

SAND88-1432  
Unlimited Release  
Printed June 1989  
Reprinted August 1991

Distribution  
UC-32

# **BLOT— A Mesh and Curve Plot Program for the Output of a Finite Element Analysis**

Amy P. Gilkey  
Applied Mechanics Division III  
John H. Glick  
Fluid Mechanics and Heat Transfer Division I  
Sandia National Laboratories  
Albuquerque, New Mexico 87185

## **Abstract**

BLOT is a graphics program for post-processing of finite element analyses output in the EXODUS database format. It is command driven with free-format input and can drive any graphics device supported by the Sandia Virtual Device Interface.

BLOT produces mesh plots with various representations of the analysis output variables. The major mesh plot capabilities are deformed mesh plots, line contours, filled (painted) contours, vector plots of two/three variables (e.g., velocity vectors), and symbol plots of scalar variables (e.g., discrete cracks). Pathlines of analysis variables can also be drawn on the mesh. BLOT's features include element selection by material, element birth and death, multiple views for combining several displays on each plot, symmetry mirroring, and node and element numbering.

BLOT can also produce X-Y curve plots of the analysis variables. BLOT generates time-versus-variable plots or variable-versus-variable plots. It also generates distance-versus-variable plots at selected time steps where the distance is the accumulated distance between pairs of nodes or element centers.

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	An Overview of the EXODUS Database .....	6
1.2	Time Step Modes .....	7
1.3	Graphics Devices .....	7
<b>2</b>	<b>Mesh Display .....</b>	<b>9</b>
2.1	MESH Subprogram .....	9
2.2	DETOUR Subprogram .....	9
2.3	PATHLINE Subprogram .....	10
2.4	Active Elements .....	10
2.5	Multiple Views .....	10
2.6	Deformed Mesh Considerations .....	11
2.7	3D Mesh Display .....	11
<b>3</b>	<b>X-Y Curve Plots .....</b>	<b>13</b>
3.1	TPLOT Subprogram .....	13
3.2	SPLIT Subprogram .....	14
<b>4</b>	<b>Command Input .....</b>	<b>15</b>
4.1	General Commands (any subprogram) .....	16
	Program Control (any subprogram, including BLOT level) .....	16
	Information (any subprogram, including BLOT level) .....	18
	Time Step Selection (any subprogram) .....	21
	Graphics Options (any subprogram) .....	24
	Plot Set Display (any subprogram) .....	26
4.2	Mesh Commands (any mesh or X-Y curve subprogram) .....	27
	Mesh View Control (any mesh or X-Y curve subprogram) .....	27
	Active Element Control (any mesh or X-Y curve subprogram) .....	29
	Multiple Views Control (any mesh or X-Y curve subprogram) .....	31
	Mesh Control (any mesh or X-Y curve subprogram) .....	32
	3D Rotation Commands (any mesh or X-Y curve subprogram) .....	34
4.3	DETOUR Commands ( DETOUR subprogram) .....	35
	Mesh Display Mode Control ( DETOUR subprogram) .....	35
	Contour Control ( DETOUR subprogram) .....	37
	Mode Specific Options ( DETOUR subprogram) .....	38
4.4	PATHLINE Commands ( PATHLINE subprogram) .....	39
4.5	X-Y Curve Commands (any X-Y curve subprogram) .....	40
	Curve Definition (any X-Y curve subprogram) .....	40
	Axis Labeling and Scaling (any X-Y curve subprogram) .....	40
	Curve Display Options (any X-Y curve subprogram) .....	42
	Neutral File Options (any X-Y curve subprogram) .....	43
	Mesh Display (any X-Y curve subprogram) .....	44
4.6	TPLOT Commands ( TPLOT subprogram) .....	45
	Curve Definition ( TPLOT subprogram) .....	45
4.7	SPLIT Commands ( SPLIT subprogram) .....	46
	Curve Definition ( SPLIT subprogram) .....	46

Mesh Display ( SPLIT subprogram) .....	47
<b>5 Plot Output .....</b>	<b>49</b>
5.1 Mesh Output .....	49
DETOUR Mesh Output .....	50
PATHLINE Mesh Output .....	50
Curve Plot Mesh Output .....	50
5.2 X-Y Curve Output .....	50
TPLOT Curve Output .....	51
SPLOT Curve Output .....	51
<b>6 Informational and Error Messages .....</b>	<b>53</b>
<b>7 Executing BLOT .....</b>	<b>55</b>
7.1 Execution Files .....	55
7.2 Special Software .....	55
References	57



# 1 Introduction

*BLOT* is a graphics program for post-processing of finite element analyses. The analysis output data is drawn from an EXODUS database [bib:exodus] file which contains history, global, nodal, and element variables at various time steps.

The *BLOT* program consists of a package of “subprograms”. In general, these subprograms either plot the mesh or plot X-Y curves.

The mesh display capabilities include element selection by material, element birth and death, multiple views for combining several displays on a single plot, symmetric mirroring, node and element numbering, and the display of node sets and side sets.

The MESH subprogram displays a single undeformed mesh plot from an X-Y curve subprogram.

The DETOUR subprogram plots the deformed mesh at user selected time steps from the database. The user can display the output variables on the mesh in one of several formats. Contours of an output variable may be plotted either with simple lines or “painted” color bands. A vector representing two variable values may be drawn at each node or element. A special “element symbol” capability plots a symbol representing a variable value (for example, a crack angle) at each element.

The PATHLINE subprogram plots “paths” of variables on the undeformed mesh. A path is represented by the mesh coordinates stored as analysis variables over user selected time steps.

The TPLOT subprogram generates time history (time-versus-variable) plots and variable-versus-variable plots, where a variable represents a result from the analysis. Both types of plots need a series of user-selected times which specify the times at which the plot values are taken. A time history curve displays the value of a variable at a time versus the time. A variable-versus-variable curve displays the value of one variable at a time versus the value of another variable at the same time.

The SPLOT subprogram plots the analysis results as a function of position in a body (for example, the value of the displacement around the circumference of a cylinder). A distance parameter is created by connecting the distances between user-selected pairs of nodes or pairs of element centers. SPLOT plots the value from the node or element center versus the accumulated distance at that node or element center.

*BLOT* also displays information about the database, including variable values. It has capabilities for limiting the amount of information displayed to the user and to write the information to a print file.

*BLOT* operates in a command / plot cycle. It first accepts commands from the user which set the plot parameters that control the plotting. Then it plots as requested. This cycle is

repeated until the user exits the program. All plot parameters remain the same between plot sets unless the user changes them.

## 1.1 An Overview of the EXODUS Database

The EXODUS database format is described briefly in Appendix [Ref: appx:exodus] and in detail in [bib:exodus]. This section discusses some of the features mentioned in this document.

The first part of the EXODUS database describes the mesh. This is in the GENESIS database format [bib:genesis] that is the input to a finite element analysis. The mesh description includes the nodal coordinates and the element connectivity. *BLOT* can process a GENESIS database to generate undeformed mesh plots.

Each element in the database is assigned to an “element block”. An element block distinguishes a material or an element type (such as a truss or quadrilateral). A specific element variable may be undefined for some element blocks, meaning that the variable has no value for elements in that element block.

The GENESIS database also may contain “node sets” and “side sets”. A node set is a collection of nodes that the user has defined. A side set is a user-defined collection of element sides described by an element number and the nodes in the side. *BLOT* can display either type of set on the mesh.

Following the mesh description is information about the database variables, then a series of time steps. Each variable represents a result of the analysis at each time step. The EXODUS database has a user-defined name associated with each variable. This name is used to reference the variable and to label any plot involving the variable. A time step consists of a time and the values for all the variables at that time.

The EXODUS format defines four types of variables:

- A history variable has a value representative of the system as a whole at each time step (e.g., the total energy).
- A global variable is the same as a history variable except that global variables are only included in “whole” time steps (explained below).
- A nodal variable has a value for every node of the mesh at each whole time step in the analysis (e.g., the displacement in the x-direction).
- An element variable has a value for every element of the finite element analysis at each whole time step (e.g., the stress in the x-direction). An element variable may be undefined for some element blocks, meaning that the variable has no value for elements in that element block.

There are two types of time steps in an EXODUS database: a “history-only” time step contains the values for the history variables only; a “whole” time step includes the values for all the variables (history, global, nodal, and element).

## 1.2 Time Step Modes

Time step selection is performed in one of the following modes:

- **Interval-Times Mode** selects time steps at uniform intervals between a minimum and a maximum time. If this mode has a delta offset, the first selected time is not the minimum time, but the minimum time plus the interval. If this mode has a zero offset, the first selected time is the minimum time.
- **All-Available-Times Mode** selects all time steps between a minimum and a maximum time.
- **User-Selected-Times Mode** selects time steps which are explicitly specified by the user.

The nearest time step from the database is chosen for each selected time. “History-only” time steps are chosen only if the current subprogram can process history variables.

## 1.3 Graphics Devices

*BLOT* can drive both a graphics terminal and a hardcopy (file) device in a single session. This allows the user to generate plots on a graphics terminal and then send selected plots to a hardcopy device. The two-device capability may not be available on all systems. The “primary” device is the graphics terminal, if one is defined, or the hardcopy device if no graphics terminal is defined.

*BLOT* can drive any graphics device supported by the Sandia Virtual Device Interface [bib:svdi].



## 2 Mesh Display

The mesh is displayed by several subprograms. The MESH subprogram displays an undeformed wireframe mesh from an X-Y curve subprogram. The DETOUR subprogram plots functions of the database variables from selected time steps on the deformed mesh. The PATHLINE subprogram plots the path of selected variables on the undeformed mesh over time.

Node and element numbers may be displayed on the mesh. Node sets and side sets may also be displayed.

The user may select/deselect elements by element block. This provides a way to make meaningful contour plots if the variable differs greatly between different element blocks (e.g., stress in foam versus stress in steel). Elements are allowed to “be born” and “die” between time steps on an individual basis. These capabilities are explained in Section [Ref: mesh:actelem] .

A single plot may provide up to four “views” of a mesh. The views may be copies of the original mesh to show different representations of the mesh (for example, the deformed mesh versus the undeformed mesh). The views may also be a reflection of the mesh around a symmetry axis, providing a way to display a whole object from a symmetric mesh that is a quarter or half of that object. This capability is explained in Section [Ref: mesh:multview]

### 2.1 MESH Subprogram

The MESH subprogram may be called from any X-Y curve subprogram to display the undeformed wireframe mesh.

### 2.2 DETOUR Subprogram

The DETOUR subprogram plots the deformed mesh at user selected time steps from the database. The mesh can be represented in two ways and the database variables can be displayed on the mesh in several formats. DETOUR operates in the following “display modes”:

- Wireframe Mesh Mode plots the mesh lines.
- Solid Mesh Mode paints the mesh, using a different color for each element block.
- Line Contour Mode plots contours of a variable on the mesh.
- Paint Contour Mode paints contours of a variable on the mesh.
- Vector Mode plots a two- or three-variable vector (depending on the dimension of the mesh) for each node or element on the mesh.
- Element Symbol Mode plots a symbol for each element on the mesh. The current version draws “cracks” at a specified angle or draws a letter indicating whether a rock joint is sliding or opening.

The output of each display mode is described in Section [Ref: out:detour] .

### **2.3 PATHLINE Subprogram**

The PATHLINE subprogram plots the path of database variables on the undeformed mesh over a set of user-selected time steps. The coordinates of each path point are taken from the value of two or three user-selected variables at the specified time step. The variables may be a set of history or global variables or a set of nodal variables at user-specified nodes. Several paths (i.e., sets of variables or sets of nodes) may be requested.

### **2.4 Active Elements**

Elements are selected in two ways: by element block and by a special birth/death flag. “Active” elements must be selected by both criteria.

One or more element blocks may be “selected”. Only elements in a selected element block contribute to any variable plot (e.g., a contour plot). On a mesh plot, lines separating elements in the same element block (in some display modes) are only drawn for elements in a selected element block.

Element variables may be undefined for some element blocks. If so, only the elements in an element block for which the variable is defined contribute to a plot involving that variable.

Elements are allowed to be born or to die. If an element is “dead”, it is ignored in all calculations and plots, including a simple mesh plot. The user specifies an element variable which gives the state of the element for each time step: the element is alive if the value is zero, dead if it is not zero. This capability can be turned on or off upon command.

### **2.5 Multiple Views**

Up to four “views” of a mesh can be displayed on a single plot. Multiple views are either symmetric or non-symmetric. Symmetric views are mirror images of the original mesh, reflected around user-defined symmetry axes. Non-symmetric views are simply copies of the original mesh. Symmetric views are not allowed for a 3D mesh.

The plot parameters that may vary between views are limited. They are explained in detail in the command descriptions in Chapter [Ref: chap:command] .

Each view has a number associated with it depending on how the views are defined. The view numbering for each possible combination of views is as follows:

Normally the default mesh window for each view is square. However, if there are two views and the non-divided axis length (derived from the default or user-defined mesh limits) is at least twice the divided axis length, the views are rectangular (filling a square). The window may be changed to an arbitrary rectangle with the **SQUARE OFF** command.

## **2.6 Deformed Mesh Considerations**

To plot a deformed mesh, a set of displacement functions and a magnification factor are needed. The displacements are magnified by the factor and then added to the coordinates.

The displacement functions are the first two or three nodal variables (one for each coordinate) if and only if these variables begin with the letter “D” and end with the last character of the coordinate name. For example, if the coordinate name is “X”, nodal variable names “DX” and “DISPX” are valid displacement function names. If the first nodal variables are not valid displacement functions, a warning is displayed and the mesh cannot be deformed.

The default magnification factor for a 2D mesh is five percent of the mesh coordinate range divided by the maximum displacement function absolute value rounded to one significant digit. The default magnification factor for a 3D mesh is always 1.0.

The magnification factor may be set by the user with the **MAGNIFY** command.

## **2.7 3D Mesh Display**

A 3D mesh can be displayed in any subprogram. Only the surface of the mesh may be displayed. The surface is defined as an element face which has no matching face. All functions of database variables apply only to the surface of the mesh. This also applies to node and element numbering and to the node set and side set display. The one exception is the pathlines which may go through the interior of the mesh.

The hidden line algorithm is not perfect and may not work on all element types. In particular it has trouble with determining the “edge” of the mesh.

The axis displayed for a 2D mesh represents the coordinates of the mesh. In 3D the axis values have little resemblance to the original mesh coordinates. The axis values are only useful for selecting a position on the displayed mesh. Since it is much simpler to use a graphics cursor, the axis values are left off the mesh plot by default.



