

Version 18R2

# Opera-3d Reference Manual

Version 18R2

Cobham Technical Services  
Network House, Langford Locks  
Kidlington, Oxfordshire, OX5 1LH, UK

May 2016

## Copyright and Trademark Information

---

Chelton Limited trading as Cobham Technical Services

Copyright © Chelton Limited 1999-2016

All Rights Reserved

This document was prepared using MadCap Flare.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Red Hat® is a registered trademark of Red Hat Incorporated in the U.S. and other countries.

PostScript® is a registered trademark of Adobe Systems Incorporated in the U.S. and other countries.

Windows® operating system is a registered trademark of Microsoft Corporation in the U.S. and other countries.

Qt® is a registered trademark of Digia Plc and/or its subsidiaries.

Graphing is based in part on the work of the Qwt project (<http://qwt.sf.net>).

ACIS and SAT are trademarks of Spatial Corporation.

I-deas is a trademark of Siemens PLM Software Inc.

LM-X is a trademark of X-Formation.

OpenGL® is a registered trademark of Silicon Graphics Inc in the U.S. and other countries.

Simulink® is a trademark of The MathWorks Inc.

Python is Copyright © 2001-2016 Python Software Foundation; All Rights Reserved.

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

# Contents

---

Copyright and Trademark Information .....	2
Contents .....	3
<b>Chapter 1 System Overview .....</b>	<b>17</b>
Introduction .....	17
Opera-3d Analysis Programs .....	18
Magnetostatic, Electrostatic and Current Flow Solvers .....	18
Dynamic Electromagnetic Solvers .....	18
Motional Electromagnetic Solver .....	18
Magnetization Solver .....	19
Charged Particle Solver .....	19
Thermal Solvers .....	19
Stress Solvers .....	19
High Frequency Solvers .....	19
Quench Solvers .....	19
Old Names of Solvers .....	20
Program Limits .....	21
Fundamental Physical Constants .....	22
<b>Chapter 2 Command Language .....</b>	<b>23</b>
Introduction .....	23
Keyboard Input .....	24
Output Files .....	25
Commands and Parameters .....	26
The Help Character (!) .....	27
Parameter Assignment .....	28
Parameter Values .....	30
Numeric Parameter Values .....	30
Expressions in Parameter Values .....	30
Character Values for Parameters .....	35
Boolean Parameter Values .....	36
List Parameter Values .....	36
Text Functions .....	37

---

Syntax .....	38
Functions .....	38
Command Interpreter Errors .....	45
Shortcuts and Reminders .....	46
Repeated Commands .....	46
Prompted Input of Parameter Values .....	46
Prompted Free Format Input .....	48
Built-in Commands .....	49
Command Prompts .....	51
Limitations .....	52
Conditional Commands .....	52
Loops .....	53
User Variable Commands .....	55
System Variable Command .....	60
User Function Command .....	60
Command Input Files .....	61
Opera Python .....	65
Dialogs .....	67
File Input/Output Commands .....	73
Commands to Run Other Programs .....	77
The \$ CD command .....	80
File Existence Command .....	80
Command File Editor .....	81
Command Separator and Comments .....	82
Command Separator   .....	82
Comments / .....	82
Euler Angles .....	83
<b>Chapter 3 The Geometric Modeller .....</b>	<b>85</b>
Introduction .....	85
Commands Issued at Startup .....	87
Topology Definitions .....	88
Introduction .....	88
Geometric Entities .....	88
Geometric Model Information .....	90
Bodies and Cells .....	90
Geometric Modelling .....	92
Modeller Functionality .....	92
Modeller Data .....	97
Display .....	101
Command History .....	101

---

Background, Symmetry and Boundary Conditions .....	102
Model Storage .....	102
Building a Model for Mesh Generation .....	103
Opera-3d Modeller Tabbed Menu Interface .....	105
Toolbar summary .....	105
Command Summary .....	115
Keyboard Shortcuts .....	120
The ANALYSISDATA Command .....	122
The ANIMATION Command .....	130
The ARC Command .....	132
The BACKGROUND Command .....	135
The BEDSTEAD Command .....	140
The BEND Command .....	143
The BHDATA Command .....	144
BH Viewer .....	145
BH Editor .....	146
Material Types .....	146
The BLEND Command .....	147
The BLOCK Command .....	148
The BOUNDARY Command .....	150
The BRICK8 Command .....	156
The BRICK20 Command .....	159
The CEDITOR Command .....	162
Circuit Editor Interaction .....	165
The CELldata Command .....	168
The CHECK Command .....	171
The CIRCUIT Command .....	173
The CLEAR Command .....	180
The COLOUR Command .....	182
The COMBINE Command .....	184
The COMMENT Command .....	186
The COMPATIBILITY Command .....	187
The CONDUCTOR Command .....	189
The CONTOUR Command .....	192
The COVER Command .....	194
The CYLINDER Command .....	195
The DBCASEDATA Command .....	197
The DELETE Command .....	200
The DRIVE Command .....	202
The EDGEDATA Command .....	205
The EMITTER Command .....	206
The END Command .....	210

---

The EXPORT Command .....	211
The EXTRACTCELLS Command .....	212
The FACEDATA Command .....	213
The FILL Command .....	216
The FILTER Command .....	219
The FITTEDCPE Command .....	222
The GUIOPTIONS Command .....	225
The HEATTRANSFER Command .....	227
The HELICALEND Command .....	230
The HIDE Command .....	233
The HISTORY Command .....	234
The IMPORT Command .....	235
The INSERTOP2FILE Command .....	236
The LABEL Command .....	238
The LCS Command .....	240
The LIST Command .....	241
The LOAD Command .....	243
The LOFT Command .....	246
The MATERIALS Command .....	248
The MESH Command .....	257
The MODEL Command .....	263
The MORPH Command .....	265
The MOUSE Command .....	267
The MULTIPHYSICS Command .....	269
The OFFSET Command .....	273
The PERIODICITY Command .....	274
The PICK Command .....	277
The PICTURE Command .....	281
The PRECISIONDATA Command .....	283
The PREVIEW Command .....	284
The PRINT Command .....	286
The PRISM Command .....	287
The RACETRACK Command .....	289
The REDO Command .....	292
The RENAME Command .....	293
The REPLAY Command .....	294
The SAVE Command .....	297
The SELECT Command .....	299
The SHELL Command .....	303
The SKETCH Command .....	304
The SOLENOID Command .....	306
The SOLVERS Command .....	309

---

The SPHERE Command .....	313
The STRAIGHT Command .....	314
The STRETCH Command .....	317
The SWEEP Command .....	318
The TANGENTIALCPE Command .....	321
The THREED Command .....	324
The TITLE Command .....	327
The TORUS Command .....	329
The TRANSFORM Command .....	330
The TWIST Command .....	334
The UNDO Command .....	335
The VARIABLE Command .....	337
The VECTOR Command .....	339
The VERTEXDATA Command .....	341
The VOLUME Command .....	342
The WCS Command .....	345
The WINDING Command .....	347
The WINDOW Command .....	351
The WIREEDGE Command .....	352
System Variables in the Modeller .....	355
<b>Chapter 4 Opera-3d Pre-Processor .....</b>	<b>362</b>
Introduction .....	362
The Pre-Processor Quick Reference Guide .....	363
Commands for Keyboard Entry – Command Line .....	363
Menu System – GUI .....	364
The Graphical User Interface .....	368
Menus .....	368
Parameter Boxes .....	369
FileBoxes and CDBoxes .....	370
DialogBoxes .....	372
MessageBoxes .....	373
ColourBoxes .....	373
Hiding and Leaving the Menus .....	374
The BHDATA Command .....	375
BHDATA Sub-commands .....	376
The CHECK Command .....	378
Checks on Hexahedral Models .....	379
Checks on other Models .....	379
The CLEAR Command .....	380
The COLOUR Command .....	381

---

The CONDUCTOR Command .....	383
The CONDUCTOR Sub-command DEFINE .....	383
The CONDUCTOR Sub-command ERASE .....	386
The CONDUCTOR Sub-command EXTERNAL .....	386
The CONDUCTOR Sub-command MODIFY .....	387
The CONDUCTOR Sub-command PRINT .....	390
The CONDUCTOR Sub-command QUIT .....	391
The CONDUCTOR Sub-command WRITE .....	391
The DEFINE Command .....	393
Finite Element Meshing .....	394
DEFINE start up sequence and modes .....	395
Aspect Ratio Searching .....	396
Defining Meshes with the GUI .....	396
Defining Meshes with the Keyboard .....	397
Construction Lines and Grids .....	397
Point Definition Mode .....	399
Facet Definition Mode .....	405
Facet Group Operations Mode .....	407
Base Plane Subdivision Mode .....	410
Extrusions Mode .....	413
Material Definition Mode .....	420
Boundary Condition Definition Mode .....	426
The DEVICE Command .....	434
The DISPLAY Command .....	436
The Coordinate Limits and the Viewing Direction .....	439
Selecting Parts of the Finite Element Model .....	439
Selecting Conductors .....	440
Other parameters .....	440
Examples .....	441
The DUMP Command .....	442
The EDIT Command .....	444
Editing Top-level Commands .....	444
Editing the DEFINE Command .....	445
The END Command .....	447
The EXTEND Command .....	448
Extrusions Mode .....	448
Material Definition Mode .....	449
Boundary Condition Definition Mode .....	449
The FILL Command .....	450
Tetrahedral meshes .....	450
Hexahedral meshes .....	450
The HELP Command .....	451

---

The IDEAS Command .....	452
Universal Data File for Input to Opera-3d .....	452
Displaying Universal File Data .....	454
The IDEAS Sub-command BOUNDARY .....	454
The IDEAS Sub-command MATERIAL .....	456
The IDEAS Sub-command QUIT .....	457
The IDEAS Sub-command READ .....	457
The LABEL Command .....	458
Entity Selection .....	459
Labelling Sub-commands in Keyboard Mode .....	460
Labelling DialogBox in Menu Mode .....	461
The MATERIALS Command .....	462
The MESH Command .....	467
Triangular meshes .....	468
Quadrilateral meshes .....	468
The MODIFY Command .....	469
Point Modification Mode .....	470
Subdivision Modification Mode .....	474
Subdivision DialogBox in Menu Mode .....	479
Material Modification Mode .....	479
Boundary Condition Modification Mode .....	480
The READ Command .....	481
The REDEFINE Command .....	483
The SLIP Command .....	484
The SOLVERS Command .....	485
Analysis types .....	487
Database files and file names .....	488
Creating a new database .....	488
Editing an existing database .....	490
SOLVERS Command Prompts .....	491
CASE sub-command .....	493
CHECK sub-command .....	494
DRIVE sub-command .....	495
MATERIAL sub-command .....	495
PERIODICITY sub-command .....	497
QUIT sub-command .....	499
SETTINGS sub-command .....	499
The TABLE Command .....	502
The THREED Command .....	503
Selecting Parts of the Finite Element Model .....	504
Selecting Conductors .....	505
Other parameters .....	505

---

The 3D Viewer Menus .....	506
The TITLE Command .....	507
The TRANSFORM Command .....	509
The WRITE Command .....	510
<b>Chapter 5 Conductors .....</b>	<b>512</b>
Introduction .....	512
Conductor Shapes .....	513
Parameter Names .....	513
Solenoids .....	514
Racetracks .....	515
Bedsteads .....	517
Helical Ends .....	519
Constant Perimeter Ends .....	520
Straight Bars .....	522
Circular Arcs .....	524
Bricks .....	525
Local Coordinate Systems .....	529
Local Coordinate System 1 .....	529
Local Coordinate System 2 .....	529
Current Density, Drive Label and Tolerance .....	531
Symmetries and Reflections .....	532
Conductors in Circuits .....	533
Including a Conductor in a Circuit .....	533
Modelling the Conductor .....	534
Volume Mesh for Biot-Savart Conductors .....	535
Conductor Commands .....	536
Modeller .....	536
Pre-Processor .....	536
Post-Processor .....	536
Conductor Files .....	538
Writing Conductor Files .....	538
Reading Conductor Files .....	538
<b>Chapter 6 Analysis Programs .....</b>	<b>539</b>
Introduction .....	539
Static Electromagnetic Field Solvers .....	541
Introduction .....	541
The Magnetostatic, Electrostatic and Current Flow Algorithm .....	541
Dynamic Electromagnetic Solvers .....	542
Introduction .....	542

---

Time Variation in the Dynamic Electromagnetic Solvers .....	542
Motional Electromagnetic Solver .....	543
Introduction .....	543
Models for the Motional EM Solver .....	543
Pre-Processor Models for the Motional EM Solver .....	543
Motion Control .....	544
Charged Particle Solver .....	548
Introduction .....	548
The Charged Particle Algorithm .....	548
Thermal Emission Models .....	549
Field Effect Emission Models .....	556
User Defined Emission Models .....	557
Plasma Emission .....	559
Emitter Current .....	561
Secondary Emission .....	562
Secondary Emission System Variables .....	563
Beam Loss and Volume Interactions .....	565
Volume Plasma Emitter .....	566
Magnetic Field of Particle Beam .....	568
Emitter Data Files .....	568
Example Data Files .....	580
High Frequency Solvers .....	585
Introduction .....	585
The Modal HF Algorithm .....	585
The Harmonic HF Algorithm .....	586
Magnetization Solver .....	589
Introduction .....	589
Material Models .....	589
Static Thermal Solver .....	591
Introduction .....	591
The Static Thermal Algorithm .....	591
Heat Tables .....	591
Boundary Conditions for the Static Thermal Solver .....	592
Transient Thermal Analysis .....	594
Introduction .....	594
The Transient Thermal Algorithm .....	594
Heat Tables .....	595
Boundary Conditions for the Transient Thermal Solver .....	595
Quench Analysis .....	597
Introduction .....	597
Modelling .....	597
Quench Thermal Analysis .....	598

---

Quench Multiphysics Analysis .....	598
Stress Analysis .....	599
Introduction .....	599
Material Properties .....	599
Internal Forces .....	599
External Forces and Constraints .....	600
Eigenvalue Frequency Range .....	600
Post-Processing .....	600
Scalar Potential Formulation .....	602
Vector Potential Formulation .....	605
Source Conductors and Circuits .....	606
The Time Harmonic Equations .....	606
The Transient Equations .....	607
Fixed Velocity Equations .....	607
The Finite Element Method .....	609
Finite Element Applications .....	612
Codes of Practice .....	612
Solution Errors .....	613
Boundary Conditions .....	615
Scalar Potential Boundary Conditions .....	615
Vector Potential Boundary Conditions .....	616
Open Boundaries .....	618
Total and Reduced Potential at the Open Boundary .....	618
Thin Plates .....	619
Surface Impedance Boundary Condition .....	620
Current Source Boundary Condition .....	622
Bulk Conductor Terminal .....	622
Electric Insulator Boundary Condition .....	623
External Forces and Constraints .....	623
Thermal Boundary Conditions .....	623
Time-Stepping .....	624
Time-Stepping Equations .....	624
Transient Driving Functions .....	624
Time-Step Control .....	626
Transient Analysis Logging and Control .....	627
Data Logging .....	627
Command File Control .....	629
Time-Step Summary .....	629
Restarting Transient Analyses .....	632
Accurate Fields .....	634
Magnetic Shielding .....	635
Advanced Features .....	636

---

Solution Fields and Field Averaging .....	640
<b>Chapter 7 Opera-3d Post-Processor .....</b>	<b>641</b>
Introduction .....	641
Commands Issued at Startup .....	642
Opera-3d Post-Processor Tabbed Menu Interface .....	643
Toolbar summary .....	643
Commands for Keyboard Entry – Command Line .....	653
Keyboard Shortcuts .....	656
Field Component Evaluation .....	657
System Variables .....	657
System Variables Defined in the Software .....	660
System Variables in Analysis Databases .....	661
Additional Field Quantities .....	665
Solution values .....	666
Circuit Values .....	667
Local Coordinate System .....	668
Results of Commands .....	668
Constants .....	669
User constants .....	670
User parameters .....	670
Expressions .....	670
Examples .....	671
Post-Processor Data Files .....	673
Opera-3d Database Files .....	673
GRID files .....	673
TABLE Files .....	674
TRACK files .....	677
The ACTIVATE Command .....	679
The ANIMATION Command .....	683
The ARC Command .....	685
The ARITHMETIC Command .....	687
The AXESVIEW Command .....	689
The BHDATA Command .....	702
The BODY Command .....	704
The BUFFER Command .....	708
The CARTESIAN Command .....	710
The CEDITOR Command .....	712
Circuit Editor Interaction .....	715
The CIRCLE Command .....	718
The CLEAR Command .....	720

---

The COLOUR Command .....	722
The COMBINE Command .....	725
The CONDUCTOR Command .....	727
The COPYCASE Command .....	733
The DATALINE Command .....	735
The DATAVECTOR Command .....	738
The DATAVECTORSET Command .....	740
The END Command .....	742
The ENERGY Command .....	743
Integration Method .....	745
Steady-state AC Models .....	745
Energy and Coenergy .....	745
The FIT Command .....	747
The FLUXLINKAGE Command .....	749
The GRAPH Command .....	751
The GRID Command .....	754
The GUIOPTIONS Command .....	756
The IDEAS Command .....	758
Finite Element Data .....	759
Results Data .....	760
Limitations for Quadratic Elements .....	760
Integration Commands .....	761
Line Integrals .....	761
Surface Integrals .....	761
Volume Integrals .....	761
The INTEGRATE Command .....	763
The ISOSURFACE Command .....	766
The LINE Command .....	768
The LOAD Command .....	770
The LOGGEDDATAFILE Command .....	772
The MAP Command .....	775
The MOUSE Command .....	779
The PICTURE Command .....	781
The PLOT Command .....	783
The POINT Command .....	787
The POLAR Command .....	788
The PRINT Command .....	790
The PROCESSLINE Command .....	791
The SELECT Command .....	793
The SET Command .....	798
Field Calculation Methods .....	800
Field Point Searching .....	801

---

Abort Field Calculations .....	802
Local Coordinate System .....	802
Steady-state AC Results .....	802
Current Density in Coils .....	803
Field Contours on Conductor Surfaces .....	803
Element Force Densities .....	803
The SHOW Command .....	804
The SIMULATION Command .....	806
The SPHERICAL Command .....	807
The SURFACE Command .....	809
The SYMMETRY Command .....	811
The SYSVARIABLE Command .....	814
Adding System Variables .....	815
Deleting System Variables .....	816
Listing System Variables .....	816
The TABLE Command .....	817
The THREED Command .....	823
The TITLE Command .....	829
The TRACK Command .....	831
The UNITS Command .....	835
System and String Variables .....	838
Internal Units .....	838
The VIEW Command .....	840
Types of VIEW .....	842
Emitter Names and Numbers .....	844
Viewing in Colour .....	845
Active System Variables .....	845
Printing Trajectory Data .....	847
The VOLUME Command .....	848
The WINDOW Command .....	851
System Variables in the Post-Processor .....	854
<b>Chapter 8 Python and Opera .....</b>	<b>866</b>
Introduction .....	866
Embedded Python .....	866
Accessing Documentation .....	867
Memory Scope .....	868
NumPy and SciPy .....	868
Installing Third Party Packages using pip .....	868
Useful resources .....	869
Limitations .....	869

---

Usage of NumPy in Opera Python .....	870
Contiguity .....	871
The operafea Embedded Python Module .....	872
Simple Usage .....	872
operafea Module Functions .....	873
The OperaObject Class .....	876
3d Solution Interface .....	878
Integrating Python Functions with Opera .....	886
Purpose .....	886
Terminology .....	886
Calculation functions .....	887
Solver Hooks .....	887
Python Arrays in Post-Processing .....	891
Opera to Python .....	891
Python to Opera .....	892
<b>Index .....</b>	<b>893</b>

# ***Chapter 1***

## **System Overview**

### **Introduction**

---

Opera-3d (an **O**Perating environment for **E**lectromagnetic **R**esearch and **A**nalysis) is a suite of multiphysics analysis programs that includes electromagnetics, stress and thermal analysis.

Finite element discretization forms the basis of the methods used in these analysis programs. This widely applicable technique for the solution of partial differential equations requires special enhancements to make it applicable to electromagnetic field calculations. Access to these features is supported by the Opera-3d Geometric Modeller and Pre-Processor. These programs provide facilities for the creation of finite element models, specification of complicated conductor geometry, definition of material characteristics including for example, nonlinear and anisotropic descriptions and graphical displays for examination of the data.

Similarly, the Opera-3d Post-Processor provides facilities necessary for calculating electromagnetic fields displaying them as graphs and contour maps. The Opera-3d Post-Processor can also calculate and display many derived quantities and can plot particle trajectories through the calculated fields.

This Reference Manual describes in detail the Opera-3d Modeller, the Pre and Post-Processors and the analysis programs. A separate chapter is given to Conductors which are used in almost all the programs.

- [Chapter 2: Command Language \[page 23\]](#)
- [Chapter 3: The Geometric Modeller \[page 85\]](#)
- [Chapter 4: Opera-3d Pre-Processor \[page 362\]](#)
- [Chapter 5: Conductors \[page 512\]](#)
- [Chapter 6: Analysis Programs \[page 539\]](#)
- [Chapter 7: Opera-3d Post-Processor \[page 641\]](#)

The rest of this chapter gives summaries of the analysis programs.

## Opera-3d Analysis Programs

---

There is a wide variety of solver types for different types of physics:

- magnetic fields,
- electric fields,
- low and high frequency electromagnetic fields,
- charged particles,
- thermal and mechanical effects, and
- coupled thermal and electromagnetic fields.

### Magnetostatic, Electrostatic and Current Flow Solvers

Three solvers are available for static magnetic and electric fields: Magnetostatic, Electrostatic and Current Flow. They are based on a formulation which uses total and reduced scalar potentials, solved using finite elements. Material properties can be nonlinear.

### Dynamic Electromagnetic Solvers

Time dependent electromagnetic fields in three dimensions are analysed using the Electromagnetic solvers. This includes the effects of eddy currents in conducting volumes. There are three analysis options:

- Fixed Velocity Electromagnetic in which eddy currents are induced in moving conductors with a specified linear or rotational velocity in the presence of a static field. Velocity EM is applicable in situations where the motion does not change the geometry, e.g. infinitely long rails or rotating disks.
- Harmonic Electromagnetic: all fields and currents oscillate at the same frequency.
- Transient Electromagnetic solver: the time variation is any function of time.

Two enhanced versions of the Transient Electromagnetic solver are described in the next two sections: Motional Electromagnetic and Magnetization.

### Motional Electromagnetic Solver

The analysis of transient electromagnetic fields in linear and rotating machines is performed by the Motional Electromagnetic solver. This includes the option of mechanical coupling to determine the speed of the moving parts.

## Magnetization Solver

Computation of the magnetization of permanent magnet materials by time varying electromagnetic fields in three dimensions including the effects of eddy currents is performed using the Magnetization solver.

## Charged Particle Solver

The analysis of electrostatic fields, taking into account the effects of space charge created by beams of charged particles, is performed by the Charged Particle solver.

## Thermal Solvers

Static and Transient Thermal fields are analysed by the thermal solvers. These fields may arise as a result of electromagnetic heating and external heat sources. Heat generated from electromagnetic heating can be included from Electromagnetic analyses. The calculated temperatures can be fed back to electromagnetic analyses in order to vary material properties.

## Stress Solvers

Displacements and stresses under load are calculated by the Static Stress solver. This can be used as a stand-alone stress analysis program, but is intended to be used to calculate the effects of electromagnetic forces or temperature changes.

The eigenvalues (modes) and eigenvectors of a model are calculated by the Modal Stress solver. This is valid in models without any internal or external forces applied.

## High Frequency Solvers

High frequency electromagnetic fields in three dimensions are analysed using the High Frequency solvers. There are two solvers:

- Harmonic HF to analyse at a predefined frequency and
- Modal HF to calculate the resonant modes of a cavity

## Quench Solvers

Simulation of a quench in a superconducting magnet is performed by the Quench solvers. These solvers model transient thermal fields including heat sources and magnetic fields produced by conductors driven by time varying circuit currents.

- Quench Thermal calculates its own magnetic fields from conductors.
- Quench Multiphysics couples with the Transient Electromagnetic solver which allows the inclusion of eddy currents in support structures.

## Old Names of Solvers

The names of the Opera-3d solvers have been changed from "operatic" names to names which indicate the physics being solved. There are still places in the Modeller, Pre and Post-Processors where the old names are used as parameter names or in option lists. The table below shows the old and new names of the solvers. Some solver names have equivalent short names.

<b>Old Name</b>		<b>New Name</b>	<b>Short Name</b>
CARMEN		Electromagnetic with Motion	Motional EM
DEMAG		Magnetization	
ELEKTRA	ELEKTRASS	Harmonic Electromagnetic	Harmonic EM
	ELEKTRATR	Transient Electromagnetic	Transient EM
	ELEKTRAVL	Fixed Velocity Electromagnetic	Velocity EM
QUENCH	stand alone	Quench Thermal	
	coupled	Quench Multiphysics	
SCALA		Charged Particle	
SOPRANO	SOPRANOEV	Modal High Frequency	Modal HF
	SOPRANOSS	Harmonic High Frequency	Harmonic HF
STRESS	STRESSEV	Modal Stress	
	STRESSST	Static Stress	
TEMPO	TEMPOST	Static Thermal	
	TEMPOTR	Transient Thermal	
TOSCA	TOSCACURR	Current Flow	
	TOSCAELEC	Electrostatic	
	TOSCAMAGN	Magnetostatic	

## Program Limits

The Modeller, analysis programs and Post-Processor have been written so that they should "grow" to accommodate any size of data up to the limit of available virtual memory (swap space). However, there are factors which limit the maximum problem size, the most significant being

- the maximum matrix size (number of non-zeroes) which is limited to  $2^{31}-2$  (=2147483646);
- the amount of physical memory required to avoid too much paging.

The largest model tested to date is a magnetostatic model with:

- 117649000 elements;
- 118370771 nodes;
- 118370770 equations;
- 1650690423 non-zeroes in the matrix.

It needed 60 GBytes of memory to solve.

The Opera-3d Pre-Processor has a limit on the maximum numbers of entities which can be created. The current limits are given in the following table:

<b>Opera-3d Pre-Processor Program Limits</b>	
internal database entities <sup>a</sup>	5000000
conductors available	no limit
conductors displayed	20000

<sup>a</sup>Entities = total number of points+lines+facets+volumes.

## Fundamental Physical Constants

---

The values of all fundamental physical constants used throughout Opera have been taken from the National Institute of Standards and Technology (NIST) Reference on Constants, Units and Uncertainty (CODATA 2014). For more information see <http://physics.nist.gov/cuu/Constants>.

# **Chapter 2**

# **Command Language**

## **Introduction**

---

The Geometric Modeller and Pre and Post-Processors of Opera-3d have a user interface which comprises both a command line and a Graphical User Interface (GUI).

The GUI generates text commands which have the same syntax as the commands which can be typed in directly at the keyboard. There are some features which can be used only from the keyboard. This chapter gives full details of the command line interface. In subsequent chapters, the keyboard commands are described with indications, where appropriate, of the corresponding GUI interaction. The use of the GUI is described in the User Guides; the use of the GUI for the Pre-Processor is given in [The Graphical User Interface \[page 368\]](#).

Within this manual, different fonts are used to differentiate between input and output of various types. The program's commands, parameters and keywords are shown in **RED**, user inputs and outputs from the program as **User Input** and **Program Output**. File names are shown as **file-name.txt** and GUI items are shown **like this**.

## Keyboard Input

---

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

name >

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

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

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

## Output Files

---

All user input and the responses from the program are stored in dialogue files:

- **Modeller\_n.ip** for the Modeller
- **Pre\_n.ip** for the Pre-Processor
- **Post\_n.ip** for the Post-Processor.

User input is written to files called

- **Modeller\_n.log** for the Modeller
- **Pre\_n.log** for the Pre-Processor
- **Post\_n.log** for the Post-Processor.

**log** files are in a format which can be used as input to the program with the **\$ COMINPUT** command. (**\$ COMINPUT** is described in [Command Input Files \[page 61\]](#).) The Pre-Processor input also written to a file called **Pre\_n.backup**. This file can be used in place of a Pre-Processor data file (**.oppre**).

A unique set of files is created for each run of the programs. The value of **n** is formed from the date, the time and the process identifier.

### Project folder

*ip, log and backup* files are stored in a sub-folder (sub-directory) of the Opera-3d project folder named *opera\_logs*.

The project folder can be changed using the **\$ PROJECTFOLDER** command. The *ip, log* and *backup* files are copied to the *opera\_logs* sub-folder of the new project folder. The syntax of the command is

- to set a new project folder:  
**\$ PROJECTFOLDER folder**  
The default value of **SET** is always **YES** so it does not need to be specified when setting a new project folder.
- to load the current project folder as the default value for the **FOLDER** parameter:  
**\$ PROJECTFOLDER SET=NO**

Additional output files can be created by the user to contain the program's usual output or user-defined output or both. The commands to do this are described in [File Input/Output Commands \[page 73\]](#).

