

OPERA-2D REFERENCE MANUAL

Vector Fields Limited
24 Bankside
Kidlington
Oxford OX5 1JE
England

Copyright © 1999-2002 by Vector Fields Limited, England

This document was prepared using Adobe[®] FrameMaker[®].

UNIX is a trademark of X/Open Company Ltd.

HPGL is a trademark of Hewlett-Packard Incorporated

PostScript is a trademark of Adobe Systems Incorporated

Windows 95/98/NT is a trademark of Microsoft Corporation

X Window System is a trademark of Massachusetts Institute of Technology

I-DEAS is a trademark of Structural Dynamics Research Corporation

All other brand or product names are trademarks or registered trademarks
of their respective companies or organisations.

CONTENTS

Chapter 1 *System Overview*

Introduction	1-1
Program Limits	1-3

Chapter 2 *User Interface*

Introduction	2-1
The Graphical User Interface	2-2
Keyboard Input	2-10
Output Files	2-11
Commands and Parameters	2-12
The Help Character (!)	2-13
Parameter Assignment	2-14
Parameter Values	2-16
Command Interpreter Errors	2-21
Confidence Level	2-22
Prompted free format Input	2-24
Built-in Commands	2-25
Command Separator and Comments	2-38
Euler Angles	2-39

Chapter 3

The Pre and Post Processor

Introduction	3-1
The Pre and Post Processor Quick Reference Guide	3-3
Using Expressions	3-11
The APPEND Command	3-21
The BHDATA Command	3-22
The CHECK Command	3-26
The CIRCLE Command	3-27
The CLEAR Command	3-30
The COLOUR Command	3-31
The COMOUTPUT Command	3-33
The CONTOUR Command	3-34
The CONVERT Command	3-38
The COPY Command	3-39
The DEVICE Command	3-41
The DIMENSION Command	3-43
The DRAW Command	3-46
The DUMP Command	3-70
The DXFOUTPUT Command	3-75
The EDIT Command	3-76
The EMIT Command	3-78
The END Command	3-85
The ERASE Command	3-86
The EXTERNALCIRCUIT Command	3-87
The EXTRA Command	3-97
The GAP Command	3-116
The GRAPH Command	3-118
The GROUP Command	3-121
The HARMONICS Command	3-124
The HELP Command	3-128
The INTAREA Command	3-129
The INTCIRCLE Command	3-133
The INTLINE Command	3-136
The LINE Command	3-139
The LMMOTION Command	3-143

The MESH Command	3-146
The MODIFY Command	3-149
The PAN Command	3-160
The PLOT Command	3-161
The POINT Command	3-165
The PRINT Command	3-167
The READ Command	3-168
The RECONSTRUCT Command	3-170
The SET Command	3-175
The SOLVE Command	3-180
The SYMMETRY Command	3-191
The TEST Command	3-193
The TITLE Command	3-195
The TRACK Command	3-197
The UNDO Command	3-201
The UNITS Command	3-202
The VECTOR Command	3-205
The VIEW Command	3-207
The WRITE Command	3-211
The ZOOM Command	3-212

Chapter 1

System Overview

Introduction

This manual describes the OPERA-2d pre and post processor, one of a suite of finite element based programs which can be used as tools in the design of electromagnetic devices of all kinds. OPERA-2d has been developed for use on PC's and workstations by Vector Fields Limited.

OPERA-2d solves a wide range of electromagnetic and electrostatic applications in 2-dimensional xy and axisymmetric coordinates. At present the suite comprises a pre and post processor, optional analysis programs and a DXF file translation program.

Pre and Post Processor

- OPERA-2d/PP: interactive pre and post processor.

Static Fields

- OPERA-2d/ST: magnetostatic and electrostatic field analysis, with non-linear materials. ST includes software to adapt the finite element mesh in order to achieve a specified accuracy.
- OPERA-2d/SP: electrostatic field analysis including the effects of space charge created by beams of particles.

Eddy Currents

- OPERA-2d/AC: steady-state ac eddy current analysis, including use of non-uniform permeabilities from OPERA-2d/ST, complex permeabilities and external circuits. AC includes software to adapt the finite element mesh in order to achieve a specified accuracy.

- OPERA-2d/TR: transient eddy current analysis, with non-linear materials, multiple drives, background dc fields and external circuits.
- OPERA-2d/VL: uniform linear or rotational motion induced eddy current analysis, where the topology does not change with time. VL includes software to adapt the finite element mesh in order to achieve a specified accuracy.

Rotating Machines

- OPERA-2d/RM: transient analysis of rotating machines, with variable speed rotor, mechanical coupling and all the features of TR.

Linear Motion

- OPERA-2d/LM: transient analysis of general motion (linear and/or rotating), driven as variable speed or with mechanical coupling, and all the features of TR.

Stress and Thermal

- OPERA-2d/SA: stress analysis including external forces and using forces calculated from electromagnetic solutions.
- OPERA-2d/TH and OPERA-2d/THTR: steady-state and transient thermal analysis using external heat fluxes and ohmic heating calculated from electromagnetic solutions.

Utility

- OPERA-2d/DXF: A program which translates a DXF file to an OPERA-2d command input (.comi) file.

The following chapters of this Reference Manual describe the pre and post processor in detail. Chapter 2 describes the Command Line and Graphical User Interfaces and Chapter 3 describes the commands which are used to prepare data for and process the results from the analysis programs.

More information on how to use the pre and post processor and analysis programs, including worked examples and tutorials, is given in the OPERA-2d User Guide.

Program Limits

The OPERA-2d programs have limits on the maximum numbers of regions, elements, etc. which can be created by the pre and post processor.

The current limits of the first three sizes are given in the following table:

OPERA-2d Program Limits			
	Size 1	Size 2	Size 3
max number of regions	2000	2000	2000
max number of elements	20000	100000	400000
max number of nodes	45000	225000	900000
max number of sides/polygon	1000	1000	1000
max number of drives (TR)	2000	2000	2000
max number of emitters (SP)	50	50	50

- Workstations are supplied with Size 2, but can be re-sized by the user.

Chapter 2

User Interface

Introduction

The interactive pre and post processor of OPERA-2d has a user interfaces which comprises both a command line, and a Graphical User Interface (GUI).

The GUI generates text commands which have the same syntax as the commands which can be typed in directly at the keyboard. There are some features which can be used only from the keyboard. This chapter gives full details of the command line interface. In subsequent chapters, the keyboard commands are described with indications, where appropriate, of the corresponding GUI interaction. The use of the GUI for the pre and post processor follows in the next section.

Within this manual, different fonts are used to differentiate between input and output of various types. The program's commands, parameters and keywords are shown in **SANS-SERIF FONT**; **input** and output from the program in **bold** and normal weight teletype font. File names are shown in a *slanted font*. GUI items are shown in a **narrow tele-type font**.

The Graphical User Interface

The GUI is built from 8 types of input window which are selected and controlled by pointing with the cursor and clicking a mouse button. Some input windows accept characters input from the keyboard. The input windows are:

- **Horizontal menu:** Only used for top level menu.
- **Vertical menu:** For selecting commands and options.
- **ParameterBox:** For entering numerical or character data.
- **DialogBox:** Combination of text inputs and switches.
- **FileBox:** For selection of files.
- **CDBox:** For selection of current directory or folder.
- **ColourBox:** For re-defining colours.

The GUI also uses MessageBoxes to display messages and questions on the display.

Menus

Menus are horizontal or vertical lists of keywords which indicate actions to be performed. Menu items are selected by pointing with the mouse and clicking its left button. When the cursor is pointing at a menu item, that item is highlighted.

Alternatively, menu items can be chosen using the keyboard arrow keys: ← and → for a horizontal menu or ↑ and ↓ for a vertical menu. When the required item is highlighted, it can be selected using the <Enter>, <Return> or ↵ keys.

Selecting a menu item can have one of several effects; the action is indicated by a symbol at the right-hand side of the item:

Symbol	Action
↓	Drop Down: this activates a sub-menu. It only exists in the top-level horizontal menu.
	Pull Right: activates a sub-menu.
	Pick and Pull: activates a sub-menu after a selection from the displayed model (see Pick below).
	Return: returns to higher-level menu.

Symbol	Action (<i>continued</i>)
	Toggle: swaps between 2 options and the symbol changes between  and  . The current state of the program is displayed.
	Option: chooses one from a set of options. The current choice is indicated by  (filled with red).
	Pick: must be followed by a selection from the displayed model. This is done by positioning the cursor (which changes shape to \oplus) over the required part of the model and pressing the left mouse button.
	Rubber-box: must be followed by selection of diagonally opposite corners of a rectangle. This is done by pressing the left mouse button with the cursor at one corner and dragging the mouse, with the button held down, releasing it at the opposite corner. The menus are automatically hidden while the rubber-box is being used.
	Action: executes a command or requests additional information via a ParameterBox or both; sometimes the menu will close after the specified action.

The **<Esc>** key can be used to escape from a menu without any actions. If the menu does not allow Pick operations, selecting from a higher level menu can also be used to close it.

Not all menu items can be used at all times. For example, it is not possible to execute post processing commands if there is no solution or no mesh. Unavailable menu items are displayed in pale-blue rather than white until they become available as the result of other commands.

Parameter Boxes

ParameterBoxes (figure 2.1)(figure 2.2) are used to input information from the

```

Size                = 10
X coordinate of eye = 2
Y coordinate of eye = 1
Z coordinate of eye = 3
Accept              Dismiss
  
```

Figure 2.1 A Typical ParameterBox

```

Minimum X coordinate = 0
Maximum X coordinate = 10
Minimum Y coordinate = 0
Maximum Y coordinate = 10
Accept              Dismiss
  
```

Figure 2.2 A Typical ParameterBox

keyboard. Default values (if they exist) are displayed and are initially highlighted. When a value is highlighted it can be replaced by the first characters typed. A value can be edited by moving the text cursor before typing any character keys.

Most ParameterBoxes have **Accept** and **Dismiss** below the list of parameters. These can be selected with the mouse to execute or escape from the command. The mouse can also be used to identify a parameter to be edited.

Editing parameter values and controlling the execution of the command can be achieved with the following keys:

- \uparrow , \downarrow and **<Tab>** can be used to move between the parameters.
- \leftarrow and \rightarrow move the text cursor within the value being edited.
- **<Enter>**, **<Return>** or \downarrow move to next parameter. If **Accept** is highlighted or there is only one parameter, the command is executed.
- **<Esc>** escapes from the command.
- **<Back-space>** or **<Delete>** delete characters.
- To toggle insert mode: **<Insert>** (Windows) or function keys **<F2>** or **<PF2>** (X-windows)
- To move to start of the value **<Home>** (Windows) or function keys **<F3>** or **<PF3>** (X-windows)

- To move to end of the value: **<End>** (Windows) or function keys **<F4>** or **<PF4>** (X-windows)

FileBoxes and CDBoxes

FileBoxes (figure 2.3)(figure 2.4) are used for selecting a file name for reading or

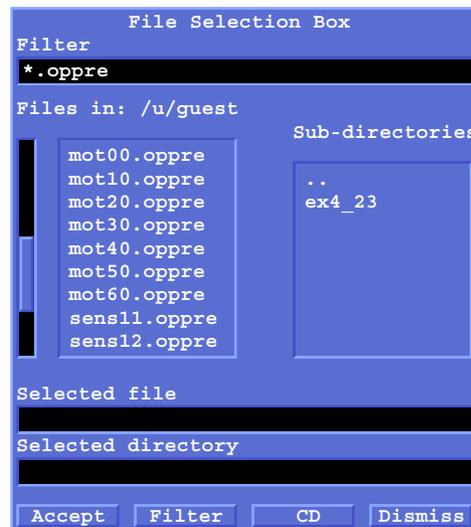


Figure 2.3 A File Selection Box

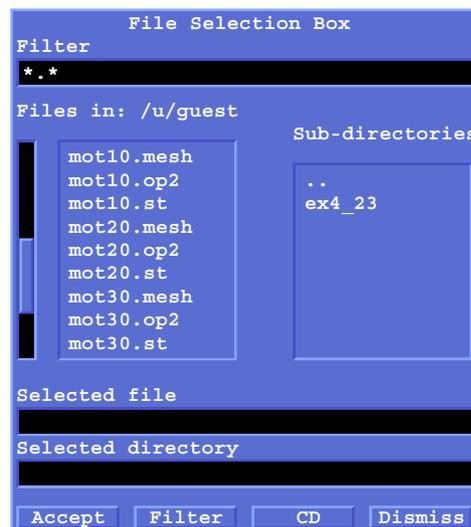


Figure 2.4 A File Selection Box

writing. The FileBox contains a filter string which specifies a subset of all files.

If the filter is edited, it can be acted on by typing **<Enter>**, **<Return>** or **↵** or by selecting the **Filter** button.

If there are more files than can be displayed in the FileBox a scroll-bar is displayed at the left side of the list of names. Similarly, if the longest file name is too long for the FileBox a scroll bar is displayed below the list of names. The size of the slider within the scroll bar indicates the proportion of the text which is displayed. The text can be scrolled in two ways:

- clicking above or below the slider in the vertical scroll bar, or to the left or right of the slider in the horizontal scroll bar, scrolls the text by one page in the direction indicated.
- dragging the slider, by pressing and holding the left mouse button while moving the mouse scrolls the text in the direction indicated while the mouse is moving.

One file should be selected from the list of files. Double-clicking (selecting the file twice in quick succession) confirms the selection. Alternatively, the selection can be confirmed by selecting the **Accept** button. The required file name can also be typed into the selection box and accepted by typing **<Enter>**, **<Return>** or **↵**.

The current directory or folder name is shown and its sub-directories are also displayed in a second selection area. The current directory can be changed by using a double-click selection in the same way as for a file name, or by typing the directory name into the selection box and typing **<Enter>**, **<Return>** or **↵**. Any change of directory in a FileBox is remembered for the next time the FileBox is used, unless the current directory is changed using the CDBox which resets the directories for all FileBoxes.

The CDBox implements the Change Directory command within the GUI. It displays a list of subdirectories, which can be selected by double-click or typing in the same way as within the FileBox. If the new directory includes a disk or device name, it can only be selected by typing the full name into the selection box. When the current directory is as required, the CDBox can be closed by typing **<Esc>** or by selecting the **Quit** button.

In FileBoxes and CDBoxes, file tree-names and directory names can be given using environment variables (UNIX and Windows only). Environment variables **\$VFDIR** (on UNIX systems) and **%VFDIR%** (on Windows systems) are defined by the software as the parent directory of the software.

DialogBoxes

DialogBoxes are used to input information using a combination of keyboard and mouse operations. Within a DialogBox there can be:

- Text-inputs: black rectangles. Initially the first text-input is selected and any characters typed will appear there. Any text-input can be selected with the mouse before typing. The **<Enter>**, **<Return>**, **↵**, **↓** or **<Tab>** keys can be used to move to the next text-input. The **↑** key can be used to move to the previous text input. Within a text-input the value can be edited using the editing and function keys defined for ParameterBoxes (see [page 2-4](#)). Selecting a down arrow button  to the right of a text input activates a FileBox to supply a file name for the text-input (see [page 2-5](#)).
- Switches: small squares or 'radio-buttons'. The switches can be on  (red) or off  (blue). The state of a switch can be changed by selecting with the mouse pointer. Turning on one switch might turn others off if the options are mutually exclusive.
- Buttons: labelled rectangles. These are used to action the selected options, or exit without issuing a command. A button can be selected using the mouse or, if it is high-lighted, with the **<Enter>**, **<Return>** or **↵** key. The **<Esc>** can also be used to exit without issuing a command.
- Scrolling lists: a list of items from which one or many can be selected using the mouse. If the list is long or wide, scroll-bars can be used to view other parts of the list in the same way as for FileBoxes ([page 2-5](#)).

It is important to remember that DialogBoxes often contain more items than are needed at a particular time. Only those items required should be selected.

MessageBoxes

MessageBoxes are used by the GUI to communicate important information to the user. There are 5 types of MessageBox each containing black text on a grey background.

- Information: the results of commands, warnings etc. These boxes are labelled with a large **i**. Information MessageBoxes can be dismissed by typing any key on the keyboard (except **<F1>**) or with the left mouse button.
- If the quantity of information exceeds the size of the window, a scroll-bar is displayed to enable the whole message to be viewed. The window will show the top of the message as the scroll is generated, but other parts of the message can be viewed by dragging the scroll bar up or down.

- Errors: these include Pick operations outside the model space and bad values in ParameterBoxes. The program gives the user another chance to perform the input if an error occurs. Error MessageBoxes are labelled with a large !. They can be dismissed by typing any key on the keyboard (except <F1>) or with the left mouse button.
- Questions: these always require a choice between **YES** and **NO**. QuestionBoxes are labelled with a large ?. They can be dismissed by selecting either the **YES** box or the **NO** box.
- Input: these always require additional information to be given by the user via a ParameterBox, a DialogBox or a FileBox which appears below the Message-Box. The boxes disappear when the information has been supplied.
- Timer: these indicate how much of an operation has been completed. Timers are only displayed for operations for which the estimated elapsed time is greater than 5 seconds. Timer boxes cannot be dismissed, but will automatically disappear when the operation is complete.

ColourBoxes

The ColourBox (figure 2.5) is used to redefine colours used on the display. It con-

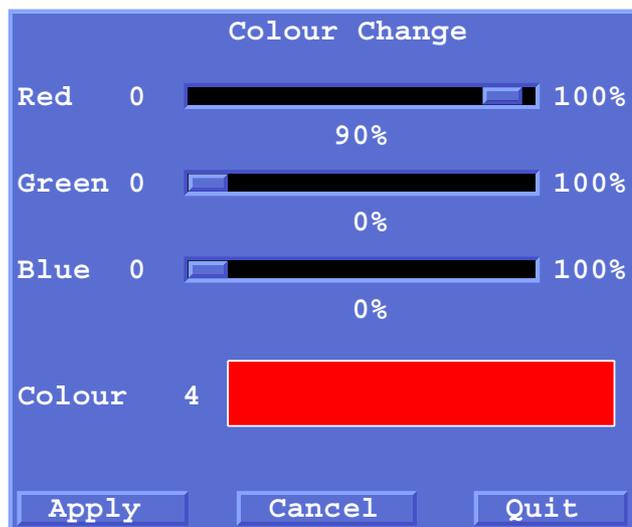


Figure 2.5 A ColourBox

sist of 3 horizontal slider bars, one each for red, green and blue, a square showing

the colour as it is changed and three buttons, **Accept**, **Cancel** and **Quit**.

The colour can be adjusted by moving the sliders in two ways:

- clicking to the left or right of the slider decreases or increases the amount of a colour by 10%.
- dragging a slider by pressing and holding the left mouse button and moving the mouse to the left or right decreases or increases the amount of a colour while the mouse is moving in proportion to the distance between the cursor and the slider.

The buttons **Accept** a colour by changing the display to use the colour in the colour square; **Cancel** the changes by restoring the colour to what it was when the ColourBox was opened; and **Quit** the ColourBox.

Note that on some types of display, the new colours are not shown until the picture is redrawn.

Hiding and Leaving the Menus

Sometimes it is necessary to hide the menus so that the complete picture can be seen. This can be done using the <F1> function key. bring the menus back again.

To leave the menus completely the top-level menu item **MENU-OFF** should be selected. To return to menu-mode, the keyboard command ^ should be given. (This is the caret character followed by <Enter>, <Return> or ↵.)

Keyboard Input

A typical keyboard input consists of a command to perform some action, together with parameters that determine how the action is performed. At other times the input is a list of 'free-format' keywords or numbers which provide additional input to an earlier command. Keyboard input is requested by a prompt of the form

```
name >
```

where `name` is the name of the program or program section being used or sometimes is a question to be answered. Alphabetic input can be in either upper or lower case.

Even in keyboard mode, some commands require graphical input. This is provided by positioning the cross-shaped cursor on the graphics window and typing a key on the keyboard or a mouse button.

Some commands are 'built-into' the command interpreter. Whenever a prompt of the form given above is issued, built-in commands can be typed, by starting the input line with `$`.

Output Files

All user input and the responses from the program are stored in a dialogue file, *Opera2d_PP_n.lp*. User input is written to a file called *Opera2d_PP_n.log*

log files are in a format which can be used as input to the program with the **\$ COMINPUT** command. Graphical input or cursor commands are included in the log files; they can be read back into the program when the **\$ COMINPUT** command is accessed from the GUI. (**\$ COMINPUT** is described on [page 2-30](#).) A unique set of files is created for each run of the programs. The lowest available value of **n** is used for all files.

lp and *log* files are stored in a sub-folder (sub-directory) of the project folder (current directory) named *opera_logs*.

Additional output files can be created by the user to contain the program's usual output or user-defined output or both. The commands to do this are described on [page 2-32](#).

Commands and Parameters

Commands and parameters control the programs. All commands and their parameters may be shortened to their minimum unambiguous form. In any case, a maximum of 16 characters is used, except for file name parameter values. For example, in order to input the **RECONSTRUCT** command, any of the following character strings could be typed: **RECONSTRUCT** or **RECONSTRUC** or **RECONSTR** or **RECONST** or **RECONS** or **RECON** or **RECO** or **REC**. The 2 characters **RE** will not be sufficient, because other commands also begin with these characters. The programs' command interpreter will return an intelligible message when an error is detected in the input. If **RE** had been input as a command the interpreter would reply:

```
DCOD Message 2: Command 'RE' ambiguous (CMND)
```

The Help Character (!)

Short help messages on the commands and their parameters can be obtained at any time by entering the help escape character; this is the exclamation mark (!). Entering a single exclamation mark on a new line will cause a list of all the commands to be displayed, together with a one line description of each command's function. Entering a command name followed by a single exclamation mark will produce a one line description of the command, followed by a list of all the command's parameters with their current value and a description of their function.

- Examples: (Not all the commands are shown here.)

```
OPERA-2d > !
Valid commands are:
HELP Obtain help on how to use OPERA-2d
DRAW Input region data
READ Read files of region and mesh data
END End OPERA-2d/PP
$... Built-in commands. Type '$ !' for a list.
OPERA-2d > read !
Read files of region and mesh data
Parameter Value Meaning
FILE File name
CASE 1 Results case number
SOL1 The name of an additional solution
table
SOL2 The name of an additional solution
table
SOL3 The name of an additional solution
table
```

Parameter Assignment

Parameter values are specified either by entering an assignment instruction

```
parameter=value
```

or positionally by entering the values for the parameters in sequence. Both forms of specification may be mixed, in which case specifying

```
xmin=value1 value2 value3
```

implies that **value2** is assigned to the next parameter after **xmin** and **value3** to the one after that. The parameter sequence for a command is fixed in the order listed by the help escape character **!**. When assignment instructions are used to specify the value of parameters the order is not important, except when expressions which reference other parameters are used (see examples on [page 2-18](#)).

Parameter assignments may be separated either by a comma or spaces; any number of spaces may be used, but if two commas are used in positional input mode this implies that the parameter value is not supplied. Whichever input mode is being used, a comma at the end of an input line implies that the command will be continued on a subsequent line. In this case the parameters entered on the first line are assigned in the program, but the action is not initiated. The first parameter on a continuation line must be assigned explicitly, i.e. using **parameter=value** syntax.

Parameters are unique to the command with which they are associated. The only exceptions to this are the parameters **COMPONENT**, **VX** and **VY**. These take expressions to define the output field quantities. In this case the expression(s) given in one command become the default values for other commands which use the parameters.

The value of the parameters associated with a command are in general initialized to sensible defaults when the programs start, although there are cases where it is not sensible to provide a default. For example, there is no default for the file name with the **READ** command. The last value used for a parameter (in a command) becomes the default value for that parameter the next time the command is used, except in cases where this could be disastrous. The exceptions are obvious, for example, with commands that delete objects the object names or numbers will not be defaulted to the last value.

- Example: Using the following **RECONSTRUCT** command as an illustration:

```
OPERA-2d > reconstruct !
```

```
Reconstruct picture of region data
Parameter Value Meaning
XMIN      0.0  Minimum X coordinate
XMAX      10.0 Maximum Y coordinate
YMIN      0.0  Minimum Y coordinate
YMAX      10.0 Maximum Y coordinate
```

- Example: Assignment instruction mode:

```
OPERA-2d > reco xmin=-5 xmax=10 ymin=-10 ymax=5
```

- Example: Positional input mode

```
OPERA-2d > reco -5 10 -10 5
```

- Example: Mixed positional and assignment

```
OPERA-2d > reco xmax=10 -10 5
```

- Example: Missing positional input, **YMIN** takes last value by default

```
OPERA-2d > reco -5 10,,5
```

Parameter Values

There are 4 types of value which may be assigned to a parameter: **Numeric**, **Expression**, **Character** and **Boolean**. Some parameters can take several value types but some combinations, such as character and expression are not allowed. Error messages indicate if an inappropriate value type has been used, e.g.

DCOD Message 19: Parameter 'TYPE' cannot take numeric values (DECODE)

Numeric Parameter Values

Numeric values are used in many commands for specifying position, size, number of objects etc. Numeric values can be integer, fixed or floating point REAL or DOUBLE PRECISION numbers.

- Examples:

```
23
1.2
3E5
-5.789E+04
2.305983743795d5
-0.04
```

Expressions in Parameter Values

Most parameters which can take numeric values can also take algebraic expressions to specify the values. Expressions used in this way for data input are a replacement for a calculator. Variables within such expressions can be other parameters, system variables or user variables (see the **\$ PARAMETER** and **\$ CONSTANT** commands on [page 2-27](#)). These input expressions are not remembered; they are evaluated and the result is stored.

Parameters which cannot take expressions as values are those which also take character values. The text functions, **%INT()** and **%REAL()** ([page 2-19](#)) provide a way of getting around this restriction.

Expressions are also used to specify user defined parameters for output field quantities in post processing. These expressions are remembered and used for evaluation when referenced. Variables in output expressions can also include the position and the field components. Full details are given in Chapter 3.

Within expressions, variable (parameter) names cannot be abbreviated. If a command parameter is used in an expression, its name must be typed in full.

The following characters can be used in expressions, with their usual FORTRAN meanings: + - / * (). No spaces can be included within expressions.

The following functions are supported, again using their usual FORTRAN definitions:

FUNCTIONS	
Arithmetic	
ABS (a)	the modulus of a
INT (a)	the largest integer whose magnitude does not exceed the magnitude of a times the sign of a
NINT (a)	the integer closest to a
MAX (a ; b)	the maximum of a and b
MIN (a ; b)	the minimum of a and b
MOD (a ; b)	the remainder when a is divided by b
SIGN (a ; b)	the modulus of a times the sign of b
Trigonometry (angles in radians)	
ACOS (a)	the angle whose cosine is a
ASIN (a)	the angle whose sine is a
ATAN (a)	the angle whose tangent is a , angle in the range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
ATAN2 (a ; b)	the angle whose tangent is b/a taking into account the signs of a and b and allowing a to be zero, angle in the range $(-\pi, \pi)$
COS (a)	the cosine of a
COSH (a)	the hyperbolic cosine of a
COTAN (a)	the cotangent of a
SIN (a)	the sine of a
SINH (a)	the hyperbolic sine of a
TAN (a)	the tangent of a
Trigonometry (angles in degrees)	
ACOSD (a)	the angle whose cosine is a
ASIND (a)	the angle whose sine is a
ATAND (a)	the angle whose tangent is a , angle in the range $(-90, 90)$

FUNCTIONS (continued)	
ATAN2D (a;b)	the angle whose tangent is b/a taking into account the signs of a and b and allowing a to be zero, angle in the range $(-180, 180)$
COSD (a)	the cosine of a
SIND (a)	the sine of a
TAND (a)	the tangent of a
Exponentials and logarithms	
EXP (a)	the value of e^a
LOG (a)	the natural logarithm of a
LOG10 (a)	the common logarithm of a
SQRT (a)	the square root of a

N.B. Functions with 2 arguments, `ATAN2` and `MOD`, use ‘;’ to separate the arguments, since ‘,’ is the separator between parameter assignments.

- Example: Parameter assignments:

```
OPERA-2d > line x1=x1+10 x2=x2+10
```

- Example: order is important since the expressions are decoded in the order given; the following commands are not equivalent:

```
OPERA-2d > line x1=y1+4 y1=y1+3
```

```
OPERA-2d > line y1=y1+3 x1=y1+4
```

- Example: Output components in post processing:

```
OPERA-2d > cont comp=sqrt(x**2+y**2)*bx
```

Character Values for Parameters

Character values are character strings, starting with an alphabetic character. In most cases the value is compared against a list of valid options. In such cases the value can be abbreviated to its minimum non-ambiguous length. Specifying the help character, ‘!’, will cause the program to give a list of the valid options.

In other cases character values are used to give file names. In such cases longer strings are permitted. For operating systems which allow file name extensions or file types, the types are added automatically, the precise type being determined from the context. On systems where file names are case sensitive, file names which are entirely upper-case are given upper-case extensions; other file names are given lower-case extensions.

File names given as tree-names can include environment variables within the directory part of the name. Variables `$VFDIR` (UNIX) and `%VFDIR%` (Windows) are defined by the software as the parent directory or folder holding the software.

Some character strings are used for titles or text messages. For these strings, the rule about the first character being alphabetic can be relaxed. However any string which contains spaces or commas must be enclosed in quotation marks (`'`). Quotation marks embedded within character strings must be paired. The GUI automatically supplies quotation marks when necessary.

Superscripts can be used in title strings. This is done by including the characters `~E` before and `~A` after the superscripts. It is not necessary to use `~A` if the string ends with superscripts.

- Examples:

```
OPERA-2d > read file=quadrupole
OPERA-2d > cont style=zone
OPERA-2d > title string='Field after 10~E-3~As'
```

The last example would give the title:

```
Field after 10-3s
```

Boolean Parameter Values

Boolean parameters take the values YES or NO and are in general used for switching features on or off. Boolean values can also be specified by `+PARAMETER` or `-PARAMETER`, being equivalent to `PARAMETER=YES` and `PARAMETER=NO`.

Some parameters can take boolean or character values

- Examples:

```
OPERA-2d > reco +label
OPERA-2d > cont auto=yes
OPERA-2d > line aver=no
```

Text functions

It is sometimes necessary to insert the value of an expression into a parameter value as a character string, for example, to include an index number in a file name or to supply a value by expression to a parameter which cannot take expressions. This can be done using two text functions:

%INT(*expression*)

%REAL(*expression*)

These functions evaluate the expressions given and replace *%function(expression)* on the command line with characters representing the value (**%REAL**) or the nearest integer to the value (**%INT**).

For example, to specify a region number using a user variable:

```
inta reg1=%int(#ireg), reg2=%int(#ireg)
```

Similarly, the text function

%EXPR(*variable*)

replaces itself with the expression which *variable* represents. The *variable* can be the parameter of a command, e.g. **COMP** or a user variable. It can be used to modify the expression. In the following example, the first component expression is modified by dividing by a constant:

```
cont comp=bmod/hmod
cont comp=%expr (comp) /mu0
```

This second command is equivalent to

```
cont comp=bmod/hmod/mu0
```

Command Interpreter Errors

The command interpreter provides input error recovery facilities. If a parameter name is mistyped, the other assignments on the input line will be performed, unless they are positional assignments whose position can not be determined, but command action will not continue. The incorrect parameter(s) can then be re-specified without having to retype the whole input line. The same applies to errors detected in the value of a parameter. The command interpreter will display any portion of the input string which it cannot recognize or which it thinks is in error so that the user can see which parameters need to be re-specified.

Confidence Level

Experienced users rely on the last used defaults and the mixed assignment and positional input modes to make efficient use of the programs. There are other useful features in the interpreter which can be used to reduce the amount which has to be typed.

Repeated commands

If the same command is being repeated many times the command name need not be supplied, providing that an assignment instruction starts the input line.

- Examples:

```
OPERA-2d > reco ymax=100
OPERA-2d > ymax=10
```

Prompted input of parameter values

The final feature of the command processor is its prompted input mode. Issuing a command followed by two help escape characters (!!) puts the command interpreter into prompt mode. Each parameter is displayed together with its default value and description. The default is accepted by pressing the **<Enter>**, **<Return>** or ↵ key, or a new value may be entered. When all the parameters have been offered the program waits for either **<Enter>**, **<Return>** or ↵ to be pressed, which then executes the command, or if '\$ABORT' is entered the command is aborted. '\$ABORT' can be used instead of any parameter value to abort the prompting at that point and not execute the command. '\$SKIP' can be used to skip over the remaining parameters and execute the command.

Note that Boolean parameters cannot be specified using **+PARAMETER** or **-PARAMETER** when in prompt mode. The character values **YES** and **NO** should be used instead.

- Example:

```
OPERA-2d > reco !!
There are 24 parameters
For each parameter:
hit return to accept default
OR enter new value
OR type $HELP for help
```

OR type \$SKIP to skip remaining parameters and execute command

OR type \$ABORT to skip remaining parameters and abort command

NO. Name Value Meaning

1 XMIN 0 Minimum X Coordinate

!! > -10

2 XMAX 10 Maximum X Coordinate

!! > 20

3 YMIN 0 Minimum Y Coordinate

!! > -10

4 YMAX 20 Maximum Y Coordinate

!! > 20

Type return to obey command, or \$ABORT to abort

!! >

Prompted free format Input

Once a specific option has been selected by command or graphical input the programs may prompt for extra input to define further parameters. In such cases the user is shown the parameters required and asked to provide values. The parameters are input in free format using **<space>** or comma as the parameter separator. The order of the parameters in this type of input is shown by the prompt, however parameters defined in the manuals as optional keywords may be specified in any order. Free format input lines cannot be continued on subsequent lines by means of a comma.

So that expressions can be used in free format input, each item is given a name according to its position on the line. #1 is the first; #2 the second etc.

Built-in Commands

‘Built-in’ commands provide control-structures (loops and conditions), user variables, command input from files and access to the operating system. ‘Built-in’ commands can be used at (almost) any prompt.

\$ at the start of an input line introduces a ‘built-in’ command or **\$**-command. There is a built-in dictionary of commands and parameters and the normal

```
$ command parameter=value ...
```

syntax can be used. The parameters have been ordered so that it is natural to use positional assignments. The parameter names are useful to provide on-line documentation using the **!** character. Except where noted, there are no default values. **\$**-commands must be specified in full and cannot be continued on subsequent lines.

Limitations

The code which implements these loops and control-structures has the following limitations. Control structures can be nested to a depth of 20 levels. The maximum number of commands from the first control command to the last **\$ END** (inclusive) is 10000.

Loops

Three types of loop are available: **\$ DO**, **\$ FOR** and **\$ WHILE**. In each case the commands between the loop command and the corresponding **\$ END** command are executed a number of times.

\$ DO-loops

The **\$ DO**-loop is similar to the FORTRAN do-loop. At the start of each execution of the loop, an index-variable is set to a value specified by a starting value, a final value and an increment. The syntax of the command is

```
prompt > $ DO index start final increment
... commands to be executed ...
prompt > $ END DO
```

The *index* should be the name of a user-variable, (up to 5 characters, beginning with #). Its value can be changed within the loop, but is always set to the correct value (c.f. **\$ CONSTANT**, see [page 2-27](#)) at the start of the loop.

start, *final* and *increment* can be specified as numerical values or expressions. Expressions are evaluated before the first pass through the loop. If *increment* is omitted it has a default value of 1.

\$ FOR-each loops

The **\$ FOR**-each loop executes a set of commands with a user-variable set in turn to each of the expressions given on the **\$ FOR** command. The syntax is

```
prompt > $ FOR index ex1 ex2 ex3 ... ex9
... commands to be executed ...
prompt > $ END FOR
```

At least one, and at most 9, expressions (ex_n) can be given. *index* is assigned in turn to each expression (c.f. **\$ PARAMETER**, see [page 2-27](#)) at the start of the loop.

\$ WHILE-loops

The **\$ WHILE**-loop executes a set of commands while a logical expression remains true. The syntax is

```
prompt > $ WHILE ex1 logical_operator ex2
... commands to be executed while logical expression is true ...
prompt > $ END WHILE
```

The logical operators are **EQ**, **NE**, **LE**, **LT**, **GE** and **GT**.

Conditional commands

Three conditional commands are available: **\$ IF**, **\$ ELIF** and **\$ ELSE**. The commands **\$ IF** and **\$ ELIF** should be followed by a logical expression. The **\$ ELSE** command has no parameters.

A **\$ IF** block (the commands executed if the logical expression is true) is terminated by a **\$ ELIF**, **\$ ELSE** or **\$ END IF** command.

A **\$ ELIF** (else-if) block is terminated by a **\$ ELIF**, **\$ ELSE** or **\$ END IF** command; it is only executed if the logical expression is true and none of the preceding **\$ IF** or **\$ ELIF** blocks at the same level have executed.

A **\$ ELSE** block is terminated by a **\$ END IF** command; it is only executed if none of the preceding **\$ IF** or **\$ ELIF** blocks at the same level have executed.

The syntax is

```
prompt > $ IF value1 logical_operator value2
... commands to be executed if logical expression is true ...
prompt > $ ELIF value1 logical_operator value2
... commands to be executed if previous blocks have not been executed and logical
expression is true ...
prompt > $ ELSE
... commands to be executed if previous blocks have not been executed ...
prompt > $ END IF
```

The logical operators are **EQ**, **NE**, **LE**, **LT**, **GE** and **GT**.

The \$ END command

The **\$ END** command ends the current block (**DO**, **FOR**, **IF** or **WHILE**). Although the block type is not logically necessary, it must be specified to ensure that the user knows which block is being **ENDED** and to help in supplying the correct number of **\$ END** commands. The syntax is

```
prompt > $ END block_type
```

User Variable Commands

The **\$ CONSTANT**, **\$ PARAMETER** and **\$ STRING** commands define user variables. Two further commands, **\$ ASK** and **\$ PROMPT**, request the user to supply values for user variables and are described on [page 2-31](#).

Each of the three commands has the same two parameters. The first defines the **NAME** of the user variable and the second the **VALUE**. If the name is used again then the value for that variable is overwritten. If no **VALUE** is given, the current value for the **NAME**d user variable is displayed. If **NAME=!** is used then all the user variables currently defined are listed.

The second parameter gives the **VALUE** for the user variable.

Numerical Variables

Numerical user variable names, defined with **\$ CONSTANT** or **\$ PARAMETER**, start with **#** and have up to 16 characters.

The **VALUE** can be a simple numeric value or can be an expression referencing other user variables or system variables. The **\$ CONSTANT** command evaluates the **VALUE** at the time the command is used and any expression is lost. The

\$ PARAMETER command stores the expression given by the **VALUE** parameter so that it can be re-evaluated each time the variable is referenced.

- Example - to define degrees to radians conversion factor:

```
OPERA-2d > $ cons #fac pi/180
Assign a value to a user variable
Name Value      Expression or Value
#FAC  0.0174533  0.0174533
OK
```

It is possible to write simple programs using the **\$ CONSTANT** and **\$ PARAMETER** commands. The user parameters are evaluated at the time they are defined and again whenever they are referenced. Thus changing a user variable definition implies a change in all user parameters which reference that variable. This is shown by the following example. Note how changing the value for #A implies a change in value for #B and #C.

- Example

```
OPERA-2d > $ cons #a 3
Assign a value to a user variable
Name Value  Expression or Value
#A   3.0    3.0
OK
OPERA-2d > $ para #b #a**2
Assign an expression to a user variable
Name Value  Expression or Value
#B   9.0    #A**2
OK
OPERA-2d > $ para #c #b-4
Assign an expression to a user variable
Name Value  Expression or Value
#C   5.0    #B-4
OK
OPERA-2d > $ cons name=! value=0
Assign an expression to a user variable
Name Value  Expression or Value
#A   3.0    3.0
#B   9.0    #A**2
#C   5.0    #B-4
OK
```

```

OPERA-2d > $ cons #a 2
Assign a value to a user variable
Name Value Expression or Value
#A 2.0 2.0
OK
OPERA-2d > $ cons n=!
Assign a value to a user variable
Name Value Expression or Value
#A 2.0 2.0
#B 4.0 #A**2
#C 0.0 #B-4
OK

```

Menu Routes to
\$PARAMETER
and
\$CONSTANT

```

OPTIONS↓
Parameters

OPTIONS↓
Constants

Options
User variables

```

The **\$ PARAMETER** and **\$ CONSTANT** commands are also available at many places in the GUI using the menu option Calculator.

Character
Variables

Character variable names, defined with the **\$ STRING** command, have up to 16 characters, starting with a letter.

The **VALUE** can be any character string. The character string can be recovered on (almost) any input line by use of the **NAME** surrounded by ampersands (&). Any quotation marks used to define the string are lost. This allows several strings to be concatenated.

There are two predefined character variables, **NOW** and **TODAY**, which can be used to obtain the current time and date.

- Example - storing a title for later use (note the use of `%real` to obtain a character representation of the value of a system variable (see [page 2-19](#)):

```

OPERA-2d > $ string t1 'Septum Magnet'
Assign a string to a user variable
Name String
T1      Septum Magnet
OK
OPERA-2d > $ string t2 '(RMS error %real(#err)%)'
Assign a string to a user variable
Name String
T2      (RMS error 5.23146%)
OK
OPERA-2d > title '&t1& &t2&' tr
Set a title for the graphics window

```

The title displayed is 'Septum Magnet (RMS error 5.23146%)'.

Command Input Files

The `$ COMINPUT` command allows commands to be read from a file and additionally sets the message output mode. If a file with no file name extension is given, the extension `comi` is assumed. The syntax is:

```
prompt > $ COMI filename mode
```

If the `$ COMINPUT` command appears in a loop, the file of commands is read each time the loop is executed. Almost any commands can be included in command input files.

Menu Route to `$ COMINPUT`:

```

FILE↓
  Commands in

```

Text Output Modes

The parameter `MODE` applies whether or not a command file is requested. In `PAGED` and `CONTINUOUS` modes, the text output continues until the next input is requested, and with `MODE=OFF` most of the normal text output does not appear at all. `MODE=PICTURES` is useful for running 'demonstration' command files, since the program pauses for an `<Enter>`, `<Return>` or `↵` before each time the graphics window is cleared, but does not stop when the text window is full. In each `MODE`, text output is written to the dialogue file.