## 3 X-Y Curve Plots

The X-Y curve plotting capability consists of two subprograms: TPLOT and SPLOT. TPLOT plots time histories of variables and variable-versus-variable plots. SPLOT plots variable values as a function of mesh position.

The user can tailor the plot legend and labels in all X-Y curve subprograms. The QA information, which includes the database title and other identifying data, can optionally be omitted. The user can also provide a plot caption and labels for both axes.

Several user-selectable options control the plot. The curves can all be overlaid on one plot or plotted one curve to a plot. Either axis may be scaled automatically or the user may define the axis limits. All the curves can optionally be scaled to the same limits. The user can specify options such as whether the curve line type varies, whether a symbol is drawn at each curve point, and whether a grid is drawn on the plot.

The curve data can be written in “neutral file” format. This file can then be read by the GRAFAID program [bib:grafaid]. GRAFAID is an interactive X-Y plot program developed for use on VAX computers under the VAX/VMS operating system. It provides an interactive capability for plotting and comparing multiple sets of X-Y data. It has numerous utilities for X-Y data analysis, manipulation, and creation. Utilities are provided for appending X-Y data sets, performing calculations involving X-Y data, editing data, enveloping data, calculating FFT's, integrating and differentiating data, windowing data, and doing statistical calculations.

A simple undeformed wireframe mesh with node and/or element numbering can be displayed from any X-Y curve subprogram.

### 3.1 TPLOT Subprogram

TPLOT generates time history plots and variable-versus-variable plots. Both types of plots need a set of user-selected database time steps which specify the times at which the plot values are taken. A time history curve displays the value of a variable at a time versus the time. A variable-versus-variable curve displays the value of one variable at a time versus the value of another variable at the same time.

Any type of database variable may be plotted. For a nodal or element variable, TPLOT plots values for a selected node or element. The user specifies the node or element number with the variable. For example, the user can plot load versus displacement for any node or stress versus strain from any element or global values of kinetic energy versus time.

The TPLOT “plot set” is the set of all defined curves for all requested variables. The curves can all be overlaid on one plot or plotted one curve to a plot.

## 3.2 SPLOT Subprogram

SPLOT plots the analysis results as a function of position in a body (for example, the value of the displacement around the circumference of a cylinder). A distance parameter is created by connecting the distances between user-selected pairs of nodes or pairs of element centers. SPLOT plots the value from the node or element center versus the accumulated distance at that node or element center.

The user first selects the nodes or elements, then the nodal or element variable. SPLOT generates plots of cumulative distances between the selected nodes versus the values of the nodal variable at those nodes or the cumulative distances between the centers of the selected elements versus the values of the element variable at those elements. A curve is generated for the variable values at each user-selected database time step.

The SPLOT “plot set” is the set of all defined curves for all requested variables and all selected times. The curves for a plot set can be plotted one curve to a plot or can be overlaid on a plot in one of two ways: the curves for all requested variables can be generated on one plot for each time step or the curves for all times steps can be generated on one plot for each variable.

## 4 Command Input

*BLOT* prompts for commands from the user with the name of the current subprogram. The program starts at the *BLOT* level. At this level the user can only request general information and enter a subprogram.

The commands are in free-format and must adhere to the following syntax rules.

- Valid delimiters are a comma or one or more blanks.
- Either lowercase or uppercase letters are acceptable, but lowercase letters are converted to uppercase except in user-defined text that appears on a plot (such as the plot caption).
- A “\$” character in any command line starts a comment. The “\$” and any characters following it on the line are ignored.
- A command may be continued over several lines with an “>” character. The “>” and any characters following it on the current line are ignored and the next line is appended to the current line.

Each command has an action keyword or “verb” followed by a variable number of parameters.

The command verb is a character string. It may be abbreviated, as long as enough characters are given to distinguish it from other commands.

The meaning and type of the parameters is dependent on the command verb. Most command parameters are optional. If an optional parameter field is blank, a command-dependent default value is supplied. Below is a description of the valid entries for parameters.

- A numeric parameter may be a real number or an integer. A real number may be in any legal FORTRAN numeric format (e.g., 1, 0.2, -1E-2). An integer parameter may be in any legal integer format.
- A string parameter is a literal character string. Most string parameters may be abbreviated.
- Variable names must be fully specified. The blank delimiter creates a problem with database variable names with embedded blanks. The program handles this by deleting all embedded blanks from the input database names. For example, the variable name “SIG R” must be entered as “SIGR”. The blank must be deleted in any references to the variable. All database names appear in uppercase without the embedded blanks in all displays.
- Screen and mesh positions may be selected with the graphics cursor (also known as the graphics locator). Cursor input is device-dependent and uses the VDI graphics locator routines. When the program prompts for the position, the user positions the graphics cursor (e.g., the crosshairs) on the screen, then selects the position by

pressing any printable keyboard character (e.g., the space bar).

- Several parameters allow a range of values. A range is in one of the following forms:
- “ $n_1$ ” selects value  $n_1$ ,
- “ $n_1$  TO  $n_2$ ” selects all values from  $n_1$  to  $n_2$ ,
- “ $n_1$  TO  $n_2$  BY  $n_3$ ” selects all values from  $n_1$  to  $n_2$  stepping by  $n_3$ , where  $n_3$  may be positive or negative.

If the upper limit of the range is greater than the maximum allowable value, the upper limit is changed to the maximum without a message.

The notation conventions used in the command descriptions are:

- The command verb is in **bold** type.
- A literal string is in all uppercase **SANSERIF** type and should be entered as shown (or abbreviated).
- The value of a parameter is represented by the parameter name in *italics*.
- A literal string in square brackets (“[ ]”) represents a parameter option which is omitted entirely (including any following comma) if not appropriate. These parameters are distinct from most parameters in that they do not require a comma as a place holder to request the default value.
- The default value of a parameter is in angle brackets (“< >”). The initial value of a parameter set by a command is usually the default parameter value. If not, the initial setting is given with the default or in the command description.

Each command section header tells which subprogram must be active for the commands in the section. The command may behave differently depending on the current subprogram. These differences are documented in the command description under a heading of “Issued from *subprogram*”.

After each command is entered, the effect of the command is displayed. For example, after a TMIN command, the time step parameters are displayed.

The current setting of a command parameter can be displayed with the SHOW command. The LIST command displays database information that may be needed in some commands (e.g., the variable names and the time step times).

## 4.1 General Commands (any subprogram)

### 4.1.1 Program Control (any subprogram, including *BLOT* level)

#### LOG

LOG requests that the log file be saved when the program is exited. Each correct

command that the user enters (excluding Information Commands) is written to the log file.

## **RESET**

RESET resets the plot parameters for the current subprogram to their initial values. All parameters are reset unless otherwise noted in the description of the command that sets the parameter. Some graphics options (see Section [Ref: gen:graphopt] ) and neutral file options (see Section [Ref: xy:neutral] ) are not reset.

## **EXIT**

The EXIT command exits immediately from the program. This is the only safe way to exit the program.

## **DETOUR**

DETOUR switches to the DETOUR subprogram. All mesh plot parameters are reset (if the current subprogram is not DETOUR) unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the *BLOT* level.

## **PATHLINE**

PATHLINE switches to the PATHLINE subprogram. All mesh plot parameters are reset (if the current subprogram is not PATHLINE) unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the *BLOT* level.

## **TPLOT**

TPLOT switches to the TPLOT subprogram and resets all X-Y curve plot parameters (if the current subprogram is not TPLOT) unless otherwise noted in the description of the command that sets the parameter.

The mesh plot parameters are reset if the current subprogram is not an X-Y curve subprogram. This effects all parameters unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the *BLOT* level.

## **SPLOT**

SPLOT switches to the SPLOT subprogram and resets all X-Y curve plot parameters (if the current subprogram is not SPLOT) unless otherwise noted in the description of the command that sets the parameter.

The mesh plot parameters are reset if the current subprogram is not an X-Y curve

subprogram. This effects all parameters unless otherwise noted in the description of the command that sets the parameter.

This command may not be abbreviated except from the *BLOT* level.

#### 4.1.2 Information (any subprogram, including *BLOT* level)

##### **SELECT *option* <No Default>**

SELECT selects the database information for the LIST and PRINT commands. These selections apply only to these two commands. Items are displayed in the order listed in the SELECT command. If the ADD parameter is present, the listed values are added to the current selection.

SELECT NODES [ADD,] *node\_range*<sub>1</sub>, *node\_range*<sub>2</sub>, ... <all nodes>

selects the nodes. Each *node\_range* must be in the range form described under Command Input.

SELECT ELEMENTS [ADD,] *element\_range*<sub>1</sub>, *element\_range*<sub>2</sub>, ... <all elements>

selects the elements. Each *element\_range* must be in the range form described under Command Input. The selected element blocks are changed to select only those blocks that include a selected element. The selected elements are reordered so that they are grouped by element block.

SELECT BLOCKS or MATERIAL [ADD,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... <all element blocks>

selects the element blocks. The *block\_id* is the element block identifier displayed by the LIST BLOCKS command. The selected elements are changed to select all the elements in the selected blocks and no elements in non-selected blocks.

SELECT NSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... <all node sets>

selects the node sets. The *set\_id* is the node set identifier displayed by the LIST NSETS command.

SELECT SSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... <all side sets>

selects the side sets. The *set\_id* is the side set identifier displayed by the LIST SSETS command.

SELECT HVARs [ADD,] *history\_variable*<sub>1</sub>, *history\_variable*<sub>2</sub>, ... <all history variables>

selects the history variables.

SELECT GVARs [ADD,] *global\_variable*<sub>1</sub>, *global\_variable*<sub>2</sub>, ... <all global variables>

selects the global variables.

```
SELECT NVARs [ADD,] nodal_variable1, nodal_variable2, ... <all nodal variables>
```

selects the nodal variables.

```
SELECT EVARs [ADD,] element_variable1, element_variable2, ... <all element variables>
```

selects the element variables.

```
SELECT READ nstep <1>
```

changes the selected time step to *nstep* steps forward (if *nstep* is positive) or backward (if *nstep* is negative) of the current selected step.

```
SELECT STEP nstep <current selected step>
```

changes the selected time step to step *nstep*.

```
SELECT TIME time <current selected step time>
```

changes the selected time step to the step that has the time nearest *time*.

### **LIST *option* <No Default>**

LIST displays the database information specified by *option* on the user's terminal. The "selected" items are specified with the **SELECT** command.

#### **LIST VARs**

displays a summary of the database. The summary includes the database title; the number of nodes, elements, and element blocks; the number of node sets and side sets; and the number of each type of variable.

#### **LIST COORDINA**

displays the coordinate values for the selected nodes.

#### **LIST MAP**

displays the element order map for all elements.

#### **LIST BLOCKs or MATERIAL**

displays a summary of the selected element blocks. The summary includes the block identifier, the number of elements in the block, the number of nodes per element, and the number of attributes per element.

#### **LIST LINK or CONNECTI**

displays the nodal connectivity of the selected elements.

#### LIST ATTRIBUT

displays the attribute values associated with the selected elements.

#### LIST NSETS

displays a summary of the selected node sets. The summary includes the set identifier and the number of nodes in the set.

#### LIST NNODES

displays the numbers of the nodes in the selected node sets.

#### LIST NFACTORS

displays the distribution factors of the nodes in the selected node sets.

#### LIST SSETS

displays a summary of the selected side sets. The summary includes the set identifier, the number of elements in the set, and the number of nodes in the set.

#### LIST SELEMS

displays the numbers of the elements in the selected side sets.

#### LIST SNODES

displays the numbers of the nodes in the selected side sets.

#### LIST SFACTORS

displays the distribution factors of the nodes in the selected side sets.

#### LIST QA

displays the QA records and the information records.

#### LIST NAMES

displays the names of the history, global, nodal, and element variables.

#### LIST HISTORY or HVARs

displays the values of the selected history variables for the selected time step.

#### LIST GLOBALS or GVARs

displays the values of the selected global variables for the selected time step.

#### LIST NODALS or NVARs

displays the values of the selected nodal variables for the selected nodes for the selected time step.

#### LIST ELEMENTS or EVARS

displays the values of the selected element variables for the selected elements for the selected time step.

#### LIST STEPS

displays the number of time steps and the minimum and maximum time step times.

#### LIST TIMES

displays the step numbers and times for all time steps on the database.

#### **PRINT *option* <no parameter>**

PRINT writes the database information specified by *option* to the print file. The valid *options* are described under the LIST command.

#### **SHOW *command* <No Default>**

SHOW displays the plot parameter settings relevant to the *command*. For example, SHOW TMIN displays the time step selection parameter. This command must be issued within a subprogram (not at the BLOT level).

SHOW TIMES and SHOW STEPS display the selected time steps.

#### **HELP *command* <no parameter>**

HELP displays information about the program command given as the parameter. If no parameter is given, all the command verbs are displayed. HELP is system-dependent and may not be available on some systems.

### **4.1.3 Time Step Selection (any subprogram)**

These commands control the time step selection as explained in Section [Ref: intro:timemode] .

The following are the time step selection parameters:

*tmin* is the minimum selected time,

- *tmax* is the maximum selected time,
- *nintv* is the number of selected time intervals, and
- *delt* is the selected time interval.

In the interval-times mode, up to *nintv* time steps at interval *delt* between *tmin* and *tmax* are selected. The mode may have a delta offset or a zero offset. With a delta offset, the first selected time is *tmin+delt*; with a zero offset, it is *tmin*.

In the interval-times mode with a delta offset, the number of selected time intervals *nintv* and the selected time interval *delt* are related mathematically by the equations:

$$delt = (tmax-tmin) / nintv \quad (1)$$

$$nintv = \text{int} ((tmin-tmax) / delt) \quad (2)$$

With a zero offset, *nintv* and *delt* are related mathematically by the equations:

$$delt = (tmax-tmin) / (nintv-1) \quad (1)$$

$$nintv = \text{int} ((tmin-tmax) / delt) + 1 \quad (2)$$

The user specifies either *nintv* or *delt*. If *nintv* is specified, *delt* is calculated using equation 1. If *delt* is specified, *nintv* is calculated using equation 2.

In the all-available-times mode, all database time steps between *tmin* and *tmax* are selected (parameters *nintv* and *delt* are ignored). In the user-selected-times mode, the specified times are selected (all parameters are ignored).