## Commands and Parameters

---

Commands and parameters control the programs. All commands and their parameters may be shortened to their minimum unambiguous form. For example, in order to input the **THREED** command, any of the following character strings could be typed: **THREED** or **THREE** or **THRE** or **TH**. The single character **T** will not be sufficient, because other commands also begin with this character. The programs' command interpreter will return a message when an error is detected in the input. If **T** had been input as a command the interpreter would reply:

```
DCOD Message 2: Command 'T' ambiguous (CMND)
```

## The Help Character (!)

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

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

Opera > !

Valid commands are:

BHDATA Define and modify BH data  
SOLVE Create or update an analysis database  
READ Read a file of Opera-3d Pre-Processor data  
END End Opera-3d Pre-Processor  
\$... Built-in commands. Type '\$ !' for a list.

Opera > **read !**

Read a file of Opera-3d Pre-Processor data

Parameter	Value	Meaning
FILE		File name

## Parameter Assignment

---

Parameter values are specified either by entering an assignment instruction

`parameter=value`

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

`xorigin=value1 value2 value3`

implies that `value2` is assigned to the next parameter after `xorigin` and `value3` to the one after that. The parameter sequence for a command is fixed in the order listed by the help escape character `!`. When assignment instructions are used to specify the value of parameters the order is not important, except when using expressions which reference other parameters (see [Example: Parameter assignments: \[page 34\]](#)).

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

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

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

- Example: Using the following `THREED` command as an illustration (only some of the parameters are shown):

```
Opera > threed !
Start OpenGL 3d viewer
Parameter Value Meaning
SIZE      10.0  Half axis length
ROTX      20.0  Rotation around X axis
ROTY      20.0  Rotation around Y axis
ROTZ      0.0   Rotation around Z axis
```

- Example: Assignment instruction mode:

```
Opera > threed size=10 rotx=20 roty=20 rotz=0
```

- Example: Positional input mode

Opera > **threed size=10 20 20 0**

- Example: Mixed positional and assignment ( **ROTY** and **ROTZ** are set by position)

Opera > **threed rotx=20 20 0 size=10**

- Example: Missing positional input, ( **ROTY** takes last value by default)

Opera > **threed size=10 rotx=20,,0**

## Parameter Values

---

There are 5 types of value which may be assigned to a parameter: **Numeric**, **Expression**, **Character**, **Boolean** and **List**. Some parameters can take several value types but some combinations, such as character and expression are not allowed. Error messages indicate if an inappropriate value type has been used, e.g.

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

### Numeric Parameter Values

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

- Examples:

23

1.2

3E5

-5.789E+04

2.305983743795d5

-0.04

### Expressions in Parameter Values

Most parameters which can take numeric values can also take algebraic expressions to specify the values. Expressions used in this way for data input are a replacement for a calculator. Variables within such expressions can be other parameters, system variables or user variables (see [User Variable Commands \[page 55\]](#)). These input expressions are not remembered; they are evaluated and the result is stored.

Some parameters which can also take character values cannot take expressions as values. The text functions, **%INT**, **%NINT** and **%REAL** [page 41] provide a way of getting around this restriction.

Expressions are also used to specify output field quantities in post-processing. These expressions are remembered and used for evaluation when referenced. Variables in output expressions can also include the position and the field components. Full details are given in [System Variables \[page 657\]](#).

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

No spaces can be included within expressions.

### Numerical arithmetic

The following characters can be used in expressions to perform numerical calculations:

+	unary plus (identity)
-	unary minus (negation)
**	exponentiation
^	exponentiation
*	multiplication
/	division
+	addition
-	subtraction

If more than one operator exists in an expression, the above table shows the order in which the operations will be performed. Parentheses can be used to group variables and operators in order to specify the order of evaluation. The two ways of indicating exponentiation (\*\* and ^) can be used interchangeably.

To avoid run time errors and results which are undefined, the following rules are applied.

In **division**, e.g.  $a/b$ :

1. if  $a$  is zero the result is zero for all values of  $b$
2. else, if  $b$  is zero a "WARNING !! Divide check fix up taken" will occur and the result will be  $a$

In **exponentiation**, e.g.  $a^{**}b$  or  $a^b$  to evaluate  $a^b$ :

1. if  $b$  is zero, the result is 1 for all values of  $a$
2. else, if  $a$  is zero, the result is zero for all values of  $b$
3. else, if  $a$  is negative, the nearest integer to  $b$  is used instead of  $b$

See [Run Time Errors \[page 35\]](#) for more information.

## Logical arithmetic

The following characters can be used in expressions to perform logical calculations:

!	unary not
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

&&	and
	or

If more than one operator exists in an expression, the above table shows the order in which the operations will be performed. Parentheses can be used to group variables and operators in order to specify the order of evaluation.

In logical expressions, numerical values are considered to be true if the nearest integer value is non-zero; otherwise they are considered false. The result of a logical expression is either 1 for true or 0 for false.

## Equality expressions

Logical expressions which compare calculated numbers should be used with care. For example `#a==#b` will only return a value of 1 if `#a` and `#b` are *exactly* the same. If `#a` and `#b` represent integer values, the equality operator will give the expected answer but in many other situations exact equality is impossible to achieve, especially if the values to be compared are the results of analysis. In such cases, an expression using a tolerance such as `abs (#a - #b) < 1 . e-8` might be a better test.

## Mixed arithmetic

Numerical and logical arithmetic can be mixed in an expression. For example: `(B>0 . 5) *B+ (B<=0 . 5) *H` will result in the value of `B` when `B` is greater than 0 . 5 and the value of `H` when `B` is less than or equal to 0 . 5.

## Functions

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

FUNCTIONS	
<b>Arithmetic</b>	
<b>ABS (a)</b>	the modulus of <code>a</code>
<b>DELTA (a)</b>	1 if <code>NINT (a) = 0</code> , otherwise 0
<b>IF (a; b; c)<sup>a</sup></b>	<code>b</code> , if <code>a</code> is true (nearest integer is non-zero) <code>c</code> , if <code>a</code> is false (nearest integer is zero)
<b>INT (a)</b>	the largest integer whose magnitude does not exceed the magnitude of <code>a</code> times the sign of <code>a</code>

---

<sup>a</sup>Care should be taken not to confuse the function `IF(a;b;c)` with the command `$ IF Conditional Commands [page 52]`

<b>FUNCTIONS</b>	
<b>MAX (a; b)</b>	the maximum of <i>a</i> and <i>b</i>
<b>MIN (a; b)</b>	the minimum of <i>a</i> and <i>b</i>
<b>MOD (a; b)</b>	the remainder when <i>a</i> is divided by <i>b</i>
<b>NINT (a)</b>	the integer closest to <i>a</i>
<b>RAN (s)</b>	a pseudo-random number; the first number in a sequence, if <i>s</i> is non-zero; the next number in the sequence, if <i>s</i> is zero. Different non-zero integer values of <i>s</i> initialize different sequences.
<b>RANGE (a; b; c)</b>	inside range: 1 if <i>b</i> <= <i>a</i> <= <i>c</i> outside range: 1 if ( <i>a</i> <= <i>c</i> or <i>a</i> >= <i>b</i> ) and <i>b</i> > <i>c</i> otherwise: 0
<b>SIGN (a; b)</b>	the modulus of <i>a</i> times the sign of <i>b</i>
<b>SWITCH (a; b; c; d)</b>	<i>b</i> , if <i>a</i> <0 <i>c</i> , if <i>a</i> =0 <i>d</i> , if <i>a</i> >0
<b>Vectors</b>	
<b>DOT2 (a; b; x; y)</b>	the inner product of vectors (a,b) and (x,y), i.e. <i>a*x+b*y</i>
<b>DOT3 (a; b; c; x; y; z)</b>	the inner product of vectors (a,b,c) and (x,y,z), i.e. <i>a*x+b*y+c*z</i>
<b>SUP2 (rax; iax; ray; iay)</b>	the supremum <sup>a</sup> of a 2-dimensional complex vector
<b>SUP3 (rax; iax; ray; iay; raz; iaz)</b>	the supremum of a 3-dimensional complex vector
<b>VMOD2 (a; b)</b>	the length of vector (a,b), i.e. <b>SQRT (a^2+b^2)</b>
<b>VMOD3 (a; b; c)</b>	the length of vector (a,b,c), i.e. <b>SQRT (a^2+b^2+c^2)</b>
<b>Trigonometry (angles in radians)</b>	
<b>ACOS (a)</b>	the angle whose cosine is <i>a</i>

<sup>a</sup>The supremum is the maximum length of the vector at any time around the ac cycle. If the x and y (and z) components of the vector have the same phase angle, the supremum is the same as the dyadic modulus; if they are out of phase, the supremum is somewhat less.

<b>FUNCTIONS</b>	
<b>ASIN (a)</b>	the angle whose sine is $a$
<b>ATAN (a)</b>	the angle whose tangent is $a$ , angle in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$
<b>ATAN2 (a; b)</b>	the angle whose tangent is $a/b$ taking into account the signs of $a$ and $b$ and allowing $b$ to be zero, angle in the range $(-\pi, \pi)$
<b>COS (a)</b>	the cosine of $a$
<b>COSH (a)</b>	the hyperbolic cosine of $a$
<b>COTAN (a)</b>	the cotangent of $a$
<b>SIN (a)</b>	the sine of $a$
<b>SINH (a)</b>	the hyperbolic sine of $a$
<b>TAN (a)</b>	the tangent of $a$
<b>Trigonometry (angles in degrees)</b>	
<b>ACOSD (a)</b>	the angle whose cosine is $a$
<b>ASIND (a)</b>	the angle whose sine is $a$
<b>ATAND (a)</b>	the angle whose tangent is $a$ , angle in the range $(-90, 90)$
<b>ATAN2D (a; b)</b>	the angle whose tangent is $a/b$ taking into account the signs of $a$ and $b$ and allowing $b$ to be zero, angle in the range $(-180, 180)$ .
<b>COSD (a)</b>	the cosine of $a$
<b>SIND (a)</b>	the sine of $a$
<b>TAND (a)</b>	the tangent of $a$
<b>Exponentials and logarithms</b>	
<b>EXP (a)</b>	the value of $e^a$
<b>LOG (a)</b>	the natural logarithm of $a$
<b>LOG10 (a)</b>	the common logarithm of $a$
<b>SQRT (a)</b>	the square root of $a$

**N.B.** Functions with multiple arguments, e.g. **ATAN2**, **MOD**, etc., use semi-colon `;` to separate the arguments, since comma `,` is the separator between parameter assignments.

User defined functions can also be added (see [User Function Command \[page 60\]](#)).

- Example: Parameter assignments:

```
Opera > block two x0+1 y0+1 z0+1 x1+2 y1+2 z1+2
```

- Example: Output components in post-processing:  
Opera > `map comp=sqrt(x^2+y^2)*bx`
- Example: Functions can be concatenated:  
Opera > `$constant #pr max(abs(#x1);abs(#x2))`
- Example: The `MIN()` and `MAX()` functions can only have two arguments. In order to find the maximum of 4 numbers, the following expressions can be used:  
Opera > `$constant #pr max(max(#a,#b);max(#c,#d))`

(See [Numerical variables \[page 56\]](#) for the meaning of # in the above examples.)

## Run Time Errors

Arithmetic operations and function evaluations can lead to run time errors, for example `sqrt(#a)` where `#a` is negative is not allowed. The following rules apply to run time errors:

- If such an error is detected as an expression is being decoded, the error will be:  
DCOD Message 73: Expression 'sqrt(#a)' cannot be evaluated. (DECODE)
- If the expression was valid when decoded, but is subsequently invalid because, for example, the value of `#a` becomes negative, the error will become:  
WARNING !! SQRT argument < zero
- If an arithmetic operation or function evaluation gives an error but the result is subsequently multiplied by zero in the same expression, the error is ignored. For example, `sqrt(#a) * (#a>0)` will not result in an error if `#a` is negative, because in that case `(#a>0)` will equal zero.

## Character Values for Parameters

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

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

File names given as tree-names can include environment variables within the directory part of the name. Variables `$VFDIR` (Linux) and `%VFDIR%` (Windows) are defined by the software as the parent directory or folder holding the software.<sup>1</sup>

Some character strings are used for titles or text messages. For these strings, the rule about the first character being alphabetic can be relaxed. However any string which contains spaces or commas

---

<sup>1</sup>The maximum length of a file name or a folder name, whether or not it includes the complete folder tree, is 255 characters.

must be enclosed in quotation marks (''). Quotation marks embedded within character strings must be paired. The GUI automatically supplies quotation marks when necessary.

## Boolean Parameter Values

Boolean parameters take the values **YES** or **NO** and are in general used for switching features on or off. Boolean values can also be specified by **+PARAMETER** or **-PARAMETER**, being equivalent to **PARAMETER=YES** and **PARAMETER=NO**. The state of a Boolean parameter can also be changed from **NO** to **YES** (or from **YES** to **NO**) with **!PARAMETER**.

Some parameters can take Boolean or character values

- Examples:

```
Opera > disp +perspective  
Opera > disp pers=yes  
Opera > disp elem=no
```

## List Parameter Values

A list parameter value allows a list of character values to be assigned. A list parameter is cleared by using

**PARAMETER=**

It is cleared, and the first value set using

**PARAMETER=Value1**

Additional parameters can then be added using

**PARAMETER+=Value2, PARAMETER+=Value3**

## Text Functions

---

It is sometimes necessary to insert the value of an expression into a parameter value as a character string, for example, to include an index number in a file name or to supply a value by expression to a parameter which cannot take expressions. Some more sophisticated command files use various text-processing techniques such as substring find and replace. These things and many others can be done using text functions. The full list is given in table 2.1.

**Table 2.1Text Functions**

Function	Purpose	see
%COMPARE	compare two character strings	<a href="#">page 38</a>
%DESCRIPTION	return the description of a variable	<a href="#">page 39</a>
%ENV	return value of an environment variable	<a href="#">page 39</a>
%EXISTSTR	test whether a string variable exists	<a href="#">page 40</a>
%EXISTVAR	test whether a user variable exists	<a href="#">page 40</a>
%EXPR	return the expression	<a href="#">page 40</a>
%FILEBASENAME	return a file name without file type from a file name or file path	<a href="#">page 40</a>
%FILECOUNT	return the number of files in a folder or tests for file existence	<a href="#">page 41</a>
%FILEPATH	return a file path from a file name	<a href="#">page 41</a>
%FIND	return position of a substring in a character string	<a href="#">page 40</a>
%INT	format a user variable or an expression as an integer	<a href="#">page 41</a>
%NINT	format a user variable or an expression as an integer	<a href="#">page 41</a>
%QUOTE	quote a string	<a href="#">page 42</a>
%REPLACE	replace a substring in a string with another substring	<a href="#">page 43</a>
%REAL	format a user variable or an expression as a floating point number	<a href="#">page 41</a>
%STRENGTH	return the number of characters in a string	<a href="#">page 43</a>
%SUBSTR	return a substring from a character string	<a href="#">page 43</a>

## Syntax

Text functions have names which start with % and which are followed by one or more arguments. Some arguments are optional. If a function is successfully processed, all the text from % to the final bracket will be replaced by the result of the function.

The arguments of the functions are either character strings or numerical values. The arguments can be supplied as string or user variable names or expressions. The functions have the following forms:

```
%function(argument1, argument2, argument3)
%function{argument1} {argument2} {argument3}
%function[argument1] [argument2] [argument3]
%function<argument1><argument2><argument3>
```

There is a choice of bracket shape, (), {}, [], or <>, to delimit the arguments. Different rules apply for round brackets and the other bracket shapes.

### Round brackets

With round brackets, (), multiple arguments must be separated with commas, for example:

```
%real(bmod, 9)
```

If a function expects an algebraic expression as its first argument, (%INT, %NINT, %REAL), nested brackets within each argument are respected. All the other functions, which expect a character string as the first argument, respect quotation marks within each argument. Except in a quoted string, round brackets must not occur in the arguments.

### Curly, square or angle brackets

With other bracket shapes, {}, [], or <>, each argument of a multiple argument function must be contained in its own pair of brackets of the same shape, for example:

```
%compare<&analysis&><magnetostatic>
```

There is no respect for bracketing or quotation marks between the brackets which delimit each argument so it is important to choose a bracket shape which is not used within the arguments.

## Functions

### %COMPARE

**%COMPARE(a,b)** compares two character strings. It replaces itself with the numerical values

- -1 if *a* is less than *b*
- 0 if *a* is the same as *b*
- 1 if *a* is greater than *b*

where *less than*, *same as* and *greater than* refer to the positions of *a* and *b* in alphabetical order.

An optional third argument can be specified. If it exists and has the value **CS** (upper or lower case), then the comparison will be case sensitive; if it is omitted or has any other value, the comparison will be case insensitive.

For example command input files can be used to ask questions. The replies can now be given using character strings rather than numbers.

The command input file below processes results at different times for a steady-state ac or transient eddy current analysis:

```
$prompt type 'Steady-state or Transient (AC or TR)'
$if !%compare(&type&,ac)
  commands to display ac results, varying phase angle
$elseif !%compare(&type&,tr)
  commands to display transient results, loading different cases
$end if
```

## %DESCRIPTION

**%DESCRIPTION(variable)** replaces itself with the description of that variable. The variable can be a numeric user variable or a string variable.

## %ENV

**%ENV(envvar)** returns the value of the environment variable **envvar**, allowing command scripts to be programmed to behave differently for different users, processors, operating systems, etc. If the given variable is not defined, an empty string is returned. The following environment variables are defined for all operating systems:

<b>Variable</b>	<b>Meaning</b>		
		<b>Windows</b>	<b>Linux</b>
<b>COMPUTERNAME</b>	Network name of computer		
<b>CPU</b>	Processor type	AMD64	x86_64
<b>HOME</b>	Home directory or folder		
<b>OS</b>	Operating system name	Windows_NT	Linux
<b>VFBATCHFOLDER</b>	Opera Manager batch folder		
<b>VFDIR</b>	Opera installation folder		
<b>USERNAME</b>	User's login name		

For example, to restrict some commands to only run on a computer with a particular processor type:

```
$constant #isx86 1-abs(%compare(%env(CPU),x86))
$if #isx86
...
$end if
```

## %EXISTSTR

**%EXISTSTR(*stringvariable*)** returns whether a string variable exists. It replaces itself with the numerical values

- 0 if *stringvariable* does not exist
- 1 if *stringvariable* does exist

## %EXISTVAR

**%EXISTVAR(*variable*)** returns whether a user variable or system variable exists. It replaces itself with the numerical values

- 0 if *variable* does not exist
- 1 if *variable* does exist

## %EXPR

**%EXPR(*variable*)** replaces itself with the expression which *variable* represents.

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

```
Opera > threed comp=b/h
Opera > threed comp=%expr(comp) /mu0
```

This second command is equivalent to

```
Opera > threed comp=b/h/mu0
```

## %FIND

**%FIND(*string*,*substring*)** returns the position of the *substring* within the *string*. The first character position is number 1. A value of zero or less means that the *string* does not contain the *substring*.

An optional third argument can be specified. If it exists and has the value **CS** (upper or lower case), then the search for the substring will be case sensitive; if it is omitted or has any other value, the search will be case insensitive.

Examples are given with **%SUBSTR** [page 43].

## %FILEBASENAME

**%FILEBASENAME(*string*)** parses the *string* as a file name and extracts the base file name, omitting the folder and extension. The file does not need to exist.

## %FILECOUNT

`%FILECOUNT(folder)` returns the number of files and folders in the named `folder`, including "." (the folder itself) and ".." (the parent folder).

`%FILECOUNT(filepath)` returns 1 if the file exists or 0 if it does not. See [File Existence Command \[page 80\]](#) for another way of checking if a file exists.

`%FILECOUNT(folder,filter)` returns the number of files in the named `folder` which match the `filter`.

- Example:

```
%filecount(.,*.op*)
```

will give the total number of `op3`, `opc` and `opcb` files in the current folder.

## %FILEPATH

`%FILEPATH(string)` parses the `string` as a file name and returns the path of the folder containing the file. If the `string` does not include a path, the project folder will be returned. The file does not need to exist.

For example, the following command sets string variable `folder` to the path to the current project folder ([See "Project folder" on page 25.](#)):

```
$string folder %filepath(anyname)
```

## %INT, %NINT and %REAL

`%INT(expression)` or `%INT(expression,width)`

`%NINT(expression)` or `%NINT(expression,width)`

`%REAL(expression)` or `%REAL(expression,width)`

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

The forms of `%REAL`, `%INT` and `%NINT` with the second argument, `width`, allow the user to specify the number of characters used in the characters string representation of the value. For example, if `#a` has the value `1234`, then

- `%int(#a)` will be replaced by `1234`
- `%int[#a][0]` will be replaced by `1234`  
(`0` has the same effect as omitting the width)
- `%int{#a}{2}` will be replaced by `1234`  
(if the width is smaller than that required to represent the number, the actual width is used)
- `%int<#a><6>` will be replaced by `001234`  
(if the width is too large, leading zeroes are included)
- `%int(#a,10)` will be replaced by `0000001234`

- `%int{#a}{15}` will be replaced by `0000001234`  
(10 is the maximum allowable width)

The width can be specified in `%REAL` in a similar way. For example, if `#b` has the value `pi*1.0e6`, then

- `%real [#b]` will be replaced by `3.14159265358979E+06`  
(15 significant figures)
- `%real (#b, 0)` will be replaced by `3.14159265358979E+06`  
(0 has the same effect as omitting the width)
- `%real{#b}{2}` will be replaced by `3.E+06`  
(if the width is smaller than that required to represent the number, at least 1 significant figure will be shown)
- `%real<#b><10>` will be replaced by `3141592.65`  
(10 characters)
- `%real [#b] [20]` will be replaced by `3.14159265358979E+06`  
(15 significant figures is the maximum)

Examples of `%INT` and `%REAL`:

- to include a number represented by a user variable in the name of another variable:

```
Opera > $constant #i 3
#i=3
Opera > $constant #position_%int(#i,3) 10
#position_003=10
```

- to include a numerical result in a graph caption:

```
Opera > graph title='Flux density at time %real[ttime][10]'
```

(For more details on the commands used in these examples see [Numerical variables \[page 56\]](#) and [The GRAPH Command \[page 751\]](#).)

## %QUOTE

`%QUOTE(string)` returns a quoted string. Unless the string already has single quotation marks as the first and last characters, quotation marks will be added at the start and end of the string and any single quotation marks within the string will be doubled so that the string can be used as a parameter value.

For example:

```
Opera > $string f 'With ''internal'' quotes.'
f='A string with 'internal' quotes.'
Opera > $string s 'And here''s another.'
s='And here's another.'
Opera > $string joined %quote<&f& &s&>
joined='With 'internal' quotes. And here's another.'
```

## %REPLACE

**%REPLACE(*string,substring1,substring2,count*)**

returns the *string* with *substring1* replaced by *substring2*. If the optional *count* is specified, the substitution will happen at most *count* times. If *count* is zero or is omitted, all possible substitutions will be made.

**%REPLACE** is case-sensitive.

For example:

```
Opera > $string temp 'Results of xx.'
temp='Results of xx.'
Opera > $string fl model_2286.log
fl='model_2286.log'
Opera > $string title '%replace<&temp&><xx><&fl&>'
title='Results of model_2286.log.'
```

## %STRLENGTH

**%STRLENGTH(*string*)**

returns the number of characters in the *string*, ignoring any trailing spaces.

## %SUBSTR

**%SUBSTR(*string,i,j*)**

returns a substring starting with the character at position *i* through to the character at position *j*.

**%SUBSTR(*string,i*)**

returns a substring starting with the character at position *i* through to the end of the string.

The first character in the *string* is at position 1.

For example:

```
$string alphabet abcdefghijklmnopqrstuvwxyz
Name      String
          ALPHABET abcdefghijklmnopqrstuvwxyz
$constant #j %find<&alphabet&><j>
#j=10
$constant #r %find(&alphabet&,R,cs)
#r=0
$constant #r %find(&alphabet&,r,cs)
#r=18
$string j2r %substr[&alphabet&][#j][#r]
Name String
          J2R jklmnopqr
$string r2z %substr{&alphabet&}{#r}
Name String
          R2Z rstuvwxyz
```

See [%DESCRIPTION \[page 39\]](#) and [User Variable Commands \[page 55\]](#) for **\$ STRING** and **\$ CONSTANT** commands.

## Command Interpreter Errors

---

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

## Shortcuts and Reminders

---

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

### Repeated Commands

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

- For example:

```
Opera > threed size=100
Opera > size=10
```

### Prompted Input of Parameter Values

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

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

- Example:

```
Opera-3d > threed !!
There are 14 parameters
For each parameter type:
    return to accept the default
    OR a new value
    OR $HELP for help
    OR $SKIP to skip remaining parameters and execute command
    OR $ABORT to skip remaining parameters and abort command
NO.      Name      Value      Meaning
1        OPTION          Display option: GETVIEW, SETVIEW or INIT
!! > SETVIEW
2        SIZE       10      Half axis length
!! > 10
```

```
3      ROTX    20      Rotation around x axis
!! >  20
4      ROTY    20      Rotation around y axis
!! >  20
```

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

## Prompted Free Format Input

---

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

In some contexts, for example coordinate input in the Point Definition Mode of the Pre-Processor, values can be omitted in free-format input so that default, or previously specified values apply. The first value is omitted by using a comma at the start of the input line; subsequent values are omitted by using repeated commas within the line or by truncating the line after one or more values.

- Example: Boundary condition input in the Pre-Processor **DEFINE** command.

Define the boundary condition for this face e.g. POT 0.0  
OP-B/C > **pot 2 all**

The following line would be rejected since the keyword potential is not followed by a value.

OP-B/C > **potential all 2**

So that expressions can be used in free format input, each item is given a name according to its position on the line. #1 is the first; #2 the second etc. These names can be used to access defaults values in the input of coordinate positions or to use the values of earlier items on the same line.

- Example: Construction line input, the value of *u2* is set in terms of *u1*:

OP-C/LINES > **line 1.34\*sin(pi/6) 10 #2+20 10 0**

## Built-in Commands

---

"Built-in" commands provide control-structures (loops and conditions), user variables, command input from files and access to the operating system. "Built-in" commands can be used at almost any prompt. The exceptions are prompts in the **\$ ASK**, **\$ PAUSE** and **\$ PROMPT** commands and in defining Euler angles using **\$EULER** in the Pre-Processor (see [Euler Angles \[page 83\]](#)).

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

**\$ command parameter=value ...**

syntax can be used. The parameters have been ordered so that it is natural to use positional assignments. The parameter names are useful to provide on-line documentation using the **!** character.

Except where noted, there are no default values. **\$**-commands can be continued on subsequent lines but the command name must be specified at the start of each command.

A full list is given in table 2.2.

**Table 2.2 \$ Commands**

Command	Purpose	see
<b>\$ ABORTCOMI</b>	abort execution of a command file	<a href="#">page 63</a>
<b>\$ ASK</b>	request a constant value from user	<a href="#">page 63</a>
<b>\$ ASKPARAMETER</b>	request an expression from user	<a href="#">page 63</a>
<b>\$ ASSIGN</b>	assign format types to the data fields	<a href="#">page 75</a>
<b>\$ BACKSPACE</b>	backspace or rewind a file	<a href="#">page 75</a>
<b>\$ BREAKERROR</b>	break out of the current block (loop etc.)	<a href="#">page 63</a>
<b>\$ CD</b>	change current working directory	<a href="#">page 80</a>
<b>\$ CLOSE</b>	close a file	<a href="#">page 73</a>
<b>\$ COMINPUT</b>	command (or Python file) input and text output control	<a href="#">page 61</a>
<b>\$ COMMANDDIR</b>	set the directory for the command file menus	<a href="#">page 72</a>
<b>\$ CONSTANT</b>	assign a value to a user variable	<a href="#">page 55</a>
<b>\$ CYCLE</b>	jump to end of a loop	<a href="#">page 54</a>
<b>\$ DIALOG</b>	create a dialog from \$ commands	<a href="#">page 67</a>
<b>\$ DISPLAYLINE</b>	print a line to the console and screen if in <b>\$ DIALOG</b> or display an object in a dialog	<a href="#">page 65</a>
<b>\$ DO</b>	begin a new DO block	<a href="#">page 53</a>

Command	Purpose	see
\$ ELIF	begin an ELSE-IF block	<a href="#">page 52</a>
\$ ELSE	begin an ELSE block	<a href="#">page 52</a>
\$ END	end current DO, FOR, IF or WHILE block	<a href="#">page 54</a>
\$ EQUATION	define a user variable as the root of an equation	<a href="#">page 56</a>
\$ ERRORHANDLER	stop execution of commands if error occurs	<a href="#">page 63</a>
\$ EXEC	run a program	<a href="#">page 77</a>
\$ EXIST	test the existence of a file or folder	<a href="#">page 80</a>
\$ EXIT	jump out of loop or command input file	<a href="#">page 55</a>
\$ FILEPROMPT	request a file name from the user and assign it to a user variable	<a href="#">page 63</a>
\$ FOR	begin a new FOR each value block	<a href="#">page 54</a>
\$ FORMAT	define the formats for writing	<a href="#">page 74</a>
\$ FUNCTION	define a multi-dimensional tabulated function	<a href="#">page 60</a>
\$ GCONSTANT	assign a value to a user variable (with no output)	<a href="#">page 55</a>
\$ GPARAMETER	assign an expression to a user variable (with no output)	<a href="#">page 55</a>
\$ GROUPBOX	group prompting commands within a \$ DIALOG	<a href="#">page 70</a>
\$ IF	begin a new IF block	<a href="#">page 52</a>
\$ LAYOUT	layout prompting commands within a \$ DIALOG	<a href="#">page 70</a>
\$ MODELDIMENSION	assign a value to a model dimension	<a href="#">page 55</a>
\$ OPEN	open a file for reading or writing	<a href="#">page 73</a>
\$ OS	execute an operating system command	<a href="#">page 78</a>
\$ PARAMETER	assign an expression to a user variable	<a href="#">page 55</a>
\$ PAUSE	pause before continuing execution	<a href="#">page 63</a>
\$ PROJECTFOLDER	change the project folder	<a href="#">page 25</a>
\$ PROMPT	request a character string from user	<a href="#">page 63</a>

Command	Purpose	see
\$ PYTHON	interact with the embedded Python interpreter	<a href="#">page 65</a>
\$ READ	read data from a file	<a href="#">page 73</a>
\$ STRING	assign a string to a user variable	<a href="#">page 59</a>
\$ SYSVAR	assign a value to a system variable	<a href="#">page 60</a>
\$ TOOLBUTTON	assign a command file to a toolbutton	<a href="#">page 72</a>
\$ WHILE	begin a new WHILE block	<a href="#">page 54</a>
\$ WRITE	write data to files	<a href="#">page 74</a>

## Command Prompts

Some \$ commands (\$ DO, \$ FOR, \$ WHILE, \$ IF, \$ COMINPUT) start control blocks. During control blocks, the command prompt changes to indicate the current block type and level number. The level number is replaced with “-” if the logic prevents the block being executed. This is illustrated in the following table which shows the prompts for a sequence of \$ IF and \$ ELSE blocks.

