

DESIGN ENVIRONMENT 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

NOTE: The symbol * indicates the command is available in both the Configuration System, and the User Sytem. Otherwise, the command is only available in the Configuration System.

Chapter 1 *Design Environment Reference Manual*

Introduction	1-1
The ANGLE Command	1-2
The BHDATA Command*	1-3
The BOUNDARY Command	1-7
The CASE Command	1-9
The CLEAR Command*	1-11
The COLOUR Command*	1-12
The CONSTRAIN Command	1-14
The CONTROL Command*	1-16
The CONVERT Command	1-19
The COPY Command	1-22
The CURVATURE Command	1-25
The DANGLE Command	1-27
The DELETE Command	1-28
The DEVICE Command*	1-30
The DIANGLE Command	1-32
The DISPLAY Command*	1-33
The DLENGTH Command	1-37
The DUMP Command*	1-38

The DVECTOR Command	1-43
The END Command*	1-44
The EXTERNALCIRCUIT Command*	1-45
The GETOP2 Command	1-55
The GRAVITY Command	1-57
The IANGLE Command	1-59
The INSERT Command	1-60
The LENGTH Command	1-61
The LIMIT Command	1-62
The LINK Command	1-64
The LIST Command	1-66
The LOAD Command*	1-68
The MATERIAL Command	1-69
The MODIFY Command	1-74
The PART Command	1-75
The POINT Command	1-80
The POLYGON Command	1-81
The POSITION Command	1-82
The POST Command	1-83
The RUNPART Command*	1-89
The SET Command	1-91
The SKETCH Command	1-93
The SOLC Command	1-95
The SOLVE Command*	1-96
The SYMMETRY Command	1-103
The TITLE Command*	1-105
The UNITS Command	1-107
The VARIABLE Command	1-110
The VECTOR Command	1-114
The WRITE Command	1-115
The ZOOM Command*	1-116

Chapter 1

Design Environment Reference Manual

Introduction

This reference manual is primarily directed at the *Configuration System*. The *User System* is operated through a subset of these commands. Commands available in both systems are marked with an *.

The **ANGLE** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Angle

Command Line Parameters:

Command	ANGLE	
Parameter	Default	Function
P1	<i>none</i>	Name of first point to be constrained.
P2	<i>none</i>	Name of second point to be constrained.
THETA	<i>none</i>	Angle used to define the constraint.

This command sets an angle constraint between two points.

The **ANGLE** command constrains two points to be fixed such that they form an angle with respect to the global x-axis. An angle of zero is defined by two points lying on or parallel to the x-axis and with the first point trailing the second point along the x-axis.

The parameter **P1** refers to the first point name that is to be constrained and **P2** refers to the second point name. The parameter **THETA** takes either an expression or numeric value as its argument and defines the angle that the two points **P1**, **P2** make with a line parallel to the x-axis.

The **ANGLE** constraint adds one constraint to the constraint list defining the model.

The **MODIFY** command on [page 1-74](#) can be used to alter the values or expressions that are used to define the angle between points **P1**, **P2**.

The **BHDATA** Command*

Menu Route:

MODEL ↓
BH or DE data → BH/DE editing

Command Line Parameters:

Command	BHDATA		
Parameter	Default	Function	
MATERIAL	IRON	Material name (up to 99 allowed).	
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.	
PACK		Expression for the packing factor for laminated materials.	
PHASE	0	Expression for the phase lag in AC complex permeability solutions.	

The OPERA-2d analysis programs use material characteristics to relate flux density and field intensity of all **MATERIALS** other than **AIR** and **CONDUCTOR**. For soft magnetic materials these 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). For hard magnetic materials the operating curve should be defined. The first value of **B** should be zero and **H** the coercive force. The easy direction of the magnet is set by the **PHASE** parameter in the region data (see **DRAW** command in *OPERA-2d Reference Manual*). 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 any case only the first 4 characters are decoded.

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 the 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.)

```
OPERA-2d -BHDA > $ do #i 1 23
OPERA-2d -BHDA > r #i h+(b-h)*0.9 h -redr
OPERA-2d -BHDA > $ end do
```

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 now 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.

$$\mu_{normal} = \frac{\mu\mu_0}{p\mu_0 + (1-p)\mu} \quad (1.1)$$

$$\mu_{tangential} = p\mu + (1-p)\mu_0 \quad (1.2)$$

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.

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

The **BOUNDARY** Command

Menu Route:

MODEL ↓
 Geometry → Set boundary condition

Command Line Parameters:

Command	BOUNDARY		
Parameter	Default	Function	
TYPE	<i>none</i>	Type of boundary condition.	
		V DV	Set potential condition. Set normal derivative of potential to zero.
		SYMMETRY	One of a symmetry (periodicity) pair.
		CLEAR	Clear any boundary conditions on side
EXPRESSION	<i>none</i>	Expression for the potential if TYPE=V	
SIDE	<i>none</i>	Side whose boundary condition is to be set.	

The **BOUNDARY** command is used to either set or remove any boundary conditions on a polygon side. The boundary conditions available are setting a potential, setting the potential derivative to zero and setting a symmetry boundary.

Initially, all sides have no boundary condition. All boundary conditions are set using the **BOUNDARY** command and applied to polygon sides. The side of a polygon is defined between two points of the polygon, and is referenced by the lower of the two node numbers defining the side (except for the last polygon side lying between the last and first node of the polygon, which is referenced by the last node number).

The different boundary condition types have a different effect depending upon the potential type being used (note that here V is used to represent either vector or scalar potential).

Vector potential	$V=0$	Tangential flux boundary
	$\frac{dV}{dn} = 0$	Normal flux boundary
Scalar potential	$V=0$	Normal flux boundary
	$\frac{dV}{dn} = 0$	Tangential flux boundary

The potential applied to a side may be an expression or numeric value. The expression may contain functions of the (x,y) coordinate position. If this is the case, the potential is automatically converted to an **EXTRA** boundary condition. This **EXTRA** condition means that potentials may be defined to vary along a particular side.

If the parameter **SIDE** is given no argument, a mouse cursor is shown over the model, and the user may assign the boundary condition as defined by **TYPE** and **EXPR** (for a potential value) to several sides by clicking the mouse cursor on to or close to the appropriate sides. Clicking the cursor twice on a polygon side results in the de-selection of that side. Once all sides have been assigned, pressing **a** will accept the sides and exit mouse cursor mode. The following keystrokes are available under mouse cursor mode.

Keystroke	Function	
a	Apply boundary condition to sides and quit	
q	Quit and do not apply any boundary conditions	
v	Set or change a variable value	
s	Solve the variational geometry	
z	Zoom the display	
	b	Set display to bounding box.
	i	Zoom in by factor of 2.
	k	Use keyboard input to set display.
	o	Zoom out by factor of 2.
	p	Pan to new centre.
	r	Redraw display.
	q	Quit
h	Bring up some help notes	

The CASE Command

Menu Route:

```

FILE ↓
    Load a CASE

FILE ↓
    Study case results
  
```

Command Line Parameters:

Command	CASE	
Parameter	Default	Function
OPERATE	BROWSE	How to open the case results file
		LOAD Load a case result data set. Load * picks last browse BROWSE Browse through the case results file. ? replaces 1 character as a wildcard browse * looks for all wildcard entries.
N1	none	The starting case data set in the .CAS file.
N2	none	The finishing case data set in the .CAS file.
FILE	none	Check results from this solution filename

The use of the case command is now redundant as the cases are no longer stored in the .CAS file. The command is kept for compatibility with existing case files. A new **CONTROL** command has been added for storing and retrieving model variations. See “The CONTROL Command*” on page 1-16.

This command is used to either import a set of specific variable settings into the Design Environment, or browse through previous results in the case results file. The .CAS file contains a record of all configurations that have been stored, converted into .op2 files and analysed. Each case contains the set of all variable values and BH data for that configuration.

The parameter **OPERATE** specifies whether the user wishes to retrieve a case file, or browse through the case results file.

To retrieve a known data set, define **OPERATE=LOAD** and set **N1** to the appropriate case number. Parameters **N2** and **FILE** are unused in this mode. Setting **N1=***

will cause the last case set to be retrieved. Once a data set has been retrieved, the variables will take the new values, the geometry will alter accordingly and any post processing results will be shown in an information box.

To browse through the case results file associated with the loaded DEM, define **OPERATE=BROWSE** and set **N1** and **N2** to the range of case numbers. Setting **N1=1** and **N2=*** will cause all cases to be browsed.

The parameter **FILE** can be used to make the browsing feature more selective in finding cases. Each set of case results in the *.CSR* file has the name of the solution file analysed to get the results. This filename can be used as the basis for searching particular results. Setting the parameter **FILE** to a known solution filename will cause the browser to find and show the post processing results for that case.

If a particular filename is not known, the **FILE** parameter need not be set, in which case, all case results will be shown for the range of case numbers given by parameters **N1** and **N2**.

If a solution filename is only partially known, wild-card characters can be used. The character '?' can be used as part of the **FILE** parameter to indicate that any filenames encountered may be browsed if they only differ by one character lying in the same position as the '?'.

Similarly, a '*' can be used as part of the **FILE** parameter to indicate that any filenames encountered may be browsed if their name only differs from the parameter given by a series of characters that could be represented by the '*'.

The data sets are written to the case file after each **CONVERT** command (see "The **CONTROL** Command*" on page 1-16) is given to generate an *OP2* file, or after every **STORE** command (see "The **SYMMETRY** Command" on page 1-103). The data sets are automatically appended to the end of the case file and the case number incremented by one. If the DEM filename changes, a new *.CAS* file is opened taking the same name as the new DEM file.

The CLEAR Command*

Menu Route:

OPTIONS ↓
Clear and reset

Command Line Parameters:

Command	CLEAR
No Parameters	

The **CLEAR** command puts the *Configuration System* or the *User System* 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 *OPERA-2d Reference Manual*) before control passes back to the user. This allows the user to supply an individual choice of default values for commands such as **UNITS** (see “[The UNITS Command](#)” on page 1-107).

The **COLOUR** Command*

Menu Route:

OPTIONS ↓
Colour settings

Command Line Parameters:

Command	COLOUR	
Parameter	Default	Function
C0	3	Colour map number for air.
C1	4	Colour map number for conductors.
C3	6	Colour map number for non-linear 1.
C4	7	Colour map number for non-linear 2.
C5	8	Colour map number for non-linear 3.
C6	9	Colour map number for non-linear 4.
C7	10	Colour map number for non-linear 5.
C8	11	Colour map number for non-linear 6.
C9	12	Colour map number for non-linear 7.
C10	13	Colour map number for non-linear 8.
C11	14	Colour map number for non-linear 9.
C12	15	Colour map number for non-linear 10.
CTEXT	1	Colour map number for text and axes.
CODE	1	Colour map number to be redefined.
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), or 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 immediate effect, if the colour number appears on the display). **RED**, **GREEN** and **BLUE** should each be in the range 0 to 1.

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 *OPERA-2d Reference Manual*)

The **CONSTRAIN** Command

Menu Route:

MODEL ↓
 Constraints → By cursor
 DEM ↓
 Dimension line → By cursor

Command Line Parameters:

Command	CONSTRAIN
No Parameters	

This command is used to set the constraints of the model through the mouse cursor rather than directly through the GUI menuing or keyboard command input.

The options available to the user are as follows:

Command	CONSTRAIN	
Keyboard	Default	Function
b	<i>none</i>	Backtrack last point
c	FIXED	Constraint type
h	<i>none</i>	Help
r	<i>none</i>	Refresh display
s	<i>none</i>	Solve geometry (See “The SOLC Command” on page 1-95.)
v	<i>none</i>	Set variable value (See “The VARIABLE Command” on page 1-110.)
z	<i>none</i>	Zoom display (See “The ZOOM Command*” on page 1-116.)
<space>	<i>none</i>	Choose point at cursor

The types of constraint allowed are as follows

FIXED point	1 point	(See “The POINT Command” on page 1-80.)
VECTOR	2 points	(See “The VECTOR Command” on page 1-114.)
DVECTOR Vector Difference	4 points	(See “The DVECTOR Command” on page 1-43.)

LENGTH	2 points	(See “The LENGTH Command” on page 1-61.)
DLENGTH Length Difference	4 points	(See “The DLENGTH Command” on page 1-37.)
ANGLE	2 points	(See “The ANGLE Command” on page 1-2.)
DANGLE Angle Difference	4 points	(See “The DANGLE Command” on page 1-27.)
IANGLE Internal Angle	3 points	(See “The IANGLE Command” on page 1-59.)
DIANGLE Internal Angle Difference	6 points	(See “The DIANGLE Command” on page 1-32.)

Once the constraint has been chosen, the nodes or reference points to be constrained need to be defined. This is done by either clicking the mouse cursor, or pressing the space-bar at the points. The order in which the points are defined is very important, as their spatial positioning play a key role in the constraint definition. For example, the two points defining an **ANGLE** constraint can result in a 180° degree error, if the first point is confused with the second point. The first point is **P1**, the second **P2** and so on until all points are defined for the constraint. The positions of the points with respect to each other are found in the commands defining the individual constraints.

