
```

**POST FILE**

**Command:**  `set_icond_post_tape`

**Description:** This command specifies that values for the current state variable initial condition are determined at run time by reading a previously generated post file.

**Keyboard Command Sequence:**

```
set_icond_post_tape
```

**USER SUB.**

**Command:**  `set_icond_usersub`

**Description:** This command specifies that MSC.Marc should call the appropriate user subroutine for this initial condition during the analysis. If the user subroutine has been invoked, there is no need to enter values for the application since they are ignored by MSC.Marc in lieu of the values provided by the subroutine.
If the user subroutine is used, the FORTRAN source code file must exist in your current working directory, so that it is compiled and linked in with MSC.Marc when you submit the job (see job_usersub_file).

**Keyboard Command Sequence:**

```
set_icond_usersub <on or off>
```

**Command:** `set_iconds`

**Description:** This command toggles the display of initial conditions.

**Keyboard Command Sequence:**

```
set_iconds <on or off>
```

**Command:** `set_image_ps_color`

**Description:** This command sets whether or not color should be used in subsequent PostScript plotting of image files. The alternative is gray scale.

Also see: image_save, render, and image_ps_print.

**Keyboard Command Sequence:**

```
set_image_ps_color <on/off>
```

Another Button with the same description:

```
GRAY
```

**Command:** `set_import_report`

**Description:** Selects the import report file.

**Keyboard Command Sequence:**

```
set_import_report <report file name>
```
INFO

Command:  **set_info_labels**
Description:  This command toggles the display of information labels.
Keyboard Command Sequence:

```
set_info_labels <on or off>
```

CREATE INSERTS

Command:  **set_insert_create**
Description:  This command toggles the creation of inserts during rebar meshing.

It is located in the MESH GENERATION->AUTOMESH->2D REBAR MESHING menu.

Keyboard Command Sequence:

```
set_insert_create <on or off>
```

ELEMENTS

Command:  **set_insert_embedded_type**
Description:  This command toggles the embedded entity type between elements and nodes.

It is located in the LINKS->INSERTS menu.

Keyboard Command Sequence:

```
set_insert_embedded_type <on or off>
```

Another Button with the same description:

```
NODES
```
Mental Help Commands in S

**TOLERANCE**

**Command:** `set_insert_tolerance`

**Description:** This command sets the Exterior Tolerance for current insert. A node is considered within a host element if the distance between the element and the node is smaller than the tolerance times average edge length of the element, unless the node is actually inside another host element.

It is located in the LINKS->INSERTS menu.

**Keyboard Command Sequence:**

```
set_insert_tolerance <value>
```

**TOLERANCE**

**Command:** `set_intersect_tol`

**Description:** This command sets the tolerance value for curve intersection checking.

**Keyboard Command Sequence:**

```
set_intersect_tol
```

**1**

**Command:** `set_light`

**Description:** This command turns a light on or off in a view. You must specify the view and the number of the light to turn on or off. There are eight lights, numbered 1 through 8. Lighting must be ON in the view in order to see the effect.

Also see: `set_lighting`.

**Keyboard Command Sequence:**

```
set_light <view> <light> <on or off>
```

Other Buttons with the same description:

- 2 through 8
- LIGHT OFF
- LIGHT ON
Command: `set_light_spotlight`

Description: This command sets whether or not a light is a spotlight in a view. Also, the light must be local for the spotlight characteristic to be relevant. You must specify the view and the number of the light. There are eight lights, numbered 1 through 8. Lighting and the given light must be ON, and it must be local in a view in order to see the effect.

Also see: `set_lighting`, `set_light`, `light_local`, `light_spot_direction`, and `light_spot_angles`.

Keyboard Command Sequence:

```
set_light_spotlight <view> <light> <on or off>
```

Another Button with the same description:

SPOTLIGHT ON

Command: `set_lighting`

Description: This command toggles the lighting feature in a view giving the model a more realistic, three-dimensional appearance. When turned ON, the default light source is from infinity, on the positive Z-axis, in view space. Currently, lighting is implemented graphically only in the OpenGL version of Mentat.

Also see: `set_light`.

Keyboard Command Sequence:

```
set_lighting <view> <on or off>
```

Other Buttons with the same description:

LIGHTING ON
OFF
ON
Command: **set_lighting_attenuation**

**Description:** This command toggles lighting attenuation in a view. Lighting attenuation is relevant when lighting is on, and it affects only the local lights in a view.

When attenuation is OFF, the strength of a local light is independent of its distance from any geometry it is illuminating.

When attenuation is ON, the strength of a local light may diminish as its distance from geometry increases.

Also see: `set_lighting`, `lighting_attenuation`, and `light_local`.

**Keyboard Command Sequence:**
```
set_lighting_attenuation <view> <on or off>
```

Another Button with the same description:

ATTENUATION ON

---

Command: **set_lightness**

**Description:** This command changes the intensity of a range of colors between two color indices. You must specify the starting index, the ending index, and the lightness value to be applied to the range. Lightness is specified as a number between zero and one.

**Keyboard Command Sequence:**
```
set_lightness <start index> <end index> <lightness value>
```
Command:  set_link_labels
Description:  This command toggles the drawing of link labels.
Keyboard Command Sequence:

   set_link_labels <on or off>

Command:  spring_option
spring_multi_option
Description:  If set to ON, the spring and damper force are always zeroed out and the
spring only acts as a link. This can be used for both linear and nonlinear
springs. In the latter case, the stabilizer flag can only be set when the
stiffness option is used for the spring.
Keyboard Command Sequence:

   spring_option stabilizer:on | off
   spring_multi_option stabilizer:on | off
**PROPERTIES**

Command:  
\[
\begin{align*}
\text{spring\_option} \\
\text{spring\_multi\_option}
\end{align*}
\]

Description:  
This option opens a sub-window to set the electrical properties of the spring. Either linear or nonlinear springs can be defined through these commands.

**Linear Spring:**
A linear spring with electrical conductivity \(E\) is specified through the following options – the other options can be ignored:

<table>
<thead>
<tr>
<th>CONDUCTIVITY</th>
<th>SET: E</th>
</tr>
</thead>
</table>

**Nonlinear Spring:**
A nonlinear spring can be specified through one of two options – the conductivity option or the current option:

<table>
<thead>
<tr>
<th>CONDUCTIVITY</th>
<th>SET: E</th>
<th>TABLE: table defined as function of time, inc. no., voltage, temp. in joule heating</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CURRENT</th>
<th>SET: C</th>
<th>TABLE: table defined as function of time, inc. no., voltage, temp. in joule heating</th>
</tr>
</thead>
</table>

**Note:** If the \text{CURRENT} option is used, use of a table with voltage as one of the independent variables is mandatory so that the table gradient can be internally calculated.

**Keyboard Command Sequence:**

\[
\begin{align*}
\text{spring\_option} & \quad \text{electrical\_type:conductivity} \mid \text{current} \\
\text{spring\_multi\_option}
\end{align*}
\]
**PROPERTIES**

**Command:**  
- `spring_option`
- `spring_multi_option`

**Description:**  
This option opens a sub-window to set the mechanical properties of the spring. Either linear or nonlinear springs and dashpots can be defined through these commands.

**NUMERICAL STABILIZER:**  
If set to ON, the spring and damper force are always zero and the spring only acts as a link. This can be used for both linear and nonlinear springs, with the stiffness option used to define the nonlinear spring.

**Linear Spring:**  
A linear spring with stiffness \( K \) and initial force \( F_0 \) is specified through the following options – the other options can be ignored:

<table>
<thead>
<tr>
<th>Stiffness</th>
<th>Set: ( K )</th>
<th>Initial Force: ( F_0 )</th>
</tr>
</thead>
</table>

**Nonlinear Spring:**  
A nonlinear spring can be specified through one of two options – the stiffness option or the force option:

<table>
<thead>
<tr>
<th>Stiffness</th>
<th>Set: ( K )</th>
<th>Table: table defined as function of time, inc. no., displacement, temp. in coupled analysis</th>
</tr>
</thead>
</table>

**OR**

<table>
<thead>
<tr>
<th>Force</th>
<th>Set: ( F )</th>
<th>Table: table defined as function of time, inc. no., displacement, temp. in coupled analysis</th>
</tr>
</thead>
</table>

**Note:**  
If the \( F \) option is used, use of a table with displacement as one of the independent variables is mandatory so that the table gradient can be internally calculated.

**Linear Dashpot:**  
A linear dashpot with damping \( C \) is specified through the following options – the other options can be ignored:

<table>
<thead>
<tr>
<th>Damping Coefficient</th>
<th>Set: ( C )</th>
</tr>
</thead>
</table>
**Nonlinear Dashpot:**
A nonlinear dashpot can be specified through one of two options – the coefficient option or the force option:

<table>
<thead>
<tr>
<th>DAMPING COEFFICIENT</th>
<th>SET: C</th>
<th>TABLE: table defined as function of time, inc. no., velocity, temp. in coupled analysis</th>
</tr>
</thead>
</table>

OR

<table>
<thead>
<tr>
<th>FORCE</th>
<th>SET: C</th>
<th>TABLE: table defined as function of time, inc. no., velocity, temp. in coupled analysis</th>
</tr>
</thead>
</table>

**Note:** If the FORCE option is used, use of a table with velocity as one of the independent variables is mandatory so that the table gradient can be internally calculated.

**Keyboard Command Sequence:**

```
spring_option       static_type:stiffness  |  force
spring_multi_option dynamic_type:damping  |  force
```
**PROPERTIES**

**Command:**
- `spring_option`
- `spring_multi_option`

**Description:**
This option opens a sub-window to set the thermal properties of the spring. Either linear or nonlinear springs can be defined through these commands.

**Linear Spring:**
A linear spring with heat transfer coefficient $H$ is specified through the following options – the other options can be ignored:

```
HEAT TRANSFER COEFFICIENT
SET: H
```

**Nonlinear Spring:**
A nonlinear spring can be specified through one of two options – the coefficient option or the flux option:

```
HEAT TRANSFER COEFFICIENT
SET: H
TABLE: table defined as function of time, inc. no., temp.
```

OR

```
FLUX
SET: Q
TABLE: table defined as function of time, inc. no., temp.
```

**Note:** If the FLUX option is used, use of a table with temperature as one of the independent variables is mandatory so that the table gradient can be internally calculated.

**Keyboard Command Sequence:**
```
spring_option thermal_type:heat | flux
spring_multi_option
```
Command: set_link_spring_type
link_multi_spring_type

Description: These commands set the type of the current spring or the type of the springs that will be created by subsequent calls to the link_multi_spring_n_to_1 and link_multi_spring_n_to_n commands. These springs can be used for mechanical, thermal, or electrical (joule) analysis. Three types are available:

1. Fixed DOF
2. To Ground
3. True Direction - only available for mechanical springs

WARNING: For coupled runs with multiple analysis passes, it is highly recommended to define mechanical, thermal, and electrical springs individually to avoid possible discrepancies in the DOF or spring types between the passes.

FIXED DOF
This is a spring that acts on selected degrees of freedom of the begin and end nodes of the spring. The force $F$ in the spring is defined as

$$ F = K(ue-ub) + C(ve-vb) + F_0 $$

where $ub$ and $ue$ are the displacement (or rotation) values of the degrees of freedom of the begin and end nodes, $vb$ and $ve$ the corresponding velocities, $K$ and $C$ the stiffness and damping coefficient of the spring and $F_0$ the initial force.

TO GROUND
This is a spring between a selected degree of freedom of a node and the fixed ground. The force $F$ in the spring is defined as

$$ F = K*ub + C*vb + F_0 $$

TRUE DIRECTION
This is a true spring between the begin and end nodes. The spring acts on all displacement degrees of freedom of the nodes and the force points in the direction of the relative position of the end node with respect to the begin node of the spring. The force $F$ in the spring is defined as

$$ F = K|Ue-Ub| + C|Ve-Vb| + F_0 $$
where $U_b$ and $U_e$ are displacement vectors of begin and end nodes, $V_e$ and $V_b$ the velocities of the nodes and where $|A|$ is the length of the vector $A$.

Note that the definitions given above are valid only if user subroutine USPRNG is not being used. The latter allows general nonlinear springs to be defined, in which the spring force depends in one of the three following nonlinear ways on the degrees of freedom of the begin and end nodes of the springs:

- $F = F(ue-ub, ve-vb) + F_0$, for fixed dof springs
- $F = F(ub, vb) + F_0$, for springs to ground
- $F = F(|U_e-U_b|, |V_e-V_b|) + F_0$, for true direction springs

**Keyboard Command Sequence:**

```
set_link_spring_type   fixed_dof|to_ground|true_dir
link_multi_spring_type fixed_dof|to_ground|true_dir
```

**Other Buttons with the same description:**

```
TO GROUND
TRUE DIRECTION
```

**Command:** `set_link_usersub`

**Description:** This command enables and disables the use of the user subroutine USPRNG in Marc jobs involving links.

If the user subroutine is used, the FORTRAN source code file must exist in your current working directory, so that it can be compiled and linked in with Marc when you submit the job.

**Keyboard Command Sequence:**

```
set_link_usersub <on or off>
```
**LINKS**

Command:  **set_links**  
Description:  This command toggles the display of links.  
Keyboard Command Sequence:  

```
set_links <yes or no>
```

**GROUND BODY**

Command:  **set_loadcase_ground_body**  
Description:  This command sets the ground body for steady state rolling. Only rigid body can be selected as ground body.  
Keyboard Command Sequence:  

```
set_loadcase_ground_body <cbody>
```

**LOADCASES**

Command:  **set_loadcase_labels**  
Description:  This command toggles the drawing of loadcase labels. When ON, labels of boundary conditions belonging to this loadcase are displayed.  
Keyboard Command Sequence:  

```
set_loadcase_labels <on or off>
```
**SPINNING BODY**

Command: `set_loadcase_spinning_body`
Description: This command sets the spinning body for steady state rolling. Only deformable body can be selected as spinning body.

Keyboard Command Sequence:
```
set_loadcase_spinning_body <cbody>
```

**LOADCASES**

Command: `set_loadcases`
Description: This command toggles the display of loadcases. When ON, boundary conditions belonging to this loadcase are displayed.

Keyboard Command Sequence:
```
set_loadcases <yes or no>
```

**MATCH CURVES**

Command: `set_match_crvs`
Description: This command toggles the `match_curves` parameter which tells that the command `clean_surface_loops` will match neighboring curves or not.

Keyboard Command Sequence:
```
set_match_crvs <on or off>
```
MERGE RENUMBER

Command: `set_merge_renumber`
Description: This command controls the renumbering of model entities during a model merge operation. When a model is renumbered, all entities will be assigned new ids based upon their creation order. A merged model will receive higher ids than the existing model.

Keyboard Command Sequence:
```
set_merge_renumber <on or off>
```

MAX TOLERANCE

Command: `set_mesh_max_tol`
Description: This command sets the max. tolerance value. Max. tolerance is used for surface geometry clean. If distance between surface trimming curve points are shorter than max. tolerance, trimming curves are added to the surface.

Keyboard Command Sequence:
```
set_mesh_max_tol
```

MIN TOLERANCE

Command: `set_mesh_min_tol`
Description: This command sets the min. tolerance value. Min. tolerance is used as a tolerance for 2-D and 3-D geometry clean.

Keyboard Command Sequence:
```
set_mesh_min_tol
```
**SURF PARAM TOL**

**Command:** `set_mesh_param_tol`

**Description:** This command sets the surface parametric space tolerance \((\text{param tolerance})\) value. \(\text{Param tolerance}\) is used for surface geometry clean.

**Keyboard Command Sequence:**

```
set_mesh_param_tol
```

---

**FORMULAS**

**Command:** `set_move_formulas`

**Description:** This command specifies the formulas used by the move commands. The variables \(x, y, z\), are the coordinates of the node or point being moved. Also, the variable \(t\) is the polar coordinate of that node or point.

**Keyboard Command Sequence:**

```
set_move_formulas <formula x> <formula y> <formula z>
```

---

**POINT**

**Command:** `set_move_point`

**Description:** This command specifies the reference point used by the move commands. You must specify the coordinates of the point.

**Keyboard Command Sequence:**

```
set_move_point <XYZ coordinates>
```
**ROTATIONS**

**Command:** set_move_rotations

**Description:** This command specifies the amount of rotation about the X, Y, and Z-axes performed by move commands.

**Keyboard Command Sequence:**

```
set_move_rotations <X-axis rotation>
<Y-axis rotation> <Z-axis rotation>
```

**SCALE FACTORS**

**Command:** set_move_scale_factors

**Description:** This command sets the scale factors in the X, Y, and Z-directions used by the move commands.

**Keyboard Command Sequence:**

```
set_move_scale_factors <X-direction factor>
<Y-direction factor> <Z-direction factor>
```

**FROM / TO**

**Command:** set_move_trans_from_to

**Description:** This command sets the translation vector for moving of objects as the displacement vector that arises when travelling from a location (FromX,FromY,FromZ) to another location (ToX,ToY,ToZ), as follows:

- TX = ToX - FromX
- TY = ToY - FromY
- TZ = ToZ - FromZ

Both locations can be entered by clicking on grid points, nodes or points. This button is located in the MESH GENERATION->MOVE menu.

**Keyboard Command Sequence:**

```
set_move_trans_from_to <FromX> <FromY> <FromZ>
<ToX>   <ToY>   <ToZ>
```
**TRANSLATIONS**

Command: `set_move_translations`

Description: This command specifies the amount of translation in the X, Y, and Z-directions performed by move commands.

Keyboard Command Sequence:
```
set_move_translations <X-axis translation> <Y-axis translation> <Z-axis translation>
```

**ATTACH INFO**

Command: `set_node_attach_labels`

Description: This command toggles the inclusion of attach information in the element node labels. If switched on, the label of an attached node includes the point number to which the node is attached, separated by an @-sign from the node number.

Also see: `attach_nodes_point`
```
detach_nodes
```

Keyboard Command Sequence:
```
set_node_attach_labels <on or off>
```

**NODES**

Command: `set_node_labels`

Description: This command toggles the display of node ids.

Keyboard Command Sequence:
```
set_node_labels <yes or no>
```
**NODES**

**Command:**  
*set_nodes*

**Description:** This command toggles the display of nodes.

**Keyboard Command Sequence:**  
*set_nodes <yes or no>*

---

**XMAX**

**Command:**  
*set_opt_plot_xmax*

**Description:** This command specifies the maximum X value to be shown in the response gradient plot. By default, this value is one.

**Keyboard Command Sequence:**  
*set_opt_plot_xmax <max x>*

---

**XMIN**

**Command:**  
*set_opt_plot_xmin*

**Description:** This command specifies the minimum X value to be shown in the response gradient plot. By default, this value is zero.

**Keyboard Command Sequence:**  
*set_opt_plot_xmin <min x>*

---

**XSTEP**

**Command:**  
*set_opt_plot_xstep*

**Description:** This command specifies the number of steps in X to be used in the response gradient plot. By default, the number of steps is ten.

**Keyboard Command Sequence:**  
*set_opt_plot_xstep <number of steps>*
YMAX

Command: set_opt_plot_ymax
Description: This command specifies the maximum Y value to be shown in the response gradient plot. By default, this value is one.

Keyboard Command Sequence:
set_opt_plot_ymax <max y>

YMIN

Command: set_opt_plot_ymin
Description: This command specifies the minimum Y value to be shown in the response gradient plot. By default, this value is zero.

Keyboard Command Sequence:
set_opt_plot_ymin <min y>

YSTEP

Command: set_opt_plot_ystep
Description: This command specifies the number of steps in Y to be used in the response gradient plot. By default, the number of steps is ten.

Keyboard Command Sequence:
set_opt_plot_ystep <number of steps>
Command: **set_orient_angle**

Description: This command sets the orientation angle for the current orientation. The orientation angle is required for all orientation types. Note that this command is not allowed for orientation type **3d_aniso**. The command **orient_rotate** should be used instead.

Keyboard Command Sequence:

```
set_orient_angle <angle in degrees>
```

Command: **set_orient_labels**

Description: This command toggles the display of element orientation labels.

Keyboard Command Sequence:

```
set_orient_labels <on or off>
```

Command: **set_orient_vector1**

Description: This command sets the first user-defined vector for the current orientation. The first user-defined vector is required for orientation types **xu_plane**, **yu_plane**, **zu_plane**, **uu_plane**, and **3d_aniso**.

Keyboard Command Sequence:

```
set_orient_vector1 <X, Y, Z vector coordinates>
```
Command: **set_orient_vector2**

**Description:** This command sets the second user-defined vector for the current orientation. The second user-defined vector is required for orientation types `uu_plane` and `3d_aniso`.

**Keyboard Command Sequence:**

```
set_orient_vector2 <X, Y, Z vector coordinates>
```

Command: **set_orients**

**Description:** This command toggles the display of element orientations.

**Keyboard Command Sequence:**

```
set_orients <on or off>
```

Command: **set_overlay_bias_factors**

**Description:** This command sets the bias factors used by the `overlay_mesh` command. Bias factors are numbers between -1 and 1. Positive bias factors skew element creation locations in the positive direction, i.e. more entities are created in the positive direction. Negative bias factors skew element creation locations in the negative direction. Biasing is parabolic. Zero factors produce evenly distributed elements. Singularities may occur if the magnitude of the bias factor is greater than one third.

**Keyboard Command Sequence:**

```
set_overlay_bias_factors <U-direction factor> <V-direction factor>
```
Command:  set_overlay_divisions
Description:  This command sets the number of divisions used by the overlay_mesh command, i.e. the number of elements created in each direction.

Keyboard Command Sequence:

```
set_overlay_divisions <number in U-direction>  
<number in V-direction>
```

---

Command:  set_pathplot_node_id
Description:  This command sets step size between path plot node labels to the specified value.
If the step size is 0, no path plot node labels will be displayed.
If step size is 1, all labels will be displayed.
If step size is 3, every third label will be displayed, etc.

Keyboard Command Sequence:

```
set_pathplot_node_id <step size>
```

---

Command:  set_pathplot_path
Description:  This command specifies a path of nodes for use in path plots. You must specify a group of nodes in the order that you wish them to appear in the path plot.

Keyboard Command Sequence:

```
set_pathplot_path  <nodes>  #
```
set_pathplot_sort

Command: **set_pathplot_sort**
Description: This command instructs the program whether or not it should sort the curves in your path plot. Sorting is useful in filled mode, so that the curves are arranged such that larger curves do not obscure smaller curves. By default, path plot curve sorting is ON.

Keyboard Command Sequence:

```
set_pathplot_sort <on or off>
```

XMAX

Command: **set_pathplot_xmax**
Description: This command specifies the maximum X value to be shown in the path plot. By default, this value is one.

Keyboard Command Sequence:

```
set_pathplot_xmax <max x>
```

XMIN

Command: **set_pathplot_xmin**
Description: This command specifies the minimum X value to be shown in the path plot. By default, this value is zero.

Keyboard Command Sequence:

```
set_pathplot_xmin <min x>
```
XSTEP

Command: set_pathplot_xstep
Description: This command specifies the number of steps in X to be used in the path plot. By default, the number of steps is ten.

Keyboard Command Sequence:
set_pathplot_xstep <number of steps>

YMAX

Command: set_pathplot_ymax
Description: This command specifies the maximum Y value to be shown in the path plot. By default, this value is one.

Keyboard Command Sequence:
set_pathplot_ymax <max y>

YMIN

Command: set_pathplot_ymin
Description: This command specifies the minimum Y value to be shown in the path plot. By default, this value is zero.

Keyboard Command Sequence:
set_pathplot_ymin <min y>

YSTEP

Command: set_pathplot_ystep
Description: This command specifies the number of steps in Y to be used in the path plot. By default, the number of steps is ten.

Keyboard Command Sequence:
set_pathplot_ystep <number of steps>

PLANE TOL.
Command: set_plane_tol
Description: This command sets the angular tolerance used by the plane select method.
Keyboard Command Sequence:

    set_plane_tol <tolerance>

Command: set_point_labels
Description: This command toggles the display of point ids.
Keyboard Command Sequence:

    set_point_labels <on or off>

Command: set_points
Description: This command toggles the display of points.
Keyboard Command Sequence:

    set_points <on or off>

Command: set_post_delta
Description: This command toggles the difference results plotting feature, which when turned on plots the difference of the current increment with the previously plotted increment.
Keyboard Command Sequence:

    set_post_delta <on/off>
**POST PROCEDURE**

Command: `set_post_procedure`

Description: This command toggles the `post_procedure_file` feature, which when turned on will execute the contents of a procedure file at each increment.

Also see help: `post_procedure_file`

Keyboard Command Sequence:

```
set_post_procedure <on/off>
```

**USE NODAL TRANSFORMATIONS**

Command: `set_tform_nqs`

Description: This command toggles the transformation of nodal quantities during the post processing of MSC.Marc post files. By default, nodal quantities are displayed in the global coordinate system. If this option is on, the nodal transformations that exist on the post file are applied.

Keyboard Command Sequence:

```
set_tform_nqs <on/off>
```

**USER TENSORS**

Command: `set_post_user_tensors`

Description: This command toggles the plotting of user defined tensors. User tensors are available when all six components of a normal tensor are available as user defined post variables. When user tensors are turned on, the additional tensor scalars and vectors will be computed from the user defined tensor and be made available for plotting.

Keyboard Command Sequence:

```
set_post_user_tensors <on/off>
```
**set_proc_echo**

**Command:** `set_proc_echo`

**Description:** This command toggles procedure command echoing. When executing a procedure with procedure echo on, commands are echoed in the dialogue area as they are executed.

**Keyboard Command Sequence:**

```
set_proc_echo <on or off>
```

**COEF.**

**Command:** `set_rbe3_ret_coef_def`

**Description:** This command in the RBE3 menu sets the default coefficient for retained nodes.

**Keyboard Command Sequence:**

```
set_rbe3_ret_coef_def <coefficient>
```

**1**

**Command:** `set_rbe3_ret_dof_def`

**Description:** This command in the RBE3 menu sets the default dof for retained nodes.

**Keyboard Command Sequence:**

```
set_rbe3_ret_dof_def <dof_id>
```

Other buttons with the same description:

```
2, 3, 4, 5, and 6
```
**TOLERANCE**

Command:  
**set_relative_tol**

Description:  This tolerance parameter is used as a relative tolerance for geometry repair and meshing. The absolute relative value is computed by multiplying the relative tolerance value by the curve length.

Keyboard Command Sequence:  

```
set_relative_tol
```

---

**TOLERANCE**

Command:  
**set_relax_tolerance**

Description:  This command sets the convergence tolerance used by the relax_nodes command.

Keyboard Command Sequence:  

```
set_relax_tolerance <tolerance>
```

---

**ANTI-ALIASING**

Command:  
**set_render_antialias**

Description:  This command sets antialiasing on or off for subsequent rendered images. Antialiasing is a technique which attempts to minimize the jaggedness and other unwanted artifacts that sometimes appear in raster images. Images generally look better with antialiasing turned on, but they may take longer to create.

Also see: render, render_antialias_tol, and render_antialias_depth.

Keyboard Command Sequence:  

```
set_render_antialias <on/off>
```
**HIGHLIGHTS**

Command: **set_render_highlights**

Description: This command sets highlights on or off for subsequent rendered images. Specular highlights can appear on shiny materials, where they reflect a light source. This command sets whether or not such highlights are included in rendered images.

Also see: render, render_standard, and set_render_shadows.

**Keyboard Command Sequence:**

```
set_render_highlights <on/off>
```

**SHADOWS**

Command: **set_render_shadows**

Description: This command sets shadows on or off for subsequent rendered images. Shadows can appear wherever a material blocks a light source. This command sets whether or not shadows are included in rendered images. Images generally look better with shadows turned on, but they may take longer to create.

Also see: render, render_standard, and set_render_highlights.

**Keyboard Command Sequence:**

```
set_render_shadows <on/off>
```

**INCREMENT**

Command: **set_renumber_increment**

Description: This command specifies the amount by which each new id number is incremented when entities are renumbered.

**Keyboard Command Sequence:**

```
set_renumber_increment <increment>
```
Command: `set_renumber_start`
Description: This command sets the first id to be used by the renumber commands.
Keyboard Command Sequence:
```
set_renumber_start <initial id>
```
Other Buttons with the same description:
- ELEMS DIRECTED
- NODES
- START

Command: `set_rm_free_crvs`
Description: This command toggles the `rm_free_curves` parameter which tells that the command `clean_surface_loops` will remove free curves or not.
Keyboard Command Sequence:
```
set_rm_free_crvs <on or off>
```

Command: `set_saturation`
Description: This command changes the saturation of a range of colors between two color indices. You must specify the starting index, the ending index, and the saturation value to be applied to the range. Saturation is specified as a number between zero and one.
Keyboard Command Sequence:
```
set_saturation <start index> <end index> <saturation value>
```
**set_save_as_version**

**Command:**  set_save_as_version  
**Description:** This command specifies the format of the model file. It only affects the save_as_model (SAVE AS) command.

**Keyboard Command Sequence:**
```
set_save_as_version
<default|ment330|ment320|ment310|ment231>
```

**set_save_formatted**

**Command:**  set_save_formatted  
**Description:** This command sets the format of database files created with the save model commands. Formatted database files are ascii text files, and are editable and portable across machine architectures. Unformatted (binary) database files are not always architecture independent but are smaller and processed more quickly.

**Keyboard Command Sequence:**
```
set_save_formatted <on or off>
```

**SELECT DISTANCE**

**Command:**  set_select_distance  
**Description:** This command sets the select distance for selecting items that fall within a certain distance of a point, curve or surface by the select_method_point_dist, select_method_curve_dist, and select_method_surface_dist methods.

This button is located under the UTILS->SELECT->METHOD menu.

**Keyboard Command Sequence:**
```
set_select_distance
```
Command: **set_solid_edge_labels**

Description: This command toggles the labeling of solid edges with their ids.

Keyboard Command Sequence:

```
set_solid_edge_labels <on or off>
```

---

Command: **set_solid_edges**

Description: This command toggles the drawing of solid edges.

Keyboard Command Sequence:

```
set_solid_edges <on or off>
```

---

Command: **set_solid_face_labels**

Description: This command toggles the labeling of solid faces with their ids.

Keyboard Command Sequence:

```
set_solid_face_labels <on or off>
```

---

Command: **set_face_normal_dev**

Description: This command will set the maximum deviation allowed between 2 normals for adjacent facets generated for the displaying of the solid surface.

Keyboard Command Sequence:

```
set_solid_face_surface_dev <value>
```
**SURFACE DEV.**

**Command:** `set_face_surface_dev`

**Description:** This command will set the allowed value for offset distance from the facets generated for display to the actual solid surface.

**Keyboard Command Sequence:**

```
set_face_surface_dev <value>
```

**...FACES**

**Command:** `set_solid_faces`

**Description:** This command toggles the drawing of solid faces.

**Keyboard Command Sequence:**

```
set_solid_faces <on or off>
```

**SOLIDS**

**Command:** `set_solid_labels`

**Description:** This command toggles the labeling of solids with their ids.

**Keyboard Command Sequence:**

```
set_solid_labels <on or off>
```

**BLOCK**

**Command:** `set_solid_type`

**Description:** This command sets the solid create type to the specified type. Subsequent `add_solids` commands will create solids of this type. Available solid types are as follows:

- block
- sphere
- cylinder
- torus
- prism

**Keyboard Command Sequence:**

```
set_solid_type <type>
```
Other Buttons with the same description:

- CYLINDER
- PRISM
- SPHERE
- TORUS

---

**Command:** set_solid_vertex_labels

**Description:** This command toggles the labeling of solid vertices with their ids.

**Keyboard Command Sequence:**

```
set_solid_vertex_labels <on or off>
```

---

**Command:** set_solid_vertices

**Description:** This command toggles the drawing of solid vertices.

**Keyboard Command Sequence:**

```
set_solid_vertices <on or off>
```

---

**Command:** set_solids

**Description:** This command toggles the drawing of solids.

**Keyboard Command Sequence:**

```
set_solids <on or off>
```
**set_surface_divisions**

**Command:** `set_surface_divisions`

**Description:** This command sets the number of divisions in each direction to be used when drawing surfaces. The more number of divisions, the more realistic surfaces will appear.

**Keyboard Command Sequence:**

```
set_surface_divisions <number of divisions>
```

**SURFACES**

**Command:** `set_surface_labels`

**Description:** This command toggles the labeling of surfaces with their ids.

**Keyboard Command Sequence:**

```
set_surface_labels <on or off>
```

**LINES**

**Command:** `set_surface_lines`

**Description:** This command specifies that surfaces edges should be drawn when in `surfaces_solid` mode.

**Keyboard Command Sequence:**

```
set_surface_lines <on or off>
```
SURFACE TYPE

Menu: SURFACE TYPE

Description: This button toggles the type of surface to create. Subsequent add_surfaces commands will create surfaces of this type. Available surface types are as follows:

- QUAD: A bilinear surface defined by four corner points.
- BEZIER: A bezier surface defined by a list of control points.
- DRIVEN: A surface defined by a driven driving curve.
- NURB: A general NURBS surface given by a complete definition: the number of control points in each direction, the curve order in each direction, a list of control points, a list of homogeneous coordinates, and a knot vector.
- RULED: A linear surface between two curves.
- SPHERE: A sphere by giving the center and radius.
- CYLINDER: A cylindrical (or conical) surface.
- SWEPT: A surface defined by a swept curve, a sweeping curve, and the number of steps.
- INTERPOLATE: A cubic NURBS surface that interpolates \( npu \times npv \) control points.
- CONNS: Coons linear surface defined by four boundary curves.
- SKIN: Smooth curve passing through a list of curves.
- SAMPLED: A surface created by a sequence of points. It is useful for creating surfaces from digitized data. Input is the origin point of the surface, a point in the first direction (to establish the first direction of the surface), then a point in the second direction of the surface (to establish the second direction of the surface), and then all the points that will define the surface (including selecting the origin point and the points in the first and second directions again).
**BEZIER**

Command: set_surface_type bezier

Surface Type: BEZIER

Description: A Bezier surface defined by a list of control points.

Note that this type of surface does in general not pass through all the control points.

Input is the point numbers (not the coordinates).

Keyboard Command Sequence:

```
set_surface_type bezier
```

**COONS**

Command: set_surface_type coons

Surface Type: COONS

Description: A Coons (linear) surface is defined by four boundary curves.

It can be described as a combination of two ruled surfaces, defined by the opposite sides.

The four curves must be given in cyclic order around the boundary.

Note that the orientation of the curves is important. They must be placed head-to-tail. Thus, it may be necessary to flip some of the curves, which can be done with the command

```
*flip_curves <list of curves>
```

available in the menu system under

MESH GENERATION->CHECK->FLIP CURVES

Keyboard Command Sequence:

```
set_surface_type coons
```
**CYLINDER**

Command:  `set_surface_type cylinder`

Surface Type: CYLINDER

Description: A cylinder (or actually a cone) defined by two points on the axis and the radii at these points.

Input is the coordinates of the points on the axis (not point numbers).

**Keyboard Command Sequence:**

```
set_surface_type cylinder
```

---

**DRIVEN**

Command:  `set_surface_type driven`

Surface Type: DRIVEN

Description: A driven surface is defined by a driven and driving curve.

The surface is generated by moving the driven curve along the driving curve.

The driving curve can be positioned anywhere in space. Thus, a translation of the driving curve does not affect the generated surface.

A surface of revolution can be created by using a circle appropriately oriented as a driving curve.

**Keyboard Command Sequence:**

```
set_surface_type driven
```
**INTERPOLATE**

Command: `set_surface_type interpolate`

Surface Type: INTERPOLATE

Description: A cubic NURBS surface that interpolates a list of points.

Input is the number of points in the u- and v-direction and the points. The points (point numbers, not coordinates) are given column wise, i.e. first by looping over rows.

Keyboard Command Sequence:

```
set_surface_type interpolate
```

---

**NURB**

Command: `set_surface_type nurb`

Surface Type: NURB

Description: A general NURBS surface.

The whole definition of the NURBS surface is given (NURBS – Nonuniform Rational B-Spline):

- the number of control points in u-direction \((npu)\)
- the number of control points in v-direction \((npv)\)
- the curve order in u-direction \((ordu)\)
- the curve order in v-direction \((ordv)\)
- a list of control points \((npu*npv)\)
- a list of homogeneous coordinates \((npu*npv)\)
- a knot vector \((npu+ordu+npv+ordv)\)

Keyboard Command Sequence:

```
set_surface_type nurb
```
**QUAD**

**Command:** `set_surface_type quad`
**Surface Type:** QUAD
**Description:** A bilinear surface defined by four corner points.

Input is the point numbers (not the coordinates).

**Keyboard Command Sequence:**
```
set_surface_type quad
```

---

**RULED**

**Command:** `set_surface_type ruled`
**Surface Type:** RULED
**Description:** A ruled surface is defined by two curves. The surface can be described as generated by a straight line that is connected to the starting ends of the two curves and moved along them.

Note that the orientation of the defining curves is important. It may be necessary to flip one of the curves, which can be done with the command
```
*flip_curves <list of curves>
```
available in the menu system under

MESH GENERATION->CHECK->FLIP CURVES

**Keyboard Command Sequence:**
```
set_surface_type ruled
```
Command:  set_surface_type sampled

Surface Type:  SAMPLED

Description:  A surface created by a sequence of points.

It is useful for creating surfaces from digitized data. Input is the origin point of the surface, a point in the first direction (to establish the first direction of the surface), then a point in the second direction of the surface (to establish the second direction of the surface), and then all the points that will define the surface (including selecting the origin point and the points in the first and second directions again).

Keyboard Command Sequence:

    set_surface_type sampled

---

Command:  set_surface_type skin

Surface Type:  SKIN

Description:  A skinned surface is defined by a list of curves.

The surface passes through all curves producing a smooth surface. If only two curves are given, it gives a ruled surface.

Note that the orientation of the defining curves is important. It may be necessary to flip some of the curves, which can be done with the command

    *flip_curves <list of curves>

available in the menu system under

    MESH GENERATION->CHECK->FLIP CURVES

Keyboard Command Sequence:

    set_surface_type skin
Command:  set_surface_type sphere
Surface Type:  SPHERE
Description:  A sphere defined by the center and the radius.
             Input is the coordinates of the center point (not a point number).
Keyboard Command Sequence:

        set_surface_type sphere

Command:  set_surface_type swept
Surface Type:  SWEPT
Description:  A swept surface is defined by a swept curve, a sweeping curve, and the
             number of steps in the surface.
Keyboard Command Sequence:

        set_surface_type swept

Command:  set_surfaces
Description:  This command toggles the drawing of surfaces.
Keyboard Command Sequence:

        set_surfaces <on or off>

Another Button with the same description:

        SURFACES
GLOBAL

Command: **set_surfint_space**

Description: This command controls the construction of the space curve during surface/surface intersection.

**Keyboard Command Sequence:**

```
set_surfint_space <on/off>
```

TRIM 1

Command: **set_surfint_trim1**

Description: This command controls the construction of the trimming curve for the first surface during surface/surface intersection.

**Keyboard Command Sequence:**

```
set_surfint_trim1 <on/off>
```

TRIM 2

Command: **set_surfint_trim2**

Description: This command controls the construction of the trimming curve for the second surface during surface/surface intersection.

**Keyboard Command Sequence:**

```
set_surfint_trim2 <on/off>
```

TOLERANCE
Command:  **set_sweep_tolerance**  
**Description:** This command sets the tolerance to be used by the sweep commands in determining coincident entities.  

**Keyboard Command Sequence:**
```
set_sweep_tolerance <tolerance>
```

Command:  **set_symmetry_normal**  
**Description:** This command specifies a normal which, along with a point set by the **set_symmetry_point** command, defines a plane about which entities are duplicated by the symmetry duplication commands.  

**Keyboard Command Sequence:**
```
set_symmetry_normal <X component> <Y component> <Z component>
```

Command:  **set_symmetry_plane_normal_from_to**  
**Description:** This command sets the normal vector of the plane about which entities are reflected by the symmetry commands as the normalized displacement vector that arises when travelling from a location (FromX,FromY,FromZ) to another location (ToX,ToY,ToZ), as follows:

\[
\begin{align*}
NX &= ToX - FromX \\
NY &= ToY - FromY \\
NZ &= ToZ - FromZ
\end{align*}
\]

Both locations can be entered by clicking on grid points, nodes or points.  
This button is located in the MESH GENERATION->SYMMETRY menu.  

**Keyboard Command Sequence:**
```
set_symmetry_plane_normal_from_to
<FromX> <FromY> <FromZ> <ToX> <ToY> <ToZ>
```
Command: **set_symmetry_point**

Description: This command specifies a point which, along with a normal set by the `set_symmetry_normal` command, defines a plane about which entities are duplicated by the symmetry duplication commands.

Keyboard Command Sequence:

```
set_symmetry_point <X coordinate> <Y coordinate> <Z coordinate>
```

---

Command: **set_table_point_id**

Description: This command sets step size between table point labels to the specified value.

If the step size is 0, no table point labels will be displayed.
If step size is 1, all labels will be displayed.
If step size is 3, every third label will be displayed, etc.

Keyboard Command Sequence:

```
set_table_point_id <step size>
```
Command: **set_table_type**

Description: This command sets the type of the current table to the specified type. The type of information conveyed by a table makes the table accessible to the analysis code for various purposes. The valid table types are as follows:

- none
- time
- temperature
- density
- plastic_strain
- stress
- strain_rate
- stress_rate

Keyboard Command Sequence:

```plaintext
set_table_type <type>
```

Command: **set_table_xmax**

Description: This command specifies the maximum X value to be shown in the table. By default, this value is one.

Keyboard Command Sequence:

```plaintext
set_table_xmax <max x>
```
Command: **set_table_xmin**  
**Description:** This command specifies the minimum X value to be shown in the table. By default, this value is zero.

**Keyboard Command Sequence:**  
```
set_table_xmin <min x>
```

Command: **set_table_xname**  
**Description:** This command specifies a name for the x-axis of the current table.

**Keyboard Command Sequence:**  
```
set_table_xname <name>
```

Command: **set_table_xstep**  
**Description:** This command specifies the number of steps in X to be used in the table. By default, the number of steps is ten.

**Keyboard Command Sequence:**  
```
set_table_xstep <number of steps>
```
Command:  **set_table_ymax**  
Description:  This command specifies the maximum Y value to be shown in the table.  
  By default, this value is one.  

Keyboard Command Sequence:  

  set_table_ymax <max y>

Command:  **set_table_ymin**  
Description:  This command specifies the minimum Y value to be shown in the table.  
  By default, this value is zero.  

Keyboard Command Sequence:  

  set_table_ymin <min y>

Command:  **set_table_yname**  
Description:  This command specifies a name for the y-axis of the current table.  

Keyboard Command Sequence:  

  set_table_yname <name>
**YSTEP**

**Command:** `set_table_ystep`  
**Description:** This command specifies the number of steps in Y to be used in the table. By default, the number of steps is ten.  
**Keyboard Command Sequence:**

```
set_table_ystep <number of steps>
```

---

**set_tetmesh_aspect**

**Command:** `set_tetmesh_aspect`  
**Description:** This command sets the desired aspect ratio when automatically meshing with tetrahedrals. The aspect ratio is defined such that a value of one is the ideal shape for an element. Values greater than one indicate that elements of less than ideal aspect ratios are allowed.  
**Keyboard Command Sequence:**

```
set_tetmesh_aspect <aspect ratio>
```

---

**TRIM BARE SURFACES**

**Command:** `set_trim_surfs`  
**Description:** This command toggles the `trim_surfaces` parameter which tells that the command `clean_surface_loops` will trim bare surfaces or not.  
**Keyboard Command Sequence:**

```
set_trim_surfs <on or off>
```
**set_trimesh_aspect**

**Command:** `set_trimesh_aspect`

**Description:** This command sets the desired aspect ratio when automatically meshing with triangles. The aspect ratio is defined such that a value of one is the ideal shape for an element. Values greater than one indicate that elements of less than ideal aspect ratios are allowed.

**Keyboard Command Sequence:**

```
set_trimesh_aspect <aspect ratio>
```

---

**set_undo**

**Command:** `set_undo`

**Description:** This command toggles the undo feature. When undo is enabled, the user may undo the last operation which changed the model (including a previous undo).

**Keyboard Command Sequence:**

```
set_undo <on or off>
```

---

**set_vecmag**

**Command:** `set_vecmag`

**Description:** This command sets the vector magnification to be used in a post vector plot.

**Keyboard Command Sequence:**

```
set_vecmag <magnification factor>
```
**set_view_repeat**

**Command:** set_view_repeat  
**Description:** This command toggles view change repeat mode. When OFF, commands that cause the view to change (e.g. rot_x_for) execute just once. With repeat mode ON, these commands execute repeatedly, until stopped with a mouse-click or keystroke. By default, view change repeat is OFF. Use the command set_view_repeat_pause to set the pause between command executions.

**Keyboard Command Sequence:**  
set_view_repeat <on or off>

**set_view_repeat_pause**

**Command:** set_view_repeat_pause  
**Description:** This command specifies the number of seconds to pause between frames when view change repeat mode is ON. By default, this number is zero.

**Keyboard Command Sequence:**  
set_view_repeat_pause <number of seconds>

**set_xcurve_increment_id**

**Command:** set_xcurve_increment_id  
**Description:** This command sets the number of (x,y) points/increment label.

**Keyboard Command Sequence:**  
set_xcurve_increment_id <value>
TITLE

Command:  set_xcurve_title
Description:  This command sets the title to the experimental data plot.
Keyboard Command Sequence:
    set_xcurve_title <title>

XMAX

Command:  set_xcurve_xmax
Description:  This command sets the right bound for the abscissa.
Keyboard Command Sequence:
    set_xcurve_xmax <value>

XMIN

Command:  set_xcurve_xmin
Description:  This command sets the left bound for the abscissa.
Keyboard Command Sequence:
    set_xcurve_xmin <value>

X-AXIS

Command:  set_xcurve_xname
Description:  This command sets the label to the abscissa.
Keyboard Command Sequence:
    set_xcurve_xname <label>
XSTEP

Command: set_xcurve_xstep
Description: This command sets the number gridlines/gridlabel in the abscissa.
Keyboard Command Sequence:
   set_xcurve_xstep <value>

YMAX

Command: set_xcurve_ymax
Description: This command sets the upper bound for the ordinate.
Keyboard Command Sequence:
Command: set_xcurve_ymax <value>

YMIN

Command: set_xcurve_ymin
Description: This command sets the lower bound for the ordinate.
Keyboard Command Sequence:
   set_xcurve_ymin <value>

Y-AXIS

Command: set_xcurve_yname
Description: This command sets the label to the ordinate.
Keyboard Command Sequence:
   set_xcurve_yname <label>
YSTEP

Command:  set_xcurve_ystep
Description:  This command sets the number gridlines/gridlabel in the ordinate.
Keyboard Command Sequence:
set_xcurve_ystep <value>

NAME

Command:  set_xy_plot_curve_name
Description:  This command sets the name of a curve to be used in the legend of the plot.
Keyboard Command Sequence:
set_xy_plot_curve_name <curve> <name>

SHOW IDS

Command:  set_xy_plot_point_id
Description:  This command sets step size between plot point labels to the specified value.
If the step size is 0, no plot point labels will be displayed.
If step size is 1, all labels will be displayed.
If step size is 3, every third label will be displayed, etc.
Keyboard Command Sequence:
set_xy_plot_point_id <step size>

TITLE

Command:  set_xy_plot_title
Description:  This command sets the title of the plot.
Keyboard Command Sequence:
set_xy_plot_title <title>
**XMAX**

**Command:** set_xy_plot_xmax

**Description:** This command specifies the maximum X value to be shown in the plot. By default, this value is one.

**Keyboard Command Sequence:**

```
set_xy_plot_xmax <max x>
```

**XMIN**

**Command:** set_xy_plot_xmin

**Description:** This command specifies the minimum X value to be shown in the plot. By default, this value is zero.

**Keyboard Command Sequence:**

```
set_xy_plot_xmin <min x>
```

**X-AXIS**

**Command:** set_xy_plot_xname

**Description:** This command sets the name of the X axis of the plot.

**Keyboard Command Sequence:**

```
set_xy_plot_xname <name>
```

**XSTEP**

**Command:** set_xy_plot_xstep

**Description:** This command specifies the number of steps in X to be used in the plot. By default, the number of steps is ten.

**Keyboard Command Sequence:**