When menus are being used the text output modes are slightly different. While a command file is being read with **MODE=CONTINUOUS**, text output appears on the text window and does not appear in GUI MessageBoxes. An additional option, **MODE=MESSAGE** causes the GUI MessageBoxes to be used.

Menu Route for Text Output Mode:

FILE↓
Commands in

Execution of command files can be interrupted using the settings of the **MODE** parameter. It can also be interrupted by inclusion of **\$ PAUSE** commands. The syntax is:

```
prompt > $ PAUSE seconds
```

\$ PAUSE waits for a number of seconds before continuing. If *seconds* is omitted or is ≤ 0 , the program waits for the user to type **<Enter>**, **<Return>** or **↵** or dismiss a MessageBox before continuing.

Prompting Commands

Command input files can contain user variables, which must be assigned values before the commands are executed. The **\$ ASK** command can be used to request the user to supply a value for a numerical variable (c.f. the **\$ CONSTANT** command). The **\$ PROMPT** command can be used to request the user to supply a value for a character variable (c.f. the **\$ STRING** command). The syntax is:

```
prompt > $ ASK #name prompt_string
prompt > $ PROMPT name prompt_string
```

The optional *prompt_string* is displayed to show what input is required. The value must be supplied at the keyboard before the program will continue.

File Existence Command

The **\$ EXIST** command tests the existence of a file. The syntax is

```
$ exist filename
```

If the file exists, the system variable **FILEEXISTS** is given the value 1; if the file does not exist, **FILEEXISTS** is set to 0.

Error Handling Commands

The **\$ ERRORHANDLER** command selects the behaviour of the command processor after an error in a **\$**-command has been detected. The default behaviour, which can be selected using **\$ errorhandler yes**, is that all commands already stored for execution are ignored.

If **\$ errorhandler no** has been specified, the programs continue to execute stored commands. (Commands are stored during execution of **\$ COMINPUT**, **\$ DO**, **\$ FOR** and **\$ WHILE** commands.)

The **\$ BREAKERROR** command causes the command processor to exit the current loop if an error has been detected. This enables command loops which read (see [page 2-32](#)) to the end of a file without knowing in advance the number of lines in the file. A typical sequence of commands could be:

```
$ constant #i 1
$ while #i eq 1
$ read ...
$ breakerror
other commands
$ end while
```

User Input/Output Commands

There are seven commands for user input and output of files. Before a file can be read or written it must be opened.

```
prompt > $ OPEN stream filename authority ± REDIRECT
```

opens a file on a logical stream number which can be in the range 1 to 10. The file can be used in 4 ways, depending on the **authority**. These are **READ** an old file, **WRITE** a new file, **OVERWRITE** an old file and **APPEND** to an old file.

If **+REDIRECT** is selected for an output file, the output which is written to the dialogue file will be written to the output file as well.

When all input or output has been completed a file can be closed to release its logical stream number or to make it available for opening with different authority.

```
prompt > $ CLOSE stream
```

closes a logical stream number in the range 1 to 10.

The **\$ READ** command takes one line from the file opened on the given logical stream number and assigns any numerical data on the line to user variables. Up to 20 variable names can be given. Any character strings on the line are ignored. The syntax is

```
prompt > $ READ stream #var1 #var2 #var3 ...
```

The **\$ WRITE** command is similar to the **\$ READ** command. The syntax is

```
prompt > $ WRITE stream data1 data2 ... ±OUTPUT
```

Up to 20 data items can be supplied.

A line of output can be built up using several **\$ WRITE** commands. If this is necessary the first **\$ WRITE** commands should have **-OUTPUT**. The last **\$ WRITE** should have **+OUTPUT**. The data from the second and subsequent **\$ WRITE** commands will be positioned after the data of the previous write commands in an internal buffer which is written and re-initialized when **+OUTPUT** is used.

Data items on a **\$ WRITE** command can be numerical, characters or user variables. Before the **\$ WRITE** command is used, the **\$ FORMAT** and **\$ ASSIGN** commands should be used to define the type of data and the style of output to be used for each item on the **\$ WRITE** command.

The **\$ FORMAT** command can be used to define up to 20 different formats for output items. In each form the *width* can be specified as zero which implies that the program should calculate a width to fit the data being written. The syntax has one of the following forms.

- To define a format for character data, truncated or padded with spaces to a particular **width**:

```
prompt > $ FORMAT number CHARACTER width
```

- To define a fixed point format for numerical data, with **decs** as the number of decimal places.

```
prompt > $ FORMAT number FIXED width decs
```

- To define a floating point format for numerical data:

```
prompt > $ FORMAT number EXPONENTIAL width
```

- To define a format for integer data:

```
prompt > $ FORMAT number INTEGER width
```

- To define a format for a user variable to display the expression defining the variable truncated or padded to a particular **width**:

```
prompt > $ FORMAT number USER width
```

- To define a character string to be output irrespective of the data on the **\$ WRITE** command (N.B. It is necessary to define **STRING** formats containing spaces to appear between other data items if those other data items are written with a width of zero.):

```
prompt > $ FORMAT number STRING width STRING=chars
```

- To list the defined formats:

```
prompt > $ FORMAT +LIST
```

The **\$ ASSIGN** command assigns format numbers to the data items of subsequent **\$ WRITE** commands. The first data item will be written with the first non-**STRING** format, the second item with the next non-**STRING** format, and so on. The syntax of the **\$ ASSIGN** command is:

```
prompt > $ ASSIGN form1 form2 form3 ...
```

Up to 20 formats can be assigned.

The final input/output command is **\$ BACKSPACE** which allows a file opened with `$ OPEN` to be backspaced or positioned at its start. The syntax is:

```
prompt > $ BACKSPACE stream records
```

where *stream* is the logical stream number and *records* is the number of records the file is to be backspaced. If *records* is given as `-1`, the file is positioned at its start.

- Example: reading and writing files.

A file *points.dat* contains:

```
10
X 0 Y 0
X 0 Y 1
X 0 Y 3
X 0 Y 5
X 0 Y 7
X 0 Y 9
X 0 Y 10
X 1 Y 1
X 1 Y 5
X 1 Y 9
```

The following commands will read the data and use the coordinates in `POINT` commands to evaluate the field and to write a corresponding output file. Note the use of comments.

```
/          Open input file for reading
$ open 1 points.dat read
/          Read the first line into user variable #np
$ read 1 #np
/          Open output file for writing
$ open 2 fields.dat write
/          Define a string format to space the output
$ form 1 string string=' '
/          Define a floating point format
$ form 2 expo 0
/          Assign format numbers for the output
$ assi 2 1 2 1 2 1 2
/          Start a loop from 1 to #np
$ do #i 1 #np
/          Read #x and #y from input file
$ read 1 #x #y
/          Evaluate fields at #x #y
poin meth=cart xp=#x yp=#y
/          Write coordinates and flux density to output file
$ write 2 x y bx by
/          End of loop
```

```
$ end do
/      Close files
$ close 1
$ close 2
```

The output file, *fields.dat* contains:

```
0.0 0.0 1.179564E-07 -0.000120992
0.0 1.0 6.322637E-08 -0.000121184
0.0 3.0 3.739035E-06 -0.000133139
0.0 5.0 -4.42901E-06 -0.00013248
0.0 7.0 -7.59639E-06 -8.48776E-05
0.0 9.0 -7.74227E-06 -2.32268E-05
0.0 10.0 0.0 0.0
1.0 1.0 4.347521E-07 -0.000121116
1.0 5.0 -1.34893E-05 -0.000141764
1.0 9.0 -3.10067E-05 -3.09315E-05
```

Operating System Commands

There are two commands to execute operating system commands:

```
prompt > $ OS str1 str2 str3 str4 str5 str6
prompt > $ CD directory
```

\$ OS passes up to 6 strings (*str_n*) which together form a command to the operating system to be executed. This can be used to issue single commands to list names of files in directories (folders), delete or rename files, etc., using the normal syntax of the operating system. The output from the commands is listed, with the usual page breaks.

The \$ OS command on UNIX systems

On UNIX systems, in order to redirect the output from the command to a file, the program adds to the command the appropriate notation:

- in c-shell: **user_command >& TeMpOsCmNdFiLe**
- in other shells: **user_command > TeMpOsCmNdFiLe 2>&1**

The contents of the file, *TeMpOsCmNdFiLe*, are then listed. For this reason, shell meta-characters within the user command should be used with care and it might be necessary to enclose the user command in parentheses. For example, to run a background command use the following syntax:

```
$OS (xterm &)
```

**The \$ OS
command on
Windows systems**

On Windows systems a limited set of Command Prompt commands have been implemented, although without full functionality. The commands available are:

- **dir** or **ls** to list files in the current folder
- **del** or **rm** to delete a file
- **mkdir** or **md** to create a new folder in the current folder
- **ren** to rename a file
- **copy** to copy a file
- **pwd** or **cd** to report the current folder

N.B. File names including spaces should be enclosed in double-quotes (").

**Additional \$ OS
Commands to
Launch OPERA**

On all operating systems additional commands are available for starting interactive and analysis programs from the interactive programs. The commands for the interactive programs are:

\$ OS operapp

and for the analysis programs:

\$ OS operaanl *program datafile mode*

where

- *program* is one of **AC**, **RM**, **SA**, **SP**, **ST**, **TH**, **THTR**, **TR** or **VL**
- *datafile* is the name of the OPERA-2d data file to be analysed
- *mode* is **FORE** or **BACK**. The interactive program waits while **FORE**ground analysis jobs are run, but can be continued or ended while **BACK**ground jobs are run.

**The \$ CD
command**

Because the **\$ OS** command spawns a new sub-process, a command such as

```
prompt > $ OS cd directory
```

has no lasting effect. The **\$ CD *directory*** command should be used instead to change the current directory or folder. Directory names can be given using environment variables. Environment variables **\$VFDIR** (on UNIX systems) and **%VFDIR%** (on Windows systems) are defined by the software as the parent directory of the software.

Menu Routes:

FILE↓
OS command

FILE↓
Change directory

Command Separator and Comments

The command separator allows several keyboard commands to be given on one input line. The command separator is the vertical bar, |.

- Example using the **RECONSTRUCT** command with mesh in only one material:

```
OPERA > reco -mesh mate=all | -erase mate=1 +mesh
```

If the first non-space character on an input line is /, the line is treated as a comment. Comments are output to the dialogue file and comments in command files are displayed as the file is being read. In menu mode, comments which start /* are displayed in GUI MessageBoxes.

In some contexts it is not possible to give comments, since for some operating systems, file names can begin with /.

Euler Angles

The program occasionally requires that the orientation of a component or a physical property be defined. Euler angles are always used to define the orientations. The following table shows the Euler angle convention used:

ANGLE	DESCRIPTION
PHI or P	Rotation about the original (global) Z axis positive rotation by right-hand screw convention i.e. from X towards Y.
THETA or T	Rotation about the local Y axis created by the PHI rotation, positive right-hand screw convention i.e. from Z' to X'.
PSI or S	Rotation about the local Z axis created by the PHI and THETA rotations, positive right-hand screw convention i.e. from X' to Y'.

Unless there are parameter names, the convention is always to define the rotation by the ordered triple **THETA PHI PSI** or **T P S**.

N.B. This order is different from the order used in the definition above.

Examples:

Coordinate Transformation	ϑ	Φ	Ψ
$XYZ_{local} = XYZ_{global}$	0	0	0
$XYZ_{local} = YZX_{global}$	90	0	90
$XYZ_{local} = ZXY_{global}$	90	90	180

Chapter 3

The Pre and Post Processor

Introduction

The OPERA-2d pre and post processor is an interactive program for preparing the data and processing the results from the OPERA-2d analysis programs.

The pre processing commands are used to create and edit two dimensional finite element models, define material characteristics for non-linear magnetic, dielectric, thermal or stress components, display pictures of the models and the mesh and output data files in the formats accepted by the analysis programs.

The post processing commands allow many field components, or expressions of field components to be viewed at points, along lines or over the cross-section of the model. Components can be integrated in one or two dimensions and particle trajectories can be calculated in three dimensions.

When the pre and post processor starts, on UNIX operating systems, the first input the user must give is 'device nomination'. This tells the program what graphics options should be used. This is described with the **DEVICE** command (page 3-41) which can be used at any time to reset the device, or change graphics options.

As the pre and post processor starts or when it is restarted with the **CLEAR** command (page 3-30), the program looks for a file called *opera2.comi*, and if it exists, executes the commands it contains. (See "Command Input Files" on page 2-30.)

The Graphical User Interface (GUI) and the style and syntax of the keyboard commands of the pre and post processor are described in Chapter 2, and the particular requirements of the analysis programs in the User Guide. The rest of this chapter describes the commands of the pre and post processor. The "The Pre and Post Processor Quick Reference Guide" on page 3-3 lists the top-level menu

items and the first level 'pull-down' menus, and each of the keyboard commands in the order in which they appear in the program. "Using Expressions" on page 3-11 explains the use of expressions and system variables. Then each command is explained in detail with references to the menu options which can be used to access them (from page 3-21 to page 3-212).

The Pre and Post Processor Quick Reference Guide

The GUI Menus

The top level menu groups the commands of the pre and post processor. The top level menu has the following items:

FILE↓	UNITS↓	OPTIONS↓	DISPLAY↓	HELP↓	FIELDS↓	MODEL↓	MENU_OFF
-------	--------	----------	----------	-------	---------	--------	----------

Each of the top level menu items gives access to a pull-down menu. This first level of menus is listed here, with references to the chapters and sections which describe the commands in detail.

The FILE Menu

The **FILE** menu contains commands which read and write files, access the operating system and end the program.

FILE↓

Data Files	
Read file	Read a data file page 3-168
Write file	Write a data file page 3-211
Start analysis	Start an analysis program using page 2-36 OPERAANL
Graph data in file	Display graph of data in a text file page 3-118
Commands in	Read commands from a file page 2-30
Dump picture	Create a picture file of current display page 3-70
Change directory	Change the current directory page 2-35
System command	Run an operating system command page 2-35
Return	Close the menu
End OPERA-2d/PP	End the pre and post processor page 3-85

The UNITS Menu

The **UNITS** menu allows units to be set to a standard combination (*S.I.*, *c.g.s.*, etc.) or for individual quantities.

UNITS↓

SI Units (magnetic)	Set the units to <i>S.I.</i> (magnetic)	page 3-202
SI Units (electric)	Set the units to <i>S.I.</i> (electric)	
CGS Units (magnetic)	Set the units to <i>c.g.s.</i>	
Imperial Units	Set the units to Imperial (where possible)	
Length unit	Set individual unit quantities	
Flux density unit		
Field intensity		
Potential unit		
Conductivity unit		
Density unit		
Power unit		
Force unit		
Energy unit		
Mass unit		
Return	Close the menu	

The *OPTIONS* Menu

The **OPTIONS** menu contains user variable commands, a group of graphical commands for colour setting, picture file output, titling etc., the **PRINT** command for listing the model data, and program reset command, **CLEAR**.

OPTIONS↓

Calculator...	Define a user variable as a ...	
Parameters	... parameter	page 2-27
Constants	... constant	page 2-27
List variables	List all user variables	
Colour settings	Change the colours of the display	page 3-31
Print model data	List the model data	page 3-167
Clear and Reset	Clear all the data and reset defaults	page 3-30
Dump picture	Create a picture file of current display	page 3-70

Graphics output	Change graphics output options page 3-41
Title	Define a title for the graphics window page 3-195
Dimension	Add dimensioning to the display page 3-42
Return	Close the menu

The *DISPLAY* Menu

The **DISPLAY** menu re-displays the picture. It allows the coordinate limits to be changed graphically by pan or zoom functions or numerically. It also controls the style and contents of the display.

DISPLAY↓

Refresh	Re-display the data page 3-170
Zoom in/out	Re-display, changing the size page 3-212
Pan	Re-display, changing the position page 3-160
	For the next display ...
Axes limits	... set the size page 3-170
Display all	... select all regions page 3-170
Region numbers	... select region numbers page 3-170
Material numbers	... select material numbers page 3-170
Region group	... select region group page 3-170
Post processing	... set style options for post processing page 3-170
Style	... choose the style page 3-170
Axes	... select the axes page 3-170
Nodes	... select nodes page 3-170
+Labels	... switch display of labels page 3-170
+Mesh	... switch display of elements page 3-170
+C_line	... switch display of construction lines page 3-170
+Back	... switch display of background region page 3-170
Return	Close the menu

The *HELP* Menu

The **HELP** menu gives information on how to use the program, the GUI and the keyboard commands. The chapter and section numbers here refer to the location of additional information on the same topics.

HELP↓

System Overview	Overview of OPERA-2d	chapter 1
The GUI	How to use the GUI	page 2-10
Command line	How to use the command line	page 2-10
Model input	How to define a model	
Boundary values	How to define boundary conditions	page 3-154
Field calculations	How to obtain field values	
System Variables	How to use system variables	page 3-11
Return	Close the menu	

The *FIELDS* Menu

The **FIELDS** menu contains all the post processing commands.

FIELDS↓

Field Calculations		
Component	Set the field component	page 3-11
Field options	Set field calculation method and problem symmetry	page 3-175
Point values	Get values at points	page 3-161
...Options	AC time option for point values	
Graphs	Get and display values along lines	page 3-27, page 3-139
Contour plot	Get and display values as contour maps	page 3-34
...Options	Options for contour maps	
Vectors	Get and display values as vectors	page 3-205
Integrals	Integrate field values	page 3-129, page 3-133, page 3-136
Harmonics	Harmonic analysis of field values	page 3-124
Trajectories	Particle trajectory calculations	page 3-197, page 3-207
Calculator	Define a user variable	page 2-27
Tables	Operate on tables of field values	page 3-98
Return	Close the menu	

The *MODEL* Menu

The **MODEL** menu contains the commands to enter data into the pre and post processor. It also includes the mesh generator and the **EXTRA** command which can be used to prepare additional data for the analysis programs.

MODEL↓

Solution Type	Choose analysis options	page 3-175
BH or DE Data	Define non-linear material data	page 3-22
Draw regions	Input model data	page 3-46
Modify regions	Edit model data	page 3-149
Boundary Cond.	Supply boundary conditions	page 3-154
Copy regions	Copy model data	page 3-39
Erase regions	Remove model data	page 3-86
Undo erase	Undo the last ERASE command	page 3-201
Change regions	Swap between region types	page 3-38
Group regions	Create and modify region groups	page 3-121
Emitter data	Create and modify space charge emitter data	page 3-78
Mesh generator	Check data and generate the mesh	page 3-26, page 3-146, page 3-193
Periodic symm.	Connect periodic boundaries	page 3-191
Extra options	Additional input data	page 3-97
Circuits	Create and modify External Circuit data.	page 3-87
Zoom display	Re-display picture	page 3-212
Calculator	Define a user variable	page 2-27
Return	Close the menu	

**Switching
Menus Off**

MENU_OFF switches the GUI off enabling keyboard command mode. The GUI can be switched on again by typing the command

```
OPERA-2d > ^
```

This is the caret character followed by **<Return>**.

Keyboard Commands

The following is a complete list of the ‘top-level’ keyboard commands which can be entered in response to the prompt

```
OPERA-2d >
```

Following sections contain complete descriptions of all the commands and sub-commands in alphabetical order.

- Help Command:

HELP	Obtain System overview, help on command interpreter, data definition commands, post processing commands and file handling commands.
-------------	---

- Mesh definition and editing commands:

DRAW	Define regions (quadrilaterals or polygons) including material properties and boundary conditions.
EMIT	Create, modify and file emitter data for the Space Charge Beam Analysis program.
MODIFY	Modify mesh data: points, subdivisions, materials and boundary conditions.
EDIT	Edit corners and sides of regions
COPY	Copy regions, with displacement and mirror imaging.
CONVERT	Convert regions from one shape code to another.
ERASE	Erase regions.
SYMMETRY	Connect region faces which have periodicity conditions.
UNDO	Return model to the state before the last ERASE or MODIFY .

- Picture display commands:

RECONSTRUCT	Display pictures of the geometry, choosing which parts are shown and how.
ZOOM	Change size of picture by zooming in or out.
PAN	Change position of picture within the graphics window.

- Data grouping and printing commands:

PRINT	Print all or some of the region data.
GROUP	Create and modify groups of region numbers.

- Material B-H characteristic, definition and editing command:

BHDATA	Enter the BH data definition and editing mode.
---------------	--

- Analysis option setting command:

SET	Set analysis options: element type, coordinate system, potential type, field type etc.
SOLVE	Set analysis program specific data.
GAP	Add a gap region to a rotating machine model.
LMMOTION	Add meshing for a linear motion model.

- Mesh generating and data checking commands:

CHECK	Check data thoroughly.
TEST	Test each region against its neighbours and add points if necessary to match faces to prepare for meshing.
MESH	Generate all finite element mesh data.

- Command to impose extra analysis conditions:

EXTRA	Sub-command mode to impose extra conditions on analysis and define stress analysis data.
--------------	--

- External Circuit command:

EXTERNALCIRCUIT	Sub-command mode to define and modify external circuit data.
------------------------	--

- Post processing commands:

POINT	Evaluate field components at a point.
LINE	Evaluate field components along a line and display as a graph.
CIRCLE	Evaluate field components along a circular arc and display as a graph.
PLOT	Evaluate field components along region faces and display as a graph.
CONTOUR	Evaluate field components over regions and display as line or coloured zone contours.
VECTOR	Evaluate field components over a quadrilateral patch and display as vectors.
INTLINE	Integrate field components along a line.
INTCIRCLE	Integrate field components along a circular arc.
INTAREA	Integrate field components over regions.
HARMONICS	Evaluate Legendre polynomial or Fourier coefficients of field components along a line.
TRACK	Evaluate particle trajectories.
VIEW	Re-display or process particle trajectories.

- Commands to read and write data files:

READ	Read a file of OPERA-2d region and mesh data and results.
APPEND	Append a perturbation solution to the solution already in the pre and post processor.
WRITE	Write a file of OPERA-2d region and mesh data and results.

COMOUTPUT	Write a file of OPERA-2d commands defining regions.
DXFOUTPUT	Write a file of DXF data defining regions.
GRAPH	Display graph of data in a text file.

- Program management commands:

DEVICE	Reset or change graphics device.
COLOUR	Redefine colour map.
CLEAR	Clear program data and re-initialize all commands.
DUMP	Write a picture file containing the current display.
TITLE	Control picture titling.
UNITS	Select units for display and evaluation.
DIMENSION	Add dimensions to the display of the model.

- Ending the program:

END	End the pre and post processor.
------------	---------------------------------

Using Expressions

The command decoder used with the pre and post processor is described fully in chapter 2. More details are given here concerning the way in which the pre and post processor makes use of the expression analyser to aid pre processing and evaluate output field components.

Any parameter which takes a numerical value (but cannot also take a character value) can be assigned to an expression in terms of other parameters, system and user variables. These are, in most cases, evaluated on input and the resulting value is stored. The exceptions to this are the numerical parameters of the **MODIFY** command and the output field component parameters. In the **MODIFY** command, any expression given as a parameter value will be evaluated for each region in turn. Thus to double the current density in the whole problem, the following command could be given:

- Example

```
OPERA-2d > modi reg1=1 * dens=dens*2
```

The system variables **AREA** and (in axisymmetry) **VOLUME** are set for each region within **MODIFY** so that material properties such as current or charge density can be set in terms of total values. (N.B. **AREA** and **VOLUME** are calculated for each region during display of that region. After **READING** a data file, values for **AREA** and **VOLUME** will be invalid until the **RECONSTRUCT** command has been used.)

Most post processing commands have parameters **COMPONENT** or **VX** and **VY**, to define scalar and vector field components to be displayed. Expressions can be used to define these output field quantities with the variables being system variables, user constants, user parameters and the parameters of the commands. The expressions are evaluated for each field point.

The expressions for **COMPONENT**, **VX** and **VY** used with any command become the default value for all other commands which use those parameters. The initial value for **COMPONENT** is **POT** and **VX** and **VY** are initially set to **BX** and **BY**.

The **\$ CONSTANT** and **\$ PARAMETER** commands can be used to perform further calculations on the results of the commands (see [page 2-27](#)).

System Variables

There are 7 sets of system variables: some are always available but others are set depending on the coordinate system (**XY** or **AXI**) and the field type (**MAGNETIC** or **ELECTRIC**) (see **SET** command, [page 3-175](#)). The final set is provided for pre processing.

The post processing system variables hold the coordinates of the field point, normal and tangential unit vectors to lines, basic field quantities, material properties, the results of post processing commands (see the sections on those commands for definitions of the quantities) and physical constants.

In addition, nodal or element values in tables, created by the **EXTRA** command ([page 3-97](#)) or **READ** from files ([page 3-168](#)), can also be used as system variables in post processing, and the **VIEW** command has its own set of variables used in the display of particle trajectories ([page 3-209](#)).

Post Processing System Variables Always Available

Post Processing Variables	
POT	the solution potential at the field point (see SET command, page 3-175 , to find meaning of POT)
MINIMUM, MAXIMUM	the minimum and maximum values of CIRCLES , LINES and CONTOURS , the maximum length of a field VECTOR
ERROR	the error in the element containing the field point in flux density units (see User Guide)
INTEGRAL, TOTAL_INTEGRAL	the integral of the COMPONENT from the INTAREA and INTLINE commands
FLUX	the potential integral from the INTAREA command
ENERGY	the energy integral from the INTAREA command
AREA	the integration area in the INTAREA command
PI	π
MU0	$\mu_0=4\pi 10^{-7}$
EPSILON0	$\epsilon_0 = \frac{1}{c^2 \mu_0}$
RMSERROR	the <i>rms</i> field error, expressed as a percentage (see User Guide)

*Post Processing
System Variables
from Analysis
Data*

Post Processing Variables	
SCALE	the scaling factor of a statics solution.
FREQ	the frequency of a steady-state ac solution
TTIME	the time of transient solution

*System variables
for magnetics in
xy*

XY Magnetics	
X, Y	the coordinates of the field point
TVX, TVY	the tangential unit vector to a line
NVX, NVY	the normal unit vector to a line
DADT	the time derivative of vector potential
HX, HY, HMOD	the field strength at the field point
HCX, HCY	the coercive force at the field point
BX, BY, BMOD	the flux density at the field point
MU	the relative permeability at the field point
HXDX	the field gradient $(\frac{\partial H}{\partial x})^x$ in the element containing the field point
HXDY	the field gradient $(\frac{\partial H}{\partial y})^x$ in the element containing the field point
HYDX	the field gradient $(\frac{\partial H}{\partial x})^y$ in the element containing the field point
HYDY	the field gradient $(\frac{\partial H}{\partial y})^y$ in the element containing the field point
J	the current density at the field point
LX, LY	the Lorentz force density ($\mathbf{J} \times \mathbf{B}$) at the field point
SIGMA	the conductivity in the element containing the field point
ALPHA	the permanent magnet or lamination direction or drive current phase angle or permeability phase angle in the element containing the field point

XY Magnetics (continued)	
VEL	the velocity of the element containing the field point
HDB	the stored energy integral $\int(\mathbf{H} \cdot \partial\mathbf{B})$ at the field point
FX, FY, TOTAL_FX, TOTAL_FY	the force integral from the INTAREA , INTCIRCLE and INTLINE commands
TORQUE, TOTAL_TORQUE	the torque integral in the INTCIRCLE and INTLINE commands
CURRENT	the total current integral from the INTAREA command
POWER	the power loss integral from the INTAREA command

*System variables
for axisymmetric
magnetics*

AXI Magnetics	
R, Z	the coordinates of the field point
TVR, TVZ	the tangential unit vector to a line
NVR, NVZ	the normal unit vector to a line
DADT	the time derivative of vector potential
HR, HZ, HMOD	the field strength at the field point
HCR, HCZ	the coercive force at the field point
BR, BZ, BMOD	the flux density at the field point
MU	the relative permeability at the field point
HRDR	the field gradient $(\frac{\partial H_r}{\partial r})$ in the element containing the field point
HRDZ	the field gradient $(\frac{\partial H_r}{\partial z})$ in the element containing the field point
HZDR	the field gradient $(\frac{\partial H_z}{\partial r})$ in the element containing the field point

AXI Magnetics (continued)	
HZDZ	the field gradient ($\frac{\partial H_z}{\partial z}$) in the element containing the field point
J	the current density at the field point
LR, LZ	the Lorentz force density ($\mathbf{J} \times \mathbf{B}$) at the field point
SIGMA	the conductivity in the element containing the field point
ALPHA	the permanent magnet or lamination direction or drive current phase angle or permeability phase angle in the element containing the field point
VEL	the velocity of the element containing the field point
HDB	the stored energy integral $\int (\mathbf{H} \cdot \partial \mathbf{B})$ at the field point
FR	the radial force integral from the INTAREA command
FZ, TOTAL_FZ	the axial force integral from the INTAREA , INTCIRCLE and INTLINE commands
CURRENT	the total current integral from the INTAREA command
POWER	the power loss integral from the INTAREA command
VOLUME	the volume of revolution from the INTAREA command

*System variables
for electrostatics
in xy*

XY Electrostatics	
X, Y	the coordinates of the field point
TVX, TVY	the tangential unit vector to a line
NVX, NVY	the normal unit vector to a line
EX, EY, EMOD	the field strength at the field point
DVX, DVY, DMOD	the electric displacement at the field point
CHARDEN	the charge density at the field point
EPSILON	the relative permittivity

XY Electrostatics (continued)	
EXDX	the field gradient $(\frac{\partial E^x}{\partial x})$ in the element containing the field point
EXDY	the field gradient $(\frac{\partial E^x}{\partial y})$ in the element containing the field point
EYDX	the field gradient $(\frac{\partial E^y}{\partial x})$ in the element containing the field point
EYDY	the field gradient $(\frac{\partial E^y}{\partial y})$ in the element containing the field point
EDD	the stored energy integral $\int(\mathbf{E} \cdot \partial\mathbf{D})$ at the field point
FX,FY,TOTAL_FX,TOTAL_FY	the force integral from the INTCIRCLE and INTLINE command
TORQUE,TOTAL_TORQUE	the torque integral from the INTCIRCLE and INTLINE command
CHARGE	the total charge integral from the INTAREA command

*System variables
for axisymmetric
electrostatics*

AXI Electrostatics	
R, Z	the coordinates of the field point
TVR, TVZ	the tangential unit vector to a line
NVR, NVZ	the normal unit vector to a line
ER, EZ, EMOD	the field strength at the field point
DVR, DVZ, DMOD	the electric displacement at the field point
CHARDEN	the charge density at the field point
EPSILON	the relative permittivity

AXI Electrostatics (continued)	
ERDR	the field gradient $(\frac{\partial E_r}{\partial r})$ in the element containing the field point
ERDZ	the field gradient $(\frac{\partial E_r}{\partial z})$ in the element containing the field point
EZDR	the field gradient $(\frac{\partial E_z}{\partial r})$ in the element containing the field point
EZDZ	the field gradient $(\frac{\partial E_z}{\partial z})$ in the element containing the field point
EDD	the stored energy integral $\int(\mathbf{E} \cdot \partial\mathbf{D})$ at the field point
FR	the radial force integral from the INTAREA command
FZ, TOTAL_FZ	the axial force integral from the INTAREA , INTCIRCLE and INTLINE commands
CHARGE	the total charge integral from the INTAREA command
VOLUME	the volume of revolution from the INTAREA command

*System Variables
for Pre
Processing*

Pre Processing	
AREA	the area of each region in the MODIFY command
REGIONS	the number of regions
VOLUME	the volume of each region in the MODIFY command (axisymmetry only)
X, Y	the old coordinates of a region corner in cursor MODIFY

User constants

User constants are defined and examined using the **\$ CONSTANT** command (See “User Variable Commands” on page 2-27.). They allow the current value of system variables or expressions to be stored for use in subsequent calculations.

User parameters

User parameters are defined and examined using the **\$ PARAMETER** command (See “User Variable Commands” on page 2-27.). They allow expressions to be stored. The value of user parameters is recalculated from the expression each time it is referenced using the current values of any other parameters or variables.

Expressions

Expressions cannot exceed 75 characters, since they cannot be continued on subsequent lines. Full details of the operators and functions allowed in expressions are described in “Expressions in Parameter Values” on page 2-16. The **\$ PARAMETER** command should be used to ‘program’ the pre and post processor if more complicated expressions are needed.

Examples

Three examples are given here.

- Example - to set the current density in a set of regions to give a total current of 1000 in each.

```
OPERA-2d > modi reg1=10 15 dens=1000/area
```

- Example - to evaluate rA for axisymmetric flux contours, assuming the solution potential is A :

```
OPERA-2d > $ para #ra r*pot
Define and store user parameters
  Name  Value  Expression or Value
#RA    0.0    R*POT
OK
```

It is sometimes necessary to store the current value of a system variable or expression. The **\$ CONSTANT** command copies the current value of a variable or expression into a user defined name. Thus to plot the percentage homogeneity of the flux density along a line the following set of commands could be used. First, the field at the reference point (10,0) is calculated and stored in a constant.

- Example - to calculate the homogeneity of flux density

```
point by meth=cart xp=10 0
$ cons #byrf by
```

The value of **BY** is now assigned to the constant **#BYRF**. Next the homogeneity can be calculated.

```
line 0 0 20 0 comp=100*(by-#byrf)/#byrf
```

Note that the **POINT**, **CIRCLE**, **LINE** and **CONTOUR** commands can calculate homogeneity using the **HOMOGENEITY**, **XREFERENCE** and **YREFERENCE** parameters.

Field Evaluation in Steady-state AC

In Steady-state ac solutions, the magnetic vector potential is expressed as a complex number at each node. The post processor calculates a complex value of flux density, field strength and current density. The post processing commands can access these complex values and functions of the values in 4 ways, set by the **TIME** parameter.

- **TIME=number**: this specifies an angle in degrees around the driving ac cycle. Thus **TIME=0** gives the in-phase component, and **TIME=90** gives the quadrature component.
- **TIME=AMPLITUDE** gives the amplitude of the field quantity.
- **TIME=PHASE** gives the phase angle of the field quantity, defined as the angle at which the field quantity is maximum.
- **TIME=TAVERAGE** gives the time average of the field quantity. For simple quantities, such as the potential or flux density, the time average is zero. For the integrand in the energy integrals, for example $\frac{\mathbf{A} \cdot \mathbf{J}}{2}$, where two oscillating quantities are multiplied together, the time average is non-zero. Such field products have the form:

$$e = a + b \cos 2\omega t + c \sin 2\omega t \quad (3.1)$$

The value of a is returned as the time average, $\sqrt{(b^2 + c^2)}$ as the amplitude, $\text{atan} \frac{c}{b}$ as the phase, and e as the value for a **TIME**= ωt .

For items such as the ‘built-in’ integrands in **INTAREA**, **INTCIRCLE** and **INTLINE**, **TIME=TAVERAGE** is available and so the integrals are evaluated correctly for all values of **TIME**. However, in evaluating functions defined by the

user for the **COMPONENT** parameter, each field quantity within the expression is evaluated with the value of **TIME** given before **COMPONENT** is evaluated and integrated. This means, for example, that

```
COMP=POT*J/2, TIME=TAVE
```

would return a value of zero. To obtain the time average integral for such **COMPONENTS**, 3 integrals should be performed at times 0, 45 and 90. If the results are assigned to user constants **#I0**, **#I45** and **#I90**, the following commands can be used to find the values of *a*, *b* and *c* as parameters **#A**, **#B** and **#C** and the values of the integral equivalent to **AMPLITUDE**, **TAVE** and **PHASE**.

- Example

```
inta time=0 comp=j*mu*hcy
$ cons #i0 integral
inta time=45
$ cons #i45 integral
inta time=90
$ cons #i90 integral
$ para #a 0.5*(#i0+#i90)
$ para #b 0.5*(#i0-#i90)
$ para #c #i45-#a
$ para #ampl sqrt(#b**2+#c**2)
$ para #tave #a
$ para #phas atan2(#c;#b)*180/pi
```

The **APPEND** Command

Menu Route:

FILE↓
Read file → Append

Command Line Parameters:

Command	APPEND	
Parameter	Default	Function
FILE	<i>none</i>	Name of OPERA-2d results file.
CASE	1	The results case number for AC and TR .
TBACKGROUND	0	Time for AC background solutions.
TPERTURBATION	0	Time for AC perturbation solutions.

The **APPEND** command reads a OPERA-2d results file into the program, adding the solution to the solution already in the program. The command is intended to add small perturbation solutions to larger background solutions, e.g. a small Steady-state ac solution added to a larger static field.

The first two parameters define the name of the **FILE** and, for Steady-state ac or Transient results files, the **CASE** number. If no file name extension is given, the extension *ac* (i.e. a steady-state ac results file) is assumed.

If either solution is Steady-state ac, the potentials will be evaluated for one point in time around the ac cycle. The time should be specified as an angle in degrees (**TBACKGROUND** and **TPERTURBATION**) with respect to the driving currents. Unless both solutions are static, the combined solution will be treated as if it had been solved with the transient analysis program and the appropriate value of time will appear at the bottom of a **RECONSTRUCT** display.

The two solutions must match in numbers of elements and nodes, solution potential, coordinate system and field type. Only the potentials and current densities of the perturbation solution are used. The region data and permeabilities of the background solution are kept.

The **BHDATA** Command

Menu Route:

MODEL↓
BH or DE data → BH/DE editing

Command Line Parameters:

Command	BHDATA		
Parameter	Default	Function	
MATERIAL	0	Material code in range 3 to 102.	
TYPE	SAME	Material type:	
		ISOTROPIC	Isotropic (one BH curve)
		LAMINATED	Laminated (one BH Curve and packing factor)
		XANISOTROPIC	The BH curve defines the μ_{xx} component of the permeability tensor.
		YANISOTROPIC	The BH curve defines the μ_{yy} component of the permeability tensor.
	SAME	Keep a previously defined material type.	
PHASE	0	Expression for the phase lag in AC complex permeability solutions.	
PACK		Expression for the packing factor for laminated materials.	

The OPERA-2d analysis programs use material characteristics to relate flux density and field intensity of all materials in the range 3 to 102.

- For *soft magnetic materials* the characteristics should be defined in the first quadrant, with the first values of B and H both zero. The curve should not extend beyond saturation magnetisation; the program extrapolates correctly. Hysteresis cannot be modelled (except by complex permeability in the Steady State AC analysis program – see User Guide).
- For *hard magnetic materials* the operating curve should be defined. The first value of B should be less than or equal to zero. The coercive force is the value of H for zero B. If the characteristic starts in the third quadrant, a second curve should be defined for the orthogonal direction. The easy direction of the mag-

net is set by the **PHASE** parameter in the region data (see the **DRAW** command, page 3-47).

Data should be provided for each material number used, even for linear (constant permeability/permittivity) analysis where the curve will be used to look up the coercive force.

In electrostatics, similar curves can be used to relate the electric field and displacement current.

The **BHDATA** command is used to create, edit or check tables of pairs of values that define the non-linear BH or DE characteristics of magnetic or dielectric materials. There must be at least 5 and not more than 50 entries in each table. Tables can be stored in files or loaded from files which are compatible with those used in other Vector Fields software.

The parameter, **MATERIAL**, specifies the material number to which the BH curve applies. Other parameters must be selected to set the material isotropy type. If a laminated material is being used, a packing factor must be given. If an anisotropic material is required two separate BH curves must be given.

The **PHASE** parameter can be used to give an expression for the complex phase lag when using the complex permeability option of the Steady-state AC Analysis Program.

On entering the **BHDATA** command, the user is presented with the existing BH curve or an empty set of axes. Points can be added or edited using the **BHDATA** command's 8 sub-commands. These allow addition of points on the current curve, modification of existing points, access to files, and data checking. The sub-commands have named parameters but are documented here assuming positional parameter input in the correct order. In the description below the sub-commands are given in upper case and the variable values in lower case. Sub-commands can be abbreviated to single letters.

In the **REPLACE** sub-command, the values of the parameters **B** and **H** are set to the current values for the point being replaced, before any expression for them is decoded. This allows expressions to be used to reset the values of **B** and **H** in terms of their old values.

- Example: to 'dilute' a BH curve; note use of the command loop and the boolean parameter, **-REDRAW**, which prevents re-drawing the graph after each replace command. (In this example, cgs units are assumed.)

```
OP2-BHDA > $ do #i 1 23
OP2-BHDA > r #i h+(b-h)*0.9 h -redr
OP2-BHDA > $ end do
```

A library of BH data files is supplied with the software in directory $\$VFDIR/bh$ (UNIX) or $\%VFDIR%\bh$ (Windows).

BHDATA Sub-commands

The sub-commands of the **BHDATA** command are described in the following table:

Sub-commands	Function
ADD <i>b h</i>	Add a new point to the end of the table. <i>b</i> and <i>h</i> are numeric values of B and H .
CHECK <i>m n</i>	Check the data and display the interpolations of the data used in analysis. <i>m</i> and <i>n</i> specify the first and last point displayed.
DELETE <i>m n</i>	Deletes the points <i>m</i> to <i>n</i> of the curve. <i>n</i> can have the value * to indicate the last point.
INSERT <i>n b h</i>	Inserts a new point after the <i>n</i> th point of the curve. <i>b</i> and <i>h</i> are numeric values of B and H .
LIST	Lists the material type, packing factor, complex permeability phase lag and BH data points associated with that material.
LOAD <i>file</i>	Loads a curve from a file. The file name extension <i>bh</i> is added to the name if no extension is given. Any points already defined are deleted.
QUIT	Leave the BHDATA command.
REPLACE <i>n b h</i>	Replaces the <i>n</i> th point of the curve. <i>b</i> and <i>h</i> are the new values of B and H . ±REDRAW can be used to control whether the curve is re-displayed.
STORE <i>file</i>	Stores the curve in a file. The file name extension <i>bh</i> is added to the name if no extension is given.