Once all points have been chosen, a prompt will automatically appear requiring the input of either a variable, expression or numeric value to define the constraint. Once this has been put in, the constraint is defined.

- Pressing **b** will result in the deletion of the last chosen point from the point list, if the wrong point is chosen.
- Pressing **n** will result in the generation of a reference point which will be one of the points **P1**, **P2**,..., depending on when the reference has been added in the sequence of point definitions. The additional reference point will be partly or fully constrained as a result of being used in the constraint definition.
- Pressing **s** will result in the geometry being solved and the display up-dated.
- Pressing **r** will result in the refreshing of the display.
- Pressing **z** will enable the user to zoom the display using the standard zoom options found in section “The ZOOM Command*” on page 1-116.
- Pressing **v** will enable the user to set a variable value.
- Pressing **<space>** will define the closest node to the cursor to be point **P1**, **P2**,.. of the points required to define the constraint.

The **CONTROL** Command*

Menu route: ANALYSIS ↓

Command Line Parameters:

Command	CONTROL		
Parameter	Default	OPTION	Function
OPTION	<i>none</i>		Option parameter to determine the action of the CONTROL command
		INIT	Clears the current analysis control set
		LOAD	Loads a model variation
		PREPARE	Prepares the models in the current analysis control for analysis and post processing
		READ	Displays results from a previous analysis control set
		RUN	Prepares the current analysis control set and launches the pre and post processor for analysis and post processing this set.
		SET	Sets the filename to be used for a new analysis control set
		SOLVE	Launches the pre and post processor for analysis and post processing a previously prepared analysis control set
		STORE	Stores a new model variation and adds it to the current analysis control set
FILE	<i>none</i>		File parameter for use with LOAD , READ , SET , SOLVE and STORE options

The **CONTROL** command is used for the control of the automatic analysis and post processing of models generated from the Design Environment.

A current control set is created using the **OPTION=SET** command. The filename is used as the basis for generating the control files necessary to run the analysis and post processing of a set of model variations.

The **OPTION=INIT** can be used to reset the current analysis control set. Any model variations stored with this set are cleared.

Once the control set name has been given, model variations, i.e. models that have got different settings for the parameters, can be added to the control set using **OPTION=STORE**, with a filename for each variation. Model variations can also be added to the current control set using the **STORE** option within the **RUNPART** command. See “The **RUNPART** Command*” on page 1-89.

Once the set of variations to be added to this analysis control set is complete, the models must be prepared using **OPTION=PREPARE**. This involves converting the variation data to OPERA-2d data and generating the model data files. Post processing command script files are also generated for each model variation. The preparation of the files also creates a command script to control the analysis and post processing. This file is named from the analysis control set name.

The pre and post processor can be started by using **OPTION=SOLVE**. This requires the name of the analysis control set that is to be used on starting the pre and post processor. The file *opera2.comi* is generated to call this the analysis control command script, and is executed upon starting the pre and post processor. Protection is added to ensure that the file *opera2.comi* and post processing command scripts are only allowed to be executed once.

NB: For UNIX systems it is necessary to ensure that the aliases for **OPERAANL** and **OPERAPP** have been set up before starting the Design Environment. This is detailed in the installation notes.

The **OPTION=RUN** combines the effects of **OPTION=PREPARE** and **OPTION=SOLVE** as both of these can be quite time consuming processes.

When post processing data, any **RESULTS** generated by the command script are stored with the model variation file, and also in a results file for the analysis control set.

Any of the model variations that have been stored can be loaded using **OPTION=LOAD**. This loads the values of parameters and material data from this model variation. If any results are known from the analysis of this model variation, they are displayed.

The analysis control set results file can be viewed using **OPTION=READ**.

A summary of the files generated is given in the table below.

Filename	Command option	Description
<i>analysis.comi</i>	SET	Command script for controlling the analysis and post processing of the analysis control set.
<i>analysis.csr</i>	SET	Analysis control set results file. Initially generated with the SET option, but added to by the post processing command scripts when they are run.
<i>variation.var</i>	STORE	Model variation file created by the STOR option. Results are appended to these files when the post processing command script is run.
<i>variation.op2</i> <i>variation.script</i>	PREPARE	Model files and post processing command script generated when preparing the analysis control set.

Some of the files are renamed after analysis and post processing to avoid re-running the same model. Additional files, the solution file and mesh file, will also be created by the analysis and post processing.

The **CONVERT** Command

Menu route:

MODEL ↓
Check OPERA-2d data

ANALYSIS ↓
Check OPERA-2d data

Command Line

Parameters:

Command	CONVERT		
Parameter	Default	Function	
SUBDIVISION	HIGH	Meshing subdivision matching option.	
		HIGH	Match neighbouring facets taking the higher mesh subdivision.
		LOW	Match neighbouring facets taking the lower mesh subdivision
		AVERAGE	Match neighbouring facets taking the average mesh subdivision.
KEEP	POT	Determine the boundary conditions to be kept on any internal region boundary	
		ALL	Keep all internal potential, derivative and symmetry boundary conditions.
		POTENTIAL	Keep all internal potentials, remove other internal boundary conditions.
		NONE	Keep no internal boundary conditions.
MESH	NO	Option to automatically generate the mesh.	
		YES	Generate the mesh
		NO	Do not generate the mesh

This command is used to **CONVERT** the DEM data into OPERA-2d data for testing the integrity and consistency of the data.

The commands checks the topological and material data of the model to ensure that the geometry is well constrained and that the material values are consistent

with the material types. Warnings are posted to the screen if inconsistencies are found.

The **SUBDIVISION** parameter sets the way in which neighbouring sides have their subdivisions matched. **SUBDIVISION=LOW** causes neighbouring sides to take the lower subdivision setting of the two sides, **SUBDIVISION=AVERAGE** causes the neighbouring sides to be assigned an average subdivision and **SUBDIVISION=HIGH** causes the sides to take on the higher subdivision setting of the sides.

The **KEEP** parameter sets the internal boundary conditions. **KEEP=ALL** keeps all internal boundary conditions without exception. **KEEP=POTENTIAL** removes all boundary conditions except for those with **BOUNDARY TYPE=V** and is useful for electrostatic models where internal boundaries are likely to be set to a driving voltage. **KEEP=NONE** removes all internal boundary conditions without exception. Any boundary that marks the physical extent of the model is unaffected by this parameter.

The **MESH** parameter allows the automatic generation of the OPERA-2d mesh data as well.

The **CONVERT** command will convert the DEM model data into standard OPERA-2d data format. The conversion can take some time if the model is particularly complex. Operations involved in the conversion are:

- Generation of the OPERA-2d geometry, material and boundary conditions
- Calculation of equivalent OPERA-2d mesh information that is applied to the region sides. Additional points can be added to the region geometry to help generate subdivision data that matches the information set up within the DEM
- Matching of symmetry sides, including the addition of extra points if necessary to ensure continuity between periodic sides.
- Conversion of quadrilateral regions with suitable subdivision values to regularly meshed shape **H** or **Q** regions.

After issuing the **CONVERT** command, a set of sub-commands is available. These commands are a subset of the standard OPERA-2d pre and post processor commands that allow viewing and checking of the OPERA-2d data created.

CONVERT Sub-commands

Sub-command	Function
CHECK	Checks the region and material data
LIST	Lists extra conditions and circuit information
MESH	Generates the mesh data
PRIN	Prints region data
QUIT	Leaves the CONVERT command
RECO	Displays the OPERA-2d data
TITLE	Adds a title to the display
WRITE	Generates an OPERA-2d data file (*.op2)
ZOOM	Zoom the display

These sub-commands are from the OPERA-2d pre and post processor. More information on the use of these commands can be found in the OPERA-2d reference manual.

OPERA-2d data files can be generated using the **WRITE** sub-command, and this file can then be read into the pre and post processor, or passed to the solver for analysis.

The COPY Command

Menu Route:

MODEL ↓
Geometry → Copy polygon

Command Line Parameters:

Command	COPY		
Parameter	Default	Function	
POLYGON	none	Polygon to be copied.	
TYPE	none	Type of polygon copy.	
		ROTATION	Copy polygon through a rotation angle.
		TRANSLATION	Copy polygon through a vector translation.
REFLECTION	Copy polygon through a plane reflection.		
CP	none	Centre point of rotation.	
ANGLE	none	Angle of rotation about CP.	
DX	none	X displacement.	
DY	none	Y displacement.	
P1	none	Start point of reflection plane.	
P2	none	End point of reflection plane.	

This command is used to create a new polygon from a master polygon. The characteristics of the new polygon are that of a slave to the master polygon. This slave characteristic is similar to that of a replication, but the slave polygon can be re-constrained and reconfigured so as to give it total independence from the master polygon. Hence it is fundamentally different from a replication.

The COPY command makes a copy of the polygon specified by POLYGON. The following coordinate transformations are made to form the slave polygon. (Only one transformation is permitted for a particular copy).

- Rotation through an angle ANGLE with respect to a fixed pivot point CP
- Reflection in a mirror plane defined by the start and end points P1 and P2
- Translation by DX, DY in the x direction and y direction

The **COPY** command creates a slave polygon which can be subsequently modified to have independent and different material properties or coordinates to the master polygon.

The slave polygon is different to the parent polygon, in that the expressions and constraints defining the geometry and material properties are unique to that slave polygon. But if the master polygon is geometrically changed, the constraints on the slave polygon will reflect the changes in the master polygon. The slave polygon changes because the constraints that have been defined now create a dependency between the slave polygon and the master. Removing or modifying these constraints will affect the dependence relationship between the master and slave polygons.

The three types of transformation available set up a variety of geometry constraints linking the master polygon to the slave polygon:

DX, DY Translation

This type of transformation copy sets up vector constraints (see “[The VECTOR Command](#)” on page 1-114) between nodes of the master and slave polygon. The vector constraints take the values of **DX** and **DY** as their parameters. These values can be expressions or numeric values.

ROTATION Transformation

This transformation sets up internal angle and difference in length constraints (see “[The LENGTH Command](#)” on page 1-61) between nodes of the master/slave polygons and the central pivot node **CP**.

If the transformation causes a node to be rotated onto itself, i.e. a node of the master polygon is used as the centre point, a simple vector constraint (see “[The VECTOR Command](#)” on page 1-114) is set up specifying zero displacement in the x and y directions.

REFLECTION Transformation

This transformation sets up a difference in internal angle constraint (see “[The DIANGLE Command](#)” on page 1-32) and a difference in length constraint (see “[The LENGTH Command](#)” on page 1-61) between nodes of the master/slave polygons and the reflection plane **P1** and **P2**.

If the transformation causes a node to be reflected onto itself, i.e. the point lies on the reflection line, a simple vector constraint (see “[The VECTOR Command](#)” on [page 1-114](#)) is set up specifying zero displacement in the x and y directions.

Care should be taken to fully constrain the points that are used for the reflection and rotation transformations, such as the **CP** centre point for rotation and points **P1**, **P2** used as the reflection line. These points are used in the master/slave constraint relationships and so changing the constraints on these points will cause the slave polygon to move. This can be alleviated by re-constraining the slave polygon and so removing the master/slave relationship.

The CURVATURE Command

Menu Route:

MODEL ↓
Geometry → Set side curvature

Command Line Parameters:

Command	CURVATURE		
Parameter	Default	Function	
SIDE	<i>none</i>	Side whose boundary condition is to be set.	
TYPE	<i>none</i>	Type of boundary condition.	
		STRAIGHT CURVATURE	Set the side to be straight Use an expression to set the value for curvature
		CENTRED	Centre the arc on a point
CENTRE	<i>none</i>	Centre point if TYPE=CENTRE	
EXPRESSION	<i>none</i>	Expression for the curvature if TYPE=CURVATURE	

The **CURVATURE** command is used to set the curvature on a polygon side. The curvature can be set to a value calculated from the **EXPRESSION** parameter, or it can be centred on the point given by the **CENTRE** parameter.

All side curvatures are set using the **CURVATURE** command and applied to polygon sides. The side of a polygon is defined between two points of the polygon. The curvature is assigned to the lower node number of the two nodes, unless the side lies between the first and last nodes of a polygon, when the higher node number (last node of the polygon) is assigned the boundary condition.

If the parameter **SIDE** is given no argument, a mouse cursor is shown over the model and the user may assign the curvature, as defined by **TYPE** and **EXPRESSION** or **CENTRE**, to several sides by clicking the mouse cursor on to or close to the appropriate sides. Clicking the cursor twice on a polygon side results in the de-selection of that side. Once all sides have been assigned, pressing **a** will accept the sides and exit mouse cursor mode. The following keystrokes are available under mouse cursor mode.