The initial mode is the interval-times mode with a delta offset. Parameters *tmin*, *tmax*, and *nintv* are set to their default values and *delt* is calculated.

### **TMIN *tmin* <minimum database time>**

TMIN sets the minimum selected time *tmin* to the specified parameter value. If the user-selected-times mode is in effect, the mode is changed to the all-available-times mode.

In interval-times mode, if *nintv* is selected (by a NINTV or ZINTV command), *delt* is calculated. If *delt* is selected (by a DELTIME command), *nintv* is calculated.

### **TMAX *tmax* <maximum database time>**

TMAX sets the maximum selected time *tmax* to the specified parameter value. If the user-selected-times mode is in effect, the mode is changed to the all-available-times mode.

In interval-times mode, if *nintv* is selected (by a NINTV or ZINTV command), *delt* is calculated. If *delt* is selected (by a DELTIME command), *nintv* is calculated.

### **NINTV *nintv* <10 or the number of database time steps - 1,> whichever is smaller**

NINTV sets the number of selected time intervals *nintv* to the specified parameter value and changes the mode to the interval-times mode with a delta offset. The selected time interval *delt* is calculated.

**ZINTV *nintv* <10 or the number of database time steps,> whichever is smaller**

ZINTV sets the number of selected time intervals *nintv* to the specified parameter value and changes the mode to the interval-times mode with a zero offset. The selected time interval *delt* is calculated.

**DELTIME *delt* <(tmax-tmin) / (nintv-1), where *nintv* is 10> or the number of database time steps, whichever is smaller**

DELTIME sets the selected time interval *delt* to the specified parameter value and changes the mode to the interval-times mode with a zero offset. The number of selected time intervals *nintv* is calculated.

**ALLTIMES**

ALLTIMES changes the mode to the all-available-times mode.

**TIMES [ADD,] *t*<sub>1</sub>, *t*<sub>2</sub>, ... <no times selected>**

TIMES changes the mode to the user-selected-times mode and selects times *t*<sub>1</sub>, *t*<sub>2</sub>, etc. The closest time step from the database is selected for each specified time.

Normally, a TIMES command selects only the listed time steps. If ADD is the first parameter, the listed steps are added to the current selected times. Any other time step selection command clears all TIMES selected times.

Up to the maximum number of time steps in the database may be specified. Times are selected in the order encountered on the database, regardless of the order the times are specified in the command. Duplicate references to a time step are ignored.

**STEPS [ADD,] *n*<sub>1</sub>, *n*<sub>2</sub>, ... <no steps selected>**

The STEPS command is equivalent to the TIMES command except that it selects time steps by the step number, not by the step time.

**HISTORY ON or OFF <ON if allowed for current> subprogram**

HISTORY controls whether history time steps are included in the selected time steps (if ON) or only whole time steps (if OFF). This command is only valid in subprograms which can process history variables (e.g., TPLOT).

For example, if the times from the database are 0.0, 0.5, 1.0, 1.5, etc.\null , the commands

```
TMIN 0.0 TMAX 5.0 NINTV 5
```

select times 1.0, 2.0, 3.0, 4.0, and 5.0. If the NINTV command is replaced by

```
ZINTV 3
```

then times 0.0, 2.5, and 5.0 are selected. If the NINTV command is replaced by

DELTIME 2.0

then times 0.0, 2.0, 4.0 are selected.

#### **4.1.4 Graphics Options (any subprogram)**

##### **QA ON or OFF <ON,> MESH or XY <current subprogram type>**

QA controls whether the QA information is displayed on the plot legend (if ON) or not (if OFF). The QA information includes the database title and the creation and modification information. The line bordering the display is also included or omitted. This command controls the mesh plot legend if the second parameter is MESH or the curve plot legend if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

##### **AXIS ON or OFF <ON, initially OFF in 3D,> MESH or XY <current subprogram type>**

AXIS controls whether the plot axes are numbered (if ON) or not (if OFF). This command controls the mesh plot axis if the second parameter is MESH or the curve plot axis if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

##### **LEGEND ON or OFF <ON,> MESH or XY <current subprogram type>**

LEGEND controls whether the plot legend (excluding the QA information) is displayed on the plot (if ON) or not (if OFF). This command controls the mesh plot legend if the second parameter is MESH or the curve plot legend if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

##### **CAPTION *line\_number* <0,> MESH or XY <current subprogram type>**

CAPTION sets the three-line plot caption (up to 80 characters in each line) which is displayed at the bottom of the plot, with each line centered. The *line\_number* is the number of the line to be changed. If *line\_number* is zero, all three lines are changed. The caption line(s) must follow on the next line(s). If no caption is defined, the plot caption is blank.

This command controls the mesh plot caption if the second parameter is MESH or the curve plot caption if the parameter is XY. The XY parameter is only valid if issued from an X-Y curve subprogram.

The initial setting is no caption defined.

##### **SOFTCHAR ON or OFF <ON,> *device* <1>**

SOFTCHAR sets the plot character output mode to software characters (if ON)

or hardware characters (if OFF). Hardware characters are drawn much faster, but may produce inferior lettering on some devices. To change the character type for a secondary device, parameter *device* must be 2.

The initial setting is software characters for a hardcopy device or hardware characters for an interactive device.

This parameter is not changed by a RESET command or by switching subprograms.

### **FONT STICK or SANSERIF or ROMAN <STICK,> *device* <1>**

FONT sets the font to be used for all software characters on a plot. The available fonts are stick, sanserif, and Roman. The stick font is much faster than the other two fonts. To change the font parameter for a secondary device, parameter *device* must be 2.

This parameter is not changed by a RESET command or by switching subprograms.

### **COLOR *ncol* <last selection>**

COLOR sets the maximum number of standard color scale colors to use on a color plot to *ncol* (up to a maximum of 6 colors). The standard color scale colors are red, green, yellow, blue, magenta, and cyan.

The initial setting is the maximum number of colors available or six, whichever is smaller.

This parameter is not changed by a RESET command or by switching subprograms.

\cmdnote DETOUR

COLOR sets the number of contours to the number of colors if the spectrum color scale is undefined.

### **SPECTRUM *ncol* <5, initially 0>**

SPECTRUM sets the maximum number of spectrum colors to use on a color plot to *ncol* (up to a maximum of 256-8 colors). The color table starting after color 8 is changed to a spectrum from blue to red. If *ncol* is zero, the spectrum color scale is undefined and the standard color scale is used.

The use of the spectrum versus the standard color scale is explained in Chapter [Ref: chap:output] .

This parameter is not changed by a RESET command or by switching subprograms.

\cmdnote DETOUR

SPECTRUM sets the number of contours to the number of colors.

### **SNAP *nsnap* <1, initially 0>**

SNAP sets the number of frames to be snapped on a camera device for each requested plot. The camera device must be “connected” in a system-dependent fashion before the program is run.

This parameter is not changed by a RESET command or by switching subprograms.

### **AUTO ON or OFF <ON, initially OFF>**

AUTO controls whether a response is requested from the user after each plot is completed on an interactive graphics terminal (if OFF) or whether all plots are displayed without user intervention (if ON). The responses expected are explained under the PLOT command.

This parameter is not changed by a RESET command or by switching subprograms.

#### **4.1.5 Plot Set Display (any subprogram)**

### **PLOT**

PLOT generates the plots requested for the current plot set on the primary graphics device. No parameters are changed when the program returns to accept commands for the next plot set.

If the program is running interactively on a graphics terminal, the user is prompted for a response after each plot. The program expects one of the following responses:

- RETURN displays the next plot.
- “Q” (for quit) aborts the plot set and the program returns immediately to command input mode.
- “H” (for hardcopy) displays the plot shown on the graphics terminal on the hardcopy device (if any) and displays the next plot on the graphics terminal.
- “T” (for text) prompts for a location and a text string. The location is selected with the graphics cursor. The key pressed to set the position determines the text location. If the key is “C”, the text string is centered over the selected location, otherwise the string is left-justified over the location. The text string is displayed on the graphics terminal and stored for output on the hardcopy device if a hardcopy plot is requested. The program again requests a response for the plot. Thus, the plot can be annotated with several text strings and then displayed on a hardcopy device.

When running interactively, the user can abort the current plot and return immediately to command input mode by issuing a system-defined interrupt (for

example, CONTROL-C on the VAX).

If the program is running in batch mode or if the AUTO parameter is on, the program plots without user intervention.

## **HARDCOPY**

HARDCOPY ends the command input for the current plot set and generates the plots requested on the hardcopy device. This command is appropriate only if two graphics devices are active. No parameters are changed when the program returns to accept commands for the next plot set.

## **4.2 Mesh Commands (any mesh or X-Y curve subprogram)**

### **4.2.1 Mesh View Control (any mesh or X-Y curve subprogram)**

These commands set view-dependent parameters (see Section [Ref: mesh:multview] ). They may be used in conjunction with the VIEW command.

The display mode commands (see Section [Ref: det:dismode] ) set these parameters.

## **EMPTY**

EMPTY sets the selected view to empty, indicating that nothing is to be drawn for that view. This command is useful with multiple views.

The only way to change an empty view to a non-empty view is with a DEFORM command or a display mode command.

## **DEFORM ON or OFF <ON>**

DEFORM sets whether the view displays the deformed mesh (if ON) or undeformed mesh (if OFF). The MAGNIFY command controls the deformation of all views.

## **DETOUR**

All display mode commands request the deformed mesh.

## **MESH and PATHLINE**

This command is not valid in these subprograms. The displacement magnification (set by the MAGNIFY command) is always zero and the mesh is always undeformed.

## **NUMBER NODES or ELEMENTS or ALL or OFF <ALL, initially OFF>**

NUMBER sets the mesh numbering for the view. If the parameter is NODES or ALL, the node numbers are displayed on the mesh. If the parameter is ELEMENTS or ALL, the element numbers are displayed on the mesh. Only

“active” elements (see Section [Ref: mesh:actelem] ) and nodes in at least one “active” element are numbered.

All display mode commands turn off the numbering.

### **MLINES ON or DOTTED or OFF <ON>**

MLINES controls whether all mesh lines (for elements in selected element blocks) are displayed (if ON or DOTTED) or only the mesh boundary and element block boundaries (if OFF). If the parameter is DOTTED, the element boundary lines are dotted.

All display mode commands set this parameter: the WIREFRAM and SOLID commands set it ON; all others set it OFF.

For example, the following command sequence:

```
XVIEW YVIEW VIEW 1 WIREFRAM VIEW 2 WIREFRAM VIEW 2
MLINES OFF VIEW 3 CONTOUR SIGR VIEW 3 MLINES DOTTED VIEW
4 CONTOUR SIGR PLOT
```

displays four views as follows:

- the mesh with all mesh lines,
- the mesh outline (the mesh boundary and element block boundaries),
- line contours on the mesh with dotted mesh lines, and
- line contours on the mesh outline.

### **BOUNDARY ON or BLACK or OFF <ON>**

BOUNDARY controls whether the mesh boundary and element block boundaries are drawn in white (if ON) or black (if BLACK) or not drawn (if OFF). It is only valid if the mesh is to be painted. BOUNDARY OFF also sets the MLINES parameter to off; BOUNDARY ON sets the MLINES parameter to the default value for the current display mode.

All display mode commands request the boundary be drawn in white. The BOUNDARY command is only valid if a solid mesh or painted contour display mode is active.

### **NSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... <all node sets> NSETS OFF <initial condition>**

NSETS selects the node sets to be displayed for the view. The side set selection is not affected. Only nodes in at least one “active” element (see Section [Ref: mesh:actelem] ) are displayed.

The *set\_id* is the node set identifier displayed by the LIST NSETS command.

If there is no parameter, all node sets are selected. If the parameter is OFF, no node sets are selected. If the first parameter is ADD, the sets listed are added to the current selected set. Otherwise, only the sets listed in the command are selected.

All display mode commands set no node sets selected.

**SSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... <all side sets>**  
**SSETS OFF <initial condition>**

The SSETS command is equivalent to the NSETS command except that it selects side sets instead of node sets. Only “active” elements (see Section [Ref: mesh:actelem] ) are displayed.

The LIST SSETS command displays the side set identifiers.

#### **4.2.2 Active Element Control (any mesh or X-Y curve subprogram)**

These commands select active elements as described in Section [Ref: mesh:actelem] .

Issued from DETOUR:

Only active elements are included in the calculation of the default contour parameters (see Section [Ref: det:contour] ) for an element variable. When an active element control command is issued and contour mode of an element variable is selected, the contour parameters are recalculated unless a contour control command was issued after the CONTOUR or PAINT command. A CONTOUR or PAINT command with the variable as a parameter forces the recalculation of the contour parameters with the new active elements.

**BLOCKS [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... <all element blocks>**  
**BLOCKS OFF**

BLOCKS selects the element blocks for active elements. Only elements in selected element blocks contribute to any variable plot (e.g., a contour plot). On a mesh plot, lines separating elements in the same element block (in some display modes) are only drawn for elements in a selected element block.

The *block\_id* is the element block identifier displayed by the LIST BLOCKS command.

If there is no parameter, all element blocks are selected. If the parameter is OFF, no element blocks are selected. If the first parameter is ADD or DELETE, the element blocks listed are added to or deleted from the current selected set. Otherwise, only the element blocks listed in the command are selected.

**MATERIAL [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... <all element blocks>**  
**MATERIAL OFF**

MATERIAL is exactly equivalent to a BLOCKS command.

**VISIBLE [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... <all element blocks>**

VISIBLE selects the element blocks that will be displayed in 3D. An element block that is not displayed is totally ignored in all plots and calculations.

The *block\_id* is the element block identifier displayed by the LIST BLOCKS command.

If there is no parameter, all element blocks are visible. If the first parameter is ADD or DELETE, the element blocks listed are added to or deleted from the current visible set. Otherwise, only the element blocks listed in the command are visible.

**DEATH ON <No Default> , *state\_variable* <last selection>  
DEATH OFF <initial condition>**

DEATH turns on or off the capability to ignore “dead” elements.

**DEATH ON**

allows elements to die and be born at the start of a time step. A “dead” element is ignored in all calculations and plots, including a simple mesh plot. “Dead” nodes can be displayed with the DEADNODE command. The *state\_variable* is an element database variable that gives the state of each element for each time step (zero if alive, otherwise dead).

The initial default *state\_variable* name is DEATH.

**DEATH OFF**

causes all elements to be alive for all time steps.

**MESH and PATHLINE**

This command is not valid in these subprograms.