```
set_xy_plot_xstep <number of steps>
```
YMAX

Command: set_xy_plot_ymax
Description: This command specifies the maximum Y value to be shown in the plot. By default, this value is one.

Keyboard Command Sequence:

\[ \text{set\_xy\_plot\_ymax} \ \langle \text{max} \ y \rangle \]

YMIN

Command: set_xy_plot_ymin
Description: This command specifies the minimum Y value to be shown in the plot. By default, this value is zero.

Keyboard Command Sequence:

\[ \text{set\_xy\_plot\_ymin} \ \langle \text{min} \ y \rangle \]

Y-AXIS

Command: set_xy_plot_yname
Description: This command sets the name of the Y axis of the plot.

Keyboard Command Sequence:

\[ \text{set\_xy\_plot\_yname} \ \langle \text{name} \rangle \]

YSTEP

Command: set_xy_plot_ystep
Description: This command specifies the number of steps in Y to be used in the plot. By default, the number of steps is ten.

Keyboard Command Sequence:

\[ \text{set\_xy\_plot\_ystep} \ \langle \text{number \ of \ steps} \rangle \]
Command: set_zbuffered

Description: This command toggles z-buffering. Z-buffering performs hidden-object removal so that graphical objects drawn in the graphics area do not cover previously drawn objects that are closer to the viewer. You may want to turn z-buffering off when you’re running Mentat on an X device and you need better performance. For 2-D models and models drawn in wireframe mode, z-buffering is not normally required.

Keyboard Command Sequence:
```
set_zbuffered <on or off>
```

Another Button with the same description:
- Z-BUFFERED

Command: shell_offset

Description: This command specifies the shell offset to be used by the expand_lines and expand_shells commands. Shell offset is the offset distance from the existing nodes to their new location when the entity is expanded.

When expanding line elements to shell elements, the offset distance is measured along a line parallel to the line element’s normal axis.

When expanding shell elements to solid elements, the offset distance is measured along a line parallel to the shell element’s normal axis (CCW is positive).

Keyboard Command Sequence:
```
shell_offset <offset>
```

Command: shell_thickness

Description: This command specifies the shell thickness to be used by the expand_lines and expand_shells commands.
When expanding line elements to shell elements, the thickness is the shell element’s width.

When expanding shell elements to solid elements, the thickness is the solid element’s 3rd dimension and is perpendicular to the shell elements face.

**Keyboard Command Sequence:**

```
shell_thickness <thickness>
```

**Menu:** SHORTCUTS

**Description:** The SHORTCUTS menu is a user-defined menu, specified in user.ms, that provides a convenient location to place buttons for often-used commands and shortcuts to other menus.

Typically, a user will customize a copy of user.ms and place that file in their Mentat working directory. (Files in the current working directory take preference over those in the menus directory.) If a binary menu file is being used, it will have to be recompiled before changes to user.ms will take effect. The binary menu file can be recompiled by running Mentat with the -compile option:

```
mentat -compile xyz.msb
```

where xyz.msb is the name of the binary menu file to be produced. The page will provide some information regarding compiling the menu file.

See the Menus/Format file for a description of the menu file language. The page also provides some information regarding the menu file language.

Your MSC.Marc Mentat support representative can also provide help on customizing menus.
Command:  **show_all_views**

Description:  This command specifies that all four views are to be displayed. To show only one model view, use the **show_view** command.

This command also activates all four views, making them respond to most view setting commands.

Also see:  **show_view** and **activate_all_views**.

Keyboard Command Sequence:

```
show_all_views
```

Another Button with the same description:

SHOW ALL VIEWS

---

Command:  **show_annotation**

Description:  This command displays information about annotations. Annotations are specified by number.

Keyboard Command Sequence:

```
show_annotation <annotation number>
```

---

Command:  **show_composite**

Description:  This command specifies that the current composite material is to be displayed. Material layers and their relative thicknesses can be seen.

Keyboard Command Sequence:

```
show_composite
```

Another Button with the same description:

SHOW COMPOSITE
Command: show_curves
Description: This command displays information about the specified curve.
Keyboard Command Sequence:
    show_curves <curve id>

Command: show_elements
Description: This command displays information about the specified element.
Keyboard Command Sequence:
    show_elements <element id>

Command: show_history
Description: This command specifies that the history plot is to be displayed. To display the model, use the show_model command.
Keyboard Command Sequence:
    show_history

Another Button with the same description:
    SHOW HISTORY
**MODEL**

*Command:* `show_model`

*Description:* This command specifies that the model is to be displayed. This command is particularly useful after viewing table, history, or path plots.

*Keyboard Command Sequence:*

```
show_model
```

Another Button with the same description:

```
SHOW MODEL
```

**NODE INFO**

*Command:* `show_nodes`

*Description:* This command displays information about the specified node.

*Keyboard Command Sequence:*

```
show_nodes <node id>
```

Another Button with the same description:

```
SHOW
```

**DESIGN PLOT**

*Command:* `show_opt_plot`

*Description:* This command specifies that the response gradient plot is to be displayed. To display the model, use the `show_model` command.

*Keyboard Command Sequence:*

```
show_opt_plot
```

Another Button with the same description:

```
SHOW DSGN PLOT
```
Command:  `show_pathplot`
Description:  This command specifies that the path plot is to be displayed. To display the model, use the `show_model` command.

Keyboard Command Sequence:
```
show_pathplot
```
Another Button with the same description:

```
SHOW PATH PLOT
```

Command:  `show_points`
Description:  This command displays information about the specified point.

Keyboard Command Sequence:
```
show_points <point id>
```
Another Button with the same description:

```
SHOW
```

Command:  `show_section`
Description:  This command specifies that the current beam section is to be displayed. The branches and their thicknesses are displayed in 2-D space.

Keyboard Command Sequence:
```
show_section
```
Another Button with the same description:

```
SHOW BEAM SECT
```
**SHOW**

Command: *show_solids*

Description: This command displays information about the specified solid.

Keyboard Command Sequence:

```
show_solids <solid id>
```

---

**SHOW**

Command: *show_surfaces*

Description: This command displays information about the specified surface.

Keyboard Command Sequence:

```
show_surfaces <surface id>
```

---

**SHOW TABLE**

Command: *show_table*

Description: This command specifies that the table is to be displayed. To display the model, use the *show_model* command.

Keyboard Command Sequence:

```
show_table
```

Another Button with the same description:

```
TABLE
```
Command: **show_view**

**Description:** This command specifies the graphics area view that is to be displayed. You must specify the number of the view to show. To show all views, use the `show_all_views` command.

This command also activates the specified view, making it respond to most view setting commands. The previously shown view is deactivated.

Also see: `show_all_views` and `activate_view`.

**Keyboard Command Sequence:**

```
show_view <1, 2, 3, or 4>
```

Other Buttons with the same description:

2, 3, and 4

---

Command: **show_xcurve**

**Description:** Toggle the display from showing the model to the curves supplied by the experimental data capability.

**Keyboard Command Sequence:**

```
show_xcurve
```

Another Button with the same description:

SHOW DATA FIT
**Command:** \texttt{xcv\_model signiorini}

**Description:** This command is used if experimental data must be fitted using the Signiorini strain energy function $W$, which is given by:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{20}(I_1 - 3)^2$$

where $I_1$ and $I_2$ are the first and second invariant of the right Cauchy-Green strain tensor and $C_{10}$, $C_{01}$, and $C_{20}$ are the material parameters to be determined.

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (\texttt{material\_type mechanical:mooney}). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**

\texttt{xcv\_model signiorini}

---

**Subroutine:** \texttt{SINCER}

**Description:** The user subroutine \texttt{SINCER} allows you to define how much an elastic stress increment exceeds the yield stress. This allows the program to accurately take large increments such that the material goes from elastic to elastic-plastic. You return the value of $F_{PLAS}$, which is the fraction of the stress increment beyond the yield surface. \texttt{SINCER} should only be used if a yield surface other than the von Mises (J2) is used in conjunction with the Generalized Plasticity option.

---

**Menu:** \texttt{SIZES}

**Description:** This menu contains a display of the numbers of entities in the model and the amount of space they occupy in the database.
Menu: MATERIAL PROPERTIES->SHAPE MEMORY ALLOYS-> MECHANICAL (AURICCHIO'S)


The input parameters are explained as follows:

- \( \text{sigAS}_s \): Initial Stress for Austenite to Martensite
- \( \text{sigAS}_f \): Final Stress for Austenite to Martensite
- \( \text{sigSA}_s \): Initial Stress for Martensite to Austenite
- \( \text{sigSA}_f \): Initial Stress for Martensite to Austenite

**Alpha:**

Alpha is a pressure factor and it is measured from the difference between the response in tension and compression.

**Case 1)**
If the behaviors in tension and compression are the same, the value is set to ZERO.

**Case 2)**
If the behaviors in tension and compression have a difference as in the classical case of SMA, the value is set to 0.1 if there is no compression data for the phase transformation.

One value for the phase transformation in compression, let’s say \( \text{sigAS}_s\text{\_compression} \) is available.

Then, alpha is calculated as follows:

\[
\alpha = \frac{\sqrt{2/3}(\text{sigAS}_s\text{\_compression} - \text{sigAS}_s)}{(\text{sigAS}_s\text{\_compression} + \text{sigAS}_s)}
\]

**epsL:**

epsL is a scalar parameter representing the maximum deformation obtainable only by detwinning of the multiple-variant martensite (or Maximum strain obtainable by variant orientation).

Classical values for epsL are in the range 0.05 and 0.10.

MSC.Marc sets the default value as 0.07.
**THERMO-MECHANICAL MODEL**

**Menu:** MATERIAL PROPERTIES->SHAPE MEMORY ALLOWS->THERMO-MECHANICAL MODEL

**Description:** The material properties for SMA is well described in the website: [http://www.sma-inc.com/NiTiProperties.html](http://www.sma-inc.com/NiTiProperties.html). The input parameters used for thermo-mechanical shape memory alloy model and recommended values based on literature survey are explained as follows:

1. Young’s modulus, Poisson’s ratios, and thermal expansion coefficients for Austenite and Martensite: $E_M, E_A, v_A, v_M, \alpha_M, \alpha_M$.
   - The recommended values:
     - $E_M = 28000-41000$ Mpa
     - $E_A = 60000-83000$ Mpa
     - $v_A = v_M = 0.33$
     - $\alpha_M: 6.6E-6$ /deg C
     - $\alpha_A: 11.0E-6$ / deg C

2. Yield stresses of the pure martensite and austenite phases: $\sigma_M$ and $\sigma_A$.
   - The recommended values:
     - $\sigma_M = 70 -140$ Mpa
     - $\sigma_A = 195 -690$ Mpa

Generally, in thermo-mechanical model, total strain is additive form with several strains like

$$E = E(e_l: \text{elastic strain}) + E(p_l: \text{plastic strain}) + E(ph: \text{phase transformation strain}) + E(th: \text{ thermo-expansion strain}).$$

If user does not want to consider plastic strain, put $\sigma_M$ and $\sigma_A$ with high value as follows:

$$\sigma_M = \sigma_A = 1.0e+20$$ Mpa

3. The “unstressed transformation temperatures”: $\text{Temp}_{Ms}, \text{Temp}_{Mf}, \text{Temp}_{As}, \text{Temp}_{Af}$. The Transformation Temperatures (TTRs) are those temperatures at which the alloy changes from the higher temperature Austenite to the lower temperature Martensite or vice versa. The typical tests to determining TTRs are Constant Load method, Differential Scanning Calorimeter (DSC) method, and Active Af method (please see the details in the...
The range of transformation temperature is –200 to 100 deg C. So, it is difficult to recommend the default values. However, here is a guideline for two different SMA materials.

- The example values for SMA:
  - **SMA 1**
    - Temp\_Ms: -100 deg C, Temp\_Mf: -50 deg C,
    - Temp\_As: 5 deg C, Temp\_Af: 20 deg C
  - **SMA 2**
    - Temp\_Ms: -145.15 deg C, Temp\_Mf: -83.15 deg C,
    - Temp\_As: -85.15 deg C, Temp\_Af: -52.15 deg C

4. The coefficients C_m and C_a:
   - Provide the stress dependency of the transformation temperature as shown in theory manual.
   - The recommended values:
     - C\_M = 6.0 Mpa / deg C, C\_A = 8.0 Mpa / deg C

5. The calibration of the transformation strains: deviatoric transformation strain, volumetric transformation strain and detwinning stress.
   - The recommended values:
     - Deviatoric transformation strain: 0.08
     - Volumetric transformation strain: 0 – 0.003.
     - Detwinning stress: 100-150 Mpa

6. The calibration of the detwinning function:
   - “g” function is to fit the uniaxial stress vs. strain curve for pure, randomly oriented martensite conducted at a temperature below the Mf temperature.
   - The recommended values for g-function:
     - g-a = -4.0
     - g-b = 2.0
     - g-c <=0
     - g-d = 2.75
     - g-e <=0
     - g-f = 3.0
     - g-0 = 1.0e+20
     (Cut off values)
     - g-max: 1.0
     - stress at g-max: 1.0e+20
SOLID AREA

Command: **solid_area**

Description: This command calculates the total surface area of a set of solids. The specified set of solids do not have to be connected; the area is calculated as the sum of the areas of the individual solids.

Keyboard Command Sequence:

```
solid_area <solid list> #
```

solid_edge_divisions

Command: **solid_edge_divisions**

Description: This command specifies the number of element edges desired when automatically meshing a solid. For triangular and tetrahedral meshes, this will be matched exactly. Quadrilateral meshes will only match when the number of divisions is even. If an odd number of edges is specified, then the number of element edges created will be greater than one specified.

Keyboard Command Sequence:

```
solid_edge_divisions <ndiv> <solid edge list>
```

solid_edge_length

Command: **solid_edge_length**

Description: This command specifies the number of element edges desired when automatically meshing a solid. The number of divisions is determined by dividing the length of the edge by the length specified in the command, and rounding up to the next integer.

Keyboard Command Sequence:

```
solid_edge_length <length> <solid edge list>
```
**ELEMENTS**

**Command:** `solid_elements`

**Description:** This command causes the specified elements to be drawn in solid color. Elements can be drawn in translucent color with the `translucent_elements` command.

**Keyboard Command Sequence:**
```
solid_elements <element list> #
```

---

**solid_max_edge_divisions**

**Command:** `solid_max_edge_divisions`

**Description:** This command assures that the number of edges divisions on every given solid edge is no greater than the given number.

**Keyboard Command Sequence:**
```
solid_max_edge_divisions <Number of Divisions> <Solid Edge List>
```

---

**solid_min_edge_divisions**

**Command:** `solid_min_edge_divisions`

**Description:** This command assures that the number of edges divisions on every given solid edge is no less than the given number.

**Keyboard Command Sequence:**
```
solid_min_edge_divisions <Number of Divisions> <Solid Edge List>
```
**solid_quadmesh_faces**

**Command:** solid_quadmesh_faces

**Description:** This command automatically creates quadrilateral meshes on the faces of solids. Mesh density is controlled by specifying the number of edge divisions desired on each edge.

**Keyboard Command Sequence:**

```
solid_quadmesh_faces <solid face list>
```

---

**SOLIDS**

**Command:** solid_solids

**Description:** This command causes the specified solids to be drawn in opaque color. Solids can be drawn in translucent color with the translucent_solids command.

**Keyboard Command Sequence:**

```
solid_solids <solids list> #
```

---

**SOLIDS FACES**

**Command:** solid_solids_faces

**Description:** This command causes the specified solids’ faces to be drawn in opaque color. Solids’ faces can be drawn in translucent color with the translucent_solids_faces command.

**Keyboard Command Sequence:**

```
solid_solids_faces <solids faces list> #
```
**SURFACES**

**Command:** **solid_surfaces**

**Description:** This command causes the specified surfaces to be drawn in solid color. Surfaces can be drawn in translucent color with the **translucent_surfaces** command.

**Keyboard Command Sequence:**

```
solid_surfaces <surface list> #
```

**solid_tetmesh_solids**

**Command:** **solid_tetmesh_solids**

**Description:** This command automatically creates a tetrahedral mesh on solid bodies. Mesh density is controlled by specifying the number of edge divisions desired on each edge. The desired aspect ratio may also be specified.

**Keyboard Command Sequence:**

```
solid_tetmesh_solids <solid list>
```

**solid_trimesh_faces**

**Command:** **solid_trimesh_faces**

**Description:** This command automatically create triangular meshes on the faces of solids. Mesh density is controlled by specifying the number of edge divisions desired on each edge. The desired aspect ratio may also be specified.

**Keyboard Command Sequence:**

```
solid_trimesh_faces <solid face list>
```
**SOLID TYPE**

**Menu:** SOLID TYPE

**Description:** This menu contains commands for setting the type of solids to be created.

**SOLID VOLUME**

**Command:** solid_volume

**Description:** This command calculates the total volume of a set of solids. The specified set of solids do not have to be connected; the volume is calculated as the sum of the volumes of the individual solids.

**Keyboard Command Sequence:**

```
solid_volume <solid list> #
```

**SOLIDS**

**Menu:** SOLIDS

**Description:** This menu contains commands that perform operations on solids.

**SOLID**

**Command:** solids_filled

**Description:** This button allows toggling between solid and wireframe display of solid. If the toggle is ON, solid faces will be displayed with solid color. If the toggle is OFF, will be displayed in wireframe mode.

**Keyboard Command Sequence:**

```
solids_filled
solids_wireframe
```
**SURFACES FLAT**

**Command:** solids_flat  
**Description:** This command causes solids faces to be drawn without Gouraud shading (color interpolation to give a smooth appearance).  
**Keyboard Command Sequence:**  
```
solids_flat <on or off>
```

**SHEAR RELAXATION**

**Command:** xcv_model relax_spectrum  
**Description:** This command is used if experimental shear relaxation data must be fitted using the following Prony series expansion for the shear modulus \( G \):

\[
G(t) = G_{\infty} + \sum_{n=1}^{N} G_n \exp\left(-\frac{t}{\tau_n}\right)
\]

where \( t \) is the time, \( N \) is the number of terms and \( G_{\infty}, G_n \) and \( \tau_n \) are material parameters to be determined.  
The data points provided by the user must give the value of the shear modulus at different time stations, which do not need to be equi-spaced.  
**Keyboard Command Sequence:**  
```
xcv_model relax_spectrum
```

**BULK RELAXATION**

**Command:** xcv_model relax_spectrum_bulk  
**Description:** This command is used if experimental bulk relaxation data must be fitted using the following Prony series expansion for the bulk modulus \( K \):

\[
K(t) = K_{\infty} + \sum_{n=1}^{N} K_n \exp\left(-\frac{t}{\tau_n}\right)
\]

where \( t \) is the time, \( N \) is the number of terms and \( K_{\infty}, K_n, \) and \( \tau_n \) are material parameters to be determined.
The data points provided by the user must give the value of the bulk modulus at different time stations, which do not need to be equi-spaced.

**Keyboard Command Sequence:**

```
xcv_model relax_spectrum_bulk
```

---

**ENERGY RELAXATION**

**Command:** `xcv_model relax_spectrum_w`

**Description:** This command is used if experimental strain energy relaxation data must be fitted using the following Prony series expansion for the strain energy $W$:

$$
W(t) = W_{\text{infinity}} + \sum_{n=1}^{N} \delta_n W_0 \exp\left(-\frac{t}{\tau_n}\right)
$$

where $t$ is the time, $N$ is the number of terms, $W_0$ is the instantaneous strain energy, and $W_{\text{infinity}}, \delta_n,$ and $\tau_n$ are material parameters to be determined.

The data points provided by the user must give the value of the strain energy at different time stations, which do not need to be equi-spaced.

**Keyboard Command Sequence:**

```
xcv_model relax_spectrum_w
```

---

**SPLIT CURVES**

**Command:** `split_curves`

**Description:** This command splits curves into multiple curves at cusp points. This is useful to assure that vertices are located only at the endpoints, and should be done before using the `break_curves` command.

**Keyboard Command Sequence:**

```
split_curves <curve_list>
```
Command:  **split_solid_faces**

**Description:** This command splits the periodic faces of solids such that they are no longer periodic. This is necessary for automatic meshing and should be used whenever periodic faces are present before meshing is performed.

**Keyboard Command Sequence:**
```
split_solid_faces <solid list>
```

Command:  **spring_dof**

**Description:** This command sets the degrees of freedom at each end of the current spring class link. You must specify the slot (end) you wish to set and the degree of freedom. The slots are numbered 0 and 1. Degrees of freedom are specified as integers starting at 1.

**Keyboard Command Sequence:**
```
spring_dof <0 or 1> <dof>
```

Command:  **spring_node**

**Description:** This command sets the nodes that make up the ends of the spring for the current spring class link. You must specify the slot (end) you wish to set and a node. The slots are numbered 0 and 1.

**Keyboard Command Sequence:**
```
spring_node <0 or 1> <node>
```
Command:  
`spring_option`  
`spring_multi_option`  

Description: This command in the SPRINGS/DASHPOTS menu sets the property type for the current spring or for springs that will be created by subsequent calls to the `link_multi_spring_n_to_1` and `link_multi_spring_n_to_n` commands.

Valid spring property types are as follows:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static_type:stiffness</code></td>
<td>spring stiffness coefficient. Table used for nonlinear springs. Initial force optionally specified for linear springs. Numerical stabilizer can be turned on to zero out spring force.</td>
</tr>
<tr>
<td><code>static_type:force</code></td>
<td>spring force – table as a function of displacement is mandatory.</td>
</tr>
<tr>
<td><code>dynamic_type:damping</code></td>
<td>dashpot damping coefficient. Table used for nonlinear springs. Numerical stabilizer can be turned on to zero out dashpot force.</td>
</tr>
<tr>
<td><code>dynamic_type:force</code></td>
<td>dashpot force – table as a function of velocity is mandatory.</td>
</tr>
<tr>
<td><code>thermal_type:heat</code></td>
<td>spring heat transfer coefficient. Table used for nonlinear springs.</td>
</tr>
<tr>
<td><code>thermal_type:flux</code></td>
<td>spring thermal flux. Table as a function of temperature is mandatory.</td>
</tr>
<tr>
<td><code>electrical_type:conductivity</code></td>
<td>spring electrical conductivity. Table used for nonlinear springs.</td>
</tr>
<tr>
<td><code>electrical_type:current</code></td>
<td>spring current. Table as a function of voltage is mandatory.</td>
</tr>
</tbody>
</table>

Keyboard Command Sequence:

```
spring_option       < >  
spring_multi_option < >  
```

Other Buttons with the same description:

DAMPING COEFFICIENT
Command:  

**spring_param**  
**spring_multi_param**

Description:  This command in the SPRINGS/DASHPOTS menu sets the property values for the current spring or for springs that will be created by subsequent calls to the **link_multi_spring_n_to_1** and **link_multi_spring_n_to_n** commands.

For a linear spring, these values are used directly. For a nonlinear spring (i.e. when a table is used in conjunction), these property values scale the values in the table.

Keyboard Command Sequence:

```
spring_param       < >
spring_multi_param < >
```

---

**NUMERICAL STABILIZER**

Command:  

**spring_option**  
**spring_multi_option**

Description:  If set to ON, the spring and damper force are always zeroed out and the spring only acts as a link. This is used for both linear and nonlinear springs. In the latter case, the stabilizer flag can only be set when the stiffness option is used for the spring.

Keyboard Command Sequence:

```
spring_option stabilizer:on | off
spring_multi_option stabilizer:on | off
```
**Command:**  
- `spring_param_table`  
- `spring_multi_param_table`

**Description:** This command in the SPRINGS/DASHPOTS menu sets the table for the current spring or for springs that are created by subsequent calls to the `link_multi_spring_n_to_1` and `link_multi_spring_n_to_n` commands.

Valid spring table properties are as follows:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stiffness</td>
<td>spring stiffness table – table can be a function of time, normalized time, increment number, displacement, and/or temperature in a coupled analysis.</td>
</tr>
<tr>
<td>static_force</td>
<td>spring force table – table as a function of displacement is mandatory. Other variables can be time, normalized time, increment number, and/or temperature in a coupled analysis.</td>
</tr>
<tr>
<td>damping</td>
<td>dashpot damping table – table can be a function of time, normalized time, increment number, velocity and/or temperature in a coupled analysis.</td>
</tr>
<tr>
<td>dynamic_force</td>
<td>dashpot force table – table as a function of velocity is mandatory. Other variables can be time, normalized time, increment number, and/or temperature in a coupled analysis.</td>
</tr>
<tr>
<td>heat</td>
<td>spring heat transfer coefficient table – table can be a function of time, normalized time, increment number and/or temperature.</td>
</tr>
<tr>
<td>flux</td>
<td>spring flux table – table as a function of temperature is mandatory. Other variables can be time, normalized time, increment number.</td>
</tr>
<tr>
<td>conductivity</td>
<td>spring electrical conductivity table – table can be a function of time, normalized time, increment number, voltage and/or temperature in a coupled analysis.</td>
</tr>
</tbody>
</table>
Subroutine: **SSTRAN**

**Description:** The user subroutine **SSTRAN** allows the transformation of a substructure. The substructure may be either rotated, mirrored, or both. **SSTRAN** is called every time you use a previously generated substructure.

**Property Name** | **Description**
--- | ---
`current` | spring current table – table as a function of voltage is mandatory. Other variables can be time, normalized time, increment number, and/or temperature in a coupled analysis.

**Keyboard Command Sequence:**