Keystroke	Function
a	Apply boundary condition to sides and quit
q	Quit and do not apply any boundary conditions

Keystroke	Function (<i>continued</i>)
v	Set or change a variable value
s	Solve the variational geometry
z	Zoom the display
	b Set display to bounding box.
	i Zoom in by factor of 2.
	k Use keyboard input to set display.
	o Zoom out by factor of 2.
	p Pan to new centre.
	r Redraw display.
	q Quit
h	Bring up some help notes

The **DANGLE** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Angle Difference

Command Line Parameters:

Command	DANGLE	
Parameter	Default	Function
P1	<i>none</i>	Name of first point to be constrained.
P2	<i>none</i>	Name of second point to be constrained.
P3	<i>none</i>	Name of third point to be constrained.
P4	<i>none</i>	Name of fourth point to be constrained.
THETA	<i>none</i>	Angle used to define the constraint.

This command sets an angle constraint between a set of four points

The **DANGLE** command constrains four points to be fixed by defining the angle from **P3-P4** relative to the line formed from **P1-P2**. This constraint can be useful in generating sets of parallel lines, by setting **THETA=0**

The parameters **P1**, **P2** refer to the start and end points of the first line, and parameters **P3**, **P4** refer to the start and end points of the second line. The direction of the lines is important when considering the angle of intersection, since a line running in the opposite direction as intended may introduce an error of 180°.

The parameter **THETA** takes either an expression or numeric value as its argument and defines the angle that the two lines generate at intersection.

The **DANGLE** constraint adds one constraint to the constraint list defining the model.

The **MODIFY** command can be used to alter the expression that is used to define **THETA**

The **DELETE** Command

Menu route: MODEL ↓
Delete object

Command Line Parameters:

Command	DELETE		
Parameter	Default	Function	
OBJECT	<i>none</i>	Type of data to be listed.	
		CONSTRAINT	Delete constraints from list.
		POINT	Delete individual points.
		POLYGON	Delete a complete polygon.
CON1	<i>none</i>	Start point of deletion.	
CON2	<i>none</i>	End point of deletion.	

This command is used to delete various types of data from the model. There are three sets of data that can be deleted: constraints, polygons and points.

The **DELETE OBJECT=CONSTRAINT** option deletes constraints from the constraint list. The parameters **CON1**, **CON2** must be specified to indicate which constraints are to be deleted. The parameters **CON1**, **CON2** take the constraint numbers as their arguments. If **CON1=CON2** only one constraint is removed, otherwise **CON1**, **CON2** defines the range from the constraint list to be deleted. Once constraints have been deleted, the numbers associated with constraints may change, and so the user should check constraint numbers before deleting further constraints.

The **DELETE OBJECT=POINT** option deletes individual points of the model. Only one point at a time may be deleted and parameter **CON1** takes the name of that point. Both reference and polygon points may be deleted in this way. If a constrained point is to be deleted, a querying message comes up so as to request confirmation of the point deletion. This is so that the user understands that any constraints attached to the point are to be deleted along with that point. The user should ensure that, if necessary, any deleted constraints are replaced.

The **DELETE OBJECT=POLYGON** option deletes complete polygons and all associated constraints and replications. Care must be taken with deleting master polygons that have been used to generate slave polygons with the **COPY** (see “[The COPY Command](#)” on page 1-22) command. The constraints relating

the master polygon to the slave will be deleted along with the master polygon, thus leaving the slave polygon totally un-constrained. This will not occur should the slave polygon have been re-constrained independently of the master polygon. Other polygons may become partially constrained through the deletion of constraints which have used points on the deleted polygon.

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.

The **DIANGLE** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Internal Angle Difference

Command Line Parameters:

Command	DIANGLE	
Parameter	Default	Function
P1	<i>none</i>	Name of first point to be constrained.
P2	<i>none</i>	Name of second point to be constrained.
P3	<i>none</i>	Name of third point to be constrained.
P4	<i>none</i>	Name of fourth point to be constrained.
P5	<i>none</i>	Name of fifth point to be constrained.
P6	<i>none</i>	Name of sixth point to be constrained.
THETA	<i>none</i>	Angle used to define the constraint.

This command sets an angle constraint between a set of six points

The **DIANGLE** command constrains six points to be fixed such that the difference in angle between the two internal angles formed by the two sets of three points is defined by **THETA**. This constraint can be useful in ensuring that various parts of a model have identical angles by setting **THETA=0**.

$$(P1, P2, P3) - (P4, P5, P6) = THETA$$

The parameters **P1**, **P2**, **P3** refer to the start, intersection and end point names of the two lines that form the first internal angle. Parameters **P4**, **P5**, **P6** refer to the start, intersection and end points names of the two lines that form the second internal angle. The parameter **THETA** defines the angular difference between the two internal angles, and takes either an expression or numeric value as its argument.

The **DIANGLE** constraint adds one constraint to the constraint list defining the model.

The **MODIFY** command can be used to alter the expression defining the angle **THETA**.

The **DISPLAY** Command*

Menu Route: DISPLAY

Command Line Parameters:

Command	DISPLAY	
Parameter	Default	Function
XMIN	0	Lower X-axis coordinate.
XMAX	0	Upper X-axis coordinate.
YMIN	0	Lower Y-axis coordinate.
YMAX	0	Upper Y-axis coordinate.
CONSTRAINT	<i>none</i>	Show constraints.
		NONE Show no constraints.
		ALL Show all constraints.
		POINT Show just POINT constraints.
		VECTOR Show just VECTOR constraints.
		DVECTOR Show just DVECTOR constraints.
		LENGTH Show just LENGTH constraints.
		DLENGTH Show just DLENGTH constraints.
		ANGLE Show just ANGLE constraints.
		DANGLE Show just DANGLE constraints.
IANGLE Show just IANGLE constraints.		
DIANGLE Show just DIANGLE constraints.		
NODES	ALL	Show nodes
		NONE Show no nodes.
		REFE Show reference nodes.
		POLY Show polygon nodes.
		ALL Show all nodes.
FILL	YES	Fill polygons
		YES Fill polygons.
		NO Just show polygon outlines.
<i>continued on next page</i>		

Command	DISPLAY (continued)		
Parameter	Default	Function	
COPY	YES	Show polygon replications	
		YES NO	Show replications. Hide replications.
LABEL	YES	Label polygons	
		YES NO	Label polygons. Hide polygon labels.
BOUND	YES	Show boundary conditions	
		YES NO	Show boundary conditions. Hide boundary conditions.
DIMENSION	ALL	Show dimensions	
		ALL NONE SAME	Show all dimensions. Hide all dimensions. Keep dimensions currently shown
		<i>+name</i>	Show the dimension for this variable
		<i>-name</i>	Hide the dimension for this variable
POLYGON	ALL	Show polygons	
		ALL NONE SAME	Show all polygons. Hide all polygons. Keep polygons currently shown
		<i>+number</i>	Show this polygon
		<i>-number</i>	Hide this polygon.
NAMES	ALL	Show polygon node names	
		YES NO	Show polygon node names. Hide polygon node names.
AXES	ALL	Draw the screen axes	
		YES NO	Show screen axes. Hide screen axes.
		OVER	Show axes over screen.

The **DISPLAY** command allows the user to set the screen display of the model. The main parameters set the minimum and maximum coordinates of the screen axes, thus sizing the modelling. Other parameters allow for more specific data to be revealed or hidden.

The first four parameters (**XMIN**, **XMAX**, **YMIN**, **YMAX**) set the coordinate limits of the picture. If **XMAX-XMIN=YMAX-YMIN**, then the value of **XMAX** is adjusted to match the aspect ratio of the graphics display so that circles appear round. These four parameters may also be set by the **ZOOM** command (see “[The ZOOM Command*](#)” on page 1-116). These four parameters may be stored as expressions.

The parameter that controls which polygons are displayed is **POLYGON**. This parameter offers total flexibility to the user through the use of **POLYGON=SAME**, **+**, **-**. The default value is **POLYGON=ALL**, which displays all the polygons. The value **POLYGON=NONE** hides all polygons. The current polygon selection is maintained if **POLYGON=SAME** is used, but the user may add to or subtract from the current selection by using the **+** and **-** options. For example

DISP POLYGON=ALL	Displays all polygons.
DISP POLYGON=SAME-1-2	Displays all polygons except polygons 1 and 2
DISP POLYGON=SAME-5+2	Displays shows all polygons except polygons 1 and 5, with polygon 2 being reinstated.
DISP POLYGON=NONE+1+3	The display shows only polygons 1 and 3.
DISP POLYGON=NONE+1..4	The display shows polygons 1,2,3 and 4
DISP POLYGON=ALL-3..5	The display shows all polygons except 3,4 and 5

This flexibility is of particular use when setting the display options for a **PART** (see “[The PART Command](#)” on page 1-75).

The use of the **+** and **-** can also be seen when displaying the dimensions (except that polygon numbers are replaced by variable names). For example:

DISP DIMENSION=NONE+LEN+ARC	Hides all dimensions except for variables LEN and ARC .
DISP DIMENSION=SAME-LEN	Hides dimension LEN from the current display.

The polygon can either be seen as filled or outlined. The filled polygons take their colour from their material number. **FILL=YES** fills the polygons, and **FILL=NO** displays polygons outlines.

The replications of a polygon, which are generated by setting the appropriate **MATERIAL** parameter (see “[The MATERIAL Command](#)” on page 1-69), can be viewed by setting **+COPY** or hidden by **-COPY**. Replications are hidden when the

boundary setting command **BOUNDARY** (see “The **BOUNDARY** Command” on page 1-7) is used. The default setting is **+COPY**.

There are two types of nodes that can be displayed - reference and polygon nodes. The **NODE** parameter can hide them all **NODE=NONE**, display them all **NODE=ALL**, only show reference nodes **NODE=REFERENCE** or only show polygon nodes **NODE=POLYGON**. The default is **NODE=ALL**.

NB: Polygon or reference points must be on display if the user wishes to apply constraints to these points in cursor mode (see “The **CONSTRAIN** Command” on page 1-14).

The polygon numbers can be hidden or displayed with the **LABEL** parameter. **+LABEL** displays polygon numbers and **-LABEL** hides polygon numbers. The default is **+LABEL**. The polygon numbers are placed in roughly the centre of the polygon area.

If a polygon has replications, the replications exhibit a slightly different polygon number. The replicated polygon has a 2-part number, the first part of which refers to the parent polygon and the second, contained within square brackets, to the replication number. The replication number increments for each replication generated from a parent polygon and can be used as a regional material property by use of the system variable, **INDEX** (page 1-71).

The boundary conditions are displayed on the polygon faces by a coloured line. The colours represent different boundary conditions. These boundary conditions can be viewed with **+BOUNDARY** or hidden with **-BOUNDARY**.

The polygon node names are displayed by **+NAME** and hidden by **-NAME**. The default is **+NAME**.

The axes that are shown to scale the model can be removed by **AXES=NO**, drawn over the model after redisplaying by **AXES=OVER** or drawn as the first item on the screen **AXES=YES**. The default option is **AXES=YES**, which occasionally means that it is obscured by the model.

The **DLENGTH** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Length Difference

Command Line Parameters:

Command	DLENGTH	
Parameter	Default	Function
P1	<i>none</i>	Starting point of the first length.
P2	<i>none</i>	End point of the first length.
P3	<i>none</i>	Starting point of the second length.
P4	<i>none</i>	End point of the second length.
DISTANCE	<i>none</i>	Length between the first and second lengths.

This command sets the difference in length between two lengths linking 2 sets of points. This constraint is of use, amongst others, in ensuring that two lengths are identical, by setting the parameter **DISTANCE=0**.

The definition of the constraint is that the length difference between the first length linking points **P1**, **P2** and the second length linking points **P3**, **P4** is equal to **DISTANCE**.

$$\mathbf{DISTANCE} = | \mathbf{P1} - \mathbf{P2} | - | \mathbf{P3} - \mathbf{P4} |$$

The parameters **P1-P4** can take either polygon or reference point names as their arguments. The parameter **DISTANCE** takes either expression or numeric values as its argument and defines the length which defines the difference between the first and second lengths. The **DLENGTH** command sets up two constraint expressions, which are added to the constraint list defining the model. The constraint list can be viewed with the **LIST CONSTRAINT** command (see “[The LIST Command](#)” on page 1-66).

The **MODIFY** command can be used to alter the expression used to define **DISTANCE**.

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 redisplay 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 PICOOUT'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).
- nf Do not fill polygons (outlines drawn).

PICOUT OPTIONS (*continued*)

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 **DVECTOR** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Vector Difference

Command Line Parameters:

Command	DVECTOR	
Parameter	Default	Function
P1	<i>none</i>	Starting point of the first vector.
P2	<i>none</i>	End point of the first vector.
P3	<i>none</i>	Starting point of the second vector.
P4	<i>none</i>	End point of the second vector.
XVECTOR	<i>none</i>	X-component of the difference vector.
YVECTOR	<i>none</i>	Y-component of the difference vector.

This command sets the difference in vector between two vectors linking 2 sets of points. This constraint is of use, amongst others, in constraining lines to be parallel with one another by setting the parameters **XVECTOR=0**, **YVECTOR=0**.