**DEADNODE ON or OFF <ON>**

DEADNODE determines whether “dead” nodes are to be displayed on all mesh plots (if ON) or not (if OFF). A dead node is a node which is not connected to any element which is alive. This parameter is active only if element birth/death is active (see the DEATH command). This command does not affect the contour calculation.

**MESH and PATHLINE**

This command is not valid in these subprograms.

### 4.2.3 Multiple Views Control (any mesh or X-Y curve subprogram)

These commands control multiple views on one plot as explained in Section [Ref: mesh:multview] . The initial setting is one view. Symmetric views are not allowed in 3D.

#### **VIEW *nview* <No Default> , *view\_dependent\_command* <No Default>**

VIEW sets a view-dependent parameter of view *nview* (1..4). If *nview* is zero, the parameter is set for all defined views. This command **VIEW** must be used to set any view-dependent parameter if there is more than one view defined. A command is view-dependent and sets a view-dependent parameter if and only if the command description states that it may be used with VIEW command.

#### **XVIEW ON or OFF <ON, initially OFF>**

XVIEW defines (if ON) or deletes (if OFF) non-symmetric vertically divided views. If the parameter is ON, new views are created to the left of the existing views, and the new views have the view-dependent parameters of the existing views beside them. If the parameter is OFF, the left views are deleted.

#### **YVIEW ON or OFF <ON, initially OFF>**

YVIEW defines (if ON) or deletes (if OFF) non-symmetric horizontally divided views. If the parameter is ON, the new views are created below the existing views, and the new views have the view-dependent parameters of the existing views above them. If the parameter is OFF, the lower views are deleted.

#### **XSYM LEFT or RIGHT <last selection,> *xsym* <last selection> XSYM OFF <initial condition>**

XSYM defines or deletes symmetric vertically divided views.

#### **XSYM LEFT or RIGHT**

defines the symmetric vertically divided views. The LEFT/RIGHT parameter determines if the new views are created to the left or right of the existing views. The existing views are reflected around the vertical symmetry axis along the horizontal coordinate *xsym*. The new views have the view-dependent parameters of the existing views beside them (unless vertically divided views are already defined).

If the LEFT/RIGHT parameter changes, *xsym* defaults to the minimum horizontal mesh coordinate (if LEFT) or to the maximum horizontal mesh coordinate (if RIGHT). Otherwise *xsym* defaults to its last selected value. The initial default is LEFT with *xsym* at the minimum horizontal coordinate value.

#### **XSYM OFF**

deletes the symmetric vertically divided views defined by a previous XSYM command. The left views are deleted if the views were defined with XSYM LEFT;

otherwise the right views are deleted (and the left views are renumbered).

**YSYM BOTTOM or TOP <last selection,> *ysym* <last selection>**  
**YSYM OFF <initial condition>**

YSYM defines or deletes symmetric horizontally divided views.

YSYM BOTTOM or TOP

defines the symmetric horizontally divided views. The BOTTOM/TOP parameter determines if the new views are created above or below the existing views. The existing views are reflected around the horizontal symmetry axis along the vertical coordinate *ysym*. The new views have the view-dependent parameters of the existing views above or below them (unless horizontally divided views are already defined).

If the BOTTOM/TOP parameter changes, *ysym* defaults to the minimum vertical mesh coordinate (if BOTTOM) or to the maximum vertical mesh coordinate (if TOP). Otherwise *ysym* defaults to its last selected value. The initial default is BOTTOM with *ysym* at the minimum vertical coordinate value.

YSYM OFF

deletes the symmetric horizontally divided views defined by a previous YSYM command. The lower views are deleted if the views were defined with YSYM BOTTOM; otherwise the upper views are deleted (and the lower views are renumbered).

**MULTTIME ON or OFF <ON, initially OFF>**

MULTTIME ON assigns a different time step to each non-empty view. Thus, if four views are defined, the first four selected times appear on the first plot (in the order that the views are ordered), the next four times are on the second plot, etc.

MULTTIME OFF assigns a single time step to all defined views.

MESH and PATHLINE

This command is not valid in these subprograms.

**4.2.4 Mesh Control (any mesh or X-Y curve subprogram)**

**MAGNIFY *dfac* <5% of mesh range divided by maximum displacement>**  
**component**

MAGNIFY causes the displacements to be magnified by *dfac* before they are added to the coordinates. If the parameter is zero, the mesh is not deformed.

MESH and PATHLINE

This command is invalid in these subprograms. The magnification is zero and may

not be set.

When switching from a subprogram which cannot display the deformed mesh to one that can, the magnification is set to the last non-zero value for this parameter.

### **HIDDEN *level* <3>**

HIDDEN controls the 3D hidden line and surface algorithm. The hidden *level* is defined below.

- 0 indicates that hidden lines are not removed. This option is not valid with painted surfaces.
- 1 requests a simple hidden surface algorithm that removes any surface (and its lines) whose normal faces away from the viewer.
- 2 requests the normal hidden node algorithm, and removes any surface (and its lines) if any node of the surface is hidden.
- 3 requests the normal hidden node algorithm and displays partial lines. A surface is considered hidden if any of its nodes are hidden.
- 4 requests the algorithm used for level 3. On painted plots, all surfaces with some hidden nodes are ordered by the Z coordinates and processed as visible surfaces.

### **ZOOM *hmin, hmax, vmin, vmax* <deformed mesh limits + 5% of range> or CURSOR**

#### **ZOOM TRANSLAT *hcen, vcen* <unchanged> or CURSOR**

#### **ZOOM EACH <default for 3D>**

#### **ZOOM MESH <default for 2D>**

ZOOM sets the mesh window limits.

#### **ZOOM *hmin, hmax, vmin, vmax* or CURSOR**

sets the horizontal mesh window limits to *hmin* and *hmax* and the vertical limits to *vmin* and *vmax*. If the first parameter is a number, the mesh limits are entered as parameters. If the first parameter is CURSOR, the mesh limits are selected with the graphics cursor. The left, bottom position (*hmin, vmin*) is selected first, then the right, top position (*hmax, vmax*).

#### **ZOOM TRANSLAT *hcen, vcen* or CURSOR**

moves the current window limits to that the window is centered on *hcen, vcen*. If the second parameter is CURSOR, the center is selected with the graphics cursor.

#### **ZOOM EACH**

causes the window limits to be reset for each plot to the smallest (expanded) window that encloses the displayed mesh.

#### **ZOOM MESH**

sets the window limits to the (expanded) deformed mesh limits or the undeformed mesh limits if all views are undeformed.

If the **SQUARE** parameter is on, the mesh window defaults to a square or two rectangles forming a square (see Section [Ref: mesh:multview] ). To accomplish this, the shorter of the horizontal or vertical range is expanded with the specified range centered on the window. If the **SQUARE** parameter is off, the specified limits exactly define a rectangular window.

### **SQUARE ON or OFF <ON>**

**SQUARE** determines if the mesh window will be square or rectangular. If the parameter is **ON**, the window limits set by the **ZOOM** command are expanded to form a square or two rectangles forming a square (see Section [Ref: mesh:multview] ). If the parameter is **OFF**, the limits of the mesh window are the limits set by the **ZOOM** command (forming a rectangular window).

### **TICK *tick* <0.0>**

**TICK** sets the axis tick interval for both mesh axes to *tick*. If *tick* is zero, the tick interval is automatically calculated from the mesh window limits.

Automatic calculation is selected when a **ZOOM** command is issued.

## **4.2.5 3D Rotation Commands (any mesh or X-Y curve subprogram)**

### **ROTATE *axis*<sub>1</sub>, *ndeg*<sub>1</sub>, *axis*<sub>2</sub>, *ndeg*<sub>2</sub>, ... <last selection> ROTATE RESET <initial condition>**

**ROTATE** rotates the 3D mesh. Each (*axis*, *ndeg*) parameter pair specifies an axis (X or Y or Z) and the number of degrees to rotate. The axis refers to the screen axis, not to the object axis. The rotations are according to right-hand rule.

Rotations are cumulative. The **ROTATE RESET** command resets to no rotation. The **EYE** command also sets the mesh rotation.

The rotation is not changed by switching subprograms, but it is reset by a **RESET** command.

**EYE** *xpos*, *ypos*, *zpos* <last selection>

**EYE** rotates the 3D mesh so that the viewer's "eye" position is at (*xpos*,*ypos*,*zpos*) (in object coordinates). The **ROTATE** command also sets the mesh rotation. The current **EYE** position is displayed after each **ROTATE** command.

The rotation is not changed by switching subprograms, but it is reset by a **RESET** command.

### **CENTER *xcen*, *ycen*, *zcen* <last selection>**

CENTER moves the 3D center of rotation to  $(xcen,ycen,zcen)$  (in object coordinates).

The center of rotation is not changed by switching subprograms, but it is reset by a RESET command.

## 4.3 DETOUR Commands ( DETOUR subprogram)

### 4.3.1 Mesh Display Mode Control ( DETOUR subprogram)

These commands set the display mode as explained in Section [Ref: mesh:detour] . The display mode is a view-dependent parameter (see Section [Ref: mesh:multview] ). The display mode commands may be used in conjunction with the VIEW command. One and only one display mode is active for each view. Thus, the display mode commands counteract one another. The initial display mode is wireframe mesh mode.

Modes which reference a variable use only values for “active” elements (see Section [Ref: mesh:actelem] ). The specified variable applies to all views (see Section [Ref: mesh:multview] ).

Display mode commands affect the mesh view control parameters (see Section [Ref: xmesh:meshview] ).

### WIREFRAM

WIREFRAM sets the view to wireframe mesh mode, which displays the mesh lines.

### SOLID

SOLID sets the view to solid mesh mode, which paints each element, using a different color for each element block. All visible element blocks are painted, even if the block is not selected.

### CONTOUR *variable* <last selected variable>

CONTOUR sets the view to line contour mode, which plots contours of the *variable* on the mesh.

The *variable* may be either a nodal or element database variable. Contours are plotted for nodal values only. An element variable is converted to a nodal variable by averaging the values for all active elements containing the node.

If a variable is specified, the contour parameters are recalculated as explained in Section [Ref: det:contour] . The contour parameters are also recalculated if the current parameters were calculated with a PAINT command, unless a contour control command was issued after the PAINT command.

### PAINT *variable* <last selected variable>

PAINT sets the view to paint contour mode, which paints contours of the *variable* on the mesh.

The *variable* may be either a nodal or element database variable. Contours are plotted for nodal values only. An element variable is converted to a nodal variable by averaging the values for all active elements containing the node.

If a variable is specified, the contour parameters are recalculated as explained in Section [Ref: det:contour] . The contour parameters are also recalculated if the current parameters were calculated with a CONTOUR command, unless a contour control command was issued after the CONTOUR command.

**VECTOR *x\_variable* or 0, *y\_variable* or 0, *z\_variable* or 0 (if 3D) <last selected variables>**

VECTOR sets the view to vector mode, which plots a vector representing the components *x\_variable*, *y\_variable*, and *z\_variable* (for a 3D mesh) on the mesh. The variables may be either nodal or element database variables, but all must be of the same type. If the variables are nodal variables, a vector is plotted at each node. If the variables are element variables, a vector is plotted at the center of each active element. If a parameter is 0, the corresponding vector component is zero (a scalar plot).

The vector lengths are controlled by the vector/symbol scale factor (see the VSCALE command). This factor is set to 1.0 by the VECTOR command.

**SYMBOL CRACK or ANGLE <No Default> , *variable* <last selected variable>**

SYMBOL sets the view to element symbol mode. This special purpose option allows the user to plot a symbol on the mesh at the center of each active element.

The *variable* is an element database variable which gives a value for each element. If the first parameter is CRACK, DETOUR plots an “X” for a negative value and an “O” for a non-zero positive value (nothing is plotted for a zero value). If the first parameter is ANGLE, DETOUR plots a line at the indicated angle (in degrees) if the angle is between -360 and 360 and an “\*” if the angle is less than -360 (nothing is plotted if the angle is greater than 360). An angle of zero plots a horizontal line; positive angles are measured counterclockwise from the horizontal.

The symbol size may be controlled the vector/symbol scale factor (see the VSCALE command). This factor is set to 1.0 by the SYMBOL command.

**variable**

A database variable name is equivalent to a display mode command if the variable name does not conflict with any command or command abbreviation. If the current display mode is wireframe or solid mesh mode, this command is equivalent to a CONTOUR command. If the display mode references a variable

(e.g., element symbol mode), this command is equivalent to the command which sets the current display mode. For example, if the view is in vector mode, the command

SIGR, SIGZ

is equivalent to the command

VECTOR SIGR, SIGZ.

#### 4.3.2 Contour Control ( DETOUR subprogram)

These commands control a contour plot. The following are the contour parameters:

- *ncntr* is the number of contours,
- *delc* is the contour interval,
- *cmin* is the minimum contour value, and
- *cmax* is the maximum contour value.

DETOUR plots *ncntr* contours at uniform intervals *delc* from *cmin* through *cmax*. The *ncntr* parameter has the highest priority, meaning that all other parameters change with respect to it. The only way to change the number of contours is with the NCNTRS command or the COLOR and SPECTRUM commands.

In line contour mode, the contour parameters are related mathematically by the equation:

$$delc = (cmax - cmin) / (ncntr - 1)$$

In paint contour mode, the contour parameters are related mathematically by the equation:

$$delc = (cmax - cmin) / ncntr$$

The contour parameters may be reset to default values when the contour variable changes or when the active element criteria changes. The number of contours *ncntr* is not changed when the default is calculated. The minimum and maximum database variable values are used to compute a rounded default contour interval *delc*. In line contour mode, the default contour limits *cmin* and *cmax* are approximately one-half *delc* inward from the database variable limits. In paint contour mode, the default contour limits *cmin* and *cmax* are slightly outside the database variable limits.

If the contour mode changes from line to painted contour mode or from painted to line contour mode and the default is not calculated, the contour maximum *cmax* is recalculated, but the rest of the parameters remain unchanged.

#### NCNTRS *ncntr* <6>

NCNTRS sets the number of contours *ncntr* to the specified parameter value. A new contour interval *delc* is calculated from the variable limits, but it is not rounded.

### **CRANGE *cmin* <last selection,> *cmax* <last selection>**

CRANGE sets the minimum contour value *cmin* and the maximum contour value *cmax* to the specified parameter values. The contour interval *delc* is calculated.