Prompt	Command	Comments
Opera>	\$if true	Start block level 1.
Opera>\$IF[1]	commands	Commands will be executed because the condition is true.
Opera>\$IF[1]	\$if false	Start block level 2.
Opera>\$IF[-]	commands	Commands will not be executed because the condition is false. The block level is replaced with -.
Opera>\$IF[-]	\$else	
Opera>\$IF[2]	commands	Commands will be executed following \$ ELSE. Block level is now shown.
Opera>\$IF[2]	\$end if	
Opera>\$IF[1]	commands	Back to block level 1.
Opera>\$IF[1]	\$else	
Opera>\$IF[-]	commands	Commands will not be executed following the \$ ELSE at level 1, so block level replaced with -.
Opera>\$IF[-]	\$end if	
Opera>	commands	Out of all blocks.

## Limitations

The code which implements loops and control-structures has the limitation that control structures can be nested to a depth of 100 levels.

## Conditional Commands

Three conditional commands are available: **\$ IF**<sup>1</sup>, **\$ ELIF** and **\$ ELSE**. The commands **\$ IF** and **\$ ELIF** should be followed by a logical expression. The **\$ ELSE** command has no parameters.

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

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

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

The syntax is

```
$ IF logical_expression
  ... commands to be executed if logical expression is true ...
$ ELIF logical_expression
  ... commands to be executed if previous blocks have not been executed and logical expression is
true ...
$ ELSE
  ... commands to be executed if previous blocks have not been executed ...
$ END IF
```

## Logical expressions

There are 2 forms of logical expression which can be used with **\$ IF** and **\$ ELIF** and also **\$ WHILE**:

1. Compare two numerical values, variables or expressions:

```
ex1 logical_operator ex2
```

The **logical\_operator** must be one of **EQ** (equal), **NE** (not equal), **LE** (less than or equal), **LT** (less than), **GE** (greater than or equal) and **GT** (greater than).

Examples:

```
TIME GE 0.01
%COMPARE (&comfilename&,step.comi) EQ 0
```

2. Use a single value, variable or expression to obtain a logical value which is considered to be true if the nearest integer is non-zero and is otherwise false (see [Logical arithmetic \[page 31\]](#)).

Example:

```
TIME>=0.01 && %COMPARE (&comfilename&,step.comi)==0
```

---

<sup>1</sup>Care should be taken not to confuse the **\$ IF** command with the **IF (a;b;c)** function (see [Functions \[page 32\]](#)).

See [Equality expressions \[page 32\]](#) for more information on the use of `==` and `EQ`.

To enable the user to create command scripts which can be run in more than one program, the following variables are set by the software and can be tested using the `$ IF` command:

- `PREPROCESSOR`, has the value 1;
- `MODELLER`, has the value 2;
- `POSTPROCESSOR`, has the value 3;
- `OPERA2DPP` has the value 4;
- `PROGRAM`, has the value of the program currently being used.

For example, using the first form of logical expression in the `$ IF` command and the second in the `$ ELIF` commands:

```
$ if program eq modeller
... commands for Opera-3d Modeller ...
$ elif program==postprocessor
... commands for Opera-3d Post-Processor
$ elif program==opera2dpp
... commands for Opera-2d/PP
$ end if
```

## Loops

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

### `$ DO-loops`

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

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

The **index** should be the name of a user-variable (c.f. `$ CONSTANT`). Its value cannot be changed within the loop and is updated to the correct value at the start of the loop. See [User Variable Commands \[page 55\]](#) and [Scoped variables \[page 58\]](#) for more information on variables.

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

## \$ FOR-each loops

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

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

At least one, and at most 9, expressions (*ex<sub>n</sub>*) can be given. **index** is assigned in turn to each expression (c.f. \$ PARAMETER) at the start of the loop. The index variable cannot be redefined within the loop. See [User Variable Commands \[page 55\]](#) and [Scoped variables \[page 58\]](#) for more information on variables.

## \$ WHILE-loops

The \$ WHILE-loop executes a set of commands while a logical expression remains true. There are 2 forms of the command. The first is:

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

where **ex1** and **ex2** are numerical values, variables or expressions and the **logical\_operator** is one of EQ (equal), NE (not equal), LE (less than or equal), LT (less than), GE (greater than or equal) and GT (greater than).

Alternatively, **ex1** can be used as a logical value, variable or expression (see [Logical arithmetic \[page 31\]](#)):

```
$ WHILE ex1
... commands to be executed while ex1 is true ...
$ END WHILE
```

See also the examples given in [Logical expressions \[page 52\]](#).

## \$ END Command

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

```
$ END block_type
```

A \$ END COMINPUT is automatically inserted at the end of a command input file. It does not need to be specified by the user.

## \$ CYCLE command

The \$ CYCLE command jumps to the end of a loop (to the \$ END command). The syntax is

```
$ CYCLE blocks
```

where *blocks* can be

- zero: the **\$ CYCLE** command is ignored.
- 1 (the value assumed if *blocks* is omitted): the **\$ CYCLE** command jumps to the **\$ END** of the current loop.
- >1: the **\$ CYCLE** command jumps to the **\$ END** of the outer loop indicated by the value of *blocks*. The current loop is 1; the next outer loop is 2, etc. Only **\$ DO**, **\$ FOR** and **\$ WHILE** loops are counted.

If *blocks* is specified by a logical expression, the **\$ CYCLE** command can be conditional. For example, if #*i* is a **\$ DO** loop variable,

```
$ cycle mod(#i;3)
```

will jump to the **\$ END** of the loop each time #*i* is divisible by 3.

**\$ CYCLE** can be used in **\$ DO**, **\$ FOR** and **\$ WHILE** loops.

## \$ EXIT command

The **\$ EXIT** command jumps out of a loop or command input file (to the command after the **\$ END** command). The syntax is

```
$ EXIT blocks
```

where *blocks* can be

- zero: the **\$ EXIT** command is ignored.
- 1 (the value assumed if *blocks* is omitted): the **\$ EXIT** command jumps out of the current block.
- >1: the **\$ EXIT** command jumps out of the outer block indicated by the value of *blocks*. The current block is 1; the next outer block is 2, etc. Only **\$ DO**, **\$ FOR**, **\$ WHILE** and **\$ COMINPUT** blocks are counted.

If *blocks* is specified by a logical expression, the **\$ EXIT** command can be conditional. For example,

```
$ exit #time>10
```

will jump out of the current block if #*time* is greater than 10.

**\$ EXIT** can be used in **\$ DO**, **\$ FOR** and **\$ WHILE** loops and **\$ COMINPUT** files.

## User Variable Commands

The **\$ CONSTANT**, **\$ EQUATION**, **\$ MODELDIMENSION**, **\$ PARAMETER** and **\$ STRING** commands define user variables. Three further commands, **\$ ASK**, **\$ ASKPARAMETER** and **\$ PROMPT**, request the user to supply values for user variables and are described in [Prompting commands \[page 63\]](#).

## Numerical variables

**\$ CONSTANT**, **\$ EQUATION**, **\$ MODELDIMENSION** and **\$ PARAMETER** define numerical variables.<sup>1</sup> They each have the same three parameters. The first defines the **NAME** of the user variable, the second the **VALUE** and the third a **DESCRIPTION** of the variable. If the name is used again then the value for that variable is overwritten. If no **VALUE** is given, the current value for the **NAMEd** user variable is displayed. If **NAME=!** is used then all the user variables currently defined are listed.

Numerical user variable names start with # and have up to 63 additional characters which can be letters, numbers or any of \_ [ ] #\$. Variable names are not case-sensitive although the programs remember the case of the characters used when a variable is defined.

The **VALUE** can be a simple numeric value or can be an numerical or logical expression referencing other user variables or system variables.

- **\$ CONSTANTs** and **\$ MODELDIMENSIONs**:

the **VALUE** is calculated at the time the command is used and any expression is lost. The **VALUE** is retained.

Model dimensions are protected constants. They are intended for use with parametric modelling and optimization. They can also be set on the command line before the programs start. They cannot be set by command file prompting commands or by reading a data file.

- Example - to define a degrees to radians conversion factor:

```
Opera > $ constant #fac pi/180 'Degrees to radians'
#fac=0.0174532925199433 (Degrees to radians)
```

- **\$ PARAMETERs**:

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

- Example - to calculate **B.H**:

```
Opera > $ parameter #bdoth bx*hx+by*hy+bz*hz 'B.H'
#bdoth=BX*HX+BY*HY+BZ*HZ=5843.0853443269 (B.H)
```

- **\$ EQUATIONs**:

each time a **\$ EQUATION** variable is referenced, the equation is solved to find the value which should be used.

Equation variables are useful for calculating the flux density in anisotropic laminated materials and the proportion of current flowing in superconducting composite materials.

Equation solution is only guaranteed for single valued monotonic non-singular equations although solution might be possible for other types of equation. The **\$ CONSTANT** command can be used to assign an initial guess for the solution of an equation.

- Example - to solve  $f(x)=\cos(x)-x=0$ :

```
Opera > $ equation #x cos (#x) -#x 'Equation example'
#X=0.739085133215161 given by COS (#X) -#X=0 (Equation example)
```

---

<sup>1</sup>There are also commands **\$ GCONSTANT** and **\$ GPARAMETER** which are identical to **\$ CONSTANT** and **\$ PARAMETER** but give no output when used to define a new variable. They are intended for use by the GUI.

## Shortcuts for constants and parameters

Shortcuts are available for the **\$ CONSTANT** and **\$ PARAMETER** commands. Constants can be defined using `#name=value`, so the following commands for defining a constant are equivalent:

```
#aaa=3
$constant #aaa 3
```

Parameters can be defined using `#name:=expression` (note ":" rather than "="), so the following commands for defining a parameter are equivalent:

```
#bbb:=bx^2+by^2
$parameter #bbb bx^2+by^2
```

A description can be supplied when using a shortcut, for example:

```
#files=%filecount(..,*.res) 'Number of res files.'
```

## Programming with user variables

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

- Example

```
Opera > $ constant #a 3
#a=3
Opera > $ parameter #b #a^2
#b=#A^2=9
Opera > $ parameter #c #b-4
#c=#B-4=5
Opera > $ parameter #d #c==0
#d=#C==0=0
Opera > $ constant name=!
#a 3
#b #A^2=9
#c #B-4=5
#D #C==0=0
Opera > $ constant #a 2
#a=2
Opera > $ constant name=!
#A 2
#B #A^2=4
#c #B-4=0
#D #C==0=1
```

## Listing and deleting

Constants, model dimensions and parameter variables can be listed or deleted using any of the corresponding user variable commands. Equation variables can only be listed or deleted using **\$ EQUATION** (unless all variables are being deleted). When a single equation variable is deleted, it becomes a constant.

The syntax is, for example:

- to list the value of one variable:  
**\$constant #variable\_name**
- to list the values of all variables:  
**\$parameter name=!**
- to delete one user variable:  
**\$constant #variable\_name +delete**
- to delete all user variables (this cannot be done inside a command loop):  
**\$constant \* +delete**

All user variables (including equation variables) can be deleted using any of the commands **\$ CONSTANT**, **\$ MODELDIMENSION** or **\$ PARAMETER** using this syntax.

## Scoped variables

If a variable (other than a model dimension or equation variable) is defined with a name starting with **#** it will be assigned a scope and will not be available outside that scope.

The scope of a variable is set by the current control block (**\$ DO**, **\$ FOR**, **\$ WHILE**, **\$ IF**, **\$ COMINPUT**) when it is first defined. For example, a scoped variable first defined in a **\$ DO** loop will be available in the loop and any control blocks within the loop, but will be deleted as soon as the loop is ended.

A scoped variable defined outside any control blocks will have program scope. No scoped variables are saved in data files.

Scoped variables cannot be used in expressions for unscoped **\$ PARAMETERS**.

## Menus and toolbuttons for user variable commands

Pre-Processor menu routes:

**OPTIONS -> Parameters**  
**OPTIONS -> Constants**

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



The toolbutton **User Variables** gives access to the user variable commands.

In the Modeler user variables are defined using [The VARIABLE Command \[page 337\]](#).

## Character or string variables

Character variable names, defined with the **\$ STRING** command, can have up to 64 characters which can be letters, numbers or any of `_[]`. The first character must be a letter. Variable names are not case-sensitive although the programs remember the case of the characters used when a variable is defined.

The **VALUE** can be any character string except that it cannot have any trailing spaces. If the string contains spaces or commas it must be surrounded with quotation marks (''). Quotation marks within the value should be paired.

The character string can be recovered on almost any input line by use of the **NAME** surrounded by ampersands (&). Any quotation marks used to define the string are lost. This allows several strings to be concatenated. It might be necessary to use **%QUOTE []** to add quotation marks around the concatenated string. See [%QUOTE \[page 42\]](#) for more information and examples of strings and quotation marks. The limitations on the use of string variables are the same as for [Built-in Commands \[page 49\]](#).

Define or change the value of a string variable and optional description with:

**\$ string name value description**

Delete a string variable with:

**\$ string name +delete**

Delete all string variables with:

**\$ string \* +delete**

List the value of one string variable with:

**\$ string name**

List the values of all string variables with:

**\$ string name=!**

There are several predefined character variables and one predefined name which can be used if necessary.

- **NOW** and **TODAY** always hold the current time and date.
- **PROJECTFOLDER** contains the current project folder and **WORKINGFOLDER** the current working folder. The working folder is normally the same as the project folder. However if a program is started from the Manager by opening a data file, the working folder will be the folder containing the data file (see the **Opera Manager User Guide**).
- **VERSION** holds the version number of the software.
- **VFEXAMPLESDIR** holds the path to the folder of example files supplied with the software.
- In the analysis programs, **SOLVEFILENAME** holds the name of the analysis data file.
- **LOADFILENAME** holds the name of the last file loaded by the Modeller or Post-Processor.
- In the Post-Processor, **TITLE** holds the first line of the user title from the loaded database.
- **YESORNO** can be defined as **YES** or **NO** to pre-answer (and therefore avoid) questions the software might ask, e.g. before over-writing a file or ending a program. This is only used during

execution of a command file (see [Command Input Files \[page 61\]](#)); it is not used for commands issued from the GUI, typed on the console or issued using **F10** in the Command File Editor.

- Example - storing a title for later use. Note the use of **%real** to obtain a character representation of the value of a system variable (see [Text Functions \[page 37\]](#)):

```
Opera > $ string t1 'Septum Magnet'
t1=Septum Magnet
Opera > $ string t2 '(RMS error %real(#err)%)'
t2=(RMS error 5.23146%)
Opera > title '&t1& &t2&' topright
The title displayed is
'Septum Magnet (RMS error 5.23146%)'
```

## Toolbutton for \$ STRING



### String Variables

## Saving user variable commands

User variable commands can be written to a file which can be used as a command input file. This can be done using the **+uservariables** option on the **\$ OPEN** command (see [\\$ OPEN \[page 73\]](#)).

## System Variable Command

The **\$ SYSVAR** command can be used to define a system variable. This is needed so that expressions can be created in terms of system variables which would not normally exist, e.g. expressions for material properties in terms of field values created during analysis. The syntax is:

```
$ SYSVAR name value
```

If the value is omitted, the system variable can still be used in expressions but those expressions cannot be evaluated.

## User Function Command

The **\$ FUNCTION** command can be used to create user defined functions with up to 3 arguments which can be used in expressions (see [Expressions in Parameter Values \[page 30\]](#)). The syntax is:

```
$ FUNCTION filename
$ FUNCTION filename function_name
```

The command reads a file to obtain the tabulated data defining the function. The format of the file is the same as the Post-Processor Table File (see [TABLE Files \[page 674\]](#)).

The file should contain 2, 3 or 4 columns of data. The name of the **last** column in the table file specifies the function name but this can be overwritten by the **a** name supplied on the command line using the second form of the command shown above. The data in **last** column in the table file also

supplies the function values. The **first** 1, 2 or 3 columns give the argument values which should be arranged linearly or as a 2 or 3 dimensional rectangular grid.

Function names can only contain alphanumeric characters or underscore () and must start with an alphabetic character.

The function is evaluated by interpolating between the supplied data:

- for argument values which match data in the file, the corresponding function value is returned.
- for argument values not at specified data points, the programs interpolates between the surrounding 2, 4 or 8 function values.
- for argument values outside the range of data values, the value of the closest data arguments is returned.

Unit conversion is carried out on the arguments and function values using unit expressions in the file. The argument and function values in the table file should all be given in internal units .

Function definitions are stored in **opc** and **op3** files and can be exported to a table file using:

**\$ FUNCTION filename function\_name +SAVE**

The functions can be listed or deleted:

- **\$ FUNCTION +LIST**  
lists functions that have been defined.
- **\$ FUNCTION NAME=function\_name +DELETE**  
deletes the named function.

## Toolbutton for \$ FUNCTION



The **\$ FUNCTION** command can be run using the toolbutton

## Command Input Files

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

**\$ COMINPUT filename mode commentstyle**

If the **\$ COMINPUT** command appears in a loop, the file of commands is read each time the loop is executed. Any command can be included in a command input file; see the documentation on the **CLEAR** command in each program for limitations on its use in command files:

- Modeller: "The **CLEAR** Command" on page 180
- Pre-Processor: "The **CLEAR** Command" on page 380
- Post-Processor: "The **CLEAR** Command" on page 720

If file ***aaaa.comi*** contains a **\$ COMINPUT** command specifying a relative path name, the program will look for the file relative to the location of the file ***aaaa.comi***.

While a command input file is open, two string variables can be used:

- **COMIBASENAME** contains the name of the file (without **.comi**).
- **COMIPATH** contains the path of the folder containing the file.

They are both set empty when there is no open command input file.

The **\$ EXIT** command [page 55] can be used in a command input file to jump out, omitting all the subsequent commands.

The **\$ COMINPUT** command can also be used to run Python files, with the **.py** extension.

### Toolbutton for \$ COMINPUT



The **\$ COMINPUT** command can be run using the toolbutton .

### Text output modes

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

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

### Comment styles

Two different styles of comment are supported:

- **commentstyle=line** means that if a line starts with a comment character, **/**, the whole line is commented;
- **commentstyle=command** means that the line is split into separate commands by the command delimiter, **|**, before looking for comments.

See [Command Separator and Comments \[page 82\]](#) for more information on comments and multiple commands.

## \$ PAUSE command

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

```
$ PAUSE seconds
```

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

## \$ ABORTCOMI command

Execution of command files can be halted using the **\$ ABORTCOMI** command. The syntax is:

```
$ ABORTCOMI
```

## Error handling commands

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

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

While **\$ ERRORHANDLER** is set to **NO**, the **\$ BREAKERROR** command causes the command processor to exit the current loop if an error has been detected. This enables command loops which read (see [File Input/Output Commands \[page 73\]](#)) to the end of a file without knowing in advance the number of lines in the file. A typical sequence of commands could be:

```
$ errorhandler no
$ constant #i 1
$ while #i
$ read ...
$ breakerror
other commands
$ end while
$ errorhandler yes
```

(note that in this example the **\$ BREAKERROR** command directly follows the **\$ READ** command).

If the error handler is switched off (**no**) in a command input file, it will be switched on again (**yes**) automatically when the last **comi** file is closed and control returns to the user.

## Prompting commands

Command input files can contain user variables, which must be assigned values before other commands in the file are executed. The **\$ ASK** command can be used to request the user to supply a value for a numerical variable (c.f. the **\$ CONSTANT** command). The **\$ ASKPARAMETER** command can be used to request the user to supply an expression (c.f. the **\$ PARAMETER** command). The

**\$PROMPT** command can be used to request the user to supply a value for a character variable (c.f. the **\$STRING** command). The syntax is:

```
$ ASK #name prompt history edit
$ ASKPARAMETER #name prompt history edit
$ PROMPT name prompt history edit
```

A value must be supplied at the keyboard before the program will continue. The optional **prompt** is displayed to show what input is required. In [Dialogs \[page 67\]](#), different settings for **history** can be used to provide a drop down list of options alongside the text entry field, a checkbox or a pushbutton instead of the text entry field. The options for **history** are:

- **no**: this generates a text entry field with no drop-down list.
- **yes**: this generates a text entry field with a drop down list which accumulates the values given each time the dialog appears.
- the name of a predefined list of objects created by the program. For example, **VF\_CIRCUITELEMENT** contains a list of the circuit elements in the Modeller. The [\\$DIALOG LIST \[page 70\]](#) command can be used to find out the names of lists which exist. If **edit** is set to **yes**, the values given are not restricted to those in the list; if **edit** is set to **no**, only values in the list can be used.
- a list of options (values separated by commas or spaces, the complete list enclosed in single quotes). If **edit** is set to **yes**, the values given are not restricted to those in the list; if **edit** is set to **no**, only values in the list can be used. For example:

```
$ prompt material 'Material name' 'cu,fe,al' no
```

will permit only **cu**, **fe** or **al** to be entered as the material name.

If an option includes spaces, then that option should be enclosed in pairs of single quotes, e.g.  
**'copper ''magnet steel'' aluminium'**

- **checkbox**: this gives values **yes** or **1** if the checkbox is ticked and **no** or **0** if it is cleared.
- **pushbutton**: gives value **yes** or **1** if the dialog is closed using this pushbutton. If the dialog is closed using any other **pushbutton**, the variable is set to **no** or **0**.

A file name can be requested along with a browse button in graphical input using the **\$FILEPROMPT** command. The syntax is

```
$ FILEPROMPT name prompt load filter
```

Setting **load** to **yes** restricts the user to existing files and setting to **no** allows the entry of new file names. The **filter** string restricts the choice of files allowed. Multiple filters are delimited by **;** ;  
e.g.,

```
$fileprompt fi 'Table' yes 'tt(*.tt);;text(*.txt)'
```

N.B. **\$ASK**, **\$ASKPARAMTER**, **\$PROMPT** cannot be used in Opera-3d Pre-Processor data files; they can only be used in command input files. The history and edit parameters are not available. The **\$FILEPROMPT** is also not available.

## Displaying information

The **\$ DISPLAYLINE** command outputs text from the program, a picture or a link. The syntax is:

```
$ DISPLAYLINE 'text'
```

- System variables, user variables and string variables can be included in the text. e.g.

```
$displayline '&stringVar& is %REAL(#uservar) '
```

- Graphics can also be displayed in [Dialogs \[page 67\]](#): if the **text** is a file name (\*.bmp, \*.gif, \*.jpg, \*.jpeg, \*.png or \*.xpm) and the file exists, the graphic will be displayed on the dialog. If the file name is a relative path name, the program will look in the folder containing the current command input file.

Sizing information can be specified with the file name as horizontal and vertical sizes in pixels. A semicolon ( ; ) must be used as the separator between the file name and the first size and between the sizes. If one of the sizes is omitted or zero, the other size will be used and the aspect ratio of the picture will be maintained. If both sizes are omitted or zero, the picture will be displayed at its original size. For example:

- **picture.png;100** will scale the picture to 100 pixels horizontally and maintain the aspect ratio.
- **picture.png;;75** will scale the picture to 75 pixels vertically and preserve the aspect ratio.
- **picture.png;125;80** will scale the picture to 125 pixels horizontally and 80 pixels vertically.
- a link to a web page, an e-mail address or a file:
  - a web page which will be displayed using the system default web browser, e.g.,  
`text=<a href="www.web.com">see my web page</a>`
  - an e-mail address, using the default mail client, e.g.,  
`text=<a href="mailto:him@his.address">e-mail him</a>`
  - a file, which will be opened by the operating system using the program associated with it, e.g.,  
`text=<a href="file:filename">open this file</a>`
- In command input files used by the transient analysis programs, **\$ DISPLAYLINE** can be used to output information to the **res** file as the analysis proceeds.

## Opera Python

Python is a powerful, yet user friendly programming language, allowing relatively straightforward access to functionality such as advanced string manipulation and numerical analysis components. The Opera interface to Python allows this to be accessed from inside Opera during all phases of the Opera solution paradigm (modelling, solving and post-processing). It is implemented as an extension to the comi programming language, providing the functionality required to perform calculations on Opera user variables and database elements inside a Python environment.

Several examples of the use of Python in Opera are provided with your Opera distribution, and the documentation provided herein should allow for basic use of the interface. For users new to Python, we recommend reading the "Introduction to Python" as provided by [www.python.org](http://www.python.org) in order to understand general Python programming before attempting to implement custom Python modules.

## \$ PYTHON command

The **\$ PYTHON** command serves as a mediator between Opera and an embedded Python interpreter and allows the execution of Python statements and Python script files.

### Command line parameters

Command	\$ PYTHON	
Parameter	Default	Function
<b>COMMAND</b>	none	Python command to be executed
<b>OPTION</b>	<b>RESET</b>	Options:
-	-	<b>RESET</b> Resets Python interpreter content to the initial state
-	-	<b>LISTCALLBACKS</b> List all hook and calculation associated Python callback objects
		<b>CLONEBUFFERS</b> Make a copy of Post-Processor field buffers available to Python
		<b>IGNOREBUFFERS</b> Do not copy field buffers to Python
<b>FILE</b>	none	Path to the Python script file to be executed
<b>ARGS</b>	none	Arguments for Python script file (for example "--arg1 --arg2 --arg3")

### Use case examples

To execute a Python statement the **COMMAND** parameter may be omitted. Both statements below assign the value 2 to the Python variable **myvar**.

```
$PYTHON COMMAND='myvar=2'
$PYTHON 'myvar=2'
```

To execute multiline Python statements:

```
$PYTHON COMMAND='myvar=2\nprint('myvar=' ,myvar) '
```

The above code assigns number '2' to the Python variable 'myvar' and prints the value of 'myvar' to the Opera output (console/file). The '\n' symbol is used to define a new line, alternatively the return carriage symbols '\r' or '\r\n' can be used for the same purpose.

To execute a Python script file:

```
$PYTHON FILE='my_script_file.py'
```

To execute a Python script file with arguments:

```
$PYTHON FILE='my_script_file.py' ARGS='arg1 arg2 arg3'
```

To reset (clear) the embedded Python interpreter either the \$ PYTHON or the CLEAR command can be used:

```
$PYTHON OPTION=RESET
```

```
CLEAR PYTHON=YES
```

To list hook and calculation associated Python functions ("Integrating Python Functions with Opera" on page 886):

```
$PYTHON OPTION=LISTCALLBACKS
```

To change the way Opera-3d/Post makes field buffers available to Python:

```
$PYTHON OPTION=CLONEBUFFERS
```

```
$PYTHON OPTION=IGNOREBUFFERS
```

## Further Information

More information about the use of Python within Opera, and documentation of available utility Python modules distributed with Opera can be found [in Python and Opera](#).

## Dialogs

The \$ DIALOG command has two functions.

- It can construct [User Dialogs \[page 67\]](#) from a sequence of prompting commands and \$ DISPLAYLINE commands.
- It can display and issue commands from [Program defined dialogs \[page 72\]](#).

## User Dialogs

User dialogs are constructed from a sequence of prompting commands enclosed between \$ DIALOG commands. Other commands can be used to define the layout of the widgets in the dialog.