The definition of the constraint is that the vector difference between the first vector linking points **P1** to **P2** and the second vector linking points **P3** to **P4** is equal to **XVECTOR**, **YVECTOR**.

$$(\mathbf{XVECTOR}, \mathbf{YVECTOR}) = \mathbf{Vector}(P1, P2) - \mathbf{Vector}(P3, P4)$$

The parameters **P1-P4** can take either polygon or reference point names as their arguments. The parameters **XVECTOR**, **YVECTOR** take either expressions or numeric values as their arguments and define the x and y components of the vector which define the difference between the first and second vectors. The **DVECTOR** command sets up two constraint expressions, which are added to the constraint list defining the model, one for the x-component and one for the y-component. The constraint list can be viewed with the **LIST CONSTRAINT** command (see “[The LIST Command](#)” on page 1-66).

The **MODIFY** command can be used to alter the expressions used to define **XVECTOR** and **YVECTOR**.

The **END** Command*

Menu Route:

FILE ↓
End DEM

Command Line Parameters:

Command	END
No Parameters	

The **END** command stops the *Configuration System*. All data files are closed.

The **EXTERNALCIRCUIT** Command*

Menu Route:

```
MODEL ↓
    External circuits
ANALYSIS ↓
    External 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 1-53
DEFINE	Define circuit data	page 1-46
DELETE	Delete circuit data	page 1-50
EDIT	Edit circuit data	page 1-48
IMPORT	Import circuit data from a SPICE type file	page 1-51
LIST	List circuit data	page 1-50
LOAD	Load circuit data	page 1-50
QUIT	Quit the EXTERNAL CIRCUIT command	page 1-54
STORE	Store circuit data	page 1-51

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	none	Number of first circuit to be deleted
C2	none	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	none	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	none	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 of 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 **GETOP2** Command

Menu Route:

File ↓
 Load .OP2 data file → Append OP2 data to DEM

File ↓
 Load .OP2 data file → Replace DEM with OP2 data

Command Line Parameters:

Command	GETOP2	
Parameter	Default	Function
NAME	<i>none</i>	Name of the <i>.OP2</i> file to be imported.
APPEND	NO	Mode of data retrieval.
		NO Overwrite and replace existing DEM data.
		YES Append <i>.OP2</i> data to existing DEM data

This command is used to import a *.OP2* data file. The data may be appended to data already existing within the Configuration System, or it may overwrite and replace the current DEM data. The geometrical and material data are recreated exactly, but the mesh data may deteriorate.

The geometry of the model is fully constrained automatically, with each node being given a **POINT** (see “The **POINT** Command” on page 1-80) constraint. The expressions used to define the constraint values are numeric and are the values taken directly from the *.OP2* data file. The user must delete constraints or modify constraint values if the topology is to be changed.

The material data is also recreated using the same data. The values from the *.OP2* data file are used as expressions directly.

The boundary conditions are recreated too. Any **EXTRA** boundary conditions (see “The **BOUNDARY** Command” on page 1-7) or **EXTRA** material conditions (see “The **MATERIAL** Command” on page 1-69) that have been used in the *.OP2* file are also recreated and applied to the relevant polygons.

The mesh data changes, in that the DEM does not hold specific facet data, rather holds the mesh data on a regional basis. Hence the subdivisions set to a region are an average of the subdivisions around the region boundary as defined in the *.OP2* file.

Replications are recreated through the generation of reference points to define centres of rotation or the start and end points of mirror lines. The reference points are given **POINT** type constraints. The angles or rotation, numbers of replications etc. are assigned to the material parameters of the replicated regions. These can be altered by using the **MATERIAL** command (see “**The MATERIAL Command**” on page 1-69).

The **GRAVITY** Command

Menu Route:

MODEL ↓
 Geometry → Mesh control point

Command Line Parameters:

Command	GRAVITY	
Parameter	Default	Function
PT	<i>none</i>	Polygon or reference point name at which mesh size is specified.
SUBDIVISION	1	The size of element at point PT
RAD0	1	The radius at which the size of the subdivision has doubled.
POLYGON	0	The polygon to which the point is associated (for replication purposes)
OPTION	LIST	Option for listing or deleting mesh control points.
		LIST List the mesh control points applied.
		DELETE Delete mesh control points
	NONE	Do not list or delete mesh control points.
N1	1	Start point for listing or deleting.
N2	*	End point for listing or deleting.

This command is used to define a particular mesh density at a specific point of interest. Normal usage of this command ensures that a particular area will receive increased mesh elements, with a gradual return to the regional mesh density further away from the point of interest. The term ‘gravity’ describes the manner in which the mesh control point increases the mesh density nearby.

The parameter **PT** defines the point name of the mesh control point. This can either be a reference or a polygon point. If a value of zero (**PT=0**) is used, a new reference point is generated, which needs constraining, but which has mesh data assigned to it.

The parameter **SUBDIVISION** sets the mesh element size at the point. This size can be defined as a parametric expression or a straight numeric value.

The area of influence of the mesh control point is defined by the parameter **RADO**. This value determines the radius at which the subdivision size has doubled. The expression used for determining the mesh size, s_r , at a distance r from the mesh control point is

$$s_r = s_0 \left(1 + \frac{r}{r_0} \right) \quad (1.3)$$

where s_0 is the subdivision size at the mesh control point, **SUBDIVISION**, and r_0 is the radius, **RADO**.

The **POLYGON** parameter automatically takes its value from the **PT** name if a polygon point is specified. However, if **PT** takes a reference point name, **POLYGON** can be left undefined or take a polygon number. This polygon number is useful if the user wishes the mesh control point to be replicated in a similar way to a neighbouring polygon.

The **OPTION** parameter allows the deletion or listing of mesh control points. The range of listing or deletion is defined by the index numbers **N1** and **N2**.

The IANGLE Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Internal Angle

Command Line Parameters:

Command	IANGLE	
Parameter	Default	Function
P1	none	Name of first point to be constrained.
P2	none	Name of second point to be constrained.
P3	none	Name of third point to be constrained.
THETA	none	Angle used to define the constraint.

This command sets an angle constraint between a set of three points.

The IANGLE command constrains three points to be fixed such that the two lines linking points P1-P2 and P2-P3 form the specified angle THETA. The point P2 can be considered to be the intersection point of the two lines.

The parameter P1 refers to the first point name that is to be constrained, P2 refers to the second point name and P3 refers to the third point name. The parameter THETA takes either an expression or numeric value as its argument and defines the angle linking the three points.

The IANGLE constraint adds one constraint to the constraint list defining the model. The constraint list can be viewed with the LIST CONSTRAINT command (see “The LIST Command” on page 1-66).

The MODIFY command can be used to alter the expression that defines THETA.

The **INSERT** Command

Menu Route:

MODEL ↓
 Geometry → Insert point in polygon

Command Line Parameters:

Command	INSERT	
Parameter	Default	Function
SIDE	<i>none</i>	Side in which point is to be inserted.

This command is used to insert a point into a polygon so as to increase the number of nodes and sides of a polygon by one.

Points may be inserted into any polygon in any type of constraint status. The parameter **SIDE** states the position, which the inserted point will occupy. All points of the polygon, before which the new point is inserted, will have their point names incremented by 0.01 and the constraints will update the point names to be consistent with the new point names. No constraint definitions are lost, but the polygon will contain a new totally un-constrained point. To demonstrate this, the inserted point is initially placed at a point such that its position can be seen to be un-constrained. Any curvature or boundary data between nodes that have been split up by an inserted node are removed and subsequently need to be redefined.

The LENGTH Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Length

Command Line Parameters:

Command	LENGTH	
Parameter	Default	Function
P1	none	First point to be constrained.
P2	none	Second point to be constrained.
DISTANCE	none	Distance between points P1, P2.

This command sets a length constraint between two points.

The **LENGTH** command constrains a particular point to be fixed at a certain distance from another point. This type of constraint is of particular use when a model has a rotational geometry, as this constraint defines the locus of a circle about which a point can circle a fixed point. This constraint is generally used in conjunction with a **ANGLE** constraint. Used on its own, this constraint can lead to multiple geometry solutions.

The definition of the constraint is:

$$| P1 - P2 | = DISTANCE$$

The parameters **P1**, **P2** refer to the point names, which will be placed a distance apart. **DISTANCE** takes either expressions or numeric values as its argument and defines the separation distance.

The **LENGTH** command sets up one constraint expression, which is added to the constraint list defining the model.

The **MODIFY** command can be used to alter the value or expression that is used to define the length between the points **P1** and **P2**.

The **LIMIT** Command

Menu Route:

MODEL ↓
Variables → Limit variables values

Command Line Parameters:

Command	LIMIT	
Parameter	Default	Function
EXP1	<i>none</i>	First variable or expression.
OPERATOR	<i>none</i>	Type of limiting constraint.
		LT Less then limit.
		LE Less than or equal to limit.
		GT Greater than limit.
		GE Greater then or equal to limit.
		NE Not equal to limit.
EQ Equal to limit.		
EXP2	<i>none</i>	Second variable or expression.
STRING	<i>none</i>	Error string message to be displayed if the limit is violated
DELETE	NO	Delete limits.
		YES Delete limits between N1 and N2
		NO Do not delete limits.
N1	<i>none</i>	First limit number.
N2	<i>none</i>	Second limit number.
LIST	NO	List limits.
		YES List limits N1 to N2
		NO Do not List limits.

This command is used to place limits on the values that a variable or expression can take. This limit can be used to prevent unstable geometry solutions or to prevent geometries that are physically impossible. The limits are further useful in restricting the degree of change which can be performed on a topology by a novice user.

If a variable is changed in such a way as to exceed a set limit, a warning message is posted to the screen. The variable which had just been changed will then be reset to its old value. Hence no changes to the geometry or the material properties

will result if a limit is exceeded. A variable may not take on a value that results in an exceeded limit.

No limit may be set if its condition at the point of setting is exceeded. The variables or expressions causing the proposed limit to exceed need to be adjusted before the limit can be properly set.

The limits that have been used in a DEM may be viewed by setting the **LIST=YES** parameter. The limits are listed in a numbered order. The number assigned to a limit is used if particular limits are to be deleted. Deleting limits is achieved by setting the parameter **DELETE=YES** and assigning the appropriate limit numbers to **N1** and **N2**.

The LINK Command

Menu route:

MODEL ↓
 Constraints → Link sides

Command Line Parameters:

Command	LINK	
Parameter	Default	Function
P1	none	Master side to be linked
P2	none	Slave side to be linked

The link command generates a link between 2 sides. In doing this it constrains the end nodes of the slave side to be fixed to the end nodes of the master side. The curvature and boundary conditions on the slave side will be taken from the master side.

The start node of P1 will not always match the start node of P2. Instead, the sides will be matched to minimise the motion of the end nodes of the slave side. This may mean that the start node of P1 will be linked to the end node of P2. Once a link has been created, this order will not change.

The two end nodes on the slave side do not need to be constrained using the standard constraining techniques as the nodes have no freedom of movement. Similarly the curvature of the slave side cannot be set.

Links can be listed by listing polygon side data, **LIST OBJECT=SIDE**

Links can be removed from a side by setting **P1=P2** for the side that is to have the link removed.

If no sides are given for P1 and P2, the sides can be selected by cursor. Options are available are:

Command	LINK
Keyboard	Function
<space>	Selects a side for linking
h	Generates a help message
q	Quits the link command

Selecting two sides with the **<space>** commands generates a link between those two sides. If the same side is selected twice with **<space>** any links on that side are deleted.

The **LIST** Command

Menu route: MODEL ↓
List objects

Command Line Parameters:

Command	LIST		
Parameter	Default	Function	
OBJECT	<i>none</i>	Type of data to be listed.	
		CONSTRAINT	List defined constraints.
		CONP	List polygon constraint data.
		CONR	List reference point constraint data.
		POLYGON	List polygon status and geometry data.
		REFERENCE	List reference point status and geometry data.
		VARIABLE	List defined variables.
	SIDE	List polygon side characteristics.	
	MATERIAL	List polygon material definitions.	
	REPLICATION	List polygon replication definitions.	
N1	<i>none</i>	Start point of list.	
N2	<i>none</i>	End point of list.	

This command is used to list out various types of data that define the model. There are seven sets of data that can be listed: constraint, polygon, reference points, variable, side, material and replication data.

The **LIST OBJECT=CONSTRAINT** option lists the constraints in the order in which the user has applied the constraints. An important aspect of the list is that a number is assigned to each constraint. This number is required when deleting or modifying specific constraints. The constraint types and arguments are listed.

The **LIST OBJECT=CONP** options lists the constraint numbers, constraint descriptions, points used for the constraints and the expressions assigned to the constraints for a single or range of polygons. This option is useful in locating the constraint numbers for particular polygon points for modification purposes.

The **LIST OBJECT=CONR** option lists the constraint numbers, constraint descriptions, points used for the constraints and the expressions assigned to the constraints for all the defined reference points. This option is useful in locating the constraint numbers for particular reference points for modification purposes.

The **LIST OBJECT=POLYGON** option lists the points of the polygons in numeric order, gives the constraint status (un-, partially, fully and over-constrained), gives constraint numbers used to calculate the position for each polygon point and states the (x,y) coordinates that the points currently occupy. This list is generated for all polygons defined between the parameter limits **N1, N2**.

The **LIST OBJECT=REFERENCE** option is similar to the **POLYGON** option, except that the data listed is given for the defined reference points. Hence constraint numbers, constraint status and (x,y) coordinates are given. All reference points are listed with this option.

The **LIST OBJECT=VARIABLE** option lists the variables defined and gives their names, text labels and their expressions or values. The variables are listed in alphabetic order for ease of identification.

The **LIST OBJECT=SIDE** option lists all the facets of the polygons in numeric order, gives any boundary conditions attached and the data used to define and set its curvature. If a side has no boundary condition and is straight, no data is listed for that side. The list is generated for all polygon sides, where the parameters **N1, N2** define which polygons are to have their sides listed.