Anisotropy in OPERA-2d

It is possible to use anisotropic materials in most of the electromagnetics analysis programs of OPERA-2d. The choice of material type is stored with the BH curve for each material. A combination of some isotropic, some laminated and some anisotropic materials is possible within a single model. Selection of material type is made from within the **BHDATA** command.

- All solvers are able to solve problems using isotropic materials only. The solutions can be either linear or non-linear.

- Models using laminated materials can be analysed using all the electromagnetics solvers except the transient or velocity (with upwinding) solvers. The analysis can be either linear or non-linear. For laminated materials a packing factor must be given. This can be an expression using only the system variables **X**, **Y**, **R** or **Z** which should evaluate to numbers between 0 (equivalent to air) and 1 (equivalent to isotropic). The direction of the normal to the laminations is given by the material property, **PHASE**.
- Laminated materials have an effective permeability that is lower than the isotropic permeability, and which is different in the planes normal and tangential to the laminations.

$$\mu_{normal} = \frac{\mu\mu_0}{p\mu_0 + (1-p)\mu} \quad (3.2)$$

$$\mu_{tangential} = p\mu + (1-p)\mu_0 \quad (3.3)$$

where p is the packing factor, μ is the isotropic permeability of the material, and μ_0 the permeability of free space.

The coercivity of laminated permanent magnets is also reduced by scaling by the packing factor.

- Anisotropic materials require 2 BH curves, one for the local X direction, one for the local Y direction. The direction of the local X and Y axes, relative to the global, is given by the **PHASE** property for each region. All the electromagnetics solvers can use anisotropic materials except transient and velocity when upwinding is used. The analysis must be non-linear.

Note that the modulus of the field strength or flux density is used to calculate the permeability for both directions.

The stress, thermal and space charge solvers ignore all reference to the material type.

The **CHECK** Command

Menu Route:

MODEL↓
 Mesh generator → Check region data

Command Line Parameters:

Command	CHECK
No Parameters	

The **CHECK** command checks the region and mesh data. Some checks can only be made after the mesh has been generated by the **MESH** command. The **CHECK** command is run automatically after the **MESH ERROR=YES** command (page 3-146), but can be run at other times as well.

The following checks are made:

- Region data: consistency between **MATERIAL** code and properties.
- BH data: do BH curves exist for each material?
- Eddy current conductors: uniform conductivity and current density.
- Elements: non-zero areas.
- Boundary conditions: at least one node with assigned potential.
- Solution: matching numbers of nodes and elements.
- Symmetry boundaries: are they all paired?

Some checks result in warnings, others in errors. If the total number of errors is non-zero, the mesh cannot be stored.

The **TEST** command (page 3-193) does geometric checks between adjacent regions.

The CIRCLE Command

Menu Route:

FIELDS↓

Graphs → Around circular arc

Command Line Parameters:

Command	CIRCLE		
Parameter	Default	Function	
RADIUS	none	Radius of arc.	
P1	none	Azimuthal coordinate at start of arc.	
P2	none	Azimuthal coordinate at end of arc.	
XCENTRE	0	X coordinate at centre of arc.	
YCENTRE	0	Y coordinate at centre of arc.	
NP	10	Number of steps along the arc, i.e. NP+1 points.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		number	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
TAVERAGE		Time average.	
AVERAGE	YES	Nodally averaged fields switch:	
		NO	Use shape function derivative fields.
		YES	Use nodally averaged field.
GRAPH	YES	Graph display switch:	
		NO	No graph.
		YES	Values displayed as graph.
PRINT	NO	Printed values switch:	
		NO	Values printed to file <i>opera2.lp</i> .
		YES	Values printed to screen and file <i>opera2.lp</i> .

Command	CIRCLE (continued)		
Parameter	Default	Function	
AUTOMATIC	YES	Automatic graph scale switch:	
		NO	Vertical axis limits set by VMIN and VMAX .
		YES	Vertical axis set automatically.
VMAX	<i>none</i>	Upper limit of vertical axis of graph AUTOMATIC=NO .	
VMIN	<i>none</i>	Lower limit of vertical axis of graph AUTOMATIC=NO .	
ERASE	YES	Old graph erasure switch:	
		NO	New graph drawn on existing axes.
		YES	Graphics window cleared and new axes drawn.
STYLE	AUTOMATIC	Line style:	
		AUTOMATIC	Program chooses a different style for each graph drawn on the same axes.
		0	Solid line.
		>0	Broken line.
COLOUR	AUTOMATIC	Line colour:	
		AUTOMATIC	Program chooses a different colour for each graph drawn on the same axes.
		>0	Colour number.
TITLE	<i>none</i>	Additional title for graph.	
HOMOGENEITY	NO	Homogeneity switch:	
		NO	Values of COMPONENT used.
		YES	Homogeneity with respect to COMPONENT value at RREF , PREF used.
RREF	0	Radial coordinate of homogeneity reference point.	
PREF	0	Azimuthal coordinate of homogeneity reference point.	

The **CIRCLE** command calculates field quantities along a circular arc, defined by its **RADIUS** and the azimuthal coordinates of the end points, (**P1**, **P2**). The circle is in a local coordinate system centred at (**XCENTRE**, **YCENTRE**). The field is evaluated at **NP**+1 points and the values are displayed as a table of numbers or a graph. The integral of the component is calculated using the trapezium rule. Calculation of field values along lines can also be done using the **LINE** command (page 3-139) and the **PLOT** command (page 3-161).

Expressions for the **COMPONENT** can use as variables any of the system variables given in page 3-12. The values used can be the actual values of **COMPONENT** or the **HOMOGENEITY** of the **COMPONENT** with respect to the reference point (**RREF**, **PREF**). The homogeneity is evaluated using, e.g.

$$A_{\text{homo}} = \frac{A(r, \phi) - A(r_{\text{ref}}, \phi_{\text{ref}})}{A(r_{\text{ref}}, \phi_{\text{ref}})} \quad (3.4)$$

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified. The values can be evaluated using nodally averaged fields (**+AVERAGE**) or shape function derivative fields (**-AVERAGE**).

The user can choose whether or not to see printed tables of values or graphs. Printed values appear on the screen and are also written to the dialogue file *opera2.lp*. The graphs can have the vertical axis limits set automatically or by the values of **VMAX** and **VMIN** with **-AUTOMATIC**.

Multiple graphs can be displayed on the same axes, the first being displayed with **+ERASE**, and the rest with **-ERASE**. The graphs can be displayed with different line **STYLE** and **COLOURS** and a **TITLE** in addition to the **COMPONENT** can be added to the key below the graph. The **TITLE** is not remembered and must be supplied each time it is needed. **TITLES** including spaces or commas should be enclosed in quotation marks (' ').

The **MINIMUM** and **MAXIMUM** values along the arc are assigned to system variables.

The **CLEAR** Command

Menu Route:

OPTIONS↓
Clear and Reset

Command Line Parameters:

Command	CLEAR
No Parameters	

The **CLEAR** command puts the pre and post processor back to the state it was in when it first started. It deletes all the data, re-initializes all variables and sets all parameters back to their default values. User variables are not deleted.

If a file called *opera2.comi* exists in the current file directory, it is opened and read as a **\$ COMINPUT** file (see [page 2-30](#)) before control passes back to the user. This allows the user to supply an individual choice of default values for commands such as **COLOUR** ([page 3-31](#)) and **UNITS** ([page 3-202](#)) or define frequently used **\$ CONSTANTS** and **\$ PARAMETERS** ([page 2-27](#)).

The COLOUR Command

Menu Route:

OPTIONS↓
Colour settings

Command Line Parameters:

Command	COLOUR	
Parameter	Default	Function
C0	3	Colour map number for material 0.
C1	4	Colour map number for material 1.
C3	6	Colour map number for material 3.
C4	7	Colour map number for material 4.
C5	8	Colour map number for material 5.
C6	9	Colour map number for material 6.
C7	10	Colour map number for material 7.
C8	11	Colour map number for material 8.
C9	12	Colour map number for material 9.
C10	13	Colour map number for material 10.
C11	14	Colour map number for material 11.
C12	15	Colour map number for material 12.
CMAP	5	Colour map number for line contour maps.
CTEXT	1	Colour map number for text and axes.
CODE	1	Colour map number to be redefined. * can be used for last contour colour.
RED	<i>none</i>	Amount of red for colour CODE.
GREEN	<i>none</i>	Amount of green for colour CODE.
BLUE	<i>none</i>	Amount of blue for colour CODE.

The COLOUR command enables the user to change any of the colours used for the display. This can be done in two ways:

- the colour map numbers associated with the parts of the display can be re-assigned (this only has an effect if the picture is re-displayed). The colour numbers associated with materials 13 to 102 cannot be changed.
- the definition of a colour map number, CODE, in terms of the RED, GREEN and BLUE components can be changed (on most displays, this has an imme-

diate effect, if the colour number appears on the display). **RED**, **GREEN** and **BLUE** should each be in the range 0 to 1.

If a value of **CODE** is given, the current values of the colour map for **CODE** are displayed. If values are also given for **RED**, **GREEN** or **BLUE**, then the colour map for **CODE** is redefined. The current values of **RED**, **GREEN** and **BLUE** can be used in expressions to define the new values.

If the value of **CODE** corresponds to either the first or last contour colour (16 and 230 on most displays with 256 colours), all the contour colours will be changed to give a rainbow of colours or a variation of shades between the colour specified and the existing colour at the other end of the spectrum. **CODE=*** can be used to specify the last contour colour.

On a 16 colour display, with colour map numbers 0 (the background) through 15, the colours 7 through 15 are also used for coloured zone contour maps. On displays with more than 16 colours, a separate set of colours (≥ 16) is used for coloured zone contours. On some displays, especially those with less than 16 colours, the default colour numbers are different; on others it may not be possible to change the colour map.

The menu colours are separate from the colours used for the display and cannot be re-defined.

WARNING: changing colours can make the display invisible. The default colours can be reinstated with the **DEVICE** command (see [page 3-41](#)).

The COMOUTPUT Command

Menu Route:

FILE↓
Write file → Commands out

Command Line Parameters:

Command	COMOUTPUT		
Parameter	Default	Function	
FILE	<i>none</i>	Name of command file.	
REG1	1	Regions to be included.	
		<i>number</i>	Regions in range REG1 to REG2.
		<i>group</i>	Regions in group named.
REG2	*	Last region to be included. * for highest region number.	
BHTABLES	YES	Include BH tables: YES or NO.	

The **COMOUTPUT** writes a file containing the minimum set of pre and post processor commands necessary to recreate the region data for some or all of the regions currently in the program. The commands included are **SET**, **UNITS**, **TEST** and a **DRAW** command for each region and optionally a **BHDATA** command for each material. Finally, **GROUP** commands are included to restore the region group definitions. A comment is inserted before each **DRAW** command giving the region number. In the assumption that the regions already match the **TEST -DRAW** command is included. Switching off the testing allows the file to be read much more quickly.

If any of the regions are polygons, the keyboard sub-commands necessary to recreate them are written to the file. Any construction lines which have been defined will be included as well.

If a **FILE** with no file name extension is given, the extension *comi* is assumed. The file may be edited with the system editor and read back into the pre and post processor using the **\$ COMINPUT** command (see [page 2-30](#)). In particular, the **LENGTH** parameter of the **UNITS** command can be changed in order to scale data which was originally input in the wrong units.

The **CONTOUR** Command

Menu Route: **FIELDS**↓
 Contour plot

Command Line Parameters:

Command	CONTOUR		
Parameter	Default	Function	
COMPONENT	POT	Field component for contours.	
LINES	10	Number of lines or zones	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		number	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
ERASE	NO	Picture erase switch:	
		NO	Old picture not erased.
		YES	Old picture erased.
LABEL	YES	Contour labelling:	
		NO	No labels.
		NUMBERS	Lines labelled with index numbers or colour scale shown.
		YES	Lines labelled with index numbers or colour scale shown.
		VALUES	Lines labelled with line values or colour scale shown.
AUTOMATIC	YES	Automatic contour value switch:	
		NO	Line values set by START , FINISH and INCREMENT .
		YES	Line values set automatically by LINES and the part of the model on the display.
START	0	Value of first line for AUTOMATIC=NO .	

Command	CONTOUR (<i>continued</i>)		
Parameter	Default	Function	
FINISH	0	Value of last line for AUTOMATIC=NO .	
INCREMENT	0	Increment between lines for AUTOMATIC=NO .	
STYLE	LINE	Contouring style:	
		COLOURED	Coloured line contours
		LINE	Monochrome line contours
		NONE	No contours (for use with DEFORMED mesh plots)
		SYMBOL	+ and - symbols
		ZONE	Coloured zone contours
REG1	1	Regions to be contoured.	
		<i>number</i>	Regions in range REG1 to REG2 .
		<i>group</i>	Regions in group named.
REG2	*	Last region to be contoured. * for highest region number.	
MATERIAL	ALL	Material number to be contoured: 0, 1, 3 to 102 or ALL .	
NOT	ANY	Material number not to be contoured: 0, 1, 3 to 102 or ANY .	
HOMOGENEITY	NO	Homogeneity switch:	
		NO	Values of COMPONENT used.
		YES	Homogeneity with respect to COMPONENT value at XREF , YREF .
XREF	0	X or R coordinate of homogeneity reference point.	
YREF	0	Y or Z coordinate of homogeneity reference point.	
DEFORMED	NO	Deformed mesh switch:	
		NO	Use correct nodal coordinates.
		YES	Add VX to X coordinates and VY to Y coordinates.
VX	BX	Field component for offset to X coordinates.	
VY	BY	Field component for offset to Y coordinates.	

The **CONTOUR** command draws contours of field quantities. The parameters allow the user to choose the **STYLE**, to set the number of lines and the values and to limit contouring to specific materials or regions of the problem.

Expressions for the **COMPONENT**, **VX** and **VY** can use as variables any of the system variables given in [page 3-12](#). For Steady-state ac solutions, each system variable in an expression is evaluated at the **TIME** specified.

The values used for contouring can be the actual values of **COMPONENT** or the **HOMOGENEITY** of the **COMPONENT** with respect to the reference point (**XREF**, **YREF**). The homogeneity is evaluated using, e.g.

$$A_{\text{homo}} = \frac{A(x, y) - A(x_{\text{ref}}, y_{\text{ref}})}{A(x_{\text{ref}}, y_{\text{ref}})} \quad (3.5)$$

The areas to be contoured can be restricted by region and material numbers or by region group name using the parameters **REG1**, **REG2**, **MATERIAL** and **NOT** (see [page 3-122](#)).

The system variables **MINIMUM** and **MAXIMUM** are set with the values of the first and last contour values, and the location of the maximum is stored in the system variables **X** and **Y** (or **R** and **Z**).

Line contours

Line contours are selected with **STYLE=LINE** or **STYLE=COLOUR**.

The values of the contour lines can be set automatically from the range of values on the part of the model displayed (**+AUTOMATIC**) and the number of **LINES** or manually (**-AUTOMATIC**) by setting the starting value, the finishing value and **INCREMENT** between the lines. If **INCREMENT=0** the number of **LINES** is used.

In **+AUTOMATIC** mode, **LINES** specifies the minimum number of lines to be drawn. An increment of 1, 2 or 5 times a power of 10 will be chosen so that the number of lines is \geq **LINES**. However, if **LINES** is negative, or the calculated number of lines would be > 100 , then precisely **|LINES|** will be drawn.

The contour line numbers can be switched on or off with **+LABEL** and **-LABEL**. Alternatively, the value of each contour can be used as a label with **LABEL=VALUE**.

Line contours can be drawn in one colour (**STYLE=LINE**) or in many colours (**STYLE=COLOUR**). **COLOURED** line contours are drawn with the colours

available for zones so that if many lines are used the effect is similar to **STYLE=ZONE**.

The use of some system variables (such as **ERROR**, the field gradients and the material properties) within the expression for **COMPONENT** will lead to fields which are discontinuous from one element to the next. Such fields may be impossible to view with line contours since few if any of the contours will cross the elements. Coloured **ZONE** contours will show these fields correctly.

Coloured zone contours

Coloured zone contours are selected with **STYLE=ZONE**.

The values of coloured zone contours can be set automatically (**+AUTOMATIC**) from the values on the part of the model displayed or manually (**-AUTOMATIC**) from the values of **START** and **FINISH**. In both cases **LINES** specifies the number of zones. The maximum number of zones is set according to the capability of the graphics hardware (see [page 3-31](#)). A scale showing the values corresponding to the colour is displayed with **+LABEL**.

Symbols

Symbol contours are selected with **STYLE=SYMBOL**. A + or – symbol is drawn at the centroid of each element. The size of the symbols is given by the values of **COMPONENT**; they are drawn in the current length units. A scaling factor may have to be included in the expression for **COMPONENT** to adjust the size of the symbols.

The range of values used for symbols can be set automatically (**+AUTOMATIC**) from the values on the part of the model displayed or manually (**-AUTOMATIC**) from the values of **START** and **FINISH**. In both cases **LINES** is ignored.

Deformed mesh plots

Deformed mesh plots are selected with **+DEFORMED**.

Contouring is performed on a deformed mesh, with the value of **VX** evaluated at each node added to the X coordinates of the nodes and **VY** evaluated at each node added to the Y coordinates of the nodes. A scaling factor may have to be included in the expressions for **VX** and **VY** to adjust the size of the deformation. Deformed meshes can be displayed without contours using **STYLE=NONE**.

The **CONVERT** Command

Menu Route: MODEL↓
Change regions

Command Line Parameters:

Command	CONVERT	
Parameter	Default	Function
REG1	<i>none</i>	Regions to be converted.
		<i>number</i> Regions in range REG1 to REG2 .
		<i>group</i> Regions in group named.
REG2	<i>none</i>	Last region to be converted. * for highest region number.
SHAPE	<i>none</i>	New shape code: BACKGROUND , H , Q or POLYGON .

The **CONVERT** command changes the shape code of 3 and 4 sided regions. The regions to be converted are given by number or group name using parameters **REG1** and **REG2** (see [page 3-122](#)). The new shape code is given by **SHAPE**.

Any region with shape **C**, **CS**, **H** or **Q** can be converted to shape **BACKGROUND** or **POLYGON**. This allows additional points to be inserted using the **MODIFY** cursor sub-command **I**.

OPERA-2d can either automatically mesh regions, producing an irregular mesh of elements, or regularly mesh quadrilateral regions. The regular meshing gives the user greater control of the element shape, and is useful for demanding applications (for example modelling skin effects accurately).

Four sided **POLYGONS** can be converted to shapes **H** or **Q**, which create a regular mesh.

The COPY Command

Menu Route: MODEL↓
Copy regions

Command Line Parameters:

Command	COPY		
Parameter	Default	Function	
REG1	none	Regions to be copied.	
		<i>number</i>	Regions in range REG1 to REG2.
		<i>group</i>	Regions in group named.
REG2	NONE	Last region to be copied. * for highest region number.	
DX	0	X displacement.	
DY	0	Y displacement.	
THETA	0	Angle of rotation or mirror plane.	
MIRROR	YES	Mirror reflection or rotation switch.	
		NO	THETA gives rotation angle.
		YES	THETA gives angle of mirror plane.

The COPY command makes copies of regions, specified by number or group name using parameters REG1 and REG2 (see page 3-122). The following coordinate transformations are made to form the new regions:

- rotation through an angle THETA with respect to local coordinate origin
- reflection in a mirror plane at an angle THETA through the local coordinate origin
- displacement by DX in the x direction
- displacement by DY in the y direction

A mirror image is created with +MIRROR; a rotated copy is made with -MIRROR.

Regions of shapes C and CS are displaced by adding DX and DY to their local coordinate system origin (XCENTRE, YCENTRE), i.e. the displacement is with respect to the global coordinate system. Other shapes are displaced with respect to their local coordinate systems.

The **COPY** command creates new regions which can be modified to have different material properties or coordinates. The **DRAW** and **MODIFY** commands (page 3-46 and page 3-149) have replication parameters which create copies of regions with displacements, rotations and reflections, but retaining the same material properties for all copies.

The **DEVICE** Command

Menu Route: OPTIONS↓
 Graphics output

Command Line Parameters:

Command	DEVICE
No Parameters	

There are two different graphics implementations of the software:

- **Windows** (available on Windows95, Windows98 and Windows NT4, Windows 2000 and Windows ME operating systems)
- **X-lib** (available on UNIX operating systems).

The **DEVICE** command is only available with X-lib graphics.

Four Graphics Options are available when the program is started and two of them can be reselected using the **DEVICE** command. The options are:

Option	Start-up	DEVICE command	Meaning
SCREEN	*	*	graphics displayed on the screen
FILE	*		all graphics commands stored in one file
BOTH	*	*	graphics on the screen and in a file
NONE	*		no graphics, except that the DUMP command can still be used to create picture files of specific pages.

If the program is started with no screen graphics (**FILE** or **NONE**), the GUI is not available and cannot be made available by the **DEVICE** command.

The initial specification of the graphics option can be stored in an environment variable (UNIX), in both cases called **VFGRAPHICS**. A valid value of **VFGRAPHICS** eliminates the initialization prompt.

Other environment variables which affect the software on UNIX are:

Variable	Meaning
VFWINDOWW	the initial window width in pixels
VFWINDOWH	the initial window height in pixels
VFINV	If this is set to INVERT , the initial setting of text and background colours will be black on white instead of the default of white on black.

On Windows systems, similar functionality can be obtained using the **Options** → **Graphics Window Preferences** menu item on the OPERA Console window.

Picture files can be read by the PICOUT program which is supplied with UNIX versions of the software and is described with the **DUMP** command.

Picture files can also be created using the **DUMP** command (see [page 3-70](#)).

The DIMENSION Command

Menu Route:

OPTIONS↓
Dimension

Command Line Parameters:

Command	DIMENSION		
Parameter	Default	Function	
OPTION	SHOW	Options:	
		ARC	Define an arc in polar coordinates.
		DELETE	Delete dimensions.
		LINE	Define a line in cartesian coordinates.
		LIST	List dimensions.
		SHOW	Display dimensions.
X1	0	X coordinate at start of line.	
Y1	0	Y coordinate at start of line.	
X2	0	X coordinate at end of line.	
Y2	0	Y coordinate at end of line.	
CURVATURE	0	Curvature of a line.	
XCENTRE	0	X coordinate at centre of arc.	
YCENTRE	0	Y coordinate at centre of arc.	
RADIUS	0	Radius of arc.	
ANGLE	0	Azimuthal coordinate at start of arc.	
THETA	0	Angle subtended by the arc.	
LTYPE	0	Line style for dimension line. 0 is solid; 1, 2, 3 ... are broken.	
LABEL	<i>none</i>	Label to be written at centre of dimension line.	
XLABEL	0	Offset in X direction from centre of dimension line for the label.	
YLABEL	0	Offset in Y direction from centre of dimension line for the label.	

Command	DIMENSION (continued)		
Parameter	Default	Function	
ARROWS	FORWARD	Type of arrows:	
		BACKWARDS	Pointing towards the start of the dimension line.
		BOTH	Arrows at both ends of the line.
		FORWARDS	Pointing to the end of the dimension line.
		NONE	No arrows.
ALENGTH	1	Length of arrows.	
ELENGTH	1	Length of bars at the ends of the dimension line.	
ETYPE	0	Line style for end bars. 0 is solid; 1, 2, 3 ... are broken.	
ESHIFT	0	Offset for the bars at the ends of the dimension line. Positive values shift the bars to the right of a LINE or away from the centre of an ARC .	
DIM1	1	First dimension number for options DELETE , LIST or SHOW .	
DIM2	*	Last dimension number for options DELETE , LIST or SHOW .	
COLOUR	1	Colour number used with OPTION=SHOW .	

The **DIMENSION** command defines, deletes, list and displays dimension lines on the model. There are two types of dimension **LINES** and **ARCS**:

- **OPTION=LINE** defines a line from (**X1**, **Y1**) to (**X2**, **Y2**) with a **CURVATURE**.
- **OPTION=ARC** defines a circular arc around (**XCENTRE**, **YCENTRE**) starting at **ANGLE** and subtending an angle of **THETA**.

The dimension lines have the following features:

- A line drawn with a style given by **LTYPE** and a **COLOUR**.
- Arrows at either (**FORWARD** or **BACKWARD**), neither (**NONE**) or **BOTH** ends of the line. The size of the arrows can be set using **ALENGTH**.
- A **LABEL** (a string of up to 80 characters) which is written centred on the centre of the line. The position of the label can be offset from the centre of the line

by (XLABEL, YLABEL). LABELS which contain spaces or commas or start with a non-alphabetic character must be enclosed in quotation marks (').

- End bars, perpendicular to the line, of length ELENGTH, drawn in style ETYPE. The end bars can be offset by ESHIFT. Positive values of ESHIFT move the end bars to the right of a LINE or away from the centre of an ARC.

The DIMENSION command will also DELETE, LIST or SHOW dimensions which have already been defined. These OPTIONS operate on a range of dimension numbers given by DIM1 and DIM2. DIM2=* indicates the highest numbered dimension.

The **DRAW** Command

The geometry of a device to be analysed by OPERA-2d is defined using the **DRAW** command as a set of polygonal areas or regions of the 2d plane. One region can be a **BACKGROUND** region which covers the whole problem space. Other regions must not overlap each other but can lie on top of the **BACKGROUND** region if it exists.

Regions (except for the background region) can have replications so that repeated structures can be defined easily.

In magnetic devices, for example, a region can represent one of the following:

- free-space;
- a conductor with a prescribed or induced current density;
- permeable material with a linear or non-linear material characteristic.

Within each region mesh generation is automatic, based only on the subdivision of the sides (or faces). There are 6 different region **SHAPE** codes which can be divided into 2 classes: quadrilaterals and polygons. The set of 60 parameters only includes parameters for four sided regions, so an additional set of sub-commands, keyboard or cursor, must be used to input the coordinates and face properties of polygons.

Since the parameters are initialized to sensible default values and the values used for one **DRAW** command become the default values for the next, many less than 60 values need to be given with each command.

Background regions are defined with **SHAPE=BACKGROUND** but in other respects they are defined in the same way as polygons.

The gap regions of rotating machines (for analysis using any of the electromagnetic solvers) should be defined with **GAP** command, or for linear motion models with the **LMMOTION** command.

There are obvious advantages in using polygonal regions, including:

- Smaller number of regions.
- Cursor input and automatic matching to existing regions.

The quadrilateral regions have some uses as well:

- Faster mesh generation.

- Predictable mesh.
- High aspect ratios possible, e.g. layers of elements in a narrow gap.

Region property parameters

Menu Routes:

MODEL↓

Draw regions → Region defaults ...material type

and

MODEL↓

Draw regions → Region defaults ... conductor data

Command Line Parameters:

Command	DRAW: Region Properties		
Parameter	Default	Function	
SHAPE	H	Shape code:	
		BACKGROUND	Background polygon (Delaunay subdivision)
		C	Quadrilateral in $R\Phi$ coordinates (Regular subdivision)
		CS	Quadrilateral in $R\Phi$ coordinates (Graded subdivision)
		H	Quadrilateral in XY coordinates (Regular subdivision)
		POLYGON	Polygon (Delaunay subdivision)
MATERIAL	0	Material code:	
		0	Air
		1	Conductor
		3-102	Non-linear material

Command	DRAW : Region Properties <i>(continued)</i>	
Parameter	Default	Function
PERM	1	Relative permeability or permittivity: the constant value used in a linear analysis. (The User Guide shows how the initial permeability is calculated in non-linear analyses.)
DENSITY	0	Current or charge density.
CONDUCTIVITY	0	Conductivity (AC, RM, TR, VL)
PHASE	0	Angle in degrees. Direction of permanent magnet (ST, RM, VL and TR) Direction of normal to laminations Phase angle of drive currents (AC). See comment on definition of phase below (page 3-49).
N	0	Conductor number.
SYMMETRY	0	Eddy current conductor symmetry: 0 Connected conductors: program calculates total current. 1 Disconnected conductors: total current set by DENSITY .
VELOCITY	0	Velocity of moving parts (VL). See comment on units below (page 3-49).

These parameters apply to ALL regions.

SHAPE codes are described in more detail in the following sub-sections. Only polygons can be drawn using the GUI.

The **MATERIAL** code is used for two purposes: first to assign different colours to parts of the model when displayed on a colour terminal, and secondly to check consistency of material properties. This second purpose is obviously more important. The program will **CHECK** (see page 3-26) the following:

- **MATERIAL=0**: must have **PERM=1**, **DENSITY=0** and **CONDUCTIVITY=0**.
- **MATERIAL=1**: must have **PERM=1**.
- **MATERIAL≥3**: must have a BH curve, and can have **CONDUCTIVITY≠0** and **DENSITY≠0**.

Definition of **PHASE angle**

In AC analysis the phase angle is defined as the angle around the ac cycle at which the current density is at its maximum value. This means that the current density has the following form: $J = J_0 \cos(\omega t - \phi)$ where J_0 is the value of **DENSITY** and ϕ is the value of **PHASE** (in degrees).

Units of **VELOCITY**

The region parameter **VELOCITY** is the Z (axisymmetric) or Y (XY symmetry) velocity of the media with respect to the field. In these cases the velocity has the units of length/sec.

In XY models the velocity can also be interpreted as an angular rotational velocity about the global coordinate system origin. In this instance the rotational velocity is still unit converted, and the velocity should be defined as ω/LENGU

Coordinates of **SHAPES H and Q**

Command Line Parameters:

Command	DRAW: H and Q coordinates	
Parameter	Default	Function
X12	<i>none</i>	X1 and X2 .
X34	<i>none</i>	X3 and X4 .
Y14	<i>none</i>	Y1 and Y4 .
Y23	<i>none</i>	Y2 and Y3 .
X1	0	X coordinate of corner 1.
Y1	0	Y coordinate of corner 1.
X2	0	X coordinate of corner 2.
Y2	0	Y coordinate of corner 2.
X3	0	X coordinate of corner 3.
Y3	0	Y coordinate of corner 3.
X4	0	X coordinate of corner 4.
Y4	0	Y coordinate of corner 4.

These parameters apply to **SHAPES H and Q** only.

SHAPES H and Q are curvilinear quadrilaterals in XY coordinates. They are defined by the local coordinates of the 4 corners (**X1**, **Y1**, ... **X4**, **Y4**), and the curvatures of the faces (**P1**, ... **P4**, described below under Face Properties, [page 3-52](#)).

For regions which are rectangular and parallel to the major coordinate axes, additional parameters are supplied: e.g. **X12** sets **X1** and **X2** simultaneously. If **X12** is specified it overrides any values for **X1** and **X2** given individually on the same command and the default values of **X1** and **X2** for subsequent commands. The same rules apply to the other paired parameters (**X34**, **Y14** and **Y23**).

The subdivision of the regions is defined by the parameters **N1**, **N2**, **N3** and **N4**, **B1**, **B2**, **B3** and **B4** described later under Face Properties (page 3-52). For **SHAPE=H** the numbers of elements on opposite faces of the quadrilateral is equal and is defined by **N1** and **N2**. For **SHAPE=Q** the numbers of elements on one pair of opposite faces is equal, and the mesh can be graded between the other pair of faces. In other words, either **N1=N3** or **N2=N4** must be satisfied.

Regions of shape **H** and **Q** cannot be drawn with the cursor, but 4-sided **POLYGONS** which can be defined graphically, can be converted to shape **H** or **Q** (see page 3-38).

Coordinates of **SHAPEs C and CS**

Command Line Parameters:

Command	DRAW: C and CS coordinates	
Parameter	Default	Function
R12	<i>none</i>	R1 and R2 .
R34	<i>none</i>	R3 and R4 .
P14	<i>none</i>	P1 and P4 .
P23	<i>none</i>	P2 and P3 .
P1	0	Φ coordinate of corner 1.
P2	0	Φ coordinate of corner 2.
P3	0	Φ coordinate of corner 3.
P4	0	Φ coordinate of corner 4.

These parameters apply to shapes **C** and **CS** only. Regions with shapes **C** and **CS** can be drawn and modified only in keyboard mode.

Shapes **C** and **CS** are annular sectors, i.e. quadrilaterals in $R\Phi$ coordinates. They are defined by the polar coordinates of the 4 corners ($r1, P1, \dots, r4, P4$). The two curved faces each have constant radius, and are set by **R12** and **R34**. The inside radius can be zero.

For regions which have radial lines connecting the inside and the outside, additional parameters, **P14** and **P23** are supplied: e.g. **P14** sets **P1** and **P4** simulta-

neously. If **P14** is specified it overrides any values for **P1** and **P4** given individually on the same command and the default values of **P1** and **P4** for subsequent commands. The same type of rules apply to **P23**.

The subdivision of the regions is defined by the parameters **N1**, **N2**, **N3** and **N4**, **B1**, **B2**, **B3** and **B4** described later under Face Properties, page 3-52. For **SHAPE=C** the numbers of elements on opposite faces of the quadrilateral is equal and is defined by **N1** and **N2**. For **SHAPE=CS** the numbers of elements on one pair of opposite faces is equal, and the mesh can be graded between the other pair of faces. In other words, either **N1=N3** or **N2=N4** must be satisfied.

Local coordinates and replications

Menu Routes:

MODEL↓

Draw regions → Region defaults ... replications

and

MODEL↓

Draw regions → Region defaults ... local coordinates

Command Line Parameters:

Command	DRAW: Local coordinates and replications	
Parameter	Default	Function
XCENTRE	0	X coordinate of local origin.
YCENTRE	0	Y coordinate of local origin.
ANGLE	0	Angle from global X axis to local X axis.
NX	1	Number of replications with X displacement.
NY	1	Number of replications with Y displacement.
DX	0	X displacement between replications.
DY	0	Y displacement between replications.
MIRROR	NO	Mirror reflection switch.
	NO	No mirror copy.
	YES	Copy reflected in mirror plane.
TMIRROR	0	Angle of mirror plane through local coordinate origin.
ROTATIONS	1	Number of replications with rotational displacement.
TROTATION	0	Angle between rotational replications.

These parameters apply to ALL regions except the **BACKGROUND** region, which cannot have replications.

The local coordinate parameters allow a region to be defined in a convenient local coordinate system and then be rotated and displaced to its correct position in global coordinates. By default the local coordinate system of a region is identical to the global system. If it has to be changed, the local origin (**XCENTRE**, **YCENTRE**) and the angle from the global to local X-axis (**ANGLE**) may be set.

The replication parameters allow a region to have multiple copies, with each copy having the same material properties and boundary conditions. There are three types of replication, which are applied in the following order: displacement in X and Y, mirror imaging, rotations. The **COPY** command also creates copies of regions; new regions created by **COPYING** can be **MODIFIED** to have different material properties and boundary conditions (see page 3-39 and page 3-149).

The displacements in X and Y are defined by the number of copies (**NX** and **NY**) and the displacement between each copy (**DX** and **DY**). This creates a total of **NX*NY** copies including the original.

Mirror images are selected with **+MIRROR**. The geometry is reflected in a plane through the local coordinate origin rotated by **TMIRROR** from the local X axis.

Rotational images are defined by the number of copies (**ROTATIONS**) and the azimuthal displacement between each copy (**TROTATION**).

All the replications are performed in the local coordinate system.

Face properties

Command Line Parameters:

These parameters only apply to the regularly meshed regions defined by **SHAPE=Q, H, C** or **CS**. In the following table *n* can take the values 1 to 4.

Command	DRAW : Face properties (not polygons)		
Parameter	Default	Function	
F_n	NO	Type of boundary condition for face <i>n</i> :	
		DV	Normal derivative of potential = DV_n .
		NO	Internal face.
		SYMM	One of symmetry (periodicity) pair.
		V	Value of potential = V_n .
VN	0	Value of potential if F_n=V .	
DVN	0	Normal derivative of potential if F_n=DV .	

Command	DRAW: Face properties (not polygons) <i>(continued)</i>	
Parameter	Default	Function
NN	1	Number of elements along face n .
BN	0.5	Subdivision bias for face n .
PN	0	Curvature of face n (shapes H and Q).

These parameters do not apply to polygons.

The face parameters set the boundary conditions, subdivisions and curvatures (**SHAPES H** and **Q** only) of quadrilateral regions. Face 1 is the side from corners 1 to 2, face 2 from corners 2 to 3, face 3 from corners 3 to 4 and face 4 from corners 4 to 1.

There are 4 types of boundary conditions set by the parameters F_n . Region faces which are internal to the mesh usually should be assigned boundary condition **NO**. This can also be used on the outside of the mesh where it is equivalent to $F_n=DV$ $DV_n=0$. Boundaries with a fixed potential should have $F_n=V$ with the value of potential set by V_n . Pairs of faces with $F_n=SYMMetry$ can be connected by the **SYMMETRY** command (page 3-191) to have either the same potential, or the same magnitude but opposite sign. The sections in the User Guide on the analysis programs give information on which boundary conditions can be used with each analysis program.

The subdivision of a region into elements is specified by the parameters **N1**, ..., **N4** and **B1**, ..., **B4**. The numbers of subdivisions (**N1**,...,**N4**) specify the number of elements along each face. **N3** and **N4** are ignored for shapes **C** and **H** which have equal subdivision on opposite faces. Shapes **CS** and **Q** must have either $N1=N3$ or $N2=N4$, i.e. one pair of opposite faces has the same subdivision. The parameters B_n specify a bias for the subdivision. If $B_n=0.5$, there will be N_n equal sized element edges along face n . If $B_n \neq 0.5$, the node positions vary quadratically along the face. A bias less than 0.5 has smaller elements nearer the first point of the face. A bias greater than 0.5 has smaller elements nearer the last point. The maximum effect of the bias can be achieved with $B_n=0$ or $B_n=1$.

The curvatures of faces of shapes **H** and **Q** are specified by **P1**,...,**P4**. The curvature is the reciprocal of radius. Curved lines are always minor arcs and follow a path to the left of the straight line if the curvature is positive, and to the right if it is negative, the direction of the line being given in the definition of the faces above. The curvature can only be seen if the subdivision is greater than 1.

POLYGON and BACKGROUND Region Input

Menu Routes:

```
MODEL↓
  Draw regions → Polygon
  Draw regions → Background
```

The parameters described in [page 3-47](#) and [page 3-52](#) only supply region properties, local coordinates and replications for regions with **SHAPE=POLYGON** and **SHAPE=BACKGROUND**. A set of sub-commands, keyboard and cursor, is provided for inputting the coordinates and face properties of polygons. There is a keyboard sub-command for switching to **CURSOR** mode and a cursor sub-command **K** for switching to keyboard mode. The program is initially in keyboard mode, but it remembers any change of mode for subsequent **DRAW** commands.

Sub-commands and Menu Items

DRAW Polygon Keyboard Sub-commands		
Sub-command	Menu Item	Function
CARTESIAN	XY input	Give coordinates of next point (X,Y) in polygon and define properties of the face created.
POLAR	RT input	Give coordinates of next point (R, Φ) in polygon and define properties of the face created.
XSHIFT	X move	Continue polygon by a line in the X direction and define properties of the face created.
YSHIFT	Y move	Continue polygon by a line in the Y direction and define properties of the face created.
RSHIFT	R move	Continue polygon by a line in the R direction and define properties of the face created.
PSHIFT	T move	Continue polygon by a line in the Φ direction and define properties of the face created.
FINISH	Close polygon	Close and store a polygon, defining properties of the last face.
ERASE	Back-track	Erase the most recent point or face of the polygon being entered.

DRAW Polygon Keyboard Sub-commands (<i>continued</i>)		
Sub-command	Menu Item	Function
CURSOR	Mouse input	Switch to cursor input of points, with construction lines and automatic matching to existing regions.
QUIT	Return	Leave the DRAW command. If there is an unfinished polygon, the user is asked to confirm the QUIT .
ZOOM	Zoom display	Re-display the picture, zooming in or out.
LINE	Enter C_lines	Define a straight construction line.
ARC	Enter C_lines	Define a circular arc construction line.
CIRCLE	Enter C_lines	Define circular construction line(s).

Cursor-hits and Menu Items

DRAW Polygon Cursor Sub-commands		
Character	Menu item	Function
<space>	At the Cursor	Start or continue the polygon with a new point at the cursor position or nearest grid point.
A	About centre	Pre-select the next point as the centre of a curved face.
C	C_line input	Define, list, delete construction lines.
E	Back-track	Erase the last point.
F	Close polygon	Close and store the current polygon.
G	Grid size (xy)	Set the size of cartesian grid. If grid size is non-zero, all <SPACE> cursor hits select the nearest grid point.
H		Obtain help.
K	Return	Switch back to keyboard input.
M	Via mid point	Pre-select the next point as the mid-point of a curved face.
N	Nearest C_line	Start or continue the polygon with a new point on the nearest construction line.
O	At old point	Start or continue the polygon with the old point nearest to the cursor.

DRAW Polygon Cursor Sub-commands (<i>continued</i>)		
Character	Menu item	Function
P	Grid size (polar)	Set the size of polar grid. If grid size is non-zero, all <SPACE> cursor hits select the nearest grid point.
Q		Abandon the current polygon and leave the DRAW command. If there is an UNFINISHED polygon, the user is asked to confirm the Q .
S	Element size	Reset the default element size.
T		Type the coordinates of the nearest point.
X	At intersection	Continue the polygon through a new point at the nearest construction line intersection.
Z	Zoom	Re-draw the picture, ZOOMING in or out.

POLYGON Input Sub-commands

The keyboard sub-commands allow complete definition of polygon coordinates and face properties (curvatures, subdivisions and boundary conditions). Seven of the sub-commands, **CARTESIAN**, **POLAR**, **XSHIFT**, **YSHIFT**, **RSHIFT**, **PSHIFT** and **FINISH** have parameters **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV**. When any of these sub-commands sets any of these parameters, the values given become the default values for all the other sub-commands. These parameters are described in full under the **CARTESIAN** sub-command.