### **DELCNTR *delc* <last selection,> *cmin* <last selection>**

DELCNTR sets the contour interval *delc* and the contour minimum *cmin* to the specified parameter values. The contour maximum *cmax* is calculated.

### **CSHIFT *cval* <*cmin*>**

CSHIFT changes the contour limits *cmin* and *cmax* so that the parameter *cval* falls exactly on the nearest contour line or contour point boundary. The contour interval *delc* is not affected.

## **4.3.3 Mode Specific Options ( DETOUR subprogram)**

### **CLABEL ON or *label\_incr* or OFF <ON>**

CLABEL controls the density of identifying letters on a contour plot. This parameter is active only in line contour mode. If the parameter is OFF, no contour letters appear on the plot. If the parameter is ON or a number, contour letters appear where a contour line intersects the mesh boundary or an element block boundary. If the parameter is a number (*label\_incr*), letters appear where a contour line intersects an element boundary, with the density of these letters decreasing as *label\_incr* increases. For example, if *label\_incr* is 1, a letter appears at every intersection with an element boundary; if *label\_incr* is 4, a letter appears at every  $\backslash$ nth 4th intersection.

### **COPEN ON or OFF <ON, initially OFF,> ON or OFF <ON, initially OFF>**

COPEN controls the plotting of values outside the contour limits on a painted contour plot. The first parameter controls whether values below the minimum parameter value should be painted in the minimum color (if ON) or left unpainted (if OFF). The second parameter controls whether values above the maximum parameter value should be painted in the maximum color (if ON) or left unpainted (if OFF).

### **CSYMBOLS *nmin* <10,> *nmax* <*nmin*>**

CSYMBOLS controls the plotting of symbols at the minimum and maximum nodal variable values on a contour plot and the display of these values on the plot legend. Parameters *nmin* and *nmax* are active only in the contour modes. Symbols are displayed at the minimum (with the values in the legend) only if the number of minimums for the plot is less than or equal to *nmin*. The maximum symbols and values are displayed only if the number of maximums is less than or equal to *nmax*. The number of minimums and maximums includes the entire mesh, not just the the mesh visible in the window. Setting *nmin* or *nmax* to zero disables the

minimum or maximum display.

#### **VSCALE *scale\_factor* <1.0>**

VSCALE sets the vector/symbol scale factor. This parameter is active only in vector or element symbol mode.

In vector mode, the magnitudes of the vectors are multiplied by *scale\_factor* and by the default vector scaling factor to determine their length. The default scaling factor is five percent of the window coordinate range divided by the maximum absolute value of all the vector component variables. Thus, the vectors maintain the same length in relation to the window size.

In element symbol mode, *scale\_factor* is multiplied by the constant symbol scaling factor to determine the symbol size.

The VECTOR and SYMBOL commands reset this parameter to 1.0.

### **4.4 PATHLINE Commands ( PATHLINE subprogram)**

#### **ADD**

ADD saves the pathlines defined in the last plot set and causes future definitions to be added to the set. This command is ignored if new pathlines have been defined for the plot set.

#### **REMOVE *pathline*<sub>1</sub>, *pathline*<sub>2</sub>, ... <No Default>**

REMOVE deletes the specified pathlines. Each parameter *pathline*<sub>*i*</sub> is the number of a pathline that is to be deleted. The pathline number is displayed with the LIST PLOT command. If *pathline*<sub>*i*</sub> is negative, the last *-pathline*<sub>*i*</sub> pathlines defined are deleted.

#### **LOCATION *x\_variable*, *y\_variable*, *z\_variable* (if 3D), <No Default> *range*<sub>1</sub>, *range*<sub>2</sub>, ... <No Default>**

LOCATION defines a pathline. The coordinates of the pathline are represented by the values of the specified variables. The variables may be history, global or nodal variables, but all must be of the same type. If the variables are nodal variables, a set of nodes must also be selected. Each *range* must be in the range form described under Command Input.

The first LOCATION command issued for a plot set starts a new set of pathlines (unless an ADD command is entered first). All following LOCATION commands are cumulative.

## 4.5 X-Y Curve Commands (any X-Y curve subprogram)

### 4.5.1 Curve Definition (any X-Y curve subprogram)

#### ADD

ADD saves the curves defined in the last plot set and causes future definitions to be added to the set. This command is ignored if new curves have been defined for the plot set.

#### REMOVE *curve*<sub>1</sub>, *curve*<sub>2</sub>, ... <No Default>

REMOVE deletes the specified curves. Each parameter *curve*<sub>*i*</sub> is the number of a curve that is to be deleted. The curve number is displayed with the LIST PLOT command. If *curve*<sub>*i*</sub> is negative, the last *-curve*<sub>*i*</sub> curves defined are deleted.

### 4.5.2 Axis Labeling and Scaling (any X-Y curve subprogram)

The axis label, scaling, and tick interval may be reset to their default values when the axis changes.

#### TPLOT

The X axis changes when a new X variable is requested. The Y axis changes when a new Y variable is requested. New variables are requested with a TYPLOT or XYPLOT command.

#### SPLOT

The X axis changes when a NODES or ELEMENTS command is issued. The Y axis changes when an SPLOT command is issued.

#### XLABEL

XLABEL sets the label (up to 40 characters) to be centered below the X axis. The label must follow on the next line. If no label is defined, a default is provided.

#### TPLOT

For time history plots, the default X axis label is "TIME". For variable-versus-variable plots, if all curves on the plot reference the same X variable, the default is the variable name; otherwise no label is provided.

#### SPLOT

The default X axis label is "DISTANCE".

The default label is selected when the X axis changes if no XLABEL command has been issued for this plot set.

## **YLABEL**

YLABEL sets the label (up to 40 characters) to be centered to the left of the Y axis. The label must follow on the next line. If no label is defined, a default is provided.

## **TPLOT**

If all curves on the plot reference the same Y variable, the default Y axis label is the variable name; otherwise no label is provided.

## **SPLOT**

If there is only one curve on the plot, the default Y axis label is the curve variable name; otherwise no label is provided.

The default label is selected when the Y axis changes if no YLABEL command has been issued for this plot set.

## **XSCALE *xmin* <last selection,> *xmax* <last selection>**

XSCALE sets the X axis minimum and maximum values to *xmin* and *xmax*. If neither value is given, the axis is automatically scaled from the data.

Automatic scaling is selected when the X axis changes if no XSCALE command has been issued for this plot set.

## **YSCALE *ymin* <last selection,> *ymax* <last selection>**

YSCALE sets the Y axis minimum and maximum values to *ymin* and *ymax*. If neither value is given, the axis is automatically scaled from the data.

Automatic scaling is selected when the Y axis changes if no YSCALE command has been issued for this plot set.

## **XTICK *xtick* <0.0>**

XTICK sets the X axis tick interval to *xtick*. If *xtick* is zero, the tick interval is automatically calculated from the data.

Automatic calculation is selected when the X axis changes if no XTICK command has been issued for this plot set.

## **YTICK *ytick* <0.0>**

YTICK sets the Y axis tick interval to *ytick*. If *ytick* is zero, the tick interval is automatically calculated from the data.

Automatic calculation is selected when the Y axis changes if no YTICK command has been issued for this plot set.

## **SAMESCAL ON or OFF <ON, initially OFF>**

**SAMESCAL** controls whether individual curves in the plot set are to have the same scale (if **ON**) or be scaled individually (if **OFF**). This parameter has no meaning if the axis scales have been set with the **XSCALE** and **YSCALE** commands. A **SAMESCAL** command overrides the **NORMAL** parameter.

**TPLOT**: If the curves are overlaid, they all have the same scale.

**SPLIT**: Since the distances are constant over the entire plot set, the X axis always has the same scale. If the parameter is **ON**, the curves for all variables and all times have the same scale.

If the parameter is **OFF** and the curves are overlaid, all the curves on one plot have the same scale, but each variable (if times are overlaid) or each time step (if variables are overlaid) is scaled individually.

#### **NORMAL ON or OFF <ON, initially OFF>**

**NORMAL** controls whether individual curves on a plot are to have the same scale (if **OFF**) or be scaled individually (if **ON**). If the parameter is **OFF**, the scale for each curve is normalized to the same scale (0..1). This parameter has no meaning if the axis scales have been set with the **XSCALE** and **YSCALE** commands. A **NORMAL** command overrides the **SAMESCAL** parameter.

**SPLIT**: Since the distances are constant over the entire plot set, the X axis always has the same scale.

#### **RATIOXY *ratio* <1.0>**

**RATIOXY** sets the X-to-Y axis length ratio to *ratio*. For example, if *ratio* is 0.5, the X axis is half as long as the Y axis.

### **4.5.3 Curve Display Options (any X-Y curve subprogram)**

#### **OVERLAY VARIABLE (ON) or TIME or OFF <VARIABLE, initially OFF>**

**OVERLAY** controls whether curves are overlaid on one plot. If the parameter is **VARIABLE**, the curves for all variables are displayed on one plot for each time step. If the parameter is **TIME**, the curves for all time steps are displayed on one plot for each requested variable. If the parameter is **OFF**, each curve is displayed on a separate plot.

**TPLOT**: The **TIME** option is not valid. If the parameter is **VARIABLE**, all curves are displayed on one plot.

**SPLIT**: If the parameter is **VARIABLE**, the curves for all variables are displayed on one plot for each time step. If the parameter is **TIME**, the curves for all time steps are displayed on one plot for each requested variable.

#### **GRID ON or OFF <ON, initially OFF>**

GRID controls whether a grid is drawn on the plot (if ON) or not (if OFF). The placement of the grid lines may be controlled with the XTICK and YTICK commands.

#### **LINES ON or VARY or OFF <ON>**

LINES controls the type of line plotted for each curve. If the parameter is ON, a solid line is plotted. If the parameter is VARY, the line type changes for each curve on a plot. The line type cycles through six types of lines: solid, dotted, dot dash, short dash, long dash, and medium dash. If the parameter is OFF, only symbols are displayed (see the SYMBOLS command).

This parameter is set ON if it is OFF and a SYMBOLS OFF command is issued.

#### **SYMBOLS VARY or *symbol\_number* or OFF <VARY, initially OFF>**

SYMBOLS controls whether symbols are plotted at each data point on a curve. If the parameter is VARY (or ON), symbols are plotted, with the symbol changing for each curve on a plot. The symbol cycles through the six symbols listed below. If the parameter is OFF, only lines are plotted (see the LINES command). If the parameter is a number, the symbol corresponding to *symbol\_number* is plotted for all curves.

The symbols with their identifying numbers are:

1) square 2) diamond 3) cross 4) x 5) triangle 6) circle

This parameter is set to VARY if it is OFF and a LINES OFF command is issued.

#### **CRVNUM FIRST or MIDDLE or LAST or OFF <LAST>**

CRVNUM controls where the curves are numbered. If the parameter is FIRST, the number appears to the left of the first point of each curve. If the parameter is MIDDLE, the number appears over the middle point of each curve. If the parameter is LAST, the number appears to the right of the last point of each curve. If the parameter is OFF, no curve numbers appear.

Curves are only numbered if the number of line types or symbols selected is less than the number of curves on the plot, regardless of the setting of this parameter.

#### **4.5.4 Neutral File Options (any X-Y curve subprogram)**

The neutral file format is the format specified for input to the GRAFAID plot program [bib:grafaid]. Each defined curve is written to the neutral file as a set of X-Y pairs. Some descriptive information is also written with each curve. Many commands (e.g., XSCALE and XLABEL) affect the output of this descriptive information. Appendix [Ref: appx:grafaid] describes the format in detail.

A name for each curve is written to the neutral file. This name can be used in GRAFAID to reference the curve. The neutral file curve name consists of a character string with a number appended. The user may set the string and the number.

### **ACURVE *acurve* <blank>**

ACURVE sets the name part of the neutral file curve name to *acurve* (up to 8 characters). The neutral file curve name is the curve name with the curve number (set by the NCURVE command) appended. This name is written to the neutral file.

This parameter is not changed by a RESET command or by switching subprograms.

### **NCURVE *ncurve* <1,> *incrv* <1>**

NCURVE sets the starting neutral file curve number to *ncurve* and the increment for this number to *incrv*. The neutral file curve name is the curve name (set by the ACURVE command) with the curve number appended. This name is written to the neutral file. The curve number is incremented after each curve is written to the neutral file.

This parameter is not changed by a RESET command or by switching subprograms.

## **NEUTRAL**

NEUTRAL ends the command input for the current plot set and writes the defined curves to the neutral file. No parameters are changed when the program returns to accept commands for the next plot set.

For example, the TPLOTT commands

```
SIGX 50 TO 250 BY 50 ACURVE SIGX NCURVE 50,50 NEUTRAL
```

assign the name SIGX50 to the first curve, SIGX100 to the second curve, etc.

### **4.5.5 Mesh Display (any X-Y curve subprogram)**

## **MESH**

MESH displays the mesh from an X-Y curve subprogram using the plot parameter settings defined for the MESH subprogram. The user may be prompted for a response after the mesh is displayed. The responses expected are explained under the PLOT command.

## 4.6 TPLOT Commands ( TPLOT subprogram)

### 4.6.1 Curve Definition ( TPLOT subprogram)

#### TYPLOT

TYPLOT defines a time history curve. The variable with its list of node or element ranges (if the variable is a nodal or element variable) must follow on the next line. The times are specified by the time step selection commands. The first TYPLOT or XYPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following TYPLOT and XYPLOT commands are cumulative.

Each node or element range must be in the range form described under Command Input. One TYPLOT command may define several curves of different nodes or elements of the same variable. For example, the command

```
TYPLOT SIGR 56, 40 TO 20 BY -10, 5
```

defines five curves of variable SIGR for elements 56, 40, 30, 20, and 5, in that order.

TYPLOT may reset the XLABEL, YLABEL, XSCALE, YSCALE, XTICK, and YTICK parameters to their default values.

**variable node\_range<sub>1</sub> or element\_range<sub>1</sub>, node\_range<sub>2</sub> or element\_range<sub>2</sub>, ...  
<no number>**

A database variable name with its list of node or element ranges (if the variable is a nodal or element variable) is equivalent to a TYPLOT command if the variable name does not conflict with any command or command abbreviation. For example, the command

```
SIGMAX, 1, 5
```

is equivalent to the command

```
TYPLOT SIGMAX, 1, 5.
```

#### XYPLOT

XYPLOT defines a variable-versus-variable curve. The X variable with its list of node or element ranges (if the variable is a nodal or element variable) must follow on the next line. The Y variable with its list of node or element ranges (if the variable is a nodal or element variable) must be on the following line. The first TYPLOT or XYPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following TYPLOT and XYPLOT commands are cumulative.