The **LIST OBJECT=MATERIAL** option lists the definitions of the material characteristics for the selected polygons. Any expressions which have been used to define a characteristic such as **DENSITY** are displayed. All the material properties as well as the **INCLUDE** parameter have their values displayed. The list is generated for all polygon, where the parameters **N1, N2** define which polygons are to be listed.

The **LIST OBJECT=REPLICATION** option lists the definitions of the polygon replications for the selected polygons. Any expressions which have been used to define a characteristic such as **NX, DX** are displayed, or as in the case of parameters **P1, P2, CP**, the point name is shown. All the replication definitions have their values displayed. The list is generated for all polygon, where the parameters **N1, N2** define which polygons are to be listed.

The **LOAD** Command*

Menu Route:

FILE ↓
Load DEM file

Command Line Parameters:

Command	LOAD	
Parameter	Default	Function
FILE	<i>none</i>	The DEM filename.
APPEND	NO	Append DEM data option.
		YES Append loaded DEM data to current data.
		NO Overwrite existing data with loaded DEM data.

This command is used to **LOAD** a DEM data file. All the data that makes up the model such as constraints, variables, material settings, expressions etc., are stored in the DEM file.

The parameter **FILE** is the DEM filename. There are two modes of loading data, which are controlled by the parameter **APPEND**. If **APPEND=NO**, any existing DEM data is overwritten and all data will be cleared. If **APPEND=YES**, the loaded DEM data is appended to existing data. Not all data is appended, however. All post processing data, unit definitions and solution data from the incoming DEM are ignored. Variables which are already in existence will not be up-dated by variables with the same name in the incoming data. Likewise, BH material names that have been defined and share the same name as BH data in the incoming data will not be up-dated. PART data will be appended, as will the appropriate PART menuing.

The MATERIAL Command

Menu Routes:

MODEL ↓
 Material data
 MODEL ↓
 Geometry → Replicate polygon

Command Line Parameters:

Command	MATERIAL		
Parameter	Default	Function	
POLYGON	<i>none</i>	Polygon identification number	
PROPERTY	AIR	Material property:	
		AIR	Air
		CONDUCTOR	Conductor
		IRON...	Non-linear material
SHAPE	POLYGON	POLYGON or BACKGROUND region.	
SUBDIVISION	1	Maximum subdivision size.	
PERMEABILITY	1	Relative permeability or permittivity	
CONDUCTIVITY	0	Conductivity.	
DENS	0	Current or charge density.	
PHASE	0	Angle in degrees.	
		Direction of permanent magnet (PF, ST and TR)	
		Direction of normal to laminations (PF)	
		Phase angle of drive currents (AC)	
		Phase angle of complex permeability (AC)	
VEL	0	Velocity of moving parts.	
N	0	Conductor number.	

Command	MATERIAL (continued)		
Parameter	Default	Function	
SYMMETRY	CONNECTED	Eddy current conductor symmetry:	
		CONNECTED	Connected conductors: program calculates total current.
		DISCONNECTED	Disconnected conductors: total current set by DENSITY.
NX	0	Number of replications with X displacement.	
DX	0	X displacement between replications.	
NY	0	Number of replications with Y displacement.	
DY	0	Y displacement between replications.	
REFLECT	NO	Mirror reflection switch.	
		NO	No mirror copy.
		YES	Copy reflected in mirror plane.
P1	none	Reflect along line starting at P1.	
P2	none	Reflect along line ending at P2.	
NR	0	Number of replications with rotational displacement.	
CP	none	Point about which rotation occurs.	
THETA	0	Angle between rotational replications.	
INCLUDE	1	Include polygon in OP2 data	
		>0	Positive value includes the polygon in op2 data file
		<0	Negative value excludes the polygon from op2 data file

The **MATERIAL** command is the material data definition command for the Configuration System. It has two sets of parameters, which relate to material properties and geometry replications. The polygon can be selected by setting the POLYGON parameter. If no value is given for this the polygon to be modified or set can be selected by cursor from those displayed on the screen.

Command	MATE
Keyboard	Function
<space>	Selects a polygon from the cursor location
h	Generates a help message
q	Quits the MATERIAL command

After selecting a polygon, the previously set values for the polygon are loaded into the parameter values. Any value that is set explicitly on the command line will overwrite a previously set value. The values of any of the parameters can then be set. The command quits when the command line sets no parameter values.

Material properties

The material **PROPERTY** takes a character string as its argument. This string is used as the pointer to a specific BH characteristic. If the **PROPERTY** name does not exist, a new pointer is created with the new name. Up to 10 material names are available to point to 10 BH curves. The BH characteristics may be defined with the **BHDATA** command (see “[The BHDATA Command*](#)” on page 1-3).

The finite element mesh is defined on a regional basis with the setting of the **SUBDIVISION** parameter. This parameter sets the maximum size of the elements used within the region. Hence a small value of **SUBDIVISION** will result in small element sizes and hence a greater number of elements within the region. Conversely, a large value of **SUBDIVISION** will result in large sized elements and hence a smaller number of elements within the region. This results in a uniform mesh density over the region. If further mesh refinement is required in a particular section of the region, the use of ‘mesh control points’ (see “[The GRAVITY Command](#)” on page 1-57) will increase the local mesh density.

The material parameters can all take either numeric values or expressions as their values. As well as this, the expressions are able to use the following system variables:

The system variable **INDEX** takes its value as the replication number. A parent polygon will have **INDEX=0**. The first replication of a polygon will have

INDEX=1 and so on. The **INDEX** value can be seen as part of the polygon labeling scheme. For example:

the region label 1[2] refers to polygon 1 replication 2, i.e. **INDEX=2**.

The **INDEX** variable can be used in the material properties to allocate varying properties to replications, which would otherwise be identical. For example:

MATE 3 PERM=1+(INDEX*1000) will set the permeability of polygon 3 as a function of **INDEX**. Hence the parent polygon will have **PERM=1**, the first replication will have **PERM=1001** etc.

The **INDEX** variable could also be used to set the **SUBDIVISION** parameter and hence generate different mesh densities between replicated regions.

The variable **AREA** takes its value as the area of the polygon. This is useful when setting the parameter **DENSITY**, as the density is automatically calculated. For example:

MATE 2 DENS=10/AREA sets the current density of region 2 given the total current is 10

The variables **X**, **Y** can be used as part of the expressions for setting the **DENSITY**, **PERMEABILITY**, **PHASE**, **CONDUCTIVITY** parameters. Using these variables implies a variation of a property as a function of the geometry. These can be set in OPERA-2d as an **EXTRA** condition (see *OPERA-2d Reference Manual*). The DEM takes the expressions with **X**, **Y** and automatically assigns it as an **EXTRA** condition. For example:

MATE 2 COND=2*X*Y sets the conductivity of region 2 as twice the product of the X and Y coordinates at any particular point of that region.

Replications

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

The replication parameters allow a region to have multiple copies, with each copy having the same material properties (except when the **INDEX** variable is used, see [page 1-71](#)) and boundary conditions. There are three types of replication, which are applied in the following order: translation in X and Y, mirror imaging and rotations. The **COPY** command also creates copies of regions; new regions

created by **COPY**ing can be modified to have different material properties, geometries and boundary conditions (see “The **COPY** Command” on page 1-22 and “The **MODIFY** Command” on page 1-74).

The translations 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. Reflected images are selected with **+REFLECT**. The geometry is reflected in a plane defined by the two points **P1** and **P2**. **P1** and **P2** can be either polygon nodes or reference points.

Rotational images are defined by the number of copies (**NR**), the azimuthal displacement between each copy (**THETA**) and the centre point **CP** of rotation. The centre point **CP** may be either a reference point of a polygon node.

The *Design Environment* treats replicated polygons as though they are independent polygons, when generating the *OP2* data file (see “The **CONTROL** Command*” on page 1-16). The geometries and material data of the polygons remain identical to the data of the replications. Hence if the *OP2* data file is read into the OPERA-2d pre processor, the user will note that all polygons are independent of each other.

The replication parameters can take expressions as well as numeric values, and so can participate in the parameterisation. For example a motor can be built which exhibits a variable number of poles by setting the parameter **NR=NUMPOL**, where **NUMPOL** is a user variable specifying the pole number.

The **INCLUDE** parameter includes or excludes the polygon depending on its value. If **INCLUDE > 0** the polygon and its replications are included in the *.OP2* data file. If **INCLUDE ≤ 0** the polygon and all its replications are excluded from the *.OP2* data file. The polygon is subsequently displayed, using the **DISPLAY** command (see section “The **MATERIAL** Command” on page 1-69), as just an outline if **INCLUDE ≤ 0**.

The **MODIFY** Command

Menu Route:

MODEL ↓
 Constraints → Modify constraint

Command Line Parameters:

Command	MODIFY	
Parameter	Default	Function
CONSTRAINT	<i>none</i>	Number specifying constraint to be modified.
VALUE	<i>none</i>	New value or expression assigned to constraint.

This command is used to modify the values or expressions that have been used to define a constraint. The command cannot change the points to which a constraint has been applied.

The parameter **CONSTRAINT** takes a number as its argument. This number is specific to a particular constraint and can be found by listing the constraints (see section “[The LIST Command](#)” on page 1-66) and noting the number given for the desired constraint.

The parameter **VALUE** is used to redefine the constraint’s argument. The parameter can take expressions or numeric values. The geometry may need to be resolved before the new constraint value take effect.

The PART Command

Menu Route:

DEM ↓

Create or modify PARTs → Create new PART

Command Line Parameters:

Command	PART	
Parameter	Default	Function
NAME	none	Name of the part to be created or altered.
UPDATE	NO	Update display associated with PART:
		NO Keep the display as presently associated with PART.
		YES Update associated display with current display settings.

The **PART** command is central to running the *User System* as it sets up the environment in which the non-expert works. A PART is defined by the expert to be a single variable or a collection of variables that together define a significant design area of the overall model. Up to 15 PARTs may be defined for any particular DEM.

For example a motor may have the rotor and stator defined as two independent parts, each of which can contain several design parameters.

The expert user is able to specify which variables and parameters are included within the PART, and hence control the design combinations available to the novice.

Each part has a set of **DISPLAY** parameters, which are used to set the display whenever a particular part is run. The parameter **UPDATE=YES** records the current **DISPLAY** parameters, such as **XMIN, XMAX,...**, the selected polygons **POLY= . .** etc., for the **PART**. Any previous display parameters are over written. Expressions as well as numeric values may be used for the **DISPLAY** parameters, as these will all be recorded by the **UPDATE=YES**. Expressions which govern the display will automatically resize the display if polygons in the part change geometry. **UPDATE=NO** will maintain the **PART** display parameters as previously set. The use of the dimension parameters of the **VARIABLE** command can be applied to the variables that are adjustable by the novice user in the part. This can be helpful to the novice since the parameters whose values are to be adjusted will be

clearly identifiable. There is no need to set the **DIMENSION** parameter of the **DISPLAY** command for the required variables, since these will be automatically included as part of the display if the variables have had their dimension lines defined and if they have been requested by the **ASKVAR** sub-command.

Once the part name has been specified, a new set of menu options become available to the user. These options are the means by which the part is configured and the variable questions set. There are four types of options that can be set in the *User System*. The first type enables a user to specify a BH data file to a non-linear material, the second enables the user to assign an expression to a variable, the third enables the user to assign either 0 or 1 to a variable through toggle buttons and finally the fourth gives the user some useful information.

In command line input mode, a sub-group of commands come available once the **PART** command has been invoked. The user remains in these sub-command environment until **QUIT** is specified. These commands can be used in a variety of data input modes - add, insert, and replace. These modes all use the same commands as listed below, but differ in the use of a command number that can be specified as a prefix to the command and parameters.

ASKBH

Sub-command	ASKBH	
Parameter	Default	Function
MATERIAL	<i>none</i>	Material requiring BH data e.g. NON1

This sub-command requests the user to supply a filename containing BH data to be attributed to a particular non-linear material. The expert user has to specify which material requires the BH data, and the text to pose the request to the novice. The novice user supplies the appropriate data file in the User System.

ASKVAR

Sub-command	ASKVAR	
Parameter	Default	Function
VARIABLE	<i>none</i>	Name of variable whose expression is to be set

This sub-command requests the novice user to supply a value or expression to a particular design parameter. The user specifies the variable to be set in the User

System. Once the variable has been requested it will appear on the **PART** display if it has been defined as a dimension. A variable may only appear in a single part.

ASKYESNO

Sub-command	ASKYESNO	
Parameter	Default	Function
VARIABLE	<i>none</i>	Name of variable whose expression is to be set to 1 or 0

This sub-command generates a toggle option that gives a boolean value to a variable of the expert's choosing. The part menu supplies a **YES** or **NO** option from this command, generating a value for the variable of 1 or 0 respectively.

TEXT

Sub-command	TEXT	
Parameter	Default	Function
TEXT	<i>none</i>	Give the user a line of information/data/help/hints

This command allows some advice to be added into the part menu structure. The text exists for information purposes only.

LIST

Sub-command	LIST
No parameters	

As an aid to setting up the options in the part, the expert user can review the options chosen by using the **LIST** sub-command. This lists out all the options and their associated parameter values.

QUIT

Sub-command	QUIT
No parameters	

In order to quit the **PART** command, the user must use the **QUIT** sub-command. This will cause the display to change to the parameters as set before the user entered the **PART** command.

The maximum number of commands that can be attached to a part is 20, of which at most 15 can be variable value requests. The attached commands are built up dynamically into the GUI and become part of the menuing environment employed by the novice in the *User System*.

The **PART** options permit the user to add, insert, replace and delete commands to the part. Each command attached to a part is allocated a number, which corresponds to the position it holds if the commands are listed using **LIST**. This command number is used to delete, replace or insert commands.

Each of the **PART** commands must have all parameter values assigned for the command to function. Incomplete **PART** commands are rejected

Add **PART** Command

If the add option is used, no command number need be specified and the added command is appended to the end of the command list. For example to add a **ASKVAR** command to the **PART** requires the user to enter:

```
ASKVAR LENG <Return>
```

Insert **PART** Command

To insert a command, the user must specify the line position at which the command is to be placed and then type of command plus any parameters. For example, to insert a **ASKVAR** at line 3 requires the user to enter:

```
3 ASKVAR VARI <Return>
```

Replace **PART** Command

To replace a command, the user must place a minus sign before the line number of the command to be replaced, state which command is to be used and set the necessary parameters. For example to replace a command at line 2 with another, the user must enter:

```
-2 TEXT 'This command is a replacement' <Return>
```

Delete **PART** Command

To delete a command, the user must place a minus sign before the line number of the command to be removed. For example to delete line 4 from a list of commands requires the user to enter:

```
-4 <Return>
```

The GUI menuing system simplifies the procedures by offering the add, insert and delete as specific menu options and automatically manipulating the line numbers as necessary.

The **POINT** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Fixed Point

Command Line Parameters:

Command	POINT	
Parameter	Default	Function
NUMBER	<i>none</i>	Polygon point to be constrained.
XPOS	<i>none</i>	X-position for the point.
YPOS	<i>none</i>	Y-position for the point.

This command sets a constraint on a polygon point. The constraint can be considered to be the most straight-forward to use, since its usage resembles standard cartesian coordinates.

The **POINT** command constrains a particular point to be fixed at a (x,y) cartesian point position with respect to the origin. All parametric models must have at least one point constrained using the **POINT** command, so as to fix the model at some distinct point in space. This effectively prevents free body motion.

The parameter **NUMBER** refers to the polygon point name that is to be constrained. The parameters **XPOS**, **YPOS** take either expressions or numeric values as their arguments. The **POINT** command sets up two constraint expressions, which are added to the constraint list defining the model. The constraint list can be viewed with the **LIST CONS** command (see “[The LIST Command](#)” on page 1-66).

The **MODIFY** command can be used to alter the values or expressions that are used to define the point’s x and y coordinates.

The **POLYGON** Command

Menu Route:

MODEL ↓
 Geometry → New Polygon

Command Line Parameters:

Command	POLYGON	
Parameter	Default	Function
NP	<i>none</i>	Number of sides to the new polygon.

This command is used to create a polygon with **NP** unconstrained points. The polygon is given default material values of air.

The new polygon is sized so as to fit the screen reasonably. The coordinates assigned to the polygon points are arbitrary, but will remain until constraints are assigned to the individual points so as to cause them to move to different positions.

The **SKETCH** command can be used instead of the **POLYGON** command to create a polygon that resembles the required shape before constraining. See “[The SKETCH Command](#)” on page 1-93.

All polygons, except for replicated polygons, must be initially generated with this command. New points can be inserted into a polygon using the **INSERT** command (see “[The INSERT Command](#)” on page 1-60). The minimum number of points for a polygon is 3, and the maximum number of points that a polygon can have is 99.

The **POSITION** Command

Menu Route:

MODEL ↓
 Geometry → Temporary point position

Command Line Parameters:

Command	POSITION	
Parameter	Default	Function
POINT	<i>none</i>	Point whose position is to be temporarily set.
XP	<i>none</i>	X-coordinate of the point.
YP	<i>none</i>	Y-coordinate of the point.

The **POSITION** command is used to temporarily set the coordinate position of a point. The coordinates override any constraints defining the point, until the geometry is re-solved. This command is of particular use if the variational geometry has difficulty finding the correct solution, as moving some points closer to the expected coordinates may aid the geometry solvers. However, if the solvers fail, it is better to change the constraint scheme to a non-ambiguous set.

Once the geometry is re-solved, points will return to the positions as defined by the constraints, unless the points are un-constrained.

The parameter **POINT** takes a node name as its argument, and the parameters **XP**, **YP** take expressions.

The **POST** Command

Menu Route: DEM ↓
Post-processing

Command Line Parameters:

Command	POST	
Parameter	Default	Function
OPTION	LIST	Specify which mode post processing command is used.
	ADD	Add a post processing command
	CHECK	Checks post processing commands N1 to N2
	DELETE	Delete a post processing command N1 to N2
	INSERT	Insert a post processing command at N1 .
	LIST	List post processing commands N1 to N2 .
	MODIFY	Modify post processing command N1 .
	WRITE	Lists full post processing <i>COMI</i> file.
N1	<i>none</i>	Start post processing command number
N2	<i>none</i>	End post processing command number
COMMAND	<i>none</i>	Post processing command to be added

A feature of the *Configuration System* is the ability to set up a post processing command script (*.comi*) file that will run in the post processor and automatically run through a series of commands, thus enabling a novice user to use the post processor without getting involved in needing to learn to use the post processor. The post processing commands are generated automatically at the same time as the generation of the OPERA-2d pre processing files.

The generation of the command scripts occurs when the CONTROL command is executed to generate the different files needed for analysing and post processing an analysis control set. See “[The CONTROL Command*](#)” on page 1-16.

The post processing command in the *Configuration System* permits an expert user to set up post processing which is attached to a parameterised model and reflects all the changes in the models geometry. Hence changes in the geometry of the

model will be reflected in the cartesian coordinates required by the post processing.

When running the **POST** command, the command is controlled by the **OPTION** parameter

- **ADD**

Post processing commands that are added are appended to the bottom of the list of commands already present. This is achieved by setting **OPTION=ADD** and **COMMAND** to the required post processing command. When this has been done, the command-line prompt changes and the parameters associated with the command are shown with their default values. The user is then able to specify particular parameters and set their values in the standard way. Once the user is satisfied with the command, pressing **<Return>** at the prompt will cause the command to be entered into the list and the Configuration System prompt to return, so as to enable the user to specify another command. For example, to set a **RECO** post processing command requires:

```
POST OPTION=ADD COMM=RECO <Return>
XMIN=0 XMAX=20 YMIN=0 YMAX=20 +MESH <Return>
<Return>
```

to Configuration System prompt

If the user wishes to cancel a post processing command during the parameter assignation, setting the option **+CANCEL** cancels the addition of the command.

- **CHECK**

Some checking of the post processing commands takes place automatically to ensure consistency. For example error messages are displayed if a **\$ IF** statement does not have a matching **\$ END IF** command.

Individual or groups of commands can be checked by using the **CHECK** option. The parameters **N1** and **N2** take the start and end numbers of the commands to be checked. For example commands 5 to 12 of a list of post processing commands can be checked by entering:

```
POST OPTION=CHECK N1=5 N2=12 <Return>
```

- **DELETE**

Post processing commands can be deleted individually or as a batch. The parameter **OPTION=DELETE** needs to be set as well as values given to **N1** and **N2**, which define the commands to be deleted. For example, deleting command 4 requires:

```
POST OPTION=DELETE N1=4 N2=4 <Return>
```

Deleting commands 6 to the end requires:

```
POST OPTION=DELETE N1=6 N2=* <Return>
```

- **INSERT**

Post processing commands can be inserted into a particular position in the command list by setting **OPTION=INSERT**, setting the parameter **N1** to the position and setting **COMMAND** to the appropriate command. For example to insert a **RECO** command at line 7 of the command list requires:

```
POST OPTION=INSERT N1=7 COMM=RECO <Return>
XMIN=10 XMAX=20 YMAX=10 <Return>
<Return>
```

- **LIST**

All commands that are entered are placed into an ordered list. This list can be viewed to aid modifying, replacing or deleting of individual commands. To view the list, the user must enter:

```
POST OPTION=LIST N1=1 N2=* <Return>
```

The parameters **N1**, **N2** can take other values so as to list only parts of the total command list.

- **MODIFY**

Post processing commands that have been added, but need adjustment can be modified by setting parameters **OPTION=MODIFY** and **N1** to the line number of the command requiring modification. Once these have been entered, the command to be adjusted is brought up with its parameters and the user is free to adjust the parameters in the standard way. Pressing **<Return>** at an empty prompt will return the user back to the main Configuration System prompt. For example to modify a **RECO** command at line 5 requires:

```
POST OPTION=MODIFY N1=5 <Return>
XMIN=1.5 YMAX=2.8 <Return>
<Return>
```

- **WRITE**

An alternative is to list the post processing command script file as it will appear. This is achieved by entering:

```
POST OPTION=WRITE <Return>
```

Post Processing Commands

The post processing commands that can be set up within the configuration system are a subset of the commands available within the post processor. The commands are not executed immediately, but are stored in order and used to generate a command script that will be executed by the post processor after the model has been

analysed using the relevant analysis module (See “The SOLVE Command*” on page 1-96)

The post processing commands in the *Configuration System* have been adapted to permit the use of point names rather than x,y coordinates, but they are identical to the OPERA-2d commands in all other respects.

Key post processing results may be written to a results file automatically by using the RESULTS post processing command. The results file has the .csr extension and takes its stem from the analysis control set (See “The CONTROL Command*” on page 1-16.).

The post processing commands available to the *Configuration System* user are listed below. They are identical to the ones found in OPERA-2d and more in-depth data on the usage of these commands can be found in the *OPERA-2d Reference Manual*. The commands are stored in a numbered list, with the number being used as a reference to the individual commands.

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

\$ Commands	
\$ASK	Asks the user to supply a value for a user variable
\$ASSIGN	Assigns a format type to a data item for writing

\$ Commands (continued)	
\$BACKSPACE	Rewind or backspace an open file
\$CD	Change directory
\$CLOSE	Closes an open stream for reading or writing
\$COMINPUT	Executes the commands in the <i>COMI</i> file
\$CONSTANT	Sets the value of a user variable to a fixed value
\$DO	Sets up a DO loop
\$ELIF	Alternative conditional execution command
\$ELSE	Conditional execution command
\$END	Closes DO , FOR , IF or WHILE loop
\$FOR	Sets up a FOR loop
\$FORMAT	Generates a format type for writing data
\$IF	Conditional execution command
\$OPEN	Opens a file on a stream for reading or writing
\$OS	Executes an operating system command
\$PARAMETER	Sets the value of a user variable to be an expression
\$PAUSE	Pauses execution
\$PROMPT	Prompts the user to define a string
\$READ	Reads the value of a variable from a file
\$STRING	Sets the value of a string variable
\$WHILE	Sets up a WHILE loop
\$WRITE	Writes data to an open file

Post processor management and display commands	
COLOUR	Sets the colour for the display
DUMP	Write a picture file containing the current display.
RECONSTRUCT	Display the model
SET	Sets post processing options for field evaluation
TITLE	Control picture titling.
UNIT	Select units for display and evaluation.

DEM specific command	
RESULTS	Write results to a control set results (.CSR) file.

Points to note for the post processing include the following:

- Point parameters, **PNT1**, **PNT2**,..., take precedence over any X and Y coordinate settings that may have been assigned. The point parameters, which are

unique to the DEM, **PNT1**, **PNT2**,... always precede the X, Y coordinates that the points set.

- Some of the post processing commands contain parameters to specify a range of regions or single region to be processed. These parameters have been amended so that they can take either material names or polygon numbers.
- If a material name is given as the parameter value, the regions which have been defined as that material are found. The post processing command is then issued, as many times as is necessary, to cover all occurrences of the material. For example

```
POST OPTION=ADD COMM=INTA
REG1=IRON
```

will cause the **INTA** command to be issued, to integrate over all regions defined as **IRON**. Groups are created with the OPERA-2d data file for each material name that has been specified.

- The parameter can also take expressions relating to specific polygons. These expression can generate a single value, which refers to a polygon and all its replications. Hence the command is issued over this range of regions. For example