The syntax is:

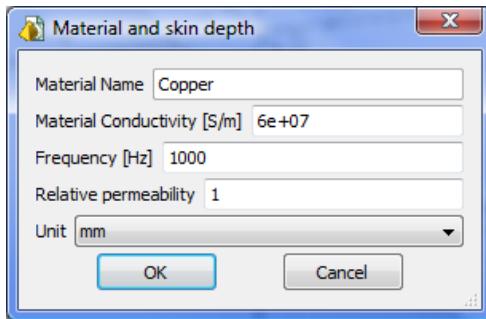
```
$ DIALOG START 'title'  
... $ask, $askparameter, $prompt and $displayline commands ...  
$ DIALOG STOP
```

The following example requests information from the user, processes it and displays a result using dialogs.

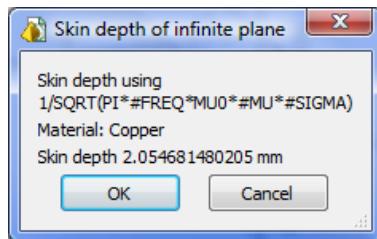
```
/ Preset some variables (only if they are not set yet)  
/ =====  
$if !%EXISTVAR(#sigma)  
  $CONSTANT NAME=#sigma VALUE=6E+7  
$END IF  
/  
$if !%EXISTVAR(#freq)  
  $CONSTANT NAME=#freq VALUE=1000  
$END IF  
/  
$if !%EXISTVAR(#MU)  
  $CONSTANT NAME=#MU VALUE=1  
$END IF  
/  
$if !%EXISTSTR(material)  
  $STRING NAME=material VALUE=Copper  
$END IF  
/  
$CONSTANT NAME=#unit_micron VALUE=1E6  
$CONSTANT NAME=#unit_mm VALUE=1E3  
$CONSTANT NAME=#unit_cm VALUE=1E2  
$CONSTANT NAME=#unit_m VALUE=1  
/  
$if !%EXISTSTR(userunit)  
  $STRING NAME=userunit VALUE=mm  
$END IF  
/  
/ Dialog to ask user information (with defaults set above)  
/ =====  
$DIALOG ACTION=START TITLE='Material and skin depth'  
$PROMPT NAME=material PROMPT='Material Name'  
$ASK NAME=#sigma PROMPT='Material Conductivity [S/m]'  
$ASK NAME=#freq PROMPT='Frequency [Hz]'  
$ASK NAME=#MU PROMPT='Relative permeability'  
$PROMPT NAME=userunit PROMPT='Unit' HISTORY='mm cm m micron' EDIT=NO  
/  
$DIALOG ACTION=STOP  
/  
/ Factor for units  
/ =====  
$STRING NAME=factor VALUE='#unit_&userunit'  
/  
/ The actual Equation  
/ =====  
$PARAMETER NAME=#skin VALUE=1/SQRT(pi*#freq*Mu0*#MU*#sigma)  
/  
/ Print out the results  
/ =====  
$DIALOG ACTION=start TITLE='Skin depth of infinite plane'  
$DISPLAYLINE TEXT='Skin depth using %expr(#skin)'  
$DISPLAYLINE TEXT='Material: &material'
```

```
$DISPLAYLINE TEXT='Skin depth %real(#skin*&factor&) &userunit&
$DIALOG ACTION=STOP
```

The first dialog looks like this:



and it displays its results like this:



## Wizard style dialogs

Options are available to allow wizard style user input. These options are used as alternatives to **\$DIALOG STOP** to close a dialog.

- **\$DIALOG FIRST**  
Creates a dialog with a **Next** button.
- **\$DIALOG NEXT**  
Creates a dialog with a **Back** and a **Next** button.
- **\$DIALOG LAST**  
Creates a dialog with a **Back** and a **Finish** button.

The identity of the button that was pressed to close the user dialog is available through the system variable **VF\_BUTTONPRESSED**. This has a value of 1 when the **Next** or **Finish** button is pressed, and a value of -1 when the **Back** button is pressed.

In all cases, the **Cancel** button is available to terminate the command input.

A wizard style user input can be created by using a command input file of the form shown below. In this a state variable, **#state**, tracks which dialog should be opened next and continues until the last dialog has been successfully finished.

```
$constant #state 0
```

```

$while #state<4
$if #state==0
$dialog start 'First dialog'
$displayline 'First dialog input'
$ask #a 'Value for #a'
$dialog first | / Next button
$constant #state #state+VF_ButtonPressed
$end if

$if #state==1
$dialog start 'Second dialog'
...
$dialog next | / Back or Next buttons
$constant #state #state+VF_ButtonPressed
$end if

$if #state==2
$dialog start 'Third dialog'
...
$dialog next
$constant #state #state+VF_ButtonPressed
$end if

$if #state==3
$dialog start 'Last dialog'
...
$dialog last | / Back or Finish buttons
$constant #state #state+VF_ButtonPressed
$end if
$end while

```

The **\$ DIALOG** command can also be used to list the history variables available to be used in the prompting commands. The syntax is

```
$ DIALOG LIST
```

The **\$ DIALOG** command and the dialog layout commands are not available in the Opera-3d Pre-Processor.

## Dialog layout

The **\$ GROUPBOX** command is used to organize objects (widgets) within a dialog. It provides a frame and a title. The syntax is

```

$ GROUPBOX START 'title' orientation strips
... $ask, $askparameter, $prompt, $displayline and other $groupbox commands ...
$ GROUPBOX STOP

```

The **orientation** of a groupbox is either **HORIZONTAL** or **VERTICAL**. A **HORIZONTAL** groupbox arranges its contents in rows, while a **VERTICAL** groupbox arranges them in columns. The number of **strips** defines the number of columns or rows.

The **\$ TABWIDGET** command creates a tabwidget in a dialog. The tabpanes are created using a set of **\$ GROUPBOX** commands between the **\$ tabwidget start** and **\$ tabwidget stop** commands.

The example below illustrates a dialog containing tabwidgets and other groupboxes. The **\$ DISPLAYLINE** commands say how each groupbox should appear.

```
$ dialog start 'MyDialog'

$ groupbox start 'group a'
$ displayline 'This is a normal groupbox.'
$ groupbox stop

$ tabwidget start
$ groupbox start 'tab a'
$ displayline 'This is a tabpane.'
$ groupbox stop

$ groupbox start 'tab b'
$ displayline 'This is another tabpane.'
$ groupbox start 'group b'
$ displayline 'This is a groupbox in "tab b".'
$ groupbox stop
$ groupbox stop

$ tabwidget stop

$ dialog stop
```

The **\$ LAYOUT** command is also used to organize widgets within a dialog. It does not provide a frame or a title. The syntax is

```
$ LAYOUT layouttype ON
... $ask, $askparameter, $prompt and $displayline commands ...
$ LAYOUT layouttype OFF
```

The **layouttype** is either **HLAYOUT** (horizontal), **VLAYOUT** (vertical) or **GRIDLAYOUT** (a grid).

A **\$ DIALOG** has an implicit **VLAYOUT** and a **\$ GROUPBOX** has an implicit **GRIDLAYOUT**.

The modifiers that can be used on a **HLAYOUT** or **VLAYOUT** are a stretchable space and a fixed space. The syntax is:

```
$ LAYOUT LAYOUTSTRETCH stretchfactor
$ LAYOUT LAYOUTSPACING spacing
```

The modifiers that can be used on a **GRIDLAYOUT** set the orientation, define the number of columns, span strips in the x and y direction and leave cells empty (skipping). The syntax is

```
$ LAYOUT ORIENTATION orientation
```

```
$ LAYOUT COLUMNS numberofcolumnsorrows
$ LAYOUT SPANX numberofcellstospan
$ LAYOUT SPANY numberofcellstospan
$ LAYOUT SKIP numberofcellstoskip
```

## Program defined dialogs

Most program defined dialog can be used by reference to its title. For example,

```
$ DIALOG SHOW 'Window Title'
```

will show the **Window Title** dialog and issue the **TITLE** command. Other examples could be:

```
$ DIALOG SHOW 'Create a Block'
```

```
$ DIALOG SHOW 'Set Magnetostatic material Properties'
```

You cannot use this command to open some of the dialogs, for example the BH Editor or the Circuit Editor.

The **\$ LAYOUT** command is not available in the Opera-3d Pre-Processor.

## Creating user toolbuttons

The **\$ TOOLBUTTON** command creates and deletes user toolbuttons in the command file toolbar.

The syntax is

```
$ TOOLBUTTON action comifilename position
```

Parameter	Default	Function	
<b>ACTION</b>	<b>LIST</b>	<b>APPENDCOMI</b>	Add to the end of list
		<b>INSERTCOMI</b>	Insert a toolbutton
		<b>LIST</b>	List all toolbuttons
		<b>DELETE</b>	Delete a toolbutton
<b>COMIFILENAME</b>		Name of the new command file to be associated with a toolbutton	
<b>POSITION</b>		Numeric position for the <b>ACTION</b> to take place	

If a picture exists with the same file name as the command file, but with the extension **.png**, **.bmp** or **xpm**, it will be used as an icon for the toolbutton. The preferred size for the picture is 16\*16 pixels, larger sizes will be scaled to fit the toolbutton.

The **\$ TOOLBUTTON** command can be accessed using:

**User Defined Toolbuttons** but the command is not available in the Opera-3d Pre-Processor.

The **\$ COMMANDDIR** command changes the location where the program searches for command files. The command files and any sub-directories are displayed under the 'Command files' top level menu. The syntax is

```
$ COMMANDDIR 'new directory'
```

If the new directory is set to ' ' then no command files are displayed.

## File Input/Output Commands

There are seven commands for user input and output of files.

### \$ OPEN

Before a file can be read or written it must be opened. The command:

```
$ OPEN stream file authority ±REDIRECT ±USERVARIABLES
```

opens a file on a logical stream number which can be in the range 1 to 50. The stream number used is stored in system variable **OPENSTREAM**. If stream has the value zero, the next available stream number will be used and the value of **OPENSTREAM** can be used for find out which stream has been opened ( see [Example \[page 75\]](#)).

The file can be used in 4 ways, depending on the **authority**. These are **READ** an old file, **WRITE** a new file, **OVERWRITE** an old file and **APPEND** to an old file.

If **+REDIRECT** is selected for an output file, the output which is written to the dialogue file will be written to the output file as well. This setting can be changed for an open file using:

```
$ OPEN stream ±REDIRECT
```

If **+USERVARIABLES** is specified for an output file, the current set of user variables will be written to the file as commands which can be read back into the software using the **\$ COMINPUT** command (see [Command Input Files \[page 61\]](#)).

### \$ CLOSE

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

```
$ CLOSE stream
```

closes a logical stream number in the range 1 to 50.

### \$ READ

The **\$ READ** command takes one line from the file opened on the given logical stream number and assigns any data on the line to user or string variables. Up to 99 variable names can be given. If no string variable names are given, any character strings on the line are ignored. The variable assignments are listed unless the **-PRINT** option is given. The syntax is:

```
$ READ stream var1 var2 var3 ... ±PRINT
```

The **\$ READ** command also saves the values read in program-defined string and numerical variables:

- string variable **readString00** holds the complete line;
- numerical variable **readValue00** holds the number of items read;

- up to 99 string variables `readString01`, `readString02`, etc., contain the items read;
- up to 99 numerical variables `readValue01`, `readValue02`, etc., contain the numerical values read or zero if an item cannot be interpreted as a number;
- the `readStringnn` and `readValue nn` variables are available until the next `$ READ` command.

## **\$ WRITE**

The `$ WRITE` command is similar to the `$ READ` command. The syntax is:

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

Up to 99 data items can be supplied.

A line of output can be built up using several `$ WRITE` commands. If this is necessary the first `$ WRITE` commands should have `-OUTPUT`. The last `$ WRITE` should have `+OUTPUT`. The data from the second and subsequent `$ WRITE` commands will be positioned after the data of the previous write commands in an internal buffer which is written and re-initialized when `+OUTPUT` is used. A different buffer is used for each open stream. The maximum buffer length is 16384 characters.

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

## **\$ FORMAT**

The `$ FORMAT` command can be used to define up to 99 different formats for output items. In each form the `width` can be specified as zero which implies that the program should calculate a width to fit the data being written. The syntax has one of the following forms.

- To define a format for character data supplied by a character value, truncated or padded with spaces to a particular `width`:  
`$ FORMAT number CHARACTER width`
- To define a format for character data supplied by the name of a string variable, truncated or padded with spaces to a particular `width`:  
`$ FORMAT number CHARACTER width +VARIABLE`
- To define a fixed point format for numerical data, with `decs` as the number of decimal places:  
`$ FORMAT number FIXED width decs`
- To define a floating point format for numerical data:  
`$ FORMAT number EXPONENTIAL width`
- To define a format for integer data:  
`$ FORMAT number INTEGER width`
- To define a format for a user variable to display the expression defining the variable truncated or padded to a particular `width`:  
`$ FORMAT number USER width`
- To define a character string to be output irrespective of the data on the `$ WRITE` command:  
`$ FORMAT number STRING width STRING=chars`

The default **STRING** is a single space. It is only necessary to specify **width** to define a string with more than 1 space, a non-empty string with trailing spaces or to reduce the length of the **STRING**. It is necessary to define **STRING** formats containing spaces to appear between other data items if those other data items are written with widths of zero.

- To list the defined formats:

```
$ FORMAT +LIST
```

## \$ ASSIGN

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

```
$ ASSIGN form1 form2 form3 ...
```

Up to 99 formats can be assigned, with no default format assumed. All format assignments can be removed using

```
$ ASSIGN 0
```

## \$ BACKSPACE

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

```
$ BACKSPACE stream records
```

where **stream** is the logical stream number and **records** is the number of records the file is to be backspaced.

- If **records** is given as -1, the file is positioned at its start (rewound).
- If **stream** is given as 0 with **records** as -1, the dialog file (**\*.lp**) is rewound.
- If **stream** is given as -1 with **records** as -1, the log file (**\*.log**) is rewound.

There is no default value for **stream**; the default value of **records** is always 1.

## Example

A file **points.dat** contains:

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

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

```
/ Open input file for reading and find stream number
$ open 0 points.dat read
$ constant #input openstream
/ Read the first line into user variable #np
$ read #input #np
/ Open output file for writing and find stream number
$ open 0 fields.dat write
$ constant #output openstream
/ Define a string format to space the output
$ format 1 string
/ Define a floating point format
$ format 2 exponential 0
/ Assign format numbers for the output
$ assign 2 1 2 1 2 1 2
/ Start a loop from 1 to #np
$ do #i 1 #np
/ Read #x and #y from input file
$ read #input #x #y
/ Evaluate fields at #x #y
poin xp=#x yp=#y zp=0
/ Write coordinates and flux density to output file
$ write #output x y bx by
/ End of loop
$ end do
/ Close files
$ close #input
$ close #output
```

The output file, **fields.dat** contains:

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

## Example

A second example showing the features mentioned above is as follows. A file **blocks.dat** is created that contains the definitions of some cubes to be defined in the Modeller. The data file contains

```
one 0 0 0 5
two 10 0 0 5
three 20 0 0 5
four 5 5 0 5
five 15 5 0 5
six 10 10 0 5
```

The following command file could be used to read **blocks.dat** and create the data:

```
$open 1 blocks.dat read
$constant #a 1
```

```
$errorhandler no
$while #a
$read 1 name #x #y #z #l -print
$breakerror
block &name& #x #y #z #x+#l #y+#l #z+#l
$end while
$cclose 1
```

(note that in this example the **\$ BREAKERROR** command directly follows the **\$ READ** command).

## Commands to Run Other Programs

There are two commands to execute other programs:

```
$ EXEC program arguments ±WAIT
$ OS str1 str2 str3 str4 str5 str6
```

### **\$ EXEC**

The **\$ EXEC** command starts a program from an interactive program or from time-stepping control command file during analysis. It has 3 parameters.

- **PROGRAM** specifies the executable program file.
- **ARGUMENTS** is a list parameter specifying the arguments.

The first argument should be given by

```
arguments='first argument'
```

Second and subsequent arguments can be added by

```
arguments+='other argument'
```

- **WAIT** is a Boolean parameter which can be used to wait until the program has completed.

**-WAIT** returns after the new process has started. This is the default.

**+WAIT** returns after the new process has completed and displays the exit code.

The functionality is similar to the **\$ OS** [page 78] command with the following differences:

- no output is reported back to the calling program except an exit code if **+WAIT** is specified;
- operating system commands cannot be run;
- no double quotes ("") are required, only single quotes around file names or arguments which contain spaces.

When an external command is executed with the **\$ EXEC** command and option **+WAIT**, the variable **EXECEXITCODE** is set to the exit code from the command. The value of **EXECEXITCODE** can be tested using the **\$ IF** command to determine whether the external command has been successful.

Example: the following commands add a data file to the top of the Manager batch queue with a post-processing command file. Note that the command name has to be used on every **\$ EXEC** command when the command is continued on a second or subsequent line.

```
$constant ##linux %compare{${ENV(OS)} {Linux}}==0

$string exefolder '%ENV(VFDIR)'
```

```

$if ##linux
// LINUX
$string manager &exefolder&/bin/opera_manager
$exec &manager&,
$exec arguments=example1.op3,
$exec arguments+=ex1_post.comi,
$exec arguments+=-top
$else
// WINDOWS
$string manager &exefolder&\bin\opera_manager.exe
$exec &manager&,
$exec arguments=example1.op3,
$exec arguments+=ex1_post.comi,
$exec arguments+=/top
$end if

```

## \$ OS

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

### \$ OS command on Linux systems

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

- in c-shell: **user\_command >& TeMpOsCmNdFiLe**
- in other shells: **user\_command > TeMpOsCmNdFiLe 2>&1**

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

**\$OS (xterm &)**

### \$ OS command on Windows systems

On Windows systems additional command names have been implemented so that the same commands exist on Windows and Linux. The additional commands and the equivalent Windows commands are:

- **ls** to list files in the current folder in the same way as **dir**
- **rm** to delete a file in the same way as **del**
- **mkdir** to create a new folder in the current folder in the same way as **md**

- `mv` to rename a file in the same way as `ren`

N.B. `mv` or `ren` cannot be used to move a file from one folder to another; the second file name on the command cannot have a path. To move between folders use `cp` or `copy` followed by `rm` or `del`.

- `cp` to copy a file in the same way as `copy`
- `pwd` to report the current folder in the same way as `cd`

**N.B.** File names including spaces should be enclosed in double-quotes ("") in `$ OS` commands. If the complete command contains more than one pair of double-quotes, the whole command should be enclosed in another set of double-quotes.

## Additional `$ OS` commands to launch opera

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

```
$ OS operapre mode datafile
$ OS operapost mode datafile options
$ OS operamodeller mode datafile options
```

- *mode* is **FORE** or **BACK**. The interactive program waits while **FORE**ground analysis jobs are run, but can be continued or ended while **BACK**ground jobs are run.
- *datafile* is optional. If it is given, it should be the name of a Pre-Processor data file (`*.oppre`), a Modeller data file (`*.opc`, `*.opcb`) or a Opera-3d database (`*.op3`) to be opened or the name of a command input file (`*.comi`) to be run.
- *options* are described in "Running Opera outside the Manager" in the ***Opera Manager User Guide***.

Analysis programs are started by adding them to the Opera Manager batch queue. The command for the analysis programs is:

```
$ OS operaanal program datafile mode comifile
```

- *program* is one of **CARMEN**, **DEMAG**, **ELEKTRASS**, **ELEKTRATR**, **ELEKTRAVL**, **QUENCH**, **SCALA**, **SOPRANOEV**, **SOPRANOS**, **STRESSST**, **STRESSEV**, **TEMPOST**, **TEMPOTR** or **TOSCA**. (The name of *program* is ignored because the Opera Manager looks into the data file to see which type of analysis is required.)
- *datafile* gives the name of the Opera-3d database (`*.op3`) to be analysed.
- *mode* is **FORE** or **BACK**. The interactive program waits while **FORE**ground analysis jobs are run; **BACK**ground analyses are added to the Opera Manager batch queue and are run under its control.
- *comifile* is an optional post-processing command input file. It can only be used with **BACK**ground analyses. If it exists, the post-processing session will be run automatically and without a window when the analysis is complete.

## The \$ CD command

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

```
$ OS cd directory
```

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

The **\$ COMMANDDIR** command is not available in the Opera-3d Pre-Processor.

## File Existence Command

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

```
$ EXIST filename
```

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

If the file exists, the **FILETYPE** system variable is also set to 1 for a normal file or 2 for a folder or directory.

See **%FILECOUNT** [page 41] for another way of checking whether a file exists.

## Command File Editor

---



The GUI gives access to the Command File Editor (toolbutton ). The editor supports many features useful for working with command files including:

- keyword high-lighting;
- running the command file currently being edited;
- executing single commands or groups of selected commands in a file using **F10**;
- copying the contents of the current log file; or subsets of the Modeller command history,
- copy, cut, paste (use **ctrl-C**, **ctrl-X** and **ctrl-V** or the right mouse button);
- search and replace, optionally through all open files;
- search through multiple files;
- open other files named in commands;
- "regular expressions" in search strings;
- commenting and un-commenting selected lines;
- command dictionary available (double-click on list to add command and parameters to the command file, or **tab** to complete words).

The Command File Editor is intended to be self-documenting with tooltips available for the toolbar buttons.

The Command File Editor is not available in the Opera-3d Pre-Processor.

## Command Separator and Comments

---

### Command Separator |

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

- Example using the LINE and PLOT commands:

```
Opera > line 0 0 0 10 0 0 | plot comp=by
```

If one of the commands is \$ COMINPUT, the command file will be opened but all the other commands on the line will be processed before any commands are read from the command input file.

- Example:

```
Opera >$cominput regions.comi | $constant #a 10
```

will be run in the following order:

1. open command file;
2. set value of #a;
3. run commands from `regions.comi`.

### Comments /

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

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

Note that if multiple commands exist on a single line (separated with |), then starting the line with a comment character (/) will prevent all the commands on that line from executing (the comment character applies to the whole line, not just to the first command on the line). This behaviour can be changed using the \$ COMINPUT command with `commentstyle=command` (see [Comment styles \[page 62\]](#)).

## Euler Angles

---

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

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

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

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

- Examples:

Coordinate Transformation	$\Theta$	$\Phi$	$\Psi$
$XYZ_{local} = XYZ_{global}$	0	0	0
$XYZ_{local} = YZX_{global}$	90	0	90
$XYZ_{local} = ZXY_{global}$	90	90	180

Whenever the ordered triple for Euler angles is to be entered into the Pre-Processor, the escape function **\$EULER** can be typed instead. The program prompts the user to define the Euler angles by specifying **THREE** points or by **SWAPPING** the coordinate axes. The 3 points required are a local origin, and points on the local Z axis and either the local X or the local Y axis. The coordinate axes can be swapped by specifying the global axes which correspond to the local X and Y axes. The global axes, in either positive or negative directions, are referred to by keywords, **POSX**, **NEGX**, **POSY**, **NEY**, **POSZ**, **NEGZ**. The Euler angles can be stored within Opera-3d under user defined names and recalled later by name.

- Example: Setting material **VECTOR** property in the Pre-Processor.

Set the material name, potential type and constants (or HELP or QUIT)  
**OP-MATERIALS > iron total vector \$euler**

Give name of Euler angle set. Type LIST for a list or NEW to define a new Euler angle set.  
**OP-EULER > new**

Give name and definition method. Methods are:

THREE - coordinates of origin, and points on z and one other axis

```
SWAP - swap global axes
OP-EULER > iron swap
Give a combination of POSX, NEGX, ..., POSZ, NEGZ to define local X, Y
and Z axes. (Only X and Y need be given.)
OP-EULER > posy posz
Input line edited to replace $EULER:
iron total vector 90 0 90
```

# ***Chapter 3***

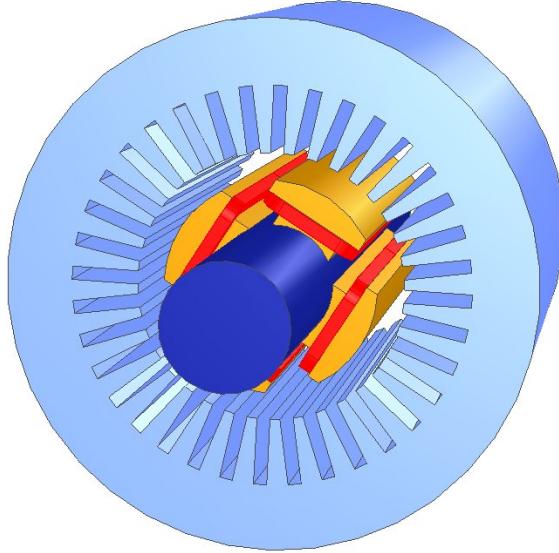
# **The Geometric Modeller**

## **Introduction**

---

The Geometric Modeller provides facilities for creating models for use with the Opera-3d analysis modules and Post-Processor.

The Modeller manipulates any defined objects through operations such as transformations and combinations. Basic objects (blocks, cylinders, spheres, cones, pyramids and toroids) can be created at any position in space. Once created they can be manipulated to reposition them. They can also be merged, intersected or subtracted from other objects in space to create more complex geometries.



*Figure 3.1 Complex geometry created with the Modeller*

This modelling technique allows many models to be created from the basic building blocks. Other more advanced techniques allow the geometry to be modified e.g. by sweeping an existing face or by morphing a cell to a different shape using an algebraic expression.

Data can be read from other geometric modelling programs or saved in formats which can be read by other geometric modelling programs (see [The LOAD Command \[page 243\]](#) and [The SAVE Command \[page 297\]](#)). The formats supported<sup>1</sup> include SAT, IGES, CATIA, STEP and Pro/E.

---

<sup>1</sup>The use of IGES, CATIA, STEP and Pro/E formats is license controlled.

## Commands Issued at Startup

---

Commands can be run and files read as the Modeller starts. They can be specified by a command input file and as command line arguments. The sequence of operations is as follows:

1. Read the initial command file. The program looks for a file called *modeller.comi*, first in the current directory and then in the user's home directory, and if it exists, executes the commands it contains (see [Command Input Files \[page 61\]](#)).
2. Load a **data\_file** given on the command line followed by a bounding-box zoom command (see [The LOAD Command \[page 243\]](#)).
3. Define model dimension variables given on the command line (see [The VARIABLE Command \[page 337\]](#)).
4. Read a **command\_file** given on the command line.
5. Run other commands given on the command line.

See the "Opera Manager User Guide", *Running Outside the Opera Manager* for details of the command line syntax.

Operation 1 above is also run when the program is restarted with [The CLEAR Command \[page 180\]](#).

# Topology Definitions

---

## Introduction

This section introduces commonly used terms and their meaning within the Modeller.

## Geometric Entities

### Primitive

A primitive is one of the 3-dimensional geometric shapes from which a model can be built. Primitives can have one of the following shapes: block, cylinder (including tube and polygonal prism), sphere, torus and wire edge. Primitives can be combined and altered in various ways to make more complex shapes.

### Vertex

This is a simple point in space, defined by a position within a cartesian coordinate system. Normally a vertex will mark a sharp change in geometry, e.g. the intersection of 2 or more edges.

### Edge

An edge is a line in space. Typically an edge will have 2 vertices defining its ends. The edge can also have an underlying geometry associated with it, e.g. it could be straight, an arc of a circle, or a more complex underlying spline curve. For some edges, e.g. circles and ellipses, the edge may have a single vertex or no end vertices at all.

### Wire edge

An edge often exists as a boundary to a face, but may exist independently as a wire edge. A wire edge can be created by copying an existing edge or by using the **WIREEDGE** command.

### Face

A set of edges, connected together at their ends, forms a loop. For most cases, such a loop forms the boundary of a face. A face also has an underlying geometry, e.g. the edges could bound part of a plane, part of a sphere or part of a more complex underlying surface definition.

There are also some cases where a face is bounded by more or less than a single loop.

- For some underlying surface geometries, e.g. a sphere or torus, there may be no bounding loop of edges required.

- For structures formed from complex operations, the face may contain internal holes. In this case there may be one or more internal loops of edges bounding interior sections of the surface to create the holes in the face.

## Sheet face

Typically a face will form part of a closed set of faces that completely encloses a volume, but it can exist independently as a sheet face. A sheet face can be created by copying a existing face or by creating a primitive (e.g. a block) with no thickness.

## Cell

A closed set of faces connected at the bounding edges forms a shell. A cell represents an enclosed volume of space bounded by one or more shells. Any point within the cell can be reached from another point within the cell by travelling a path that does not pass through a face. For some complex geometries there may be more than a single shell of faces bounding a cell, e.g. where a section has been removed from the interior of a volume, leaving an internal set of faces restricting the volume occupied by the cell.

## Body

A body is a collection of any of the topological entities discussed above. All objects within a body are topologically linked, so that parts of a body cannot be moved without considering the impact on other parts of the body. For example a body may contain 2 cells. These cells each occupy their own space. Movement of one cell within the body could make the cells overlap which would cause the cells to become invalid.

However, if there are 2 bodies, each with one cell, it is permissible to have the bodies overlapping as there is no connection between the bodies. Before forming a final model, the bodies will be merged into a single body to ensure that the topology forms a single valid structure.

## Local coordinate system

A local coordinate system (LCS) is specified by a position for the origin, and a rotation of the local axes with respect to the global coordinate system.

## Conductor

Special primitives can be defined in the Modeller to define standard conductor shapes. These include racetrack coils, solenoids and others.

## Entity

The term entity is used to include any of the terms defined above.