In keyboard mode, a polygon must be started by a **CARTESIAN** or **POLAR** sub-command, giving the initial coordinates. This first sub-command only defines a point and the parameters for the face properties are ignored. Subsequent sub-commands **CARTESIAN**, **POLAR**, **XSHIFT**, **YSHIFT**, **RSHIFT** or **PSHIFT** define a point and a face connecting the previous point to the new one. Polygons must be closed by a **FINISH** sub-command which defines the last face of the polygon.

The ARC Sub-command

Menu Route:

MODEL↓

Draw regions	→ Draw Polygon	→Enter C_lines
	Draw Background	→Enter C_lines

Command Line**Parameters:**

Sub-Command	ARC	
Parameter	Default	Function
XCENTRE	0	X coordinate at centre of polar coordinate system.
YCENTRE	0	Y coordinate at centre of polar coordinate system.
R1	0	R coordinate at start of line.
P1	0	Φ coordinate at start of line.
R2	0	R coordinate at end of line.
P2	0	Φ coordinate at end of line.

The **ARC** sub-command creates a construction line: a straight line in a polar coordinate system centered on (**XCENTRE**, **YCENTRE**). It starts at (**R1**, **P1**) and extends to (**R2**, **P2**). If **R1** and **R2** are the same, the line will be an arc of a circle. If **P1** and **P2** are the same, it will be a radial line. Further details on the use of construction lines are given in [page 3-68](#).

The CARTESIAN Sub-command**Menu Route:**

MODEL ↓
 Draw regions → Draw Polygon → XY input
 Draw Background → XY input

Command Line**Parameters:**

Sub-Command	CARTESIAN		
Parameter	Default	Function	
XP	0	X-coordinate of first or next point in polygon.	
YP	0	Y-coordinate of first or next point in polygon.	
CURVATURE	0	Curvature of face from previous point.	
N	1	Number of elements along face.	
BIAS	0.5	Subdivision bias:	
		<0.5	Smaller elements nearer previous point.
		>0.5	Smaller elements nearer new point.

Sub-Command	CARTESIAN (continued)		
Parameter	Default	Function	
F	NO	Type of boundary condition:	
		DV	Normal derivative of potential = DV .
		NO	Internal face.
		SYMM	One of symmetry (periodicity) pair.
	V	Value of potential = V .	
V	0	Value of potential if F=V .	
DV	0	Normal derivative of potential if F=DV .	

The **CARTESIAN** sub-command specifies either the first point (**XP**, **YP**) in a polygon, in which case **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV** are ignored, or a face extending from the previous point to the new point (**XP**, **YP**). **XP** and **YP** are defined in local coordinates. For second or subsequent points the default values of **XP** and **YP** are the coordinates of the previous point. This enables definition of a diagonal line by expressions in terms of **XP** and **YP**.

- Example - to define a line at 45°:
OP2-POLY > **cart xp+10 yp+10**

The **CURVATURE** of the face is defined as the reciprocal of radius. A minor arc is always assumed, the curved line going to the left of the straight line if the curvature is positive. The curvature can only be seen if the subdivision of the line, **N** is greater than 1.

The subdivision of the line is set by two parameters, **N** and **BIAS**. If **BIAS=0.5**, there will be **N** equal size element edges along the face. If **BIAS≠0.5**, the node positions vary quadratically along the face. A bias less than 0.5 has smaller elements nearer the previous point. A bias greater than 0.5 has smaller elements nearer the new point. The maximum effect of the bias can be achieved with **BIAS=0** and **BIAS=1**.

There are 4 types of boundary conditions set by the parameter **F**. Region faces which are internal to the mesh usually should be assigned boundary condition **NO**. This can also be used on the outside of the mesh where it is equivalent to **F=DV DV=0**. Boundaries with a fixed potential should have **F=V** with the value of potential set by **V**. Pairs of faces with **F=SYMMETRY** can be connected by the **SYMMETRY** command to have either the same potential, or the same magnitude but opposite sign. The sections of this Reference Manual on the analysis programs give information on which boundary conditions can be used with each analysis program.

The CIRCLE Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon →Enter C_lines
 Draw Background →Enter C_lines

Command Line Parameters:

Sub-Command	CIRCLE		
Parameter	Default	Function	
TYPE	FILLET	Type of circular construction line:	
		C1	Circle defined by centre and radius.
		C2	Circle defined by centre and a point on the circumference.
	FILLET	Fillets between two straight construction lines.	

The **CIRCLE** sub-command creates construction lines which form complete circles. The construction lines are stores as arcs. There are 3 types of circle:

- **C1** is defined by one point (the centre) and its radius. The graphics cursor is used to select the centre. A **<space>** cursor hit selects the nearest polygon point, or if there are no points, the cursor position or nearest grid position. A **Q** cursor hit aborts the definition. The program prompts for a value for the radius.
- **C2** is defined by two points: the first defines the centre and the second a point on the circumference. **<space>** cursor hits select the nearest polygon points, or if there are no points, the cursor positions or nearest grid positions. A **Q** cursor hit aborts the definition.
- **FILLETS** are circles tangential to intersecting straight construction lines. They are defined by selecting the two straight lines using **<space>** cursor hits. A **Q** cursor hit aborts the definition. Four circles are defined, each one consisting of two construction lines, which are the major and minor arcs connecting the points where the circles and straight lines touch.

Further details on the use of construction lines are give in [page 3-68](#).

The **CURSOR** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → Mouse input
 Draw Background → Mouse input

Command Line Parameters:

Sub-Command	CURSOR	
Parameter	Default	Function
SIZE	1	Default element size.

The **CURSOR** sub-command switches from keyboard input to cursor input of polygons. The cursor sub-command **K** can be used to switch back again.

The parameter **SIZE** selects the default element size. This is used to set the number of subdivisions on non-matching region faces. **N** is given the smallest integer such that **N*SIZE** is greater than the length of the face.

The **ERASE** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → Back-track
 Draw Background → Back-track

Command Line Parameters:

Sub-Command	ERASE
No Parameters	

The **ERASE** sub-command erases the most recent point or face of the polygon being entered.

The FINISH Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → Close polygon
 Draw Background → Close polygon

Command Line Parameters:

Sub-Command	FINISH
Parameters	
CURVATURE, N, BIAS, F, V and DV AS in CARTESIAN sub-command.	

The **FINISH** sub-command creates the last face of a polygon by connecting the last point entered to the first point. The parameters specify the face properties of the last face. The polygon is stored.

Control remains with the polygon input sub-command mode for the input of more polygons. The **QUIT** sub-command should be used to leave the **DRAW** command.

The parameters **CURVATURE, N, BIAS, F, V** and **DV** are defined under the **CARTESIAN** sub-command.

The LINE Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → Enter C_lines
 Draw Background → Enter C_lines

Command Line Parameters:

Sub-Command	LINE	
Parameter	Default	Function
X1	0	X coordinate at start of line.
Y1	0	Y coordinate at start of line.
X2	0	X coordinate at end of line.

Sub-Command	LINE (<i>continued</i>)	
Parameter	Default	Function
Y2	0	Y coordinate at end of line.
ANGLE	0	Rotation angle around start point.

The **LINE** sub-command creates a construction line: a straight line from (**X1**, **Y1**) to (**X2**, **Y2**) rotated by **ANGLE** around (**X1**, **Y1**). Further details on the use of construction lines are give in [page 3-68](#).

The **POLAR** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → RT input
 Draw Background → RT input

Command Line Parameters:

Sub-Command	POLAR	
Parameter	Default	Function
RP	0	Radial coordinate of first or next point in polygon.
PP	0	Φ -coordinate of first or next point in polygon.
CURVATURE , N , BIAS , F , V and DV as in CARTESIAN sub-command		

The **POLAR** sub-command specifies either the first point (**RP**, **PP**) in a polygon, in which case **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV** are ignored, or a face extending from the previous point to the new point (**RP**, **PP**). **RP** and **PP** are defined in polar coordinates with respect to the local coordinate origin. For second or subsequent points the default values of **RP** and **PP** are the coordinates of the previous point. This enables the definition of the coordinates of the next point by expressions in terms of **RP** and **PP**.

The parameters **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV** are defined under the **CARTESIAN** sub-command.

The **PSHIFT** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → T move
 Draw Background → T move

Command Line Parameters:

Sub-Command	PSHIFT	
Parameter	Default	Function
DP	0	Increment in Φ -coordinate for next point in polygon.
CURVATURE, N, BIAS, F, V and DV as in CARTESIAN sub-command.		

The **PSHIFT** sub-command specifies the second or subsequent point of a polygon in terms of an increment in the local azimuthal coordinate of the point with respect to the previous point. The radial coordinate remains the same as that of the previous point.

The parameters **CURVATURE, N, BIAS, F, V** and **DV** are defined under the **CARTESIAN** sub-command.

The **QUIT** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → Return
 Draw Background → Return

Command Line Parameters:

Sub-Command	QUIT
No Parameters	

The **QUIT** sub-command leaves the **DRAW** command. If there is an unfinished polygon, the user is asked to confirm the **QUIT**.

The **RSHIFT** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → R move
 Draw Background → R move

Command Line Parameters:

Sub-Command	RSHIFT	
Parameter	Default	Function
DR	0	Increment in R-coordinate for next point in polygon.
CURVATURE, N, BIAS, F, V and DV as in CARTESIAN sub-command.		

The **RSHIFT** sub-command specifies the second or subsequent point of a polygon in terms of an increment in the local radial coordinate of the point with respect to the previous point. The azimuthal coordinate remains the same as that of the previous point.

The parameters **CURVATURE, N, BIAS, F, V** and **DV** are defined under the **CARTESIAN** sub-command.

The **XSHIFT** Sub-command

Menu Route:

MODEL↓
 Draw regions → Draw Polygon → X move
 Draw Background → X move

Command Line Parameters:

Sub-Command	XSHIFT	
Parameter	Default	Function
DX	0	Increment in X-coordinate for next point in polygon.
CURVATURE, N, BIAS, F, V and DV as in CARTESIAN sub-command.		

The **XSHIFT** sub-command specifies the second or subsequent point of a polygon in terms of an increment in the local X coordinate of the point with respect to the previous point. The Y coordinate remains the same as that of the previous point.

The parameters **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV** are defined under the **CARTESIAN** sub-command.

The **YSHIFT** Sub-command

Menu Route:

```
MODEL↓
  Draw regions → Draw Polygon    → Y move
                Draw Background → Y move
```

Command Line Parameters:

Sub-Command	YSHIFT	
Parameter	Default	Function
DY	0	Increment in Y-coordinate for next point in polygon.
CURVATURE , N , BIAS , F , V and DV as in CARTESIAN sub-command.		

The **YSHIFT** sub-command specifies the second or subsequent point of a polygon in terms of an increment in the local Y coordinate of the point with respect to the previous point. The X coordinate remains the same as that of the previous point.

The parameters **CURVATURE**, **N**, **BIAS**, **F**, **V** and **DV** are defined under the **CARTESIAN** sub-command.

The **ZOOM** Sub-command

Menu Route:

```
MODEL↓
  Draw regions → Draw Polygon    → Zoom display
                Draw Background → Zoom display
```

Command Line Parameters:

Command	ZOOM
No Parameters	

The **ZOOM** sub-command is identical to the top-level command of the same name (see [page 3-212](#)). It re-displays the picture of the geometry, zooming in or out.

POLYGON Input Cursor Sub-commands

Menu Routes:

```
MODEL↓
  Draw regions → Draw Polygon    → Mouse Input
MODEL↓
  Draw regions → Draw Background → Mouse Input
```

The cursor sub-commands allow the definition of the next point in a polygon in three ways: a new point at the cursor position (cursor hit **<space>**), an existing point (cursor hit **O**), or a construction line point (cursor hits **N** and **X**). Polygons are closed and stored with **F**.

A ‘snap to grid’ facility can be switched on by setting the cartesian or polar grid size greater than zero. When the grid is switched on, all **<space>** cursor hits take the nearest grid point. Switching on a cartesian grid also switches off a polar grid and *vice versa*. The grid sizes can be set or re-set following a **G** or **P** cursor hit.

Other cursor hits allow points to be erased (**E**) or their current coordinates displayed (**T**), or keyboard mode can be used instead after cursor hit **K**. Construction line input follows a cursor hit **C**. The picture can be re-displayed using the **Z** cursor hit which uses the **ZOOM** command (see [page 3-212](#)). The **DRAW** command can be left by cursor hit **Q**.

In cursor mode, if a new face exactly matches an existing face, the subdivision and curvature information is copied from the existing face. The subdivision of new faces (which do not match existing faces) is given by the element **SIZE** parameter of the **CURSOR** keyboard sub-command or reset by the **S** cursor sub-command. Curvature can be set by using additional points as the centre of curvature (cursor sub-command **A**) or as a mid-point on a curve (cursor sub-command **M**). Such additional points are pre-selected using **A** or **M**; their positions can then be given using any of **<space>**, **N**, **O**, or **X**; they do not form part of the polygon definition, but are only used to calculate the curvature of the face.

When a region has been completed its geometry is checked against existing regions and additional points are added if necessary to make the regions match. The way in which this is done can be controlled with the **TEST** command (page 3-193).

Cursor hits and Menu Items

DRAW Polygon Cursor Sub-commands		
Character	Menu item	Function
<space>	At the Cursor	Start or continue the polygon with a new point at the cursor position or grid point.
A	About centre	Pre-select the next point as the centre of a curved face.
C	C_line input	Define, list, delete construction lines.
E	Back-track	Erase the last point.
F	Close polygon	Close and store the current polygon.
G	Grid size (xy)	Set the size of cartesian grid. If grid size is non-zero, all <SPACE> cursor hits select the nearest grid point.
H		Obtain help.
K	Return	Switch back to keyboard input.
M	Via mid point	Pre-select the next point as the mid-point of a curved face.
N	Nearest C_line	Start or continue the polygon with a new point on the nearest construction line.
O	At old point	Start or continue the polygon with the old point nearest to the cursor.
P	Grid size (polar)	Set the size of polar grid. If grid size is non-zero, all <SPACE> cursor hits select the nearest grid point.
Q		Abandon the current polygon and leave the DRAW command. If there is an UNFINISHED polygon, the user is asked to confirm the Q .
S	Element size	Reset the default element size.
T		Type the coordinates of the nearest point.

DRAW Polygon Cursor Sub-commands (<i>continued</i>)		
Character	Menu item	Function
X	At intersection	Continue the polygon through a new point at the nearest construction line intersection.
Z	Zoom	Re-draw the picture, ZOOM in or out.

Construction lines

Construction line definition follows cursor hit **C**. A set of 5 keyboard sub-commands can be used to define new lines, list existing lines and delete redundant lines.

Construction lines are straight lines (**LINE**) or arcs (**ARC**). The arc is a linear interpolation in a cylindrical polar coordinate system (i.e. $\phi=mr+c$ with m and c constant). It is only an arc of a circle if the radial coordinates of the end points are the same. The definitions can be given numerically or by choosing, with the cursor, points (region corners) which have already been defined: lines are defined by two **<space>** cursor hits; arcs are defined by either a **C** and two **<space>**s to specify the centre and two end points or by 3 **<space>** cursor hits to specify points on a circle. Cursor defined lines extend beyond the points which define them. Cursor defined arcs are minor arcs if defined by centre and end points or complete circles if defined by 3 points on the circumference.

Construction lines can also be input using the keyboard sub-commands **ARC**, **CIRCLE** and **LINE**.

The sub-commands and their syntax are:

Construction Line Sub-commands	
Command	Parameters and Function
ARC	$xc\ yc\ r1\ \phi1\ r2\ \phi2$ An arc centered on (xc, yc) starting at polar coordinates $(r1, \phi1)$ and ending at $(r2, \phi2)$. Both r and ϕ vary linearly between the end points.
ARC	CURSOR An arc defined by cursor hits: Either: select points for centre with C and end points with <space> or: select 3 points on circumference of circle with <space> .

Construction Line Sub-commands (<i>continued</i>)	
Command	Parameters and Function
CIRCLE	C1 An arc representing a complete circle defined by a centre point chosen with a <SPACE> cursor hit and radius, given in response to a prompt.
CIRCLE	C2 An arc representing a complete circle defined by a centre point and a circumferential point chosen with <space> cursor hits.
CIRCLE	FILLET Arcs representing 4 complete circles which are tangential to two intersecting straight lines, chosen with <space> cursor hits.
DELETE	Delete construction lines, chosen by <space> cursor hit. Use cursor hit Q to quit.
LINE	$x1\ y1\ x2\ y2\ angle$ A straight line from $(x1, y1)$ to $(x2, y2)$ rotated by <i>angle</i> around $(x1, y1)$.
LINE	CURSOR A straight line through and beyond 2 existing points chosen by <space> cursor hits.
PRINT	List all the construction lines.
QUIT	End the definition of construction lines.

Points can be defined on the nearest construction line using cursor hit **N**, or at the nearest intersection of construction lines using cursor hit **X**. The end points of the construction lines are also counted as intersections.

The **DUMP** Command

Menu Route:

OPTIONS ↓
Dump picture

Command Line Parameters:

Command	DUMP	
Parameter	Default	Function
FILE	none	Name of file to contain the picture.
TYPE	POSTSCRIPT	Graphics language:
	HPGL	Hewlett-Packard Graphics Language
	PICOUT	Vector Fields PICOUT Graphics Language
	POSTSCRIPT	Adobe PostScript
SIZE	A4	Paper sizes: HPGL can use A, A3, A4, B, A0D, A1D, A2D, A3D or A4D ; PostScript can use A4, A or USER .
LLX	0	X-coordinate of lower-left corner in mm (TYPE=POST, SIZE=USER).
LLY	0	Y-coordinate of lower-left corner in mm (TYPE=POST, SIZE=USER).
URX	0	X-coordinate of upper-right corner in mm (TYPE=POST, SIZE=USER).
URY	0	Y-coordinate of upper-right corner in mm (TYPE=POST, SIZE=USER).
COLOUR	YES	Colour PostScript: NO implies grey-scale.
FILL	NO	Filled polygons in HPGL: YES or NO .
ORIENT	LANDSCAPE	PostScript paper orientation: LANDSCAPE or PORTRAIT .
SWAP	YES	Swap black and white in PostScript:
	NO	Colours appear as on screen.
	YES	Black and white are swapped.

The **DUMP** command copies the graphics commands used to create the current display to a **FILE** in three different formats. If no filename extension is given, extensions *.ps*, *.hgl* or *.pic* are added as appropriate.

- Adobe PostScript (**TYPE=POST**): This has options for paper size, colour and orientation.
 - There are two standard paper sizes: European **A4** and American **A**. Alternatively **SIZE=USER** allows the image to be printed at any size and any position on the page by the specification of the coordinates of the lower-left and upper-right corners in mm (parameter **LLX**, **LLY**, **URX**, **URY**).
 - If **+COLOUR** is selected, the screen colours will be matched on the paper as closely as possible, except that black and white can be optionally **SWAPPED**. (Black is any colour with an intensity of less than $\frac{1}{256}$ for Red, Green and Blue; white is any colour with an intensity of more than $\frac{255}{256}$ for Red, Green and Blue – see the **COLOUR** command.)
 - Grey-scale pictures can be created in several ways: the colours could be changed to grey using the **COLOUR** command before **DUMPING** (see the **COLOUR** command); a colour picture could be sent to a grey-scale printer; or a grey-scale picture could be stored using **-COLOUR**. In this last case, the grey levels are calculated using the formula $1 - \frac{r + g + b}{3}$. Thus dark colours on the screen become light colours on the paper.
 - Pictures can be orientated in two ways: **PORTRAIT** pictures have the X horizontal along the bottom of the page and the vertical axis up the left-hand side; **LANDSCAPE** pictures have the horizontal axis up the right-hand side of the page and the vertical axis from right to left along the bottom of the page.
- HPGL (**TYPE=HPGL**) pictures are intended for pen-plotters. There is a larger selection of sizes available: the sizes with **D** appended are for drum plotters, the other sizes are for flat-bed plotters.
 - The only other option allows polygon-fill to be selected (**± FILL**). If selected, polygons are filled using shading with parallel horizontal or vertical lines.
- PICOUT (**TYPE=PICOUT**) pictures can be re-displayed or translated using the PICOUT program supplied by Vector Fields for OpenVMS and UNIX systems. It reads graphics files and redisplayes them on the screen or converts them to PostScript or HPGL for printing or plotting. The PICOUT program prompts the user for the name of the input file and (if necessary) the name of the output file. It also prompts for options such as paper size, colour or monochrome, etc.,

although in some implementations these can be supplied by command line options (*-cc*) which are shown below.

Instructions for running PICOOUT are included in the Implementation Notes supplied with the software.

PICOOUT prompts the user for the input it needs. On UNIX systems, some choices can be specified as command line options, for example *-sc* below. Replies to PICOOUT's prompts can be abbreviated.

PICOOUT always prompts for the name of an input file. There are 3 output options: Screen (*-sc*), PostScript (*-ps*) and HPGL (*-hp*).

- **Screen:** If output to the screen is chosen, there are no further options. On workstations and X-terminals, after the graphics window has been created, its size can be adjusted using the window manager and the pictures will be scaled to fit the new size. The original aspect ratio of the pictures will be maintained by increasing either the top or right side margin. PICOOUT pauses for *<carriage-return>* to continue or *Q* to quit at the end of each picture.
- **PostScript:** If PostScript output is chosen, PICOOUT prompts for the name of the file to contain the PostScript program. If the file already exists it will be overwritten. When PICOOUT has finished, this file can be sent directly to a PostScript printer. PostScript output can be written to multiple files with a single page in each file (*-s*) or to one file with many pages in the file (*-m*). The advantage of multiple files is that a single page can be printed without printing all the others. The filenames for the single page, multiple file option are formed by appending *.1*, *.2*, *.3* etc. to the name given at the output file prompt. On some operating systems, it is necessary to name the output file with a single part name in order to make use of this facility.

PostScript output can be created for different paper sizes. There are 2 standard sizes: European A4 (*-a4*) and American A (*-a*) for which PICOOUT knows the paper size. Non-standard user defined paper sizes (*-u llx lly urx ury*) can also be given by specifying the coordinates of the lower left and upper right corners of the drawing area in millimetres.

PostScript output can be orientated on the page in landscape format (*-l*) or portrait format (*-p*). For paper sizes A and A4, portrait format fits 2 pictures on each page.

PostScript output can be in colour (*-c*) or greyscale (*-g*). Colour output assumes white paper, so the black and white of the original screen picture are swapped. Greyscale intensities are inverted so that dark colours appear pale grey and bright colours as dark grey. Some greyscale printer can interpret colour commands and therefore give a choice of using

PICOUT's algorithm for colour to greyscale conversion or the printer's algorithm.

- **HPGL:** If HPGL output is chosen, PICOUT prompts for the name of the file to contain the HPGL commands. Each page of output is written to a separate file. The filenames are formed by appending .1, .2, .3 etc. to the name given at the output file prompt. If a file already exists it will be overwritten. When PICOUT has finished, the files can be sent directly to a HPGL printer or plotter. On some operating systems, it is necessary to name the output file with a simple name so that the extension can be added.

HPGL output can be created for a variety of paper sizes. For flat-bed plotters, sizes A (**-a**), A3 (**-a3**), A4 (**-a4**) and B (**-b**) are available. For drum plotters, HPGL output can be produced for sizes A0D (**-a0d**), A1D (**-a1d**), A2D (**-a2d**), A3D (**-a3d**) and A4D (**-a4d**). The difference between the flat-bed and drum sizes is that the coordinate system origin is at the centre of the plot on drum plotters and at the bottom left corner on flat bed plotters. No PS (paper size) command is put into the files.

HPGL output can include extra line drawing commands to shade filled polygons (**-f**) or polygons can be represented by the outlines alone (**-nf**).

PICOUT OPTIONS

Output style:

- sc Screen. PICOUT redraws the pictures onto the workstation screen.
- ps PostScript. PICOUT creates a file or files of PostScript commands.
- hp HPGL. PICOUT creates files of HPGL commands.

Paper size (ignored with **-sc**):

- size PostScript output can be produced for standard paper sizes A (**-a**) and A4 (**-a4**) or for a user choice of paper size, (**-ullx lly urx ury**). The integer numbers given by *llx*, *lly*, *urx* and *ury* specify the position of the lower left and upper right corners of the picture in millimetres.
- size[d] For flat-bed plotters, HPGL output can be produced for sizes **a**, **a4**, **a3** and **b**; for drum plotters, HPGL output can be produced for sizes **a0d**, **a1d**, **a2d**, **a3d** and **a4d** (**d** for 'drum'.)

Polygon fill (HPGL only):

- f Fill polygons (solid shading).

PICOUT OPTIONS (*continued*)

-nf Do not fill polygons (outlines drawn).

Colour or greyscale (PostScript only):

-c Colour. White paper is assumed, so black and white are swapped.

-g Greyscale. Intensities are inverted, so that dark colours appear as pale grey and bright colours as dark grey.

Picture orientation (PostScript only):

-l Landscape (one picture on each page).

-p Portrait (two half-size pictures on each page).

File output (PostScript only):

-s Single page per file. Each picture is written to a separate file, with filenames generated from the output file name given by the user.

-m All pages in one file.

Help:

-h For help (all other options ignored).

The **DXFOUTPUT** Command

Menu Route:

FILE↓
Write file → DXF output

Command Line Parameters:

Command	DXFOUTPUT	
Parameter	Default	Function
FILE	<i>none</i>	Name of DXF file.

The **DXFOUTPUT** writes a file containing DXF data which describes the faces of regions which are boundaries between different materials. If a **FILE** with no file name extension is given, the extension *dxf* is assumed.

The **EDIT** Command

Menu Route:

MODEL↓

Modify regions → Modify Points by number
 → Modify Sides by number

Command Line Parameters:

Command	EDIT	
Parameter	Default	Function
REG1	<i>none</i>	Regions to be edited.
		<i>number</i> Regions in range REG1 to REG2 .
		<i>group</i> Regions in group named.
REG2	<i>none</i>	Last region to be edited. * for highest region number.
OPTION	<i>none</i>	Option:
		CORNER Edit region corners.
		SIDE Edit region sides
NUMBER	<i>none</i>	Number of corner or side to be edited. * for all corners or side.
NEWX	<i>none</i>	New x-coordinate of corner. This can be an expression in terms of old coordinates.
NEWY	<i>none</i>	New y-coordinate of corner. This can be an expression in terms of old coordinates.
NEIGHBOURS	YES	Edit corners of neighbouring regions:
		NO Only edit regions give by NUMBER .
		YES Also edit region neighbours.
N	<i>old value</i>	Number of elements along face.
F	<i>old value</i>	Type of boundary condition for face:
		DV Normal derivative of potential = DV .
		NO Internal face.
		SYMM One of symmetry (periodicity) pair.
		V Value of potential = V .
V	<i>old value</i>	Value of potential if F=V .

Command	EDIT (<i>continued</i>)	
Parameter	Default	Function
DV	<i>old value</i>	Normal derivative of potential if F=DV .
BIAS	<i>old value</i>	Subdivision bias for face:
CURVATURE	<i>old value</i>	Curvature of face (not for SHAPE s C and CS).

The **EDIT** command modifies the regions of the model, adjusting the coordinates of the corners of the regions or the parameters of the sides of the regions. These and other region properties can also be changed using the **MODIFY** command (page 3-149).

The region or regions to be modified are specified using parameters **REG1** and **REG2** (see page 3-122) and the corner or side number using parameter **NUMBER**. If **NUMBER=***, all sides or corners of the selected regions are edited.

- **OPTION=CORNER**: the coordinates of the region corners are edited. Corners of regions with shape code **C** or **CS** cannot be edited.

When region corners are shared by more than one region, the neighbouring regions can be edited as well (**+NEIGHBOUR**) or left at their original locations (**-NEIGHBOUR**). Using **-NEIGHBOUR** also prevents shared corners being edited more than once when multiple regions are edited.

The new coordinates for the region corners (**XNEW** and **YNEW**) can be given as expressions in terms of the old coordinates (**X** and **Y**). For example, the following command can be used to scale the whole model from inches to cm:

```
edit reg1=1 reg2=* opti=corn xnew=x*2.54,
ynew=y*2.54 -neig
```

- **OPTION=SIDE**: the properties of the region sides are edited. Sides of regions with any shape code can be edited.

Only the parameters specified on a command are edited. The region values of parameters not specified remain unchanged.

New values of **N**, **V**, **DV**, **BIAS** and **CURVATURE** can be specified in terms of their existing values. Regions which share sides with the regions selected for editing are edited as well. The curvatures of sides of regions with shape code **C** or **CS** cannot be edited.

The following command can be used to double the discretisation of the whole model:

```
edit reg1=1 reg2=* opti=side n=n*2
```

The **EMIT** Command

Menu Route: MODEL↓
Emitter data

Command Line Parameters:

Command	EMIT
No Parameters	

There are no parameters with the **EMIT** command; it introduces a set of sub-commands which allow emitter files to be created, read, filed and edited. The **EMIT** sub-commands are as follows:

CREATE	To create a new emitter or line segment.
DELETE	To delete an emitter or line segment.
EDIT	To edit and modify emitter data.
LIST	To list the emitter data.
QUIT	Quit the EMIT command.
READ	To read an old emitter data file.
STORE	To write an emitter data file.
ZOOM	Re-draw the picture, zooming in or out (see page 3-212).

The **CREATE** Sub-command

Menu Route: MODEL↓
Emitter data → Create emit data

Command Line Parameters:

Sub-Command	CREATE		
Parameter	Default	Function	
MODE	EMITTER	Data to be created:	
		EMITTER	Create a new emitter.
		SEGMENT	Create a new line segment for emitter NUMBER .
		PICK	Create a new line segment by selection for emitter NUMBER .
NUMBER	1	Emitter number	

The **CREATE** sub-command is used to create emitters and line segments. The parameter **MODE** controls what type of item is to be created.

- **MODE=EMITTER**: a new emitter is generated with default settings and with the next available emitter number. This number will be 1 if it is the first emitter to be created. The new emitter will take initial default values which will need to be edited with the **EDIT** sub-command.
- **MODE=SEGMENT**: a new segment is added to emitter **NUMBER**. The new line segment will take initial default values which will need to be edited with the **EDIT** sub-command.
- **MODE=PICK**: a new segment is created from a region side which can be selected using the cursor (see “Pick Operations” on page 3-152). The segment is added to emitter **NUMBER**. Only one facet may be selected. The following cursor hit characters may be used:

CREATE Command Cursor Hits	
Cursor hit	Function
<space>	Select a region facet.
A	Accept selected facet and exit CREATE sub-command.
Q	Quit CREATE sub-command without creating a line segment.

A line segment which has been generated by picking a region facet will contain the identical subdivision, bias and curvature data as the facet. This ensures correct matching between region facets and an emitter. Other properties must be set with the **EDIT** sub-command.

The DELETE Sub-command

Menu Route:

MODEL↓
Emitter data → Delete emit data

Command Line Parameters:

Sub-Command	DELETE		
Parameter	Default	Function	
MODE	EMITTER	Data to be deleted:	
		EMITTER	Delete an emitter.
		SEGMENT	Delete a line segment in emitter NUMBER.
NUMBER	none	Emitter to be deleted, or emitter from which a segment is deleted.	
NSEGMENT	none	Line segment number to be deleted	

The **DELETE** sub command is used to delete emitters and line segments. The parameter **MODE** controls what type of item is to be deleted.

- To delete an emitter, use **MODE=EMIT**. The parameter **NUMBER** sets which emitter is to be deleted.
- To delete a line segment from an emitter, use **MODE=SEGMENT** and specify the emitter number with the **NUMBER** parameter and the segment number with **NSEG**.

The EDIT Sub-command

Menu Route:

MODEL↓
Emitter data → Edit emit data

Command Line Parameters:

Sub-Command	EDIT	
Parameter	Default	Function
NUMBER	1	The emitter number to be edited.
STEP	none	Maximum step length in trajectory calculation.

Sub-Command	EDIT (<i>continued</i>)		
Parameter	Default	Function	
TOLERANCE	<i>none</i>	Absolute tolerance for the trajectory calculation.	
DISTANCE	<i>none</i>	Normal sampling distance.	
ETYPE	<i>none</i>	Emitter type:	
		0	Thermal saturation limit
		1	1d Langmuir/Fry limit
		2	1d Child's Law limit
		3	Specified current density
		4	Fowler-Nordheim field emission
		5	Schottky field emission
		6	Extended Schottky field emission
		7	Automatic selection of 4, 5 or 6
		8	1d Langmuir/Fry limit (no virtual cathode)
		10	Maxwell normal velocity sampling
		11	Maxwell normal and tangential velocity sampling
102	Plasma free surface model		
TEMPERATURE	<i>none</i>	Emitter temperature (Kelvin).	
WORK	<i>none</i>	Emitter work function (Volt).	
CONSTANT	<i>none</i>	Emission constant (amp cm ⁻²).	
MASS	<i>none</i>	Particle mass in electron mass units.	
CHARGES	<i>none</i>	Number of charge quanta carried by the particle.	
NTANGENTIAL	<i>none</i>	Number of sample bins to sample tangential velocity.	
NNORMAL	<i>none</i>	Number of sample bins to sample normal velocity.	
LENGTH	<i>none</i>	Maximum distance between sample rays.	
NSEGMENTS	<i>none</i>	The line segment number to be edited.	
GTYPE	<i>none</i>	Line segment geometry type.	
X1	<i>none</i>	Starting X(R) coordinate.	
Y1	<i>none</i>	Starting Y(Z) coordinate.	
X2	<i>none</i>	Final X(R) coordinate.	
Y2	<i>none</i>	Final Y(Z) coordinate.	
CURVATURE	<i>none</i>	Curvature of line segment.	
BIAS	<i>none</i>	Bias of line segment.	

Sub-Command	EDIT (<i>continued</i>)	
Parameter	Default	Function
SUBDIVISIONS	<i>none</i>	Line segment subdivision.
TOGGLE	NO	Toggle line segment curvature.

The **EDIT** sub-command contains all the parameters required to define an emitter and line segments. The parameter **NUMBER** specifies which emitter is to be modified, and the parameter **NSEG** specifies which line segment is to be edited. An emitter needs to be created or available before it can be edited. The **EDIT** sub-command is generally used after the **CREATE** sub-command has been used to create a new emitter containing default values.

- Emitter values are edited by running the **EDIT** sub-command and specifying which emitter to edit with the **NUMBER** parameter. This will cause the **EDIT** sub-command and its parameters to be loaded up with the current emitter values. An **EDIT** sub-command prompt will appear and the parameter values can be modified by giving the parameter with a new value or expression.
- Line segment values are edited by running the **EDIT** sub-command, specifying the emitter to edit with the **NUMBER** parameter and specifying the line segment number with the **NSEG** parameter. This will cause the **EDIT** sub-command and its parameters to be loaded up with the specified emitter values and line segment numbers. An **EDIT** sub-command prompt will appear and the parameter values can be modified by giving the parameter with a new value or expression.
- The **TOGGLE** parameter can be used to toggle the curvature of a specified line segment. This can be useful if the incorrect curvature has been set for the line segment and it needs to be quickly inverted.

Once the appropriate emitter data and line segment data has been defined, a blank line given to the command prompt will exit the **EDIT** sub-command.

The **LIST** Sub-command

Menu Route:

MODEL↓
Emitter data → List emit data

Command Line Parameters:

Command	LIST
No Parameters	

The **LIST** command lists all the data for the emitters and line segments that have been defined.

The **QUIT** Sub-command

Menu Route:

MODEL↓
Emit → Return

Command Line Parameters:

Command	QUIT
No Parameters	

There are no parameters with the **QUIT** command. It allows the user to exit the **EMIT** sub-command and return to the top-level command input.

The **READ** Sub-command

Menu Route:

MODEL↓
Emitter data → Read an emit file

Command Line Parameters:

Sub-Command	READ	
Parameter	Default	Function
FILE	<i>none</i>	The file name of the emitter file to be read.
ADD	NO	Add the file to the data already present.

The **READ** sub-command reads in an emitter data file. The parameter **FILE** takes the file name of the required emitter file. The four letter extension *emit* is added by default to the file name if no extension is specified.

The **ADD** parameter controls whether the file data is to be appended to or overwrite any existing data. Setting **ADD=YES** will append the file data and the default setting (**ADD=NO**) will replace the existing emitter data with the new data.

Once the data is read in, it will be available to be listed and modified as required.

The **STORE** Sub-command

Menu Route:

MODEL↓
Emitter data → Store an emit file

Command Line Parameters:

Sub-Command	STORE	
Parameter	Default	Function
FILE	<i>none</i>	The name of the emitter file to be created/overwritten.

The **STORE** sub-command creates (or overwrites) a emitter data file. The parameter **FILE** takes the file name of the required emitter file. The four letter extension *emit* is added by default to the file name if no extension is specified. The emitter file can be used as part of a Space Charge analysis.

The **ZOOM** Sub-command

Menu Route:

MODEL↓
Emit → Zoom display

Command Line Parameters:

Command	ZOOM
No Parameters	

The **ZOOM** sub-command is identical to the top-level command of the same name (see [page 3-212](#)). It re-displays the picture of the geometry, zooming in or out. Line segments are displayed as white lines containing white crosses indicating the number of subdivisions, subdivision bias and red arrows indicating the direction of beam emission.

The **END** Command

Menu Route:

FILE↓
End OPERA-2d/PP

*Command Line
Parameters:*

Command	END
No Parameters	

The **END** command stops the pre and post processor. All data files are closed.

The **ERASE** Command

Menu Route: MODEL↓
Erase regions

Command Line Parameters:

Command	ERASE		
Parameter	Default	Function	
REG1	<i>none</i>	Regions to be erased.	
		number	Regions in range REG1 to REG2 .
		group	Regions in group named.
REG2	<i>none</i>	Last region to be erased. * for highest region number.	
ERASE	NO	Clear graphics window and re-display switch:	
		NO	Regions selectively erased from display.
		YES	Graphics window cleared and picture re-displayed.

The **ERASE** command erases the regions specified by number or group name using parameters **REG1** and **REG2** (see [page 3-122](#)). The remaining regions are renumbered to form a contiguous set starting at 1.

If neither **REG1** nor **REG2** are specified the regions to be erased can be picked from the display. In keyboard mode, the cursor commands are **<space>** to select a region, **H** for help and **Q** to quit (see [page 3-152](#)).

The regions erased are selectively erased from the graphics window (**-ERASE**) or the complete picture can be erased and the remaining regions re-displayed (**+ERASE**).

Before each **ERASE** command, the current set of regions is saved in memory so that it can be restored using the **UNDO** command ([page 3-201](#)) if regions are erased in error.

The **EXTERNALCIRCUIT** Command

Menu Route: MODEL↓
Circuits

Command line Parameters:

Commands	EXTERNALCIRCUIT
No Parameters	

External Circuits

External circuit configurations can be entered for Steady State analysis (AC), Transient Analysis (TR) and for the Rotating Machine solver (RM). Circuit configurations are ignored in all other solvers.

External circuit data consists of power supply characteristics (voltage, resistance, inductance and capacitance), the length of the circuit (for XY models only), and OPERA conductors. The OPERA conductors are defined by having a conductor number (also referred to as the circuit label or the region parameter N). More than one region can have the same conductor number.

In AC analysis, the circuits are driven by the frequency defined in the analysis data (the **SOLVE** command), which is a cosine source, and a phase which is defined in the circuit data. In TR and RM analysis all the drive data is defined in the analysis data.

There are two types of external circuit, **FILAMENTARY** and **EDDY** current circuits and up to 100 circuits can be defined.

Filamentary circuits

- The conductors consist of many turns of stranded wire.
- The current density is uniform over the conductor (the conductivity of the region must be zero), i.e. the skin effect is neglected.
- The properties are defined as the number of turns and the resistance per unit length of the wire.
- The external impedance can be zero, if the conductor resistance is non-zero.

Eddy Current Circuits

- Conductors are ‘massive’ and each turn is modelled separately.
- Eddy currents can flow in the conductors.
- The property of the conductor is defined by the conductivity of the region.
- The external resistance must be non zero.

Two external circuits can share common components (resistances, inductances or OPERA conductor). Common resistances and inductances are entered with the **EDIT** sub-command once the circuits have been defined. A conductor which exists in more than one circuit is defined in each circuit.

Full details of the way such data is used are given in the User Guide.

The following list of sub-commands are available:

EXTERNAL CIRCUITS Sub-commands		
Sub-command	Function	
CHECK	Check circuit data agrees with the model data	page 3-95
DEFINE	Define circuit data	page 3-88
DELETE	Delete circuit data	page 3-92
EDIT	Edit circuit data	page 3-90
IMPORT	Import circuit data from a SPICE type file	page 3-93
LIST	List circuit data	page 3-92
LOAD	Load circuit data	page 3-92
QUIT	Quit the EXTERNAL CIRCUIT command	page 3-96
STORE	Store circuit data	page 3-93

DEFINE Sub-command

Command line Parameters:

Sub-Command	DEFINE	
Parameter	Default	Function
SYMMETRY	1	The number of similar conductors in series, implied by boundary conditions.
LENGTH	<i>none</i>	Length of the circuit (XY problems only).

Sub-Command	DEFINE (continued)	
Parameter	Default	Function
VOLTAGE	<i>none</i>	Sum of the voltage sources in the circuit (Peak value for AC problems).
PHASE	0	Phase of supply voltage (AC problems only).
RESISTANCE	0	Resistance which is in series and is only in this circuit
INDUCTANCE	0	Inductance which is in series and is only in this circuit
CAPACITANCE	0	Capacitance which is in series and is only in this circuit (0 is equivalent to no capacitors)
INITIALCAPVOLT	0	Initial voltage across a capacitor (transient solvers only)
TYPE	FILAMENTARY	Circuit type: EDDY Current or FILAMENTARY

The **DEFINE** sub-command allows input in two stages. First the global circuit parameters are defined, where the power supply characteristics are entered as **VOLTAGE**, **PHASE**, **RESISTANCE** and **INDUCTANCE**. To calculate the inductance of the finite element model, the program also needs to know (for XY problems only) the **LENGTH** of the model. To correctly calculate the current in models which have symmetry implied by boundary conditions **SYMMETRY** is defined. If half the circuit is included in the finite element model, **SYMMETRY=2**, if a quarter, **SYMMETRY=4**, etc., providing that the coils are wound in series.