```
spring_param_table < >
spring_multi_param_table < >
```

Command: **store_curves**

**Description:** This command creates a set from a list of curves. You must specify the set name and a list of curves.

**Keyboard Command Sequence:**

```
store_curves <set name> <curve list> #
```
Command: store_edges
Description: This command creates a set from a list of edges. You must specify the set name and a list of edges.

Keyboard Command Sequence:

    store_edges <set name> <edge list> #

Command: store_elements
Description: This command creates a set from a list of elements. You must specify the set name and a list of elements.

Keyboard Command Sequence:

    store_elements <set name> <element list> #

Command: store_faces
Description: This command creates a set from a list of faces. You must specify the set name and a list of faces.

Keyboard Command Sequence:

    store_faces <set name> <face list> #
**STORE NODE PATH**

**Command:** `store_node_path`

**Description:** This command creates a set from a path of nodes. You must specify the set name and key nodes in a continuous path of nodes. The order of the nodes is maintained in the set.

**Keyboard Command Sequence:**

```
store_node_path <set name> <node path> #
```

---

**STORE**

**Command:** `store_nodes`

**Description:** This command creates a set from a list of nodes. You must specify the set name and a list of nodes.

**Keyboard Command Sequence:**

```
store_nodes <set name> <node list> #
```

---

**STORE ORDERED**

**Command:** `store_nodes_ordered`

**Description:** This command creates an ordered set of nodes. You must specify the set name and the nodes in the set. The order of the nodes is maintained in the set.

**Keyboard Command Sequence:**

```
store_nodes_ordered <set name> <nodes> #
```
**Store**

**Command:** store_points

**Description:** This command creates a set from a list of points. You must specify the set name and a list of points.

**Keyboard Command Sequence:**

```plaintext
store_points <set name> <point list> #
```

**Store**

**Command:** store_surfaces

**Description:** This command creates a set from a list of surfaces. You must specify the set name and a list of surfaces.

**Keyboard Command Sequence:**

```plaintext
store_surfaces <set name> <surface list> #
```

**Stress PT ID’s**

**Command:** stress_pt_labels

**Description:** This command toggles the drawing of stress points on beam sections.

**Keyboard Command Sequence:**

```plaintext
stress_pt_labels <on or off>
```

**Stretch**

**Menu:** STRETCH

**Description:** This menu contains commands that relocate nodes along a straight line.
Command: **stretch_bias**

**Description:** This command specifies the bias factor used by the `stretch_nodes` command. The bias factor is a number between -1 and 1 which affects the positioning of nodes. A positive bias factor causes the nodes to be stretched toward the first specified node. A negative bias factor causes the nodes to be stretched toward the last node specified. Biasing is parabolic. A bias factor of zero (the default) causes the nodes to be spaced equally along the path.

Singularities may occur if the magnitude of the bias factor is greater than one third.

**Keyboard Command Sequence:**

```
stretch_bias <bias factor>
```
BIAS FACTORS

Command: sub_bias_factors
Description: This command sets the bias factors used by the subdivide command. Bias factors are specified for each of the element’s principle directions. (For line elements, only the U-direction factor is used. For two-dimensional elements, only the U- and V-direction factors are used.) Bias factors are numbers between -1 and 1.

Positive factors skew new element creation position in the element’s positive direction, negative factors skew it in the negative direction, and 0 bias factors produce evenly distributed elements.

Keyboard Command Sequence:

    sub_bias_factors <U-direction factor>
    <V-direction factor> <W-direction factor>

DIVISIONS

Command: sub_divisions
Description: This command sets the number of elements that are created from every element specified for the subdivide command.

Keyboard Command Sequence:

    sub_divisions <number of elements in U>
    <number of elements in V>
    <number of elements in W>

SUBDIVIDE

Menu: SUBDIVIDE
Description: This menu contains commands that generate elements by subdividing existing one, two, or three-dimensional elements.
**CURVES**

Command: **subdivide_curves**

Description: This command breaks specified curves into smaller curves. The number of curves created is controlled by the number of subdivisions set by the `sub_divisions` command. The size and location of the created curves is controlled by the bias factors set by the `sub_bias_factors` command.

Keyboard Command Sequence:

```
subdivide_curves <curve list> 
```

---

**ELEMENTS**

Command: **subdivide_elements**

Description: This command breaks specified elements into smaller elements. The number of elements created is controlled by the number of subdivisions set by the `sub_divisions` command. The size and location of the created elements is controlled by the bias factors set by the `sub_bias_factors` command.

Keyboard Command Sequence:

```
subdivide_elements <element list> 
```

---

**ELEMENTS TO HEX**

Command: **subdivide_elements2hex**

Description: This command converts TETRA4 and PENTA6 elements into HEX8 elements. Each TETRA4 element is converted into four HEX8 elements, and each PENTA6 element is converted into six HEX8 element.

In order to keep the mesh conformity, each existing HEX8 element is also broken into eight smaller HEX8 elements. The subdivision is not controlled by DIVISION NUMBERS and BIAS FACTORS.

Keyboard Command Sequence:

```
subdivide_elements2hex <element list> 
```
**ELEMENTS TO QUAD**

**Command:** `subdivide_elements2quad`

**Description:** This command converts TRIA3 elements into QUAD4 elements by adding one node interior to each triangle and three nodes on the three edges. Therefore, each triangular element is converted into three quad elements.

In order to keep the mesh conformity, each existing QUAD4 element is also broken into four smaller quad elements. The subdivision is not controlled by DIVISION NUMBERS and BIAS FACTORS.

**Keyboard Command Sequence:**

```
subdivide_elements2quad <element list> #
```

---

**RESET**

**Command:** `subdivide_reset`

**Description:** This command resets the subdivide divisions and bias factors used by the subdivide command to their default value.

**Keyboard Command Sequence:**

```
subdivide_reset
```
**SUBMIT 1**

**Command:** submit_job

**Description:** This command submits a job to the MSC.Marc analysis program. It first writes an input deck with a name of the form model_job.dat and then starts the job. The job submission files submit1, submit2, and submit3, located in the tools directory, may be tailored by the user. You must specify the job submission number (1, 2, or 3) which specifies which job submission file to use.

If a submit_job is done after opening a database file (mfd file) and MSC.Marc Mentat responds with Job already running!, do a RESET (job_submit_reset) and then submit the job again.

**Keyboard Command Sequence:**

    submit_job <1, 2, or 3>

Other Buttons with the same description:

    SUBMIT 2
    SUBMIT 3

**SURFACES TO INTERPOLATED**

**Command:** surface_interpolated

**Description:** This command converts surfaces into interpolated surfaces. The original surface is sampled at the user defined subdivisions and an interpolated surface is passed through them. The resulting surface is a bicubic NURBS surface with C1 and C2 continuity.

**Keyboard Command Sequence:**

    surface_interpolated <surface list> #
**SURFACES TO POLYQUADS**

**Command:** `surface_polyquads`

**Description:** This command converts the specified surfaces into polyquads (surfaces composed of a regular array quadrilateral patches).

**Keyboard Command Sequence:**

```
surface_polyquads <surface list> #
```

---

**SURFACE TYPE**

**Menu:** SURFACE TYPE

**Description:** This menu contains commands for setting the type of geometric surfaces to be created with the `add_surfaces` command.

Available surface types are as follows:

- **QUAD**: A bilinear surface defined by four corner points.
- **BEZIER**: A bezier surface defined by a list of control points.
- **DRIVEN**: A surface defined by a driven curve and a driving curve.
- **NURB**: A general NURBS surface given by a complete definition: the number of control points in each direction, the curve order in each direction, a list of control points, a list of homogeneous coordinates, and a knot vector.
- **RULED**: A linear surface between two curves.
- **SPHERE**: A sphere by giving the center and radius.
- **CYLINDER**: A cylindrical (or conical) surface.
- **SWEPT**: A surface defined by a swept curve, a sweeping curve, and the number of steps.
- **INTERPOLATE**: A cubic NURBS surface that interpolates `npu*npv` control points.
- **COONS**: Coons linear surface defined by four boundary curves.
- **SKIN**: Smooth curve passing through a list of curves.
Mentat Help Commands in S

Mentat Help Reference

SURFACES FLAT

Command: surfaces_flat

Description: This command causes surface faces to be drawn without Gouraud shading (color interpolation to give a smooth appearance).

Keyboard Command Sequence:

surfaces_flat <on or off>

SURFACES FLAT

Command: surfaces_solid

Description: This button allows toggling between solid and wireframe display of surfaces. If the toggle is ON, surfaces will be displayed with solid color. If the toggle is OFF, surfaces will be displayed in wireframe mode. In this mode, surface faces will be represented by a diagonal line indicating the first edge.

Keyboard Command Sequence:

surfaces_solid
surfaces_wireframe

Another Button with the same description:

SURF'S

SAMPLED
A surface created by a sequence of points. It is useful for creating surfaces from digitized data. Input is the origin point of the surface, a point in the first direction (to establish the first direction of the surface), then a point in the second direction of the surface (to establish the second direction of the surface), and then all the points that will define the surface (including selecting the origin point and the points in the first and second directions again).
**Menu:** SWEEP

**Description:** This menu contains commands for cleaning up the model. There are commands for removing coincident entities, renumbering entities, and removing unused entities. Entities affected by these commands include points, curves, surfaces, nodes, and elements.

**Command:** sweep_all

**Description:** This command removes coincident points, curves, surfaces, nodes, and elements from the mesh. These entities are considered to be coincident if they occupy the same space within the tolerance distance specified with the set_sweep_tolerance command. This command removes coincident entities throughout the model; i.e. no list of entities is required.

**Keyboard Command Sequence:**

```
sweep_all
```

**Command:** sweep_curves

**Description:** This command removes coincident curves from the specified list. Curves are coincident if they have identical points, weights, and knots. It is generally recommended to remove coincident points with the sweep_points command before executing this command. All but one of each set of coincident curves are removed.

**Keyboard Command Sequence:**

```
sweep_curves <curve list> #
```
**ELEMENTS**

Command: **sweep_elements**

Description: This command removes coincident elements from the specified list. Elements are coincident if they have identical nodes and are of the same class. It is generally recommended to remove coincident nodes with the `sweep_nodes` command before executing this command. All but one of each set of coincident elements are removed.

Keyboard Command Sequence:

```
sweep_elements <element list> #
```

**NODES**

Command: **sweep_nodes**

Description: This command removes coincident nodes from the specified list. Nodes are considered to be coincident if they are located closer than the tolerance distance specified with the `set_sweep_tolerance` command. All but one of each set of coincident nodes are removed.

Keyboard Command Sequence:

```
sweep_nodes <node list> #
```

Another Button with the same description:

SWEEP OUTLINE NODES

**POINTS**

Command: **sweep_points**

Description: This command removes coincident points from the specified list. Points are considered to be coincident if they are located closer than the tolerance distance specified with the `set_sweep_tolerance` command. All but one of each set of coincident points are removed.

Keyboard Command Sequence:

```
sweep_points <point list> #
```
**SURFACES**

**Command:** `sweep_surfaces`

**Description:** This command removes coincident surfaces from the specified list. Surfaces are coincident if they have identical points, weights, and knots. It is generally recommended to remove coincident points with the `sweep_points` command before executing this command.

**Keyboard Command Sequence:**

```
sweep_surfaces <surface list> #
```
pick, box pick, and polygon pick methods if the `symmetry_combined` command is executed. This allows, for example, to simultaneously duplicate elements and surfaces, but not curves. Wildcards like `all_existing` and `all_selected` can also be used with this command to indicate all existing or all selected items of the active types.

The plane of symmetry is defined by a point and a normal vector, set by the `set_symmetry_point` and `set_symmetry_normal` commands, respectively.

Also see: `symmetry_nodes`  
`symmetry_elements`  
`symmetry_points`  
`symmetry_curves`  
`symmetry_surfaces`  
`symmetry_solids`  
`symmetry_ties`  
`symmetry_servos`  
`symmetry_springs`  
`symmetry_rbe2s`  
`symmetry_rbe3s`

This button is located in the MESH GENERATION->SYMMETRY menu.

**Keyboard Command Sequence:**

```
symmetry_combined <item list> #
set_symmetry_combined <item type> <on|off>
```

---

**Command:** `symmetry_curves`

**Description:** This command duplicates the specified curves about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command. The control points associated with the curve are also duplicated.

**Keyboard Command Sequence:**

```
symmetry_curves <curve list> #
```
Command: *symmetry_elements*

Description: This command duplicates the specified elements about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.

Keyboard Command Sequence:

```
symmetry_elements <element list> 
```

Command: *symmetry_nodes*

Description: This command duplicates the specified nodes about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.

Keyboard Command Sequence:

```
symmetry_nodes <node list> 
```

Command: *symmetry_points*

Description: This command duplicates the specified points about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.

Keyboard Command Sequence:

```
symmetry_points <point list> 
```
**Command:** `symmetry_rbe2s`  
**Description:** This command duplicates the specified RBE2’s about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.  

This button is located in the MESH GENERATION->SYMMETRY menu.  

**Keyboard Command Sequence:**  
`symmetry_rbe2s <rbe2 list> #`

---

**Command:** `symmetry_rbe3s`  
**Description:** This command duplicates the specified RBE3’s about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.  

This button is located in the MESH GENERATION->SYMMETRY menu.  

**Keyboard Command Sequence:**  
`symmetry_rbe3s <rbe3 list> #`

---

**Command:** `symmetry_reset`  
**Description:** This command resets the symmetry plane point and normal used by the symmetry commands to their default values.  

**Keyboard Command Sequence:**  
`symmetry_reset`
SERVOS

Command: symmetry_servos
Description: This command duplicates the specified servo links about the plane defined by the point set by the set_symmetry_point command and a normal set by the set_symmetry_normal command.

This button is located in the MESH GENERATION->SYMMETRY menu.

Keyboard Command Sequence:

symmetry_servos <servo link list> #

SOLIDS

Command: symmetry_solids
Description: This command duplicates the specified solids about the plane defined by the point set by the set_symmetry_point command and a normal set by the set_symmetry_normal command.

Keyboard Command Sequence:

symmetry_solids <solid list> #

SPRINGS

Command: symmetry_springs
Description: This command duplicates the specified springs about the plane defined by the point set by the set_symmetry_point command and a normal set by the set_symmetry_normal command.

This button is located in the MESH GENERATION->SYMMETRY menu.

Keyboard Command Sequence:

symmetry_springs <spring list> #
**SURFACES**

Command: `symmetry_surfaces`

Description: This command duplicates the specified surfaces about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.

Keyboard Command Sequence:

```
symmetry_surfaces <surface list> #
```

**TIES**

Command: `symmetry_ties`

Description: This command duplicates the specified nodal ties about the plane defined by the point set by the `set_symmetry_point` command and a normal set by the `set_symmetry_normal` command.

This button is located in the MESH GENERATION->SYMMETRY menu.

Keyboard Command Sequence:

```
symmetry_ties <tie list> #
```

**ALIGN**

Command: `system_align`

Description: This command defines a user coordinate system using the three specified points. You must specify the origin, a point on the U-axis, and a point on the V-axis. All three points are specified in global coordinates.

Keyboard Command Sequence:

```
system_align <XYZ coordinates of origin>
<XYZ coordinates of U-axis point>
<XYZ coordinates of V-axis point>
```
Command: **system_command**

**Description:** This command executes the specified command at the system level as though you were entering it from the operating system command line. The command and the output will be displayed in the dialogue area. On UNIX systems, the command is executed by the C shell (csh).

**Keyboard Command Sequence:**

```
system_command <csh command>
<command arguments (if any)>
```

---

**CYLINDRICAL**

Command: **system_cylindrical**

**Description:** This command sets the type of the current coordinate system to cylindrical. In the cylindrical coordinate system, coordinates are specified by radius, angle of deflection, and position in the Z-direction.

**Keyboard Command Sequence:**

```
system_cylindrical
```

---

**RECTANGULAR**

Command: **system_rectangular**

**Description:** This command sets the type of the current coordinate system to rectangular. In the rectangular coordinate system, coordinates are specified by position in X, Y, and Z-directions.

**Keyboard Command Sequence:**

```
system_rectangular
```
**RESET**

**Command:** `system_reset`

**Description:** This command resets the coordinate system to the default global rectangular coordinate system.

**Keyboard Command Sequence:**

```
system_reset
```

---

**ROTATE**

**Command:** `system_rotate`

**Description:** This command rotates the coordinate system about the global X, Y, and Z-axes in that order. You must specify the rotations in degrees.

**Keyboard Command Sequence:**

```
system_rotate <X-axis rotation> <Y-axis rotation> <Z-axis rotation>
```

---

**SYSTEM SHELL**

**Command:** `system_shell`

**Description:** This command invokes an operating system shell from within MSC.Marc Mentat. This allows you to execute system commands without leaving MSC.Marc Mentat. To return from the shell after you have finished with the shell, type a carriage return. On UNIX systems, you must specify the type of shell you want, C shell (csh) or Bourne shell (sh).

**Keyboard Command Sequence:**

```
system_shell <csh or sh>
```
**SPHERICAL**

Command:  `system_spherical`

Description: This command sets the type of the current coordinate system to spherical. In the spherical coordinate system, coordinates are specified by radius, angle from the horizontal axis, and angle from the vertical axis.

Keyboard Command Sequence:

```
system_spherical
```

---

**TRANSLATE**

Command:  `system_translate`

Description: This command translates the current coordinate system in the global X, Y, and Z-directions by the specified translations.

Keyboard Command Sequence:

```
system_translate <X translation> <Y translation> <Z translation>
```
Mentat Help Commands in T

**TABLE**

**Menu:** TABLE

**Description:** This menu contains commands for defining tables. Tables represent functions that can be used to define the variation of a quantity versus other parameters such as variables can be time, increment, temperature, density, plastic strain, stress, strain rate, and stress rate.

**ADD POINT**

**Command:** `table_add`

**Description:** This command adds a data point to the current table. The command adds another data point to the function $y = f(x)$. You must specify the tabular $x$ and $y$ coordinates of the point to add.

**Keyboard Command Sequence:**

```
table_add <x value> <y value>
```
**CLEAR**

**Command:** \texttt{table\_clear}

**Description:** This command clears the current table. All data points are removed from the table.

**Keyboard Command Sequence:**

\texttt{table\_clear}

---

**DIFFERENTIATE**

**Command:** \texttt{table\_differentiate}

**Description:** This command numerically differentiates the function represented by the values in the current table and makes the new values in the table.

Also see the \texttt{table\_integrate} command.

Use this command cautiously, as differentiation of tabular values is numerically unstable. (For example, if you differentiate a table twice and integrate twice, you may not end up with the same values that you started with.)

**Keyboard Command Sequence:**

\texttt{table\_differentiate}

---

**EDIT**

**Command:** \texttt{table\_edit}

**Description:** This command edits an existing tabular point. You must specify the point to be edited by number, and new tabular x and y coordinates of the point.

**Keyboard Command Sequence:**

\texttt{table\_edit <point number> <x value> <y value>}

Command: **table_fill**

Description: This command shifts and scales all table point values in the x and y-directions, so that the values fill the x and y limits of the current table.

Also see: **table_fit**, **table_shift**, and **table_scale**.

Keyboard Command Sequence:

```
   table_fill
```

Command: **table_filled**

```
   table_filled
   table_unfilled
```

Description: These commands specify whether or not areas under tabular functions be filled with color.

Keyboard Command Sequence:

```
   table_filled
   table_unfilled
```

Command: **table_fit**

Description: This command adjusts the limits of the current table so that all table point values fit within those limits.

Also see: **table_fill**, **set_table_xmin**, **set_table_xmax**, **set_table_ymin**, and **set_table_ymax**.

Keyboard Command Sequence:

```
   table_fit
```
Command: **table_formula**

Description: This command specifies a formula to be used as the function represented by the table. You must specify an arithmetic expression in terms of the variable \( x \).

Formulas should be entered as Fortran-like expressions. The expression can include transcendentals (\( \text{SIN}, \text{COS}, \text{TAN}, \text{ASIN}, \text{ACOS}, \text{ATAN} \)). The \( \text{LOG} \) keyword returns a natural logarithm. The parameters \( \pi \) and \( e \) are also available. Addition, multiplication, subtraction, and division are represented as +, *, -, /, respectively. The power of a number is represented by the symbol \(^\)\( \). You can use nesting to a level of 20 by enclosing expressions in parentheses.

Examples of legal expressions follow:

\[
1.2\times10^5 \times (3 \times (\sin(x) / 12.34)) / 2.0
\]
\[
x^{\pi}
\]
\[
1.23^{\pi \times \sin(3 \times 1.0\times10^{-2})}
\]

Keyboard Command Sequence:

```
table_formula <formula expression>
```

Command: **table_integrate**

Description: This command numerically integrates the function represented by the values in the current table and makes the new values in the table.

Also see the **table_differentiate** command.

Keyboard Command Sequence:

```
table_integrate
```
**NAME**

**Command:** table_name

**Description:** This command sets or changes the name of the current table.

**Keyboard Command Sequence:**

table_name <name>

---

**READ**

**Command:** table_read

**Description:** This command reads the specified file containing tabular data into a table. Note that this command creates a new entry in the list of tables and makes it the current table. To read tabular data into the current table, use the table_read_curr command.

The file must contain data of the correct form in order to be read successfully.

The tabular data file can be created by hand or produced by the table_write command and is written in free format. Comments are allowed in the file by placing a # at the beginning of the line. The order of the data is significant; its format (i.e. position on the line, number of items on each line) is not.

An example of a tabular data file is given below.