## Working coordinate system

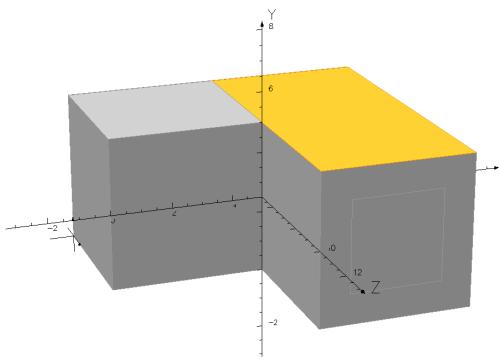
A local coordinate system can be chosen as a Working Coordinate System (WCS). While a WCS is active, values given for operations are in the Working Coordinate System, rather than in the Global Coordinate System.

## Geometric Model Information

The geometric model held within the Modeller consists of a set of bodies, with their component cells, faces, edges and vertices, Local Coordinate Systems and Conductors.

The information about what is contained within the model is always available in the status bar at the bottom right of the screen. Each number represents the number of entities of a specific type, defined in the current model. Leaving the cursor on any of the entries in the status bar will bring up a tooltip giving more information about its meaning.

For the model, in [Figure 3.2](#), the status bar information at the bottom right of the window, shows that the model consists of 3 bodies; 3 cells; 18 faces (with 1 of them picked); 36 edges; 24 vertices; no conductors.



Component | 3 | 3 | 18(1) | 36 | 24 | 0 | Global coordinate system

*Figure 3.2 Entities displayed by the Modeller*

## Bodies and Cells

A common confusion when first using the Modeller is the misunderstanding of differences between bodies and cells.

When creating a body using a basic building object, e.g. a block, the object created is a single body, containing 1 cell, 6 faces, 12 edges and 8 vertices. The close correspondence between a body and a cell in this instance makes it hard to differentiate between them.

However, after bodies have been combined using non-regular operations, a single body may contain more than 1 cell, and may contain no cells if they have only sheet faces.

Whenever performing geometric operations, they are almost always performed on a body rather than on a cell. This is because the topology of connected entities must remain consistent within a body. By trying to move a cell within a body, all connected faces and edges would be affected and may well become invalid.

Similarly, if material properties were defined on a body, only one material would be available for each body, making separation of materials within a body very difficult.

# Geometric Modelling

---

## Modeller Functionality

### Creating new objects

Basic primitive commands exist for creating objects. The type of objects that can be created are:

Type	Command	Description
Block	The BLOCK Command [page 148]	Creates a cuboid block. Specify one corner and its opposite to define a cuboid.
Cylinder, cone, n-sided prism, tube	The CYLINDER Command [page 195]	Specify the positions at the centre of the base and centre of the top as well as the radius. Hollow cylinders (tubes) can be created by specifying the thickness of the walls. Ellipsoidal cylinders and cones can be created by giving major and minor axes or a top radius of zero. Regular prisms can be created by specifying the number of sides.
Sphere	The SPHERE Command [page 313]	Give the centre of the sphere and the radius.
Torus	The TORUS Command [page 329]	Give the centre, the major and minor radii.
Regular n-sided prisms and pyramids	The PRISM Command [page 287]	Creates a regular $n$ -sided polygon with corner points on an ellipse (of given major and minor radius), and extends this in the third dimension a given height with a taper given by a major radius at the top.
Wire edge	The WIREEDGE Command [page 352]	Creates a wire edge which can be straight or circular. Circular edges can be specified using 3 points on the edge or 2 points and the centre of curvature. N.B. The material name and radius properties of a wire edge must be set before it becomes a wire.

These objects are created in the current Working Coordinate System. Local Coordinate Systems can be defined, and one of them selected to be the Working Coordinate System. All primitives and transformations are created within the Working Coordinate System.

A standard set of conductors can also be created. This is done through separate commands, e.g. the **RACETRACK**, **BEDSTEAD** commands. These conductors do not form a physical part of the model and are not included within the finite element mesh. They can be selected for transformation, repositioning and copying. For more details see the **CONDUCTOR** section and the individual conductor commands.

## Sketching objects

An alternative means of creating any of the standard shape primitives is to use the 2D sketching



facilities. With sketching selected through use of the **Sketching Active** toolbutton (), subsequent selection of any of the basic shape primitives through use of their respective toolbar buttons will display a large grid over the xy-plane. The spacing of the grid-lines for this grid may be modified using [The SKETCH Command \[page 304\]](#).

A red spot, confined to grid line intersections, will then be seen to track the mouse position as it moves over the grid. Double-clicking over this spot will initiate the coordinate specifications for the chosen geometry primitive. Subsequent mouse movement will then outline the geometry that will be created by double-clicking of the mouse to specify further required coordinates.

The number of points that will need to be specified will depend on the geometry primitive that was initially selected, but upon specification of the last, the usual dialog for the geometry primitive will appear, allowing the user to confirm or modify the coordinates defined by the sketching.

## Modifying objects

Once created, objects within the model can be picked for modification. There are 2 ways to pick objects:

- by graphical interaction, or
- by picking objects that have certain attributes associated with them.

Once an object is picked, operations can be performed upon that object. In many cases it is possible to pick more than one object and to operate upon all of the picked objects.

The operation may perform differently depending upon the type of object picked, or may have no effect if the picked objects are not of suitable type.

For example: the copy operation (see [The TRANSFORM Command \[page 330\]](#)) can be applied to all picked bodies to create transformed copies of these bodies. If a cells have been picked the process creates a new body containing the single cell for each cell that has been copied. The new bodies are created to ensure that the body's topology is not corrupted; this might happen if the cell were copied into a space already occupied by another part of the body. A single face will behave in the same way

as the cell, but creates a sheet face. Similarly a copied edge will become a wire-edge. Vertices are not operated upon.

The topological entities that are affected by each command are listed in detail later in this manual

## Graphical interaction

Graphical interaction is controlled by [The FILTER Command \[page 219\]](#). This determines what type of topological object can be picked by positioning the cursor over it and double-clicking with the left mouse button. The options for the filter command include: vertex, edge, face, cell, body.

As well as the topological entities described, conductors and Local Coordinate Systems can be picked and modified.

When double clicking, the system generates a command to pick an entity using its unique name. Such commands can be safely used in command scripts unless changes in model dimensions lead to changes in topology.

## Picking by attribute

Each entity can have attribute data associated with it. These data will vary according to the entity type. For example, bodies can have a name associated with them, and a set of labels that can be applied. Cells can have a material label, a volume label, a potential type, an element type, a mesh control size and a set of labels. Faces can have a boundary condition label, a mesh control size, layering information and a set of labels. Edges and vertices can only have a mesh control size and a set of labels.

[The PICK Command \[page 277\]](#) can use these attributes and labels to identify parts of a model for modification, viewing, etc.

## System variables

The following system variables hold the numbers of picked entities: **PICKEDBODIES**, **PICKEDCELLS**, **PICKEDFACES**, **PICKEDEDGES**, **PICKEDVERTICES** and **PICKEDCONDUCTORS**.

## Modifying objects geometry

The main command for modifying the geometry is [The TRANSFORM Command \[page 330\]](#). This command can be applied to bodies, cells etc. A limited form can also be applied to conductors and Local Coordinate Systems. All transformations occur within the Working Coordinate System. Transformations available include:

Transformation	Description
DISPLACE	Re-position by displacement
ROTATION	Rotation about an axis vector (through the origin) by an angle, or rotation through a set of Euler angles
REFLECTION (body only)	Reflect in a plane whose normal is specified (through the origin)
SCALE (body only)	Scale the body by anisotropic scale factors

More complex geometric changes can be performed using the morphing commands on bodies. These commands change the underlying geometric shape of the object. Morphing commands available include:

Command	Description
The BEND Command [page 143]	Bend a body through a fixed angle
The MORPH Command [page 265]	Apply any functional transformation
The STRETCH Command [page 317]	Stretch part of a body between 2 points
The TWIST Command [page 334]	Twist part of a body between 2 points

## Creating complex bodies

More complex structures can be formed by performing boolean operations upon 2 or more bodies using [The COMBINE Command \[page 184\]](#). The boolean operations are:

Operation	Description
UNION	Merge the bodies so that the resulting body has all the components that existed in <b>any</b> of the original bodies.
INTERSECTION	Merge the bodies so that the resulting body has only the components that existed in <b>all</b> of the original bodies.
SUBTRACTION	Changes the first picked body by removing from it the intersection with each of the other picked bodies in turn. The other picked bodies are then deleted.
TRIM	As subtraction, but the other picked bodies are not deleted.
CUTAWAY	Leaves the first body unchanged and changes each of the other picked bodies by removing its intersection with the first body.

The order of picking bodies is important:

- **UNION** and **INTERSECTION**: the order is unimportant for the resulting geometry, but the name of the resulting body will be taken from the first one picked.
- **SUBTRACTION** and **TRIM**: the second body picked is subtracted from the first body picked. If more than two bodies have been picked, the third body picked is subtracted from the result of the first subtraction, and so on.
- **CUTAWAY**: the first body is subtracted from each of the others in turn.

For all operations, there is an option to regularize the result. Regularizing removes internal faces and edges that are formed from the operation. This is most easily seen during union, but can also be useful for subtraction.

Note: If a body is fully contained within another body with no adjoining or overlapping faces, then it will remain unmodified after a union operation without regularization regardless of the data storage level.

## Extending geometry

An existing planar face within the model may be picked and swept through space to extend the volume occupied by a body. When using [The SWEEP Command \[page 318\]](#) there are 4 options available:

Sweep Operation	Description
DISTANCE	Sweep the face by a fixed distance normal to the surface
VECTOR	Sweep the face through a specified vector
ROTATION	Sweep the face around an axis, passing through a point, by a given angle
PATH	Sweep the face along a path formed from edges of the model

An option exists to keep the original face, in which case the sweep operation will form a new cell within the body.

The operation can also apply a taper so that the face expands or contracts at a constant angular rate.

When sweeping, care should be taken to avoid self intersections and formation of incorrect topologies. Where possible the sweep operation will try to fix these inconsistencies, but it cannot be guaranteed to work in all cases.

## Blending and chamfering

The edges between adjoining faces can be blended to form a smooth interface between the faces, or chamfered to trim the edge. Both of these operations are achieved using [The BLEND Command \[page 147\]](#).

Note that such blending and chamfering may only be applied to non-manifold geometry.

## Offsetting and shelling

A body can be grown or shrunk by offsetting one or more faces by a positive or negative distance. A solid body can be hollowed out to become a shell with a given thickness. For more details see [The OFFSET Command \[page 273\]](#) and [The SHELL Command \[page 303\]](#).

## Removing geometry features

Sections of the model can be deleted. Any picked body, conductor or Local Coordinate System can be deleted from the model.

Other topological entities can also be deleted in certain cases.

- A vertex can be removed if the edges that link to it all have the same underlying geometric curve, e.g. represent adjoining parts of an arc of a circle.
- An edge can be removed if all faces touched by that edge have the same underlying surface geometry.
- Internal or external sheet faces can be deleted from a body. A sheet face is one that exists but supports no volume e.g. the product of copying a single face from a body creates such a face. An option to regularize allows any edges that are not needed to maintain the topology of the body to be removed from the body.
- External faces can be removed if the switch to allow this is activated. The removal will only occur if the resulting hole formed in the body can be patched. This cannot occur with faces that form part of a non-manifold geometry.
- Deleting a cell will remove all external faces that exist to support that cell. Any internal faces will now be external. An option to regularize allows removal of any edges and vertices that are no longer needed to maintain the topology of the body.

For more details, see [The DELETE Command \[page 200\]](#).

## Checking and correcting topology

In some instances, particularly after some more complex operations, or with data from other applications, the geometry and topology of the structure will be incorrect. [The CHECK Command \[page 171\]](#) exists to test for potential problems. Any entity that fails the check command will be flagged with a **SYSTEM** label, and the body to which it belongs will also be flagged. These bodies can be displayed and a warning message will give an indication of why the problem occurred. The **CHECK** command contains additional facilities which can in some circumstances repair some of these problems. In some instances it may be necessary to remove and recreate any entities that exhibit problems, as these may cause future operations to fail.

## Modeller Data

Within the Modeller it is necessary to set material properties, mesh control information, volume properties and boundary conditions.

This information is used when creating the data for analysis by one of the analysis modules. It is also used to assist in visualization of the model, as different data can be chosen as the basis for the display.

This data is stored with each entity with which it is associated, and is different for different types of topological entity.

## Body data

A body can have:

- a name: this is used for specifically naming bodies.
- a unique name: in general this is automatically generated and is based on the name of the body. When a body is copied, a new unique name is generated for the copy unless the user specifies a **COPYIDENTIFIER** (see [The TRANSFORM Command \[page 330\]](#)).
- additional labels: these are used for grouping sections of the model for visualization and modification.

When merging bodies the data from the body that is picked first is always maintained.

## Cell data

A cell can have the following data attached to it:

- a material label  
This is a label that is used to reference the material properties associated with the cell. By default cells are labelled **AIR**, but other labels can be applied, and the properties e.g. permeability or BH curve label, conductivity, permittivity, can be defined for each material label.
- Potential type (**AUTOMATIC, TOTAL, REDUCED, VECTOR**)
- Element type (**LINEAR, QUADRATIC**)
- Volume properties label  
A volume property label can be attached to cells and stores other data associated with the cell. This can include velocity, local orientation, packing factor etc.
- Mesh control size, surface normal and normal distance tolerance  
Used to control the maximum size of mesh that is generated within this cell.
- Group labels for stator, gap and rotor in [Motional Electromagnetic Solver \[page 543\]](#).
- Additional labels  
Used for labelling and grouping sections of the model for visualization and modification. In particular, labels are used for
  - emitter data in SCALA analysis (see [The EMITTER Command \[page 206\]](#));
  - for selection in the Post-Processor (see [The SELECT Command \[page 793\]](#)).
- A data storage level

When merging bodies with boolean operations, the resulting cells will normally be formed from a combination of the initial cells. In such a case it is not clear which of the initial data sets should be kept with each newly formed cell. The data storage level is used to resolve this conflict, and the data set with the highest storage level is kept. Where the data level is the same, the result is ambiguous and the data is formed by merging the possible data sets.

## Face data

A face can have the following data attached to it:

- Boundary condition label

This is a label that is used to reference the boundary condition data for this face. In many cases it is more convenient to specify boundary conditions using [The BACKGROUND Command \[page 135\]](#). Boundary condition labels are needed in order for space charge emitter data to be assigned (see [The EMITTER Command \[page 206\]](#)).

- Element type ([LINEAR](#), [QUADRATIC](#))

- Mesh control size, surface normal and normal distance tolerance

Used to control the maximum size of element in the finite element mesh that is generated on this face.

- Additional labels

Used for grouping sections of the model for visualization and modification. Labels are available for selection in the Post-Processor (see [The SELECT Command \[page 793\]](#)).

- Layering

Used to create thin layers on either or both sides of the surface with consistent surface meshing for models where thin geometries are important.

- A data storage level

As with cells, merging may produce ambiguity in face properties. The data set with the greater storage level will be kept.

## Edge and vertex data

An edge or a vertex can have the following data attached to it:

- Mesh control size

Used to control the maximum size of elements in the finite element mesh that is generated on this face.

- Additional labels

Used for grouping sections of the model for visualization and modification.

- A data storage level

As with cells, merging may produce ambiguity in edge properties. The data set with the greater storage level will be kept.

## Setting data properties

The different commands for setting properties are:

Command	Description
The LABEL Command [page 238]	Adds and removes labels for all picked objects
The RENAME Command [page 293]	Gives names to and renames bodies and Local Coordinate Systems
The CELldata Command [page 168]	Sets the properties of picked cells
The FACEDATA Command [page 213]	Sets the properties of picked faces
The EDGEDATA Command [page 205]	Sets properties of picked edges
The VERTEXDATA Command [page 341]	Sets properties of picked vertices
The MATERIALS Command [page 462]	Defines the data associated with a material label
The VOLUME Command [page 342]	Defines the data for each volume label
The BOUNDARY Command [page 150]	Defines the boundary condition data for each boundary label
The BHDATA Command [page 144]	Defines the BH curve file to be associated with a BH curve label (used when defining the material properties)
The EMITTER Command [page 206]	Defines emission characteristics on a face or volume

Properties of one or more entities of the same type can be set with one command. Any data the entities have in common will be shown in the dialog. Common data can be changed and the data set. If a data item is left blank, the original data is left unchanged.

## Listing data

The LIST Command [page 241] can be used to list data associated with the picked entity. The FILTER Command [page 219] can also be used to adjust the effect of double clicking over a part of the model, so that the LIST command is automatically called instead of picking the item for modification.

## Display

The display uses an OpenGL interface to allow visualization of the model. The model can be rotated, translated or zoomed using the mouse controls, toolbuttons, and [The THREED Command \[page 324\]](#).

### View selection

The display of the model is controlled by [The SELECT Command \[page 299\]](#). This allows individual items to be selected for display or to be explicitly hidden from view. There are many data items attached to objects, and these can be used to control the display of the model. The **SELECT** command allows the user to display or hide any object that has data of a certain type, e.g. any face with any boundary label, or more explicitly e.g. any face with a particular boundary label.

### Contour display

Some data attributes can be contoured on surfaces of the model. The contours will only be shown on parts of the model that are visible through the normal display selection. The **CONTOUR** command gives greater detail.

### Vector display

Some data attributes can be displayed as vectors within the model. The vectors will only be shown on parts of the model that are visible through the normal display selection. The **VECTOR** command gives greater detail.

## Command History

The Modeller maintains a history of model changes. Most commands will generate an entry within the history stream. [The UNDO Command \[page 335\]](#) is available to return to any previously noted state of the Modeller. After an **UNDO** command, [The REDO Command \[page 292\]](#) can undo the changes of the **UNDO** command, until a new command has been issued. At this point the **REDO** command becomes invalid, and the history between the current position and the position at which the **UNDO** command was issued is removed.

The [REPLAY Command \[page 294\]](#) allows the history stream to be rewound to an earlier command. The user input of this command can be edited and the command replayed. All further downstream operations are then actioned to rebuild the adjusted model.

Certain commands such as **THREED**, **FILTER** and **SAVE** do not affect the model and do not create an entry in the history stream.

Other commands, such as [The SELECT Command \[page 299\]](#), generate changes. However, when the command is issued several times in succession, the effects are merged into a single state within the history stream.

## Background, Symmetry and Boundary Conditions

In most cases, geometries for electromagnetic field analysis need:

- extending to provide surrounding air space which must be terminated with appropriate boundary conditions; and
- cutting so that symmetrical geometries are reduced to the minimum symmetrical part, with appropriate boundary conditions on the cut planes.

Both of these operations can be done using [The BACKGROUND Command \[page 135\]](#) removing the need for boundary condition labels.

If the model symmetry is specified using [The BACKGROUND Command \[page 135\]](#), the symmetry information is stored in the database and can be used by the Post-Processor to reconstruct the complete model.

There are 2 other ways in which the surrounding air space, boundary conditions and symmetry can be prepared.

- A body with the name **BACKGROUND** can be defined. When the model is being prepared for mesh generation (see [The MODEL Command \[page 263\]](#)) the program will retain only the intersection of the model with the body named **BACKGROUND**. Any boundary conditions on the **BACKGROUND** will be applied to the model if the data storage level is high enough.
- The whole model can be built from separate bodies with their own boundary conditions. This method is much more laborious but does allow greater control over the model and mesh sizes.

## Model Storage

The data within the Modeller can be saved at any time using [The SAVE Command \[page 297\]](#). Once saved a file can be opened using [The LOAD Command \[page 243\]](#). This will clear any existing data from Modeller and will overwrite it with data from the new file. The file can then be saved after modifications. A complete model, including the finite element mesh can be saved in a binary file.

If no file has been loaded, a name must be given the first time the model is saved.

Data from a file can also be inserted into a currently open model. This will add the topology and data from the file into the open model.

Individual component bodies can be picked and exported to data file.

The standard data formats are the **opc** file containing the Opera-3d Modeller components and the **opcb** file which includes the finite element mesh data. ACIS **sat** data files containing geometry and data from other applications may be used within the Opera-3d Modeller. It should be noted that in some cases, such data are not suitable for use with finite element analysis and may cause problems.

## Building a Model for Mesh Generation

To generate a continuous mesh through the volume of space modelled it is necessary to have a single body. This body is made up of multiple cells containing different material data representing the different components of the structure being modelled.

With such a potentially complex single body, it is difficult to modify the geometry and topology of the structure. To make modelling easier, it is preferable to have many simple bodies, possibly overlapping if necessary, that can be easily moved, scaled etc. as required.

To make the transition from multiple bodies to a single body suitable for meshing, there is [The MODEL Command \[page 263\]](#). This command makes a copy of each body (or component) within the model and merges them using a boolean union operation without regularization to ensure that internal material boundaries are maintained. The single body formed (the model body) will have the properties defined on the components, adjusted where necessary to take account of conflicts by using the data storage level. The model created should be suitable for meshing and database creation. If the model is not correct, it is possible to adjust the properties and topology, although these changes will be lost when returning to component view.

Additionally layering information attached to some faces can be used to create thin layers. These layers are forced to have the same topology as the master face, so that the mesh can be transferred directly between them, improving the quality of the tetrahedral mesh in these regions.

Once created, if changes are again required, the model body can be deleted, and the view will return to the component view. If adjustments have been made to the model, these would be lost when returning to component view.

### Mesh generation

Mesh generation is a two stage process. [The MESH Command \[page 257\]](#) generates surface mesh of triangles or quadrilaterals over all faces of the model. The element size is controlled by the mesh size, the surface shape, and the normal distance tolerances. The smallest of each of these is used to control the local mesh size. This command can only be used when the model has been created. The mesh can be displayed as outlines on the model display.

[The FILL Command \[page 216\]](#) creates the volume elements throughout the model space, and can only be used once a valid surface mesh exists.

Changes to the model topology will destroy the surface mesh, as will returning to component view. It is possible to [UNDO](#) back to the last meshes formed, but meshes prior to this will have been overwritten.

### Database generation

General settings for analysis control are entered using the [ANALYSISDATA](#) command. The options used are dependent on the analysis type.

Control of the drive data for transient solvers is set using the **DRIVE** command, along with the phase angle for the harmonic solvers.

Output times for transient solvers, multiple frequencies for harmonic solvers and scaling factors for some of the static solvers are set using the **DBCASEDATA** command.

When the model is ready with a volume mesh created, the database can be prepared using the **SOLVERS** command.

## Opera-3d Modeller Tabbed Menu Interface

The Opera-3d Modeller Tabbed Menu Interface has 3 Tabs of commands:

- **Work:** contains commands for opening and saving files, history, user defined variables and display.
- **Modelling:** contains commands for creating, modifying and manipulating models.
- **Analysis:** contains to define analysis specific data, generate the mesh and create analysis databases.

**Display buttons** are displayed at the bottom of the window and include commands to manipulate the display and perform other common operations.

### Toolbar summary

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
3d display options	Work	View			THREED
3d display options	Display buttons				THREED
Add additional labels	Modelling	Properties			LABEL
Add command files as toolbuttons	Work	Command Files	User Defined Toolbuttons		\$TOOLBUTTON
Add simulation to analysis database	Analysis	Database			SOLVERS
Alter body by sweeping the picked face	Modelling	Operation			SWEEP
Apply filter to picked entities	Modelling	Manipulate			PICK
Blend or chamfer the selected edges	Modelling	Operation			BLEND
Change display colours	Display buttons				COLOUR
Change picking on entities to new filter type	Modelling	Manipulate			PICK

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Change project folder	Work	File			\$PROJECTFOLDER
Change the function of the mouse buttons	Work	Options			MOUSE
Check picked bodies for topological problems	Modelling	Operation			CHECK
Circuit loops	Analysis	Excitation	Circuit		CIRCUIT
Clear all data	Work	File			CLEAR
Combine Bodies	Modelling	Operation	Combine Bodies		COMBINE
Coordinate Systems	Modelling	Operation	Coordinate Systems		WCS
Copy and transform selected bodies	Modelling	Operation			COPY, TRANSFORM
Create a block	Modelling	Create			BLOCK
Create a cylinder or cone	Modelling	Create			CYLINDER
Create a local coordinate system (LCS)	Modelling	Create			LCS
Create a new 8-node brick conductor section	Modelling	Conductors	New		BRICK8
Create a new 20-node brick conductor section	Modelling	Conductors	New		BRICK20
Create a new arc conductor	Modelling	Conductors	New		ARC
Create a new bedstead conductor	Modelling	Conductors	New		BEDSTEAD
Create a new fitted constant perimeter end conductor	Modelling	Conductors	New		FITTEDCPE
Create a new helical end conductor	Modelling	Conductors	New		HELICALEND

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Create a new racetrack conductor	Modelling	Conductors	New		RACETRACK
Create a new solenoid conductor	Modelling	Conductors	New		SOLENOID
Create a new straight bar conductor	Modelling	Conductors	New		STRAIGHT
Create a new tangential constant perimeter end conductor	Modelling	Conductors	New		TANGEMNTIALCPE
Create a sheet face by covering the picked edges	Modelling	Operation			COVER
Create a sphere	Modelling	Create			SPHERE
Create a torus	Modelling	Create			TORUS
Create a wire-edge	Modelling	Create			WIREEDGE
Create analysis database	Analysis	Database			MODEL, MESH, FILL, SOLVERS
Create and view animations	Work	Images			ANIMATION
Create model body	Analysis	Database			MODEL
Cutaway overlap, with regularization	Modelling	Operation	Combine Bodies		COMBINE
Cutaway overlap, without regularization	Modelling	Operation	Combine Bodies		COMBINE
Cycle axis display mode	Display buttons				WINDOW
Default display selection	Work	View			SELECT
Default display selection	Display buttons				SELECT
Define and list tabular functions	Work	User Defined			\$FUNCTION
Define and list user strings	Work	User Defined			\$STRING

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Define and list user variables	Work	User Defined			VARIABLE
Define which bodies are included in the model body	Work	Options			MODEL
Delete model body	Analysis	Database			MODEL
Delete picked entities	Modelling	Operation			DELETE
Display contours of volume properties and boundary conditions in the model	Work	View			CONTOUR
Display vectors of boundary conditions and face and volume properties on the model	Work	View			VECTOR
Edit sketch mode settings	Modelling	Create			SKETCH
Export conductors to file	Modelling	Conductors			EXPORT
Export picked bodies to a file	Work	File			SAVE
Extract cells to form new bodies	Modelling	Operation			EXTRACTCELLS
Generate surface mesh	Analysis	Database			MESH
Generate volume mesh	Analysis	Database			FILL
Grow or shrink a body by offsetting picked faces	Modelling	Operation			OFFSET
Hide currently picked entities	Modelling	Manipulate			SELECT
Hide entity on double-click	Modelling	Manipulate	Action		FILTER
Hollow out a body to create a shell	Modelling	Operation			SHELL

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Import conductors from file	Modelling	Conductors			IMPORT
Insert Opera-2d file	Modelling	Create			INSERTOP2FILE
Insert Opera-2d file	Work	File			INSERTOP2FILE
Insert Opera-3d or CAD file	Modelling	Create			LOAD
Insert Opera-3d or CAD file	Work	File			LOAD
Intersection, with regularization	Modelling	Operation	Combine Bodies		COMBINE
Launch the analysis setup tool to configure solutions for this model	Analysis	Properties			
Launch the Post-Processor	Analysis	Launch			\$OS OPERAPOST
List entity properties on double-click	Modelling	Manipulate	Action		FILTER
List properties of picked entities	Modelling	Properties			LIST
Loft between picked faces	Modelling	Operation			LOFT
Modify common parameters for picked conductors	Modelling	Conductors	Modify		CONDUCTOR
Modify picked 8-node brick conductors	Modelling	Conductors	Modify		BRICK8
Modify picked 20-node brick conductors	Modelling	Conductors	Modify		BRICK20
Modify picked arc conductors	Modelling	Conductors	Modify		ARC
Modify picked bedstead conductors	Modelling	Conductors	Modify		BEDSTEAD
Modify picked fitted constant perimeter end conductors	Modelling	Conductors	Modify		FITTEDCPE

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Modify picked helical end conductors	Modelling	Conductors	Modify		HELICALEND
Modify picked racetrack conductors	Modelling	Conductors	Modify		RACETRACK
Modify picked solenoid conductors	Modelling	Conductors	Modify		SOLENOID
Modify picked straight bar conductors	Modelling	Conductors	Modify		STRAIGHT
Modify picked tangential constant perimeter end conductors	Modelling	Conductors	Modify		TANGENTIALCPE
Morph	Modelling	Operation			MORPH
Move WCS to picked entities	Modelling	Operation	Coordinate Systems		WCS
Open a file	Work	File	Open		LOAD
Open Reference Manual	Work	File			
Open the Circuit Editor	Analysis	Excitation	Circuit		CEDITOR
Open the Command File Editor	Work	Command Files			
Pick all entities of current filter type	Modelling	Manipulate			PICK
Pick entities by property	Modelling	Manipulate			PICK
Pick entity on double-click	Modelling	Manipulate	Action		FILTER
Print	Work	Images	Screen Output		PRINT
Rebuild model by updating previous commands	Work	History			REPLAY
Redo	Display buttons				REDO

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Redo	Work	History			REDO
Refresh the view	Display buttons				THREED
Rename picked bodies	Modelling	Properties			RENAME
Rename picked LCS	Modelling	Operation	Coordinate Systems		RENAME
Revert model to a previous state	Work	History			HISTORY
Revert to saved model	Work	File			CLEAR
Run command file	Work	Command Files	Run Comi		\$COMINPUT
Run recent command files	Work	File	Open		LOAD
Run recent command files	Work	Command Files	Run Comi		\$COMINPUT
Run the 3d Machines Environment	Work	Environments			
Run the Transformers Environment	Work	Environments			
Run the Winding Tool	Work	Environments			
Save image to Clipboard	Work	Images	Screen Output		PICTURE
Save image to file	Work	Images	Screen Output		PICTURE
Save model with mesh	Work	File			SAVE
Save model with new filename	Work	File			SAVE
Save picked components to a file	Work	File			SAVE
Save the model	Work	File			SAVE

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Select components for display	Work	View			SELECT
Select components for display	Display buttons				SELECT
Set boundary conditions	Analysis	Properties			BOUNDARY
Set cell properties	Modelling	Properties			CELLDATA
Set drive properties	Analysis	Excitation			DRIVE
Set edge properties	Modelling	Properties			EDGEDATA
Set entity filter to bodies	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to cells	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to conductors	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to edges	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to faces	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to LCS	Modelling	Manipulate	Pick Type		FILTER
Set entity filter to vertices	Modelling	Manipulate	Pick Type		FILTER
Set face properties	Modelling	Properties			FACE DATA
Set geometric modelling tolerances	Work	Options			PRECISION DATA
Set material properties	Analysis	Properties			MATERIALS
Set model symmetry	Analysis	Properties			BACKGROUND
Set nonlinear magnetic material properties (BH)	Analysis	Properties			BH DATA

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Set properties for space charge emitters	Analysis	Excitation			EMITTER
Set the title of the display	Work	View			TILE
Set vertex properties	Modelling	Properties			VERTEXDATA
Set volume properties	Analysis	Properties			VOLUME
Set WCS to named LCS	Modelling	Operation	Coordinate Systems		WCS
Start analysis in Opera batch queue	Analysis	Launch			\$OS OPERAANL
Subtraction, with regularization	Modelling	Operation	Combine Bodies		COMBINE
Subtraction, without regularization	Modelling	Operation	Combine Bodies		COMBINE
Switch off picking	Modelling	Manipulate	Pick Type		FILTER
Toggle mesh display	Display buttons				WINDOW
Toggle outline view of model	Display buttons				WINDOW
Toggle sketch mode	Modelling	Create			SKETCH
Toggle solid view of model	Display buttons				WINDOW
Toggle visibility of vectors	Display buttons				WINDOW
Transform WCS	Modelling	Operation	Coordinate Systems		TRANSFORM
Trim overlap, with regularization	Modelling	Operation	Combine Bodies		COMBINE
Trim overlap, without regularization	Modelling	Operation	Combine Bodies		COMBINE
Undo	Work	History			UNDO

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Undo	Display buttons				UNDO
Unhide hidden entities	Modelling	Manipulate			SELECT
Union, with regularization	Modelling	Operation	Combine Bodies		COMBINE
Union, without regularization	Modelling	Operation	Combine Bodies		COMBINE
Unpick picked entities	Modelling	Manipulate			PICK
Use global coordinate system	Modelling	Operation	Coordinate Systems		WCS
User defined toolbuttons	Work	Command Files	User Defined Toolbuttons		

## Command Summary

---

### Primitive construction commands

Command	Description
BLOCK	Create a new cuboid body.
CYLINDER	Create a new cylinder / cone / prism body.
PRISM	Create an n-sided prism or pyramid body.
SPHERE	Create a new spherical body.
TORUS	Create a new toroid.
WIREEDGE	Create a new wire edge.
SKETCH	Set/display a 2D sketching plane.