Secondly the properties of each conductor are entered.

FUNCTION	<i>none</i>	Function to take place on conductor data: ADD , BACKTRACK (over write previous conductor) or QUIT
CONDUCTOR	<i>none</i>	Conductor Number
SENSE	<i>none</i>	Direction of the current in the conductor, GO or RETURN
TURNS	<i>none</i>	Number of turns (Filamentary circuits only must be 1 for Eddy current circuits)
UNITRESISTANCE	<i>none</i>	Resistance per unit length of the wire in ohms. (Filamentary circuits only)

The program prompts for the conductor numbers (region parameter **N**) of the conductors which form the circuit along with:

- A function, whether to **ADD** a new conductor, **BACKTRACK** to a previous conductor or **QUIT**.
- the conductor number (circuit label, or region parameter **N**)
- the sense of a conductor, whether it is **GO** or **RETURN**
- the number of turns (1 for an eddy current conductor)
- the resistance/unit length of the wire (filamentary circuits only).

Once defined, all the data can be modified using the **EDIT** command.

EDIT Sub-command

Command line Parameters:

Sub-Command	EDIT	
Parameter	Default	Function
C1	<i>none</i>	Circuit number to be edited
TYPE	<i>none</i>	Circuit type, EDDY current or FILAMENTARY
SYMMETRY	<i>none</i>	The number of similar conductors in series, implied by boundary conditions.
LENGTH	<i>none</i>	Length of the magnet (XY-symmetry only).
VOLTAGE	<i>none</i>	Sum of the voltage sources in the circuit (Peak value for AC analysis).
PHASE	<i>none</i>	Phase of the supply voltage, AC problems only.
RESISTANCE	<i>none</i>	Resistance which is in series and is only in this circuit
INDUCTANCE	<i>none</i>	Inductance which is in series and is only in this circuit
CAPACITANCE	0	Capacitance which is in series and is only in this circuit (0 is equivalent to no capacitors)
INITIALCAPVOLT	0	Initial voltage across a capacitor (transient analysis only)

Sub-Command	EDIT (<i>continued</i>)	
Parameter	Default	Function
FUNCTION	<i>none</i>	FUNcTION to take place on conductor data, UPDATE , DELETE , ADD , INSERT or LIST
ITEM	<i>none</i>	The item number to edit, add after or insert before
CONDUCTOR	<i>none</i>	Conductor Number
SENSE	<i>none</i>	Direction of the current in the conductor, GO or RETURN
URNS	<i>none</i>	Number of turns (Filamentary circuits only must be 1 for Eddy current circuits)
UNITRESIST	<i>none</i>	Resistance per unit length of the wire (Filamentary circuits only)
RTWO	<i>none</i>	A second circuit which shares a common resistance (CORESISTANCE)
CORESISTANCE	<i>none</i>	The common resistance shared with RTWO
LTWO	<i>none</i>	A second circuit which shares a common inductance (COINDUCTANCE)
COINDUCTANCE	<i>none</i>	The common inductance shared with LTWO
CTWO	<i>none</i>	A second circuit which shares a common capacitance
COCAPACITANCE	<i>none</i>	The common capacitance shared with CTWO
COINITIALCAPVOLT	<i>none</i>	The common initial voltage of the capacitor shared with CTWO
SWITCH	<i>none</i>	Special option for transient solvers: MONITOR : Print the current at each time step MAX : Stop the analysis when the absolute values of the current in the circuit is a maximum

The global parameters (circuit type, symmetry etc.) can be changed to new values using the **EDIT** command. The conductor properties are modified, by specifying the **ITEM** number of the conductor. The item numbers for the conductors are found by using the **LIST FUNCTION** of the **EDIT** command or the **LIST** sub-command.

Different circuits may 'share' common components; conductors from the OPERA model, or resistances and inductances. The common conductors which are shared are entered in both circuits, whereas the common resistors and inductors which are shared are added into one or other of the circuits using the edit command. Both global circuit parameter for both circuits must have been defined before shared components can be added.

DELETE Sub-command

Command line Parameters:

Sub-Command	DELETE	
Parameter	Default	Function
C1	<i>none</i>	Number of first circuit to be deleted
C2	<i>none</i>	Number of last circuit to be deleted

This command removes the circuit data from **C1** to **C2**. The remaining circuits are renumbered to replace any which have been deleted. To delete all the circuits in a model, **C1** is specified as 1 and **C2** as *.

LIST Sub-command

Command line Parameters:

Sub-Command	LIST	
Parameter	Default	Function
C1	<i>none</i>	Number of first circuit to be listed

This command describes all the circuit data. If a solution is present the current in the circuit will also be displayed. To list all the circuits in a model, **C1** is specified as 1 and **C2** as *.

LOAD Sub-command

Command line Parameters:

Sub-Command	LOAD	
Parameter	Default	Function
FILE	<i>none</i>	Filename for the circuit data (The default file extension is '.lop')

The external circuits definitions can be read into the OPERA datafile. If External Circuits are already in the datafile, the new circuits will be added at the end of the list.

STORE Sub-command

Sub-Command	STORE	
Parameter	Default	Function
FILE	<i>none</i>	Filename for the circuit data (The default file extension is <i>'.lop'</i>)

The external circuits definitions can be saved into a file, this allows a circuit definition to be exchanged between models.

IMPORT Sub-command

Sub-Command	IMPORT	
Parameter	Default	Function
FILE	<i>none</i>	Filename for the SPICE type data (The default file extension is <i>'.cir'</i>)

This command allows SPICE type data to be imported into OPERA, the data is then converted by the program to circuit data. This feature is useful if using inter-connecting circuits where the definition of the circuits is most easily achieved by analysing the circuit as a 'net list'.

The SPICE type file is an ASCII file which uses the comment lines in the files to describe the OPERA circuit details which are not available in the conventional SPICE format (comments start with *'**)

A SPICE type file contains a label for each component, the 2 nodes between which it is located and a value for the component i.e.

```
R4 1 2 10k
```

describes resistor 4, located between nodes 1 and 2, with a value of 10 KOhms.

There are four SPICE type components which are interpreted in OPERA.

SPICE type component	Description	Meaning of 'value'
O	OPERA conductor	Conductor number (other properties are taken from the comment lines *TURNS and *RESIST)
VIN	Voltage Source	Voltage (volts)
R	Resistor	Resistance (ohms)
L	Inductor	Inductance (Henries)

Comment	Default	Description
*TYPE	FILAMENTARY	Circuit type, EDDY or FILAMENTARY
*SYMMETRY	1	Symmetry of the circuits
*LENGTH	<i>none</i>	Length of the Circuits
*PHASE	0	Phase of the all Voltage sources
*SENSE	<i>none</i>	Defines the sense of one conductor in a circuit
*TURNS	1	Turns per conductor (set to 1 for EDDY)
*RESIST	0	Resistance per unit length for the conductor (set to 0 for EDDY)

Only one comment can be placed on any one line. The comments must be placed in the file before the lines which they affect.

The first line in the file is the title (and is ignored) and the file must finish with a the line

.end

A SPICE type file can only contain **FILAMENTARY** or **EDDY** current circuits and the **LENGTH** and **SYMMETRY** must be the same through out the file. One conductor in each circuit must have its **SENSE** defined as either **GO** or **RETURN**.

In a SPICE type file, the component values use the code letters defined below to describe the exponent of the values.

Exponent Symbol	Value	Description
F	1e-15	femto
P	1e-12	pico

Exponent Symbol	Value	Description
N	1e-9	nano
U	1e-6	micro
M	1e-3	milli
K	1e3	kilo
MEG	1e6	mega
G	1e9	giga
T	1e12	tera

An example SPICE type circuit is given below, it describes a simple circuit of two conductors (number 1 and 8, where 8 is the **GO** conductor) with an external resistance of 2 ohms and Voltage of 180millivolts.

```
* Simple 2 conductor circuit
*LENGTH          4
*SYMMETRY        1
*PHASE           0
*TURNS           1
*RESIST          0
*TYPE            EDDY
Vin1              1  2  180M
R1                4  1  2
O1                2  3  1
*SENSE           GO
O2                3  4  8
.end
```

This produces an OPERA circuit as follows (displayed using the **LIST** sub-command).

```
Circuit1: symmetry=1, length4.0 (CM), Eddy Current
Power supply=0.18 Volts, Phase angle=0.0 Degrees
Total resistance=2 Ohms, Inductance=0.0 Henries
Conductors:
item, conductor, sense,      turns, resistance/CM
  1      8      Go      1      0.0
  2      1      Return    1      0.0
```

CHECK Sub-command

This command has no parameters and checks all the External Circuits for errors. This sub-command is also called during mesh generation, so all errors must be fixed before a model can be analysed.

QUIT Sub-command

This command has no parameters and causes the program to leave the **EXTERNALCIRCUIT** command.

The **EXTRA** Command

Menu Route: MODEL↓
Extra options

Command Line Parameters:

Command	EXTRA
No Parameters	

The **EXTRA** command imposes additional conditions on data for analysis. These data are of two types:

- Non-uniform electromagnetic material properties and boundary conditions which cannot be supported by the standard region data.
- Material properties and external load tables and boundary conditions for the OPERA-2d stress and thermal analysis programs.

There are no parameters with the **EXTRA** command; it introduces a set of sub-commands. The **EXTRA** sub-commands are as follows:

EXTRA Sub-commands	
CURSOR	To switch to cursor selection of nodes, faces, elements and regions.
DELETE	To delete EXTRA conditions.
ELEMENT	To identify an element and supply one of its electromagnetic properties.
EXPORT	To save a solution for part of the model.
FACE	To identify a region face and supply its boundary condition.
IMPORT	To load a partial solution file.
LIST	To list EXTRA conditions and tables.
LOAD	To load a valid table creating a new system variable
MATERIAL	To define or edit material property tables for stress and thermal analysis.
NODE	To identify a node and supply its boundary condition.
PROCESS	To process a partial solution file.
QUIT	Leave the EXTRA command.
REGION	To identify a region or region group and supply one of its electromagnetic properties.

EXTRA Sub-commands <i>(continued)</i>	
SAVE	To save a table created using the TABLE sub-command into a file
TABLE	To set up tables of nodal or element values. This is primarily intended for storing body forces for stress analysis and power densities for thermal analysis.
ZOOM	Re-display the picture, zooming in or out.

Boundary Conditions and Electromagnetic Material Properties

EXTRA conditions can be applied to nodes, faces, elements or regions.

On nodes, either individually or all nodes on a face, the boundary conditions **V** (fixed potential) or **DV** (fixed normal derivative of potential) can be set. In addition, boundary conditions for stress analysis and thermal analysis can be given. For stress analysis, this allows the imposition of a **SKEW** angle, a **LOAD** or a **CONSTRAINT**. For thermal analysis, this allows the imposition of constant **TEMPERATURE** or **THERMAL** cooling conditions.

For elements, either individually or all elements of a region, current or charge **DENSITY**, **PERMEABILITY** or **PERMITTIVITY**, **PHASE** angle and **CONDUCTIVITY** can be set.

In each case (nodes or elements) the values given can be expressed in terms of the coordinates (system variables **X**, **Y** or **R**, **Z**). Such expressions are evaluated for each node or element centroid, allowing spatial variations of properties. The only other system variable which can be used in these expressions is **PI**.

N.B. User variables should not be used in **EXTRA** expressions.

The **EXTRA** sub-commands should be used to define the extra conditions after the **MESH** has been generated successfully. The command overwrites the appropriate node and element values in the mesh. If for any reason the **MESH** command has to be used after **EXTRA** conditions have been imposed, the **EXTRA** data is not lost, but is re-imposed automatically. In this case **EXTRA** data should be checked, since node and element numbers might be changed by re-meshing.

Tables

Tables are the means of transferring extra nodal or element based data between OPERA-2d analysis programs and the pre and post processor. The tables are pri-

marily intended for storing the element values of Lorentz force density or power density as input to the stress and thermal analysis programs. Tables can also be used while post processing to create additional system variables from the values of expressions at every node. The system variables thus created can be used in any post processing command. If names are given for system variables to hold the spatial derivatives then these can also be used in post processing. Tables can be stored in and loaded from files using the **SAVE** and **LOAD** sub-commands (page 3-106 and page 3-112) so that values created by one analysis can be loaded into another, provided that the numbers of nodes and elements match.

Mechanical and Thermal Material Properties

The material properties necessary for stress analysis include Young's modulus, Poisson's ratio etc. The only properties required for thermal analysis are the values of the thermal conductivity tensor and the material angle relative to the global x-axis. Tables of properties can be defined, edited, stored and loaded from file, and associated with the region material numbers.

Command Line Parameters:

The **CURSOR** Sub-command

Sub-Command	CURSOR
No Parameters	

The **CURSOR** sub-command switches to cursor mode for identification of nodes, faces, elements and regions (see page 3-152). The cursor sub-commands correspond to the keyboard sub-commands, except that, after identification of the node, face, element or region the program prompts the user to supply the required condition.

EXTRA Command Cursor Hits	
Cursor hit	Function
E	Identify the element containing the cursor for an EXTRA condition.
F	Identify the nearest face of the region containing the cursor for an EXTRA condition.
H	Display help message explaining all the cursor options.
N	Identify the nearest node to the cursor for an EXTRA condition.
L	List all EXTRA conditions.
Q	Leave cursor mode.

EXTRA Command Cursor Hits (continued)	
Cursor hit	Function
R	Identify the region containing the cursor for an EXTRA condition.
Z	Re-draw the picture, zooming in or out (see page 3-212).

The DELETE Sub-command

Menu Routes:

MODEL↓

Extra options → Extra conditions → Delete conditions

Command Line Parameters:

Sub-Command	DELETE	
Parameter	Default	Function
TYPE	CONDITION	Type of EXTRA data: COND EXTRA conditions.
C1	<i>none</i>	First data item to be deleted.
C2	<i>none</i>	Last data item to be deleted. * for highest number.

The **DELETE** command removes some or all of the **EXTRA** conditions specified by **TYPE**, **C1** and **C2**. The remaining data are renumbered to form a contiguous set starting at 1.

The ELEMENT Sub-command

Menu Route:

MODEL↓

Extra

options → Extra

conditions → Electromagnetic

materials → Pick an element

Command Line Parameters:

Sub-Command	ELEMENT		
Parameter	Default	Function	
XE	<i>none</i>	X-coordinate of a point inside the element.	
YE	<i>none</i>	Y-coordinate of a point inside the element.	
CONDITION	PHASE	Name of the material property to be set.	
		DENSITY	Current or charge density.
		PERM	Permeability or permittivity.
		PHASE	Angle in degrees.
	CONDUCTIVITY	Conductivity.	
FUNCTION	0	Function or value for property.	

The **ELEMENT** command identifies an element for an **EXTRA** condition. The element is identified by the coordinates of a point within it (**XE**, **YE**).

If the point is found within an element, the **FUNCTION** for the chosen **CONDITION** is evaluated with the coordinates of the centroid of the element assigned to system variables **X** and **Y** or **R** and **Z**. The only other system variable which can be used in **FUNCTION** expressions is **PI**.

Full details of the material properties are given under the **DRAW** Command, Region property parameters, [page 3-47](#).

The EXPORT Sub-command

Menu Route:

MODEL ↓

Extra options → Solution tables → Export a Table

Command Line Parameters:

Sub-Command	EXPORT		
Parameter	Default	Function	
FILEOUT		Name of Table File	
REG1	<i>none</i>	Regions in table.	
		number	Regions in range REG1 to REG2 .
		group	Regions in group named.
REG2	*	Last region. * for highest region number.	

Sub-Command	EXPORT (<i>continued</i>)		
Parameter	Default	Function	
NODAL	YES	Options:	
		YES	Table is created from nodal values
		NO	Table is created from element values.
TRANSX	0	X translation stored in the export file	
TRANSY	0	Y translation stored in the export file	
ROTATION	0	Rotation (in degrees) stored in the export file	
SOL1		Name of a solution table to be exported	
SOL2		Name of a solution table to be exported	
SOL3		Name of a solution table to be exported	
SOL4		Name of a solution table to be exported	
SOL5		Name of a solution table to be exported	
SOL6		Name of a solution table to be exported	
SOL7		Name of a solution table to be exported	
SOL8		Name of a solution table to be exported	
SOL8		Name of a solution table to be exported	
SOL9		Name of a solution table to be exported	
SOL10		Name of a solution table to be exported	
NODEDUMP	NO	Options:	
		YES	Dump the node (or element) numbers in the solution along with the X and Y coordinates
		NO	Do nothing
NELD	NO	Options:	
		YES	Dump the node element database at the end of the table file
		NO	DO nothing.

The **EXPORT** command stores **TABLE** values in a file. Up to 10 tables can be selected. An exported table file can contain either element or nodal based tables; the values are stored with the node or element centroid coordinates. An exported table can only be read back in to the program using the **IMPORT** sub-command.

The X, Y and rotation information stored in the command are used by the **PROCESS** sub-command. The original node positions are stored in the table file.

The node (or element) number can be written into the file if the **NODEDUMP** option is selected. The **NELD** option outputs the node numbers of each element along with the material number of each element.

The **FACE** Sub-command

Menu Routes:

```

MODEL↓
  Extra
  options → Extra
            conditions → Electromagnetic
                          boundaries → Pick a side
    
```

and

```

MODEL↓
  Extra
  options → Extra
            conditions → Stress/thermal
                          boundaries → Pick a side
    
```

Command Line Parameters:

Sub-Command	FACE	
Parameter	Default	Function
REGION	<i>none</i>	Region number.
FACE	<i>none</i>	Face number within the region.
CONDITION	V	Name of the boundary condition to be set.
		CONSTRAINT Constrain nodal coordinate (stress analysis).
		DV Fixed normal derivative of potential.
		LOAD External pressure on face (stress analysis).
		SKEW Skewed constraint (stress analysis).
		TEMPERATURE Fixed temperature (thermal analysis).
		THERMAL Heat transfer condition (thermal analysis).
		V Fixed value of potential.

Sub-Command	FACE	
Parameter	Default	Function
FUN1	none	Function or value for scalar, X component or thermal condition.
FUN2	none	Function or value for Y component or thermal condition.
FUN3	none	Function or value for thermal condition.

The **FACE** command identifies a region face for an **EXTRA** condition. The face is identified by the **REGION** number and the **FACE** number within the region.

If the face exists, the function **FUN1** and, if necessary, **FUN2** and **FUN3**, for the chosen **CONDITION** is evaluated for each node of the face with the coordinates of the node assigned to system variables **X** and **Y** or **R** and **Z**. The only other system variable which can be used in **FUN_n** expressions is **PI**.

The stress analysis conditions, **CONSTRAINT** and **LOAD** take a maximum of two functions. One of the functions can be omitted for constrained nodes to indicate a constraint in one direction only.

- In XY symmetry, a load on a face is used as a surface traction, i.e. the total force applied per unit length is the load multiplied by the length of the face.
- In axisymmetry, a load on a face is a pressure, and is multiplied by the surface area to give the force.

The **THERMAL** heat transfer condition takes three functions, which are used for q , α and T_0 in the boundary condition

$$k \frac{\partial T}{\partial n} = q + \alpha(T - T_0) \quad (3.6)$$

where k is the material thermal conductivity (see User Guide).

The **IMPORT** Sub-command

Menu Route:

MODEL ↓

Extra options → Solution tables → Import a Table

Command Line Parameters:

Sub-Command	IMPORT	
Parameter	Default	Function
FILEIN		Name of Table File
OVERWRITE	YES	Options:
		YES
		Imported values replace previous values
		NO
		Imported values are added to the previous values.

The **IMPORT** sub-command reads **TABLE** values in a file. If the **TABLE** file contains tables which do not already exist in the program, new tables are created. If the **TABLE** file contains nodal values the table created will be a node table, if the **TABLE** file contains element based values, the table created will be an element based table.

In order to import values, the program compares the coordinates in the **TABLE** file with the node (or element centroid) positions. It is an error if the position in the file and the program mesh do not match. A user can import a table from one model to another by using the **PROCESS** sub-command.

The **LIST** Sub-command

Menu Routes:

MODEL↓
Extra options → Extra conditions → List conditions

and

MODEL↓
Extra options → Solution tables → List tables

Command Line Parameters:

Sub-Command	LIST	
Parameter	Default	Function
TYPE	CONDITIONS	Type of EXTRA data:
		CONDITIONS EXTRA conditions.
		TABLE Tables.

The **LIST** sub-command prints all existing **EXTRA** conditions or tables to the screen.

The **LOAD** Sub-command

Menu Routes:

MODEL↓

Extra options → Solution tables → Load a table

Command Line Parameters:

Sub-Command	LOAD	
Parameter	Default	Function
FILE	<i>none</i>	Name of the table file

The **LOAD** sub-command loads a file containing a valid table into the pre and post processor. Once loaded the data contained in the table may be displayed using the system variable associated with the table.

The **MATERIAL** Sub-command

Menu Routes:

MODEL↓

Extra options → Stress/thermal materials

Command Line Parameters:

Sub-Command	MATERIAL		
Parameter	Default	Function	
OPTION	DEFINE	Option:	
		DEFINE	Define a new material property table.
		EDIT	Edit an existing material property table.
		LIST	List the properties of a material.
		LOAD	Load a material property table from a file.
	STOR	Store a material property table in a file.	

Sub-Command	MATERIAL (<i>continued</i>)	
Parameter	Default	Function
NUMBER	3	Material number: 1 or 3 to 102 .
FILE	<i>none</i>	File name for LOAD or STORE .

The **MATERIAL** sub-command handles the material properties necessary for the OPERA-2d stress analysis program.

It allows material property tables to be associated with the material numbers (1 and 3 to 102) in the region data. Material property tables cannot be associated with material 0 which is used for air.

The **MATERIAL** sub-command is controlled by the **OPTION** parameter which can be set to **DEFINE**, **EDIT**, **LIST**, **LOAD** or **STORE**.

When a new property table is defined, the program prompts for a material name and type. The name exists only for the user's benefit. The following material types are available:

- **ISOTROPIC** (stress analysis)
- **ORTHOTROPIC** (stress analysis)
- **TRANSVERSELY** isotropic (stress analysis)
- **AXISYMMETRIC**-stratified (stress analysis)
- **THERMAL** (thermal analysis)
- **NULL** (parts to be omitted from the analysis)

When defining or editing, another set of sub-commands can be used to give values to the required properties. These are:

Stress Analysis Material Properties	
ANGLE	The angle between principal and global axes.
CSTRESS	The maximum allowable compressive stress.
DENSITY	The mass density.
POISSON	The Poisson's ratio.
SHEAR	The shear modulus.
STSRESS	The maximum allowable shear stress.
THERMAL	The thermal expansion integrals.
TSTRESS	The maximum allowable tensile stress.
YOUNG	The Young's modulus.

Thermal Analysis Material Properties	
ANGLE	The angle between principal and global axes.
DENSITY	The mass density.
SPECIFIC	The specific heat capacity.
THERMAL	The thermal conductivity tensor.

In the case of some of the properties, the number of values required depends on the type of material being defined. The keywords **CLEAR**, **HELP**, **LIST**, **QUIT** and **TYPE** can be used while defining properties to remove all properties, request help, list properties, leave the command or change the material type.

- Example:

```

OPERA-2d > extra
Supply extra conditions on analysis
OP2-EXTRA > mate defi 3
Define/Edit material property tables (Stress Analysis)
Give material name and type.
Types are (ISOTropic, ORTHotropic, TRANsversely isotropic,
          AXISymmetric-stratified or NULL)
OP2-MATE > steel iso
Give property name and values, or type HELP, LIST
or QUIT
OP2-MATE > pois help
Poisson's Ratio: Nu
OP2-MATE > pois 1.e10
...

```

The User Guide gives information on which properties must be supplied for each type of material.

The **LOAD** and **STORE** options allow material property tables to be stored in files independent of other OPERA-2d data. They can thus be transferred from one model to another. If no file name extension is provided, the extension *mate* is used.

The **NODE** Sub-command

Menu Routes:

```

MODEL↓
  Extra
  options → Extra
            conditions → Electromagnetic
                        boundaries → Pick a node

```

```

MODEL↓
  Extra
  options → Extra
            conditions → Stress/thermal
                       boundaries → Pick a node

```

Command Line Parameters:

Sub-Command	NODE	
Parameter	Default	Function
XN	<i>none</i>	Approximate x-coordinate of node.
YN	<i>none</i>	Approximate y-coordinate of node.
CONDITION	V	Name of the boundary condition to be set.
		CONSTRAIN Constrain nodal coordinate (stress analysis).
		DV Fixed normal derivative of potential.
		LOAD External force on node (stress analysis).
		SKEW Skewed constraint (stress analysis).
		TEMPERATURE Fixed temperature (thermal analysis).
		V Fixed value of potential.
FUN1	<i>none</i>	Function or value for scalar or X component of condition.
FUN2	<i>none</i>	Function or value for Y component of condition.

The **NODE** command identifies a node for an **EXTRA** condition. The node is identified by the coordinates **XN**, **YN**. The program selects the node closest to that point.

The function **FUN1** and, if necessary, **FUN2**, for the chosen **CONDITION** is evaluated for the node with the coordinates of the node assigned to system variables **X** and **Y** or **R** and **Z**. The only other system variable which can be used in **FUN n** expressions is **PI**.

The stress analysis conditions, **CONSTRAIN** and **LOAD** take a maximum of two functions. One of the functions can be omitted for constrained nodes to indicate a constraint in one direction only.

- In XY symmetry, a load on a node is used as a force per unit length.
- In axisymmetry, a load on a node is multiplied by $2\pi r$ to give the force.

The **PROCESS** Sub-command

Menu Route:

MODEL ↓

Extra options → Solution tables → Process a Table

Command Line Parameters:

Sub-Command	PROCESS	
Parameter	Default	Function
FILEIN		Name of input Table File
FILEOUT		Name of output Table File
TRANSX	0	X translation stored in the export file
TRANSY	0	Y translation stored in the export file
ROTATION	0	Rotation (in degrees) stored in the export file
SOL1		Name of a solution table to be processed
SOL2		Name of a solution table to be processed
SOL3		Name of a solution table to be processed
SOL4		Name of a solution table to be processed
SOL5		Name of a solution table to be processed
SOL6		Name of a solution table to be processed
SOL7		Name of a solution table to be processed
SOL8		Name of a solution table to be processed
SOL8		Name of a solution table to be processed
SOL9		Name of a solution table to be processed
SOL10		Name of a solution table to be processed

The **PROCESS** sub-command calculates the **TABLE** values at positions determined by a **TABLE** file, and writes out the value in a second **TABLE** file. Up to 10 tables can be selected for processing. The imported table can contain either nodal or element based values. If the imported table contains component values these are discarded.

The X, Y and rotation information supplied by the command is applied to the coordinates in the **TABLE** file *after* the transformation contained in the **TABLE** file has been applied. If Vector components are chosen (for example BX and BY) the components will be rotated by the rotation angles. Both components must be processed for this function to succeed

This sub-command is primarily used to allow table files from one model to be read into a second model, which has a different geometry. The transformation option allows tables at one location in one model to be moved to a different location in another. A typical use is to import the magnetic fields from a magnetic solution into an electrostatic solution (which has a different mesh) for the space charge analysis module, e.g.:

1. Model 1 (electrostatic): **EXPORT** the node positions of the regions of interest in *nodes.table*.
2. Model 2 (magnetostatic): Use the **TABLE** sub-command to create tables of **XBFLUX** and **YBFLUX** from **BX** and **BY** at all nodes of model.
3. Model 2: **PROCESS** the tables **XBFLUX** and **YBFLUX** using *nodes.table* and writing *flux.table*.
4. Model 1: **IMPORT** *flux.table*.

The **QUIT** Sub-command

Command Line Parameters:

Sub-Command	QUIT
No Parameters	

The **QUIT** sub-command terminates the **EXTRA** command and returns to top-level command input.

The **REGION** Sub-command

Menu Route:

```

MODEL↓
  Extra
  options → Extra
            conditions → Electromagnetic
                        materials → Pick a region
                                → Region group

```

Command Line Parameters:

Sub-Command	REGION		
Parameter	Default	Function	
REGION	none	Region number or group name.	
CONDITION	PHASE	Name of the material property to be set.	
		DENSITY	Current or charge density.
		PERM	Permeability or permittivity.
		PHASE	Angle in degrees.
	CONDUCTIVITY	Conductivity or packing factor.	
FUNCTION	0	Function or value for property.	

The **REGION** command identifies a region or a region group for an **EXTRA** condition. A single region can be identified by its **REGION** number. Multiple regions can be specified using a region group name (see [page 3-122](#)).

The **FUNCTION** for the chosen **CONDITION** is evaluated for each element of the region, with the coordinates of the centroid of the element assigned to system variables **X** and **Y** or **R** and **Z**. The only other system variable which can be used in **FUNCTION** expressions is **PI**.

Full details of the material properties are given under the **DRAW** Command, Region property parameters, [page 3-47](#).

The SAVE Sub-command

Menu Routes:

MODEL ↓

Extra options → Solution tables → Save a table

Command Line Parameters:

Sub-Command	SAVE	
Parameter	Default	Function
FILE	none	Name of the table file
TABLE	none	Name or number of the table.

The **SAVE** sub-command stores a table created using the **TABLE** sub-command to a file. This file can be loaded into another model with the same mesh using the **LOAD** subcommand.

The TABLE Sub-command

Menu Route:

MODEL↓

Extra options → Solution tables

Command Line Parameters:

Sub-Command	TABLE		
Parameter	Default	Function	
COMPONENT	POT	Field component expression.	
NUMBER	1	Table number (1 to 5).	
OPTION	NEW	Options:	
		ADD	Add values to an existing table.
		NEW	Create or overwrite table.
		SUBTRACT	Subtract values from and existing table.
NAME	none	Name for table and system variable.	
XDERIVATIVE	none	Name for system variable for X derivative of COMPONENT.	
YDERIVATIVE	none	Name for system variable for Y derivative of COMPONENT.	
TYPE	NODAL	Basis of values stored:	
		AVERAGE	Nodally averaged values.
		ELEMENT	Element centroid values.
		MAXX	X or R component of Maxwell stress integral around each element.
		MAXY	Y or Z component of Maxwell stress integral around each element.
		NODAL	Nodal values.
REG1	1	Regions in table.	
		number	Regions in range REG1 to REG2.
		group	Regions in group named.

Sub-Command	TABLE (continued)		
Parameter	Default	Function	
REG2	*	Last region. * for highest region number.	
MATERIAL	ALL	Material number to be included: 0, 1, 3 to 102 or ALL.	
NOT	ANY	Material number to be excluded: 0, 1, 3 to 102 or ANY. ANY implies none excluded.	
UNIT	none	Expression defining the units of the field component.	
TIME	COMPLEX	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		number	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		COMPLEX	Complex values stored.
		PHASE	Phase angle.
		TAVEVARAGE	Time average.

The **TABLE** command stores values of a field **COMPONENT** in a table. Tables can be accessed by post processing commands by the system variable names assigned with the parameters **NAME**, **XDERIVATIVE** and **YDERIVATIVE**. They can also be read from and written to files using the parameters **SOLn** of the **READ** and **WRITE** commands (see page 3-168 and page 3-70).

The **TABLE** command is primarily intended for creating tables of body force densities for the stress analysis program or heat sources for the thermal analysis program.

- Stress Analysis: Two tables, with names **XLOAD** and **YLOAD**, or **RLOAD** and **ZLOAD**, should be created with **TYPE=ELEMENT** and components representing the force densities required. Alternatively, for body forces in magnetic material, **TYPE=MAXX** and **TYPE=MAXY** can be used to calculate the Maxwell Stress integral around each element (**COMPONENT** is ignored if **TYPE=MAXX** or **TYPE=MAXY**).
- Thermal Analysis: One table called **HEAT** should be created with element values of the power density, and for transient thermal analysis a second table with nodal values of the initial temperature.

Tables can be restricted to certain region numbers (**REG1** and **REG2**) or material numbers (**MATERIAL** and **NOT**) or to a region group (see page 3-121). Zeroes are used for all other regions. Further field components can be added to or sub-

tracted from existing tables, allowing a table to be built from different functions in different materials.

If required a **UNITS** expression can be given, allowing the values to be scaled appropriately whenever the **UNITS** are changed (see [page 3-202](#)). The expression should be formed from the names **CONDU**, **DENSU**, **ENERU**, **FIELU**, **FLUXU**, **FORCU**, **LENGU**, **MASSU**, **POTEU** and **POWEU**. For example, units of a force density per unit length should be given as **FORCU/LENGU**3**.

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified or, if **TIME=COMPLEX** the values at times 0 and 90 are both stored. (See “[Field Evaluation in Steady-state AC](#)” on [page 3-19](#).)

The ZOOM Sub-command

Command Line Parameters:

Sub-Command	ZOOM
No Parameters	

The **ZOOM** sub-command is identical to the top-level command of the same name (see [page 3-212](#)). It re-displays the picture of the geometry, zooming in or out.

The **GAP** Command

Menu Route:

MODEL↓
 Draw regions → Rotating m/c Air Gap

Command Line Parameters:

Command	GAP		
Parameter	Default	Function	
INCLUDE	NO	Action:	
		YES	Includes an air gap for a rotating machine within the model.
		NO	There is no air gap in the model
RADIUS	0	The radius of the air gap	
SYMMETRY	0	The rotational symmetry of the rotating machine.	

This command creates the air gap region within a rotating machine model. The **GAP** command allows the automatic generation of a finite element mesh to fill the air gap between the rotor and stator parts of a model.

Region Parameters

The **GAP** region is either present or not within a model. It can be included within the model by setting the **INCLUDE** parameter to **YES**, and removed by setting it to **NO**.

The **RADIUS** parameter specifies a mid gap radius. This radius must not pass through any regions, i.e. it must be outside the maximum radius of any point on the rotor, and inside the minimum radius of any point on the stator.

The **SYMMETRY** parameter is an integer value specifying both the rotational symmetry of the model and the sign of the periodicity. For example a model representing 1/4 of a full machine, would have a **SYMMETRY** of 4 (if the periodicity was positive) or -4 (if the periodicity was negative). If the whole machine has been modelled, the value must be 1. It is essential that the value of this parameter correctly matches the symmetry of the model.

Use of the Gap Region

The gap region is primarily intended for use with the rotating machine analysis program (RM), but can also be used to greatly simplify the modelling of the air gap of a rotating machine within all the analysis programs, unless mesh adaptation is required.

When generating the mesh with a gap region present, the outside edge of the rotor and the inside edge of the stator are found. These are then connected to form a triangular mesh over the air gap region. If model symmetry exists, the sides of both rotor and stator will be found, and periodicity will be applied to pair the nodes on either side of the section of the motor that has been modelled. No symmetry boundary conditions or periodicity commands need to be applied.

It is recommended that the radius of the outside edge of the rotor and the inside edge of the stator be constant, and that the subdivisions on rotor and stator sides of the gap produce elements of similar size. This will avoid very poorly shaped elements within the gap region.

The gap region can only be displayed when the mesh has been generated.

The **GRAPH** Command

Menu Route:

FILE↓
Graph data in text file

Command Line Parameters:

Command	GRAPH		
Parameter	Default	Function	
FILE	none	File name	
XVALUE	COL1	Expression for x-axis values.	
XLABEL	X	Label for x-axis.	
XMINIMUM	*	X-axis minimum value. Use * for automatic scaling.	
XMAXIMUM	*	X-axis maximum value. Use * for automatic scaling.	
YVALUE	COL2	Expression for y-axis values.	
YLABEL	Y	Label for y-axis.	
YMINIMUM	*	Y-axis minimum value. Use * for automatic scaling.	
YMAXIMUM	*	Y-axis maximum value. Use * for automatic scaling.	
TITLE	<i>none</i>	Additional title for line.	
STYLE	AUTOMATIC	AUTOMATIC	Program chooses a different style for each graph drawn on the same axes.
		0	Solid line.
		>0	Broken line.
COLOUR	AUTOMATIC	AUTOMATIC	Program chooses a different colour for each graph drawn on the same axes.
		>0	Colour number.

Command	GRAPH (continued)		
Parameter	Default	Function	
SYMBOL	AUTOMATIC	Symbols at data points:	
		AUTOMATIC	Program chooses a different symbol for each graph drawn on the same axes.
		0	No symbol.
		>0	Symbols number.
ERASE	YES	Old graph erasure switch:	
		NO	New line drawn on existing axes.
		YES	Graphics window cleared and new axes drawn.
OPTION	VALUES	Display option:	
		DERIVATIVES	Display the first derivatives of the data
		INTERPOLATIONS	Display cubic-spline interpolations between data values.
		VALUES	Display straight lines between data values.

The **GRAPH** command is a general purpose command for displaying graphs of data read from external files.

The parameters of the **GRAPH** command control the axes limits (**XMIN**, **XMAX**, **YMIN** and **YMAX**), the line **STYLE** and **COLOUR** and whether **SYMBOLS** should be displayed at the data points. For graphs with more than one line, the second and subsequent lines should be drawn with **-ERASE**. A line **TITLE** can be specified for each line. The line titles appear in a legend at the bottom left corner of the graph.

The data format expected by the graphs command is flexible. The file should consist of up to 20 columns of numbers, with up to 1000 values in each column. However, text can be embedded within the numerical data. Each line of the file is parsed into a maximum of 20 fields separated by spaces or commas. The fields are identified as character data or numerical data. If there is numerical data on a line it is kept; character data is ignored. The number of columns is given by the number of numerical data items on the first line which contains any numbers. If subsequent lines contain less numerical values, the number of columns is reduced.

The values for the x and y coordinates of the points plotted on the graphs can be calculated using expressions in terms of corresponding entries from the columns. The simplest use would be to use the first column for the x-coordinates and the second for the y-coordinates (**XVAL=COL1**, **YVAL=COL2**), but much more complicated expressions can be used. For example, the percentage difference between two columns could be calculated and displayed as a graph using **YVAL=100*(COL3-COL2)/COL2**. The **ROW** number can also be used in expressions.

The data evaluated from the expressions **XVAL** and **YVAL** can be displayed in 3 ways:

- **OPTION=VALUE** shows straight lines between the evaluated data points.
- **OPTION=INTERPOLATIONS** uses cubic-spline interpolations between the data points.
- **OPTION=DERIVATIVES** shows the first derivative of the cubic-spline interpolations.

INTERPOLATIONS and **DERIVATIVES** can only be used if the expression for **XVAL** results in a monotonically increasing set of values.

The **GROUP** Command

Menu Route: MODEL↓
Group regions

Command Line Parameters:

Command	GROUP		
Parameter	Default	Function	
NAME	<i>none</i>	Name of group	
ACTION	CREATE	Action:	
		ADD	Add regions to the group.
		CREATE	Create a new empty group.
		CURSOR	Switch to cursor input to select regions to be added to or removed from the group.
		DESTROY	Destroy the group.
		HILIGHT	High-light the group.
		LIST	List the members of the group.
		LOLIGHT	Low-light the group.
		REMOVE	Remove regions from the group.
REG1	1	Regions to be added or removed.	
		number	Regions in range REG1 to REG2 .
		group	Regions in group named.
REG2	*	Last region to be added or removed.	
MATERIAL	ALL	Material number to be added or removed: 0 , 1 , 3 to 102 or ALL .	
NOT	ANY	Material number not to be added or removed: 0 , 1 , 3 to 102 or ANY .	

The **GROUP** command creates and modifies region groups. Region groups are named lists of region numbers. Regions are not automatically assigned to any group, but can be added into or remove from any number of groups using the **GROUP** command.

Region Parameters

In most commands which operate on regions, the regions can be specified by number or by group name, as follows:

- **REG1=number.** If a parameter **REG1** is given a numerical value, it refers to the first region in the range **REG1** to **REG2**. The command then operates on all the regions from **REG1** to **REG2**. **REG2** can be given the value ***** to indicate the highest numbered region. In some commands, parameters **MATERIAL** and **NOT** can also be used to include only one material within the range or exclude one material from the range.

N.B. The text function **%INT()** must be used to evaluate expressions or user variables as values for **REG1** or **REG2** (SEE page 2-19).

- **REG1=group_name.** If a parameter **REG1** is given a character value, it refers to a region group. In this case the command operates on the members of that group. The parameters **REG2**, **MATERIAL** and **NOT** are ignored.

Region groups definitions are read from and stored in data files by the **READ** and **WRITE** commands (see page 3-168 and page 3-211).

GROUP Command Actions

The first parameter of the **GROUP** command is the **NAME**. This can be any character string of up to 8 characters. It must start with a letter. *In menu mode*, the names of existing groups appear in a DialogBox so that one can be selected. *In keyboard mode*, use **NAME=!** to obtain a list of group names.

The **GROUP** command has 8 actions, selected by the **ACTION** parameter:

- **CREATE:** this creates an empty group with a new **NAME**.
- **DESTROY:** this destroys an existing group by removing all its members and deleting its name from the list. It does not **ERASE** the regions.
- **ADD:** this adds the regions specified to the group. The regions can be specified by number or by group name as described in page 3-122 above. It does not matter if some of the regions are already members of the group.
- **REMOVE:** this removes the regions specified from the group. The regions can be specified by number or by group name as described on page 3-122 above. It does not matter if some of the regions are not members of the group.
- **HIGHLIGHT:** this repaints the members of the group in the high-light colour.
- **LOLIGHT:** this repaints the members of the group in their material colours.
- **LIST:** this lists the numbers of the regions which are members of the group.

- **CURSOR**: this switches to cursor input so that regions to be added or removed from the group can be picked from the display (see [page 3-152](#)). The group is high-lighted on entry to cursor mode and low-lighted on exit. There are 4 valid cursor hit characters: **A** for add, **R** for remove, **H** for help and **Q** for quit.