Each X and Y node or element range must be in the range form described under

Command Input. One XYPLOT command may define several curves of different nodes or elements of one X and one Y variable. If multiple nodes or elements are requested on both the X and Y variables, the number of nodes or elements selected for the X and Y variables must be equal. For example, the command

```
XYPLOT SIGMAX, 1 SIGZ, 10 TO 20 BY 5, 30
```

defines four curves of SIGMAX at element 1 along the X axis versus SIGZ at elements 10, 15, 20, and 30 along the Y axis.

XYPLOT may reset the XLABEL, YLABEL, XSCALE, YSCALE, XTICK, and YTICK parameters to their default values.

## 4.7 SPLOT Commands ( SPLOT subprogram)

### 4.7.1 Curve Definition ( SPLOT subprogram)

**NODES [ADD,] [PATH,] *node\_range*<sub>1</sub>, *node\_range*<sub>2</sub>, ... <No Default>**

NODES selects the nodes which define the plotting distances and the variable values to plot. A NODES or ELEMENTS command must be issued before an SYPLOT command.

Each *node\_range* must be in the range form described under Command Input. For example, NODES 40 TO 20 BY -10, 7 selects nodes 40, 30, 20, and 7, in that order.

If ADD is the first parameter, the listed nodes are added to the current selection. A NODES command with the ADD parameter is preferable to continuing a NODES command over two lines.

If the parameter PATH proceeds any node number, nodes are selected which form a “path” between each pair of specified nodes. The next node along the path is the node connected to the current path node that is closest to the line connecting the path endpoints and closer to the endpoint than the current node. Connected nodes are in the same element (including nodes which are diagonal from each other).

The existing defined curves are deleted if and only if the selected curve variables are not nodal variables.

NODES may reset the XLABEL, XSCALE, and XTICK parameters to their default values.

**ELEMENTS [ADD,] [PATH,] *element\_range*<sub>1</sub>, *element\_range*<sub>2</sub>, ... <No Default>**

The ELEMENTS command is equivalent to the NODES command except that it selects elements instead of nodes. Element paths may be defined. Connected elements have at least one node in common.

**SYPLOT *variable* <No Default>**

SYMPLOT defines a distance-versus-variable curve. A NODES or ELEMENTS command must be issued before this command. The *variable* must be a nodal database variable if nodes are selected or an element database variable if elements are selected. The first SYMPLOT command issued for a plot set starts a new set of curves (unless an ADD command is entered first). All following SYMPLOT commands are cumulative.

SYMPLOT may reset the YLABEL, YSCALE, and YTICK parameters to their default values.

### **variable**

A database variable name is equivalent to a SYMPLOT command if the variable name does not conflict with any command or command abbreviation. For example, the command

SIGMAX

is equivalent to the command

SYMPLOT SIGMAX.

### **4.7.2 Mesh Display ( SPLOT subprogram)**

#### **ECHO**

ECHO displays the mesh showing the selected nodes or elements. A NODES or ELEMENTS command must be issued before this command. The mesh window is scaled to form a box around the selected nodes or elements. The selected nodes or elements are numbered and connected by arrows.



## 5 Plot Output

### 5.1 Mesh Output

Each plot is displayed on the graphics device. The user may be prompted for a response after each plot (see the PLOT command).

For plotting purposes, a mesh line is classified as one of three types:

- a mesh boundary line, which is part of only one element,
- an element block boundary line, which is shared by elements in different element blocks or
- an element boundary line, which is shared by elements in the same element block.

This classification includes only elements which are alive (see Section [Ref: mesh:actelem]).

Each element block is assigned a color from the standard (*not* spectrum) color scale which remains constant throughout the run. An element's color is the color associated with its element block. A node's color is the color of all the element's containing the node, or white if the elements are in different element blocks.

Mesh boundary lines and element block boundary lines are drawn in white. Element boundary lines are normally drawn in the element's color; if the mesh is painted (e.g., in solid mesh mode or paint contour mode), they are drawn in black.

On devices which allow different line widths, the mesh boundary lines are drawn as wide lines, and the element block boundary lines are slightly thinner than the mesh boundary lines but wider than the element boundary lines. If the element boundary lines are drawn in a mode that references a variable, the element boundary lines are very thin.

If dead nodes are selected (with the DEATH and DEADNODE commands), each dead node is shown as a dot in the node's color.

If node numbers are requested, each number appears in white to the upper right of the node. If element numbers are requested, each number appears in the element's color in the center of the element.

If node sets are displayed, each node in a selected set is marked with an X in the color assigned to the node set. The identifiers of selected sets appear in the plot legend in the color assigned to the set.

If side sets are displayed, the nodes that make up a side in a selected set are connected by lines in the color assigned to the side set. The first two nodes displayed for each side are

connected with an arrow to indicate the node ordering within the set. The identifiers of selected sets appear in the plot legend in the color assigned to the set.

### **5.1.1 DETOUR Mesh Output**

The display mode and the **MLINES** and **BOUNDARY** commands determine which lines are drawn on the plot. Mesh boundary lines and element block boundary lines are drawn by default in all display modes. Element boundary lines are drawn by default in the wireframe and solid mesh modes. These lines are drawn only for element blocks that are selected. If the mesh is painted (e.g., in solid mesh mode or paint contour mode), these lines are drawn in black.

In wireframe mesh mode, only the mesh is drawn.

In solid mesh mode, each element is painted in the element's color.

In contour mode, the spectrum color scale is used if selected. In line contour mode, the contour lines are in color and are labeled with a letter (controlled by the **CLABEL** command). The value corresponding to each color and letter is listed in the plot legend. In paint contour mode, the contour areas are "painted" in color. The value range corresponding to each color is shown in the plot legend. In either contour mode, the location of the minimum and maximum nodal values for a time step may be shown with symbols, and the values listed in the plot legend (controlled by the **CSYMBOLS** command). The minimum symbol is a circle with a cross in it; the maximum symbol is an 8-point star.

In vector mode, a vector indicating the magnitude and direction of the two plot variable values is drawn for each node or active element. If nodal variables are given, the vector is drawn from the node in the node's color. If element variables are given, the vector is drawn from the center of the element in the element's color. The vector size is controlled with the **VSCALE** command.

In element symbol mode, a symbol is drawn at the center of each active element in the element's color. The symbol drawn is described under the **SYMBOL** command. The symbol size is controlled with the **VSCALE** command.

### **5.1.2 PATHLINE Mesh Output**

#### **5.1.3 Curve Plot Mesh Output**

An undeformed wireframe mesh plot may be output from an X-Y curve subprogram.

If the **ECHO** command from **SPLOT** is used to draw the mesh, selected nodes and elements are connected with yellow arrows.

## **5.2 X-Y Curve Output**

Each plot is displayed on the graphics device. The user may be prompted for a response after each plot (see the **PLOT** command).

Each curve is plotted and identified in the legend. Each curve has an associated color, line type, and symbol (controlled by the **COLOR**, **LINES**, and **SYMBOLS** commands). The spectrum color scale is used, if selected. On a multiple curve plot, each of these characteristics is incremented independently (cycling if necessary). Some characteristics may be constant for all curves (for example, if no symbols or solid lines only are selected). The characteristics are reset at the start of each plot, and thus are the same for the first curve on each plot in a plot set. If lines are selected, consecutive points on the curve are connected with a straight line of the appropriate type. If symbols are selected, they appear at each point on the curve. The curves on a plot are numbered if they cannot be distinguished by the line type and symbol. Thus if the line type and symbol are constant or if more than six curves are on the plot, all curves are numbered. The placement of the curve number is determined by the **CRVNUM** command.

The line type is included in the legend on multiple curve plots if it varies. The symbol is included if it varies. If the curves are numbered, the number is included in the legend.

### **5.2.1 Tplot Curve Output**

The legend lists the time and the variable (for a time history curve) or both variables (for a variable-versus-variable curve) for each curve in the curve color.

### **5.2.2 Splot Curve Output**

The legend lists the variable and time for each curve in the curve color.

If the curves are not overlaid, all the curves for each selected variable are plotted before proceeding to the next selected time.



## 6 Informational and Error Messages

Error messages may appear during the program initialization, during command input or during the plotting. If a fatal error occurs, the program aborts.

An error message of the following form may appear in response to a command:

```
*** ERROR - message
*** WARNING - message.
```

If an error occurs, the command is usually ignored. If only a warning is printed, the command is usually performed. If the message is not sufficiently informative, the appropriate command description may be helpful. The display after the command shows the effect of the command.

*BLOT* expects a valid database; a bad database can cause very unusual behavior. If a database format error is discovered before the time steps, the program prints an error of the following format:

DATABASE ERROR - Reading *database* item

and aborts. If a database format error is found while scanning the time steps, the following error message is printed:

WARNING - End-of-file during time steps.

The time step with the error and all following time steps are ignored, but the program continues and the valid time steps are available for processing.

A random file is used to store the database variables for efficiency reasons. If the random file cannot be opened, a message of the following format is printed:

WARNING - Random file cannot be opened, *message*

and processing continues. Program performance may be impaired if the random file cannot be opened, but the program should produce correct results.

The program allocates memory dynamically as it is needed. If the system runs out of memory, the following message is printed:

FATAL ERROR - Too much dynamic memory requested

and the program aborts. The user should first try to obtain more memory on the system. Another solution is to run the program in a less memory-intensive fashion. For example, cutting down the mesh size or requesting fewer curves in a plot set may require less memory.

*BLOT* has certain programmer-defined limitations (for example, the number of curves that may be defined). The limits are not specified in this manual since they may change. In most cases the limits are chosen to be more than adequate. If the user exceeds a limit, a message is printed. If the user feels the limit is too restrictive, the code sponsor should be notified so the limit may be raised in future releases of *BLOT*.

## 7 Executing *BLOT*

The details of executing *BLOT* are dependent on the system being used. The system manager of any system that runs *BLOT* should provide a supplement to this manual that explains how to run the program on that particular system. Site supplements for all currently supported systems are in Appendix [Ref: appx:site] .

### 7.1 Execution Files

The table below summarizes *BLOT*'s file usage.

**Figure 1:**

Description	Unit Number	Type	File Format
User input	standard input	input	Section
User output	standard output	output	ASCII
EXODUS database	11	input	exodus]
Neutral file	20	optional output	grafaid]
Print file	21	optional output	ASCII
Log file	99	optional output	ASCII
Scratch random file	90	optional output	binary

All files must be connected to the appropriate unit before *BLOT* run. Each file (except standard input and output) is opened with the name retrieved by the EXNAME routine of the SUPES library [bib:supes].

The plots are generated on the connected graphics device(s). The graphics terminal device name (if any) should be assigned to symbol 1 (see EXNAME) and the hardcopy device name (if any) should be assigned to symbol 2. *BLOT* retrieves the device names with EXNAME. The appropriate graphic device drivers are linked to the program in a system-dependent fashion.

### 7.2 Special Software

*BLOT* is written in ANSI FORTRAN-77 [bib:f77] with the exception of the following system-dependent features:

- the VAX VMS help facility,
- the OPEN options for the files, and
- the use of ASCII characters that are not in the FORTRAN standard character set.

*BLOT* uses the following software packages:

- the Sandia Virtual Device Interface ( SVDI) [bib:svdi] with extensions for dual virtual devices ( DVDI) [bib:dvdi] (if supported),
- the PLT graphics package [bib:plt] which uses the SVDI, and
- the SUPES package [bib:supes] which includes dynamic memory allocation, a free-field reader, and FORTRAN extensions.

## References

- [bib:detour]Amy P. Gilkey and Dennis P. Flanagan, “DETOUR – A Deformed Mesh / Contour Plot Program,” SAND86-0914, \SNLA , July 1987.
- [bib:tplot]Amy P. Gilkey, “TPLOT – A Time History or X-Y Plot Program for the Output of a Finite Element Analysis,” SAND86-0883, \SNLA , August 1986.
- [bib:spplot]Amy P. Gilkey, “SPLOT – A Distances-versus-variable Plot Program for the Output of a Finite Element Analysis,” SAND86-0882, \SNLA , August 1986.
- [bib:grope]Amy P. Gilkey, “GROPE – A GENESIS / EXODUS Database Examination Program,” Internal Memorandum RS1523/88/02, \SNL , in preparation.
- [bib:oldtplot]Zelma E. Beisinger and Charles M. Stone, “TPLOT2: A Flexible X-Y Plotting Program For Use With Finite Element Software,” SAND80-2508, \SNLA , February 1981.
- [bib:oldsplot]Mary R. Sagartz, “SPLOT: A Plot Program for Finite Element Data, Function vs. Distance,” SAND84-1406, \SNLA , January 1985.
- [bib:exodus]William C. Mills-Curran, Amy P. Gilkey, Dennis P. Flanagan, “EXODUS: A Finite Element File Format for Pre- and Post-Processing,” SAND87-2997, \SNLA , in preparation.
- [bib:genesis]Lee M. Taylor, Dennis P. Flanagan, and William C. Curran, “The GENESIS Finite Element Mesh File Format,” SAND86-0910, \SNLA , May 1986.
- [bib:grafaid]C. R. Adams, “GRAFAID Code User Manual Version 2.0,” SAND84-1725, \SNLA , September 1985.
- [bib:svdi]Applied Computer Graphics Division, “Functional Specification of the Sandia Virtual Device Interface (SVDI),” SAND81-1900, \SNLA , December 1981.
- [bib:dvdi]Dennis P. Flanagan, “Dual Device, Shared Library, and Switchable Device Systems for the SVDI,” SAND88-1341, \SNLA , in preparation.
- [bib:f77]*American National Standard Programming Language FORTRAN*, American National Standards Institute, ANSI X3.9-1978, New York, 1978.
- [bib:plt]Colin B. Selleck and Mark P. Sears, “PLT – A Multipurpose Plotting Package,” SAND83-0627, \SNLA , August 1983.
- [bib:supes]John R. Red-Horse, William C. Curran, and Dennis P. Flanagan, “SUPES Version 2.1 – A Software Utility Package for the Engineering Sciences ,” SAND90-0247, \SNLA , May 1990.