### File and general commands

Command	Description
CLEAR	Clear and reset all data from the model.
COMPATIBILITY	Allow superseded features to be used.
END	End this session of the Modeller.
HISTORY	Control the size of the history stream
INSERTOP2FILE	Insert Opera-2d data as sheet faces or swept volumes.
LOAD	Load model data from a file
REDO	Return changes that have been undone.
REPLAY	Replay commands from the history stream with different data if required.
SAVE	Save model data to a file.
UNDO	Return the model to a previous state.

### Display and selection commands

<b>Command</b>	<b>Description</b>
<b>ANIMATION</b>	Record or play back animations.
<b>COLOUR</b>	Set the colour for display of items.
<b>CONTOUR</b>	Display contours of components on displayed entities.
<b>FILTER</b>	Set a filter and command for graphical selection.
<b>GUIOPTIONS</b>	Set window preferences.
<b>HIDE</b>	Hide picked entities from display.
<b>MOUSE</b>	Change the operation of the mouse buttons.
<b>PICK</b>	Pick specific model entities for modification.
<b>PICTURE</b>	Save picture to file or copy to "clipboard".
<b>PRINT</b>	Send picture to a printer.
<b>SELECT</b>	Select the criteria which determine what is included in the display.
<b>THREED</b>	Create a display window for model visualization.
<b>TITLE</b>	Add date, time and title to the display.
<b>VECTOR</b>	Display vectors: material properties, boundary conditions or current directions.
<b>WINDOW</b>	Control display of parts of the 3D view.

## Conductor Commands

<b>Command</b>	<b>Description</b>
<b>ARC</b>	Create or modify an arc conductor.
<b>BEDSTEAD</b>	Create or modify a bedstead conductor.
<b>BRICK8</b>	Create or modify an 8-node brick conductor.
<b>BRICK20</b>	Create or modify a 20-node brick conductor.
<b>CONDUCTOR</b>	Modify common conductor data.
<b>EXPORT</b>	Export conductor data to a file.
<b>FITTEDCPE</b>	Create or modify a fitted constant perimeter end conductor.
<b>HELICALEND</b>	Create or modify a helical end conductor.
<b>IMPORT</b>	Import conductor data from a file.

Command	Description
RACETRACK	Create or modify a racetrack conductor.
SOLENOID	Create or modify a solenoid conductor.
STRAIGHT	Create or modify a straight bar conductor.
TANGENTIALCPE	Create or modify a tangential constant perimeter end conductor.

### Local coordinate system commands

Command	Description
LCS	Create a new Local Coordinate System (LCS).
WCS	Set a Local Coordinate System as the Working Coordinate System (WCS).

### Topology / geometry modification

Command	Description
BACKGROUND	Define a background region with symmetry and boundary conditions.
BEND	Bend a body by through a fixed angle.
BLEND	Add blends or chamfers at edges.
CHECK	Check the topology of parts of the model.
COMBINE	Combine (union, subtract, etc.) two or more bodies.
EXTRACTCELLS	Turn picked cells into single cell bodies.
DELETE	Delete a body or parts of a body.
LOFT	Create a new body by filling the space between two picked faces.
COVER	Create a new sheet face enclosed by a polygon of picked edges.
MORPH	Apply any functional transformation.
OFFSET	Make bodies to grow or shrink by offsetting picked faces.
PRECISIONDATA	Set geometric tolerances.

Command	Description
SHELL	Hollow out a body to leave a shell.
STRETCH	Stretch part of a body between 2 points.
SWEEP	Sweep a face to form a volume.
TRANSFORM	Transform or copy existing bodies, conductors and Local Coordinate Systems.
TWIST	Twist part of a body between 2 points.

## Data commands

Command	Description
BHDATA	Set the BH data for a BHDATA label.
BOUNDARY	Set the boundary conditions for a boundary label.
CELldata	Set the properties associated with cells.
CEDITOR	Start Circuit Editor.
CIRCUIT	View/edit circuits.
EDGEDATA	Set the properties associated with edges.
EMITTER	Define data of space charge emitters.
FACEDATA	Set the properties associated with faces.
LABEL	Attach a label to any item.
LIST	List data associated with any item.
MATERIALS	Set the material properties for a material label.
RENAME	Rename a body or Local Coordinate System.
VERTEXDATA	Set the properties associated with vertices.
VOLUME	Set the properties for a volume label.

## Analysis database preparation commands

Command	Description
ANALYSISDATA	Set analysis options used by the analysis modules.
COMMENT	Enter comments into analysis database.
DBCASEDATA	Set simulation output frequencies and times.

Command	Description
DRIVE	Set drive information for sources.
FILL	Create a volume mesh.
MESH	Create a surface mesh.
MODEL	Create (and delete) a model suitable for analysis, from the components.
MULTIPHYSICS	Set the stages of a multiphysics analysis.
PERIODICITY	Define model periodicity.
SOLVERS	Create a database for analysis.
VARIABLE	Define and store user defined variables.

## Keyboard Shortcuts

---

The Tabbed Menu Interface (TMI) can be used with keyboard shortcuts which have been defined for some of the more commonly used functions. The full set of shortcuts is as follows.

<b>&lt;space&gt;</b>	Cycle through entities which appear on top of each other on the display
<b>&lt;CR&gt;</b>	Pick the high-lighted entity - same as <b>P</b>
<b>B</b>	Change the picking filter to Bodies
<b>C</b>	Change the picking filter to Cells
<b>D</b>	3d Display
<b>E</b>	Change the picking filter to Edges
<b>F</b>	Change the picking filter to Faces
<b>H</b>	Hide picked entities
<b>I</b>	Display high-lighted entity information - same as <b>L</b>
<b>K</b>	Change the picking filter to Conductors
<b>L</b>	Display high-lighted entity information - same as <b>I</b>
<b>P</b>	Pick the high-lighted entity - same as <CR>
<b>S</b>	Select Items to Display
<b>T</b>	Copy and Transform
<b>V</b>	Change the picking filter to Vertices
<b>Ctrl A</b>	Pick all entities of current filter type
<b>Ctrl C</b>	Copy to clipboard
<b>Ctrl P</b>	Print
<b>Ctrl O</b>	Open a file
<b>Ctrl S</b>	Save the model
<b>Ctrl Y</b>	Redo
<b>Ctrl Z</b>	Undo
<b>Ctrl 0</b>	Reset display to initial view
<b>Ctrl 1</b>	View from +X
<b>Ctrl 2</b>	View from -X
<b>Ctrl 3</b>	View from +Y
<b>Ctrl 4</b>	View from -Y
<b>Ctrl 5</b>	View from +Z

<b>Ctrl 6</b>	View from -Z
<b>Del</b>	Delete
<b>F5</b>	Refresh
<b>F6</b>	Default selection

## The ANALYSISDATA Command

---

### Summary

Sets the options for the analysis modules.

### Command line parameters

Command	ANALYSISDATA		
Parameter	Default	Function	
OPTION	<i>none</i>	ACTIVATE	Set menus and dialogs for the specified PROGRAM.
		LOAD	Recover stored values as default parameter values.
		SET	Set new values.
PROGRAM	<i>none</i>	Analysis program type	
		CARMEN	Motional EM
		DEMAG	Magnetization
		ELEKTRASS	Harmonic EM
		ELEKTRATR	Transient EM
		ELEKTRAVL	Velocity EM
		MULTIPHYSICS	Multiple analyses of different types
		QUENCH	Quench
		SCALA	Charged Particle

Command	ANALYSISDATA		
Parameter	Default	Function	
		<b>SOPRANOEV</b>	Modal HF
		<b>SOPRANOSS</b>	Harmonic HF
		<b>STRESSEV</b>	Modal Stress
		<b>STRESSST</b>	Static Stress
		<b>TEMPOST</b>	Static Thermal
		<b>TEMPOTR</b>	Transient Thermal
		<b>TOSCACURR</b>	Current Flow
		<b>TOSCAELEC</b>	Electrostatic
		<b>TOSCAMAGN</b>	Magnetostatic
<b>LINEAR</b>	<b>YES</b>	<b>YES</b>	Linear permeability for analysis
		<b>NO</b>	Nonlinear permeability (Magnetostatic and Dynamic Electromagnetic solvers)
<b>NLITERTYPE</b>	<b>NEWTON</b>	Nonlinear update method	
		<b>NEWTON</b>	Newton-Raphson nonlinear updates
		<b>SIMPLE</b>	Use simple updates
<b>NITERATIONS</b>	21	Maximum number of nonlinear iterations	
<b>TOLERANCE</b>	0.001	Nonlinear iteration convergence tolerance	
<b>RELAXATION</b>	1.0	Charged particle analysis relaxation factor	

Command	ANALYSISDATA		
Parameter	Default	Function	
<b>SAVEITERATIONS</b>	<b>NO</b>	Save solution and tracks <sup>1</sup> :	
		<b>ALL</b>	Save both after every iteration
		<b>NO</b>	Save both at the end of analysis
		<b>NTH</b>	Save both every <i>n</i> th iteration
		<b>SOLUTION</b>	Save solution after every iteration and tracks at the end of analysis
		<b>TRACKS</b>	Save tracks after every iteration and solution at the end of analysis
<b>SAVENTH</b>	<b>1</b>	Value of <i>n</i> for <b>SAVEITERATIONS=NTH</b>	
<b>RHS</b>	<b>ADAPTIVE</b>	RHS coil calculation method	
		<b>SIMPLE</b>	Simple coil integration
		<b>ADAPTIVE</b>	Adaptive coil integration
<b>HXEXT</b>	<b>0</b>	Externally applied H field for Magnetostatic, Charged Particle and Electromagnetic solvers	
<b>HYEXT</b>	<b>0</b>		
<b>HZEXT</b>	<b>0</b>		
<b>DRIVELABEL</b>		Drive label for external fields in Magnetostatic, Harmonic, Motional and Transient Electromagnetic and Magnetization solvers	
<b>SCALEDRAVE</b>	<b>ALL</b>	Apply drive scaling factors to drives with the specified label. Use <b>SCALEDRAVE=ALL</b> to scale all drives.	
<b>UPWINDING</b>	<b>YES</b>	<b>NO</b>	No upwinding for Velocity EM
		<b>YES</b>	Use upwinding
<b>ITPTSTEP</b>	<b>21</b>	Nonlinear iterations per time-step Motional and Transient EM and Magnetization solvers	

<sup>1</sup>**SAVEITERATIONS** and **SAVENTH** are intended for use in Charged Particle analyses but can be used in any nonlinear analysis. The additional solutions saved in other analyses will only contain the potentials and not the fields.

Command	ANALYSISDATA			
Parameter	Default	Function		
<b>UPDATE</b>	<b>SIMPLE</b>	Time update method for transient solvers:		
		<b>SIMPLE</b>	Simple theta method with fixed steps	
		<b>ADAPTIVE</b>	Adaptive step length theta method	
<b>MAXADERR</b>	0.1	Maximum percentage error for adaptive updates		
<b>DELTAT</b>	0.01	Initial or fixed time-step		
<b>RPM</b>	3000	This parameter is no longer used.		
<b>FREQ1</b>	1.0e9	Lower limit on range of frequencies for Modal HF		
	1.0e-3	Lower limit on range of frequencies for Modal Stress		
<b>FREQ2</b>	2.0e9	Upper limit on range of frequencies for Modal HF		
	10000	Upper limit on range of frequencies for Modal Stress		
<b>NEV</b>	1	Number of eigenvalues to be found		
<b>CONVTOL</b>	1.0e-8	Convergence tolerance for linear equation solutions		
<b>POTENTIALCUT</b>	<b>YES</b>	<b>YES</b>	Use automatic potential cuts	
		<b>NO</b>	Do not use automatic potential cuts	
<b>THERMALTEMP</b>		Initial temperature for Transient Thermal. Specify an expression or <b>T</b> to use nodal values already stored in the database.		
<b>LOGLIST</b>		List of variable names to be logged during transient analysis		
<b>USEBSCONDUCTORS</b>		<b>BSCONDUCTORS</b>	Magnetic field calculation used during Quench analysis.	
		<b>NONE</b>		
		<b>ELEKTRATR</b>		
<b>BEAMTOLERANCE</b>	1e-3	Tolerance on Charged Particle beam calculations		
<b>BEAMMAXSTEP</b>		Maximum step length used for Charged Particle beam calculations		
<b>BEAMMAGFIELDS</b>	<b>NO</b>	Include self-magnetic fields from Charged Particle beams		
<b>SECONDARYGEN</b>	1	Number of generations of secondary emission to be included.		
<b>LDACTIVE</b>	No	Include lossy dielectric effects in the analysis.		

Command	ANALYSISDATA	
Parameter	Default	Function
LDITERATIONS	11	No of iterations of current flow solutions for Charged Particle analysis.
LDTOLERANCE	1e-3	Tolerance for convergence of Charged Particle current flow solutions
ROTATION	YES	Include rotation in Motional EM analysis: YES or NO.
DISPLACEMENT	YES	Include linear motion in Motional EM analysis: YES or NO.
GRAVITY	NONE	Include gravitational forces in stress analysis: MINUS_X Gravity in the negative X direction. MINUS_Y Gravity in the negative Y direction. MINUS_Z Gravity in the negative Z direction. NONE No gravity PLUS_X Gravity in the positive X direction. PLUS_Y Gravity in the positive Y direction. PLUS_Z Gravity in the positive Z direction.
USETHERMALEXPANSION	NO	Include thermal expansion in Static Stress analysis: YES or NO.
STRESSREFERENCETEMP		Expression for reference temperature for thermal expansion in Static Stress analysis.
STRESSOPERATINGTEMP		Expression for operating temperature for thermal expansion in Static Stress analysis.
USEGENERALEXPANSION	NO	Include general expansion in Static Stress analysis: YES or NO.
USEDDEFORMEDMESH	NO	Use deformed mesh: YES or NO.
STAGE	1	Stage number in a multiphysics analysis.

## Notes

This command sets the options for the different analysis program. Each program has its own independent set of data which is configured by this command.

- **OPTION=ACTIVATE PROGRAM=***program\_name* will adjust the menus for specified analysis type.
- **OPTION=LOAD** will load the stored data into the default values for the parameters.
- **OPTION=SET** will change the stored values to those set by the user.

The last program type specified for **OPTION=ACTIVATE** or **OPTION=SET** will be the active analysis type. Property menus, e.g. material properties, boundary conditions, model symmetry, etc. will be configured for the active analysis type. An analysis database created from the geometric model will be solved by the active analysis type.

The **ANALYSISDATA** command sets various analysis options. From the menus, only those options relevant to the current simulation are available. See [Analysis Programs \[page 539\]](#) for more information.

- **Linear or Nonlinear:** Linear analyses use constant material properties; nonlinear analyses update the material properties, depending on the solution and re-solve. Nonlinear analyses can include some linear materials and linear analyses can include nonlinear materials. In this latter case, the material property characteristic (e.g. BH curve) is used to supply the value of the material property (permeability). The slope of the curve at zero flux-density is used.

Nonlinear analyses use either Newton-Raphson or simple iterations as specified by **NLITERTYPE**. The iterations continue until convergence or the maximum number of iterations (**NITERATIONS**) has been reached. Two convergence tests are used:

- the maximum and rms changes in the solution must be less than **TOLERANCE**;
- the residual (see [Nonlinear materials \[page 611\]](#)) must be less than **TOLERANCE** squared and divided by 100 (e.g. **TOLERANCE=0 . 001** gives a residual tolerance of  $0.001^2/100=10^{-8}$ ).

The iterations converge if either test is satisfied.

- **RHS Integrals:** The line and surface integrals of coil fields which are part of magnetostatics and electromagnetic analyses can be done using simple integrals, which only use the values of coil fields at the nodes or using adaptive integrals which use additional field points in order to reduce the errors.
- **External Fields** can be added to Magnetostatics, Charged Particle and Electromagnetic analyses. For Harmonic and Transient EM, a **DRIVELABEL** is needed to link to the drive function to be associated with the external fields. See footnotes in [Scalar Potential Formulation \[page 602\]](#) and [Vector Potential Formulation \[page 605\]](#) for more information on the use of external fields.
- **Drive Scaling** factors which are set using [The DBCASEDATA Command \[page 197\]](#) can be applied to static analyses. The factors can be applied to **ALL** drives or to an individual drive chosen by its label using parameter **SCALEDRAVE**.
- **Automatic Potential Cuts** can be used in Magnetostatic and Electromagnetic analyses to automatically insert potential cuts to avoid having multiply connected volumes, where a loop of total potential volume encloses a non-zero net current.
- **Charged Particle Solver Iterations** converge to a consistent set of particle trajectories and fields. The number of iterations and under-relaxation factor can be set. The fields and particle trajectories are saved at the end of the analysis. Additional outputs can be selected using the parameters **SAVESOLUTIONS** and **SAVENTH**.

Additional control of beam tolerance and step size is available in **BEAMTOLERANCE** and **BEAMMAXSTEP**.

- **Charged Particle self magnetic fields** from the beam currents can be included using **BEAMMAGFIELDS**.
- **Upwinding** is a technique to improve Velocity EM analysis. The solver reports whether upwinding is required or not.
- **Transient Time-stepping** can use **SIMPLE** fixed time-steps or **ADAPTIVE** time-stepping to achieve a given accuracy (specified as a percentage).
- **Data logging** in transient solvers uses a list of variables created with the **LOGLIST** parameter. Multiple instances of the **LOGLIST** parameter can be used in one command:

**LOGLIST=** empties the list.

**LOGLIST+=name** adds **name** to the list.

**name** can be any system or user variable which is updated by the time-stepping process. For more information, see [Transient Analysis Logging and Control \[page 627\]](#).

- **Motion** in Motional EM can be **ROTATION**, **DISPLACEMENT** or both. If both rotation and displacement are selected, displacement is limited to the z-direction.
- **Initial temperature** in Transient Thermal and Quench solvers can be specified as a constant value or as **T** to pick up nodal values of temperature already stored in the database from an earlier analysis or added as nodal values of **RT** with the Post-Processor using [The TABLE Command \[page 817\]](#).
- For a **Quench analysis**, the calculation of the magnetic fields is determined using the **USEBSCONDUCTORS** parameter.
  - **USEBSCONDUCTORS=NONE**  
No magnetic fields are included in the quench calculation.
  - **USEBSCONDUCTORS=BSCONDUCTORS**  
The magnetic field values are calculated from the defined conductor primitives. This option is suitable for coil only models. These conductors will normally be circuit element conductors. Fields are calculated using the Biot-Savart integral, but with a current density that is calculated from the current in the circuit element.
  - **USEBSCONDUCTORS=ELEKTRATR**  
The magnetic fields can be calculated from a coupled solution with Transient EM. This should be used where the magnetic field will be affected by ferromagnetic materials or induced eddy currents.
- **Lossy dielectric** solutions are available for Current Flow, Electrostatic and Charged Particle solutions. If active (**LDACTIVE=YES**), regions with conductivity will form part of a current flow solution to determine the electric potential distribution, before calculating the electrostatic field distribution in non-conducting dielectrics. Where conducting regions have no assigned voltage boundary, the potential is floating and will be determined from the electrostatic solution.  
For Charged Particle solutions, the current from beams hitting conducting regions will form part of a nonlinear solution of the current flow, with the currents from the beams being updated for each iteration.

- **Modal HF** and **Modal Stress** eigen-frequencies are calculated in the range **FREQ1** to **FREQ2**, limited to **NEV** frequencies. If **FREQ2** is set to be the same as **FREQ1**, there is no upper limit to the frequency range; the only limit is **NEV**.
- **Static Stress Analysis** has 3 optional features:
  - gravitational forces can be switched on, with the direction of gravity aligned to any of the major coordinate directions, positive or negative.
  - thermal expansion can be switched on, with expressions for the reference and operating temperatures. These can be expressed in terms of values stored in the database from an earlier analysis or added by the Post-Processor using [The TABLE Command \[page 817\]](#). The expansion coefficients are supplied using [The MATERIALS Command \[page 248\]](#).
  - general expansion can be switched on. The expansion integrals are supplied using [The MATERIALS Command \[page 248\]](#).
- **Multiphysics**. In a multiphysics analysis, several analysis stages can be defined before any of them is run. The **STAGE** parameter should be used to identify which stage number for which the parameters are being set.
- **Deformed mesh**. If the database contains an earlier stress analysis simulation, the node coordinates in the new simulation can be updated by the displacements calculated by the Static Stress solver. This option is selected with **USEDEFORMEDMESH=YES**.

Each analysis program uses only a subset of the analysis parameters.

## The **ANIMATION** Command

---

### Summary

Record or play back an animation file.

### Toolbutton



### Command line parameters

Command	<b>ANIMATION</b>		
Parameter	Default	Function	
<b>ACTION</b>	<b>NEW</b>	Action on animation file:	
		<b>NEW</b>	Create or overwrite an animation.
		<b>ADD</b>	Add to or create an animation.
		<b>PLAY</b>	Play back a stored animation.
<b>FILE</b>	<i>none</i>	File name to be recorded or played back.	
<b>DELAY</b>	0	The delay in seconds between frames when playing back an animation.	
<b>SCALE</b>	100	The picture size of the animation (in percent of the original size) when playing back.	
<b>LOOP</b>	<b>NO</b>	<b>NO</b>	Stop at the end of the animation.
		<b>YES</b>	Restart at the beginning.
<b>TIMETOPLAY</b>	0	The time in seconds after which the animation viewer will close (0 keeps the viewer open).	

### Notes

The **ANIMATION** command can be used to record or play back an animation file. An animation file is a series of compressed images stored in the "Portable Network Graphics" format using a **png** extension.

The following **ACTIONs** are used to add the current screen display to an animation file or play back an existing file:

- **NEW**: creates a new file or overwrites an existing file with one image.

- **ADD**: creates a new file with one image or adds a new image to an existing file.
- **PLAY**: launches an animation viewer.

The play-back parameters can be used to adjust the initial behaviour of the animation viewer:

- the **DELAY** between frames,
- the size of the play-back window relative to the original window size (**SCALE** as a percentage),
- whether the animation will play once (**LOOP=NO**) or restart at the beginning when it gets to the end (**LOOP=YES**), and
- how long the animation will play for (**TIMETOPLAY** in seconds).

The animation viewer has toolbuttons to change the way the animation is displayed (it can be stopped, single stepped or played in a continuous loop). The currently displayed frame in the animation viewer can be copied to the system clipboard.

## The ARC Command

---

### Summary

Create or modify arc conductors.

### Toolbutton



### Command line parameters

Command	ARC	
Parameter	Default	Function
OPTION	NEW	NEW
		MODIFY
		LOAD
DRIVELABEL		Name for the arc drive label
LCNAME		Name for Local Coordinate System for coordinate system 1
SYMMETRY		Rotational symmetry about global Z axis
XCEN2		Origin of coordinate system 2
YCEN2		
ZCEN2		
THETA2		Euler angles defining orientation of coordinate system 2
PHI2		
PSI2		
RXY		Reflection symmetries in XY, YZ and ZX planes.
RYZ		
RZX		
WIDTH		Axial cross-sectional size
THICKNESS		Radial cross-sectional size

Command	ARC		
Parameter	Default	Function	
R1		Radius of the arc	
ANGLE		Angle subtended by the arc	
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
TOLERANCE		Field calculation tolerance	
INCIRCUIT		Is the conductor part of a circuit:	
	NO	The conductor has defined current density.	
	YES	The current in the conductor is determined by a circuit.	
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.	
CIRCUITELEMENT		The name of circuit element this conductor is part of	
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation	
MODELCOMPONENT		NO	Do not convert to meshable cells.
	REGULAR	Convert to meshable cells which will be regularly meshed.	
	YES	Convert to meshable cells which meet the mesh size criteria.	
MESHSIZE		The size of the mesh to be used when meshing	
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new arc conductor or modifies existing arcs.

- **OPTION=NEW** creates a new arc conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked arcs. If the picked arcs do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking arc conductors (see [The PICK Command \[page 277\]](#)) to be modified and before **ARC OPTION=MODIFY**.

- **OPTION=MODIFY** changes the conductor data of all of the picked arcs to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Circular Arcs \[page 524\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **BACKGROUND** Command

---

### Summary

Defines a background region with symmetry and boundary conditions.

### Toolbutton



### Command line parameters

Command	BACKGROUND	
Parameter	Default	Function
OPTION	LOAD	LOAD or SET the Background data.
SHAPE	NONE	Shape of Background. Valid options are <b>NONE</b> , <b>BLOCK</b> , <b>CYLINDER</b> , <b>SPHERE</b> , <b>TRIM</b> and <b>REFLECTIONTRIM</b> .
SCALEX	10	Scales in X, Y, Z directions.
SCALEY		
SCALEZ		
SCALER	10	Scale in R direction.
XYSYMMETRYPLANE	NO	Reflections on coordinate planes: YES or NO.
YZSYMMETRYPLANE		
ZXSYMMETRYPLANE		
ROTZNUM	1	The order of rotational symmetry around Z. This must be a positive integer.
EMRXY		Type of electromagnetic boundary condition on the coordinate planes: <b>NORMELEC</b> , <b>NORMMAGN</b> , <b>TANGELEC</b> or <b>TANGMAGN</b> .
EMRYZ		
EMRZX		
EMROZTYPE		Type of electromagnetic symmetry condition on rotations: <b>NEGATIVE</b> , <b>NORMELEC</b> , <b>NORMMAGN</b> , <b>POSITIVE</b> , <b>TANGELEC</b> or <b>TANGMAG</b> .