```
POST OPTION=ADD COMM=INTA
REG1=2
```

will cause **INTA** command to be issued, to integrate over polygon 2 and all its replications. Alternatively the parameter can be assigned an expression so as to process specific region. This is achieved by stating the polygon and replication number. For example

```
POST OPTION=ADD COMM=INTA
INTA REG1=4 [2]
```

will cause the **INTA** command to be issued so as to integrate over replication 2 of polygon 4.

- Care must be taken if the **UNIT** command is used as a post processing command, since the values assigned to the point coordinates of commands such as **INTLINE** will not reflect the unit changes.

The **RUNPART** Command*

Menu Route:

DEM ↓
 Demonstrate a Part
 ANALYSIS ↓
 Set PART data

Command Line Parameters:

Command	RUNPART	
Parameter	Default	Function
NAME	<i>none</i>	Name of the part to be run
PA01	<i>none</i>	Parameters whose use will depend upon the part selected through name
PA02		
PA03		
PA04		
PA05		
PA06		
PA07		
PA08		
PA09		
PA10		
PA11		
PA12		
PA13		
PA14		
PA15		
STORE	NO	Store this set of variable values as a new model variation.
	NO	Don't store the model variation
	YES	Store the model variation
FILE	<i>none</i>	Filename to be used if storing the model variation or a range of model variation
MIN	<i>none</i>	Start value for a range variable

Command	RUNPART (continued)	
Parameter	Default	Function
MAX	none	Finish value for a range variable
INCREMENT	none	Increment by which a range value increases between MIN and MAX values

The **RUNPART** command is the main command used within the *User System* and is also available within the *Configuration System*.

The **RUNPART** command can be executed with only the first parameter set. This loads the **PART** that has been created with this name. See “The **PART** Command” on page 1-75.

After selecting the **PART** that is to be used, the values of parameters **PA01** to **PA15** are set to defaults for variables that are set within the part. (See “**ASKVAR**” on page 1-76. See “**ASKYESNO**” on page 1-77.) By modifying the value of the relevant parameter, the value of the associated parameter is modified to the new value. The **RUNPART** command finishes when **<return>** is selected with no parameter values set on the command line.

When setting the value of variables in the *part*, a range variable can be created by setting the value of the variable to be *. In such a case the values of **MIN**, **MAX** and **INCREMENT** must be set to define the range of values used by the variable. A model for each value of the range variable will be created.

When creating updating the commands, the **STORE** parameter allows the current model variation to be stored to the file given by the **FILE** parameter. If a range variable is being used, the range of model variations will be created and each stored to a separate file, the name being generated by suffixing an integer number to the filename given. The **STORE** parameter may only be set to **YES** if an analysis control set name has been specified. See “The **CONTROL** Command*” on page 1-16.

The SET Command

Menu Route:

MODEL ↓
Solution Settings

Command Line Parameters:

Command	SET		
Parameter	Default	Function	
SYMMETRY	XY	Coordinate system:	
		AXIS XY	Axisymmetry (R≡X, Z≡Y). Cartesian.
SOLN	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 for electrostatics.	
ELEMENT	LINEAR	Element type:	
		LINEAR QUADRATIC	First order elements. Second order elements.
FIELD	MAGNETIC	Field type:	
		ELECTRIC MAGNETIC	Electrostatics. Magnetics.

The **SET** command defines parameters which affect the way the analysis programs operate. After the **SET** command has been used, or data has been **LOADED** from a file (see “**The LOAD Command***” on page 1-68), the current values become the new default values.

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.

Several different solution potentials (**SOLN**) can be used with the *Design Environment*. They determine the meaning of the system variable **POT** within the *pre- and post processor*.

Solution Potentials and the System Variable POT	
SOLN	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 *Design Environment* can use **LINEAR**, 3 noded triangular or **QUADRATIC** elements with mid-side nodes as well. The quadratic elements have straight sides, as the mid-side node will be automatically placed on the edge already defined.

The **FIELD** type controls the setting of the system variable names and several constants used in post processing.

The **SKETCH** Command

Menu route:

MODEL ↓
 Geometry → Sketch polygon

Command line parameters:

Command	SKETCH		
Parameter	Default	Function	
AUTO	NO	Automatically constrain the polygon when sketching	
		NO	Do not add constraints
		YES	Add constraints

The **SKETCH** command allows a polygon to be drawn by using the cursor to select the point positions in the polygon. By default these points will all be unconstrained, so constraints should be added to define their position. The **AUTO** parameter can be used to automatically constrain points to a fixed position. See “The **POINT** Command” on page 1-80. These constraints are added as the polygon is closed.

All sketching is done using the cursor. The cursor hits available are:

Command	SKETCH
Keyboard	Function
<space>	Add a point at the cursor
a	Switch auto-constrain on
b	Backtrack last point
c	Close the polygon
g	Create a cartesian grid of points
h	Display a help message
o	Switch auto-constrain off
p	Create a polar grid of points
q	Quit the SKETCH command
z	Zoom display (see “The ZOOM Command*” on page 1-116)

By using the cursor, the series of points used to construct the polygon can be added by selecting with **<space>**. Auto constraining can be toggled by **a** and **o**. To complete the polygon select **c**. If a mistake is made points can be removed by backtracking with **b**.

A cartesian or polar grid of points can be displayed using **g** or **p**. When the grid is displayed, any new points will be positioned at the nearest point on the grid. A prompt will be displayed to allow the size of the grid to be set. Setting a size of zero will deactivate the grid. The display must be redrawn using zoom to clear existing grid points from the display.

The **SKETCH** command can be left by selecting **q**. If a polygon is part way through being created, the program will ask if you wish to keep the unfinished polygon.

The **SOLC** Command

Menu Route:

MODEL ↓
 Constraints → Solve constraints

Command Line Parameters:

Command	SOLC	
Parameter	Default	Function
CHANGE	YES	Solve the geometry after every change.

This command is used to invoke the variational geometry solver to solve the constraints as given and generate the model topology. Occasionally, the command may need to be used more than once if the topology is particularly difficult to solve. If this occurs, the constraint scheme should be adjusted.

The only parameter attached to the **SOLC** command is **CHANGE**. This parameter allows the solver to be switched on or off. The default setting is **+CHAnge**, which means that after every addition of constraint or change of constraint or variable the topology is automatically re-solved and updated. Setting **-CHAnge** means that the topology is not updated automatically and will only be updated if the **SOLC** command is employed.

NB: if the **RUNPART** command is being used, the variational geometry will automatically solve for any change in constraint or variable, irrespective of the setting of **CHANGE**.

The **SOLVE** Command*

Menu Route:

```

MODEL ↓
    Analysis Data
ANALYSIS ↓
    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
		SP Space charge analysis
		ST Static 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 analysis
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	Output times at which a solution will be stored

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), 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	Use linear analysis	
			NO	Use non-linear analysis
			YES	Use linear analysis
NITERATIONS	21	ST/ AC/ TR/ VL	Maximum number of non-linear iterations that will be used	
TOLERANCE	0.001	ST/ AC/ TR/ VL	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
ADACCURACY	5	ST/ AC/ VL	Final accuracy at which adaptive refinement will stop	
TSTEP	0	TR	Fixed time step	
TSTOLERANCE	0.001	TR	Tolerance for adaptive time stepping	

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
TSTYPE	ADAPT	TR	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
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/ SP	Restart from the current solution	
			NO	Use non-linear analysis
			YES	Use linear analysis
TSINITIAL	0	TR	Initial time step for adaptive timestepping.	

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.

DRIVE Sub-command

The **DRIVE** sub-command is for use only with the transientmodule, 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 $t < 0$ where $f(0)$ 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

QUIT Sub-command

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

The SYMMETRY Command

Menu Route:

MODEL ↓
Periodic Symmetry

Command Line Parameters:

Command	SYMMETRY	
Parameter	Default	Function
OPTION	<i>none</i>	Polygon to be copied.
		POSITIVE Define a positive transformation.
		NEGATIVE Define a negative transformation.
		LIST List the transformations defined.
DELETE Delete all defined transformations.		
CP	<i>none</i>	Centre point of rotation.
ANGLE	<i>none</i>	Angle of rotation about CP .
DX	<i>none</i>	X displacement.
DY	<i>none</i>	Y displacement.

This command is used to define the symmetry transformations that will match master and slave facets that have the symmetry boundary condition. The transformation can consist of a rotation about a fixed point followed by a translation. The type of transformation can be defined as positive or negative.

The setting of **OPTION=POSITIVE** indicates that the potential of the master facet will be transferred to the slave potential without any change in magnitude or phase.

The setting of **OPTION=NEGATIVE** indicates that the potential of the master facet will be transferred to the slave with a 180° phase shift - hence effectively the negative of the master potential.

The setting of **OPTION=LIST** lists out all the defined transformation.

The setting of **OPTION=DELETE** deletes all the defined transformations without exception.

The transformation is carried out by firstly rotating symmetry facets by an amount set by **ANGLE** about centre point **CP** and secondly translating by the vector **DX**,

DY. The order is important since a rotation followed by a translation will not necessarily give the same transformation as a translation followed by a rotation.

The parameter **CP** can be given either a polygon point name or a reference point name as its argument.

The parameters **ANGLE, DX, DY** can be given either expression or numeric values as their arguments, and any expression is stored and evaluated as it is used.

During conversion to OPERA-2d data, symmetry sides are matched. If necessary, points are inserted into faces to ensure that edges and subdivision values can be matched across the symmetry condition. See [“The CONVERT Command” on page 1-19](#).

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 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
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 **VARIABLE** Command

Menu Route:

MODEL ↓
 Variables
 DEM ↓
 Dimension lines → Define/edit dimension

Command Line Parameters:

Command	VARIABLE	
Parameter	Default	Function
NAME	<i>none</i>	Variable name.
VALUE	<i>none</i>	Either a numeric value or an expression
DESCRIPTION	<i>none</i>	Text string used to describe the variable.
TYPE	NONE	Type of dimensioning arrow.
		NONE No dimensioning arrow assigned.
		LINE Straight dimension arrow assigned.
		ARC Curved dimension arrow with a given curvature.
		CENTRE Curved dimension centred on a point.
		CENA Anti-clockwise curved dimension centred on a point.
CENC Clock-wise curved dimension centred on a point.		
P1	<i>none</i>	Starting point for the dimension arrow.
P2	<i>none</i>	End point for the dimension arrow.
CP	<i>none</i>	Centre point for the dimension arrow (TYPE=CENTRE/CENA/CENC).
CURVATURE	0	Curvature of dimension arrow (TYPE=ARC).
XLABEL	0	Expression for the x-shift of dimension label.
YLABEL	0	Expression for the y-shift of dimension label.
DEFAULT	YES	Use default line style for dimension line.

Command	VARIABLE (continued)		
Parameter	Default	Function	
		YES	Default line style used.
		NO	User line style used.
LTYPE	0	Line type for dimension (0-solid,1-16 dotted).	
ETYPE	0	Line type for the end-bar (0-solid,1-16 dotted).	
ARROWS	BOTH	Dimension arrows as FORWARD , BACKWARD , NONE or BOTH .	
ALENGTH	1	Expression for the length of the arrow heads.	
ELENGTH	1	Expression for the length of the end-bars.	
ESHIFT	0	Expression for the shift of the end-bars.	
DISPLACEMENT	0	Expression for the displacement of the dimension line.	

This command is central to the parameterised model in that this command sets up the variables that are used to build the expressions used in setting material properties and the variational geometry constraint expressions. Any variable name can be used, the proviso being that the first character is a character and not a numeral or # symbol. There are a few names that are not permitted, such as system variable names, and should the user choose these, error messages are generated and the variable is disallowed. All variable names must be less than or equal to 7 characters in length.

Variables can take numeric values as their argument or other variables and expressions.

The first two parameters are the most important, as these set the variable name and expression. The variable name and expression are the minimum requirements to define a variable.

The text string, parameter **DESCRIPTION**, is a label that describes the variable and its usage. This parameter need not be set, but if the variable is to be set in the *User System* by a novice using the **PART** command, then the description is an essential piece of information. The text descriptor is used as part of the menuing structure of the **PART** command and is read by the novice user before setting the variable expression. See “The **PART** Command” on page 1-75.

The other parameters are used to set up dimensioning lines that can clearly label a length or angle that takes the variable as its value. These parameters need not be set, since not all variables need be shown as dimensions.

The dimensioning line can be used to show the user the precise location and usage of a particular variable, in setting the geometry. The dimensions should be used in a similar way to the dimensions that can be seen on a typical engineering drawing.

The dimension arrows are implemented by constraining two points that define the start and end point of the arrow. These points **P1** and **P2** can be points from the model or reference points set up by the user. If a dimension is requested for a variable (**TYPE=LINE, ARC, . . .**), and points **P1** and **P2** are undefined, 2 unconstrained reference points are automatically assigned as the start and end points. These reference points must then be constrained onto the model.