```
# Title
table1
# X-axis Label
X
# Y-axis Label
Y
# Type
0
# Steps in X and Y
10 10
# X-min, X-max, Y-min, Ymax
0.0 1.0 0.0 1.0
# Number of Data Points
11
# Data Points: X, Y, Point ID
```
Keyboard Command Sequence:

```
table_read <file name>
```

**Command:**  `table_reeval`

**Description:** This command reevaluates the current table’s formula (set with the `table_formula` command) and redisplays the table. This command is useful after changing table limits and step sizes.

Also see: `set_table_xmin`, `set_table_xmax`, `set_table_ymin`, `set_table_ymax`, `set_table_xstep`, and `set_table_ystep`.

Keyboard Command Sequence:

```
table_reeval
```

---

**Command:**  `table_remove`

**Description:** This command removes a data point from the current table. You must specify the point to be removed by number.

Keyboard Command Sequence:

```
table_remove <point number>
```
**READ**

**Command:**
- `table_read`
- `table_write`
- `table_read_raw`
- `table_write_raw`

**Description:** These commands read and write tables to a given file. Normally, all information about the table is read or written. In raw mode only xy pairs of data are used, all other information being excluded. This mode is to support transferring data from other programs.

**Keyboard Command Sequence:**
```
table_read <filename>
table_write <filename>
table_read_raw <filename>
table_write_raw <filename>
```

Another Button with the same description:
- WRITE

**SCALE**

**Command:** `table_scale`

**Description:** This command scales all table point values in the x and y-directions by the specified factors.

**Keyboard Command Sequence:**
```
table_scale <x factor> <y factor>
```
**COMMAND: table_shift**

**Description:** This command shifts all table point values in the x and y-directions by the specified values.

**Keyboard Command Sequence:**

```
table_shift <x shift> <y shift>
```

**COMMAND: table_swapxy**

**Description:** This command swaps the x and y values of the current table.

**Keyboard Command Sequence:**

```
table_swapxy
```

**COMMAND: table_filled table_unfilled**

**Description:** These commands specify whether or not areas under tabular functions be filled with color.

**Keyboard Command Sequence:**

```
table_filled
table_unfilled
```
**table_write**

**Command:** table_write

**Description:** This command writes a file that contains tabular data from the current table. Tabular data files can be read into the current table with the table_read command. You must specify the name of the file to write the data to.

**Keyboard Command Sequence:**

table_write <file name>

---

**X FORMULA**

**Command:** table_xformula

**Description:** This command specifies the formula that is used for displaying the x values of the current table. The x values are not modified; the formula only affects their plot positions.

Formulas should be entered as Fortran-like expressions. The expression can include transcendentals (SIN, COS, TAN, ASIN, ACOS, and ATAN). The LOG keyword returns a natural logarithm. The parameter PI is also available. Addition, multiplication, subtraction, and division are represented as +, *, -, /, respectively. The power of a number is represented by the symbol **. You can use nesting to a level of 20 by enclosing expressions in parentheses.

Examples of legal expressions follow:

- \(1.2e+05 \times (3 \times (\sin(x) / 12.34)) / 2.0\)
- \(x^{**}\text{PI}\)
- \(1.23 ** (\text{PI} \times \sin(3 \times 1.0e-02))\)

**Keyboard Command Sequence:**

table_xformula <formula expression>
Y FORMULA

Command: table_yformula

Description: This command specifies the formula that is used for displaying the y values of the current table. The y values are not modified; the formula only affects their plot positions.

Formulas should be entered as Fortran-like expressions. The expression can include transcendentals (SIN, COS, TAN, ASIN, ACOS, and ATAN). The LOG keyword returns a natural logarithm. The parameter PI is also available. Addition, multiplication, subtraction, and division are represented as +, *, -, /, respectively. The power of a number is represented by the symbol **. You can use nesting to a level of 20 by enclosing expressions in parentheses.

Examples of legal expressions follow:

- \(1.2e+05 \times (3 \times (\sin(x) / 12.34)) / 2.0\)
- \(x^{\times \pi}\)
- \(1.23^{\times (\pi \times \sin(3 \times 1.0e-02))}\)

Keyboard Command Sequence:

```
table_yformula <formula expression>
```

---

tally

Command: tally

Description: This command prints out a tally of every command in Mentat and the number of times each command has been executed in the current session. The tally is written to a file. You must specify the file name.

Keyboard Command Sequence:

```
tally <file name>
```
**TENSOF**

**Subroutine:**  TENSOF  

**Description:** The user subroutine TENSOF allows the definition of the post-failure behavior at an integration point. By default, the reduction of the cracking stress to zero is a linear function of the crack strain.  

TENSOF is automatically called for every crack in the analysis.

---

**THIRD ORDER DEFORMATION**

**Command:** xcv_model third_order_deformation  

**Description:** This command is used if experimental data must be fitted using the third order deformation strain energy function $W$, which is given by:

$$W = C10*(I1 - 3) + C01*(I2 - 3) + C11*(I1 - 3)*(I2 - 3) + C20*(I1 - 3)^2 + C30*(I1 - 3)^3$$

where $I1$ and $I2$ are the first and second invariant of the right Cauchy-Green strain tensor and $C10, C01, C11, C20,$ and $C30$ are the material parameters to be determined.  

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (material_type mechanical:mooney). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**  

xcv_model third_order_deformation
**TYPE**

**Command:** tie_type  
**Description:** This command sets the Marc tie type for the current nodal tie class link. Tie types are specified as integers. See *MSC.Marc Volume C: Program Input Manual* for valid tie types.

**Keyboard Command Sequence:**
```
tie_type <tie type>
```

**DOF**

**Command:** tied_dof  
**Description:** This command sets the tied degree of freedom for the current servo link class link. Degrees of freedom are specified as integers starting at 1.

**Keyboard Command Sequence:**
```
tied_dof <tied dof>
```

**NODE**

**Command:** tied_node  
**Description:** This command sets the tied node for the current nodal tie or servo link class link.

**Keyboard Command Sequence:**
```
tied_node <tied node>```
Command: **touching_bodies**

**Description:** Sets the current entry in the table to be modified. At least one of the bodies must be deformable.

The bodies are entered either by typing their names on the command line or by clicking on the entries in the pop-up menu.

**Keyboard Command Sequence:**

```
touching_bodies <first contact body>
<second contact body>
```

---

Command: **trans_camera_cspace_c_for**

**Description:** This command translates the camera along the camera space X, Y, and Z-axes, each in the positive direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_cspace_c_for
```
Command: **trans_camera_cspace_c_rev**

**Description:** This command translates the camera along the camera space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_cspace_c_rev
```

---

Command: **trans_camera_cspace_x_for**

**Description:** This command translates the camera along the camera space X-axis, in the positive direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_cspace_x_for
```
Command:  \textit{trans\_camera\_cspace\_x\_rev}

Description: This command translates the camera along the camera space X-axis, in the negative direction, by a distance set with the \textit{trans\_camera\_increment} command.

This command acts on all the currently active views.

Also see: \textit{trans\_camera\_increment}, \textit{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_camera\_cspace\_x\_rev}

---

Command:  \textit{trans\_camera\_cspace\_y\_for}

Description: This command translates the camera along the camera space Y-axis, in the positive direction, by a distance set with the \textit{trans\_camera\_increment} command.

This command acts on all the currently active views.

Also see: \textit{trans\_camera\_increment}, \textit{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_camera\_cspace\_y\_for}
Command: `trans_camera_cspace_y_rev`

**Description:** This command translates the camera along the camera space Y-axis, in the negative direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_cspace_y_rev
```
Command: **trans_camera_cspace_z_rev**

Description: This command translates the camera along the camera space Z-axis, in the negative direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_cspace_z_rev
```

---

Command: **trans_camera_increment**

Description: This command allows you to set the translation increment angle that will be used by subsequent camera translation commands such as `trans_camera_cspace_x_for`.

See the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_increment <delta>
```
Command: **trans_camera_mspace_c_for**

**Description:** This command translates the camera along the model space X, Y, and Z-axes, each in the positive direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_mspace_c_for
```

---

Command: **trans_camera_mspace_c_rev**

**Description:** This command translates the camera along the model space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_mspace_c_rev
```
Command:  trans_camera_mspace_x_for
Description:  This command translates the camera along the model space X-axis, in the positive direction, by a distance set with the trans_camera_increment command. This command acts on all the currently active views. Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_camera_mspace_x_for

Command:  trans_camera_mspace_x_rev
Description:  This command translates the camera along the model space X-axis, in the negative direction, by a distance set with the trans_camera_increment command. This command acts on all the currently active views. Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_camera_mspace_x_rev
Command: trans_camera_mspace_y_for

Description: This command translates the camera along the model space Y-axis, in the positive direction, by a distance set with the trans_camera_increment command.

This command acts on all the currently active views.

Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

trans_camera_mspace_y_for

Command: trans_camera_mspace_y_rev

Description: This command translates the camera along the model space Y-axis, in the negative direction, by a distance set with the trans_camera_increment command.

This command acts on all the currently active views.

Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

trans_camera_mspace_y_rev
Command:  trans_model_mspace_z_for
Description:  This command translates the model along the model space Z-axis, in the positive direction, by a distance set with the trans_model_increment command.

This command acts on all the currently active views.

Also see: trans_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_model_mspace_z_for

Command:  trans_camera_mspace_z_rev
Description:  This command translates the camera along the model space Z-axis, in the negative direction, by a distance set with the trans_camera_increment command.

This command acts on all the currently active views.

Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_camera_mspace_z_rev
Command: **trans_camera_vspace_c_for**

Description: This command translates the camera along the view space X, Y, and Z-axes, each in the positive direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_camera_vspace_c_for
```

---

Command: **trans_camera_vspace_c_rev**

Description: This command translates the camera along the view space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_camera_vspace_c_rev
```
Command: `trans_camera_vspace_x_for`

Description: This command translates the camera along the view space X-axis, in the positive direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_camera_vspace_x_for
```

Command: `trans_camera_vspace_x_rev`

Description: This command translates the camera along the view space X-axis, in the negative direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

See also: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_camera_vspace_x_rev
```
Command: **trans_camera_vspace_y_for**

Description: This command translates the camera along the view space Y-axis, in the positive direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_vspace_y_for
```

Command: **trans_camera_vspace_y_rev**

Description: This command translates the camera along the view space Y-axis, in the negative direction, by a distance set with the `trans_camera_increment` command.

This command acts on all the currently active views.

Also see: `trans_camera_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_camera_vspace_y_rev
```
Command: trans_camera_vspace_z_for
Description: This command translates the camera along the view space Z-axis, in the positive direction, by a distance set with the trans_camera_increment command.
This command acts on all the currently active views.
Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_camera_vspace_z_for

Command: trans_camera_vspace_z_rev
Description: This command translates the camera along the view space Z-axis, in the negative direction, by a distance set with the trans_camera_increment command.
This command acts on all the currently active views.
Also see: trans_camera_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
trans_camera_vspace_z_rev
Command: **trans_model_cspace_c_for**

**Description:** This command translates the model along the camera space X, Y, and Z-axes, each in the positive direction, and each by a distance set with the `trans_model_increment` command. This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_cspace_c_for
```

---

Command: **trans_model_cspace_c_rev**

**Description:** This command translates the model along the camera space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_model_increment` command. This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_cspace_c_rev
```
Command:  trans_model_cspace_x_for
Description:  This command translates the model along the camera space X-axis, in the positive direction, by a distance set with the trans_model_increment command.
                          
This command acts on all the currently active views.

Also see: trans_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
                    trans_model_cspace_x_for
Another Button with the same description:
                    X+

Command:  trans_model_cspace_x_rev
Description:  This command translates the model along the camera space X-axis, in the negative direction, by a distance set with the trans_model_increment command.
                          
This command acts on all the currently active views.

See also trans_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:
                    trans_model_cspace_x_rev
Another Button with the same description:
                    X-
Command:  trans_model_cspace_y_for
Description:  This command translates the model along the camera space Y-axis, in the positive direction, by a distance set with the trans_model_increment command.

This command acts on all the currently active views.

Also see: trans_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

   trans_model_cspace_y_for

Another Button with the same description:

   Y+

Command:  trans_model_cspace_y_rev
Description:  This command translates the model along the camera space Y-axis, in the negative direction, by a distance set with the trans_model_increment command.

This command acts on all the currently active views.

See also trans_model_increment, activate_view, and the topic viewing for a general discussion of viewing.

Keyboard Command Sequence:

   trans_model_cspace_y_rev

Another Button with the same description:

   Y-
Command: `trans_model_cspace_z_for`
Description: This command translates the model along the camera space Z-axis, in the positive direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:
```
trans_model_cspace_z_for
```

Another Button with the same description:

`Z+`

Command: `trans_model_cspace_z_rev`
Description: This command translates the model along the camera space Z-axis, in the negative direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:
```
trans_model_cspace_z_rev
```

Another Button with the same description:

`Z-`
**Command:** trans_model_increment  

**Description:** This command allows you to set the translation increment angle that is used by subsequent viewing model translation command such as trans_model_cspace_x_for.  

See the topic viewing for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_increment <delta>
```
Command:  `trans_model_mspace_c_rev`

Description: This command translates the model along the model space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_model_mspace_c_rev
```

---

Command:  `trans_model_mspace_x_for`

Description: This command translates the model along the model space X-axis, in the positive direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_model_mspace_x_for
```
Command:  `trans_model_mspace_x_rev`

Description:  This command translates the model along the model space X-axis, in the negative direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

`trans_model_mspace_x_rev`

---

Command:  `trans_model_mspace_y_for`

Description:  This command translates the model along the model space Y-axis, in the positive direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

`trans_model_mspace_y_for`
Command: *trans_model_mspace_y_rev*

Description: This command translates the model along the model space Y-axis, in the negative direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
trans_model_mspace_y_rev
```
Command: \texttt{trans\_model\_mspace\_z\_rev}

Description: This command translates the model along the model space Z-axis, in the negative direction, by a distance set with the \texttt{trans\_model\_increment} command.

This command acts on all the currently active views.

See also \texttt{trans\_model\_increment}, \texttt{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_model\_mspace\_z\_rev}

Command: \texttt{trans\_model\_vspace\_c\_for}

Description: This command translates the model along the view space X, Y, and Z-axes, each in the positive direction, and each by a distance set with the \texttt{trans\_model\_increment} command.

This command acts on all the currently active views.

Also see: \texttt{trans\_model\_increment}, \texttt{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_model\_vspace\_c\_for}
Command: **trans_model_vspace_c_rev**

Description: This command translates the model along the view space X, Y, and Z-axes, each in the negative direction, and each by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```plaintext
trans_model_vspace_c_rev
```

---

Command: **trans_model_vspace_x_for**

Description: This command translates the model along the view space X-axis, in the positive direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic *viewing* for a general discussion of viewing.

**Keyboard Command Sequence:**

```plaintext
trans_model_vspace_x_for
```
Command: \texttt{trans\_model\_vspace\_x\_rev}

Description: This command translates the model along the view space X-axis, in the negative direction, by a distance set with the \texttt{trans\_model\_increment} command.

This command acts on all the currently active views.

Also see: \texttt{trans\_model\_increment}, \texttt{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_model\_vspace\_x\_rev}

---

Command: \texttt{trans\_model\_vspace\_y\_for}

Description: This command translates the model along the view space Y-axis, in the positive direction, by a distance set with the \texttt{trans\_model\_increment} command.

This command acts on all the currently active views.

Also see: \texttt{trans\_model\_increment}, \texttt{activate\_view}, and the topic \textit{viewing} for a general discussion of viewing.

Keyboard Command Sequence:

\texttt{trans\_model\_vspace\_y\_for}
Command: **trans_model_vspace_y_rev**

**Description:** This command translates the model along the view space Y-axis, in the negative direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic **viewing** for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_vspace_y_rev
```

---

Command: **trans_model_vspace_z_for**

**Description:** This command translates the model along the view space Z-axis, in the positive direction, by a distance set with the `trans_model_increment` command.

This command acts on all the currently active views.

Also see: `trans_model_increment`, `activate_view`, and the topic **viewing** for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_vspace_z_for
```
Command: **trans_model_vspace_z_rev**

**Description:** This command translates the model along the view space Z-axis, in the negative direction, by a distance set with the **trans_model_increment** command.

This command acts on all the currently active views.

Also see: **trans_model_increment**, **activate_view**, and the topic **viewing** for a general discussion of viewing.

**Keyboard Command Sequence:**

```
trans_model_vspace_z_rev
```
**RESET**

Command: `transform_reset`

Description: This command resets the current transformation to its default state.

Keyboard Command Sequence:

```
transform_reset
```

**ROTATE**

Command: `transform_rotate`

Description: This command rotates the current transformation about the X, Y, and Z-axes, in that order. You must specify the rotations in degrees.

Keyboard Command Sequence:

```
transform_rotate <X rotation> <Y rotation> <Z rotation>
```

**TRANSFORMS**

Menu: TRANSFORMS

Description: This menu contains commands for defining transformations and applying them to boundary condition application values at nodes.

**ELEMENTS**

Command: `translucent_elements`

Description: This command specifies that the given elements be drawn with translucent color.

Keyboard Command Sequence:

```
translucent_elements <element list> #
```
**SOLIDS**

Command:  `translucent_solids`

Description:  This command specifies that the given solids be drawn with translucent color.

Keyboard Command Sequence:

```
translucent_solids <solids list> #
```

**SOLIDS FACES**

Command:  `translucent_solids_faces`

Description:  This command specifies that the given solids faces be drawn with translucent color.

Keyboard Command Sequence:

```
translucent_solids_faces <solids faces list> #
```

**SURFACES**

Command:  `translucent_surfaces`

Description:  This command specifies that the given surfaces be drawn with translucent color.

Keyboard Command Sequence:

```
translucent_surfaces <surface list> #
```
**TRIM BOX**

Command: **trim_box**

Description: This command places four trimming curves on a surface. The curves are defined in parametric coordinates to be a rectangle parallel to the local coordinate system. The command requires the input of the lower left position, and the width and height of the rectangle.

Keyboard Command Sequence:
```
trim_box <surface> <X> <Y> <W> <H>
```

**TRIM CIRCLE**

Command: **trim_circle**

Description: This command places a trimming curve on a surface. The curve is defined in parametric coordinates to be circular, and requires an origin and radius.

Keyboard Command Sequence:
```
trim_circle <surface> <center x> <center y> <radius>
```

**SURFACE QUAD MESH**

Command: **trim_mesh**

Description: This command automatically creates a mesh on a trimmed surface. This command is similar to the overlay_mesh command but uses the trimming curves of the specified surface as the boundary.

Keyboard Command Sequence:
```
trim_mesh <surface>
```
**TRIM OUTER**

**Command:** trim_outer

**Description:** This command places trimming curves on the edges of surfaces. This is useful when creating trimmed surfaces with holes in them.

**Keyboard Command Sequence:**

```
trim_outer <surface list> #
```

---

**TRIM SURFACES**

**Command:** trim_surfaces

**Description:** This command places 4 trimming curves on the edges of surfaces.

**Keyboard Command Sequence:**

```
trim_surfaces <surface list> #
```

---

**TRSFAC**

**Subroutine:** TRSFAC

**Description:** The user subroutine TRSFAC allows you to define a shift function for a particular thermal rheologically simple viscoelastic material group. TRSFAC is only available in conjunction with the hereditary integral form of viscoelastic constitutive representation.
Mentat Help Commands in U

**Subroutine:** UBEAM

**Description:** The user subroutine UBEAM allows you to define nonlinear elastic cross-section properties as a function of generalized elastic strains and state variables for beam element 52 or beam element 98.

UBEAM is used in conjunction with the hypoelastic option. You must use the HYPOELASTIC model definition option.
**UBEAR**

**Subroutine:** UBEAR  
**Description:** The user subroutine UBEAR requires information about the direction cosines of the lubricant normal.

In bearing analysis, the lubricant is modelled by a planar mesh due to the absence of pressure gradients across the film height. The program will integrate the obtained pressure distribution over the entire mesh. This will yield a set of equivalent consistent nodal forces perpendicular to the lubricant. In order to calculate the load capacity of a particular bearing system, these forces must be transformed to the global coordinate system.

UBEAR is called for each node.

---

**UCOMPL**

**Subroutine:** UCOMPL  
**Description:** The user subroutine UCOMPL allows you to input a real (elastic) and imaginary (damping) stress strain relation for complex harmonic analysis. If not used, only the real portion will be formed in the conventional manner.

UCOMPL is called for all elements, integration points, and layers in a harmonic subincrement.

---

**UCRACK**

**Subroutine:** UCRACK  
**Description:** The user subroutine UCRACK allows the input of a constant or a temperature dependent ultimate stress at each integration point of an element for cracking analysis. In addition, you can define the strain softening modulus and the crushing strain.
Subroutine: UENERG

Description: The user subroutine UENERG allows you to define your own elastic strain energy function for incompressible materials. Normally, the five constant second order model is entered using the MOONEY model definition option. This option must still be used to invoke UENERG.

Subroutine: UFAIL

Description: The user subroutine UFAIL allows you to calculate your own scalar failure criterion. To call UFAIL, you must specify failure criterion type UFAIL in the FAIL DATA model definition option. UFAIL will then be called for every integration point associated with the material id specified in the FAIL DATA option.

Subroutine: UFCONN

Description: The user subroutine UFCONN may be used to modify (or expand) input given through use of the CONNECTIVITY option, or as an internal connectivity generator. You must input the model definition option UFCONN, followed by a line giving the elements for which UFCONN will be used. The program will then call UFCONN for each element in the series, so that the connectivity of that element may be modified or generated. The UFCONN option may be repeated as many times as necessary.
Subroutine: **UFORMS**

**Description:** The user subroutine **UFORMS** allows the definition of a constraint condition. To distinguish user constraints from the program’s built-in constraints, those constraints formed by the user in **UFORMS** must be of type less than zero (**ISTYP** in the subroutine: columns 1-5 of data block 2 of the **TYING** model definition option). The constraint conditions may be supplied by using the subroutine **UFORMS**.

Subroutine: **UFOUR**

**Description:** The user subroutine **UFOUR** allows input of a function \( F(\theta) \) where it can be expressed analytically. The values of \( F(\theta) \) are then passed into a Marc routine that calculates the Fourier expansion coefficients.

Subroutine: **UFRIC**

**Description:** The user subroutine **UFRIC** allows you to define the variable friction coefficients, or friction factors, in conjunction with the model definition option **CONTACT**. Its call is triggered by the option **UFRICTION**.

**UFRIC** is called for every element containing nodes that are in contact with surfaces at the nodes. These calls are made every iteration both during the assembly phase and during the stress recovery phase.
Subroutine: **UFRORD**

**Description:** The user subroutine **UFRORD** may be used to modify (or expand) coordinate change input. You must input the rezoning command **UFRORD**, followed by a command giving a list of nodes for which **UFRORD** will be used. The program will then call **UFRORD** for each node in the list, so that the coordinates for that node may be modified or generated. **UFRORD** may be repeated as many times as necessary.

---

Subroutine: **UFXORD**

**Description:** The user subroutine **UFXORD** may be used to modify (or expand) coordinates input through use of the COORDINATE option, or as an internal coordinate generator. You must input the model definition option **UFXORD**, followed by a line giving the nodes for which **UFXORD** will be used. The program will then call **UFXORD** for each node in the list, so that the coordinates of that node may be modified or generated. The **UFXORD** option may be repeated as many times as necessary.

---

Subroutine: **UGROOV**

**Description:** The user subroutine **UGROOV** allows you to specify position-dependent groove depths which may need to be included at each integration point in bearing analyses. This allows discontinuous film thicknesses to be applied to increase the load carrying capacity. The grooves may be defined in the GEOMETRY option. **UGROOV** may also be used for selecting elements which are located at grooves in the event that complex groove patterns have to be modelled.

**UGROOV** is called for each integration point.
Subroutine: **UHTCOE**

Description: The user subroutine **UHTCOE** allows you to define variable film coefficients and sink temperatures of free surfaces in conjunction with the options CONTACT and COUPLE. Its call is triggered by the option OHTCOE.

**UHTCOE** is called at every element surface containing nodes that are on free body boundary, and for each surface at the trapezoidal rule integration points (i.e. the nodes). These calls are made every iteration both during the assembly phase and the recovery phase of the heat transfer pass of a COUPLED analysis.

Subroutine: **UHTCON**

Description: The user subroutine **UHTCON** allows you to define variable film coefficients of surfaces that are in contact with other surfaces, in conjunction with the options CONTACT and COUPLE. Its call is triggered by the option OHTCON.

**UHTCON** is called at every element surface containing nodes that are on free body boundary, and for each surface at the trapezoidal rule integration points (i.e. the nodes). These calls are made every iteration both during the assembly phase and the recovery phase of the heat transfer pass of a COUPLED analysis.

Subroutine: **UINSTR**

Description: The user subroutine **UINSTR** is called in a loop over all the elements in the mesh when the ISTRESS option is used. Note that this routine is called twice for each point. During the first call, the user-defined stress vector is used to define the net nodal force. During the second call, the user-defined stress vector is used to define the initial stress at each point.
**UMOONY**

**Subroutine:** UMOONY  
**Description:** The user subroutine UMOONY allows you to redefine the constants used in the strain energy function. This data is normally entered through the MOONEY model definition option.

**UNALIAS**

**Command:** unalias  
**Description:** This command deletes an alias from the list of aliases.  
**Keyboard Command Sequence:** 
unalias <alias name>

**UNDEFINE**

**Command:** undefine  
**Description:** This command deletes a parameter from the list of parameters.  
**Keyboard Command Sequence:** 
undefine <parameter name>

**FULL**

**Command:** undeformed_edges_full  
**Description:** This command specifies that all edges on the outline of the visible portion of undeformed model be displayed.  
**Keyboard Command Sequence:** 
undeformed_edges_full
**OUTLINE**

Command: **undeformed_edges_outline**

Description: This command specifies that only edges on the outline of the visible portion of undeformed model be displayed.

Keyboard Command Sequence:

```
undeformed_edges_outline
```

---

**SURFACE**

Command: **undeformed_edges_surface**

Description: This command specifies that only edges on the surface of the visible portion of undeformed model be displayed.

Keyboard Command Sequence:

```
undeformed_edges_surface
```

---

**UNDO**

Command: **UNDO**

Description: The command UNDO allows you to cancel the last change made to your model. UNDO works only for actual changes to the model, not for changes made to your view of the model (i.e. UNDO will restore any deleted elements, but will not cancel any rotations or pans).
**UNEWTN**

**Subroutine:** UNEWTN  
**Description:** The user subroutine UNEWTN allows you to define the viscosity at a particular spatial location when the R-P FLOW option is used to solve Newtonian and Non-Newtonian laminar incompressible fluid analyses. An Eulerian approach is then used to solve for the nodal velocities.

---

**UNMAP TRIMMING CURVES**

**Command:** unmap_curves  
**Description:** This command unmaps trimming curves from a chosen surface back to 2-D space. It can help users clean trimming curves manually.  
**Keyboard Command Sequence:**
```
unmap_curves <surface list>
```

---

**UNORST**

**Subroutine:** UNORST  
**Description:** The user subroutine UNORST allows you to define the normal stress at each node in contact instead of using the extrapolated value from the integration points.  

UNORST is only called for user defined elements and considers the effects of Coulomb friction for CONTACT analysis. The magnitude of the user-defined normal stress must be in the local system of the patch with which the nodal point is in contact.
Command:  unpost_elements
Description:  This command indicates which elements should not be post plotted when a post plotting option is turned on.

Keyboard Command Sequence:

        unpost_elements <element list>

Command:  unpost_nodes
Description:  This command indicates which nodes should not be post plotted when a post plotting option is turned on (for Vectors, Symbols, and Numerics). Normally this is controlled by the unpost_elements command, however, this command allows individual nodes to be controlled. By default, all nodes have been ADDed. Therefore, to view specific nodes use the REM button to remove the nodes that are not to be displayed.

Note that once a node has been removed, it cannot be selected. Therefore, using the ADD button will require that the node numbers be typed in.

Keyboard Command Sequence:

        unpost_nodes <node list>

Command:  untransform_nodes
Description:  This command clears the transform for the specified list of nodes.

Keyboard Command Sequence:

        untransform_nodes <node list> #
Description: This button scrolls the list upward.

**update_axito3d**

Description: This option allows the `axito3d` to update load table curves. For applied load, the origin of the table curve coordinate is shifted along the time axis; for displacement boundary condition, the origin is shifted along the curve.

**update_job**

Command: update_job

Description: This command updates the status of the current job.

Keyboard Command Sequence:

```
update_job
```

**UPDNOD**

Subroutine: UPDNOD

Description: The user subroutine `UPDNOD` is used in conjunction with Eulerian flow solutions (e.g., R-P FLOW) to update the mesh after a velocity field has been found. You may access the velocity field and hence redefine the nodal coordinates.

`UPDNOD` is called in a loop over all the nodes in the mesh at the end of the convergent step of the flow calculation.
**Subroutine: UPHI**

**Description:** The user subroutine **UPHI** allows you to input PHI functions where they can be expressed analytically. The values of PHI are then passed into a routine where they are used in calculation of the Laplace transform for harmonic analysis.

---

**Subroutine: URESTR**

**Description:** The user subroutine **URESTR** allows you to include nonuniform restrictor coefficients and pump pressures, and to modify the restrictor coefficient and pump pressure previously input in the data file. **URESTR** is called at each increment for each integration point.

---

**Subroutine: USELEM**

**Description:** The user subroutine **USELEM** allows you to calculate your own finite element stiffness or mass matrix. **USELEM** may also be used as interface with other numerical techniques. **USELEM** is called multiple times for a particular element, since several matrices are generally required in the finite element calculation.
Subroutine: **CREDE**

Description: The user subroutine **CREDE** allows you to input prespecified state variables. The simplest option allows you to specify temperature increments throughout the mesh. Through the use of the STATE VARS parameter line, the number of state variables per point in the structure may be increased. For example, radiation fluxes (in reactor core problems), pore pressures (in soil mechanics) may be included. The program always assumes temperature is the first state variable given at a particular point or since the first state variable is used in conjunction with the tables of temperature dependence input specified in the TEMPERATURE EFFECTS option, and the first state variable is used to compute thermal strains. All state variables are available to all constitutive routines – see subroutines CRPLAW and VSWELL later in this section.

Subroutine **CREDE** will be called once per element in a loop over the elements when the THERMAL LOADS option is activated.

Subroutine: **CRPLAW**

Description: For creep analysis, the user subroutine **CRPLAW** allows you to specify the creep strain rate. **CRPLAW** will be called as required during the analysis.

For visco-plasticity analysis, the user subroutine **CRPLAW** may also be used for calculating the visco-plastic behavior. All the creep quantities are treated as viscoplastic strain quantities when the appropriate flag is set in the CREEP option.
Subroutine: CRPVIS
Description: The user subroutine CRPVIS allows you to include a general Kelvin model (in addition to the nonlinear Maxwell-type model allowed in the CREEP option).

Subroutine: CUPFLX
Description: The user subroutine CUPFLX allows you to modify the default routine for the calculation of the internal heat generated due to inelastic energy dissipation. This routine is only used if a coupled thermal-mechanical analysis is being performed and a distributed FLUX type 101 is chosen.

Subroutine: FILM
Description: The user subroutine FILM allows you to include nonuniform film coefficients and sink temperatures for the calculation of convection or radiation boundary conditions.

FILM is called at each time step for each integration point on each element surface given in the FILMS model definition set, and allows you to modify the film coefficient and sink temperature.

Subroutine: FLOW
Description: The user subroutine FLOW allows you to modify mass flow rate, inlet temperature, and film coefficient in heat transfer analyses involving fluid channel elements. Both the inlet temperature and mass flow rate can be dependent on time; the film coefficient can also be a function of streamline distance.
Subroutine: **FLUX**  
**Description:** The user subroutine **FLUX** allows you to specify surface or body fluxes as functions of time, temperature, or position for heat transfer analysis. The use of this subroutine is flagged by the appropriate flux type in the DIST FLUXES or FLUXES input option where the type chosen depends on element type.

---

Subroutine: **FORCDF**  
**Description:** The user subroutine **FORCDF** is used to input frequency dependent point loads or displacement histories. (Use user subroutine **FORCEM** for distributed loads.)

**FORCDF** is flagged by introducing a model definition set (**FORCDF**) listing the node numbers for which this routine is called. Then, at each harmonic increment of the analysis, for each of the nodes on the list, the subroutine is called. Displacement and load arrays are available for stress analysis. For dynamic analysis, velocity, and acceleration are available. For nodes without kinematic boundary conditions, you may define increments of point loads (thus overwriting any point load input in the same nodes in the point loads block). For nodes with kinematic boundary conditions (i.e. listed in the **FIXED DISP** or **DISP CHANGE** options) you may define increments of harmonic displacement.
**FORCDT**

**Subroutine:** FORCDT

**Description:** The user subroutine FORCDT is used to input time dependent point loads or displacement histories. (User subroutine FORCEM is used for distributed loads.)

FORCDT is flagged by introducing a model definition set (FORCDT) listing the node numbers for which this routine is called. Then, at each increment of the analysis, for each of the nodes on the list, the subroutine is called. Displacement and load arrays are available for stress analysis. For dynamic analysis, velocity, and acceleration are available. For nodes without kinematic boundary conditions, you may define increments of point loads (thus overwriting any point load input in the same nodes in the point loads block). For nodes with kinematic boundary conditions (i.e. listed in the FIXED DISP or DISP CHANGE options) you may define increments of displacement.

**FORCEM**

**Subroutine:** FORCEM

**Description:** The user subroutine FORCEM is used to input nonuniform distributed loads. This routine may be used to specify the load magnitude as a function of position and/or time.

FORCEM is called during the calculation of the equivalent nodal loads, at each integration point needed to calculate the loads specified in the DIST LOADS. If data blocks are read by the user’s FORCEM, they should therefore appear after the corresponding END OPTION or CONTINUE options in the input deck.
GAPT

Subroutine: GAPT

Description: The user subroutine GAPT allows you to redefine gap temperature (TGAP) based on the nodal temperatures T1 and T2 in conjunction with the CONRAD GAP option. If the gap temperature (TGAP) is greater than or equal to the gap closure temperature (TCLOSE), then the gap is closed. Otherwise the gap is open.

GAPU

Subroutine: GAPU

Description: The user subroutine GAPU allows you to input or modify the direction and closure distance of gap element type 12 and 97 based on the current position of the end nodes of the element. This makes it possible to model contact sliding along curved surfaces which may occur in the analysis of metal forming problems. Note that although the gap direction and closing distance can be changed, GAPU does not allow for finite sliding of two meshes with respect to each other, since the load transfer path is unchanged. In addition, GAPU allows for specification of a nonlinear relationship between the normal force and the maximum friction force instead of the regular linear Coulomb relation.

Note: If this option is used to change the direction of the gap, then friction should NOT be included.

GAPU also allows you to specify certain tolerances to control gap closure and friction iterations. Note that this last feature will generally not be used.
Genstr

Subroutine: **GENSTR**

**Description:** The user subroutine **GENSTR** allows you to enter the generalized stress-strain law for shells and beams which are conventionally integrated through their thickness. This is often convenient in composite analysis, where the experimental information is for the total material, not individual plies. **GENSTR** is activated using the SHELL SECT parameter lines. As no layer integration is performed, the number of layers may be set to one.

Hooklw

Subroutine: **HOOKLW**

**Description:** The user subroutine **HOOKLW** is an alternative mechanism to user subroutine **ANELAS**. In this routine, you supply the elastic stress-strain law. This law is given in terms of the coordinate system defined in the ORIENTATION option. You should insure that the stress-strain law is symmetric. Note that this routine will be called for each integration point of those elements that have anisotropic properties. You may define either the stress-strain relation or the compliance strain-stress relation. The returned value of argument IMOD must be set accordingly.

Hookvi

Subroutine: **HOOKVI**

**Description:** The user subroutine **HOOKVI** allows you to modify the time dependent properties of an orthotropic material specified through the VISCELORTH model definition option.

**HOOKVI** is automatically called for every material defined in that option.
**HYPELA**

**Subroutine:** HYPELA

**Description:** The user subroutine HYPELA allows you to define elastic constants as functions of elastic strain and state variables.

**INITPL**

**Subroutine:** INITPL

**Description:** The user subroutine INITPL is called in a loop over all the elements in the mesh when the INITIAL PLASTIC option appears in the model definition options with a 2 in column 10 of the second data blocks of that option. It is often necessary to enter the amount of previously accumulated plastic strain. This initial value is only used in the work (strain) hardening calculation.

**INITSV**

**Subroutine:** INITSV

**Description:** The user subroutine INITSV allows you to define initial values of state variables. This routine will be called in a loop over all the elements in the mesh when the INITIAL STATE option (appears in the model definition options with a 2 in column 10 of the second data blocks of that option).

**INTCRD**

**Subroutine:** INTCRD

**Description:** The user subroutine INTCRD makes the integration point coordinates for the stiffness matrix available at each increment. You may save them in any form convenient for your postprocessing.
Subroutine: MOTION

Description: The user subroutine MOTION allows you to define nonuniform rigid surface motions, in conjunction with the option CONTACT. Its call is triggered by the model definition option UMOTION.

User subroutine MOTION is called during the calculations at the beginning of each time increment, and you return surface velocities for that increment. Imposed displacement increments at nodal points in contact with rigid surfaces are obtained from the velocity multiplied by the time increment. The surface path becomes an explicit forward integration of velocities. Therefore, caution should be used when there are abrupt changes in surface path direction or abrupt changes in velocity, by making time increments as small as necessary.

If, at the start of the analysis, a surface is placed apart from the body to be deformed, subroutine MOTION will also be used in the approaching phase, in time increments of 0.1 seconds.

Subroutine: NASSOC

Description: The user subroutine NASSOC allows you to calculate a strain increment using a flow rule differing from the normality rule of plasticity, which is the default used by the program.

Subroutine: NEWSV

Description: The user subroutine NEWSV allows you to define the new values of any state variable for the end of the current step. This subroutine will be called in a loop over all the elements in the mesh when requested through the CHANGE STATE option.
**Subroutine: ORIENT**

**Description:** The user subroutine **ORIENT** allows you to supply a preferred orientation so that **ANELAS**, **HOOKLW**, **ANPLAS**, and **ANKOND** may supply anisotropic material constants in this orientation.

---

**Subroutine: SEPFOR**

**Description:** The user subroutine **SEPFOR** allows you to define the separation force in conjunction with the model definition option **CONTACT**. The separation forces, **FNORM** and **FTANG**, are either calculated by the program or entered through the **CONTACT** option, and are then passed into this subroutine. You decide whether this magnitude at the current increment is good enough to determine whether separation will occur.

**FNORM** is the normal reaction force above which a node in contact will separate from a surface. Any compressive or negative value indicates real contact, while a positive reaction force indicates a tendency to separate. The default is taken as the maximum value of the residual force in the structure for the current increment. This value can be reset in the input deck. Defining a small value may result in an increased number of iterations. Defining a very large value will eliminate the possibility of separation.

**FTANG** is the tangential force used to determine whether a nodal point positioned at a convex corner of a surface should be sliding from patch to patch or remain on its current patch. The default value is half of **FNORM**. These two default reaction forces vary from increment to increment.
Subroutine: SEPSTR

Description: The user subroutine SEPSTR allows you to define the separation stress in conjunction with the model definition option CONTACT. The separation stresses, SNORM and STANG, are either calculated by the program or entered through the CONTACT option, and are then passed into this subroutine. You decide whether this magnitude at the current increment is good enough to determine whether separation will occur.

SNORM is the stress normal to the surface above which a node in contact will separate from another body. Any compressive or negative value indicates real contact, while a positive reaction stress indicates a tendency to separate. The default is taken as the maximum value of the residual force in the structure for the current increment divided by an effective area. This value can be reset in the input line. Defining a too small value can result in an increased number of iterations. Defining a very large value will eliminate the possibility of separation.

STANG is the tangential stress used to determine whether a nodal point positioned at a convex corner of a surface should be sliding from patch to patch or remain on its current patch. The default value is half of SNORM. These two default values vary from increment to increment.

Subroutine: TRSFAC

Description: The user subroutine TRSFAC allows you to define a shift function for a particular thermal rheologically simple viscoelastic material group.

TRSFAC is only available in conjunction with the hereditary integral form of viscoelastic constitutive representation.
**UBEAM**

**Subroutine:** UBEAM

**Description:** The user subroutine UBEAM allows you to define nonlinear elastic cross-section properties as a function of generalized elastic strains and state variables for beam element 52 or beam element 98.

UBEAM is used in conjunction with the hypoelastic option. You must use the HYPOELASTIC model definition option.

---

**UBEAR**

**Subroutine:** UBEAR

**Description:** The user subroutine UBEAR requires information about the direction cosines of the lubricant normal.

In bearing analysis, the lubricant is modelled by a planar mesh due to the absence of pressure gradients across the film height. The program will integrate the obtained pressure distribution over the entire mesh. This will yield a set of equivalent consistent nodal forces perpendicular to the lubricant. In order to calculate the load capacity of a particular bearing system, these forces must be transformed to the global coordinate system.

UBEAR is called for each node.

---

**UCOMPL**

**Subroutine:** UCOMPL

**Description:** The user subroutine UCOMPL allows you to input a real (elastic) and imaginary (damping) stress strain relation for complex harmonic analysis. If not used, only the real portion will be formed in the conventional manner.

UCOMPL is called for all elements, integration points, and layers in a harmonic subincrement.
Subroutine: UCONTACT
Description: The user subroutine UCONTACT allows you to calculate your own contact conditions with rigid surfaces based upon the Marc contact algorithm. This subroutine requires the use of the UCONTACT model definition option and is triggered by this option.

In general, the four stages in Marc contact procedures are:

– checking contact
– enforce contact constraint
– check separation
– check penetration.

You can either specify contact conditions at one stage and let the Marc program do the work in the remainder of stages or you can substitute your calculations for all stages.

Subroutine: UCRACK
Description: The user subroutine UCRACK allows the input of a constant or a temperature dependent ultimate stress at each integration point of an element for cracking analysis. In addition, you can define the strain softening modulus and the crushing strain.

Subroutine: UENERG
Description: The user subroutine UENERG allows you to define your own elastic strain energy function for incompressible materials. Normally, the five constant second order model is entered using the MOONEY model definition option. This option must still be used to invoke UENERG.
**Subroutine: UFAIL**

**Description:** The user subroutine **UFAIL** allows you to calculate your own scalar failure criterion. To call **UFAIL**, you must specify failure criterion type **UFAIL** in the **FAIL DATA** model definition option. **UFAIL** will then be called for every integration point associated with the material id specified in the **FAIL DATA** option.

---

**Subroutine: UFCCONN**

**Description:** The user subroutine **UFCCONN** may be used to modify (or expand) input given through use of the **CONNECTIVITY** option, or as an internal connectivity generator. You must input the model definition option **UFCCONN**, followed by a line giving the elements for which **UFCCONN** will be used. The program will then call **UFCCONN** for each element in the series, so that the connectivity of that element may be modified or generated. The **UFCCONN** option may be repeated as many times as necessary.

---

**Subroutine: UFORMS**

**Description:** The user subroutine **UFORMS** allows the definition of a constraint condition. To distinguish user constraints from the program’s built-in constraints, those constraints formed by the user in **UFORMS** must be of type less than zero (**ISTYP** in the subroutine: columns 1-5 of data block 2 of the **TYING** model definition option). The constraint conditions may be supplied by using the subroutine **UFORMS**.
Subroutine:  **UFOUR**

**Description:** The user subroutine **UFOUR** allows input of a function $F(\theta)$ where it can be expressed analytically. The values of $F(\theta)$ are then passed into a Marc routine that calculates the Fourier expansion coefficients.

Subroutine:  **UFRIC**

**Description:** The user subroutine **UFRIC** allows you to define the variable friction coefficients, or friction factors, in conjunction with the model definition option **CONTACT**. Its call is triggered by the option **UFRICTION**. **UFRIC** is called for every element containing nodes that are in contact with surfaces at the nodes. These calls are made every iteration both during the assembly phase and during the stress recovery phase.

Subroutine:  **UFRORD**

**Description:** The user subroutine **UFRORD** may be used to modify (or expand) coordinate change input. You must input the rezoning command **UFRORD**, followed by a command giving a list of nodes for which **UFRORD** will be used. The program will then call **UFRORD** for each node in the list, so that the coordinates for that node may be modified or generated. **UFRORD** may be repeated as many times as necessary.
**USER SUBROUTINE UFXORD**

**Subroutine:** UFXORD  
**Description:** The user subroutine UFXORD may be used to modify (or expand) coordinates input through use of the COORDINATE option, or as an internal coordinate generator. You must input the model definition option UFXORD, followed by a line giving the nodes for which UFXORD will be used. The program will then call UFXORD for each node in the list, so that the coordinates of that node may be modified or generated. The UFXORD option may be repeated as many times as necessary.

**UGROOV**

**Subroutine:** UGROOV  
**Description:** The user subroutine UGROOV allows you to specify position-dependent groove depths which may need to be included at each integration point in bearing analyses. This allows discontinuous film thicknesses to be applied to increase the load carrying capacity. The grooves may be defined in the GEOMETRY option. UGROOV may also be used for selecting elements which are located at grooves in the event that complex groove patterns have to be modelled.  

UGROOV is called for each integration point.

**UHTCOE**

**Subroutine:** UHTCOE  
**Description:** The user subroutine UHTCOE allows you to define variable film coefficients and sink temperatures of free surfaces in conjunction with the options CONTACT and COUPLE. Its call is triggered by this button.  

UHTCOE is called at every element surface containing nodes that are on free body boundary, and for each surface at the trapezoidal rule integration points (i.e. the nodes). These calls are made every iteration, both during the assembly phase and the recovery phase of the heat transfer pass of a COUPLED analysis.
Subroutine: UHTCON
Description: The user subroutine UHTCON allows you to define variable film coefficients of surfaces that are in contact with other surfaces, in conjunction with the options CONTACT and COUPLE. Its call is triggered by this button.

UHTCON is called at every element surface containing nodes that are on free body boundary, and for each surface at the trapezoidal rule integration points (i.e. the nodes). These calls are made every iteration, both during the assembly phase and the recovery phase of the heat transfer pass of a COUPLED analysis.

Subroutine: UINSTR
Description: The user subroutine UINSTR is called in a loop over all the elements in the mesh when the ISTRESS option is used. Note that this routine is called twice for each point. During the first call, the user-defined stress vector is used to define the net nodal force. During the second call, the user-defined stress vector is used to define the initial stress at each point.

Subroutine: UMOONY
Description: The user subroutine UMOONY allows you to redefine the constants used in the strain energy function. This data is normally entered through the MOONEY model definition option.
Subroutine: **UNEWTN**

Description: The user subroutine **UNEWTN** allows you to define the viscosity at a particular spatial location when the R-P FLOW option is used to solve Newtonian and Non-Newtonian laminar incompressible fluid analyses. An Eularian approach is then used to solve for the nodal velocities.

Subroutine: **UNORST**

Description: The user subroutine **UNORST** allows you to define the normal stress at each node in contact instead of using the extrapolated value from the integration points.

**UNORST** is only called for user defined elements and considers the effects of Coulomb friction for CONTACT analysis. The magnitude of the user-defined normal stress must be in the local system of the patch with which the nodal point is in contact.

Subroutine: **UPDNOD**

Description: The user subroutine **UPDNOD** is used in conjunction with Eulerian flow solutions (e.g. R-P FLOW) to update the mesh after a velocity field has been found. You may access the velocity field and hence redefine the nodal coordinates.

**UPDNOD** is called in a loop over all the nodes in the mesh at the end of the convergent step of the flow calculation.
**Subroutine: UPHI**

**Description:** The user subroutine UPHI allows you to input PHI functions where they can be expressed analytically. The values of PHI are then passed into a routine where they are used in calculation of the Laplace transform for harmonic analysis.

**Subroutine: URESTR**

**Description:** The user subroutine URESTR allows you to include nonuniform restrictor coefficients and pump pressures, and to modify the restrictor coefficient and pump pressure previously input in the data file. URESTR is called at each increment for each integration point.

**Subroutine: USELEM**

**Description:** The user subroutine USELEM allows you to calculate your own finite element stiffness or mass matrix. USELEM may also be used as interface with other numerical techniques. USELEM is called multiple times for a particular element, since several matrices are generally required in the finite element calculation.

**Subroutine: VSWELL**

**Description:** The user subroutine VSWELL allows you to include pure swelling (dilatational) creep in Marc.
Subroutine: **WKSLP**

**Description:** The user subroutine **WKSLP** makes it possible for you to program the yield stress and the corresponding workhardening slope directly as a function of equivalent plastic strain and temperatures. See WORK HARD model definition option. You need to define the value of the slope of the equivalent stress vs. equivalent plastic strain. The current yield stress may be defined also. The specification of the latter is optional. If the value of the current yield is not given here, the program will calculate it from the initial yield value and the workhardening slopes defined in this routine.

Subroutine: **YIEL**

**Description:** The user subroutine **YIEL** allows you to define the yield stress based on the current workhardening and other state variables.

Subroutine: **ZERO**

**Description:** The user subroutine **ZERO** allows you to calculate the equivalent yield stress based on the current total stresses. The function **ZERO** in the program applies the von Mises yield criterion as a default. You may substitute another yield criterion by writing a new function **ZERO**. The MOHRC option should not be used when a new function **ZERO** is used because of the danger of taking into account the effects of hydrostatic pressure twice.
**USHELL**

Subroutine: **USHELL**

Description: The user subroutine **USHELL** allows the specification of the thickness of shell elements for each integration point. **USHELL** is called twice for each increment of analysis. It is not advised to change the thickness during an analysis (i.e. after **RESTART**).

**USHRET**

Subroutine: **USHRET**

Description: The user subroutine **USHRET** allows the definition of the residual shear stiffness for a cracked integration point in a cracking analysis. The factor is defined as the factor with which the initial shear stiffness is multiplied. With this routine, you can define the shear retention factor to be, for instance, a function of the crack strain. **USHRET** is automatically called for each existing crack.

**USINC**

Subroutine: **USINC**

Description: The user subroutine **USINC** allows you to input initial displacements and velocities for stress analysis and initial temperatures for heat transfer analysis. You give the values for all degrees of freedom in vector F. This routine is used with either the INITIAL DISPLACEMENTS, INITIAL VELOCITY, or the INITIAL TEMPERATURE option. This routine will be called for every node in the structure if it is used.
**USPRNG**

**Subroutine:** USPRNG  
**Description:** The user subroutine USPRNG allows you to introduce nonlinear spring constants for use with the SPRING and FOUNDATION options, and input of nonlinear damping if the dashpot option is to be used. Your coding must supply both the ratio of the current value of spring stiffness to the data input value and the total spring force.

For dynamic analyses, the ratio of damping coefficient can also be provided. The data value of the spring/dashpot constant, total time, and the element or spring number are made available to the subroutine.

For harmonic analyses, the spring/dashpot constants can be a function of the frequency. USPRNG is accessible whenever either the SPRING or the FOUNDATION option is used.

**USSD**

**Subroutine:** USSD  
**Description:** The user subroutine USSD allows you to input the spectral density function for the frequencies required in the spectrum response calculation. These frequencies will be obtained by performing a modal analysis.

**UTHICK**

**Subroutine:** UTHICK  
**Description:** The user subroutine UTHICK allows you to define, or to redefine, previously specified nodal thicknesses. In addition, UTHICK may be used to defined thickness increments in incremental analysis or within subincrements when evaluating damping and/or stiffness coefficients. In order to enable the specification of thickness increments as a function of previously calculated bearing properties, you have access to the latter quantities in this routine.

UTHICK is called for each node in the mesh.
Main Menu: UTILS

Description: This menu contains utility Mentat commands.

---

Subroutine: UTRANS

Description: The user subroutine UTRANS allows the specification of a local coordinate system for user-specified nodes. The node numbers are given in the UTRANFORM model definition option.

UTRANS is called multiple times for each increment of analysis. The local coordinate system can be modified (updated) at each increment to facilitate the input of complex boundary conditions. Incremental nodal displacements and reaction forces will be output in both the local and global coordinate systems. All total nodal quantities will be output in the global system.

---

Subroutine: UVELOC

Description: The user subroutine UVELOC allows you to specify or redefine previously specified nodal velocity vectors.

UVELOC is called for each node.
Mentat Help Commands in V

**Command:** version

**Description:** This command prints the current version name of the program.

**Keyboard Command Sequence:**

```
version
```

**Description:** These commands perform the computation of viewfactors for radiation boundary conditions. This computation must be performed before an analysis job is submitted which involves the effects of radiation heat transfer. A file is written containing the viewfactors for this purpose.

The computation of viewfactors is implemented using the Monte Carlo scheme. Rays are emitted randomly from each surface and intersected with every other radiating surface, finding the first intersecting surface for
each ray. The viewfactor for each incident surface is simply the fraction of emitted rays which have hit that surface.

**Commands:**

```markdown
vfc_sym_activate <plane>
vfc_sym_deactivate <plane>
vfc_sym_point <plane> <x y z>
vfc_sym_normal <plane> <x y z>
```

Three symmetry planes, defined by a position and a normal, are provided. Each plane may be individually activated and deactivated.