Command	BACKGROUND	
Parameter	Default	Function
THERMALRXY		Type of thermal boundary condition on the coordinate planes: <b>INSULATOR</b> .
THERMALRYZ		
THERMALRZX		
THERMALROTZTYPE		Type of thermal symmetry condition on rotations: <b>INSULATOR</b> or <b>POSITIVE</b> .
STRESSRXY		Type of stress boundary condition on the coordinate planes: <b>FIXED</b> or <b>NORMFIXED</b> .
STRESSRYZ		
STRESSRZX		
STRESSROTZTYPE		Type of electromagnetic symmetry condition on rotations: <b>FIXED</b> , <b>NORMFIXED</b> , <b>POSITIVE</b> or <b>TANGFIXED</b> .
BCRIGHT	DEFAULT	Boundary condition on far field Right (+X) and Left (-X) surfaces.
BCLEFT		
BCTOP	DEFAULT	Boundary condition on far field Top (+Y) and Bottom (-Y) surfaces.
BCBOTTOM		
BCFRONT	DEFAULT	Boundary condition on far field Front (+Z) and Back (-Z) surfaces.
BCBACK		
BCRADIUS	DEFAULT	Boundary condition on far field Radius (R) surface.
<b>The following parameters are deprecated and should not be used with new models. See <a href="#">Deprecated parameters [page 139]</a>.</b>		
RXY	NONE	Reflections on coordinate planes. Valid options are <b>NONE</b> , <b>TANGMAGN</b> , <b>NORMMAGN</b> , <b>TANGELEC</b> , <b>NORMELEC</b> , and <b>INSULATOR</b>
RYZ		
RZX		
ROTZTYPE	POSITIVE	Type of rotational symmetry around Z. Valid options are <b>NONE</b> , <b>TANGMAGN</b> , <b>NORMMAGN</b> , <b>TANGELEC</b> , <b>NORMELEC</b> , <b>INSULATOR</b> , <b>POSITIVE</b> and <b>NEGATIVE</b> .

## Notes

The **BACKGROUND** command defines the symmetry associated with the model, enforces this symmetry where necessary and creates a background region to assist in modelling external air regions.

The information on model symmetry is used within the analysis and post-processing for calculations requiring the full model.

Some options are specific or to an analysis type, or have restrictions due to the analysis type. The background options are stored for each analysis type. Therefore, changing the analysis type requires that the background options are set up for the new analysis type.

## Background space

Background space is added when creating the Model Body. The limits of the existing model are scaled to determine the size of background space. **SCALEX**, **SCALEY** and **SCALEZ** are used to calculate the size of a **BLOCK**. **SCALEZ** and **SCALER** are used with **CYLINDER**. The size of a **SPHERE** only uses **SCALER**.

A rectangular bounding box is used to determine the size and the centre of the existing model.

- **SHAPE=BLOCK**: the sizes of background block in the three coordinate directions are calculated by corresponding sizes of the bounding box while the block is located at the centre of the box.
- **SHAPE=CYLINDER**: the corners of the bounding box are used to measure the largest radius from the Z axis. The cylinder is centred on the midpoint of Z axis.
- **SHAPE=SPHERE**: the corners of the bounding box are used to measure the largest radius from the coordinate origin. The sphere is located at the origin.
- **SHAPE=TRIM**: does not introduce any background space but reduces the existing model according to the reflection and rotational symmetry settings.
- **SHAPE=REFLECTIONTRIM**: does not introduce any background space but reduces the existing model according to the reflection symmetry settings.

If a rotational symmetry has been specified it is not used to trim the model. Therefore, the user must ensure that only the sector of the full model is defined in the component model, so that appropriate symmetry boundary conditions can be identified to match the given rotation.

This option is useful where the model has symmetry boundaries that are easiest to model using non-planar limits, e.g. a skewed tooth motor model.

## Symmetries

Symmetry options reduce the model by removing parts which are reflected or rotated copies. For example, a model with **ROTZNUM=4** would be trimmed to include only the first quadrant, i.e. between 0 and 360/**ROTZNUM** degrees.

Reflections on XY and ZX coordinate plane can be combined with a rotational symmetry, but rotational symmetry is not allowed with reflections in the YZ plane.

Reflection and rotational symmetry boundary conditions can be specified for different analysis types. In the following description, *analysis* can be **EM**, **THERMAL** or **STRESS**.

Reflection symmetry can be applied on individual coordinate planes using **analysisRXY**, **analysisRYZ** and **analysisRZX**. Symmetries are:

- electromagnetic analysis: tangential or normal field;
- thermal analysis: only **INSULATOR** symmetry is available;
- stress analysis: coordinates can be fixed in all directions or in the normal direction.

Rotational symmetry around global Z axis can be defined by **ROTZNUM** and **analysisROTZTYPE**. The maximum value of **ROTZNUM** is 1023. The available symmetries specified by **analysisROTZTYPE** are:

- electromagnetic analysis:
  - **POSITIVE** or **NEGATIVE**, if there is no ZX reflection (**ZXSYMMETRYPLANE=NO**)<sup>1</sup>;
  - tangential or normal field, if there is a ZX reflection (**ZXSYMMETRYPLANE=YES**);
- thermal analysis: insulator or positive;
- stress analysis: positive or fixed in all directions, the normal direction or tangential directions.

Note the following definitions which apply to potentials, temperatures or displacements:

- **POSITIVE** will set values on paired faces to be the same.
- **NEGATIVE** (electromagnetic potentials only) will set the potentials on paired faces to have the same magnitude but opposite sign. This should be used in models which have currents flowing in opposite directions in alternate rotated copies.

Refer to [The BOUNDARY Command \[page 150\]](#) and [The PERIODICITY Command \[page 274\]](#).

## Far-field boundary conditions

The **BACKGROUND** command creates boundary condition labels, for each far-field surface corresponding to the background **SHAPE** and the symmetry parameters. For example, a cylindrical background will have labels **Farfield Radius** and **Farfield Front (+Z)**; if **XYSYMMETRYPLANE** is set to **NO** there will also be a label **Farfield Back (-Z)**.

The parameters **BCFRONT**, **BCRADIUS**, etc. can be used to set the boundary condition on these labels using parameter values other than **DEFAULT**. Non-default values are only available for electromagnetics: **NORMELEC**, **NORMMAGN**, **TANGELEC**, and **TANGMAGN**.

The boundary condition on these labels can also be set using [The BOUNDARY Command \[page 150\]](#).

## Bodies called **BACKGROUND** and the **BACKGROUND** command

The background space created by users using primitives with name of **BACKGROUND** is still valid with the **BACKGROUND** command.

If a model already has user-defined background, the **BACKGROUND** command will add further background space to the model when creating the Model Body (see [The MODEL Command \[page 263\]](#)).

---

<sup>1</sup>Tangential and normal fields can be used when there is no ZX reflection and will give the correct solution in the section which is analysed. However, it might not be possible for the Post-Processor to determine the sign of the fields in the symmetry copies of the mesh.

## Deprecated parameters

Parameters **RXY**, **RYZ**, **RZX** and **ROTZTYPE** have been superseded by parameters specific to the 3 analyses type: **EM**, **THERMAL** and **STRESS**. Values in parameters **RXY**, **RYZ**, **RZX** and **ROTZTYPE** are no longer stored with the model in data files but if they are used in command files, the values for these parameters will be interpreted and be used in precedence over any data assigned to the new parameters.

## The **BEDSTEAD** Command

---

### Summary

Create or modify bedstead conductors.

### Toolbutton



### Command line parameters

Command	<b>BEDSTEAD</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>NEW</b>	<b>NEW</b>	Creates a new bedstead conductor
		<b>MODIFY</b>	Modifies properties of the picked bedstead conductors
		<b>LOAD</b>	Loads defaults from picked conductors
<b>DRIVELABEL</b>		Name for the bedstead drive label	
<b>LCNAME</b>		Name for Local Coordinate System for coordinate system 1	
<b>SYMMETRY</b>		Rotational symmetry about global Z axis	
<b>XCEN2</b>		Origin of coordinate system 2	
<b>YCEN2</b>			
<b>ZCEN2</b>			
<b>THETA2</b>		Euler angles defining orientation of coordinate system 2	
<b>PHI2</b>			
<b>PSI2</b>			
<b>RXY</b>		Reflection symmetries in XY, YZ and ZX planes	
<b>RYZ</b>			
<b>RZX</b>			
<b>XP1</b>		Local coordinates of lower, inside corner	
<b>YP1</b>			

Command	<b>BEDSTEAD</b>		
Parameter	Default	Function	
<b>WIDTH</b>		Local x cross-sectional width	
<b>THICKNESS</b>		Local y cross-sectional height	
<b>H1</b>		Half-length of the straight	
<b>H2</b>		Half-length of the upright	
<b>R1</b>		Radius of the arc	
<b>R2</b>		Radius of the upper bend	
<b>CURD</b>		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
<b>TOLERANCE</b>		Field calculation tolerance	
<b>INCIRCUIT</b>		Is the conductor part of a circuit:	
		<b>NO</b>	The conductor has defined current density.
		<b>YES</b>	The current in the conductor is determined by a circuit.
<b>REVERSE</b>		Reverse the connections to this conductor in its circuit: <b>YES</b> or <b>NO</b> .	
<b>CIRCUITELEMENT</b>		The name of circuit element this conductor is part of.	
<b>GROUPLABEL</b>		The group of which the conductor is a part for a Motional EM simulation	
<b>MODELCOMPONENT</b>		<b>NO</b>	Do not convert to meshable cells.
		<b>REGULAR</b>	Convert to meshable cells which will be regularly meshed.
		<b>YES</b>	Convert to meshable cells which meet the mesh size criteria.
<b>MESHSIZE</b>		The size of the mesh to be used when meshing	
<b>MESHLFACTOR</b>		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
<b>KEEP</b>	<b>NO</b>	<b>NO</b>	Clear the list of picked items
		<b>YES</b>	Keep the list of picked items for further modification

## Notes

This command creates a new bedstead conductor or modifies existing bedsteads.

- **OPTION=NEW** creates a new bedstead conductor.

- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked bedsteads. If the picked bedsteads do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking solenoids (see [The PICK Command \[page 277\]](#)) to be modified and before **BEDSTEAD OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked bedsteads to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Bedsteads \[page 517\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **BEND** Command

---

### Summary

Bends a body through a specified angle.

### Toolbutton



### Command line parameters

Command	<b>BEND</b>			
Parameter	Default	Function		
<b>LCSNAME</b>		Name of local coordinate system defining the bend orientation		
<b>RADIUS</b>		Radius of the bend		
<b>ANGLE</b>		Angle of the bend		
<b>CENTRE</b>	<b>NO</b>	<b>NO</b>	Centre the bend	
		<b>YES</b>		
<b>KEEP</b>	<b>NO</b>	<b>NO</b>	Keep the body for further operations	
		<b>YES</b>		

### Notes

This command operates on a set of picked bodies and bends them around a cylinder. The local coordinate system given by **LCSNAME** specifies the axes around which the bend occurs. The portion of the body along the positive W axis of the LCS is bent around an axis parallel with the U axis, but shifted by a distance **RADIUS** along the V axis.

The bodies are bent through the **ANGLE** given. If the bodies do not extend far enough in the W direction, the bend may not complete the full angle specified.

If **CENTRE=YES**, the bend angle will be centred on the origin of the LCS. If **CENTRE=NO**, the bend will start at the origin of the LCS.

## The BHDATA Command

---

### Summary

Assigns BH curve properties to a BH data label

### Toolbutton



### Command line parameters

Command	BHDATA		
Parameter	Default	Function	
OPTION		ADD	Adds a point to the end of the BH curve
		DELETE	Deletes a BH curve label
		EDIT	Starts the BH editor window
		INIT	Clears the data from the BH curve
		LIST	Lists the BH data
		LOAD	Loads BH data from file
		NEW	Creates a new BH data label
		SAVE	Saves BH data in a file
		VIEW	Starts the BH viewer window
LABEL		BH data label to be used by the command.	
H		New value of H to be added	
B		New value of B to be added	
CGS	YES	NO	The BH curve will be defined in CGS units
		YES	
FILE		File name for LOAD and SAVE options.	

### Notes

The BHDATA command allows editing of BH data associated with BH curve labels.

For all commands, the label being edited or created should be specified in the LABEL parameter.

- option=new will create a new BH data label.

- `option=edit` will start the interactive BH editor window (note that this editor does not directly operate the **BHDATA** command, but will generate a consistent set of commands upon closure).
- `option=init` will clear the BH data associated with the label. The units being used by the curve should be specified on the command line with this option.
- `option=add` adds a point to the end of the BH data associated with the label.
- `option=delete` deletes a BH curve associated with a label.
- `option=list` will list the data associated with the label.
- `option=load` will load data from a BH data file (`*.bh`).
- `option=save` will save the BH data created using options **NEW** and **ADD** in a file. The file name extension will be **bh**. If the file already exists, it will be overwritten.
- `option=view` will start the interactive BH viewer.

## BH Viewer

The BH viewer is an interactive tool for viewing properties of the BH data associated with a label. The viewer allows views of different graphs generated from the data. These graphs include:

- $BvH$  (data values)
- $BvH$  (interpolated values)
- $MvH$  (interpolated values)
- $\mu vH$  (interpolated values)
- $\frac{\partial \mu}{\partial H} vH$  (interpolated values)

The interpolated graphs are displayed for both positive and negative values of  $H-H_c$ , but will only be available if there are no errors found in the data. The graphs are interactive and the display can be zoomed using

- the left mouse button to drag a box for zooming a local area,
- moving the mouse up or down, with the middle mouse held down, zooms in or out on the centre of the screen.
- Moving the mouse with the right button held down, pans the picture. This can also be achieved using the scroll bars.

Zoom bounding box and previous views can be obtained from the view menu. Additionally a set view menu allows an explicit size to be defined, and allows the aspect ratio to be locked for subsequent zoom operations.

The units in which the data is displayed (CGS or SI) can be changed from the options menu.

The data being displayed can be exported to a new BH data file (**bh**).

## BH Editor

The BH editor incorporates all the facilities of the BH viewer.

In addition it provides a spreadsheet style interface in which the underlying data can be edited.

- Individual values can be changed by editing the entry in the table.
- Data can be imported from a BH data file (**bh**) provided by the user or from the library of BH data files supplied with the software in folder **%VFDIR%\bh** (Windows) or **\$VFDIR/bh** (Linux).
- Changes can be undone or re-applied.
- Data from other sources may be copied in from the Windows clipboard if it is in the correct format (tab delimited data). Data may be exported in similar format.

All changes automatically update the graphs in the viewer.

## Material Types

### Soft magnetic materials

For *soft magnetic materials* the characteristics should be defined in the first quadrant, with the first values of **B** and **H** both zero. The curve should not extend beyond saturation magnetization; the program extra-polates correctly. The BH Viewer displays  $\delta\mathbf{B}/\delta\mathbf{H}$  from the last 2 points on a BH curve and **M** from the last point on the curve to help users to define sufficient, but not too many, data points.

### Hard magnetic materials

For *hard magnetic materials* the operating curve should be defined. The first value of **B** should be less than or equal to zero. The coercive force is the value of **H** for zero **B**. (This should be one of the defined points on the BH curve.) If the characteristic starts in the third quadrant, additional curves should be defined for the orthogonal directions. The easy direction of the magnet is set by the Z direction of the volume local coordinate system (see [The VOLUME Command \[page 342\]](#)).

### Hysteretic materials

For *hysteretic materials*, the demagnetization curve should be defined. It should start in the third quadrant and end in the first quadrant. The extreme values should have the same magnitude, i.e.  $\mathbf{B}_{\min} = -\mathbf{B}_{\max}$  and  $\mathbf{H}_{\min} = -\mathbf{H}_{\max}$ . The defined points should include a point at  $\mathbf{B}=0$ .

The material should be defined to be hysteretic by setting **MULINEARITY=HYSTERETICMODEL** in [The MATERIALS Command \[page 248\]](#).

Hysteretic materials can be used in any of the transient electromagnetic solvers (Transient EM, Motional EM and Magnetization) but a license for Magnetization is required. Hysteresis can also be approximated by complex permeability in Harmonic EM.

## The BLEND Command

---

### Summary

Creates a blend or chamfer between adjoining faces along an edge or a chamfer at a vertex.

### Toolbutton



### Command line parameters

Command	<b>BLEND</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>CHAMFER</b>	<b>BLEND</b>	Create a blend
		<b>CHAMFER</b>	Create a chamfer
<b>RADIUS</b>		Radius of the blend	
<b>LEFTCHAMFER</b>		Distance between chamfer edge and original edge or vertex	
<b>RIGHTCHAMFER</b>		Distance between chamfer edge on second face and original edge	

### Notes

This command operates on a set of picked edges or vertices. The edges must be non-manifold (i.e. there must not be more than two faces meeting at the edge).

If **OPTION=BLEND**, then for each picked edge a tangential join is formed. The join is cylindrical in nature, with radius given by the parameter **RADIUS**.

If **OPTION=CHAMFER**, any picked **edge** is planed off to give a planar face between the two faces, with each of the original faces cut by the **LEFTCHAMFER** or **RIGHTCHAMFER** distances. At present there is no way to distinguish which value is applied to which face, so the operation should **PREVIEW**ed and re-applied with the values switched if they are incorrect at the first attempt.

If **OPTION=CHAMFER**, any picked **vertex** is planed off to give a planar polygonal face joining the faces which previously met at the vertex. The polygonal face cuts the edges at a distance **LEFTCHAMFER** from the vertex.

When applying the command to multiple edges, the result may differ from the result of applying the command to each edge individually. The **BLEND** command may fail if no suitable blend or chamfer surface can be found.

## The **BLOCK** Command

---

### Summary

Creates a cuboid block from the data supplied.

### Toolbutton



### Command line parameters

Command	<b>BLOCK</b>	
Parameter	Default	Function
<b>NAME</b>		Attaches this name to the cuboid body formed
<b>X0</b>		Coordinates of corner of the cuboid
<b>Y0</b>		
<b>Z0</b>		
<b>X1</b>		Coordinates of opposite point in the cuboid
<b>Y1</b>		
<b>Z1</b>		
<b>MATERIALLABEL</b>		Set the material label of the new block
<b>LEVEL</b>		Set the data storage level of the new block
<b>UNIQUENAME</b>		Force a specific unique name for the block created

### Notes

The **BLOCK** command creates a rectangular parallelepiped from the coordinates of opposite corners. A rectangular sheet face is formed if **X0=X1** or **Y0=Y1** or **Z0=Z1**.

The coordinates specified are in the Working Coordinate System.

The single **CELL** created by the **BLOCK** command is given a material label and data storage level if specified in the **MATERIALLABEL** and **LEVEL** parameters. See "The CELldata Command" on page 168. If the dimensions of the block create a sheet face, the properties specified are attached to the **FACE** created. See "The FACEDATA Command" on page 213.

When a block is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

## The **BOUNDARY** Command

### Summary

Sets the properties associated with a boundary label.

### Toolbutton



### Command line parameters 1: general

Command	BOUNDARY		
Parameter	Default	Function	
OPTION		PICK	Adds a boundary label to a list to be set
		UNPICK	Clears the list of picked boundary labels
		RESET	Clears the data from the picked boundary labels
		MODIFY	Sets the data for the picked boundary labels
		DELETE	Deletes the picked labels
BOUNDARYLABEL		Boundary label to be picked	

## Command line parameters 2: electromagnetic conditions

Command	<b>BOUNDARY</b>	
Parameter	Default	Function
<b>CONDITION</b>		Electromagnetic boundary condition:
		<b>CURRENTSOURCE</b> Current source boundary.
		<b>DPDN</b> Assigned derivative of scalar potential.
		<b>DVDN</b> Assigned derivative of voltage.
		<b>ELECTRICINSULATOR</b> Electric insulator.
		<b>EVECTOR</b> Assigned electric field.
		<b>IEVECTOR</b> Assigned incident electric field.
		<b>IVECTOR</b> Assigned incident vector potential.
		<b>MIXED</b> Mixed potential condition.
		<b>NONE</b> None.
		<b>NORMELEC</b> Normal electric.
		<b>NORMELEV</b> Normal electric with assigned constant voltage.
		<b>NORMMAGN</b> Normal magnetic.
		<b>NORMMAGP</b> Normal magnetic with assigned constant potential.
		<b>PEC</b> Perfect conductor.
		<b>POTENTIAL</b> Functional magnetic scalar potential.
		<b>RADIATION</b> Radiation.
		<b>SYMMETRY</b> Symmetry boundary.
		<b>TANGELEC</b> Tangential electric.
		<b>TANGMAGN</b> Tangential magnetic.
		<b>TERMINAL</b> Terminal of a bulk eddy current conductor.
		<b>THINPLATE</b> Thin plate with a specified material and thickness (magnetostatics only).
		<b>VECTOR</b> Assigned vector potential.
		<b>VOLTAGE</b> Functional voltage.
<b>VOLTAGE</b>		Functional electric scalar potential.

Command	<b>BOUNDARY</b>	
Parameter	Default	Function
DVOLTAGE		Derivative of electric scalar potential.
MPOTENTIAL		Functional magnetic scalar potential.
DMPOTENTIAL		Derivative of magnetic scalar potential.
AX		Components of magnetic vector potential.
AY		
AZ		
INAX		Components of incident magnetic vector potential.
INAY		
INAZ		
EX		Components of electric field.
EY		
EZ		
INEX		Components of incident electric field.
INEY		
INEZ		
ZMULTREALPART		Real and imaginary parts of a factor multiplying the wave impedance of free-space to match a particular wave-guide mode.
ZMULTIMAGPART		
PMIXA		Mixed derivative and potential condition coefficients.
PMIXB		
CMPOTENTIAL		Value for constant magnetic scalar potential.
CVOLTAGE		Value for constant voltage.
CURRENTSOURCE		Value of current flowing through a current source boundary. A positive value indicates current flowing into a meshed volume.
DRIVELABEL		Drive label for functional time variation of assigned values.
POLARIZATION		Bulk conductor terminal polarization:  BOTH Positive and negative. NEGATIVE Negative. POSITIVE Positive.
CIRCUITELEMENT		The name of the bulk conductor circuit element.
PLATEMATERIAL		The material label to used for a <b>THINPLATE</b> .
PLATETHICKNESS		The thickness of the a <b>THINPLATE</b> .

### Command line parameters 3: thermal conditions

Command	BOUNDARY	
Parameter	Default	Function
<b>THERMALCONDITION</b>		Thermal boundary condition:
		<b>COMBINED</b> Combined radiation, heat transfer and heat flux.
		<b>FLUX</b> Specified heat flux.
		<b>INSULATOR</b> Perfect insulator.
		<b>NONE</b> No condition applied.
		<b>RADIATION</b> Radiation.
		<b>THERMALCONTACT</b> Thermal contact.
		<b>TEMPERATURE</b> Specified temperature.
		<b>TRANSFER</b> Specified heat transfer.
<b>TEMPERATURE</b>		Function specifying temperature for use with <b>TEMPERATURE</b> condition.
<b>HEATFLUX</b>		Function specifying heat flux for use with <b>FLUX</b> condition.
<b>HEATTRANSFER</b>		Function specifying heat transfer coefficient for use with <b>TRANSFER</b> condition.
<b>AMBIENTTEMP</b>		Function specifying ambient temperature for use with <b>RADIATION</b> and <b>TRANSFER</b> conditions.
<b>EMISSIVITY</b>		Surface emissivity for a <b>RADIATION</b> condition, in the range 0 to 1.
<b>THERMALCONTACT</b>		Thermal contact conductance.

### Command line parameters 4: stress conditions

Command	BOUNDARY	
Parameter	Default	Function
<b>STRESSCONDITION</b>		Stress analysis boundary condition:
		<b>DISPLACEMENT</b> The surface has a specified displacements in all directions.
		<b>FIXED</b> The surface is cannot move in any direction.
		<b>FREE</b> The surface is free to move.
		<b>NONE</b> The surface has no conditions.
		<b>NORMALDISPLACEMENT</b> The surface has a specified normal displacement.
		<b>NORMFIXED</b> The surface is cannot move normally.
		<b>TANGFIXED</b> The surface is cannot move tangentially.
<b>NORMALDISPLACEMENT</b>		Function specifying the normal displacement of the surface.
<b>DISPLACEMENTX</b>		Function specifying the displacement of the surface.
<b>DISPLACEMENTY</b>		
<b>DISPLACEMENTZ</b>		
<b>STRESSLOAD</b>		Stress analysis load:
		<b>NONE</b> The surface has no load conditions.
		<b>PRESSURE</b> A pressure on the surface.
		<b>TRACTION</b> A force in any direction.
<b>PRESSURE</b>		Function specifying the pressure on the surface.
<b>TRACTIONX</b>		Functions specifying the force on the surface.
<b>TRACTIONY</b>		
<b>TRACTIONZ</b>		

## Notes

This command defines the boundary conditions for use by the analysis programs.

A set of boundary labels is picked using the command repeatedly, with **option=pick** and a **BOUNDARYLABEL** specified. A boundary label can be removed from the set using **option=unpick**. If no **BOUNDARYLABEL** is given, the set is emptied.

Issuing the command with **option=modify** will modify the properties of the set of picked boundary labels to the new values given in the parameters. The value of a property associated with the boundary labels is unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked boundary labels. If the data of one of these parameters is unset, or the picked boundary labels do not share the same value, then the parameter value is left clear.

**option=reset** will clear the properties associated with all of the picked boundary labels.

The properties of all boundary labels can be listed using **option=list**.

Boundary labels that are not used, i.e. have no face referencing them can be deleted using **option-n=delete**. Deleting a boundary label that is in use will clear its properties.

All parameters can be specified. The values of **CONDITION**, **THERMALCONDITION**, **STRESSCONDITION** and **STRESSLOAD** determine which will be used. Many of the conditions have associated data which can be supplied using parameters with the same names as the condition.

The **DRIVELABEL** allows value and functional based boundary conditions to be driven in transient solvers or assigned a phase lag in harmonic solvers.

See [Labels \[page 238\]](#) for more information on valid labels and also [Boundary Conditions \[page 615\]](#).

## The **BRICK8** Command

---

### Summary

Create or modify 8-node bricks.

### Toolbutton



### Command line parameters

Command	BRICK8			
Parameter	Default	Function		
OPTION	NEW	NEW	Creates a new 8-node brick conductor	
		MODIFY	Modifies properties of the picked 8-node brick conductors	
		LOAD	Loads defaults from picked conductors	
DRIVELABEL		Name for the drive label		
LCNAME		Name for Local Coordinate System for coordinate system 1		
SYMMETRY		Rotational symmetry about global Z axis		
XCEN2		Origin of coordinate system 2		
YCEN2				
ZCEN2				
THETA2		Euler angles defining orientation of coordinate system 2		
PHI2				
PSI2				
RXY		Reflection symmetries in XY, YZ and ZX planes		
RYZ				
RZX				

Command	<b>BRICK8</b>		
Parameter	Default	Function	
XP1		First point defining the conductor corners	
YP1			
ZP1			
...			
XP8		Last point defining the conductor corners	
YP8			
ZP8			
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
TOLERANCE		Field calculation tolerance	
INCIRCUIT		Is the conductor part of a circuit:	
	NO	The conductor has defined current density.	
	YES	The current in the conductor is determined by a circuit.	
REVERSE		Reverse the connections to this conductor in its circuit: <b>YES</b> or <b>NO</b> .	
CIRCUITELEMENT		The name of circuit element this conductor is part of.	
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation	
MODELCOMPONENT		NO	Do not convert to meshable cells.
	REGULAR	Convert to meshable cells which will be regularly meshed.	
	YES	Convert to meshable cells which meet the mesh size criteria.	
MESHSIZE		The size of the mesh to be used when meshing	
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new 8-node brick conductor or modifies existing 8-node bricks.

- **OPTION=NEW** creates a new 8-node brick conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked 8-node bricks. If the picked 8-node bricks do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking 8-node bricks (see [The PICK Command \[page 277\]](#)) to be modified and before **BRICK8 OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked 8-node bricks to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Bricks \[page 525\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **BRICK20** Command

---

### Summary

Create or modify 20-node bricks.

### Toolbutton



### Command line parameters

Command	BRICK20		
Parameter	Default	Function	
OPTION	NEW	NEW	Creates a new 20-node brick conductor
		MODIFY	Modifies properties of the picked 20-node brick conductors
		LOAD	Loads defaults from picked conductors
DRIVELABEL		Name for the drive label	
LCNAME		Name for Local Coordinate System for coordinate system 1	
SYMMETRY		Rotational symmetry about global Z axis	
XCEN2		Origin of coordinate system 2	
YCEN2			
ZCEN2			
THETA2		Euler angles defining orientation of coordinate system 2	
PHI2			
PSI2			
RXY			
RYZ		Reflection symmetries in XY, YZ and ZX planes	
RZX			

Command	<b>BRICK20</b>		
Parameter	Default	Function	
XP1		First point defining the conductor corners	
YP1			
ZP1			
XP20		Last point defining the conductor corners	
YP20			
ZP20			
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
TOLERANCE		Field calculation tolerance	
INCIRCUIT		Is the conductor part of a circuit:	
		NO	The conductor has defined current density.
		YES	The current in the conductor is determined by a circuit.
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.	
CIRCUITELEMENT		The name of circuit element this conductor is part of.	
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation	
MODELCOMPONENT		NO	Do not convert to meshable cells.
		REGULAR	Convert to meshable cells which will be regularly meshed.
		YES	Convert to meshable cells which meet the mesh size criteria.
MESHSIZE		The size of the mesh to be used when meshing	
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new 20-node brick conductor or modifies existing 20-node bricks.