The dimension can be either straight or curved. There are four types of curve available. **TYPE=LINE** is a straight line directly linking the points **P1** and **P2**. The simplest curve is the **TYPE=ARC**, which is an arc linking points **P1** and **P2** with a curvature given by parameter **CURVATURE**. The curve **TYPE=CENTRE** is an arc of maximum 180° centred on a point defined by parameter **CP**. An arc will always follow the minor arc path given two points on a radius and the centre point. The curves **TYPE=CENA, CENC** offer the opportunity to specify the direction that the dimension arrow should take (anti-clockwise or clockwise). This directional control means that major arc paths are available to the user.

The dimension is automatically labelled with the variable name, which is placed in a shaded area located in the middle of the dimension line itself. The automatic placement may not be ideal for the model, and so two parameters **XLABEL, YLABEL** can be used to move the label from its placement point. These parameters can take numeric or expression values.

The type of lines that are drawn for the line and end-bars may be varied using the parameters **LTYPE** for the line and **ETYPE** for the end-bar. The default line type is given as **LTYPE=0** and **ETYPE=0**, which is a solid line. There are 15 other line styles which give a variety of dotted and broken line styles. These can be obtained by setting the parameters to the appropriate number (1-15).

The arrows that are placed on the dimension line are defaulted as arrows at both ends, (**ARROWS=BOTH**). The arrows can be totally removed by setting **ARROWS=NONE**, or to having only one arrow **ARROWS=FORWARD** or **ARROWS=BACKWARD**. The arrow sizes can be changed by modifying the parameter **ALENGTH**, which sets the length of the arrow. If the dimension is too short to contain the line end-arrows, the dimension will invert and the arrows will point inwards.

The end-bars are placed at the two ends of the dimension line. The length of the end-bars can be modified by changing the parameter **ELENGTH** and the end-bars may be shifted up or down with the parameter **ESHIFT**.

In general most of these parameters need not be modified. If the user uses a particular set of parameter values on a continual basis, the default settings of the parameters can be customised. This is achieved by stating no variable name when running the VARIable command, but setting **VARIABLE NAME=***. This will enable the defaults to be reset by setting the appropriate parameters - **XLABEL, YLABEL,... ESHIFT**. Once the user is satisfied with the default settings they can be applied to variables by using the **+DEFAULT** parameter of the **VARIABLE** command.

The user must specify **-DEFAULT** if a different line style to the default is required. The parameters can then be customised as described above.

The **VECTOR** Command

Menu Route:

MODEL ↓
 Constraints → By Keyboard → Vector

Command Line Parameters:

Command	VECTOR	
Parameter	Default	Function
P1	<i>none</i>	First polygon point to be constrained.
P2	<i>none</i>	Second polygon point to be constrained.
XVECTOR	<i>none</i>	X-component of the vector.
YVECTOR	<i>none</i>	Y-component of the vector.

This command sets a vector constraint between two points.

The **VECTOR** command constrains a particular point to be fixed at a certain vector (x,y) from another point. This type of constraint is very stable and cannot lead to multiple solutions existing between the defined points. The definition of the constraint is that point

$$P2 - P1 = (XVECTOR, YVECTOR)$$

The parameter **P1**, **P2** refer to the point names that are to be constrained. **P1** refers to the point name whose position will be taken as the starting point of the constraining vector. The parameters **XVECTOR**, **YVECTOR** take either expressions or numeric values as their arguments and defined the x and y components of the vector linking the two points. The **VECTOR** command sets up two constraint expressions, which are added to the constraint list defining the model. The constraint list can be viewed with the **LIST CONSTRAINT** command (see “The **LIST Command**” on page 1-66).

The **MODIFY** command can be used to alter the values or expressions that are used to define the vector between points **P1** and **P2**.

The **WRITE** Command

Menu Routes:

```

FILE ↓
    Save DEM
FILE ↓
    Save as new DEM
  
```

Command Line Parameters:

Command	WRITE	
Parameter	Default	Function
FILE	<i>none</i>	The DEM filename.

This command is used to **WRITE** a DEM data file. All the data that makes up the model such as constraints, variables, material settings, expressions etc., are stored in the DEM file.

The parameter **FILE** is the DEM filename and is important since the filename given will be used as the stem for the *.CAS* (case) and *.CSR* (case results) files. If the filename given already exists, the user is asked if the old file should be overwritten. If the user wishes to retain the old file, entering **NO** at the overwrite prompt will terminate the **WRITE** command.

The ZOOM Command*

Menu Routes:

DISPLAY ↓
Zoom

The **ZOOM** command is also available in many other menus where refreshing or zooming into the display may be necessary.

Command Line Parameters:

Command	ZOOM
No Parameters	

ZOOM executes a **DISPLAY** command (see “The **DISPLAY** Command*” on page 1-33) having set the first 4 parameters (**XMIN**, **XMAX**, **YMIN**, **YMAX**) by use of the cursor. The other **DISPLAY** parameters are unchanged. **ZOOM** has 6 options:

- zoom in: To zoom in two cursor hits must be given, defining diagonally opposite corners of a rectangular area of the graphics window. The first corner is defined by a <space> cursor hit. The second corner is defined with a **B**; the picture within the rectangle will be made Bigger to fill the graphics window. When executed from the menus, the two cursor hits are supplied by means of a rubber-box.
- zoom out: To zoom out two cursor hits must be given, defining diagonally opposite corners of a rectangular area of graphics window. The first corner is defined by a <space> cursor hit. The second corner is defined with a **S**; the current picture will be made Smallier to fit inside the rectangle. When executed from the menus, the two cursor hits are supplied by means of a rubber-box.
- zoom in by a factor of 2: To zoom in with the same aspect ratio, halving the values of (**XMAX-XMIN**) and (**YMAX-YMIN**), the single cursor hit **I** should be used. The position of the cursor defines a point which is moved to the centre of the display.
- zoom out by a factor of 2: To zoom out with the same aspect ratio, doubling the values of (**XMAX-XMIN**) and (**YMAX-YMIN**), the single cursor hit **O** should be used. The position of the cursor defines a point which is moved to the centre of the display.
- pan: To pan the picture so that a new point is moved to the centre of the display, the single cursor hit **P** should be used.

- bounding rectangle: To display everything within the bounding rectangle of the geometry, the single cursor hit **B** should be used.
- keyboard limits: To use the keyboard, the single cursor hit **K** should be used. The program prompts the user to type the coordinates of diagonally opposite corners of the rectangle to be re-displayed.
- existing limits: To **DISPLAY** the display with the existing limits the single cursor hit **R** should be used.
- quit without a **DISPLAY**: cursor hit **Q**.

INDEX

Symbols

\$DO loop 1-4

-A-

adaptive refinement options 1-99
analysis 1-16
analysis options 1-98
ANGLE command 1-2
angle constraint 1-2
anisotropy 1-4
automatic analysis 1-16
axisymmetry 1-91

-B-

BH tables 1-3
BHDATA command 1-3
boundary conditions
 BOUNDARY command 1-7
 periodicity 1-103
 symmetry 1-103

-C-

cartesian coordinates 1-91
CASE command 1-9
CASE sub-command of **SOLVE** 1-97
charge density
 units 1-109
circuits 1-45
CLEAR command 1-11
coercive force 1-3

COLOUR command 1-12
complex permeability 1-3
conductivity 1-72
 units 1-109
conductor numbers 1-45
CONSTRAIN command 1-14
constraint
 angle 1-2, 1-14
 angle difference 1-14, 1-27
 fixed point 1-80
 internal angle 1-14, 1-59
 internal angle difference 1-14, 1-32
 length 1-14, 1-61
 length difference 1-14, 1-37
 vector 1-14, 1-114
 vector difference 1-14, 1-43
constraints
 copying polygons 1-23
 deleting 1-28
 importing op2 data 1-55
 listing 1-66
 modifying 1-74
 solving 1-95
CONTROL command 1-16
control files 1-86
CONVERT command 1-19
COPY command 1-22
copying regions 1-22
 reflection 1-22
 rotation 1-22

translation 1-22
 creating a new polygon 1-81, 1-93
 current density 1-72
 units 1-108, 1-109
CURVATURE Command 1-25

-D-

DANGLE command 1-27
 data files
 time-tables 1-102
DATA sub-command of **SOLVE** 1-98
 defining a new variable 1-110
DELETE command 1-28
 deleting constraints 1-28
 deleting points 1-28
 deleting polygons 1-28
Design Environment Modules 1-68
DEVICE command 1-30
DIANGLE command 1-32
 dimension arrows 1-112
 dimension display 1-35
 dimensioning a variable 1-111
DISPLAY command 1-33
 display options 1-35
 display size 1-35, 1-116
DLENGTH command 1-37
DRIVE sub-command of **SOLVE** 1-100
DUMP command 1-38
DVECTOR command 1-43

-E-

easy direction 1-3
 electrostatics 1-91
END command 1-44
 energy
 units 1-109
 environment variables 1-30
 expression limits 1-62
 external circuits 1-45
EXTRA
 boundary condition 1-8

conductivity 1-72
 density 1-72
 permeability 1-72
 phase 1-72

-F-

field strength
 units 1-109
 files
 BH data 1-4
 case files 1-19
 command input 1-11, 1-83
 control 1-16, 1-86
 dem files 1-68, 1-115
 model variation 1-86
 OPERA-2d data files 1-16, 1-19, 1-55
 post-processing command scripts ... 1-16,
 1-83
 results 1-86
 writing 1-115
 fixed point constraint 1-80
 flux density
 units 1-109
 force
 units 1-109
 frequencies 1-97

-G-

GETOP2 command 1-55
 GO conductor 1-48
GRAVITY command 1-57

-H-

hard magnetic materials 1-3
 hysteresis 1-3

-I-

IANGLE command 1-59
 importing op2 data 1-55
INCLUDE parameter 1-73
INDEX variable 1-72

inductance 1-46
INSERT command 1-60
 inserting points 1-60
 internal angle difference constraint 1-32
 isotropic materials 1-4

-L-

length
 units 1-109
LENGTH command 1-61
 length difference constraint 1-37
 length of circuit 1-48
LIMIT command 1-62
 limiting variable values 1-62
 linear analysis 1-98
 linear elements 1-92
LINK Command 1-64
LIST command 1-66
 listing constraints 1-66
 listing points 1-66
LOAD command 1-68
 loading a DEM file 1-68
LOOP command 1-45

-M-

mass
 units 1-109
MATERIAL command 1-71
 material name 1-71
 menus for the *User System* 1-78, 1-89
 mesh
 control points 1-57
 polygon subdivision 1-57, 1-69
 model variation files 1-86
MODIFY command 1-74
 modifying a variable value 1-110
 modifying constraints 1-74
 multiple cases 1-97

-N-

non-linear analysis 1-98

-O-

opera2.comi 1-11
 OPERA-2d data files 1-19
 output time 1-97

-P-

packing factor 1-4
 region parameter 1-3
PART
 ASKBH sub-command 1-76
 ASKVAR sub-command 1-77
 ASKYESNO sub-command 1-77
 demonstrating the menus 1-89
 display 1-75
 TEXT sub-command 1-77
PART command 1-75
 periodicity 1-103
 permeability 1-72
 phase 1-72
 phase angle
 complex permeability 1-3
POINT command 1-80
 point constraint 1-80
 points
 deleting 1-28
 listing 1-66
 point insertion 1-60
 temporary positioning 1-82
POLYGON command 1-81
 polygons
 copying 1-22
 creating 1-81, 1-93
 deleting 1-28
 listing constraint data 1-66
 listing side data 1-66
 material properties 1-69
 replicating 1-69
 sketching 1-93
POSITION command 1-82
POST command 1-83
 post-processing 1-16, 1-86

- potential
 - units 1-109
- potential type 1-92
- power
 - units 1-109
- Q-**
- quadratic elements 1-92
- R-**
- refreshing the display 1-116
- region properties 1-69
- replicating regions 1-73
- restart solutions 1-100
- results files 1-86
- RETURN conductor 1-48
- RUNPART** command 1-89
- S-**
- saturation magnetisation 1-3
- saving a *dem* file 1-115
- scale factor 1-97
- SET** command 1-91
- SKETCH** command 1-93
- SOLC** command 1-95
- SOLVE** command 1-96
- solving constraints 1-95
- space charge analysis options 1-100
- SPICE type data 1-51
- stored energy
 - units 1-109
- storing a *dem* 1-115
- subdivision size 1-57
- SYMMETRY** command 1-103
- T-**
- testing OPERA-2d data files 1-19
- time-table files 1-102
- timetables 1-101
- TITLE** Command 1-105
- titles 1-105
- transient drive functions 1-100
- transient time step options 1-99
- U-**
- UNITS** command 1-107
- User System settings* 1-78
- V-**
- VARIABLE** command 1-110
- variables
 - defining 1-110
 - descriptions 1-110
 - dimension lines 1-110
 - limiting 1-62
 - modifying 1-110
 - setting values 1-89
 - using in the *User System* 1-75
- variational geometry 1-95
- VECTOR** command 1-114
- vector constraint 1-114
- vector difference constraint 1-43
- VFGRAHICS** 1-30
- VFINV** 1-31
- VFWINDOWH** 1-31
- VFWINDOWW** 1-31
- Voltage driven circuits 1-45
- W-**
- WRITE** command 1-115
- Z-**
- ZOOM** command 1-116