```markdown
vfc_type_2d
vfc_type_3d
vfc_type_ax
```

The geometry may be interpreted as being two-dimensional, three-dimensional, or axisymmetric. For the two-dimensional and axisymmetric cases, only the element edges which are part of a radiation boundary condition are used in the computation. For three dimensions, only element faces are used.

```markdown
vfc_nray
```

This command controls the number or rays emitted by each object. A larger number of emitted rays gives a more accurate estimate of the viewfactor.

```markdown
vfc_start
vfc_stop
vfc_reset
```

These commands control the computation itself. Obviously, `vfc_start` begins or restarts the computation and `vfc_stop` halts the computation. `vfc_reset` stops the computation and resets it to the beginning, throwing away all previously computed results. A reset is automatically done whenever any of the above commands is issued.

```markdown
vfc_file
```

This command sets the name of the file where the results are to be placed. This must be set before the beginning of viewfactor computation. The format of the file is:

**Block 1 – Header**

Line 1

10 int iver Version #
10 int nobj Number of objects
10 int nray Number of rays used in computation

**Block 2 – Objects**
Line 1 repeated nobj times
10 int obj Object number
10 int eid Element id
10 int face Face or edge number
15 float tinf Temperature at infinity top
15 float tinf Temperature at infinity bottom

**Block 3 – View Factors repeated nobj times**
Line 1
10 int obj Emitting object number
10 int nz Number of nonzero viewfactors
Line 2 repeated nz times
10 int obj Incident object number
15 float vfs[4] Four viewfactors

<table>
<thead>
<tr>
<th>Emit</th>
<th>Incident</th>
</tr>
</thead>
<tbody>
<tr>
<td>out</td>
<td>out</td>
</tr>
<tr>
<td>out</td>
<td>in</td>
</tr>
<tr>
<td>in</td>
<td>out</td>
</tr>
<tr>
<td>in</td>
<td>in</td>
</tr>
</tbody>
</table>

where: out – outer normal of element according to connectivity
in – the other side

**Notes:** For line elements out means the right hand side as you travel from node 1 to node 2.
For shells out is defined by the right hand rule for the connectivity of the nodes. This is the same as the front side in Mentat.
LOOKAT POINT DISTANCE

Command: `view_lookat_distance`

Description: For all active views, this command moves the lookat point to a specified distance from the viewpoint. The lookat point is moved towards or away from the viewpoint along the line which connects them.

This command is useful for changing the depth of the lookat point, so that dynamic rotations will take place about a specified depth. You can use the left mouse button to pick an item which is at the desired depth from your viewpoint.

Keyboard Command Sequence:

```
view_lookat_distance <distance>
view_lookat_distance <ML>
```

SET ANGLES

Command: `view_model_angles`

Description: This command sets absolutely the viewing rotation angles for the model, while leaving the viewing model scale and translations alone. All camera settings remain unchanged by this command. You must specify separate X, Y, and Z-rotation angles in degrees.

First, the model is realigned with the view space axes, thus establishing absolutely no rotation for the model. Then the model is rotated about the view space X-axis by the specified X-angle, then rotated about the view space Y-axis by the specified Y-angle, then rotated about the view space Z-axis by the specified Z-angle.

This command acts on all the currently active views.

Also see: `activate_view`, `view_scale_factor`, `view_model_translation`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
view_model_angles <x angle> <y angle> <z angle>
```
**SET TRANSLATION**

Command:  `view_model_translation`

Description: This command sets absolutely the viewing translation for the model, while leaving the viewing model scale and rotations alone. All camera settings remain unchanged by this command. You must specify the absolute translation X, Y, and Z in view space from the view space origin. Any preexisting viewing translation is replaced by the given translation. This command acts on all the currently active views.

Also see: `activate_view`, `view_scale_factor`, `view_model_angles`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
view_model_translation <x> <y> <z>
```

**ORTHOGRAPHIC**

Command:  `view_orthographic`

Description: This command specifies that all active views be displayed with orthographic projection. With orthographic projection, lines that are parallel in the model will appear parallel in the view. Views can be made to be shown in perspective with the `view_perspective` command.

Views are activated and deactivated with the commands `activate_view`, `deactivate_view`, and `activate_all_views`.

Keyboard Command Sequence:

```
view_orthographic
```
**PERSPECTIVE**

Command: `view_perspective`

Description: This command specifies that all active views be displayed with perspective projection. With perspective projection, lines that are parallel in the model will appear to converge towards a point at infinity.

Views are activated and deactivated with the commands `activate_view`, `deactivate_view`, and `activate_all_views`.

Keyboard Command Sequence:

`view_perspective`

---

**SCALE FACTOR**

Command: `view_scale_factor`

Description: This command sets the current viewing scale setting for the model. The `scale_model_up` and `scale_model_down` commands will increase or decrease this setting.

This command acts on all the currently active views.

Also see: `activate_view`, `view_scale_factor`, `view_model_angles`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

`view_scale_factor <value>`

---

**VIEW SETTINGS**

Menu: SETTINGS

Description: This menu contains commands for setting the rotation, translation, and zoom increments.
**VIEWPOINT DISTANCE**

Command: `view_viewpoint_distance`

Description: For all active views, this command moves the viewpoint to a specified distance from the lookat point. The viewpoint is moved towards or away from the lookat point along the line which connects them.

You can use the left mouse button to pick an item which is at the desired depth from your lookat point.

Keyboard Command Sequence:

```
view_viewpoint_distance <distance>
view_viewpoint_distance <ML>
```

**ZOOM FACTOR**

Command: `view_zoom_factor`

Description: This command sets the current viewing scale setting for the model. The `zoom_model_in` and `zoom_model_out` commands will increase or decrease this setting.

This command acts on all the currently active views.

Also see: `activate_view`, `view_zoom_factor`, `view_model_angles`, and the topic `viewing` for a general discussion of viewing.

Keyboard Command Sequence:

```
view_zoom_factor <value>
```
**VIEW**

**Menu:** VIEW

**Description:** This menu contains commands for changing the view of the model. The viewing commands refer to three different spaces called model space, view space, and camera space.

*View space* can be thought of as a room containing the model, a virtual camera, and the lights used in lighting and rendering.

*Model space* is the database space of the model, with the origin always at the model origin, and its axes always aligned with the model’s axes.

*Camera space* is the space of the graphics window, with its origin at the viewpoint, and Y pointing up, X pointing to the right, and Z pointing out of the screen.

Certain viewing commands move the model around the room (view space), while other commands move the camera around. All the lighting commands which control the location of lights are also referring to view space.

**Command:** `visible_all`

**Description:** This command makes the all nodes, elements, points, curves, surfaces, and solids visible.

**Keyboard Command Sequence:**

```
visible_all
```

**Command:** `visible_curves`

**Description:** This command causes the specified curves to be made visible. Curves can be made invisible with the `invisible_curves` command.

**Keyboard Command Sequence:**

```
visible_curves <curve list> #
```
ELEMENTS

Command:  **visible_elements**
Description:  This command causes the specified elements to be made visible. Elements can be made invisible with the **invisible_elements** command.

Keyboard Command Sequence:

```
visible_elements
```

MAKE VISIBLE

Command:  **visible_selected**
Description:  This command causes the currently selected entities to be made visible. Selected entities can be made invisible with the **invisible_selected** command.

Keyboard Command Sequence:

```
visible_selected
```

SOLIDS

Command:  **visible_solids**
Description:  This command causes the specified solids to be made visible. Solids can be made invisible with the **invisible_solids** command.

Keyboard Command Sequence:

```
visible_solids <solid list> #
```
**SURFACES**

**Command:** visible_surfaces

**Description:** This command causes the specified surfaces to be made visible. Surfaces can be made invisible with the invisible_surfaces command.

**Keyboard Command Sequence:**
```
visible_surfaces <surface list> #
```

---

**VISUALIZATION**

Menu: VISUALIZATION

**Description:** This menu contains menus for PLOT, COLORS, VIEW, LIGHTING, and RENDER settings.

**PLOT**
Contains commands for plotting nodes, elements, points, curves, surfaces, solids, cavities, boundary conditions, initial conditions, links, rbe2’s, rbe3’s, material orientations, and loadcases. Elements, surfaces, and solids may be plotted in wireframe or solid mode.

Visibility, opaqueness, and translucency of entities may be specified. The legend toggled on or off, and annotations may be specified at user defined locations.

The identification of backfaces, element classes and types, boundary conditions, initial conditions, inserts, materials, geometric properties, contact bodies, global and local remeshing bodies, domains and sets may be selected.

**COLORS**
The colors for the background, graphs, annotations, contour lines, grid, points, curves, surfaces, elements, backfaces, boundary conditions, selected entities and attached entities may be specified.

**VIEW**
This menu contains commands for setting the camera view options, model view options, displaying views 1-4 or showing all views. The view repeat mode, zooming in/out, scaling up/down, and viewing in either orthographic or perspective mode may be specified.
LIGHTING
The light settings (OpenGL version only) may be specified for any of the views. Up to 8 lights may be enabled and their color may be specified. The lights may be static or dynamic, local, infinite, or a spotlight. The light direction may be specified directly or moved using the mouse.

RENDER
This menu contains commands for setting the renderer options, such as renderer resolution, anti-aliasing on/off, anti-aliasing tolerance and depth, pixel size, line and symbol radius. The render view may also be selected and also the image size.

VSWELL

Subroutine: VSWELL

Description: The user subroutine VSWELL allows you to include pure swelling (dilatational) creep in Marc.
Mentat Help Commands in W

**WRITE**

Command:  write_marc

Description:  This command writes a Marc input file with data from the current model database. You must specify the name of the file. Conventionally, Marc input file names end with .dat.

Keyboard Command Sequence:

```
write_marc <MARC file name>
```

**COLOR FILE**

Command:  write_postscript_color

Description:  This command writes a color PostScript file with data currently displayed in the graphics window. You must specify the name of the PostScript file.

Keyboard Command Sequence:

```
write_postscript_color <file name>
```
**Command:** write_postscript_gray

**Description:** This command writes a gray scale PostScript file with data currently displayed in the graphics window. You must specify the name of the PostScript file.

**Keyboard Command Sequence:**

```
write_postscript_gray <file name>
```
Mentat Help Commands in X

**CLEAR CURVES**

Command: xcurve_clear
Description: Clear all curves.
Keyboard Command Sequence: xcurve_clear

**BIAXIAL**

Command: xcurve_dual_biax
Description: Toggle the plotting of the predicted stress response for the biaxial deformation mode ON/OFF.
Keyboard Command Sequence: xcurve_dual_biax <on or off>
**PLANAR SHEAR**

Command: `xcurve_dual_plshr`

Description: Toggle the plotting of the predicted stress response for the planar shear deformation mode ON/OFF.

Keyboard Command Sequence:

```
xcurve_dual_plshr <on or off>
```

**SIMPLE SHEAR**

Command: `xcurve_dual_sishr`

Description: Toggle the plotting of the predicted stress response for the simple shear deformation mode ON/OFF.

Keyboard Command Sequence:

```
xcurve_dual_sishr <on or off>
```

**UNIAXIAL**

Command: `xcurve_dual_uniax`

Description: Toggle the plotting of the predicted stress response for the uniaxial deformation mode ON/OFF.

Keyboard Command Sequence:

```
xcurve_dual_uniax <on or off>
```
**VOLUMETRIC**

Command:  `xcurve_dual_vol`

Description: Toggle the plotting of the predicted stress response for the volumetric deformation mode ON/OFF.

Keyboard Command Sequence:
`xcurve_dual_vol <on or off>`

**SCALE AXES**

Command:  `xcurve_fill`

Description: Set both the abscissa and ordinate bounds.

Keyboard Command Sequence:
`xcurve_fill`

**SAVE CURVES**

Command:  `xcurve_save`

Description: This command will make a response curve provided by subsequent calibrations to supplant those of an earlier fit.

Keyboard Command Sequence:
`xcurve_save <on or off>`

**APPLY**

Command:  `xcv_apply`

Description: This command copies the computed material model coefficients into a material model: it either creates a new one or changes the existing one.

Keyboard Command Sequence:
`xcv_apply`
**Command:** `xcv_biax_vol`  

**Description:** Volumetric data of an equibiaxial test can be given as:  
- The ratio of the current thickness and the original thickness \((t/t_0)\). The button text for this case will be \(t/t_0\).
  
or:
- The ratio of the current volume and the original volume \((V/V_0)\). The button text for this case will be \(V/V_0\).

By default, Mentat expects the volumetric data to be in terms of \(t/t_0\). If the actual data is in terms of \(V/V_0\), use `xcv_biax_vol` to switch to this type of input.

**Keyboard Command Sequence:**

```plaintext
xcv_biax_vol <on or off>
```

---

**Command:** `xcv_checks`  

**Description:** This command activates mathematical checks for Ogden and Foam materials and causes the data fitter to discard the coefficients when one of the mathematical conditions on them is not satisfied. These conditions are considered to be very strict and at times no set of coefficients may be found, or the fit may be very poor.

**Keyboard Command Sequence:**

```plaintext
xcv_checks <on/off>
```
UNIAXIAL

Command: **xcv_color**

Description: This command sets the red, green, and blue color for the index of the associated Experimental Curve Fitting line.

Keyboard Command Sequence:
```
color <index> <red> <green> <blue>
```

Other Buttons with the same description:
- BIAXIAL
- PLANAR SHEAR
- SIMPLE SHEAR
- VOLUMETRIC
- CONSTANT STRAIN DAMAGE
- INCREASING STRAIN DAM.
- STRESS RELAXATION
- BULK RELAXATION
- W RELAXATION
- 192-207

COMPUTE

Command: **xcv_compute**

Description: This command starts the data fitting program with the selected data. Upon fitting, the measured and fitted curves are displayed and the corresponding material model coefficients and the least squares error are reported.

For the Mooney-Rivlin, Ogden, Foam, Arruda-Boyce, and Gent models, the response in the modes for which no data is measured, is predicted. Notice that if volumetric data is relevant in order to predict a uniaxial, biaxial or planar shear mode, this is calculated using the constraint of a zero stress component in a direction perpendicular to the direction of the measured stress component. If this calculation fails, the predicted mode will contain zero stresses.

If the data set contains a large number of entries, or if the model is highly nonlinear and/or contains many coefficients, then the fitting procedure may take some time.
Since the curve fitting procedure does not use weighting factors per data point, it might be useful to have many data points near regions where an accurate response is desired.

**Keyboard Command Sequence:**

```
xcv_compute
```

---

**Command:** `xcv_extrapolation`

**Description:** For Mooney-Rivlin, Ogden, Foam, Arruda-Boyce, and Gent models, this command gives the possibility to get the response of the material outside the range of measurements. This might be important if the deformations of the structure to be analyzed exceed those of the experiments. The user needs to set the new left and right bounds up to which the extrapolation will be performed. Notice that when volumetric information is provided, outside the range of measurements the volumetric data is calculated based on linear extrapolation using the two closest measured data points.

**Keyboard Command Sequence:**

```
xcv_extrapolation <on/off>
```

Another Button with the same description:

EXTRAPOLATION

---

**Command:** `xcv_custom`

**Description:** Customized material model.

**Keyboard Command Sequence:**

```
xcv_custom
```
CONTINUOUS DAMAGE

Command: `xcv_discontinuous`
Description: This command tells the data fitter that one is fitting the data set to the discontinuous damage model. The defaults setting is on. Also it is useful to compare single data set between the two modes. When turned off, it confirms that the correct value for the free energy at maximum strain, has been entered.

Keyboard Command Sequence:
```
xcv_discontinuous <yes or no>
```

FREE ENERGY

Command: `xcv_energy`
Description: This command enters value of free energy at maximum strain amplitude. This value is needed to calibrate the continuous damage model. The code utilizes this value to calculate the arclength of the strain energy path.

Keyboard Command Sequence:
```
xcv_energy <value>
```

xcv_err_abs

Command: `xcv_err_abs`
Description: The least squares error to be minimized during data fitting can be based on absolute or relative errors:

\[
\text{Err}_{\text{Abs}} = \sum \left\{ \left[ \text{data\_measured}(i) - \text{data\_calculated}(i) \right]^2 \right\}_i
\]
\[
\text{Err}_{\text{Rel}} = \sum \left\{ \left[ 1 - \frac{\text{data\_calculated}(i)}{\text{data\_measured}(i)} \right]^2 \right\}_i
\]

This command can be used to switch between relative (default) and absolute errors.

Keyboard Command Sequence:
```
xcv_err_abs <on or off>
```
**ERROR LIMIT**

**Command:** `xcv_err_lim`  
**Description:** During curve fitting, an optimal set of material coefficients is searched for using the Downhill Simplex method. If the least squares error (also see `xcv_err_abs`) corresponding to a set of material coefficients is larger than the error limit, a new minimum will be searched for, unless the maximum number of iterations has been reached (also see `xcv_niter`). This command sets the error limit. If, upon fitting, the reported error is larger than the error limit, searching for a new minimum has been terminated due to reaching the maximum number of iterations.

**Keyboard Command Sequence:**
```
xcv_err_lim <error limit>
```
FICTIVE POISSON’S RATIO

Command: \texttt{xcv\_ipois}

Description: Toggle this value to ON in order to use the fictive Poisson’s ratio to create volumetric information.

Keyboard Command Sequence:
\texttt{xcv\_ipois \langle on or off\rangle}

LEFT BOUND

Command: \texttt{xcv\_left\_bound}

Description: Specify the left bound up to which to perform the extrapolation. If the user prescribes a value within the range of the data set, it will be ignored.

Keyboard Command Sequence:
\texttt{xcv\_left\_bound \langle extrapolation left bound\rangle}

USE ALL DATA

Command: \texttt{xcv\_mode\_all}

Description: This command uses all the available data sets in fitting the model.

Keyboard Command Sequence:
\texttt{xcv\_all\_modes \langle on/off\rangle}
Command:  `xcv_mode_biaxial`

Description:  This command uses the available biaxial data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

Keyboard Command Sequence:

`xcv_mode_biaxial <on/off>`

---

Command:  `xcv_mode_damage`

Description:  This command uses the available damage data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

Keyboard Command Sequence:

`xcv_mode_damage <on or off>`

Another Button with the same description:

INCREASING STRAIN
**PLANAR SHEAR**

**Command:** `xcv_mode_planarshear`

**Description:** This command uses the available planar shear data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

**Keyboard Command Sequence:**

```
xcv_mode_planar_shear <on/off>
```

---

**RELAXATION**

**Command:** `xcv_mode_relaxation`

**Description:** This command uses the available (shear) relaxation data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

**Keyboard Command Sequence:**

```
xcv_mode_relaxation <on or off>
```
Command: **xcv_mode_relaxation_bulk**

Description: This command uses the available (bulk) relaxation data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

**Keyboard Command Sequence:**

```
xcv_mode_relaxation_bulk <on or off>
```

---

Command: **xcv_mode_relaxation_w**

Description: This command uses the available (energy) relaxation data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

**Keyboard Command Sequence:**

```
xcv_mode_relaxation_w <on or off>
```
SIMPLE SHEAR

Command:  `xcv_mode_simpleshear`

Description: This command uses the available simple shear data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

Keyboard Command Sequence:

`xcv_mode simple_shear <on/off>`

UNIAXIAL

Command:  `xcv_mode_uniaxial`

Description: This command uses the available uniaxial data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it.

Keyboard Command Sequence:

`xcv_mode uniaxial <on/off>`
**VOLUMETRIC**

**Command:** `xcv_mode_volumetric`

**Description:** This command uses the available volumetric data set in fitting the model.

If this button is cross-hatched, then the specific test does not have a table associated with it and cannot be selected. Go back to the EXPERIMENTAL DATA FITTING menu and associate a table with the experiment if you wish to activate it. This type of test data cannot be used for Mooney-Rivlin, Arruda-Boyce, and Gent material models.

**Keyboard Command Sequence:**

```
xcv_mode_volumetric <on/off>
```
Command: **xcv_plshr_vol**

**Description:** Volumetric data of a planar shear test can be given as:

- The ratio of the current thickness and the original thickness \((t/t_0)\). The button text for this case will be \(t/t_0\).

  or:

- The ratio of the current volume and the original volume \((V/V_0)\). The button text for this case will be \(V/V_0\).

  By default, Mentat expects the volumetric data to be in terms of \(t/t_0\). If the actual data is in terms of \(V/V_0\), use **xcv_plshr_vol** to switch to this type of input.

**Keyboard Command Sequence:**

```
xcv_plshr_vol <on or off>
```
**EVALUATE**

**Command:** `xcv_recalc`

**Description:** This option prompts the user to enter the coefficients of the material model and displays the response curves. Notice that, although no fit is performed, at least one set of test data must be selected. This feature is available for the Mooney-Rivlin, Ogden, Foam, Arruda-Boyce, and Gent material models.

**Keyboard Command Sequence:**

```
xcv_recalc <coefficients of material model>
```

**RESET**

**Command:** `xcv_reset`

**Description:** This command resets the experimental data fitting parameters to their default values and clears the results.

**Keyboard Command Sequence:**

```
xcv_reset
```

**RIGHT BOUND**

**Command:** `xcv_right_bound`

**Description:** Specify the right bound up to which to perform the extrapolation. If the user prescribes a value within the range of the data set, it will be ignored.

**Keyboard Command Sequence:**

```
xcv_right_bound <extrapolation right bound>
```
UNIAXIAL

Command: xcv_table

Description: Associate a table with the specified test. Depending on the test data, the table should contain the following data:

- For time independent elastomeric materials (uniaxial, biaxial, planar shear, simple shear, and volumetric test):

  eng. strain_1, eng. stress_1, volumetric data_1
  eng. strain_2, eng. stress_2, volumetric data_2
  ...
  ...
  ...
  eng. strain_n, eng. stress_n, volumetric data_n

  If, for uniaxial, biaxial or planar shear data, the third column is left empty, the material is assumed to be incompressible.

- For continuous damage (resulting from a constant strain amplitude test):

  cycle_1, str_energy_d_1
  cycle_2, str_energy_d_2
  ...
  ...
  ...
  cycle_n, str_energy_d_n

  In addition, before fitting the data, the free energy (which is the strain energy density corresponding to the undamaged state) is required. Notice that the data points should not include the range of cycles at which damage did not start to evaluate.

- For discontinuous damage (resulting from an increasing strain amplitude test):

  str_energy_d_1, strain_energy_d_1/str_energy_d_1_undamaged
  str_energy_d_2, strain_energy_d_2/str_energy_d_2_undamaged
  ...
  ...
  ...
  str_energy_d_n, strain_energy_d_n/str_energy_d_n_undamaged

  Notice that the data points should not include the range of cycles at which damage did not start to evaluate.
For viscoelastic material behavior (shear relaxation, bulk relaxation, and energy relaxation test):

\[
\text{time}_1, \text{value}_1 \text{ (shear modulus, bulk modulus or strain energy)} \\
\text{time}_2, \text{value}_2 \text{ (shear modulus, bulk modulus or strain energy)} \\
\vdots \\
\text{time}_n, \text{value}_n \text{ (shear modulus, bulk modulus or strain energy)}
\]

Keyboard Command Sequence:

```
xcv_table <data fit mode> <table name>
```

Other Buttons with the same description:

BIAXIAL
PLANAR SHEAR
SIMPLE SHEAR
VOLUMETRIC
BULK RELAX.
CONSTANT
ENERGY RELAX.
INCREASING
SHEAR RELAX.

---

**CONVERGENCE TOLERANCE**

Command: **xcv_tol**

**Description:** The data fitter uses the Downhill Simplex method to find a (local) minimum. This method uses several sets of material coefficients and calculates the corresponding objective functions. Based on the values of the objective functions, the sets of material coefficients are modified. This process is terminated when:

\[
2 \times \text{abs}((\text{fh} - \text{fl}) / (\text{abs}((\text{fh}) + \text{abs}(\text{fl})))) < \text{convergence tolerance}
\]

in which \(\text{fh}\) is the highest and \(\text{fl}\) is the lowest objective function found so far. The process is also terminated after 2000 trials if the convergence tolerance has not been reached. This command sets the convergence tolerance.

**Keyboard Command Sequence:**

```
xcv_tol <convergence tolerance>
```
Command: **xcv_uniax_vol**

**Description:** Volumetric data of a uniaxial test can be given as:

- The ratio of the current cross sectional area and the original cross sectional area (A/A0).

  or:

- The ratio of the current volume and the original volume (V/V0).

By default, Mentat expects the volumetric data to be in terms of A/A0. If the actual data is in terms of V/V0, use `xcv_uniax_vol` to switch to this type of input.

**Keyboard Command Sequence:**

```
xcv_mode_damage <on or off>
```

Another Button with the same description:

V/V0

---

Command: **xdump**

**Description:** This command dumps the invokes of the Xwindow dumper from within Mentat. The utility uses the Xwindows utilities `xwd` and `xpr`. The `xwd` utility is invoked from the `xdump.gen` script located in the tools directory.

After you enter the `xdump` command you will have five seconds to refresh the display and click on the window you wish to dump. If everything is set up correctly, your window’s contents will be printed on the device connected to your system.

**IMPORTANT:**

The `xdump.gen` script must be tailored to your specific hardware and software environment by someone who is familiar with the use of shell scripts and redirection of I/O to ports or to devices on a network. Since the specifics of such environments are often very complicated, we cannot supply a definitive set of tools for general use.
Keyboard Command Sequence:

`xdump1`

Another Button with the same description:

`XDUMP 2`

Command: `set_xy_draw_crv_histplot curve_type`

Description: This command sets the curve type for history plot.

Keyboard Command Sequence:

`set_xy_draw_crv_histplot <curve or bar or scatter>`

Other Buttons with the same description:

BAR
SCATTER

Command: `set_xy_draw_crv_pathplot curve_type`

Description: This command sets the curve type for path plot.

Keyboard Command Sequence:

`set_xy_draw_crv_pathplot <curve or bar or scatter>`

Other Buttons with the same description:

BAR
SCATTER
**CURVES**

Command: `set_xy_draw_crv_table curve_type`

Description: This command sets the curve type for table plot.

Keyboard Command Sequence:

```
set_xy_draw_crv_table <curve or bar or scatter>
```

Other buttons with the same description:

- BARS
- SCATTER

**CURVES**

Command: `set_xy_draw_crv_xcurve curve_type`

Description: This command sets the curve type for the plot.

Keyboard Command Sequence:

```
set_xy_draw_crv_xcurve <curve or bar or scatter>
```

Another Button with the same description:

- SCATTER

**GENERALIZED XY PLOT**

Menu: GENERALIZED XY PLOT

Description: Generalized XY-Plot has the ability to collect plots from various plotters: History plot, Design plot, Path plot, Table and Data Fit plot. The overlapping feature allows the user to compare plots.

In the GET CURVES FROM section, the user can collect plots from various sources.

The CURVE OPERATIONS section has some basic manipulation functions of the plots.

The LIMITS section allows the change of the plot range.

The LABEL section allows the change of plot setting.
### CURVES

**Command:**  `xy_plot_draw_crv curve_type`  
**Description:** This command sets the curve type for the plot.  
**Keyboard Command Sequence:**  
```
xy_plot_draw_crv_xy_plot <curves or bar or scatter>
```

Other Buttons with the same description:  
- BARS  
- SCATTER

### FILLED

**Command:**  `xy_plot_filled`  
`xy_plot_unfilled`  
**Description:** These commands specify whether or not areas under tabular functions be filled with color.  
**Keyboard Command Sequence:**  
```
xy_plot_filled  
xy_plot_unfilled
```

### FIT

**Command:**  `xy_plot_fit`  
**Description:** This command adjusts the limits of the current plots so that all point values fit within those limits.  
**Keyboard Command Sequence:**  
```
xy_plot_fit
```
Command: `get_history_plots`

Description: This command sends all of the plots in the respective plotter to the Generalized XY plotter. Here are the respective plotters:

<table>
<thead>
<tr>
<th>Button</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY</td>
<td><code>get_history_plots</code></td>
</tr>
<tr>
<td>PATH</td>
<td><code>get_path_plots</code></td>
</tr>
<tr>
<td>RESPONSE GRAD/DESIGN VAR</td>
<td><code>get_opt_plots</code></td>
</tr>
<tr>
<td>TABLE</td>
<td><code>get_table_plots</code></td>
</tr>
<tr>
<td>EXPERIMENTAL DATA FIT</td>
<td><code>get_xcurve_plots</code></td>
</tr>
</tbody>
</table>

Keyboard Command Sequence:

```
get_history_plots
get_path_plots
get_opt_plots
get_table_plots
get_xcurve_plots
```

Other Buttons with the same description:

- PATH
- DESIGN PLOT
- TABLE PLOT
- EXPERIMENTAL DATA FIT
Command:  
```plaintext
get_history_plots
get_path_plots
get_opt_plots
get_table_plots
get_xcurve_plots
```

Description: The greater sign > exists in several different menus. This button copies all of the plots in the current plotter’s (e.g. table, history plot, path plot, experimental data fit, and response grad/design var) to the Generalized XY plotter.

Keyboard Command Sequence:
```plaintext
get_history_plots
get_path_plots
get_opt_plots
get_table_plots
get_xcurve_plots
```

Command:  
```plaintext
xy_plot_legend
```

Description: This command toggles the drawing of the plot legend.

Keyboard Command Sequence:
```plaintext
xy_plot_legend
```
Command:  

```
set_xy_plot_scale
set_xy_plot_rot
set_xy_plot_trans
set_xy_plot_swap_xy
```

Description: Followings are the definitions for the commands.

- **scale**
  This sets the scale factors in the X and Y axes for the selected plot. These factors are multiple with each coordinates. The result will increase or decrease the plot range about the inputted factors.

- **rot**
  This sets the rotational factors (in degrees) in the X and Y axes for the selected plot. This is useful for flipping the plot across an axis.

- **trans**
  This sets the translational factors in the X and Y axes for the selected plot. This is used to extend (or expand) the points in certain directions.

- **swap_xy**
  This swaps X and Y axes values for the selected plot.

Keyboard Command Sequence:

```
set_xy_plot_scale
set_xy_plot_rot
set_xy_plot_trans
set_xy_plot_swap_xy
```

Other Buttons with the same description:

- SCALE FACTORS
- SWAP X and Y AXES
- TRANSLATIONS
CLEAR

Command:  xy_plot_copy  
          xy_plot_remove  
          xy_plot_clear  
          copy_to_clipboard

Description:  Followings are the definitions for the commands.

  copy          Duplicating a plot.
  remove        Deleting a plot.
  clear         Delete all plots.
  copy_to_clipboard  On NT, copy the tabular data to the OS’s clipboard.

Keyboard Command Sequence:

  xy_plot_copy  
  xy_plot_remove  
  xy_plot_clear  
  copy_to_clipboard

RESET

Command:  xy_plot_reset_label

Description:  This command resets the title of the plot, the name of the X-axis and the name of the Y-axis to their default values.

Keyboard Command Sequence:

  xy_plot_reset_label

SHOW XY PLOT

Command:  show_xy_plot

Description:  This command specifies that the XY Plot is to be displayed. It is particularly useful for toggling between different plot settings.

Keyboard Command Sequence:

  show_xy_plot
Commands Begin with Y
Mentat Help Commands in Y

**Command:** `xcv_model yeoh`

**Description:** This command is used if experimental data must be fitted using the Yeoh strain energy function $W$, which is given by:

$$ W = C_{10}*(I_1 - 3) + C_{20}*(I_1 - 3)^2 + C_{30}*(I_1 - 3)^3 $$

where $I_1$ is the first invariant of the right Cauchy-Green strain tensor and $C_{10}$, $C_{20}$, and $C_{30}$ are the material parameters to be determined.

For this model, a volumetric test cannot be supplied. The user may enter a bulk modulus (material_type mechanical: mooney). If no bulk modulus is given, Marc assumes (nearly) incompressible material behavior.

**Keyboard Command Sequence:**

```
xcv_model yeoh
```

**Subroutine:** `YIEL`

**Description:** The user subroutine `YIEL` allows you to define the yield stress based on the current workhardening and other state variables.
**Subroutine: ZERO**

**Description:** The user subroutine ZERO allows you to calculate the equivalent yield stress based on the current total stresses. The function ZERO in the program applies the von Mises yield criterion as a default. You may substitute another yield criterion by writing a new function ZERO. The MOHRC option should not be used when a new function ZERO is used because of the danger of taking into account the effects of hydrostatic pressure twice.
Command: **zoom_box**  
**Description:** This command zooms in on the selected portion of the model.  
**Keyboard Command Sequence:**  
```
zoombox <use mouse to select box coordinates>
```

---

Command: **zoom_in**  
**Description:** This command zooms in all active views using the current zoom increment set by the **zoom_increment** command. Zooming in multiplies the views’ zoom factors by the zoom increment.  
**Keyboard Command Sequence:**  
```
zoom_in
```

Another Button with the same description:  
```
ZOOM IN
```

---

Command: **zoom_increment**  
**Description:** This command sets the zoom factor used by the commands **zoom_in** and **zoom_out**.  
**Keyboard Command Sequence:**  
```
zoom_increment <zoom increment>
```
Command:  zoom_out

Description:  This command zooms out all active views using the current zoom increment set by the zoom_increment command. Zooming out divides the views’ zoom factors by the zoom increment.

Keyboard Command Sequence:

    zoom_out

Another Button with the same description:

    ZOOM OUT
Appendix A: MSC.Marc Mentat Arithmetic and Database Functions

Arithmetic and Database Functions
- Arithmetic Functions
- Database Functions
- Post File Database Functions
- Set Functions
Arithmetic and Database Functions

This chapter describes the Arithmetic and Database function supported by MSC.Marc Mentat.

Arithmetic Functions

The following are the MSC.Marc Mentat Arithmetic Functions which can be used in the `py_get_float` or `py_get_int` functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cos(arg1)</code></td>
<td>Cosine of arg1 radians</td>
</tr>
<tr>
<td><code>sin(arg1)</code></td>
<td>Sine of arg1 radians</td>
</tr>
<tr>
<td><code>tan(arg1)</code></td>
<td>Tangent of arg1 radians</td>
</tr>
<tr>
<td><code>dcos(arg1)</code></td>
<td>Cosine of arg1 degrees</td>
</tr>
<tr>
<td><code>dsin(arg1)</code></td>
<td>Sine of arg1 degrees</td>
</tr>
<tr>
<td><code>dtan(arg1)</code></td>
<td>Tangent of arg1 degrees</td>
</tr>
<tr>
<td><code>acos(arg1)</code></td>
<td>Arccosine of arg1 radians</td>
</tr>
<tr>
<td><code>asin(arg1)</code></td>
<td>Arcsine of arg1 radians</td>
</tr>
<tr>
<td><code>atan(arg1)</code></td>
<td>Arctangent of arg1 radians</td>
</tr>
<tr>
<td><code>atan2(arg1,arg2)</code></td>
<td>Arctangent of (arg1/arg2) radians</td>
</tr>
<tr>
<td><code>dacos(arg1)</code></td>
<td>Arccosine of arg1 degrees</td>
</tr>
<tr>
<td><code>dasin(arg1)</code></td>
<td>Arcsine of arg1 degrees</td>
</tr>
<tr>
<td><code>datan(arg1)</code></td>
<td>Arctangent of arg1 degrees</td>
</tr>
<tr>
<td><code>datan2(arg1,arg2)</code></td>
<td>Arctangent of (arg1/arg2) degrees</td>
</tr>
<tr>
<td><code>log(arg1)</code></td>
<td>Base-10 logarithm of arg1</td>
</tr>
<tr>
<td><code>ln(arg1)</code></td>
<td>Natural logarithm of arg1</td>
</tr>
<tr>
<td><code>exp(arg1)</code></td>
<td>E to the power arg1</td>
</tr>
<tr>
<td><code>cosh(arg1)</code></td>
<td>Hyperbolic cosine of arg1</td>
</tr>
<tr>
<td><code>sinh(arg1)</code></td>
<td>Hyperbolic sine of arg1</td>
</tr>
<tr>
<td><code>tanh(arg1)</code></td>
<td>Hyperbolic tangent of arg1</td>
</tr>
<tr>
<td><code>acosh(arg1)</code></td>
<td>Inverse hyperbolic cosine of arg1</td>
</tr>
<tr>
<td><code>asinh(arg1)</code></td>
<td>Inverse hyperbolic sin of arg1</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>atanh(arg1)</td>
<td>Inverse hyperbolic tangent of arg1</td>
</tr>
<tr>
<td>sqrt(arg1)</td>
<td>Square root of arg1</td>
</tr>
<tr>
<td>rad(arg1)</td>
<td>Angle in radians of arg1 degrees</td>
</tr>
<tr>
<td>deg(arg1)</td>
<td>Angle in degrees of arg1 radians</td>
</tr>
<tr>
<td>abs(arg1)</td>
<td>Absolute value of arg1</td>
</tr>
<tr>
<td>int(arg1)</td>
<td>Largest integral value not greater than arg1</td>
</tr>
<tr>
<td>frac(arg1)</td>
<td>Fractional part of arg1</td>
</tr>
<tr>
<td>max(arg1,arg2)</td>
<td>Maximum of arg1 and arg2</td>
</tr>
<tr>
<td>min(arg1,arg2)</td>
<td>Minimum of arg1 and arg2</td>
</tr>
<tr>
<td>dist2d(arg1,arg2, arg3, arg4)</td>
<td>Distance in 2-D space between a point with coordinates (arg1,arg2) and a point with coordinates (arg3,arg4)</td>
</tr>
<tr>
<td>dist3d(arg1,arg2, arg3, arg4, arg5, arg6)</td>
<td>Distance in 3-D space between a point with coordinates (arg1,arg2, arg3) and a point with coordinates (arg4,arg5,arg6)</td>
</tr>
</tbody>
</table>
Database Functions

The following are the MSC.Marc Mentat Database Functions which can be used in the `py_get_float` or `py_get_int` functions. Note that the `job_name` and `job_title` functions must be used in a call to the `py_get_string` function and is available to Python scripts only.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>npoints()</code></td>
<td>Number of points in database</td>
</tr>
<tr>
<td><code>point_id(ARG1)</code></td>
<td>Id of ARG1-th point in database</td>
</tr>
<tr>
<td><code>max_point_id()</code></td>
<td>Largest point id in database</td>
</tr>
<tr>
<td><code>point_surface_id(ARG1)</code></td>
<td>Id of surface trimmed by point arg1</td>
</tr>
<tr>
<td><code>point_x(ARG1)</code></td>
<td>Global X-coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_y(ARG1)</code></td>
<td>Global Y-coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_z(ARG1)</code></td>
<td>Global Z-coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_u1(ARG1)</code></td>
<td>First user coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_u2(ARG1)</code></td>
<td>Second user coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_u3(ARG1)</code></td>
<td>Third user coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_s1(ARG1)</code></td>
<td>First surface parametric coordinate of point arg1</td>
</tr>
<tr>
<td><code>point_s2(ARG1)</code></td>
<td>Second surface parametric coordinate of point arg1</td>
</tr>
<tr>
<td><code>ncurves()</code></td>
<td>Number of curves in database</td>
</tr>
<tr>
<td><code>curve_id(ARG1)</code></td>
<td>Id of ARG1-th curve in database</td>
</tr>
<tr>
<td><code>max_curve_id()</code></td>
<td>Largest curve id in database</td>
</tr>
<tr>
<td><code>ncurve_points(ARG1)</code></td>
<td>Number of points of curve arg1</td>
</tr>
<tr>
<td><code>curve_point_id(ARG1,ARG2)</code></td>
<td>Id of ARG2-th point of curve arg1</td>
</tr>
<tr>
<td><code>curve_surface_id(ARG1)</code></td>
<td>Id of surface trimmed by curve arg1</td>
</tr>
<tr>
<td><code>curve_length(ARG1)</code></td>
<td>Length of curve arg1</td>
</tr>
<tr>
<td><code>curve_ndiv(ARG1)</code></td>
<td>Number of divisions of curve arg1</td>
</tr>
<tr>
<td><code>nsurfaces()</code></td>
<td>Number of surfaces in database</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>surface_id(ARG1)</td>
<td>Id of ARG1-th surface in database</td>
</tr>
<tr>
<td>max_surface_id()</td>
<td>Largest surface id in database</td>
</tr>
<tr>
<td>nsurface_curves(arg1)</td>
<td>Number of trimming curves of surface arg1</td>
</tr>
<tr>
<td>surface_curve_id(arg1,ARG2)</td>
<td>Id of ARG2-th curve in database that trims surface arg1</td>
</tr>
<tr>
<td>nsurface_points_x(arg1)</td>
<td>Number of defining points in first parametric direction of surface arg1</td>
</tr>
<tr>
<td>nsurface_points_y(arg1)</td>
<td>Number of defining points in second parametric direction of surface arg1</td>
</tr>
<tr>
<td>surface_point_id(arg1,ARG2,Arg3)</td>
<td>Id of (ARG2, ARG3) - the defining point of surface arg1</td>
</tr>
<tr>
<td>nsolids()</td>
<td>Number of solids in database</td>
</tr>
<tr>
<td>solid_id(ARG1)</td>
<td>Id of ARG1-th solid in database</td>
</tr>
<tr>
<td>nsolid_lumps(arg1)</td>
<td>Number of lumps of solid arg1</td>
</tr>
<tr>
<td>nsolid_faces(arg1)</td>
<td>Number of faces of solid arg1</td>
</tr>
<tr>
<td>nsolid_edges(arg1)</td>
<td>Number of edges of solid arg1</td>
</tr>
<tr>
<td>nsolid_vertices(arg1)</td>
<td>Number of vertices of solid arg1</td>
</tr>
<tr>
<td>solid_area(arg1)</td>
<td>Surface area of solid arg1</td>
</tr>
<tr>
<td>solid_volume(arg1)</td>
<td>Volume of solid arg1</td>
</tr>
<tr>
<td>nnodes()</td>
<td>Number of nodes in database</td>
</tr>
<tr>
<td>node_id(ARG1)</td>
<td>Id of ARG1-th node in database</td>
</tr>
<tr>
<td>max_node_id()</td>
<td>Largest node id in database</td>
</tr>
<tr>
<td>node_x(arg1)</td>
<td>Global X-coordinate of node arg1</td>
</tr>
<tr>
<td>node_y(arg1)</td>
<td>Global Y-coordinate of node arg1</td>
</tr>
<tr>
<td>node_z(arg1)</td>
<td>Global Z-coordinate of node arg1</td>
</tr>
<tr>
<td>node_u1(arg1)</td>
<td>First user coordinate of node arg1</td>
</tr>
<tr>
<td>node_u2(arg1)</td>
<td>Second user coordinate of node arg1</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>node_u3(arg1)</td>
<td>Third user coordinate of node arg1</td>
</tr>
<tr>
<td>nelements( )</td>
<td>Number of elements in database</td>
</tr>
<tr>
<td>element_id(ARG1)</td>
<td>Id of ARG1-th element in database</td>
</tr>
<tr>
<td>max_element_id( )</td>
<td>Largest element id in database</td>
</tr>
<tr>
<td>element_node_id(arg1,ARG2)</td>
<td>Id of ARG2-th node of element arg1</td>
</tr>
<tr>
<td>job_name( )</td>
<td>Name of the job (py_get_string only).</td>
</tr>
<tr>
<td>job_title( )</td>
<td>Title of the job (py_get_string only).</td>
</tr>
</tbody>
</table>
Post File Database Functions

The following are the MSC.Marc Post File Database Functions which can be used in the `py_get_float` or `py_get_int` functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>displace_x(arg1)</code></td>
<td>Global X displacement of node arg1</td>
</tr>
<tr>
<td><code>displace_y(arg1)</code></td>
<td>Global Y displacement of node arg1</td>
</tr>
<tr>
<td><code>displace_z(arg1)</code></td>
<td>Global Z displacement of node arg1</td>
</tr>
<tr>
<td><code>post_increments()</code></td>
<td>If a post file is opened, it returns the number of increments in the post file.</td>
</tr>
<tr>
<td><code>post_node_extra(arg1)</code></td>
<td>Returns 1 if node arg1 is an extra (dummy) node, 0 if a valid node</td>
</tr>
<tr>
<td><code>scalar_1(arg1)</code></td>
<td>First scalar component of node arg1</td>
</tr>
<tr>
<td><code>scalar_2(arg1)</code></td>
<td>Second scalar component of node arg1</td>
</tr>
<tr>
<td><code>scalar_max_element()</code></td>
<td>Returns the first element in which the scalar quantity is the maximum value.</td>
</tr>
<tr>
<td><code>scalar_min_element()</code></td>
<td>Returns the first element in which the scalar quantity is the minimum value.</td>
</tr>
<tr>
<td><code>scalar_max_node()</code></td>
<td>Returns the first node in which the scalar quantity is the maximum value.</td>
</tr>
<tr>
<td><code>scalar_min_node()</code></td>
<td>Returns the first node in which the scalar quantity is the minimum value.</td>
</tr>
<tr>
<td><code>scalar_pos(x,y,z)</code></td>
<td>Returns the value of the scalar quantity that is being postprocessed at position (x,y,z), if that position is located somewhere in the finite element mesh or 0 if the position is outside the finite element mesh. The value is obtained by interpolating the closest element.</td>
</tr>
</tbody>
</table>
## Set Functions

The following are the MSC.Marc Mentat Set functions, available to Python scripts only, which can be used in a call to the `py_get_float` or `py_get_int` functions. The `set_name` and `set_type` functions must be used in a call to the `py_get_string` function.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>is_curve_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a curve set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_edge_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a edge set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_element_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a element set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_face_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a face set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_node_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a node set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_point_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a point set, 0 otherwise.</td>
</tr>
<tr>
<td><code>is_surface_set(arg1)</code></td>
<td>Returns 1 if set arg1 is a surface set, 0 otherwise.</td>
</tr>
<tr>
<td><code>nsets()</code></td>
<td>Number of sets in database</td>
</tr>
<tr>
<td><code>nset_entries(arg1)</code></td>
<td>Number of set entries for set arg1</td>
</tr>
<tr>
<td><code>set_edge(arg1, ARG2)</code></td>
<td>The id of the ARG2-th edge set entry in edge set arg1.</td>
</tr>
<tr>
<td><code>set_entry(arg1, ARG2)</code></td>
<td>The value of the ARG2-th set entry in set arg1.</td>
</tr>
<tr>
<td><code>set_face(arg1, ARG2)</code></td>
<td>The id of the ARG2-th face set entry in face set arg1.</td>
</tr>
<tr>
<td><code>set_id(ARG1)</code></td>
<td>Id of ARG1-th set in database</td>
</tr>
<tr>
<td><code>set_name(arg1)</code></td>
<td>Name of set arg1. This is a string value, and can only be used in <code>py_get_string</code>.</td>
</tr>
<tr>
<td><code>set_type(arg1)</code></td>
<td>Type of set arg1. This is a string value, and can only be used in <code>py_get_string</code>. Returned values will be one of the following strings: node, element, point, edge, curve, face, surface.</td>
</tr>
</tbody>
</table>