## A The EXODUS Database Format

The following code segment reads an EXODUS database. The first segment is the GENESIS database format.

```
C  --Open the EXODUS database file
      NDB = 9
      OPEN (UNIT=NDB, ..., STATUS='OLD', FORM='UNFORMATTED')
C  --Read the title
      READ (NDB) TITLE
C  --TITLE - the title of the database (CHARACTER*80)
C  --Read the database sizing parameters
      READ (NDB) NUMNP, NDIM, NUMEL, NELBLK,
      &  NUMNPS, LNPSNL, NUMESS, LESSEL, LESSNL, NVERSN
C  --NUMNP - the number of nodes
C  --NDIM - the number of coordinates per node
C  --NUMEL - the number of elements
C  --NELBLK - the number of element blocks
C  --NUMNPS - the number of node sets
C  --LNPSNL - the length of the node sets node list
C  --NUMESS - the number of side sets
C  --LESSEL - the length of the side sets element list
C  --LESSNL - the length of the side sets node list
C  --NVERSN - the file format version number
C  --Read the nodal coordinates
      READ (NDB) ((CORD(INP,I), INP=1,NUMNP), I=1,NDIM)
C  --Read the element order map (each element must be listed once)
      READ (NDB) (MAPEL(IEL), IEL=1,NUMEL)
C  --Read the element blocks
      DO 100 IEB = 1, NELBLK
C  --Read the sizing parameters for this element block
      READ (NDB) IDELB, NUMELB, NUMLNK, NATRIB
C  --IDELB - the element block identification (must be unique)
C  --NUMELB - the number of elements in this block
C  -- (the sum of NUMELB for all blocks must equal NUMEL)
C  --NUMLNK - the number of nodes defining the connectivity
C  -- for an element in this block
C  --NATRIB - the number of element attributes for an element
C  -- in this block
C  --Read the connectivity for all elements in this block
      READ (NDB) ((LINK(J,I), J=1,NUMLNK, I=1,NUMELB)
C  --Read the attributes for all elements in this block
      READ (NDB) ((ATRIB(J,I), J=1,NATRIB, I=1,NUMELB)
100 CONTINUE
C  --Read the node sets
      READ (NDB) (IDNPS(I), I=1,NUMNPS)
C  --IDNPS - the ID of each node set
      READ (NDB) (NNNPS(I), I=1,NUMNPS)
C  --NNNPS - the number of nodes in each node set
      READ (NDB) (IXNNPS(I), I=1,NUMNPS)
C  --IXNNPS - the index of the first node in each node set
C  -- (in LTNNPS and FACNPS)
      READ (NDB) (LTNNPS(I), I=1,LNPSNL)
```

```

C      --LTNNPS - the nodes in all the node sets
      READ (NDB) (FACNPS(I), I=1, LNPSNL)
C      --FACNPS - the factor for each node in LTNNPS
C  --Read the side sets
      READ (NDB) (IDESS(I), I=1, NUMESS)
C      --IDESS - the ID of each side set
      READ (NDB) (NEESS(I), I=1, NUMESS)
C      --NEESS - the number of elements in each side set
      READ (NDB) (NNESS(I), I=1, NUMESS)
C      --NNESS - the number of nodes in each side set
      READ (NDB) (IXEESS(I), I=1, NUMESS)
C      --IXEESS - the index of the first element in each side set
C      --      (in LTEESS)
      READ (NDB) (IXNESS(I), I=1, NUMESS)
C      --IXNESS - the index of the first node in each side set
C      --      (in LTNESS and FACESS)
      READ (NDB) (LTEESS(I), I=1, LESSEL)
C      --LTEESS - the elements in all the side sets
      READ (NDB) (LTNESS(I), I=1, LESSNL)
C      --LTNESS - the nodes in all the side sets
      READ (NDB) (FACESS(I), I=1, LESSNL)
C      --FACESS - the factor for each node in LTNESS

```

A valid GENESIS database may end at this point or at any point until the number of variables is read.

```

C  --Read the QA header information
      READ (NDB, END=900) NQAREC
C      --NQAREC - the number of QA records (must be at least 1)
      DO 110 IQA = 1, MAX(1, NQAREC)
          READ (NDB) (QATITL(I, IQA), I=1, 4)
C          --QATITL - the QA title records; each record contains:
C          --  1) analysis code name (CHARACTER*8)
C          --  2) analysis code qa descriptor (CHARACTER*8)
C          --  3) analysis date (CHARACTER*8)
C          --  4) analysis time (CHARACTER*8)
110 CONTINUE
C  --Read the optional header text
      READ (NDB, END=900) NINFO
C      --NINFO - the number of information records
      DO 120 I = 1, NINFO
          READ (NDB) INFO(I)
C          --INFO - extra information records (optional) that contain
C          -- any supportive documentation that the analysis code
C          -- developer wishes (CHARACTER*80)
120 CONTINUE
C  --Read the coordinate names
      READ (NDB, END=900) (NAMECO(I), I=1, NDIM)
C      --NAMECO - the coordinate names (CHARACTER*8)
C  --Read the element type names
      READ (NDB, END=900) (NAMELB(I), I=1, NELBLK)
C      --NAMELB - the element type names (CHARACTER*8)

```

The GENESIS section of the database ends at this point.

```

C  --Read the history, global, nodal, and element variable information
    READ (NDB, END=900) NVARHI, NVARGL, NVARNP, NVAREL
C  --NVARHI - the number of history variables
C  --NVARGL - the number of global variables
C  --NVARNP - the number of nodal variables
C  --NVAREL - the number of element variables
    READ (NDB)
    & (NAMEHV(I), I=1,NVARHI),
    & (NAMEGV(I), I=1,NVARGL),
    & (NAMENV(I), I=1,NVARNP),
    & (NAMEEV(I), I=1,NVAREL)
C  --NAMEHI - the history variable names (CHARACTER*8)
C  --NAMEGV - the global variable names (CHARACTER*8)
C  --NAMENV - the nodal variable names (CHARACTER*8)
C  --NAMEEV - the element variable names (CHARACTER*8)
    READ (NDB) ((ISEVOK(I,J), I=1,NVAREL), J=1,NELBLK)
C  --ISEVOK - the name truth table for the element blocks;
C  -- ISEVOK(i,j) refers to variable i of element block j;
C  -- the value is 0 if and only if data will NOT be output for
C  -- variable i for element block j (otherwise the value is 1)
C  --Read the time steps
130 CONTINUE
    READ (NDB, END=900) TIME, HISTFL
C  --TIME - the time step value
C  --HISTFL - the time step type flag:
C  -- 0.0 for all variables output ("whole" time step) else
C  -- only history variables output ("history-only" time step)
C
    READ (NDB) (VALHV(IVAR), IVAR=1,NVARGL)
C  --VALHV - the history values for the current time step
    IF (HISTFL .EQ. 0.0) THEN
        READ (NDB) (VALGV(IVAR), IVAR=1,NVARGL)
C  --VALGV - the global values for the current time step
        DO 140 IVAR = 1, NVARNP
            READ (NDB) (VALNV(INP,IVAR), INP=1,NUMNP)
C  --VALNV - the nodal variables at each node
C  -- for the current time step
140 CONTINUE
            DO 160 IBLK = 1, NELBLK
                DO 150 IVAR = 1, NVAREL
                    IF (ISEVOK(IVAR,IBLK) .NE. 0) THEN
                        READ (NDB) (VALEV(IEL,IVAR,IBLK),
C  --VALEV - the element variables at each element
C  -- for the current time step
                            & IEL=1,NUMELB(IBLK))
                    END IF
250 CONTINUE
260 CONTINUE
                END IF
            END IF
        END IF
C  --Handle time step data
        ...
        GOTO 130
    END IF
900 CONTINUE
C  --Handle end of file on database ...

```



## B The GRAFAID Neutral File Format

The GRAFAID neutral file format is described in Section 9-2 of the GRAFAID Code User Manual [bib:grafaid]. The following description covers only those parts that are relevant to the *BLOT* program.

The neutral file is a sequential file written in ASCII format. Each record is free format with fields separated by a comma.

The general structure of the GRAFAID neutral file is:

```
Neutral file title record (optional) \nth 1st Curve Definition Record
Set \nth 2nd Curve Definition Record Set ... \nthNthCurveDefinition
Record Set END NEUTRAL FILE
```

The neutral file title contains the name of the program which wrote the neutral file ( *BLOT* ) and the name, date, and time of the programs that created and last modified the database.

The format of the records in a curve definition record set is:

```
BEGIN CURVE,curve-name ntitle,title1 title2 ... titlentitle x-label y-label x-min,x-max,y-min,y-
max,npts,aux-data
```

```
axis-type,x-type,aux-type x1,y1 x2,y2 ... xnpts,ynpts END CURVE,curve-name
```

```
\newlength \graftag \newlength \grafdesc \settowidth \graftag curve-name 15 characters
```

```
\addtolength \grafdesc -\graftag \addtolength \grafdesc -\tabbingsep
```

Field	Type	Description
<i>curve-name</i>	15 characters	The curve name, set by the ACURVE and NCURVE commands.
<i>ntitle</i>	integer	The number of title lines (5 maximum).
<i>title<sub>i</sub></i>	80 characters	The \nth ith curve title line. If the user has requested QA information with the QA command, the database title is the first title line. The caption lines set by the CAPTION command are the next title lines. The last title line is a description of the curve.
<i>x-label</i>	40 characters	The X axis label, set by the XLABEL command.
<i>y-label</i>	40 characters	The Y axis label, set by the YLABEL command.
<i>x-min</i>	real	The X axis lower plot limit, set by the XSCALE command.
<i>x-max</i>	real	The X axis upper plot limit, set by the XSCALE command.
<i>y-min</i>	real	The Y axis lower plot limit, set by the YSCALE command.
<i>y-max</i>	real	The Y axis upper plot limit, set by the YSCALE command.
<i>npts</i>	integer	The number of curve data X-Y pairs.
<i>aux-data</i>	1 character	F to indicate that no auxiliary data exists.

<i>axis-type</i>	4 characters	NOLO to indicate a linear scale on both axes.
<i>x-type</i> data is monotonic.	4 characters	NONM if the X data is nonmonotonic or MONO if the X data is monotonic.
<i>aux-type</i>	1 character	Blank field to indicate that no auxiliary data exists.
$x_i$	real	The $i$ th X data value.
$y_i$	real	The $i$ th Y data value.

## C Special Plot Text Capabilities

There are several special capabilities available for user-defined text that appears on a plot (such as the plot caption). For example, the text can include subscripted text, Greek letters, and special symbols. The user requests a capability by inserting an “escape sequence” in the text. The escape sequence starts with a backslash (“\”) and (unless otherwise noted) must end with a space which is deleted from the text string. Escape sequences are translated only when **software** characters are plotted. If an invalid escape sequence is entered, an error message appears when the plot is drawn and the sequence is ignored. The following escape sequences are available:

Figure 2:

<code>\XXXXXXX</code>		<code>\CSQ</code>	centered square
<code>\SQ</code>	square	<code>\CSQ</code>	centered square
<code>\DI</code>	diamond	<code>\CDI</code>	centered diamond
<code>\CS</code>	cross	<code>\CCS</code>	centered cross
<code>\X</code>	X	<code>\CX</code>	centered X
<code>\TR</code>	triangle	<code>\CTR</code>	centered triangle
<code>\CI</code>	circle	<code>\CCI</code>	centered circle
<code>\DO</code>	dot	<code>\CDO</code>	centered dot
<code>\LO</code>	logo	<code>\CLO</code>	centered logo
<code>\SLINE</code>	solid line	<code>\DLINE</code>	dotted line
<code>\DDLIN</code>	dot-dash line	<code>\SDLIN</code>	short dash line
<code>\LDLIN</code>	long dash line	<code>\MDLIN</code>	medium dash line
<code>\NEQ</code>	not equal to		
<code>\LEQ</code>	less than or equal to	<code>\GEQ</code>	greater than or equal to
<code>\NLEQ</code>	not less than or equal to	<code>\NGEQ</code>	not greater than or equal to
<code>\NLT</code>	not less than	<code>\NGT</code>	not greater than
<code>\LL</code>	much less than	<code>\GG</code>	much greater than
<code>\PLUSMIN</code>	plus or minus	<code>\PRIME</code>	prime
<code>\SUM</code>	summation	<code>\APPROX</code>	approximation sign
<code>\\</code>	single backslash (with no ending space)		
<code>\GR</code>	Greek font	<code>\ENG</code>	normal font
<code>\^</code>	start superscript (with no ending space)		

**Figure 2:**

<code>\XXXXXXX</code>		<code>\CSQ</code>	centered square
<code>\_</code>	start subscript (with no ending space)		
<code>\-</code>	end superscript or subscript (with no ending space)		

## D Command Summary

### GENERAL COMMANDS

Subprogram Control (page [Pageref: gen:control])

#### LOG

requests that the log file be saved.

#### RESET

resets all plot parameters to their initial values.

#### EXIT

exits immediately from the program.

#### DETOUR

switches to the DETOUR subprogram.

#### PATHLINE

switches to the PATHLINE subprogram.

#### TPLOT

switches to the TPLOT subprogram.

#### SPLOT

switches to the SPLOT subprogram.

Information (page [Pageref: gen:info])

#### SELECT *option*

selects database information for the LIST and PRINT commands.

#### LIST *option*

displays database information to the user's terminal.

#### PRINT *option*

prints database information to a file.

**SHOW** *command*

displays plot parameters relevant to the command.

**HELP** *command*

displays information about a program command.

#### Time Step Selection (page [Pageref: gen:timesel])

**TMIN** *tmin*

sets the minimum selected time to *tmin*.

**TMAX** *tmax*

sets the maximum selected time to *tmax*.

**NINTV** *nintv*

sets the number of selected time intervals to *nintv* (delta offset).

**ZINTV** *nintv*

sets the number of selected time intervals to *nintv* (zero offset).

**DELTIME** *delt*

sets the selected time interval to *delt*.

**ALLTIMES**

selects all time steps between *tmin* and *tmax*.

**TIMES** [ADD,]  $t_1, t_2, \dots$

selects times  $t_1, t_2$ , etc.

**STEPS** [ADD,]  $n_1, n_2, \dots$

selects time steps  $n_1, n_2$ , etc.

**HISTORY ON/OFF**

controls whether history time steps are included in the selected time steps or only whole time steps.

Graphics Options (page [Pageref: gen:graphopt])

QA ON/OFF, MESH/XY

controls whether the QA information is displayed on the plot legend.

AXIS ON/OFF, MESH/XY

controls whether the plot axes are numbered.

LEGEND ON/OFF, MESH/XY

controls whether the legend (excluding the QA information) is displayed on the plot legend.