The **HARMONICS** Command

Menu Route:

FIELDS↓
Harmonics

Command Line Parameters:

Command	HARMONICS		
Parameter	Default	Function	
X1	<i>none</i>	X coordinate at start of line.	
Y1	<i>none</i>	Y coordinate at start of line.	
X2	<i>none</i>	X coordinate at end of line.	
Y2	<i>none</i>	Y coordinate at end of line.	
CURVATURE	0	Curvature of line.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		number	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
PERIOD	1	Multiplying factor to create full period (1, 2 or 4).	
NP	16	Number of sampling points.	
TYPE	LEGENDRE	Type of series:	
		FOURIER	Fourier series.
		LEGENDRE	Legendre polynomials.
SERIES	EVEN	Symmetry of Fourier series: EVEN or ODD .	

The **HARMONIC** analysis command can be used to fit either a Fourier or Legendre polynomial series to a function calculated along a line. Calculation of a harmonic series fit to the potential, fields or any function of the fields (e.g. forces) is useful in a wide range of applications. In 2D XY symmetry, a fourier series expansion is particularly useful because this is the analytic series solution to the Laplace

equation. Similarly, the Legendre polynomial series is the analytic solution in axisymmetric coordinates.

The Parameters

The parameters, **X1, Y1, X2, Y2** and **CURVATURE** define the line on which the selected function is calculated, and then fitted to a series (X is equivalent to r and Y to z in axisymmetry).

Other parameters allow the number of sampling points to be set (**NP**) and give the symmetry of fields for **FOURIER** fitting. The **SERIES** parameter is only important if an incomplete period is being used for Fourier analysis. The following examples illustrate two of the possible combinations of **PERIOD** and **SERIES**:

- If the line covers half the period, **PERIOD=2**. If the values in the second half of the period are the same as in the first half then **SERIES** should be set to **EVEN**.
- If the line covers a quarter of the period, **PERIOD=4**. If the values in the second quarter of the period are the negative of the values in the first quarter then **SERIES** should be set to **ODD**.

Expressions for the **COMPONENT** can use as variables any of the system variables given in [page 3-12](#). For Steady-state ac solutions, each system variable in an expression is evaluated at the **TIME** specified.

The Integrals

When the Fourier series are fitted, the coefficients of the Sine (α) and Cosine (β) terms of the series expansion are calculated and printed. These terms are calculated by performing the following integrals:

$$\alpha_n = \sum_{m=1}^p \left[\frac{S_m}{(2\pi n)} \int_0^p \frac{1}{p} \sin(2\pi n s) F(s) ds \right] [n = 0] \quad (3.7)$$

$$\alpha_n = \sum_{m=1}^p \left[\frac{2S_m}{(2\pi n)} \int_0^p \frac{1}{p} \sin(2\pi n s) F(s) ds \right] [n \neq 0] \quad (3.8)$$

$$\beta_n = \sum_{m=1}^p \left[\frac{S_m}{(2\pi n)} \int_0^p \frac{1}{p} \cos(2\pi n s) F(s) ds \right] [n = 0] \quad (3.9)$$

$$\beta_n = \sum_{m=1}^P \left[\frac{2S_m}{(2\pi n)} \int_0^{\frac{1}{2}} \cos(2\pi ns) F(s) ds \right] [n \neq 0] \quad (3.10)$$

where

- n is the order of the harmonic
- s is the normalised measure of distance along the line
- $F(s)$ is the **COMPONENT** that is being fitted
- P is equal to the value of the **PERIOD** parameter
- S_m is a sign function equal to +1 or -1.

Two commonly used cases are:

- **PERIOD=4** with **SERIES=EVEN**: this implies $S_1=1$ $S_2=-1$ $S_3=-1$ and $S_4=1$.
- **PERIOD=4** with **SERIES=ODD**: this implies $S_1=1$ $S_2=1$ $S_3=-1$ and $S_4=-1$.

When Legendre polynomial series are fitted, the Sine and Cosine functions in the above integrals are replaced by $P^n(\sin())$ and $P^n(\cos())$ respectively.

It should be noted that the coefficients of the series are not scaled in any way. The coefficients give the series that represents the function on the line used for calculation. In 2D applications if the command is used to calculate a series expansion of the field in the aperture of a magnet, where the analytic series solution of Laplace's equation has the form

$$A_z = \sum_n a_n r^n \sin(n\Theta) + b_n r^n \cos(n\Theta) \quad (3.11)$$

the $\sin(\alpha_n)$ and $\cos(\beta_n)$ coefficients are related to a_n and b_n as follows

$$\alpha_n = a_n r^n \quad (3.12)$$

$$\beta_n = b_n r^n \quad (3.13)$$

when the fitting is performed on a line which is part of a circle radius r . The same is true when Legendre polynomials are fitted, if the fit is intended to give the terms of a series expansion of the field, then the line used for the calculation must form part of the surface of a sphere and the coefficients must be scaled by the appropriate power of radius. The coefficients that are printed describe the function on the surface of the sphere (this is in fact the usual way of quoting the results

in axisymmetric cases, where the harmonic errors are quoted on a particular diameter sphere).

Practical hints

- When this command is used to determine the coefficients of a series expansion of the field in the aperture of a magnet, it should be noted that it is good practice to use a large radius for the function evaluation. The high order harmonics will only make a small contribution to the field at a small radius and may therefore be invisible (or subject to large errors due to numerical inaccuracies).
- To determine the Legendre polynomial expansion of the field on a sphere, the line used to calculate the fields should start from a point on axis. For example, if the field has symmetry about the plane $Z=0$, the command to fit the Z component of the field, on a sphere radius 25 units, is
**HARM TYPE=LEGENDRE X1=0 Y1=25 X2=25 Y2=0,
CURV=1/25 PERIOD=2 COMP=BZ**
- If the function has a zero order term and the fit is only performed on part of the full period, then the zero order term must be subtracted. Otherwise, the sign reflection used to extend the function to the full period will not produce the expected result.

The **HELP** Command

Menu Route: **HELP**

*Command Line
Parameters:*

Command	HELP
No Parameters	

The **HELP** command gives help to remind users of several of the features of the program. The topics are:

- **System Overview:** this gives an overview of OPERA-2d and introduces the other **HELP** topics.
- **Command Interpreter:** this summarizes the syntax and built-in help features of the command decoder, including details of sub-commands and cursor commands. More information is given in chapter 2.
- **Data Definition Commands:** this summarizes the commands used for entering data.
- **Post processing Commands:** this summarizes the post processing commands and lists the system variables currently available.
- **File Handling Commands:** this summarizes the commands for reading, writing, deleting, renaming and listing files.

The program prompts for the name of a topic: **SYSTEM**, **COMMAND**, **DEFINE**, **POST**, **SYSTEM** or **QUIT**.

When accessed from the GUI, the **HELP** command has different topics explaining the use of the program and the menu interface.

The INTAREA Command

Menu Route: FIELDS↓
 Integrals → Over area

Command Line Parameters:

Command	INTAREA		
Parameter	Default	Function	
REG1	none	Regions to be integrated.	
		<i>number</i>	Regions in range REG1 to REG2.
		<i>group</i>	Regions in group named.
REG2	*	Last region to be integrated. * for highest region number.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		<i>number</i>	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.

INTAREA integrates field quantities over areas. The areas are limited to regions, specified by number or group name using parameters REG1 and REG2 (see page 3-122). The ‘built-in’ integrals vary according to the field type, symmetry and solution potential. One other function, set by COMPONENT is also integrated. Expressions for the COMPONENT can use as variables any of the system variables given on page 3-12.

For Steady-state ac solutions, each system variable in an expression for COMPONENT is evaluated at the TIME specified before COMPONENT is evaluated and integrated. However, the integrands for energy, force and power are computed correctly for all values of TIME (see page 3-19).

The values of the integrals are assigned to the system variables ENERGY, ENERGY1, FLUX, FX, FR, FY, FZ and POWER. The result of integrating

COMPONENT is assigned to **INTEGRAL**. The total area of the regions is assigned to **AREA** and, in axisymmetry, the total volume to **VOLUME**.

When results containing complex permeability from OPERA-2d/AC are processed, an additional hysteresis power-loss term is added into the time-average values of **POWER** and the formula for time-average **ENERGY** is modified to take into account the phase angle between **B** and **H**.

When OPERA-2d is used to solve electrostatic problems, a conversion factor of $\frac{16\pi^2\epsilon_0}{100\mu_0}$ is included in the energy calculation. If a current flow problem is being solved, the electrostatic energy should be multiplied by $\frac{200}{\epsilon_0} = 2.258818148 \times 10^{13}$.

INTAREA Integrals		
1: FIELD=MAGNETIC, SYMMETRY=XY, SOLUTION=V		
Integral	Explanation	System Variable
$\int \phi ds$	integral of potential/unit length	FLUX
$\int \frac{\mathbf{B} \cdot \mathbf{H}}{2} ds$	stored energy/unit length	ENERGY

INTAREA Integrals		
2: FIELD=MAGNETIC, SYMMETRY=XY, SOLUTION=A		
Integral	Explanation	System Variable
$\int A_z ds$	integral of potential/unit length	FLUX
$\int \mathbf{J} ds$	total current	CURRENT
$\int \frac{\mathbf{A} \cdot \mathbf{J}}{2} ds$	stored energy/unit length	ENERGY
$\int \frac{\mathbf{B} \cdot \mathbf{H}}{2} ds$	stored energy/unit length	ENERGY1
$\int \frac{ \mathbf{J} ^2}{\sigma} ds$	power loss/unit length	POWER
$\int \mathbf{J} \times \mathbf{B} ds$	force/unit length	FX, FY

INTAREA Integrals		
3: FIELD=ELECTRIC, SYMMETRY=XY, SOLUTION=V		
Integral	Explanation	System Variable
$\int \phi ds$	integral of potential/unit length	FLUX
$\int \rho ds$	charge/unit length	CHARGE
$\int \frac{\rho \phi}{2} ds$	stored energy/unit length	ENERGY
$\int \frac{\mathbf{D} \cdot \mathbf{E}}{2} ds$	stored energy/unit length	ENERGY1
$\int \rho E ds$	force/unit length	FX, FY

INTAREA Integrals		
4: FIELD=MAGNETIC, SYMMETRY=AXI, SOLUTION=A, RA OR AXI		
Integral	Explanation	System Variable
$2\pi \int r A ds$	integral of potential	FLUX
$\int \mathbf{J} ds$	total current	CURRENT
$\pi \int r \mathbf{A} \cdot \mathbf{J} ds$	stored energy	ENERGY
$\pi \int r \mathbf{B} \cdot \mathbf{H} ds$	stored energy	ENERGY1
$2\pi \int r \frac{ \mathbf{J} ^2}{\sigma} ds$	power loss	POWER
$2\pi \int r \mathbf{J} \times \mathbf{B} ds$	force	FR, FZ

INTAREA Integrals		
5: FIELD=ELECTRIC, SYMMETRY=AXI, SOLUTION=V		
Integral	Explanation	System Variable
$2\pi \int r \phi ds$	integral of potential	FLUX
$2\pi \int r \rho ds$	total charge	CHARGE

INTAREA Integrals <i>(continued)</i>		
5: FIELD=ELECTRIC, SYMMETRY=AXI, SOLUTION=V		
$\pi \int r \rho \phi ds$	stored energy	ENERGY
$\pi \int r \mathbf{D} \cdot \mathbf{E} ds$	stored energy	ENERGY1
$2\pi \int r \rho \mathbf{E} ds$	force	FR, FZ

The INTCIRCLE Command

Menu Route:

FIELDS↓

Integrals → Around circular arc

Command Line Parameters:

Command	INTCIRCLE		
Parameter	Default	Function	
RADIUS	<i>none</i>	Radius of arc.	
P1	<i>none</i>	Azimuthal coordinate at start of arc.	
P2	<i>none</i>	Azimuthal coordinate at end of arc.	
XCENTRE	0	X coordinate at centre of arc.	
YCENTRE	0	Y coordinate at centre of arc.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		<i>number</i>	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
TOLERANCE	0.01	If TOLERANCE <1 it is used as the tolerance in a Runge-Kutta integration. If TOLERANCE ≥1 it is used as the number of points in a Simpson's Rule integration.	
RACTION	0	Radial coordinate of torque action point.	
PACTION	0	Azimuthal coordinate of torque action point.	
ACCUMULATE	ZERO	Accumulator options:	
		ADD	Add integrals to accumulator.
		SUBTRACT	Subtract integrals from accumulator.
		ZERO	Zero accumulator, then add integrals.

Command	INTCIRCLE		
Parameter	Default	Function	
AVERAGE	YES	Nodally averaged fields switch:	
		NO	Use shape function derivative fields.
		YES	Use nodally averaged field.

The **INTCIRCLE** command integrates field quantities along a circular arc, defined by its **RADIUS** and the azimuthal coordinates of the end points, (**P1**, **P2**). Integrals along can also be done using the **INTLINE** command ([page 3-136](#)).

The functions integrated are:

- in xy coordinates: the Maxwell Stress, to give the x and y force per unit length and the anticlockwise torque per unit length with respect to the action point (**REACTION**, **PACTION**).
- in axisymmetry: the Maxwell Stress, to give the z force.
- one other **COMPONENT**.

Expressions for the **COMPONENT** can use as variables any of the system variables given in [page 3-12](#).

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified. However, the integrals for force and torque are computed correctly for all values of **TIME**. (See [page 3-19](#).)

The values of the integrals are assigned to the system variables **INTEGRAL** and in xy coordinates, **FX**, **FY** and **TORQUE**, or in axisymmetry, **FZ**.

The integrals can be evaluated using nodally averaged fields (**+AVERAGE**) or shape function derivative fields (**-AVERAGE**). The **TOLERANCE** can also be set to a value less than 1 to achieve a particular accuracy on the integration using a Runge-Kutta method or to a number greater than 1 to set the number of steps in a Simpson's Rule integration.

For best results, it is recommended that integration paths are completely in air and do not pass through elements which touch the surface of any iron or dielectric. The reason for this is that close to positions where the surface tangent direction changes, the errors will be large. If the line passes through non-air materials, the Maxwell stresses are evaluated as if the line is a vanishingly narrow cut in the material. This should be used with care since singular points at the ends of the cut can lead to a loss of accuracy.

The total force on a body should be computed from the Maxwell Stress integral along series of lines encircling the body in an anticlockwise direction. To enable the forces to be summed easily, an accumulator can be initialized to **ZERO** and the integrals added to or subtracted from it. **INTCIRCLE** and **INTLINE** share the same accumulator. The accumulated values are stored in system variables: **TOTAL_FX**, **TOTAL_FY**, **TOTAL_FZ**, **TOTAL_TORQUE** and **TOTAL_INTEGRAL**.

The **INTLINE** Command

Menu Route:

FIELDS↓
Integrals → Along line

Command Line Parameters:

Command	INTLINE		
Parameter	Default	Function	
X1	<i>none</i>	X coordinate at start of line.	
Y1	<i>none</i>	Y coordinate at start of line.	
X2	<i>none</i>	X coordinate at end of line.	
Y2	<i>none</i>	Y coordinate at end of line.	
CURVATURE	0	Curvature of line.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		<i>number</i>	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
TOLERANCE	0.01	If TOLERANCE <1 it is used as the tolerance in a Runge-Kutta integration. If TOLERANCE ≥1 it is used as the number of points in a Simpson's Rule integration.	
XACTION	0	X coordinate of torque action point.	
YACTION	0	Y coordinate of torque action point.	
ACCUMULATE	ZERO	Accumulator options:	
		ADD	Add integrals to accumulator.
		SUBTRACT	Subtract integrals from accumulator.
		ZERO	Zero accumulator, then add integrals.

Command	INTLINE (continued)		
Parameter	Default	Function	
AVERAGE	YES	Nodally averaged fields switch:	
		NO	Use shape function derivative fields.
		YES	Use nodally averaged field.

The **INTLINE** command integrates field quantities along a line, defined by its end points, (X1, Y1) and (X2, Y2) and its **CURVATURE**. Integrals on circular arcs can also be done using the **INTCIRCLE** command (page 3-133).

The functions integrated are:

- in xy coordinates: the Maxwell Stress, to give the x and y force per unit length and the anticlockwise torque per unit length with respect to the action point (XACTION, YACTION).
- in axisymmetry: the Maxwell Stress, to give the z force.
- one other **COMPONENT**.

Expressions for the **COMPONENT** can use as variables any of the system variables given in page 3-12.

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified. However, the integrals for force and torque are computed correctly for all values of **TIME**. (See page 3-19.)

The values of the integrals are assigned to the system variables **INTEGRAL** and in xy coordinates, **FX**, **FY** and **TORQUE**, or in axisymmetry, **FZ**.

The integrals can be evaluated using nodally averaged fields (**+AVERAGE**) or shape function derivative fields (**-AVERAGE**). The **TOLERANCE** can also be set to a value less than 1 to achieve a particular accuracy on the integration using a Runge-Kutta method or to a number greater than 1 to set the number of steps in a Simpson's Rule integration.

For best results, it is recommended that integration paths are completely in air and do not pass through elements which touch the surface of any iron or dielectric. The reason for this is that close to positions where the surface tangent direction changes, the errors will be large. If the line passes through non-air materials, the Maxwell stresses are evaluated as if the line is a vanishingly narrow cut in the material. This should be used with care since singular points at the ends of the cut can lead to a loss of accuracy.

The total force on a body should be computed from the Maxwell Stress integral along series of lines encircling the body in an anticlockwise direction. To enable the forces to be summed easily, an accumulator can be initialized to **ZERO** and the integrals added to or subtracted from it. **INTLINE** and **INTCIRCLE** share the same accumulator. The accumulated values are stored in system variables: **TOTAL_FX**, **TOTAL_FY**, **TOTAL_FZ**, **TOTAL_TORQUE** and **TOTAL_INTEGRAL**.

The **LINE** Command

Menu Route: **FIELDS**↓
 Graphs → **Along line**

Command Line Parameters:

Command	LINE	
Parameter	Default	Function
X1	<i>none</i>	X coordinate at start of line.
Y1	<i>none</i>	Y coordinate at start of line.
X2	<i>none</i>	X coordinate at end of line.
Y2	<i>none</i>	Y coordinate at end of line.
NP	10	Number of steps along line, i.e. NP +1 points.
CURVATURE	0	Curvature of line.
COMPONENT	POT	Expression of field components.
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.
		<i>number</i> Angle in degrees around ac cycle.
		AMPLITUDE Amplitude.
		PHASE Phase angle.
		TAVERAGE Time average.
AVERAGE	YES	Nodally averaged fields switch:
		NO Use shape function derivative fields.
		YES Use nodally averaged field.
GRAPH	YES	Graph display switch:
		NO No graph.
		YES Values displayed as graph.

Command	LINE (continued)		
Parameter	Default	Function	
PRINT	NO	Printed values switch:	
		NO	Values printed to file <i>opera2.lp</i> .
		YES	Values printed to screen and file <i>opera2.lp</i> .
AUTOMATIC	YES	Automatic graph scale switch:	
		NO	Vertical axis limits set by VMIN and VMAX .
		YES	Vertical axis set automatically.
VMAX	<i>none</i>	Upper limit of vertical axis of graph AUTOMATIC=NO .	
VMIN	<i>none</i>	Lower limit of vertical axis of graph AUTOMATIC=NO .	
ERASE	YES	Old graph erasure switch:	
		NO	New line drawn on existing axes.
		YES	Graphics window cleared and new axes drawn.
STYLE	AUTOMATIC	Line style:	
		AUTOMATIC	Program chooses a different style for each graph drawn on the same axes.
		0	Solid line.
		>0	Broken line.
COLOUR	AUTOMATIC	Line colour:	
		AUTOMATIC	Program chooses a different colour for each graph drawn on the same axes.
		>0	Colour number.
TITLE	<i>none</i>	Additional title for line.	

Command	LINE (continued)		
Parameter	Default	Function	
HOMOGENEITY	NO	Homogeneity switch:	
		NO	Values of COMPONENT used.
		YES	Homogeneity with respect to COMPONENT value at XREF, YREF used.
XREF	0	X or R coordinate of homogeneity reference point.	
YREF	0	Y or Z coordinate of homogeneity reference point.	

The **LINE** command calculates field quantities along a line, defined by its end points, (X1, Y1) and (X2, Y2) and its **CURVATURE** and displays them as tables of numbers and graphs. The field is evaluated at **NP+1** points. The integral of the component is calculated using the trapezium rule. Calculations of field values along lines are also available with the **CIRCLE** command (page 3-27) and the **PLOT** command (page 3-161).

Expressions for the **COMPONENT** can use as variables any of the system variables given in page 3-12. The values used can be the actual values of **COMPONENT** or the **HOMOGENEITY** of the **COMPONENT** with respect to the reference point (XREF, YREF). The homogeneity is evaluated using, e.g.

$$A_{\text{homo}} = \frac{A(x, y) - A(x_{\text{ref}}, y_{\text{ref}})}{A(x_{\text{ref}}, y_{\text{ref}})} \quad (3.14)$$

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified. The values can be evaluated using nodally averaged fields (**+AVERAGE**) or shape function derivative fields (**-AVERAGE**).

The user can choose whether or not to see printed tables of values or graphs. Printed values appear on the screen and are also written to the dialogue file *opera2.lp*. The graphs can have the vertical axis limits set automatically or by the values of **VMAX** and **VMIN** with **-AUTOMATIC**.

Multiple lines can be displayed on the same axes, the first being displayed with **+ERASE**, and the rest with **-ERASE**. The lines can be displayed with different line **STYLE** and **COLOURS** and a **TITLE** in addition to the **COMPONENT** can be added to the key below the graph. The **TITLE** is not remembered and must be

supplied each time it is needed. Titles including spaces or commas should be enclosed in quotation marks ('').

The **MINIMUM** and **MAXIMUM** values along the line are assigned to system variables.

The LMMOTION Command

Menu Route:

MODEL↓

Draw regions → Linear motion air gap

Command Line Parameters:

Sub-Command	LMMOTION		
Parameter	Default	Function	
MESHING	NO	NO	Mesh assuming that there will be no motion
		YES	Mesh using the groups specified
MOVINGGROUP	<i>none</i>	Group name for regions that are moving	
MEDIUMGROUP	<i>none</i>	Group name for the regions that allow movement	
CENTREX	0.0	Coordinate about which rotation occurs	
CENTREY	0.0		

Notes

The **LMMOTION** command allows the mesh generation to allow for moving meshes required by the **LM** solver.

Parameters

The **LMMOTION** command allows a more flexible mesh generation necessary for linear motion problems (LM). If **MESHING=YES** then this meshing technique will be used to generate the finite element mesh.

This mesh generation technique allows the positioning of a set of moving regions over another set of regions that allow movement. The moving regions are grouped together within the group defined in parameter **MOVINGGROUP**. The regions that allow movement are grouped together within the group defined in the **MEDIUMGROUP** parameter.

If rotational motion is required, the point of rotation is given by **CENTREX** and **CENTREY** parameters. If this is combined with linear motion, the centre of rotation is also moved.

Use of the LM Gap region

The linear motion meshing must be used with the **LM** solver. It can also be used to generate meshes for other solvers. It is particularly useful where positional variations are being studied, as it allows modifications to a model to be simplified.

Some important features regarding **LMMOTION** meshing:

- Any number of regions may be grouped within the **MOVINGGROUP**, and where the regions within this group are joined along a region side, continuity of subdivision information must be enforced.
- Any number of regions may be grouped within the **MEDIUMGROUP**, and where the regions within this group are joined along a region side, continuity of subdivision information must be enforced. Mesh continuity with other regions must also be enforced as normal.
- All regions in the **MOVINGGROUP** must lie within the regions in the **MEDIUMGROUP**.
- There is no requirement for mesh continuity between regions in the **MOVINGGROUP** and those in the **MEDIUMGROUP**.
- Regions in the **MOVINGGROUP** may touch or share an external boundary of the model, but may not go beyond the boundary.
- Regions in the **MOVINGGROUP** must not touch any boundary of a region that is not part of the **MEDIUMGROUP**.
- Regions in the **MEDIUMGROUP** should all be defined with the properties of air.
- Periodicity and non-zero potential boundary conditions may not be used.
- It is recommended that the element size at the boundary of the **MOVINGGROUP** is comparable with the element size within the **MEDIUMGROUP**, so that the element edges forming the region boundary during the remeshing process are similar in size.

The following is a brief summary of the meshing process:

- All regions except those in **MOVINGGROUP** are meshed normally. Those in **MOVINGGROUP** are meshed independently.
- All elements in the **MEDIUMGROUP** that lie underneath the **MOVINGGROUP** regions are removed and the element edges forming the boundary of the hole found. The boundary edges of the **MOVINGGROUP** elements are added to this set.
- The region formed from this set of element edges is meshed, and the new elements added into the final mesh to form a gap between moving and static parts.

- The elements from the moving regions are also included in the final mesh.

During the LM solution process, the moving regions are repositioned and the final 2 stages of this process are repeated for each new position. It is therefore very important that the regions grouped within the **MEDIUMGROUP** extend to allow sufficient motion.

The MESH Command

Menu Routes:

Equivalent to **MESH ERRORCHECK=NO:**

FILE↓

Read file → Re-mesh

Equivalent to **MESH ERRORCHECK=YES:**

MODEL↓

Mesh generator → Generate mesh

Command Line Parameters:

Command	MESH	
Parameter	Default	Function
TOLERANCE	5.0E-5	Coincidence tolerance on joining regions together.
DISPLAY	NO	NO Polygon mesh display switch:
		YES No mesh generator display. Display the activity of polygon mesh generator.
ERRORCHECK	NO	Check and display errors in the mesh:
		NO No error checking. YES Error checking.
DELETE	NO	Delete existing mesh and solution data:
		NO Do not delete. YES Delete.

The complete finite element mesh can be generated at any time by use of the **MESH** command. This forms the data necessary to analyse the problem or interrogate the results, i.e. node table, element definition table, element properties, boundary nodes, etc. It is essential to use the mesh command after drawing the regions, and before writing a data file for analysis, in order to check the data given. The nodes with boundary conditions other than **NO** and the mesh in a **BACKGROUND** region can be displayed using the **RECONSTRUCT** command only after the mesh has been generated (see page 3-170).

As each region is added to the mesh, the program removes redundant nodes where two regions meet. This is done by coalescing nodes which are closer than the tolerance. By default, the tolerance is

$$5 \times 10^{-5} \times |(\text{maximum coordinate of region being added})|$$

but the factor, 5×10^{-5} can be changed using the **TOLERANCE** parameter. The tolerance is included in any data file written, and if the mesh has to be generated for post processing, the value read from the results file will be used unless a value is given on the **MESH** command.

During mesh generation in **POLYGONAL** regions, the activity of the mesh generator can be displayed. This shows the initial mesh, insertion of extra nodes and relaxation of node positions. The **DISPLAY** parameter switches this facility on and off.

- If **ERRORCHECK=YES**, the program displays the boundary of the mesh, to show whether redundant nodes have been eliminated satisfactorily where regions touch. This display consists of all element edges which do not occur twice. Thus adjacent regions which do not match are also shown. In keyboard mode, the user has opportunity to display part of the boundary by changing the area shown in the same way as in the **ZOOM** command. The error check display can be redisplayed using

DISPLAY↓

Style → **Mesh error check**
 → **Return**
Refresh

equivalent to the command, **RECO FILL=ERROR** (see page 3-170). Errors can be corrected automatically using the **TEST** command (page 3-193) or by hand using the **MODIFY** command (page 3-149).

- Also with **ERRORCHECK=YES**, the nodes with boundary conditions other than **NO** are displayed. The **MESH** command then automatically runs the **CHECK** command to check the data. If any errors are found the mesh will not be stored.
- The option **ERRORCHECK=NO** allows the mesh to be generated without affecting the display. This can be useful in overlaying results from several files, when the mesh files have been deleted.

Any command which changes the stored data (**DRAW, COPY, ERASE, MODIFY, SET, SYMMETRY, TEST**) can also destroy the mesh data.

After loading in an RM solution, the model can not be modified until the mesh and the solution data has been removed with **DELETE=YES**.

If the mesh data exists when the **WRITE** command is used, the mesh data will also be stored in a file. Similarly, an appropriately named mesh file can be **READ** into

the program, at the same time as the region data. This avoids expensive recalculation of mesh data (see [page 3-168](#) and [page 3-211](#)).

The program maintains an 'extra' title block (see [page 3-195](#)) at the right of the graphics window including 'No Mesh' or the numbers of nodes and elements as appropriate.

The **MODIFY** Command

The **MODIFY** command is the main data editing command for the pre and post processor. Corners and face properties of regions can also be edited using the **EDIT** command (page 3-76).

MODIFY can be used in two ways: keyboard or cursor. In keyboard mode, parameters are used to specify which regions are to be modified and to define new values for some of the parameters of those regions. In cursor mode a point (region corner), face or region can be chosen by pointing to it on the graphics window, and then new values can be given for position, face properties or region properties. Cursor mode is selected by issuing the **MODIFY** command with no parameters.

Before each **MODIFY** command, the current set of regions is saved in memory so that it can be restored using the **UNDO** command (page 3-201) if regions are modified in error.

Keyboard **MODIFY**

Menu Route:

```
MODEL↓
  Modify regions      → Region group
                    → Multiple regions
```

If any parameters are given on a **MODIFY** command, keyboard mode is assumed.

The **MODIFY** command has 63 parameters, the last 60 of which are the same as the parameters of the **DRAW** command (page 3-46). The first 3 select the range of regions to be **MODIFIED** and whether the picture of the regions should be reconstructed (**RECO**) afterwards (page 3-170).

Command Line Parameters (first 3):

Command	MODIFY : The first 3 parameters	
Parameter	Default	Function
REG1	<i>none</i>	Regions to be modified.
		<i>number</i> Regions in range REG1 to REG2 .
		<i>group</i> Regions in group named.

Command	MODIFY: The first 3 parameters (continued)		
Parameter	Default	Function	
REG2	REG1	Last region to be modified. * for highest region number.	
ERASE	NO	Clear graphics window and re-display switch:	
		NO	Regions MODIFIED are erased and re-drawn.
		YES	Graphics window cleared and picture re-displayed.

The regions to be modified are specified by number or group name using the parameters REG1 and REG2 (see page 3-122). If REG1=number and REG2 is omitted, REG2 is assumed to be the same as REG1 and only that one region is modified.

After modifications, the changed regions can be selectively erased from the display and re-drawn (-ERASE), or the complete picture of the regions can be reconstructed (RECO +ERASE).

The remaining 60 parameters have no default values. This means that if a parameter is not given a value, the regions retain the value they had before the command. Values for numerical parameters can be given as expressions in terms of other parameters of the same region, the expressions being re-evaluated for each region in turn. The system variable AREA, the total area of the region including replications, can also be included in such expressions (see page 3-17).

The section on the DRAW (page 3-46) command defines these parameters. There are no parameters for MODIFYING coordinates or face properties of regions with SHAPE=POLYGON or SHAPE=BACKGROUND. These can only be changed in cursor mode or by using the EDIT command (page 3-76). The SHAPE code of a region can only be changed between shapes C and CS or between shapes H and Q.

The first 4 characters of the 60 parameters are:

SHAP	MATE	PERM	DENS	COND	PHAS
N	SYMM	VELO	X12	X34	Y14
Y23	X1	Y1	X2	Y2	X3
Y3	X4	Y4	R12	R34	P14
P23	P1	P2	P3	P4	XCEN
YCEN	ANGL	NX	NY	DX	DY
MIRR	TMIR	ROTA	TROT	F1	F2
F3	F4	V1	V2	V3	V4

DV1	DV2	DV3	DV4	N1	N2
N3	N4	B1	B2	B3	B4

In menu mode the option **Region group** presents Parameter Boxes with subsets of the parameters. Values should only be given for the parameters which should be changed.

In menu mode the option **Multiple regions** allows the values of **REG1** and **REG2** and up to 4 parameter names and values to be entered into a DialogBox.

Cursor MODIFY

If no parameters are given on a **MODIFY** command, cursor mode is assumed.

In the cursor mode of the **MODIFY** command, items (points, faces or regions) to be changed are selected by pointing to them on the display with the cursor, and using one of the single character cursor commands. Other cursor commands allow extra points to be inserted into polygons, the picture to be re-drawn or to obtain help.

Region corners can be moved with the **N**, **P** and **T** cursor hits. If a region corner is selected to be moved with **N**, all regions which share that corner are modified. If **P** is used for selection, then only the region containing the cursor position is changed. Cursor hit **T** introduces a transformation operation, allowing many region corners to be moved in the same way.

Faces can be selected for changes to their parameters with the **F** cursor hit. When an internal face is modified, both regions which share the face are modified, and any change in subdivision is carried through the mesh as far as possible by continuing to the neighbours on opposite sides of regions with **SHAPE=C** or **SHAPE=H**.

Faces of polygons can be divided into two equal sections with the **I** cursor hit (**I**nsert new point). The new faces have the same curvature, bias and boundary conditions as the old face, and half the subdivision (rounded up).

MODIFY
Command
Cursor Hits

The following cursor hit characters can be used to select parts of the data to be modified:

MODIFY Command Cursor Hits	
Cursor hit	Function
A	Apply a transformation to a region corner. The transformation must be set with the T cursor hit first.
E	Clear the graphics window and RECONSTRUCT the picture. The program prompts for RECONSTRUCT parameters which can be set if necessary. Type <Return> to use existing settings.
F	Select for modification the closest face of the region containing the cursor.
H	Display help message explaining all the cursor options.
I	Select the closest face of the region containing the cursor for insertion of a new point half way along face.
N or P	Select for modification the closest corner of the region containing the cursor. Only one region modified with P ; neighbouring regions changed with N .
Q	Leave MODIFY command.
R	Select for modification the region containing the cursor.
T	Start a region corner transformation operation.
V	Toggle between picking in all regions and picking in visible regions only.
Z	Execute the ZOOM (see page 3-212) command to change size of the picture. The program prompts for ZOOM command cursor hits.

After each cursor hit which needs further input, appropriate prompts are given. The format of such input is described below.

Pick Operations

In all 'pick' operations to select a point, face or region the program first finds the region which contains the cursor. By default, only the displayed regions are searched. This setting can be toggled to make the program search all regions or to return to the default setting.

This affects pick operations in **EMIT**, **ERASE**, **EXTRA**, **PLOT**, **GROUP**, **MODIFY** and **SYMMETRY** commands as well.

Menu Route: MODEL↓
 Modify regions → Search all regions
 → Only visible regions

Without GUI The V cursor hit toggles the setting of the searching algorithm.

Modifying a Region Face with GUI

Menu Routes: MODEL↓
 Modify regions → Modify Sides by picking

Three options are given:

- **All properties of one side** allows the user to pick one side and the set values for the subdivisions, curvature, bias and boundary condition.
- **Subdivision of many sides** first requests the subdivision. This can be given as a number or as an expression in terms of the existing subdivision; e.g. a subdivision of **N*2** will double the existing value. The program then allows the user to apply the subdivision to as many region sides as required. This can be done in 3 ways:
 - **-with old bias** does not change the bias.
 - **-with bias=cursor** uses the position of the cursor with respect to side to determine the bias. The cursor should be closer to the end where smaller elements are required.
 - **-with bias=0.5** sets the bias so that element sizes are uniform along the side.
- **Swap curvature** changes the sign of the curvature of the sides which are subsequently selected.

The menu option **Modify Sides by number** accesses the **EDIT** command (page 3-76).

Menu Route: MODEL↓
 Modify regions → Modify Points ...
 ... by picking → Insert in side

This divides a polygon side into 2 equal parts.

Modifying Boundary Conditions with the GUI

Menu Route:

MODEL↓
Boundary cond.

Boundary condition should be set using the appropriate choice of **SCALAR POTENTIAL** or **VECTOR POTENTIAL**. For each potential, the following boundary conditions can be applied:

- Normal field strength or flux density forced to be zero.
- Tangential field strength or flux density forced to be zero.
- Periodic symmetry.
- Fixed value of potential. A value of potential is requested before the sides are selected.
- No boundary condition.

In each case the condition can be applied to as many sides as necessary.

Modifying a face without the GUI

A face is selected by positioning the cursor inside a region close to the required face and typing the **F** cursor hit. If the cursor is positioned nearer to one end of the face, its position can be used to set the subdivision **BIAS** such that there will be smaller elements towards that end of the face (**BIAS=CURSOR**).

After selection, the region number and face number are reported, together with the value of **BIAS** indicated by the cursor position with respect to the ends of the face. The user can then set any of the parameters for the face, type **!** to request current values or **<Return>** to keep current values.

Any changes to subdivision (**N** and **BIAS**), or **CURVATURE** are copied to any other region which shares the edge. The subdivision is also copied to the neighbours on opposite faces of regions with **SHAPE=C** or **SHAPE=H**, continuing through the regions as far as possible. The neighbours of replications of any regions which are changed will not necessarily be changed. It is not possible to change the **CURVATURE** of a face belonging to or shared with a region with **SHAPE=C** or **SHAPE=CS**.

The parameters are:

MODIFY Face Properties			
Parameter	Default	Function	
N	<i>old value</i>	Number of elements along face.	
F	<i>old value</i>	Type of boundary condition for face:	
		DV	Normal derivative of potential = DV .
		NO	Internal face.
		SYMMETRY	One of symmetry (periodicity) pair.
V	<i>old value</i>	Value of potential = V .	
DV	<i>old value</i>	Normal derivative of potential if F=DV .	
BIAS	<i>old value</i>	Subdivision bias for face:	
		<i>number</i>	Set value explicitly.
		CURSOR	Take value implied by cursor position.
CURVATURE	<i>old value</i>	Curvature of face (Shapes H , Q and POLYGON).	

Modifying a Single Region Corner

Region corners can be moved singly, or many corners can be moved under the same transformation. The first option allows a single region corner to be selected and repositioned. Neighbouring regions can be change simultaneously, if required.

Menu Route:

```
MODEL↓
  Modify regions → Modify Points ...
                  ... by picking → For all regions
```

This selects a point to be moved. All regions which use the point will be affected.

```
MODEL↓
  Modify regions → Modify Points ...
                  ... by picking → Only for region
```

This selects a point to be moved. Only the region containing the cursor when the point is picked will be affected.

In both cases the point can be repositioned using the menu items in the table below.

The menu option **Modify Points by number** accesses the **EDIT** command (page 3-76).

Without GUI:

A single point (region corner) is selected by positioning the cursor inside a region close to the required corner and typing the **N** or **P** cursor hit. After selection the point is high-lighted, and the cursor reappears with a new set of cursor commands for repositioning. This includes a set of options similar to that used in polygon input (page 3-66), including **K** for keyboard input, **O** for nearest old point and **C**, **N** and **X** for use of construction lines (see page 3-68).

A 'snap to grid' facility can be switched on by setting the cartesian or polar grid size greater than zero. When the grid is switched on, all **<space>** cursor hits take the nearest grid point. Switching on a cartesian grid also switches off a polar grid and *vice versa*. The grid sizes can be set or re-set following a **G** or **P** cursor hit.

In keyboard input, following cursor hit **K**, expressions can be given for the new position in terms of the old position, using system variables **X** and **Y**.

If the **N** cursor hit is used and the region corner is shared by more than one region, all those regions, except **SHAPES C** and **CS**, will be changed.

The full set of cursor commands is given below.

Repositioning a Region Corner

Point Repositioning Options		
Cursor hit	Menu Option	Function
<space>	At cursor	Reposition point at a new point at the cursor position or the nearest grid point.
C		Input, list or delete construction lines (see page 3-68).
G	Set grid size (xy)	Set the size of cartesian grid. If grid size is non-zero, all <space> cursor hits select the nearest grid point.
H		Display help message explaining all the cursor options.

Point Repositioning Options (<i>continued</i>)		
Cursor hit	Menu Option	Function
K	XY text input	Switch to keyboard input. New coordinates maybe given in free format using expressions in terms of old X and Y if necessary.
N	Nearest C_line	Reposition point at a new point on the nearest construction line.
O	At old point	Reposition point at the old point nearest the cursor position.
P	Set grid size (polar)	Set the size of polar grid. If grid size is non-zero, all <space> cursor hits select the nearest grid point.
Q	Quit	Leave point where it is.
T	Print old XY	Type the coordinates of the point.
X	At intersection	Reposition point at a new point at the closest construction line intersection or end.
Z	Zoom display	Execute the ZOOM (see page 3-212) command to change size of the picture.

Transforming Many Region Corners

This option allows many region corners to be selected and then repositioned using a transformation, which can be expressed in terms of the old coordinates.

Menu Route:

```
MODEL↓
  Modify regions → Modify Points ...
                  ... by picking → Transform many points
```

The program prompts for expressions which define the new values of **X** and **Y** in terms of their old values and the allows the transformation to be applied to as many points as necessary.

Without GUI:

The transformation operation is selected with the **T** cursor hit (the position of the cursor does not matter). The program prompts the user to supply expressions for the new coordinate positions in terms of the old. For example, **x+1 y+2** will move selected points by one unit in x and 2 units in y. The user can then select as

many region corners as required an apply the transformation to them with **A** cursor hits. All regions shared by the points (except shapes **C** and **CS**) are changed, and the display is immediately updated to show the new positions.

Modifying Region Properties

Menu Route:

MODEL↓

Modify regions → Modify Region → Pick Region

After selecting a region the appropriate group of parameters can be selected and modified.

Without GUI:

A region is selected by positioning the cursor inside it and typing the **R** cursor hit. After selection the region is high-lighted. Any parameter of the region can then be reset.

The parameters are divided into groups, each group having a sub-command: **REGION** for region properties, **COORDINATES** for coordinates, **FACE** for face properties, and **LOCAL** for local coordinates and replications. The default values of each parameters are the current values for the region selected. Only the region selected is modified.

The coordinates and face properties of polygons can not be changed in this way. Their coordinates can only be modified with cursor hit **N** and their face properties with cursor hit **F**.

The system variable **AREA**, the total area of the region including replications, can also be included in expressions for numeric parameters (see [page 3-17](#)).

The sub-commands and their parameters are:

Region MODIFY Sub-Commands	
Sub-Command	Function and Parameters
COORDINATES	Coordinates of SHAPES C and CS R12, R34, P14, P23, P1, P2, P3, P4
COORDINATES	Coordinates of SHAPES H and Q X12, X34, Y14, Y23, X1, Y1, X2, Y2, X3, Y3, X4, Y4

Region MODIFY Sub-Commands <i>(continued)</i>	
Sub-Command	Function and Parameters
FACE	Face properties (not SHAPE=POLYGON or SHAPE=BACKGROUND) F1, F2, F3, F4, V1, V2, V3, V4, DV1, DV2, DV3, DV4, N1, N2, N3, N4, B1, B2, B3, B4, P1, P2, P3, P4
LOCAL	Local Coordinates and replications XCENTRE, YCENTRE, ANGLE, NX, NY, DX, DY, MIRROR, TMIRROR, ROTATIONS, TROTATION
QUIT	Finish MODIFYING this region.
REGION	Region properties SHAPE (not available for polygons), MATERIAL , PERM , DENSITY , CONDUCTIVITY , PHASE , N , SYMMETRY , VELOCITY

The parameters are fully described under the **DRAW** command, [page 3-46](#).

The PAN Command

Menu Route: DISPLAY↓
 Pan

*Command Line
Parameters:*

Command	PAN
No Parameters	

PAN executes a RECONSTRUCT command (see [page 3-170](#)) having set the first 4 parameters (XMIN, XMAX, YMIN, YMAX) by use of the cursor.

Two cursor hits must be given (any character can be used): the first identifies a picture coordinate, and the second a window coordinate. The parameters are adjusted and the picture moved so that the picture coordinate appears at the window coordinate. The size and aspect ratio of the picture and other RECONSTRUCT parameters are unchanged.

The **PLOT** Command

Menu Route:

FIELDS↓

Graphs → Along region sides

Command Line Parameters:

Command	PLOT		
Parameter	Default	Function	
NEW	YES	New graphs switch:	
		NO	Use the points from the last CIRCLE , PLOT or LINE command.
		YES	Create a new line from region sides.
CURVATURE	0	Curvature of line.	
COMPONENT	POT	Expression of field components.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		number	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
AVERAGE	YES	Nodally averaged fields switch:	
		NO	Use shape function derivative fields.
	YES	Use nodally averaged field.	
GRAPH	YES	Graph display switch:	
		NO	No graph.
		YES	Values displayed as graph.

Command	PLOT (continued)		
Parameter	Default	Function	
PRINT	NO	Printed values switch:	
		NO	Values printed to file <i>opera2.lp</i> .
		YES	Values printed to screen and file <i>opera2.lp</i> .
AUTOMATIC	YES	Automatic graph scale switch:	
		NO	Vertical axis limits set by VMIN and VMAX .
		YES	Vertical axis set automatically.
VMAX	<i>none</i>	Upper limit of vertical axis of graph AUTOMATIC=NO .	
VMIN	<i>none</i>	Lower limit of vertical axis of graph AUTOMATIC=NO .	
ERASE	YES	Old graph erasure switch:	
		NO	New line drawn on existing axes.
		YES	Graphics window cleared and new axes drawn.
STYLE	AUTOMATIC	Line style:	
		AUTOMATIC	Program chooses a different style for each graph drawn on the same axes.
		0	Solid line.
		>0	Broken line.
COLOUR	AUTOMATIC	Line colour:	
		AUTOMATIC	Program chooses a different colour for each graph drawn on the same axes.
		>0	Colour number.
TITLE	<i>none</i>	Additional title for line.	

Command PLOT (continued)			
Parameter	Default	Function	
HOMOGENEITY	NO	Homogeneity switch:	
		NO	Values of COMPONENT used.
		YES	Homogeneity with respect to COMPONENT value at XREF, YREF used.
XREF	0	X or R coordinate of homogeneity reference point.	
YREF	0	Y or Z coordinate of homogeneity reference point.	

New PLOTS

The **PLOT NEW=YES** command calculates field quantities along region sides, and displays them as tables of numbers and graphs. The integral of the component is calculated using the trapezium rule. The region sides are selected graphically with a **<space>** cursor hit (see page 3-152). The cursor should be positioned inside the region, next to the required side. If the region side is an interface between different materials, where the normal and tangential components of fields would be discontinuous, it is important to select the side from within the appropriate region. Up to 1000 region sides can be selected to form a continuous line. After the last region side has been selected, the cursor hit **Q** requests that the fields be calculated at each node. Cursor hit **A** can be used to abort the command.

Cursor-hits and Menu Items

PLOT Cursor Sub-commands		
Character	Menu item	Function
<space>	Select region side	Add the region side to the line.
E	Deselect region side	Remove the last selected region side from the line.
Q	Draw graph	If regions sides have been selected, evaluate fields and display values. Otherwise abort the PLOT command.
A	Return	Abort the PLOT command.
Z	Zoom	Re-draw the picture, Zooming in or out.

Old PLOTS

The **PLOT NEW=NO** command calculates new field quantities at the set of points defined by the last **CIRCLE**, **PLOT NEW=YES** or **LINE** command. The integral of the component is calculated using the trapezium rule.

Other parameters

All the other parameters apply to **PLOT NEW=YES** and **PLOT NEW=NO**.

Expressions for the **COMPONENT** can use as variables any of the system variables given in page 3-12. The values used can be the actual values of **COMPONENT** or the **HOMOGENEITY** of the **COMPONENT** with respect to the reference point (**XREF**, **YREF**). The homogeneity is evaluated using, e.g.

$$A_{\text{homo}} = \frac{A(x, y) - A(x_{\text{ref}}, y_{\text{ref}})}{A(x_{\text{ref}}, y_{\text{ref}})} \quad (3.15)$$

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified. The values can be evaluated using nodally averaged fields (**+AVERAGE**) or shape function derivative fields (**-AVERAGE**).

The user can choose whether or not to see **PRINT**ed tables of values or **GRAPHS**. **PRINT**ed values appear on the screen and are also written to the dialogue file *opera2.lp*. The graphs can have the vertical axis limits set **AUTOMATIC**ally or by the values of **VMAX** and **VMIN** with **-AUTOMATIC**.

Multiple lines can be displayed on the same axes, the first being displayed with **+ERASE**, and the rest with **-ERASE**. The lines can be displayed with different line **STYLE** and **COLOUR**s and a **TITLE** in addition to the **COMPONENT** can be added to the key below the graph. The **TITLE** is not remembered and must be supplied each time it is needed. **TITLES** including spaces or commas should be enclosed in quotation marks (' ').

The **MINIMUM** and **MAXIMUM** values along the line are assigned to system variables.

The POINT Command

Menu Route: **FIELDS**↓
 Point values

Command Line Parameters:

Command	POINT		
Parameter	Default	Function	
COMPONENT	POT	Field component for contours.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		<i>number</i>	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
METHOD	CURSOR	Method of supplying field point coordinates:	
		CARTESIAN	XP and YP are cartesian coordinates.
		CURSOR	Use cursor.
		POLAR	XP and YP are polar coordinates.
XP	0	X or R coordinate of field point.	
YP	0	Y or Φ coordinate of field point.	
HOMOGENEITY	NO	Homogeneity switch:	
		NO	Values of COMPONENT .
		YES	Homogeneity with respect to COMPONENT value at XREF , YREF .
XREF	0	X or R coordinate of homogeneity reference point.	
YREF	0	Y or Z coordinate of homogeneity reference point.	

The **POINT** command calculates field quantities at a point. The point coordinates can be given in **CARTESIAN** or **POLAR** coordinates or by a cursor hit. If **METHOD=CURSOR** is used, many points can be given by positioning the cursor and typing any character except **Q** which ends the command.

Expressions for the **COMPONENT** can use as variables any of the system variables given in [page 3-12](#). The value displayed can be the actual value of **COMPONENT** or the **HOMOGENEITY** of the **COMPONENT** with respect to the reference point (**XREF, YREF**). The homogeneity is evaluated using, e.g.

$$A_{\text{homo}} = \frac{A(x, y) - A(x_{\text{ref}}, y_{\text{ref}})}{A(x_{\text{ref}}, y_{\text{ref}})} \quad (3.16)$$

All system variables relevant to the field point are set.

For Steady-state ac solutions, each system variable in an expression for **COMPONENT** is evaluated at the **TIME** specified (see [page 3-19](#)).

The PRINT Command

Menu Route:

OPTIONS↓
Print model data

Command Line Parameters:

Command	PRINT	
Parameter	Default	Function
REG1	1	Regions to be printed.
		<i>number</i> Regions in range REG1 to REG2.
		<i>group</i> Regions in group named.
REG2	*	Last region to be printed. * for highest region number.

The **PRINT** command lists the data of the regions on the display and into the file *Opera2d_PP_n.lp*. The regions are given by number or group name using parameters **REG1** and **REG2** (see [page 3-122](#)).

The data is divided into sections and the user is prompted to type **R** for region properties (including the areas and volumes of the regions), **C** for coordinates, **F** for face properties, **L** for local coordinates and replications or **Q** to leave the **PRINT** command.

The **PRINT** command displays the cross sectional area and the rotational volume (for **SYMM=AXI**) of the regions if the geometry has been displayed on the screen. Hence, after reading a data file, values of area and volume will be invalid until the **RECONSTRUCT** command ([page 3-170](#)) has been used.

The **READ** Command

Menu Route:

FILE↓
Read file → Read model

Command Line Parameters:

Command	READ	
Parameter	Default	Function
FILE	<i>none</i>	Name of OPERA-2d data or results file.
CASE	1	The results case number for AC and TR.
SOL1	<i>none</i>	The name of an additional solution table.
SOL2	<i>none</i>	The name of an additional solution table.
SOL3	<i>none</i>	The name of an additional solution table.
SOL4	<i>none</i>	The name of an additional solution table.
SOL5	<i>none</i>	The name of an additional solution table.
SOL6	<i>none</i>	The name of an additional solution table.
SOL7	<i>none</i>	The name of an additional solution table.
SOL8	<i>none</i>	The name of an additional solution table.
SOL9	<i>none</i>	The name of an additional solution table.
SO10	<i>none</i>	The name of an additional solution table.

The **READ** command reads a OPERA-2d data or results file into the program. There are two parameters which define the name of the **FILE** and, for Steady State ac or Transient results files, the **CASE** number. If no file name extension is given, the extension *op2* (i.e. a data file) is assumed. To read a results file, the full file name including file name extension (*an*) must be given, where *an* is one of the two letter analysis program names, *ac*, *rm*, *sa*, *sp*, *st*, *th*, *tr*, or *vl*.

Before reading the file specified by the command, the program looks for a mesh file with the same file name but the extension *mesh*. If such a file exists, it will be read first. Mesh files are binary files containing the complete finite element mesh data.

OPERA-2d data files contain:

- solution parameters
- region data
- BH data

- symmetry conditions
- data set by the **EXTRA** command including the current units set
- region groups
- user variables

Results file also contain this data with the solution potentials, current/charge densities and permeabilities/permittivities appended. Thus all the data necessary for adjusting the specification of a problem is available during a post processing session.

The stress, thermal and space charge beam analysis program are supplied with data and can return solutions to the pre and post processor in tables. Up to 10 tables can be loaded from a data or results file by giving their names to parameters **SOL*n***. The values in tables can be accessed by the post processing commands by their system variable names. For more information about tables, see [page 3-113](#).

The **RECONSTRUCT** Command

Menu Route: DISPLAY↓
 Refresh

**Command Line
 Parameters:**

Command	RECONSTRUCT	
Parameter	Default	Function
XMIN	0	Minimum X coordinate.
XMAX	10	Maximum X coordinate.
YMIN	0	Minimum Y coordinate.
YMAX	10	Maximum Y coordinate.
MATERIAL	ALL	Material(s) to be displayed:
	<i>number</i>	0, 1, 3, ..., 102: material code.
	ALL	All materials.
NOT	ANY	Material number to be omitted.
	<i>number</i>	0, 1, 3, ..., 102: material code.
	ANY	No materials to be omitted.
MESH	NO	Mesh display switch:
	DISPLAY	Activity of POLYGON mesh generator is displayed.
	NO	No mesh is displayed.
	YES	Mesh elements displayed.
REG1	1	Regions to be displayed.
	<i>number</i>	Regions in range REG1 to REG2 .
	<i>group</i>	Regions in group named.
REG2	*	Last region to be displayed. *for highest region number.

Command	RECONSTRUCT (continued)		
Parameter	Default	Function	
NODES	CORNERS	Node display switch:	
		ALL	All nodes displayed.
		BOUNDARY	All nodes with boundary conditions.
		CORNERS	Corner nodes of regions.
	NONE	No nodes displayed.	
FILL	REGION	Colour fill switch:	
		BOUNDARY	Line display of material outlines.
		ELEMENTS	Elements filled with material colour and MATERIAL outlines drawn.
		ERROR	Redraw the MESH command error check display.
		NO	Line display of regions.
		MATERIAL	Regions filled with material colour and MATERIAL outlines drawn.
		REGION	Regions filled with material colour and REGION outlines drawn.
		YES	Regions filled with material colour with no outlines.
LABEL	YES	Region numbering switch:	
		NO	No labels.
	YES	Regions labelled with region numbers.	
AXES	YES	Axes display switch:	
		NO	No axes.
		REDRAW	Axes re-drawn after regions.
	YES	Axes displayed once.	
ERASE	YES	Picture erase switch:	
		NO	Old picture not erased.
	YES	Old picture erased.	

Command	RECONSTRUCT <i>(continued)</i>		
Parameter	Default	Function	
CLINES	YES	Construction line switch:	
		NO	Construction lines not displayed.
		YES	Construction lines displayed.
BACKGROUND	YES	Background region switch:	
		NO	Background region not displayed.
		YES	Background region displayed.

The **RECONSTRUCT** command displays pictures of the model. The parameters allow the scaling of the picture to the graphics window to be set and the type of picture to be set in various ways.

Picture Coordinates

The first four parameters (**XMIN**, **XMAX**, **YMIN**, **YMAX**) set the coordinate limits of the picture. If $\mathbf{XMAX - XMIN} \approx \mathbf{YMAX - YMIN}$, then the value of **XMAX** is adjusted to match the aspect ratio of the graphics display so that circles will appear round.

If $\mathbf{XMAX} \leq \mathbf{XMIN}$ or $\mathbf{YMAX} \leq \mathbf{YMIN}$, the program calculates the bounding rectangle which fits the selected regions and displays within that.

These parameters can also be set by the **PAN** and **ZOOM** commands (page 3-160 and page 3-212).

Regions and Materials

There are four parameters which control which regions are displayed. **REG1** and **REG2** set a range of region numbers. **MATERIAL** sets a material number, the only material to be displayed, or **ALL** for all materials. **NOT** sets the number of a material to be omitted from the display, or **ANY** for no omissions.

Alternatively, **REG1** can be used to specify a region group name. In this case **REG2**, **MATERIAL** and **NOT** are ignored (see page 3-122).

Picture Style

The style of the picture is set by the **FILL** parameter, using the Menu Route:

Menu Route:

DISPLAY↓
Style

The following options are available in the sub-menu. The keyboard equivalents are also listed.

- **Line drawing** or **-FILL** gives a line drawing of all region boundaries.
- **Material boundary** or **FILL=BOUNDARY** gives a line drawing of the material boundaries. This is only possible if the mesh exists. If the mesh does not exist, **FILL=BOUNDARY** is the same as **-FILL**.
- **Fill (no outlines)** or **+FILL** gives filled areas (on a suitable graphics device). Other **FILL** options specify how filled areas are outlined.
- **Fill regions** or **FILL=REGION** gives filled areas with the outline of each region displayed as well.
- **Fill materials** or **FILL=MATERIAL** also gives filled areas, with the outline of the materials displayed. This is only possible if the mesh exists. If the mesh does not exist, **FILL=MATERIAL** is the same as **FILL=REGION**.
- **Fill elements** or **FILL=ELEMENTS** has the same effect as **FILL=MATERIAL** but since each element is filled, it can cope with meshes produced by the adaptive analysis programs, in which, for curved sided regions, the elements might not fit the original region shape.
- **Mesh error check** or **FILL=ERROR** redisplay the error check display, which is given by the **MESH** command (see [page 3-146](#)).

Labels, Construction Lines and Axes

Region number **LABELS** can be switched on or off with **±LABEL**, the construction lines (used in the commands **DRAW SHAPE=POLYGON**, see “**POLYGON and BACKGROUND Region Input**” on [page 3-54](#), or **MODIFY**, see “**The MODIFY Command**” on [page 3-149](#)) with **±CLINE**, and the background region with **±BACKGROUND**. With **FILLED** displays, the axes are obliterated by the regions. They can be re-displayed after the regions with **AXES=REDRAW**. Additional **RECONSTRUCT** commands can be given without clearing the old picture first with **-ERASE**.

Nodes and Elements

The finite element mesh can be displayed over the regions. The elements can be displayed at any time with **+MESH**. However, the mesh of a background region can only be displayed if the mesh has been generated with the **MESH** command. If the mesh is generated for the display, the activity of the **POLYGONAL** mesh generator can be shown with **MESH=DISPLAY**.

CORNER nodes (i.e. the defining points) of regions, and nodes with a **BOUNDARY** condition can be displayed at any time. If the mesh exists (see [page 3-146](#)) **ALL** the **NODES** can be displayed. **NODES=NONE** prevents even the corner nodes from being displayed.

The SET Command

Menu Routes: MODEL↓
 Solution Type

and

 FIELDS↓
 Field options

Command Line Parameters:

Command	SET		
Parameter	Default	Function	
SYMMETRY	XY	Coordinate system:	
		AXISYMMETRY	Axisymmetry (R≡X, Z≡Y).
		XY	Cartesian.
SOLUTION	AT	Solution potential:	
		AT	Total vector potential.
		AXI	Modified rA for axisymmetric magnetics.
		RA	rA for axisymmetric magnetics.
	V	Reduced scalar potential for magnetostatics or scalar potential forelectrostatics.	
ELEMENT	LINEAR	Element type:	
		LINEAR	First order elements.
		QUADRATIC	Second order elements.
FREQUENCY	1	This parameter is no longer used. Use the SOLVE command to set frequency (page 3-180).	

Command	SET (continued)		
Parameter	Default	Function	
FIELD	MAGNETIC	Field type:	
		ELECTRIC	Electrostatics.
		MAGNETIC	Magnetics.
HXX	0	X component of magnetic field added to integral fields	
HYY	0	Y component of magnetic field added to integral fields	
REFXAXIS	NO	Reflection of model in X or R axis	
		INVE	Reflection with zero tangential field
		NO	No reflection
		YES	Reflection with zero normal field
REFYAXIS	NO	Reflection of model in Y or Z axis	
		INVERSE	Reflection with zero tangential field
		NO	No reflection
		YES	Reflection with zero normal field
ROTATIONS	1	The number of rotational copies of the mesh, around the global origin.	
ROTSIGN	POSITIVE	Sign of alternate rotational copies:	
		NEGATIVE	Alternate copies have negative current density.
		POSITIVE	All copies have the same sign of current density.
FLDTYPE	TOTAL	Integral field option:	
		COIL	Only include fields from driving currents
		TOTAL	Include fields from all currents and magnetisation.

Command	SET (continued)		
Parameter	Default	Function	
INTEGRAL	NO	Field calculation method:	
		NO	Fields from finite element mesh
		YES	Fields from integration of current density and magnetisation.

The **SET** command has three groups of parameters. The first affects the way the analysis programs operate. The other two affect how the post processing commands work. **SET** parameters are stored in OPERA-2d data files. After data has been **READ** from a file (see page 3-168), the values from the file become the new default values.

Analysis Program Options

The **SYMMETRY** parameter is used to choose between infinite 2-dimensional **XY** coordinates or **AXISYMMETRIC** coordinates. In axisymmetry, R is equivalent to X and Z to Y. Changing **SYMMETRY** destroys any solution in the program.

Several different solution potentials (**SOLUTION**) can be used with OPERA-2d. They determine the meaning of the system variable **POT**.

Solution Potentials and the System Variable POT	
SOLUTION	POT
AT	The vector potential: A_z or A_θ
AXI	Radial coordinate \times vector potential: rA_θ
RA	Radial coordinate \times vector potential: rA_θ
V	Reduced magnetic or electric scalar potential: ϕ

The sections on the analysis programs in the User Guide describe these potentials in more detail and make recommendations concerning which should be used. Changing **SOLUTION** destroys any solution in the program.

OPERA-2d can use **LINEAR**, 3 noded triangular elements or **QUADRATIC** elements with mid-side nodes as well. The quadratic elements have straight sides. Changing **ELEMENT** destroys the mesh and any solution in the program.

The **FIELD** type controls the setting of the system variable names and several constants used in post processing. The value of **FIELD** can be changed without affecting any solution in the program.

Field Symmetry Options

The field symmetry options allow fields to be calculated in symmetry copies of the finite element mesh. The symmetry copies of the mesh are defined by reflections in the major coordinate axes and by rotations around the global origin. The reflections are defined by parameters **REFXAXIS** and **REFYAXIS** which affect the reflections in the horizontal and vertical axes of the displayed geometry. The parameter values select which component of magnetic field changes sign across the axis, i.e. has a zero value on the axis:

- **REFX=INVERSE**: the tangential field is zero.
- **REFX=YES**: the normal field is zero.
- **REFX=NO**: there is no reflected copy of the mesh.

Rotational copies of the mesh are specified by the number of copies (**ROTATIONS**) and their sign (**ROTSIGN**). If the direction of the currents reverses in alternate copies **ROTSIGN** should be set to **NEGATIVE**.

Results from Stress or Thermal analysis or other tables can also be calculated in the symmetry copies of the mesh. However, their signs remain the same as in the mesh itself.

Integral Field Options

By default, the electromagnetic fields (**B** and **H**) are calculated by the post processing routines by interpolation of nodal values, using the finite element shape functions. These nodal values are calculated by averaging the appropriate derivatives of potential from the elements surrounding each node.

Electromagnetic fields can be also calculated at any point in space by integration of the current densities and magnetisations in the current carrying and permeable elements of the mesh. This is selected using

SET +INTEGRAL, FLDTYPE=TOTAL

In order to obtain the correct values it is necessary to specify the symmetry copies of the mesh using the parameters **REFXAXIS**, **REFYAXIS**, **ROTATIONS** and **ROTSIGN**, page 3-178.

If the finite element solution includes the effect of a uniform external field implied by the boundary conditions, this can be added to the calculated field by setting non-zero values to **HXX** and **HYY** as appropriate.

It is also possible to calculate the fields from the specified current densities in the mesh, ignoring induced currents and magnetisations. This is selected using **FLDTYPE=COILONLY**.

Only **B** and **H** are calculated by integration. When **+INTEGRAL** is selected, other field quantities are calculated by interpolation.

The **SOLVE** Command

Menu Route:

FILE↓
Write file → Analysis data

Command line Parameters:

Command	SOLVE	
Parameter	Default	Function
TYPE	<i>none</i>	Module to be used for the analysis of the model
		AC Steady-state AC analysis
		LM Linear Motion analysis
		RM Rotating Machine analysis
		SA Stress analysis
		SP Space charge analysis
		ST Static analysis
		TH Thermal analysis
		THTR Thermal transient analysis
		TR Transient time varying analysis
		VL Velocity analysis

The **SOLVE** command has a single parameter for the type of analysis module that is to be used for calculation of the field solution. This command must be executed before writing the analysis file, and the correct solver **TYPE** must be specified.

Other options are available as sub-commands within the **SOLVE** command, allowing various parameters to be set for the analysis program.

SOLVE Keyboard Sub-commands	
Sub-command	Function
CASE	Give case data, either scale factors, frequencies or output times, for the analysis program
DATA	Give data specific to the options available in the analysis programs
DRIVE	Select the drive functions to be used by the drives in transient, linear motion and rotating machine analysis
LOGFILE	Create an additional output file from TR/LM/RM.
QUIT	Quit the SOLVE command

CASE Sub-command

The **CASE** sub-command allows multiple cases to be set up and solved in a single analysis file. The different modules have different case types available.

Modules	Case type
ST/VL/SP	Scale factor used to multiply all drive sources (current density, non-zero potential boundaries)
AC	Frequencies for which the steady state harmonic analysis will be run
TR/LM/RM/THTR	Output times at which a solution will be stored
SA/TH	Multiple cases are not available

The solution file can contain multiple solutions (one for each case value set) that can be read in by setting the **CASE** parameter of the **READ** command.

Command line Parameters:

Sub-command	CASE	
Parameter	Default	Description
COMMAND	ADD	Command option for generating the list of cases
		ADD Adds a new case value to the end of the list of cases
		INSERT Inserts a new case at the position given
		REPLACE Replaces the case value at the position given
		DELETE Deletes the case value from the position specified
	LIST	Lists the current case list
POSITION	<i>none</i>	Sets the position in the case list for inserting, replacing or deleting values
VALUE	<i>none</i>	Sets the value for adding, inserting or replacing

For output times (TR, LM and RM), all values in the list are ordered in numerical order, so inserting and adding may have the same effect. Any duplicate values are removed from the list. If adaption is being used, the adaption will happen on the first case in the list only. All subsequent cases will be solved using this mesh.

DATA Sub-command

Command line Parameters:

Sub-command	DATA			
Parameter	Default	Modules	Function	
LINEAR	YES	ST/ AC/ TR/ VL/ LM/ RM	Use linear analysis	
			NO	Use non-linear analysis
			YES	Use linear analysis
NITERATIONS	21	ST/ AC/ TR/ VL/ LM/ RM	Maximum number of non-linear iterations that will be used	
TOLERANCE	0.001	ST/ AC/ TR/ VL/ LM/ RM	Tolerance for convergence of the non-linear iterations	
ITTYPE	NEWTON	ST/ VL	Iteration type to be used for non-linear updates	
			NEWTON	Use Newton-Raphson non-linear updates
			SIMPLE	Use simple iteration updates
CMU	NO	AC	Use complex permeability	
MUTYPE	DC	AC	Use a DC or Incremental permeability	
			DC	Use a DC permeability
			INCREMENT	Use an incremental permeability

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
VELTYPE	LINEAR	VL	Use a linear or rotational velocity (see comment on units below)	
			LINEAR	Region velocity is linearly directed along the Y or Z axis
			ROTATION	Region velocity is interpreted as a rotational velocity around the origin
UPWINDING	NONE	VL	Selects the type of upwinding to be used	
			NONE	Do not use any upwinding
			HUGHES	Use Hughes upwinding (not available in axi-symmetric problems)
			STREAM	Use streamline upwinding
ADELEMENTS	*	ST/ AC/ VL	Maximum number of elements that will be used in the model when using adaptive refinement	
ADITERATIONS	0	ST/ AC/ VL	Maximum number of refinement iterations that will be used	
			0	Do not use any adaptive mesh refinement
ADACCUARCY	5	ST/ AC/ VL	Final accuracy at which adaptive refinement will stop	
TSTEP	0	TR/ LM/ RM	Fixed time step	

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
TSTOLERANCE	0.001	TR/LM/ RM	Tolerance for adaptive time stepping	
TSTYPE	ADAPT	TR/LM/ RM	Selects the type of time stepping algorithm to be used	
			FIXED	Uses fixed time steps (TSTEP)
			ADAPT	Adapts the time steps until the tolerance is reached
RMVELOCITY	3000	RM	Rotational velocity (RPM) for the rotating machines module	
SPRELAXATION	0.001	SP	Initial under-relaxation factor	
SPITERATIONS	30	SP	Maximum number of iterations that will be used	
SPTOLERANCE	0.001	SP	Tolerance for convergence of the space charge iterations	
RESTART	NO	ST/ AC/ TR/ VL/ LM/ RM/ SP	Restart from the current solution	
			NO	Use non-linear analysis
			YES	Use linear analysis
SATYPE	STRESS	SA	Selects the type of stress analysis to be used	
			STRESS	Use plain stress
			STRAIN	Use plain strain
			AXISYMM	Use axisymmetric stratified
TSINITIAL	0	LM/ RM/ TR	Initial time step for adaptive timestepping.	

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
MOTIONTYPE	FIXED	LM/ RM	LM/ RM velocity:	
			COUPLED	Velocity calculated using mechanical coupling.
			FIXED	Constant velocity (RM only)
			VARIABLE	Velocity calculated from expression in a command input file
MOTIONCOMI		LM/RM	Name of command input file for COUPLED or VARIABLE velocity	
INERTIA	1	RM	Moment of inertia of rotor in COUPLED velocity.	
FTORQUE	0	RM	Frictional torque applied to rotor in COUPLED velocity.	
ATORQUE	0	RM	Torque applied to the rotor in COUPLED velocity	
VTORQUE	0	RM	Speed varying torque coefficient in COUPLED velocity	
LENGTH	0	RM	Magnetic length of the rotor.	

The **DATA** sub-command can be used to specify any options required by the chosen analysis program. The options that must be set will vary depending upon the analysis program.

RM Options

The Rotating Machine Analysis program can be used with **FIXED**, **COUPLED** or **VARIABLE** velocity:

- **FIXED**: the value of **RMVELOCITY** is used for all time.
- simple **COUPLED**: the velocity is calculated by the program from the forces in the electromagnetic fields and the following mechanical properties of the rotor:
 - the **INERTIA**
 - the frictional torque (**FTORQUE**)

- an additional applied torque (**ATORQUE**)
- a speed varying torque (**VTORQUE***speed)
- the **LENGTH** of the rotor.
- complex **COUPLED**: More complex mechanical systems can be modelled using a command input file given by the **MOTIONCOMI** parameter. If the name of file is given, the file should calculate **#ACCEL**, the acceleration of the rotor from the system variables:
 - **RMTORQUE**: the torque per unit length from the electromagnetic fields.
 - **RMANGLE**: the position of the rotor (radians).
 - **RMSPEED**: the rotation speed (radians/second).
 - **TTIME**: the time (seconds).
 - **RMLENGTH**: the device length.
 - **RMINERTIA**: the moment of inertia.
 - **RMFTORQUE**: the frictional torque.
 - **RMVTORQUE**: the speed varying torque.
 - **RMATORQUE**: the applied torque.
 - the total torque
 - = $RMTORQUE * RMLENGTH + RMSPEED * RMVTORQUE + RMATORQUE - RMFTORQUE$
 - with the sign of **RMFTORQUE** set so that it opposes motion.

A command input file equivalent to the default mechanical system would contain:

```
$CONSTANT #TORQUE1 RMTORQUE*RMLENGTH
$CONSTANT #TORQUE2 RMSPEED*RMVTORQUE
$CONSTANT #TORQUE3 RMATORQUE-RMFTORQUE
$CONSTANT #TORQUE #TORQUE1+#TORQUE2+#TORQUE3
$CONSTANT #ACCEL #TORQUE/RMINERTIA
```

A more complex mechanical system could be modelled using a command input file containing:

```
$CONSTANT #TORQUE1 RMTORQUE*RMLENGTH
$CONSTANT #TORQUE2 RMSPEED*RMVTORQUE
$CONSTANT #TORQUE3 RMATORQUE-RMFTORQUE
$IF RMSPEED LE 3600/60*2*PI
  $CONSTANT #TORQUE #TORQUE1+#TORQUE2+#TORQUE3
$ELSE
  $CONSTANT #TORQUE #TORQUE1-RMFTORQUE
$END IF
$CONSTANT #ACCEL #TORQUE/RMINERTIA
```

- **VARIABLE**: the speed of the rotor is calculated using a command input file given by **MOTIONCOMI** which specifies the speed as a function of time, in radians/second, e.g.

```
$CONSTANT #SPEED 100*TTIME
```

Full details of the RM facilities are given in the User Guide.

LM Options

The Linear Motion Analysis program can be used with coupled or variable velocity:

- **COUPLED**: Complex mechanical systems can be modelled using a command input file given by the **MOTIONCOMI** parameter. The file should calculate **#ACCELX**, **#ACCELY** and **#ROTACCEL**, the acceleration of the rotor from the system variables:
 - **LMXFORCE**, **LMYFORCE**, **LMZFORCE**, **LMTORQUE**: the force and torque from the electromagnetic fields.
 - **LMXSHIFT**, **LMYSHIFT**, **LMZSHIFT**, **LMROTANGLE**: the position of the moving part.
 - **LMXSPEED**, **LMYSPEED**, **LMROTSPEED**: the speeds.
 - **TTIME**: the time (seconds).
- **VARIABLE**: the speed of the moving part is calculated using a command input file given by **MOTIONCOMI** which specifies the speed as a function of time, e.g.


```
$constant #speedx 100/ttime
$constant #speedy 0
$constant #rotspeed 0
```

Full details of the LM facilities are given in the User Guide.

DRIVE Sub-command

The **DRIVE** sub-command is for use only with the transient, linear motion and rotating machine analysis modules, and can be used to set the list of drive functions applied to the current sources and non-zero potential boundaries.

*Command line
Parameters:*

Sub-Command	DRIVE		
Parameter	Default	Description	
NUMBER	<i>none</i>	Number of the drive function to be set	
		LIST	Option to output the list of drive numbers that can be set
TYPE	<i>none</i>	Type of drive that will be used for the selected drive number	
		DC	DC drive, $f(t) = 1$
		STEP	Step at $t=0$ $f(t) = 0: t < 0$ $f(t) = 1: t \geq 0$
		RAMP	Ramps from $f(t)=0$ at $t=0$ to $f(t)=1$ at $t=val1$
		COSINE	Cosine function of frequency $val1$, with phase $val2$
		SINE	Sine function of frequency $val1$, with phase $val2$
		EXPONENTIAL	Exponential increase with time constant given by $val1$
		PEAK	Function generating a peak at time given by $val1$
		TTON	Use a timetable of points with $f(t)=0$ at $t < 0$
TTOFF	Use a timetable of points with $f(t)=f(0)$ at $t < 0$ $f(0)$ is the value of the function in the table file at time $t=0$.		
VAL1	<i>none</i>	First value for the different functions	
VAL2	<i>none</i>	Second value for the functions (only used by COSINE and SINE)	
FILE	<i>none</i>	File name for use with timetable drive types TTON and TTOFF	

The program generates a list of all conductor labels that have been defined, together with any external circuits that have been created, and assigns each of these a number. This list can be seen by using the command **DRIVE NUMBER=LIST**

The drive type for any of these can be set by the **DRIVE** command, together with the correct parameters. A default drive is also available. This is used to set any

drive that has conductor number 0, as well as any of the drive labels that has no drive function specified.

Time-table files should be created by an editor. They consist of 2 columns of numbers in free-format, the first giving values of time and the second of function value. Consecutive lines with the same value of time introduce a discontinuity of function value or time-derivative into the cubic-spline fitting. The data in time-table files can be displayed using the pre and post processor **GRAPH** command

LOGFILE Sub-command

Command line Parameters:

Sub-Command	LOGFILE		
Parameter	Default	Description	
FILE	<i>none</i>	Name of log file.	
COL1	TTIME	Expression for column 1 of log file.	
COL2	RMANGLE	Expression for column 2 of log file.	
COL3	<i>none</i>	Expression for column 3 of log file.	
COL4	<i>none</i>	Expression for column 4 of log file.	
COL5	<i>none</i>	Expression for column 5 of log file.	
COL6	<i>none</i>	Expression for column 6 of log file.	
COL7	<i>none</i>	Expression for column 7 of log file.	
COL8	<i>none</i>	Expression for column 8 of log file.	
LOGGING	NO	Log file options:	
		NO	No log file
		SOLUTION	Logging to file with name derived from solution file name.
		YES	Logging to FILE

The **LOGFILE** sub-command is for use only with the transient, linear motion and rotating machine analysis modules, and can be used to request an additional output file which tabulates the progress of the analysis. The command specifies the name of the file and up to 8 system variables or expressions which will be evaluated and added to the file at every time step.

The system variables available are those which can be used in coupled velocity command files (See “RM Options” on page 3-185. See “LM Options” on page 3-187.).

QUIT Sub-command

This command has no parameters and causes the program to leave the **SOLVE** command.

The SYMMETRY Command

Menu Route:

MODEL↓
Periodic Symm.

Command Line Parameters:

Command	SYMMETRY	
Parameter	Default	Function
ANGLE	none	Rotation angle.
DX	none	X displacement.
DY	none	Y displacement.
CONNECTION	POSITIVE	Connection between master and slave:
	POSITIVE	Master potential = slave potential.
	NEGATIVE	Master potential = – slave potential.
DELETE	NO	Delete old connections:
	NO	Keep old connections.
	YES	Delete old connections.
LIST	YES	List connections:
	NO	No list.
	YES	List connections.

The SYMMETRY command connects together region faces which have the periodicity boundary condition SYMMETRY. It can be used in two ways: keyboard or cursor. Faces of regions with replications cannot be in symmetry pairs.

Keyboard mode is used if values are given for any of the parameters DX, DY and ANGLE. In keyboard mode, parameters are used to specify a transformation consisting of a displacement (DX, DY) and a rotation (ANGLE) which are applied to each region face with boundary condition SYMMETRY. If the transformed face matches an untransformed SYMMETRY face in position, curvature and subdivision, the two faces are CONNECTED so that the slave potential is the same as (POSITIVE) or opposite in sign to (NEGATIVE) the master.

Any SYMMETRY pairs existing before the command is issued can be deleted with +DELETE and the pairs connected at the end of the command can be listed with +LIST.

Cursor mode is used if no values are given for any of the parameters **DX**, **DY** and **ANGLE**. In cursor mode, pairs of region faces are selected using the cursor (see page 3-152). The first set of cursor commands allows the selection of the master face; a second set allows one end of the face to be selected. The two sets of cursor commands are used a second time to select the slave face and one end of it. Finally the connection, **POSITIVE** or **NEGATIVE** must be given.

SYMMETRY Face Selection Cursor Hits

The following cursor hit characters can be used to select master or slave faces:

SYMMETRY Command Cursor Hits 1	
Cursor hit	Function
D	Delete the closest face from the list of symmetry pairs. The selection must be confirmed with YES , NO or ALL . (Only available while selecting a master face.)
F	Select the closest face of region containing the cursor as a master. The program prompts for first corner and slave selection cursor hits.
H	Display help message explaining all the cursor options.
L	List the current symmetry pairs. (Only available while selecting a master face.)
Q	Leave SYMMETRY command (while selecting a master face). Forget master face (while selecting a slave face).
Z	Execute the ZOOM (see page 3-212) command to change size of the picture. The program prompts for ZOOM command cursor hits.

SYMMETRY First Node Selection Cursor Hits

The following cursor hit characters can be used to select the first node of a master or slave face:

SYMMETRY Command Cursor Hits 2	
Cursor hit	Function
<space>	Select the nearest end node of the face.
H	Display help message explaining all the cursor options.
Q	Reject previously selected face.

The **TEST** Command

Menu Routes:

MODEL↓
 Draw region → Test new region
 Mesh generator → Check region matching

Command Line Parameters:

Command	TEST			
Parameter	Default	Function		
REG1	<i>none</i>	First region to be tested.		
REG2	<i>none</i>	Last region to be tested. * for highest region number.		
POINTS	BOTH	Where should points be added?		
		BOTH	In both regions.	
		NONE	In neither region.	
		OTHER	In other regions.	
		SELF	In region being tested.	
		QUERY	YES	Ask user first?
		NO	Points added without asking.	
		YES	User is asked before regions are changed.	
SIDES	OVER	Which region's sides should be changed?		
		NO	Neither region.	
		OVER	Change other region.	
		UNDER	Change this region.	
MATCH	YES	Curvature matching switch:		
		NO	Curvatures need not match.	
		YES	Curvatures must match.	
DRAW	YES	Apply test during DRAW command switch:		
		NO	Do not test new regions.	
		YES	Test each region as it is drawn.	

The **TEST** command compares each region from **REG1** to **REG2** against all other regions in that range and optionally adds extra points and adjusts subdivisions and curvatures so that the regions match. The **+DRAW** option allows the **TEST** command to operate automatically on each region as it is drawn. The **TEST** command can be used explicitly to correct errors which might have occurred if the region topology has been changed using **COPY**, **ERASE** or **MODIFY**.

Inserting Points

If two regions have sides which are collinear but do not share the same end points, extra points can be added into the regions so that the sides match exactly. It is usually best to allow the program to add points into whichever region it chooses (**POINT=BOTH**). However in some circumstances it might be better to restrict the program to adding points into either the region being tested (**POINT=SELF**) or the region it is being compared with (**POINT=OTHER**). If the program finds that an extra point is required, the user can accept or reject it if **QUERY=YES**.

The point adding tests can be omitted by setting **POINT=NONE**.

Side Matching

If two regions have sides which share the same end points, the subdivisions and curvatures of one can be copied to the other. If while testing a region, there is a mis-match with a neighbouring regions and **SIDES=OVER**, the neighbouring region is changed; if **SIDES=UNDER**, the region being tested is changed. If **MATCH=YES** the curvatures as well as the end points must match, and only the subdivisions will change.

The side matching tests can be omitted with **SIDES=NO**.

Boundary Conditions

The **TEST** command also checks boundary conditions on adjoining faces of regions. Any discrepancies are reported (e.g. two different values of potential). These errors must be corrected by modifying the boundary conditions.

The **TITLE** Command

Menu Route:

OPTIONS ↓
Title

Command Line Parameters:

Command	TITLE	
Parameter	Default	Function
STRING	<i>none</i>	A graphics window title.
POSITION	TL	Graphics window title position:
		BC Bottom centre.
		BL Bottom left.
		BR Bottom right.
		TC Top centre.
		TR Top right.
KEEP	YES	Title preservation switch:
		NO Only display title once.
		YES Display title on subsequent pictures as well.
NOW	YES	First appearance switch:
		NO Display after next graphics window clear.
		YES Display immediately.
DATE	YES	Date, time and page number switch:
		NO Date, time and page number not displayed.
		YES Date, time and page number displayed.
EXTRA	YES	Display additional model information (if available):
		NO Additional model information not displayed.
		YES Additional model information displayed.

The **TITLE** command controls the display of titles and other information on the graphics window. The items which can be controlled are an additional title, and the default labelling of the graphics window with date, time and page number.