CAPTION *line\_number*, MESH/XY

sets the plot caption.

SOFTCHAR ON/OFF, *device*

selects software characters or hardware characters.

FONT STICK/SANSERIF/ROMAN, *device*

selects the font to use for software characters.

COLOR *ncol*

sets the maximum number of default color scale colors to use.

SPECTRUM *ncol*

sets the maximum number of spectrum colors to use.

SNAP *nsnap*

controls the number of frames snapped on a camera device.

AUTO ON/OFF

controls whether a response is requested after each plot.

Plot Set Display (page [Pageref: gen:display])

## PLOT

generates the plots requested on the primary graphics device.

## HARDCOPY

generates the plots requested on the hardcopy device.

## MESH COMMANDS

Mesh View Control (page [Pageref: xmesh:meshview])

## EMPTY

sets empty view.

## DEFORM ON/OFF

controls whether the mesh is deformed or undeformed.

## NUMBER NODES/ELEMENTS/ALL/OFF

controls whether the nodes and/or elements are numbered on the mesh.

## MLINES ON/DOTTED/OFF

controls whether all mesh lines are displayed or only the mesh boundary and element block boundaries.

## BOUNDARY ON/BLACK/OFF

controls whether the mesh boundary and element block boundaries are displayed on painted mesh plots.

## NSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... NSETS OFF

controls whether node sets are displayed on the mesh.

## SSETS [ADD,] *set\_id*<sub>1</sub>, *set\_id*<sub>2</sub>, ... SSETS OFF

controls whether side sets are displayed on the mesh.

Active Element Control (page [Pageref: xmesh:actelem])

## BLOCKS [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... BLOCKS OFF

selects the element blocks for active elements.

**MATERIAL** [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ... **MATERIAL OFF**

is equivalent to a **BLOCKS** command.

**VISIBLE** [ADD or DELETE,] *block\_id*<sub>1</sub>, *block\_id*<sub>2</sub>, ...

selects the element blocks that are visible in 3D.

**DEATH ON**, *element\_variable* **DEATH OFF**

controls whether “dead” elements are ignored.

**DEADNODE ON/OFF**

controls the display of “dead” nodes.

Multiple Views Control (page [Pageref: xmesh:multview])

**VIEW** *nview*, *view\_dependent\_command*

sets the view-dependent parameters of view *nview*.

**XVIEW ON/OFF**

defines or deletes non-symmetric vertically divided views.

**YVIEW ON/OFF**

defines or deletes non-symmetric horizontally divided views.

**XSYM LEFT/RIGHT**, *xsym* **XSYM OFF**

defines or deletes the symmetric vertically divided views.

**YSYM BOTTOM/TOP**, *ysym* **YSYM OFF**

defines or deletes the symmetric horizontally divided views.

**MULTTIME ON/OFF**

assigns different time steps to each view or the same step to all views.

Mesh Control (page [Pageref: xmesh:control])

**MAGNIFY** *dfac*

causes the displacements to be magnified by *dfac* before they are added to the coordinates.

**HIDDEN** *level*

controls the 3D hidden line and surface algorithm.

**ZOOM** *hmin, hmax, vmin, vmax* or **CURSOR ZOOM TRANSLAT** *hcen, vcen* or  
**CURSOR ZOOM EACH ZOOM MESH**

sets the mesh window limits.

**SQUARE ON/OFF**

determines if the mesh window is square or defined exactly by the **ZOOM** limits.

**TICK** *tick*

sets the axis tick interval for both mesh axes.

3D Rotation Commands (page [Pageref: xmesh:rotate])

**ROTATE** *axis<sub>1</sub>, ndeg<sub>1</sub>, axis<sub>2</sub>, ndeg<sub>2</sub>, ...* **ROTATE RESET**

sets the 3D mesh rotation.

**EYE** *xpos, ypos, zpos*

sets the 3D mesh rotation by position.

**CENTER** *xcen, ycen, zcen*

sets the 3D center of rotation.

## DETOUR COMMANDS

Mesh Display Mode Control (page [Pageref: det:dismode])

**WIREFRAM**

sets wireframe mesh mode.

**SOLID**

sets solid mesh mode which paints the mesh.

**CONTOUR** *variable*

sets line contour mode which plots contours of the variable on the mesh.

**PAINT** *variable*

sets paint contour mode which paints contours of the variable on the mesh.

**VECTOR** *x\_variable, y\_variable*

sets vector mode which plots a vector representing the two variables on the mesh for each node or active element.

**SYMBOL CRACK/ANGLE**, *element\_variable*

sets element symbol mode which plots a symbol on the mesh for each active element.

*variable*

is equivalent to a **CONTOUR** command if in wireframe or solid mesh mode; otherwise equivalent to the command which sets the current display mode.

Contour Control (page [Pageref: det:contour])

**NCNTRS** *ncntr*

sets the number of contours to *ncntr*.

**CRANGE** *cmin, cmax*

sets the minimum contour value to *cmin* and the maximum contour value to *cmax*.

**DELCNTR** *delc, cmin*

sets the contour interval to *delc* and the minimum contour value to *cmin*.

**CSHIFT** *cval*

sets the contour limits *cmin* and *cmax* so that *cval* falls on the nearest contour line or paint boundary.

Mode Specific Options (page [Pageref: det:disopt])

**CLABEL ON/label\_incr/OFF**

sets the contour labeling increment.

## COPEN ON/OFF, ON/OFF

controls the plotting of values outside contour limits on a painted contour plot.

## CSYMBOLS *nmin, nmax*

controls the plotting of symbols at the minimum and maximum nodal variable values on a contour plot.

## VSCALE *vector\_scale*

sets the vector/symbol scale factor.

## PATHLINE COMMANDS

### ADD

saves the pathlines defined in the last plot set.

### REMOVE *pathline<sub>1</sub>, pathline<sub>2</sub>, ...* <No Default>

deletes the specified pathlines or deletes the last *-pathline<sub>i</sub>* pathlines defined.

## X-Y CURVE COMMANDS

Curve Definition (page [Pageref: xy:def])

### ADD

saves the curves defined in the last plot set.

### REMOVE *curve<sub>1</sub>, curve<sub>2</sub>, ...* <No Default>

deletes the specified curves or deletes the last *-curve<sub>i</sub>* curves defined.

Axis Labeling and Scaling (page [Pageref: xy:axis])

### XLABEL

sets the X axis label.

### YLABEL

sets the Y axis label.

### XSCALE *xmin, xmax*

sets the X axis limits.

**YSCALE** *ymin, ymax*

sets the Y axis limits.

**XTICK** *xtick*

sets the X axis tick interval.

**YTICK** *ytick*

sets the Y axis tick interval.

**SAMESCAL ON/OFF**

controls whether all curves have the same scale.

**NORMAL ON/OFF**

controls whether each curve is to be scaled individually or all curves on a plot are to have the same scale.

**RATIOXY** *ratio*

sets the X-to-Y axis length ratio.

Curve Display Options (page [Pageref: xy:display])

**OVERLAY VARIABLE/TIME/OFF**

controls whether all curves are displayed on the same plot.

**GRID ON/OFF**

controls whether a grid is drawn on the plot.

**LINES ON/VARY/OFF**

controls the type of line to plot.

**SYMBOLS VARY/*symbol\_number*/OFF**

controls whether symbols are drawn at each curve point.

**CRVNUM FIRST/MIDDLE/LAST/OFF**

controls where the curves are numbered.

Neutral File Options (page [Pageref: xy:neutral])

ACURVE *acurve*

sets the name part of the neutral file curve name.

NCURVE *ncurve, incrv*

sets the starting curve number and increment for the neutral file.

NEUTRAL

writes the defined curves to the neutral file.

Mesh Display (page [Pageref: xy:mesh])

MESH

displays the mesh.

## TPLOT COMMANDS

Curve Definition (page [Pageref: tplot:def])

TYPLOT

defines a time history curve.

*variable node\_range<sub>1</sub> or element\_range<sub>1</sub>, node\_range<sub>2</sub> or element\_range<sub>2</sub>, ...*

is equivalent to a TYPLOT command.

XYPLOT

defines a variable-versus-variable curve.

## SPLOT COMMANDS

Curve Definition (page [Pageref: splot:def])

NODES [ADD,] [PATH,] *node\_range<sub>1</sub> node\_range<sub>2</sub> ...*

selects the node numbers for the nodal variables.

ELEMENTS [ADD,] [PATH,] *element\_range<sub>1</sub> element\_range<sub>2</sub> ...*

selects the element numbers for the element variables.

## SYPLOT

defines a distance-versus-variable curve.

*variable*

is equivalent to a SYPLOT command.

Mesh Display (page [Pageref: splot:mesh])

## ECHO

displays the mesh with the nodes/elements selected with the NODES or ELEMENTS command.



## E Sample Plots

The plots on the following pages were generated with the sequence of *BLOT* commands that precede the plots. Note that the examples are from different databases.

### DETOUR Deformed Mesh, Contour, and Vector Plots

```
DETOUR      $ Switch to DETOUR subprogram
TIME 2.104E-6 $ Select database time
ZOOM 0, 5E-3, -1E-3, 4E-3 $ Set window limits
WIREFRAM    $ Request a wireframe mesh plot (deformed)
CAPTION     $ Define three-line plot caption
*** DETOUR EXAMPLE 1 ***
Deformed wireframe mesh plot
PLOT        $ Display the deformed wireframe mesh
CONTOUR VONMISES $ Request a contour plot of variable VONMISES
CAPTION     $ Define three-line plot caption
*** DETOUR EXAMPLE 2 ***
Contour plot of von Mises stress
PLOT        $ Display the contour plot
VECTOR VELX, VELY $ Request a vector plot of VELX and VELY
CAPTION     $ Define three-line plot caption
*** DETOUR EXAMPLE 3 ***
Vector plot of velocities
PLOT        $ Display the vector plot
```

\notetome Example produced by DETOUREX.COM

### MESH Symmetric versus Non-symmetric Views

```
DETOUR      $ Switch to DETOUR subprogram
TIME 12E-3  $ Select database time
WIREFRAM    $ Request a wireframe mesh plot
YVIEW       $ Create non-symmetric views (bottom and top)
VIEW 2 DEFORM ON $ Set top view to deformed
VIEW 4 DEFORM OFF $ Set bottom view to undeformed
XSYM        $ Create symmetric views (side by side)
CAPTION     $ Define three-line plot caption
*** DETOUR EXAMPLE 4 ***
(Symmetric versus non-symmetric views)
```

Deformed versus undeformed mesh plot

PLOT           \$ Plot the views

\notetome Example produced by DETOUREX.COM

#### MESH Element Birth/Death

DETOUR       \$ Switch to DETOUR subprogram

TIME .115E-3 \$ Select database time

ZOOM 0,1.75,0,1.75 \$ Set window limits

WIREFRAME \$ Request a wireframe mesh plot

XSYM LEFT 0.0 \$ Create two symmetric views

YSYM BOTTOM 0.0 \$ Create four symmetric views

CAPTION     \$ Define three-line plot caption

\*\*\* DETOUR EXAMPLE 6 \*\*\*

(Four symmetric views)

Deformed mesh with all elements

PLOT           \$ Plot the views

DEATH ON STATUS \$ Request element birth/death

CAPTION     \$ Define three-line plot caption

\*\*\* DETOUR EXAMPLE 7 \*\*\*

(Four symmetric views)

Deformed mesh with element birth/death

PLOT           \$ Plot the views

\notetome Example produced by DETOUREX.COM

#### MESH Node and Element Numbering

TPLOT       \$ Switch to TPLOT subprogram

NUMBER ALL \$ Number both nodes and elements

ZOOM .000,.010,.000,.016 \$ Zoom so numbers are readable

CAPTION     \$ Define three-line plot caption

\*\*\* MESH EXAMPLE 1 \*\*\*

Numbered nodes and elements on mesh plot

MESH         \$ Display single time step of mesh

\notetome add node and side sets \notetome add pathline examples \notetome add 3D plot  
with rotation and contours

#### TPLOT Time History Curve

```

TPLOT          $ Switch to TPLOT subprogram
SIGXX 50 TO 250 BY 50 $ Define 5 time history curves
                $ at elements 50,100,150,200,250
OVERLAY       $ Overlay all curves on one plot
LINE VARY     $ Draw each curve with a different line type
CAPTION      $ Define three-line plot caption
*** TPLOT EXAMPLE 1 ***
Overlaid time history curves
PLOT         $ Plot the curves defined above

```

#### TPLOT Variable-versus-variable Curve

```

TPLOT          $ Switch to TPLOT subprogram
XYPLOT       $ Define a variable-versus-variable curve
STRAIN, 2    $ X variable: STRAIN at element 2
VONMISES, 2  $ Y variable: VONMISES at element 2
GRID         $ Draw grid
SYMBOLS     $ Draw symbols at each point on curve
XSCALE - .03, .45 $ Set X axis limits
YSCALE -50, 800 $ Set Y axis limits
YLABEL      $ Define Y axis label
VONMISES STRESS
CAPTION     $ Define three-line plot caption
*** TPLOT EXAMPLE 2 ***
Variable-versus-variable curve
PLOT         $ Plot the curve defined above

```

#### SPLOT Overlaid Curves

```

SPLOT          $ Switch to SPLOT subprogram
NODES 1 TO 306 BY 6 $ Select nodes 1,7,13,....,301
ACCLX         $ Define curve of nodal variable ACCLX
ACCLY         $ Define curve of nodal variable ACCLY
TIMES 8.0E-6 $ Select time 8E-6 only
OVERLAY VARIABLE $ Overlay curves for both variables on one plot
SYMBOLS     $ Draw symbols at each point on curve
YSCALE -37E9, 5E9 $ Set Y axis limits
CAPTION     $ Define three-line plot caption
*** SPLOT EXAMPLE 1 ***

```

```
Two overlaid nodal variable curves
PLOT          $ Plot the curves defined above
RESET        $ Clear all parameters (for example, symbols)
ELEMENTS 3 TO 250 BY 5 $ Select elements 3,8,13,...,248
SYMPLOT SIGXX $ Define curve of element variable SIGXX
NINTV 6      $ Select 6 times over the database
OVERLAY TIME $ Overlay curves for all times on one plot
LINE VARY   $ Draw each curve with a different line type
CAPTION    $ Define three-line plot caption
*** SPLOT EXAMPLE 2 ***
Overlaid time curves of element variable SIGXX
PLOT          $ Plot the curves defined above
```