- an additional title, **STRING**, can be **POSITION**ed at any one of 6 places on the graphics window. It can be displayed immediately (**+NOW**) or after the next graphics window clear (**-NOW**). It can be used once (**-KEEP**) or kept for sub-

sequent pictures (**+KEEP**). **STRING**s which contain spaces or commas or start with a non-alphabetic character must be enclosed in quotation marks (').

- the display of the **DATE**, time and page number can be switched on or off on subsequent pictures with **+DATE** and **-DATE**.
- additional information about the model (if available) can be switched on or off on subsequent pictures with **+EXTRA** and **-EXTRA**.

The TRACK Command

Menu Route: **FIELDS**↓
 Trajectories → Calculate

Command Line Parameters:

Command	TRACK		
Parameter	Default	Function	
XORR	0	Initial coordinate of the particle.	
		X in SYMMETRY=XY problems.	
		R in SYMMETRY=AXISYMMETRY problems.	
YORT	0	Initial coordinate of the particle.	
		Y in SYMMETRY=XY problems.	
		ϑ in SYMMETRY=AXISYMMETRY problems.	
Z0	0	Initial Z-coordinate of the particle.	
THETA	0	ϑ Euler angle defining particle direction.	
PHI	0	Φ Euler angle defining particle direction.	
PSI	0	Ψ Euler angle defining particle direction.	
VOLTS	1	Acceleration voltage or initial energy.	
CHARGES	-1	CHARGE on the particle in elementary charge units. -1 is the charge on an electron.	
MASS	1	Particle rest MASS in electron rest mass units.	
STEP	1	STEP length between output points.	
NSTEP	100	Number of steps to be calculated.	
TOLERANCE	0.01	Accuracy required.	
OPTION	RAY	Options:	
		DIPOLE	Use DIPOLE force on magnetic particle.
		FLUX	Follow a FLUX line.
		RAY	Use electromagnetic forces on charged particle.
		TEST	A TEST pattern of charged particles defined by PATTERN and LINE .
PATTERN	1.E-4	Size of the rectangular grid test pattern.	
LINES	5	Number of X and Y lines in test pattern.	
FILE	<i>none</i>	Name of the TRACK file.	

Command	TRACK (continued)	
Parameter	Default	Function
STATUS	NEW	STATUS of the TRACK file.
		NEW Create a NEW file.
		OLD Append data to an OLD file.
		CLEAR Overwrite data in an OLD file.

The **TRACK** command calculates the trajectories through the field. There are four options:

- A single particle (**OPTION=RAY**) subject to magnetic forces,

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B} \quad (3.17)$$

or electric forces

$$\mathbf{F} = q\mathbf{E} \quad (3.18)$$

- A **TEST** pattern of many rays.
- A single unsaturated spheroidal particle subject to magnetic **DIPOLE** forces,

$$\mathbf{F} = \mathbf{M} \cdot \nabla \mathbf{H} \quad (3.19)$$

The parameter **CHARGES** should be set to $10^{-8}\mu v$, where μ is relative permeability of the particle and v its volume in cm^3 .

- A magnetic or electric **FLUX** line.

The charged particle trajectory calculations include full relativistic correction.

Each option starts at the point **X0, Y0, Z0**. Particles have an initial velocity in the direction defined by the Euler angles **THETA, PHI** and **PSI** (see [page 2-39](#)).

The initial coordinates and direction are defined in a 3 dimensional **XYZ** or **RθZ** coordinate system. In **XY** problems, $Z=0$ is the plane of the screen; in **AXISYMMETRY** $\vartheta=0$ is the plane of the screen.

The initial energy (in electron **VOLTS**), number of **CHARGES** and **MASS** of the particles can be set.

The **TEST** pattern option allows a beam of particles to be fired from the plane normal to the initial particle direction, from the intersection points of a square orthogonal grid.

The calculation is controlled by the **STEP** length along the trajectory, the number of steps (**NSTEP**) and the relative **TOLERANCE** which applies to coordinates and velocities. **STEP** only determines the distance between the displayed points

on the trajectory, the **TOLERANCE** is achieved by adaptive integration. **STEP** is measured along the trajectory and hence the total trajectory length calculated is **STEP*NSTEP**.

The trajectory coordinates are stored in a binary file. When the file status is set to **NEW**, a new file will be created to contain the data. Setting status to **OLD** causes the trajectories to be appended to an existing file. The third option **CLEAR** causes an old file to be overwritten. The format of the track file is described in the section on **TRACK** Files below.

The trajectories are drawn on the current display as they are calculated. Subsequently the contents of the **TRACK** files can be **VIEWED** in a number of ways (see [page 3-207](#)).

TRACK Files

The **TRACK** command stores the trajectory coordinates in a binary unformatted file which can be read by the **VIEW** command (see [page 3-207](#)). Track files have file name extension *track*.

There is no indication of the number of tracks in the file; the data must be read with end and error control to detect when all data has been read.

The contents of the file is as follows:

Record 1: 20 **INTEGERS**. In track files created by the post processor, only the first value is important. It contains the number of points on the track, **NSTEP**.

Record 2: 20 **DOUBLE PRECISION** values, **RTRAK**.

RTRAK(1) is the current associated with the track.

RTRAK(2) is the mass.

RTRAK(3) is the charge.

RTRAK(4) is the step-length

RTRAK(5-20) can be ignored.

Record 3: **NSTEP** **DOUBLE PRECISION** values, x coordinates of the track points.

Record 4: **NSTEP** **DOUBLE PRECISION** values, y coordinates of the track points.

Record 5: **NSTEP** **DOUBLE PRECISION** values, z coordinates of the track points.

Record 6: **NSTEP** **DOUBLE PRECISION** values, x-component of the velocity.

Record 7: `NSTEP DOUBLE PRECISION` values, y-component of the velocity.

Record 8: `NSTEP DOUBLE PRECISION` values, z-component of the velocity.

On UNIX systems, if `NSTEP` is greater than 500, records 3 to 8 are blocked into sub-records which contain 500 `DOUBLE PRECISION` words.

The UNDO Command

Menu Routes:

MODEL↓
Modify regions → Undo modify

MODEL↓
Undo erase

Command Line

Parameters:

Command	UNDO
No Parameters	

Before each **EDIT**, **ERASE** and **MODIFY** command (see [page 3-76](#), [page 3-86](#) and [page 3-149](#)), a copy of the current state of the regions is saved in memory. The **UNDO** command reinstates this copy. The **UNDO** should immediately follow the **EDIT**, **ERASE** or **MODIFY** since other commands will invalidate the copy.

The **UNITS** Command

Menu Route: **UNITS**

*Command Line
Parameters:*

Command	UNITS		
Parameter	Default	Function	
LENGTH	METRE	Unit for length:	
		CM	centimetre
		INCH	inch
		METRE	metre
		MICRON	micron
		MM	millimetre
FLUX	TESLA	Unit for flux density:	
		CM2	coulomb metre ⁻² (electrostatics)
		KGAUSS	kilogauss
		TESLA	tesla
FIELD	AM	Unit for field strength:	
		AM	ampere metre ⁻¹
		OERSTED	oersted
		VCM	volt centimetre ⁻¹ (electrostatics)
		VM	volt metre ⁻¹ (electrostatics)
		VMM	volt millimetre ⁻¹ (electrostatics)
POTENTIAL	WBM	Unit for scalar potential:	
		AMPERE	ampere
		GCM	gauss centimetre
		OCM	oersted centimetre
		VOLT	volt (electrostatics)
		WBM	weber metre ⁻¹

Command	UNITS (continued)		
Parameter	Default	Function	
CONDUCTIVITY	SM	Unit for conductivity:	
		SCM	siemen centimetre ⁻¹
		SM	siemen metre ⁻¹
		SMM	siemen millimetre ⁻¹
		SMU	siemen micron ⁻¹
DENSITY	AM2	Unit for current/charge density:	
		ACM2	ampere centimetre ⁻²
		AIN2	ampere inch ⁻²
		AM2	ampere metre ⁻²
		AMM2	ampere millimetre ⁻²
		AMU2	ampere micron ⁻²
		CCM3	coulomb centimetre ⁻³
MCM3	micro coulomb metre ⁻³		
POWER	WATT	Unit for power:	
		ERGS	erg second
		HP	horse power
		WATT	watt
FORCE	NEWTON	Unit for force:	
		DYNE	dyne
		GRAMME	gramme force
		KGF	kilogramme force
		LBF	pound force
		NEWTON	newton
ENERGY	JOULE	Unit for energy:	
		BTU	British Thermal Unit
		ERG	erg
		CALORIE	calorie
		JOULE	joule
MASS	KG	Unit for mass:	
		GRAMME	gramme
		KG	kilogramme
		LB	pound

The **UNITS** command sets the units to be used to interpret user input and display geometric and field data. Each of the parameters can be set to one of a set of pre-defined character strings corresponding to commonly used units.

The default set of units is *S.I.* Note that if the result of a post processing calculation gives a numerical quantity whose units are the product of these basic quantities, any changes made to the units will also be reflected in the compound unit. For example, if the user has set the length unit to **MM** but left the force in **NEWTON**, torque values computed by the **INTLINE** command will be in *N-mm*, not *N-m*.

The current units are stored in OPERA-2d data and results files, and are reinstated when a file is **READ** (see page 3-168). The **UNITS** command is also included in a **COMOUTPUT** file (see page 3-33). The parameter values can be changed before reading the file back into the program with **\$ COMINPUT** (see page 2-30) in order to scale the data.

The **VIEW** command does not use the **DENSITY** unit but has its own unit for the display of current density: it uses amp **LENGTH**⁻²(page 3-210).

The VECTOR Command

Menu Route:

FIELDS↓
Vectors

Command Line Parameters:

Command	VECTOR		
Parameter	Default	Function	
X_n	0	X or R coordinate of n th point defining patch. Points 1 to 4 are corners; points 5 to 8 are mid-side points.	
Y_n	0	Y or Z coordinate of n th point defining patch. Points 1 to 4 are corners; points 5 to 8 are mid-side points.	
CORNERS	4	Number of points defining patch: 4 or 8.	
N1	10	Number of points between corners 1 and 2, and between corners 3 and 4.	
N2	10	Number of points between corners 1 and 4 and between corners 2 and 3.	
VX	BX	Expression for x-component of vectors.	
VY	BY	Expression for y-component of vectors.	
TIME	0	For Steady-state ac solutions only: time at which basic field quantities are evaluated.	
		<i>number</i>	Angle in degrees around ac cycle.
		AMPLITUDE	Amplitude.
		PHASE	Phase angle.
		TAVERAGE	Time average.
ERASE	NO	Picture erase switch:	
		NO	Old picture not erased.
		YES	Old picture erased.
VECTOR	YES	Vector display switch:	
		NO	No vector display.
		YES	Vectors displayed.
PRINT	NO	Printed values switch:	
		NO	No printed values.
		YES	Values printed to file <i>opera2.lp</i> .

Command	VECTOR		
Parameter	Default	Function	
LABEL	YES	Vector component labelling switch:	
		NO	No labels.
		YES	Component expressions shown.

The **VECTOR** command evaluates field quantities on 4 or 8-noded patches. The results can be displayed as vectors at each field point and the values can be printed in the file *Opera2d_PP_n.lp*.

The patch is specified by the points (**X1,Y1**, ..., **X8,Y8**). The mid-side points are ignored if **CORNERS=4**, and included if **CORNERS=8**. Point 5 is on side 1 (point 1 to point 2); point 6 is on side 2 (point 2 to point 3); point 7 is on side 3 (point 3 to point 4); point 8 is on side 4 (point 4 to point 1).

It does not matter if the patch extends beyond the limits of the mesh.

The field quantities are evaluated at **N1*N2** points. The coordinates of points other than the corner and mid-side points are found by interpolation using 4 or 8-noded isoparametric two-dimensional finite element shape functions.

Expressions for the vector components, **VX** and **VY** can use as variables any of the system variables given in [page 3-12](#). For Steady-state ac solutions, each system variable in an expression is evaluated at the **TIME** specified. The size of the vectors in current length units gives the magnitude of the vector quantity (**VX,VY**). Vectors can be scaled by using a scaling factor in the expressions for **VX** and **VY**. The system variable **MAXIMUM** is set to the length of the longest vector.

The other parameters are switches: **±VECTOR** controls whether vectors are displayed, and if they are **±LABEL** controls whether the expressions are displayed on the graphics window; **±PRINT** controls whether the component values are written to the file *opera2.lp*.

The **VIEW** Command

Menu Route:

FIELDS↓
Trajectories → Display

Command Line Parameters:

Command	VIEW	
Parameter	Default	Function
FILE	<i>none</i>	Name of file containing trajectories.
PLOT	DISPLAY	Type of output required.
		DISPLAY DISPLAY trajectories on the current view of the model.
		INTERSECT Draw graphs of the intersections of the trajectories with a line.
		TRACKS Draw graphs of the trajectories.
X1	0.0	X or R coordinate of the start of the intersection line.
Y1	0.0	Y or Z coordinate of the start of the intersection line.
X2	0.0	X or R coordinate of the end of the intersection line.
Y2	0.0	Y or Z coordinate of the end of the intersection line.
CURVATURE	0.0	Curvature of the intersection line.
XAXIS	Y	The variable plotted on the horizontal axis of the graph.
YAXIS	X	The variable plotted on the vertical axis of the graph.
XMIN	*	Lower limit for graph horizontal axis. (* for automatic setting of limit.)
XMAX	*	Upper limit for graph horizontal axis. (* for automatic setting of limit.)
YMIN	*	Lower limit for graph vertical axis. (* for automatic setting of limit.)

Command	VIEW (continued)		
Parameter	Default	Function	
YMAX	*	Upper limit for graph vertical axis. (* for automatic setting of limit.)	
LINE	YES	Graph plotting style:	
		NO	Plot using symbols.
		YES	Plot using lines.
COLOUR	YES	Use of colour in the displays:	
		FUNCTION	Colours represent the values of COMPONENT .
		NO	Use text colour.
		YES	Colours represent trajectory numbers.
COMPONENT	POT	Expression used to assign colours to the trajectories when COLOUR=FUNCTION . The value is also printed when printing is enabled.	
PRINT	NO	Printing switch:	
		NO	No printing.
		YES	All points or intersections printed to dialogue file.
SAMPLE	8	Sample size used to calculate current density for PLOT=INTERSECTIONS .	
ERASE	YES	Erase previous display (only for PLOT=INTESECTIONS and PLOT=TRACKS)	
		NO	Add to existing display.
		YES	Erase display first.

The **VIEW** command re-displays and processes trajectories calculated by the **TRACK** command (page 3-197), or the Space Charge analysis program, and stored in **TRACK** files (binary data files as described in the page 3-199).

Types of **VIEW**

- **PLOT=DISPLAY**: this option allows the trajectories to be displayed on the current view of the model using the text colour or multiple colours (page 3-209).
- **PLOT=TRACKS**: this option displays graphs of the trajectories. The variables plotted on the axes of the graphs can be selected from the set of active sys-

tem variables (page 3-209). For example to display axisymmetric results in the ZX plane, with Z on the horizontal axis of the graph, use

VIEW XAXIS=Z, YAXIS=X

By default, the axis limits are automatically set to contain the functions plotted but limits can be specified.

- **PLOT=INTERSECTIONS**: this option displays graphs of the intersections of the trajectories with a line which can be curved. The line is specified by its start (X1, Y1) and end points (X2, Y2) and **CURVATURE**. The intersections are plotted as graphs using the set of active system variables (page 3-209). When the intersections are calculated, the current density is also computed for each intersection point. This is evaluated statistically using varying sample sizes up to a maximum given by the **SAMPLE** parameter. The expected error is also calculated and will be displayed using error bars if **YAXIS=J** is selected. The sample size should be less than $\frac{1}{10}$ of the number of intersections. If larger values are used the results will be smooth but the errors will be larger.

Viewing in COLOUR

In each of the **PLOTTING** options, the lines can be displayed using the text colour (**COLOUR=NO**), with colours representing the trajectory number (**COLOUR=YES**) or with colours representing some characteristic of the position along the trajectory (**COLOUR=FUNCTION**).

Functional colours are selected using the value of the **COMPONENT** expression. **COMPONENT** can be assigned to expressions in terms of the active system variables listed below. For example,

COLOUR=FUNCTION, COMPONENT=TOF

will display the graphs coloured according to the time of flight from the start points.

Active System Variables

The **VIEW** command has its own set of system variables which are available in expressions for the graph plotting variables (**XAXIS** and **YAXIS**) and the **COMPONENT** used to determine the colours of the lines.

System Variables for PLOT=DISPLAY and PLOT=TRACKS	
X	X coordinates of points on the trajectory
Y	Y coordinates of points on the trajectory

System Variables for PLOT=DISPLAY and PLOT=TRACKS	
Z	Z coordinates of points on the trajectory
R	Radial coordinate of points on trajectory (axisymmetry only)
VELX	X component of particle velocity
VELY	Y component of particle velocity
VELZ	Z component of particle velocity
CURRENT	The current in the track (beamlet).
TOF	Time of flight from the start of the trajectory.
Q	Electronic charge on the particle.
M	Mass of the particle in electron units.

System Variables for PLOT=INTERSECTIONS	
X	X coordinates of the intersection points
Y	Y coordinates of the intersection points
Z	Z coordinates of the intersection points
J	The calculated current density in the beam defined by the set of tracks in the track file (units are amp length_unit ⁻²)
VELX	X component of particle velocity
VELY	Y component of particle velocity
VELZ	Z component of particle velocity
CURRENT	The current in the track (beamlet).
TOF	Time of flight from the start of the trajectory.
Q	Electronic charge on the particle.
M	Mass of the particle in electron units.

Printing Trajectory Data

When printing is selected (**PRINT=YES**) the following data is output to the dialogue file, *Opera2d_PP_n.lp*:

- **PLOT=DISPLAY** and **PLOT=TRACKS**, for all points in all trajectories:
X Y Z VELX VELY VELZ COMPONENT
- **PLOT=INTERSECTIONS**, for all intersections:
CURRENT X Y Z VELX VELY VELZ COMPONENT

N.B. The unit of current density is amp length_unit⁻².

The **WRITE** Command

Menu Route:

FILE↓
Write file → Write model

Command Line Parameters:

Command	WRITE	
Parameter	Default	Function
FILE	<i>none</i>	Name of OPERA-2d data or results file.
SOL1	<i>none</i>	The name of an additional solution table.
SOL2	<i>none</i>	The name of an additional solution table.
SOL3	<i>none</i>	The name of an additional solution table.
SOL4	<i>none</i>	The name of an additional solution table.
SOL5	<i>none</i>	The name of an additional solution table.
SOL6	<i>none</i>	The name of an additional solution table.
SOL7	<i>none</i>	The name of an additional solution table.
SOL8	<i>none</i>	The name of an additional solution table.
SOL9	<i>none</i>	The name of an additional solution table.
SOL10	<i>none</i>	The name of an additional solution table.

The **WRITE** command writes a OPERA-2d data file. The first parameter defines the name of the **FILE**. If no file name extension is given, and extension *op2* is assumed. If a solution exists it will be included in the file. If the mesh data exists it will also be written to a file. Mesh file names have the extension *mesh*.

OPERA-2d data files consist of the solution properties (**SET** command), the region data, the **SYMMETRY** pairs, the BH curves and **EXTRA** data including the current units set. If a solution exists, the potentials, current/charge densities and permeabilities/permittivities are appended. The files are formatted, but include little text to help them to be understood.

The thermal, stress and space charge beam analysis programs are supplied with data and can return their solutions to the pre and post processor in tables. Up to 10 tables can be written to a data file by giving their names to parameters **SOLn**. For more information on setting up tables, see [page 3-113](#).

The ZOOM Command

Menu Routes:

DISPLAY↓
Zoom

The **ZOOM** command is also available at many points under **MODEL** and **FIELDS** menus. The menu items are called **ZOOM** or **ZOOM DISPLAY**.

Command Line Parameters:

Command	ZOOM
No Parameters	

ZOOM executes a **RECONSTRUCT** command (see page 3-170) having set the first 4 parameters (**XMIN**, **XMAX**, **YMIN**, **YMAX**) by use of the cursor. The other **RECONSTRUCT** parameters are unchanged. **ZOOM** has 7 options which should be used in the following ways:

- ... **zoom in** Define a rectangular area of the graphics window using a rubber-box. The part of the picture inside the rectangle will be made bigger to fill the graphics window.
- *In keyboard mode*, the first corner of a rectangular area should be defined with **<space>** cursor hit and the second corner with a **B**. The part of the picture inside the rectangle will be made bigger to fill the graphics window.
- ... **zoom out** Define a rectangular area of the graphics window using a rubber-box. The whole picture will be made smaller to fit inside the rectangle.
- *In keyboard mode*, the first corner of a rectangular area should be defined with **<space>** cursor hit and the second corner with an **S**. The whole picture will be made Smaller to fit inside the rectangle.
- ... **zoom in * 2** Select a point to be moved to the centre of the display. The picture will be redrawn with the range of horizontal and vertical coordinates halved and centred on the selected point.
- *In keyboard mode*, cursor hit **I** should be used to identify the point for the centre of the picture and zoom in.
- ... **zoom out * 2** Select a point to be moved to the centre of the display. The picture will be redrawn with the range of horizontal and vertical coordinates doubled and centred on the selected point.

- *In keyboard mode*, cursor hit **O** should be used to identify the point for the centre of the picture and zoom out.
- ... **pan** Select a point to be moved to the centre of the display. The picture will be redrawn with the same range of horizontal and vertical coordinates but centred on the selected point.
- *In keyboard mode*, cursor hit **P** should be used to identify the point which will become the new centre of the picture.
- **Bounding box** No further input is required. The program calculates the coordinate limits to include all the regions selected for display.
- *In keyboard mode*, cursor hit **B** should be used anywhere in the graphics window to request bounding box.
- **Previous size** No further input is required. This option cycles back through up to 20 sets of coordinate limits previously used in the program.
- *In keyboard mode*, cursor hit **E** should be used anywhere in the graphics window to request an earlier size.
- **Same size** No further input is required. This option redraws the picture with no change to the coordinate limits.
- *In keyboard mode*, cursor hit **R** should be used anywhere in the graphics window to request redraw.
- **Text input** Enter the values of the minimum and maximum values of X and Y into the ParameterBox.
- *In keyboard mode*, cursor hit **K** should be used anywhere in the graphics window to request keyboard. The program prompts for the coordinates of opposite corners to be given.
- **Return** No further input is required. This option returns to the higher level menu without redrawing the picture.
- *In keyboard mode*, cursor hit **Q** should be used anywhere in the graphics window to request quit.

ZOOM is available as a sub-command in the **DRAW**, **MODIFY** and **SYMMETRY** commands ([page 3-46](#), [page 3-149](#) and [page 3-191](#)).

INDEX

Symbols

!	2-13
!!	2-22
#	
free format variables	2-24
user variables	2-27
\$ ABORT	2-22
\$ ASK	2-31
\$ ASSIGN	2-33
\$ BACKSPACE	2-34
\$ CD	2-35
\$ CLOSE	2-32
\$ COMINPUT	2-30
\$ CONSTANT	2-27, 3-11
\$ DO-loops	2-25
\$ ELIF	2-26
\$ ELSE	2-26
\$ END	2-27
\$ FOR -each loops	2-26
\$ FORMAT	2-33
\$ IF	2-26
\$ OPEN	2-32
\$ OS	2-35
\$ PARAMETER	2-27, 3-11
\$ PAUSE	2-31
\$ PROMPT	2-31
\$ READ	2-32
\$ SKIP	2-22
\$ STRING	2-27
\$ WHILE -loops	2-26

\$ WRITE	2-32
\$-commands	2-25
\$VFDIR	
<i>see</i> VFDIR	
%INT, %REAL, %EXPR	2-19
%VFDIR%	
<i>see</i> VFDIR	
,, comma	2-14
/, comment	2-38
;, semicolon	2-18
, command separator	2-38
', quotation mark	2-19

-A-

abbreviation	2-12, 2-17
ac	3-19
adaptive refinement options	3-183
air gap	3-116
amplitude	3-19
analysis options	3-182
angle	
region parameter	3-48
system variable	3-13, 3-14
anisotropy	3-23
APPEND command	3-21
ARC sub-command	
in DRAW command	3-57
AREA	3-11, 3-12, 3-17, 3-130
in MODIFY command	3-150, 3-158
area	

integral 3-129
 printing 3-167
 arrow keys 2-2, 2-4

ASK

see **\$ ASK**

ASSIGN

see **\$ ASSIGN**

axisymmetry
 selecting 3-177

-B-

background
 fields 3-21
 region 3-46

BACKSPACE

see **\$ BACKSPACE**

BH tables 3-22

BHDATA command 3-22

bias 3-53

body force 3-114

boolean

parameters 2-19

boundary conditions 3-98

display 3-174

modifying 3-155

polygons 3-58

printing 3-167

region parameters 3-53

BR, BZ, BMOD 3-14

built-in commands 2-25

BX, BY, BMOD 3-13

-C-

calculator 2-29

cartesian coordinates

selecting 3-177

CARTESIAN sub-command

of **DRAW** command 3-57

CASE sub-command

of **SOLVE** command 3-181

CD

see **\$ CD**

CDBox 2-2, 2-6

character values 2-18

character variables 2-29

CHARDEN 3-15, 3-16

CHARGE

system variable 3-15, 3-16

charge density

region parameter 3-48

system variable 3-15, 3-16

units 3-204

CHECK command 3-26

CIRCLE command 3-27

CIRCLE sub-command

of **DRAW** command 3-59

circuits 3-87

CLEAR command 3-30

CLOSE

see **\$ CLOSE**

coercive force 3-22

system variable 3-13, 3-14

COLOUR command 3-31

ColourBox 2-2, 2-8

COMINPUT

see **\$ COMINPUT**

command line interface 2-1

command separator 2-38

commands 2-12

built-in 2-25

comments 2-38

COMOUTPUT command 3-33

complex permeability 3-22, 3-130

COMPONENT 2-14, 3-11

compound units 3-204

conditional commands 2-26

conductivity

region parameter 3-48

system variable 3-13, 3-14

units 3-204

conductor numbers 3-48, 3-87

CONSTANT

- see* **\$ CONSTANT**
 - constants 3-18
 - construction lines
 - displaying 3-173
 - modifying coordinates 3-156
 - polygon input 3-68
 - continuation lines 2-14
 - CONTOUR** command 3-34
 - CONVERT** command 3-38
 - coordinates
 - printing 3-167
 - COPY** command 3-39
 - CREATE** sub-command
 - of **EMIT** command 3-78
 - CURRENT** 3-13, 3-14, 3-210
 - current
 - system variable 3-13, 3-14
 - current density
 - region parameter 3-48
 - system variable 3-13, 3-14
 - units 3-203, 3-204, 3-210
 - current flow 3-130
 - cursor commands
 - DRAW** polygons 3-55, 3-67
 - ERASE** 3-86
 - EXTRA** 3-99
 - GROUP** 3-123
 - MODIFY** 3-151
 - PAN** 3-160
 - picking 3-152
 - POINT** 3-166
 - SYMMETRY** 3-192
 - ZOOM** 3-212
 - CURSOR** sub-command
 - of **DRAW** command 3-60
 - of **EXTRA** command 3-99
 - curvature 3-53, 3-58
- D-**
- DADT** 3-13, 3-14
 - data files 3-168
 - command input 2-30
 - time-tables 3-189
 - DATA** sub-command
 - of **SOLVE** command 3-182
 - deformed mesh plots 3-37
 - DELETE** sub-command
 - of **EMIT** command 3-80
 - density 3-107
 - DEVICE** command 3-41
 - device nomination 3-1
 - DialogBox 2-2, 2-7, 2-8, 3-122, 3-151
 - DIMENSION** command 3-43
 - displacement 3-39
 - DO**
 - see* **\$ DO**
 - double-click 2-6
 - DRAW** 3-46
 - DRAW** command 3-46
 - DRIVE** sub-command
 - of **SOLVE** command 3-187, 3-189
 - DUMP** command 3-70
 - DVR, DVZ, DMOD** 3-16
 - DVX, DVY, DMOD** 3-15
 - DXFOUTPUT** command 3-75
- E-**
- easy direction 3-22
 - EDD** 3-15, 3-16
 - eddy current conductors 3-26, 3-48
 - EDIT** command 3-76
 - EDIT** sub-command
 - of **EMIT** command 3-80
 - electric displacement
 - system variable 3-15, 3-16
 - electrostatics 3-176
 - element type
 - selecting 3-177
 - ELIF**
 - see* **\$ ELIF**
 - ELSE**
 - see* **\$ ELSE**

- EMIT** command 3-78
END command 3-85
END-DO
 see **\$ END**
END-FOR
 see **\$ END**
END-IF
 see **\$ END**
END-WHILE
 see **\$ END**
ENERGY 3-12, 3-129
 energy
 system variable 3-13
 units 3-204
ENERGY1 3-129
 environment variables 2-6, 2-19, 2-36, 3-41
EPSILON 3-15, 3-16
EPSILON0 3-12
ER, EZ, EMOD 3-16
ERASE command 3-86
ERASE sub-command
 of **DRAW** command 3-60
ERDR, ERDZ, EZDR, EZDZ 3-16
ERROR 3-12
 Euler angles 2-39
EX, EY, EMOD 3-15
EXDX, EXDY, EYDX, EYDY 3-15
EXPORT sub-command
 of **EXTRA** command 3-101
 expressions 2-16, 3-11, 3-18
EXTERNALCIRCUIT command 3-87
EXTRA command 3-97

 -F-
 face properties 3-52
 printing 3-167
 field averaging .. 3-29, 3-134, 3-137, 3-141,
 3-164, 3-178
 field gradient
 system variable ... 3-13, 3-14, 3-15, 3-16

 field point coordinates 3-13, 3-14, 3-15,
 3-16
 field strength
 system variable 3-13, 3-14, 3-15, 3-16
 units 3-204
 field type
 selecting 3-178
 field values
 along circular arcs 3-29
 along lines 3-141
 along region sides 3-163
 area integrals 3-129
 at points 3-166
 contours 3-36
 line integrals 3-134, 3-137
 vectors 3-206
FileBox 2-2, 2-5, 2-8
 filenames 2-18
 files
 BH 3-23
 command input 3-30, 3-33
 data 3-168, 3-211
 dialogue 3-29, 3-141, 3-164, 3-167,
 3-206
 mesh 3-147, 3-168, 3-211
 results 3-21, 3-168, 3-211
FINISH sub-command
 of **DRAW** command 3-61
FLUX 3-12, 3-129
 flux density
 system variable 3-13, 3-14
 units 3-204
FOR
 see **\$ FOR**
 force 3-134, 3-137
 system variable 3-13, 3-14, 3-15, 3-16
 units 3-204
 force density
 system variable 3-13, 3-14
FORMAT
 see **\$ FORMAT**

- Fourier series 3-125
FR, FZ 3-14, 3-16, 3-129
 free format input 2-24
FREQ 3-13
 frequencies 3-181
FX, FY 3-13, 3-15, 3-129, 3-134, 3-137
FZ 3-134, 3-137
- G-**
- gap 3-116
GAP command 3-116
 gap regions 3-46
GO conductor 3-90
GRAPH command 3-118
 graphical user interface 2-1
 graphs 3-29, 3-141, 3-164
 grid 3-66, 3-156
GROUP command 3-121
 GUI 2-1
- H-**
- hard magnetic materials 3-22
HARMONICS command 3-124
HCR, HCZ 3-14
HCX, HCY 3-13
HDB 3-13, 3-14
 help character 2-13
HELP command 3-128
 hide menus 2-9
 homogeneity 3-29, 3-36, 3-141, 3-164,
 3-166
 horizontal menu 2-2
HR, HZ, HMOD 3-14
HRDR, HRDZ, HZDR, HZDZ 3-14
HX, HY, HMOD 3-13
HXDX, HXDY, HYDX, HYDY 3-13
 hysteresis 3-22, 3-130
- I-**
- IF**
 see **\$ IF**
- IMPORT** sub-command
 of **EXTRA** command 3-104
 inductance 3-88
INTAREA command 3-129
INTCIRCLE command 3-133
INTEGRAL 3-12, 3-130, 3-134, 3-137
 integral
 area 3-129
 circular arc 3-133
 current 3-131
 flux 3-131
 force 3-131, 3-132, 3-134, 3-137
 Fourier coefficients 3-125, 3-126
 Legendre coefficients 3-126
 line 3-136
 power 3-131
 stored energy 3-130, 3-131, 3-132
 torque 3-134, 3-137
 integral fields 3-178
INTLINE command 3-136
 Isotropic materials 3-23
- J-**
- J** 3-13, 3-14, 3-210
- L-**
- Legendre polynomials 3-125
 length
 units 3-204
 length of circuit 3-90
 limits 1-3
LINE command 3-139
 line contours 3-36
 line integral 3-133, 3-136
LINE sub-command
 of **DRAW** command 3-61
 linear analysis 3-182
 linear elements 3-177
 linear motion 3-143
LIST sub-command
 of **EMIT** command 3-82

LM solver 3-143
LMMOTION command 3-143
LOAD sub-command
 of **EXTRA** command 3-106
 local coordinate system
 modifying 3-159
 local coordinates
 in **COPY** command 3-39
 printing 3-167
 region parameters 3-51
 loops 2-25, 3-23
 Lorentz force
 system variable 3-13, 3-14
LR, LZ 3-14
LX, LY 3-13

 -M-
M 3-210
 mass
 units 3-204
 material angle
 region parameter 3-48
 system variable 3-13, 3-14
 material code 3-48
 material properties 3-98, 3-99
 electromagnetics 3-48
 stress analysis 3-99, 3-107
 thermal analysis 3-99, 3-107
MATERIAL sub-command
 of **EXTRA** command 3-106
MAXIMUM 3-12, 3-29, 3-142, 3-164
 Maxwell stress 3-114, 3-134, 3-137
 menu interface 2-1
 menu off 2-9
 menus 2-2
 mesh
 display 3-147, 3-174
 generation 3-143, 3-146
 linear motion 3-143
MESH command 3-146
 mesh files 3-168

MessageBox 2-2, 2-7, 2-31, 2-38
MINIMUM 3-12, 3-29, 3-142, 3-164
MODIFY command 3-149
 motion 3-143
 moving regions 3-143
MU 3-13, 3-14
MUO 3-12
 multiple cases 3-181

-N-

narrow gaps 3-47
 nodally averaged fields *see* field averaging
 3-178
 nodes 3-174
 non-linear analysis 3-182
 normal unit vector 3-13, 3-14, 3-15, 3-16
NOW 2-29
 numeric values 2-16
 numerical variables 2-27
NVR, NVZ 3-14, 3-16
NVX, NVY 3-13, 3-15

-O-

OPEN
 see **\$ OPEN**
opera2.comi 3-1, 3-30
Opera2d_PP_n.log 2-11
Opera2d_PP_n.lp 2-11, 3-167, 3-206, 3-210
 operaanl 2-36
 operating system commands 2-35
OS
 see **\$ OS**
 output time 3-181

-P-

packing factor 3-23
 region parameter 3-22, 3-48
PAN command 3-160
PARAMETER
 see **\$ PARAMETER**

- ParameterBox 2-2, 2-3, 2-4, 2-8, 3-151, 3-213
 - parameters 2-12, 3-18
 - particle trajectories 3-198, 3-208
 - PAUSE**
 see **\$ PAUSE**
 - periodicity 3-117
 - periodicity conditions 3-53, 3-58, 3-191
 - permeability
 - region parameter 3-48
 - system variable 3-13, 3-14
 - permittivity
 - region parameter 3-48
 - system variable 3-15, 3-16
 - perturbation fields 3-21
 - PHASE** 3-13, 3-14
 - phase 3-19
 - phase angle
 - complex permeability 3-22
 - region parameter 3-48
 - system variable 3-13, 3-14
 - PI** 3-12
 - pick operations 3-152
 - picking 2-3
 - pictures 3-172
 - PLOT** command 3-161
 - POINT** command 3-165
 - point transformation 3-151
 - Poisson's ratio 3-107
 - POLAR** sub-command
 - of **DRAW** command 3-62
 - polygons 3-46
 - positional assignments 2-14
 - POT** 3-12
 - potential
 - selecting for solution 3-177
 - system variable 3-12
 - units 3-204
 - POWER** 3-13, 3-14, 3-129
 - power
 - system variable 3-13, 3-14
 - units 3-204
 - PRINT** command 3-167
 - PROCESS** sub-command
 - of **EXTRA** command 3-110
 - program sizes 1-3
 - PROMPT**
 see **\$ PROMPT**
 - prompting
 - for parameter values 2-22
 - in command input files 2-31
 - PSHIFT** sub-command
 - of **DRAW** command 3-63
- Q-
- Q** 3-210
 - quadratic elements 3-177
 - quadrilateral regions 3-46
 - QUIT** sub-command
 - of **DRAW** command 3-63
 - of **EMIT** command 3-83
 - of **EXTRA** command 3-111
- R-
- R, Z** 3-14, 3-16, 3-210
 - radio-buttons 2-7
 - radius 3-116
 - READ**
 see **\$ READ**
 - READ** command 3-168
 - READ** sub-command
 - of **EMIT** command 3-83
 - RECONSTRUCT** command 3-170
 - reflection 3-39
 - region coordinates 3-49
 - region corners 3-151
 - region matching 3-193
 - region properties
 - printing 3-167
 - REGIONS** 3-17
 - regions 3-46
 - relative permeability

system variable 3-13, 3-14
 relative permittivity
 system variable 3-15, 3-16
 replications
 copying regions 3-39
 printing 3-167
 region parameters 3-51
 restart solutions 3-184
 results files 3-168
RETURN conductor 3-90
RLOAD, ZLOAD 3-114
 RM 3-117
 rotating machine 3-117
 rotating machines
 see gap regions
 rotating machines velocity 3-184
 rotation 3-39
 rotor 3-116
RSHIFT sub-command
 of **DRAW** command 3-64
 rubber-box 2-3

-S-

saturation magnetisation 3-22
SAVE sub-command
 of **EXTRA** command 3-112
SCALE 3-13
 scale factor 3-181
 scaling 3-33
 scroll bars 2-6, 2-7
 selecting 2-2
SET command 3-175
SHAPE codes 3-46, 3-48
 shear modulus 3-107
SIGMA 3-13, 3-14
 slider bars 2-8
 snap to grid 3-66, 3-156
SOLVE command 3-180
 space charge analysis options 3-184
 space charge beam analysis
 magnetic field tables 3-211

specific heat capacity 3-107
 SPICE type data 3-93
 stator 3-116
 steady-state ac
 post processing 3-19
STORE sub-command
 of **EMIT** command 3-84
 stored energy
 system variable 3-14, 3-15, 3-16
 units 3-204
 stress analysis
 load tables 3-114, 3-211
 loads and constraints . 3-98, 3-104, 3-110
 material properties 3-99, 3-107
 solution tables 3-169
 stress analysis options 3-184
STRING
 see **\$ STRING**
 subdivision 3-53
 superscripts 2-19
 symbol contours 3-37
SYMMETRY
 boundary conditions ... 3-53, 3-58, 3-191
 eddy current conductors 3-48
 of problem (**XY** or **AXI**) 3-177
 symmetry 3-116
SYMMETRY command 3-191
 system variables 3-12
 VIEW command 3-209

-T-

TABLE sub-command
 of **EXTRA** command 3-113
 tables
 creating 3-98
 post processing 3-12
 reading 3-169
 writing 3-211
 tangential unit vector 3-13, 3-14, 3-15, 3-16
TEST command 3-193
 text function 2-19

text output modes 2-30

thermal analysis

- boundary conditions 3-98
- heat source tables 3-114
- material properties 3-99, 3-107
- solution tables 3-169
- temperature tables 3-211

thermal conductivity 3-107

thermal expansion integral 3-107

time average 3-19

time derivative 3-13, 3-14

time of flight 3-210

TIME parameter 3-19

timers 2-8

time-table files 3-189

timetables 3-188

TITLE command 3-195

titles 3-29, 3-142, 3-164, 3-195

TODAY 2-29

TOF 3-210

tolerance 3-146

TORQUE 3-13, 3-15, 3-134, 3-137

torque 3-134, 3-137

- system variable 3-13, 3-15

TRACK command 3-197

trajectories 3-197, 3-207

transient drive functions 3-187

transient time step options 3-183

TTIME 3-13

TVR, TVZ 3-14, 3-16

TVX, TVY 3-13, 3-15

-U-

UNDO command 3-201

UNITS command 3-202

user constants 3-18

user interface 2-1

user parameters 3-18

user variables 2-27, 3-18

-V-

variables 3-18

- character 2-29
- in expressions 2-16, 2-17
- in free format input 2-24
- numerical 2-27

VECTOR command 3-205

VELOCITY 3-13, 3-14

velocity

- region parameter 3-48
- system variable 3-13, 3-14

VELX 3-210

VELY 3-210

VELZ 3-210

vertical menu 2-2

VFDIR 2-6, 2-19, 2-36

VFGRAPHICS 3-41

VFINV 3-42

VFWINDOWH 3-42

VFWINDOWW 3-42

VIEW command 3-207

Voltage driven circuits 3-87

VOLUME 3-11, 3-14, 3-16, 3-17, 3-130

volume

- printing 3-167
- system variable 3-14, 3-16

VX 3-11

VX and **VY** 2-14

VY 3-11

-W-

WHILE

- see* **\$ WHILE**

WRITE

- see* **\$ WRITE**

WRITE command 3-211

-X-

X, Y 3-13, 3-15, 3-17, 3-209, 3-210

XLOAD, YLOAD 3-114

XSHIFT sub-command

- of **DRAW** command 3-64
- Y-
- Young's modulus 3-107
- YSHIFT** sub-command
 - of **DRAW** command 3-65
- Z-
- zone contours 3-37
- ZOOM** command 3-212
- ZOOM** sub-command
 - of **DRAW** command 3-66
 - of **EMIT** command 3-84
 - of **EXTRA** command 3-115