- **OPTION=NEW** creates a new 20-node brick conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked 20-node bricks. If the picked 20-node bricks do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking 20-node bricks (see [The PICK Command \[page 277\]](#)) to be modified and before **BRICK20 OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked 20-node bricks to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Bricks \[page 525\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **CEDITOR** Command

---

### Summary

Control the Circuit Editor.

### Toolbutton



### Command line parameters

Command	<b>CEDITOR</b>	
Parameter	Function	
<b>OPTION</b>	Command option:	
	<b>CLEAR</b>	Clear all circuit data and close the Circuit Editor window.
	<b>CLOSE</b>	Close the Circuit Editor window.
	<b>COPYCOMPONENT</b>	Copy a circuit component.
	<b>IMPORT</b>	Import a circuit from a file adding to existing data.
	<b>LOAD</b>	Clear all circuit data and load a circuit from a file.
	<b>MODIFY</b>	Modify the value of the property of a component.
	<b>PLACE</b>	Position an unused component in the circuit.
	<b>PRINT</b>	Create a picture of the circuit in a file.
	<b>REVERSE</b>	Change the direction of a component in the circuit.
	<b>SAVE</b>	Save circuit data in a file.
	<b>START</b>	Open the Circuit Editor window with any existing circuit information.
	<b>UPDATE</b>	Copy any changes made interactively into the model.

<b>FILENAME</b>	Name of circuit data file for <b>LOAD</b> , <b>IMPORT</b> or <b>SAVE</b> . Name of picture file for <b>PRINT</b> .
<b>COMPONENTNAME</b>	Name of component for <b>MODIFY</b> .
<b>PROPERTYNAME</b>	Name of property for <b>MODIFY</b> .
<b>VALUE</b>	New value of property for <b>MODIFY</b> .
<b>SOURCECOMPONENT</b>	Name of the source component in <b>COPYCOMPONENT</b> or the adjacent component in <b>PLACE</b> .
<b>PREFIX</b>	A prefix to be added to the names of components <b>IMPORT</b> ed.

## Notes

The **CEDITOR** command controls the Circuit Editor. The Circuit Editor is designed for interactive use. Once started, wires and components can be drawn on the Circuit Editor window (see [Figure 3.3, below](#)) and properties can be assigned to components using its menus, toolbars and property sheets.

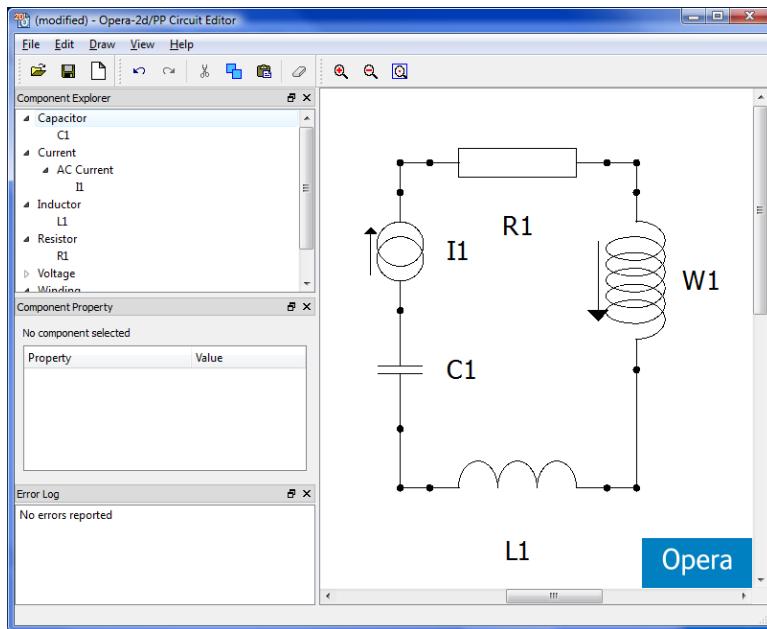


Figure 3.3 A simple RLC circuit including an Opera winding

The figure shows a circuit from Opera-2d. However, the circuit editor is the same and circuit data files (**\*.vfc**) are compatible between Opera-2d and Opera-3d. Some of the options shown depend on the current analysis type; for example **Functional Voltage** is not available if harmonic analysis has been selected in the Modeller.

When the Circuit Editor is closed, the data is transferred to the Modeller and will be saved to any data file (\*.**opc**). The Circuit Editor can save and load its own data files (\*.**vfc**) so that circuit information can be transferred between models.

## Options

In normal interactive use, the Modeller issues the command:

```
ceditor option=start
```

After the circuit data has been prepared (see [Circuit Editor Interaction \[page 715\]](#)), the action of closing the window gives the option of accepting or cancelling the changes in the model.

The other **OPTIONs** allow the Circuit Editor to be controlled from the command line or a command file. The topology of a circuit can only be modified interactively but the properties of circuit components can be changed by command line.

Options to control the Circuit Editor window:

- **START**: to open the Circuit Editor window.
- **CLOSE**: to close the Circuit Editor window, accepting any changes made interactively since the last **UPDATE**.
- **PRINT**: to create a picture file of the circuit. The file format is determined by the file type of the **FILENAME**. Files are overwritten without asking.

The following **OPTIONs** do not require the Circuit Editor window to be open:

- **CLEAR**: to delete all circuit data and close the Circuit Editor window if it is open. Any circuit data stored in the application will also be cleared.
- **MODIFY**: to change the value of a property of a component, e.g.  
`ceditor option=modify component=r1 resistance 10`

**MODIFY** is designed for scripting, not for interactive use. Component names and property names are not case sensitive but they cannot be abbreviated. Names which contain spaces must be enclosed in single quotation marks, e.g.

```
propertyname='resistance per unit length'
```

If **MODIFY** is used to change a property value of a component, it is necessary for the component to be deselected and reselected before the changed property can be seen in the Circuit Editor window.

Changes made with **MODIFY** cannot normally be undone. The behaviour is undefined if the circuit is also being modified graphically.

- **COPYCOMPONENT**: to copy the component given by **SOURCECOMPONENT** to create a new component with the name given by **COMPONENTNAME**. The new component can then be added to the circuit using the **PLACE** option.
- **PLACE**: to insert an unused component given by **COMPONENTNAME** so that it is placed immediately after the component given by **SOURCECOMPONENT**.
- **REVERSE**: to reverse the component given by **COMPONENTNAME** in the circuit.

- **UPDATE:** to copy any changes made interactively into the model<sup>1</sup>. It is necessary to **UPDATE** after a set of **MODIFY** or **LOAD** commands to accept changes, otherwise a question will be asked to confirm the changes.
- **LOAD, IMPORT** and **SAVE:** to access Circuit Editor data files (\*.vfc).  
**INSERT** can rename the components read from a **vfc** file so that they do not match the names of existing components. This is done by adding a **PREFIX** to the names of the new components. The **PREFIX** can be empty if none of the names in the **IMPORT**ed circuit is already used by components in the existing circuit.

The **CEDITOR** command cannot be used if existing circuit data has been defined using the **CIRCUIT** command. To access the **CEDITOR** command all previously defined circuit data must be cleared.

However the **CIRCUIT** command can be used to view the circuit data created by the **CEDITOR** command, if there are no errors in the circuit.

## Circuit Editor Interaction

### Drawing wires

Double clicking in empty space on the diagram can be used to start a new wire. Once started, double-clicking continues the wire. Double-clicking on an existing wire makes the new wire join the exiting wire at the position clicked. Wires can pass over an existing wire, and no connection node will be formed.

As the wire is being drawn, a preview of the wires will be shown if it is possible to create a connection between the start and end points with 1 or 2 wires.

Press the left button to complete the wire.

Cancel the drawing by moving out of the view pane or by clicking the right mouse button.

### Drawing components

Position the mouse over a component or component type in the left hand list view.

Press and hold the left mouse button and drag the component into the diagram. The position of the component will be displayed as the cursor moves if it is possible to drop the component at that position. Release the left mouse button to drop the component into the diagram at the cursor position.

Components can be dropped onto empty space or onto wires that are sufficiently long to include the component. If positioned on a vertical wire, the component will be rotated to form connections with the wire.

To cancel the drop move the cursor back into the left hand component list view.

---

<sup>1</sup>The **UPDATE** option is not available in the Post-Processor.

Double clicking over a component will select the component for property modification in the left hand property sheet. Values can be entered into the mutual inductance matrix for the whole circuit using the menu route: **Edit -> Set Mutual Inductance**.

## Opera-3d windings

Circuit winding elements available in the Modeller are displayed in the **Component Explorer** under **Windings**. These should be dragged into the relevant positions in the circuit diagram. The mesh type of the winding can only be set in the Modeller.

## Opera-3d bulk eddy current conductors

Bulk Eddy current conductors available in the Modeller are displayed in the **Component Explorer** under **Bulk Conductor**. These should be dragged into the relevant positions in the circuit diagram. Values for the symmetry factors for the length and cross-sectional area of the conductor must be set if the complete conductor is modelled by a symmetric sub-section.

## Selecting components to move them

Components can be selected by:

- Pressing the left mouse button while over a component.
- Pressing the **Ctrl** or **Shift** key while selecting a second component will allow multiple components to be selected. If **Ctrl** or **Shift** is not pressed, all other components will be deselected when the mouse button is released.
- Pressing the left mouse button over empty space in the diagram, holding and dragging to create a selection area.

If the selection area starts from left and moves right, components that are completely enclosed in the selection area will be selected.

If the selection area starts at the right and is dragged left, any component that intersects the selection area will be included.

## Moving or copying selected components

Select the components to be moved as above.

Press and hold the left mouse button while the cursor is over one of these items. As the cursor is moved a preview of the moved components will be displayed. The items will not be displayed if the position to which they are moved is invalid.

Release the left mouse button when the components are in the correct place.

By default, connections between components will be maintained for the moving parts.

To disconnect the moving components press the **Ctrl** key while moving the cursor or when the button is released.

To take a copy of the components and leave the existing components in-situ, press the **shift** key while moving and dropping the components.

The mouse button press will perform a select on a new component, allowing movement of a single component by pressing and moving as a single operation.

## Copy/Cut and Paste

Selected items can be copied by using the **Edit -> Copy** menu item (**Ctrl+C**).

Items can be pasted by pressing **Edit -> Paste** menu item (**Ctrl+V**). On pressing the paste menu item, the Circuit Editor enters a paste positioning mode. A preview of where the components will be pasted is shown as the cursor moves over the diagram.

To position the components, press the left mouse button. To cancel the paste operation, press the right mouse button.

## Other operations

Other operations can be performed on components once they are selected using a context menu which appears after a right mouse button click.

Components can be:

- Reversed, connections are maintained.
- Rotated, connections are broken and the operation will only happen if the rotated component can be positioned in its new orientation.
- Deleted or Removed. Removed components are removed from the diagram but remain in the component list view.

Finally, a set of operations is available to help with building and saving the circuits on file:

- Unlimited Undo and Redo until a further change is made.
- Save, Load or Import a transferable circuit file (**\*.vfc**).
- Export to Opera-3d as a command file (**\*.comi**).
- Generate a report containing details of all the components in a **html** file.

## The CELldata Command

---

### Summary

Sets properties of picked cells.

### Toolbutton



### Command line parameters

Command	CELldata			
Parameter	Default	Function		
OPTION	MODIFY	MODIFY	Applies new values to the picked cells.	
		RESET	Clears all data and sets to the default values.	
MATERIALLABEL	See notes	Material label:		
POTENTIAL	See notes	DEFAULT	Potential type of the volume.	
		REDUCED		
		TOTAL		
ELEMENTTYPE	See notes	LINEAR	Type of finite element in the volume.	
		QUADRATIC		
VOLUMELABEL	See notes	Volume property label.		
GROUPLABEL	See notes	Group label used by the Motional EM solver.		
SIZE	See notes	Mesh control size.		
NORMALTOL	See notes	Maximum normal angle between mesh nodes.		
SURFACETOL	See notes	Maximum deviation of the mesh from the surface.		

<b>LEVEL</b>	See notes	Data storage level for the cell data.	
<b>ELEMSHAPEPREF</b>	See notes	Element shape preference:	
		<b>HEXORPRISM</b>	Hexahedra or prisms.
		<b>NONE</b>	No preference.
		<b>TETRAHEDRAL</b>	Tetrahedra.

## Notes

This command is used to set or clear the properties of all picked cells.

When a cell is created, it has default properties of

- **MATERIAL=AIR**,
- **POTENTIAL=DEFAULT**,
- **ELEMENTTYPE=LINEAR**,
- **ELEMSHAPEPREF=NONE**.

These properties are also set when the cell's data is cleared using **OPTION=RESET**.

**OPTION=MODIFY** can be used to change the value of all of the picked cells. Any unset parameter values are not modified. The new value of parameters that have been set replace the existing values in the data attached to the cells.

Each time a cell is picked the value of each of the parameters is updated to be the common value of all the picked cells. If the data of one of these parameters is unset, or the picked cells do not share the same value, then the parameter value is left clear.

After issuing the command with **OPTION=MODIFY**, the list of picked items is cleared. Issuing the command with **OPTION=RESET** keeps the same set of picked objects.

The **MATERIALLABEL** parameter controls the material label attached to a cell. The properties associated with such a label are set using the **MATERIALS** command.

The **ELEMENTTYPE** parameter allows the user to choose to have linear or quadratic elements in a cell when creating the database with mixed elements. See [The SOLVERS Command \[page 309\]](#) command for greater detail.

The **ELEMSHAPEPREF** parameter sets a preference for the element shape in a cell. The choices are as follows.

- **NONE**: no preference; the cell can be meshed using tetrahedra, hexahedra or prisms.
- **TETRAHEDRAL**: use tetrahedral elements. This preference will be respected and the cell will be meshed with tetrahedra.
- **HEXORPRISM**: use hexahedra or prism elements. The cell will be meshed with hexahedra or prisms if:
  - the cell has a suitable topology for hexahedra or prisms, and

- the MESH command TYPE parameter is not set to TETRAHEDRAL.

If either of these conditions is not met, tetrahedra will be used.

See [The MESH Command \[page 257\]](#) for more information on how the cell preferences are used when the mesh is generated.

The POTENTIAL parameter defines the potential type that is to be used in elements created in the cell. POTENTIAL=DEFAULT will give a REDUCED potential for cells with MATERIALLABEL=AIR, and TOTAL potential for all others. The use of the different potentials is dependent upon the analysis program. (AUTOMATIC is an alias for DEFAULT, but should not be used for new models.)

The VOLUMELABEL parameter holds the volume property label that may be attached to a cell. The properties associated with such a label are set using the VOLUME command.

The GROUPLABEL parameter must be set for each cell when building a model for CARMEN analysis. For other analysis types it is not used. Every cell with the same GROUPLABEL is considered to be part of the same group. Each of these groups can be moved independently as a rigid body. A set of cells separating groups with other group labels must be given GROUPLABEL=GAP. Having 2 neighbouring cells with different group labels will cause an error in the Motional Em solver unless one of these labels is GAP.

The SIZE, NORMALTOL and SURFACETOL parameters control the mesh size of elements in the cell and near the faces of the cell, when generating the surface and volume mesh.

The LEVEL parameter controls the storage of data when there is a conflict during the merging of multiple cells. The data set with the greater level will be maintained. The result of merging 2 cells with the same level is indeterminate.

## The **CHECK** Command

---

### Summary

Checks a picked body for topological problems.

### Toolbutton



### Command line parameters

Command	<b>CHECK</b>		
Parameter	Default	Function	
<b>LEVEL</b>	<b>HIGH</b>	<b>LOW</b>	Level of detail used in the checking
		<b>MEDIUM</b>	
		<b>HIGH</b>	
		<b>FULL</b>	
<b>REPAIR</b>	<b>NO</b>	<b>NO</b>	Attempt to repair errors found during checking
		<b>YES</b>	
<b>FILENAME</b>			Output the results of the checking to the specified file

### Notes

This command checks for topological problems of a single picked body. If problems exist, the objects are given a **SYSTEM** label and information regarding the problem is also attached to the body, or constituent parts of the body. This can be seen by listing the data on the object through the **FILTER COMMAND=LIST** command.

Generally the parameter **LEVEL** should be **HIGH** or **FULL**, as this will report most possible errors within the model. If errors exist, **repair=yes** can be used to try to fix them. The repair might improve the condition of the model by improving the model's geometry and by creating tolerant entities. However, tolerant entities might themselves fail during some operations, so might not be a cure for all models. Very short edges and sliver faces will also be removed by the repair.

The **REPAIR** option can only be used on component bodies, not on the model body.

If problems exist with a body, it may be necessary to undo the operation that caused the problem and try an alternative method to generate the model. Failure to do this could cause the body to fail in future operations.

There are many reasons why a body fails the check command. The output of the check command can be output to a **FILENAME** if more details are required.

## The **CIRCUIT** Command

### Summary

Define circuits.

### Toolbutton



### Command line parameters

Command	<b>CIRCUIT</b>	
Parameter	Function	
OPTION	CHOOSEELEMENT	Set the active element in the circuit dialog
	CHOOSETYPE	Set the active element type in the circuit dialog
	CLEAR	Remove all circuit data
	DELETEELEMENT	Delete a circuit element
	INIT	Initialize the circuit dialog
	LOOPADD	Add an element to a current loop
	LOOPDELETE	Remove an element from a loop
	NEWLOOP	Create a new loop
	REVERSE	Reverse the direction of an element in a loop
SETELEMENT	Create or modify a new circuit element	
ELEMENTNAME	Name of element to be added, changed, deleted etc.	

Command	CIRCUIT				
Parameter	Function				
TYPE	BULKCONDUCTOR	Bulk eddy current conductor			
	CAPACITOR	Capacitor			
	CURRENT	Current source			
	DIODE	A piece-wise linear or exponential diode			
	INDUCTOR	Inductor			
	MUTUALCOUPLING	Inductive coupling			
	RESISTOR	Resistor			
	SWITCH	High resistance switch			
	VOLTAGE	Voltage source			
	WINDING	Circuit winding			
RESISTANCE	Resistance of the resistive element				
INDUCTANCE	Inductance of the inductive element				
INITIALCURR	Initial current for transient solutions				
CAPACITANCE	Capacitance of the capacitive element				
INITIALCAPVOLT	Initial voltage on capacitor				
VOLTAGE	Voltage from the voltage source element				
CURRENT	Current from the current source element				
RPUL	Resistance Per Unit Length of the winding element				
URNS	Number of turns in the winding element				
FILAMENTS	Number of representative filaments in the winding element				
SYMMETRY	Symmetry adjustment factor needed to create the full winding				
DRIVELABEL	Name of the drive function controlling the source strength				
LOOPNAME	Name of the current loop being controlled				
LOOPELEMENT	Name of active element in the current loop				
NEWLOOPNAME	Name for a new loop being created				
COUPLEDELEMENT	Circuit element to be linked through a mutual inductance				
USEMODELSSYMMETRY	YES	NO	User defines the symmetry value		
		YES	Use the model symmetry set. See "The BACKGROUND Command" on page 135.		

Command	<b>CIRCUIT</b>	
Parameter	Function	
<b>DIODEMODELTYPE</b>	Type of diode:	
	<b>EXPONENTIAL</b>	Exponential current-voltage characteristic, set by <b>ISATURATION</b> and <b>VTCOEFF</b>
	<b>PIECEWISELINEAR</b>	Two-resistance model switching between <b>ROPEN</b> and <b>RCLOSED</b> at voltage <b>VSWITCH</b>
<b>ISATURATION</b>	Diode saturation current	
<b>VTCOEFF</b>	Diode thermal voltage multiplicative constant	
<b>VSWITCH</b>	Diode switching voltage	
<b>ROPEN</b>	Diode resistance when open	
<b>RCLOSED</b>	Diode resistance when closed	
<b>SWITCHTESTTYPE</b>	Type of test on <b>SWITCH FUNCTION</b> to change state of switch:	
	<b>CLOSEDGEZERO</b>	Close if greater than or equal to zero
	<b>CLOSEDGTZERO</b>	Close if greater than zero
	<b>CLOSEDIFTRUE</b>	Close if true (nearest integer is non-zero)
	<b>OPENGEZERO</b>	Open if greater than or equal to zero
	<b>OPENGZERO</b>	Open if greater than zero
	<b>OPENIFTRUE</b>	Open if true (nearest integer is non-zero)
<b>SWITCHFUNCTION</b>	Expression evaluated to open or close the switch	
<b>SWITCHOPENRES</b>	Resistance of switch when open	
<b>SYMMETRYLENGTH</b>	Symmetry factor to create full length of bulk eddy current conductor.	
<b>USEMODELSYMMETRYLENGTH</b>	Use model symmetry rather than <b>SYMMETRYLENGTH</b> : <b>YES</b> or <b>NO</b>	
<b>SYMMETRYLENGTH</b>	Symmetry factor to create full cross-section of bulk eddy current conductor.	
<b>USEMODELSYMMETRYSECTION</b>	Use model symmetry rather than <b>SYMMETRYSECTION</b> : <b>YES</b> or <b>NO</b>	

## Notes

Circuits can be defined using the **CIRCUIT** command or by [The CEDITOR Command \[page 162\]](#). If the circuit has been defined using the **CEDITOR** command, the **CIRCUIT** command can only be used to view the data. This is best done using the GUI which displays the current values as each component is selected.

If the **CIRCUIT** command is used to define a new circuit, it has two main roles.

- To define the circuit elements (resistors, capacitors, inductors, switches, diodes, voltage or current sources) and the properties of the circuit conductors defined in the model.
- To specify the connectivity of the elements in the circuit by defining a set of current loops.

The creation of components and the specification of the circuit loops can be performed at any time before the analysis database is created. There are however two exceptions. Before a current loop can be created all the components forming the loop must be defined. The circuit conductors must be created and their properties defined before the model body is generated. This second rule is a consequence of the method used to represent the circuits within the finite element analysis.

A circuit element is created by issuing the command with **OPTION=SETELEMENT**. The **TYPE** parameter is used to define whether the element is a resistor, inductor etc. and the **ELEMENTNAME** parameter is used to specify a unique name for the component. Any properties of the component can be set by assigning a value to the appropriate parameters for the component being created. For example to create a new 100 ohm resistor named R1 the following command can be issued,

```
CIRCUIT OPTION=SETELEMENT TYPE=RESISTOR,
ELEMENTNAME='R1' RESISTANCE=100
```

Ten different circuit element types can be defined,

1. Resistors.

The resistance of the component is defined using the **RESISTANCE** parameter. This value can be functional if required.

2. Inductors.

The inductance of the component is defined using the **INDUCTANCE** parameter. This value can be functional if required.

A value of initial current can be defined. If set, this value is used as an initial condition to define currents in the starting solution for the transient solution.

3. Capacitors.

The capacitance of the component is defined using the **CAPACITANCE** parameter. This value can be functional if required.

In transient analysis, capacitors can have an initial voltage applied using **INITIALCAPVOLT**.

4. Voltage sources.

The **VOLTAGE** parameter can be used to set the supply voltage. An optional drive label can also be defined, **DRIVELABEL**. This value can be functional if required.

5. Current sources.

The **CURRENT** parameter can be used to set the current. An optional drive label can also be defined, **DRIVELABEL**. This value can be functional if required.

## 6. Windings.

Windings form a link between the finite element model and the circuit. Windings are represented in the model by a conductor or a set of conductors that have been created as a circuit component with a circuit element name. There are 2 representations of winding circuit elements within the finite element solution.

- Volume meshed

Created from conductors that are defined as circuit elements and have been set to be volume meshed.

These conductors are represented within the finite element solution by an integral over the elements which have been created to represent the conductors.

In general these will provide a more accurate calculation of the inductance and field solution local to the conductor than the equivalent filament mesh winding. Generation of volume meshed conductors will increase the complexity of the model body and the time taken to create it.

- Filamentary

Created by conductors that are defined as circuit elements but are not set to be volume meshed.

These are represented by one or more closed loops of edges within the conductor cross-section.

Filamentary and volume meshed conductors may not be combined within a single circuit winding element.

A circuit winding represents a number of **TURNS** of wire. The **SYMMETRY** parameter can be used to adjust the model when symmetry boundary conditions are used and the circuit conductor does not lie entirely within the finite element mesh. The **SYMMETRY** parameter allows representation of series wound images of the winding that are not included within the finite element mesh. If half the winding is included in the mesh then **SYMMETRY=2**, if a quarter then **SYMMETRY=4** etc. By default this is set to the model symmetry, [See "The BACKGROUND Command" on page 135](#).

For volume meshed circuit windings, the properties of the wire, i.e. conductivity and wire cross-section are defined as material properties of a material label with the same name as the circuit element. If these values are unset, the winding is treated as lossless.

For filamentary windings, the resistance per unit length of the winding, is defined through the **RPUL** parameter. The **FILAMENTS** parameter controls the representation of the winding in the finite element mesh. For example **FILAMENTS=1** means that the winding will be represented in the mesh by a single filament through the centre of the conductor. To improve the accuracy of models a higher number of filaments can be used. **FILAMENTS=2** will give a set of 4 filaments etc.

For a **QUENCH** analysis, volume meshed elements must be used, and an additional wire material property defining the critical current, probably as a function of temperature, T, and flux density, B, should be given.

## 7. Bulk Eddy Current Conductors also link to the finite element model. Bulk conductors are modelled as one or two boundary condition surfaces which form either or both of the input and output terminals of the conductor. The resistance and inductance of the conductor are calculated by its material properties and its position in the finite element model.

If the symmetry of the finite element model reduces the length or cross-sectional area of a bulk conductor compared to the conductor which should appear in the circuit, this should be specified using the parameters **SYMMETRYLENGTH**, **USEMODEL<sub>S</sub>YMMETRYLENGTH**, **SYMMETRYXSECTION** and **USEMODEL<sub>S</sub>YMMETRYXSECTION**. For example, if symmetry of the model cuts the length of the conductor by a factor of 4 and its cross-sectional area by a factor of 2, use

**SYMMETRYLENGTH=4 , SYMMETRYXSECTION=2**

#### 8. Mutual coupling elements.

These define a mutual inductance term between 2 inductive elements.

- An **ELEMENTNAME** and a **COUPLEDELEMENT** must be specified. These circuit elements must be either of type **INDUCTOR** or **WINDING**. A value of **INDUCTANCE** defines the inductance linking these 2 circuit elements.
- For electromagnetic analyses, the self and mutual inductance between circuit elements is assumed to be calculated as part of the electromagnetic solution, so mutual coupling terms between 2 winding elements are ignored.
- For a Quench analysis, giving the same element name in **ELEMENTNAME** and **COUPLEDELEMENT** parameters allows a self-inductance value to be given for winding elements.

#### 9. Switches.

These are modelled by a zero (closed) or large resistance (**SWITCHOPENRES**). The switch operates when a user defined expression (**SWITCHFUNCTION**) is in a particular range of values given by **SWITCHTESTTYPE**. (See **SWITCHTESTTYPE** [page 175] for the possible tests.)

In general, switching should not depend on a function of a closely related circuit property, e.g. using the value of current through a switch to cause it to switch will be highly unstable as the discontinuity in resistance will prevent convergence between nonlinear iterations.

#### 10. Diodes.

There are two models of diode:

- **DIODEMODELTYPE=PIECEWISELINEAR** switches between two resistance values: **ROPEN** when the voltage is less than or equal to **VSWITCH** and **RCLOSED** when the voltage is greater than **VSWITCH**.
- **DIODEMODELTYPE=EXPONENTIAL** uses values of saturation current ( $I_s$ ) and thermal voltage coefficient ( $n$ ) to define an exponential current-voltage characteristic:

$$V = nV_T \log\left(\left(1 + \frac{I}{I_s}\right)\right) \quad (3.1)$$

where  $V_T$  is the thermal voltage at room temperature ( $V_T = kT/q_e = 25.8$  mV, where  $k$  is the n constant). The user must supply values of **VTCOEFF=n** and **ISATURATION=Is**. The resulting resistance ( $V/I$ ) is limited by the values of **RCLOSED** and **ROPEN**.

Nonlinear analysis must be used to converge to the true solution.

The **CIRCUIT** command with **OPTION=SETELEMENT** can be used to modify the properties of components that have already been defined. A component can be deleted by issuing the command with

**OPTION=DELETEELEMENT** and naming the element to be deleted using the **ELEMENTNAME** parameter.

The circuit is defined in terms of loops. These loops are created using **OPTION=NEWLOOP** and giving a unique name for the loop through the **NEWLOOPNAME** parameter. For example, the following line will create a new loop named Loop1.

```
CIRCUIT OPTION=NEWLOOP NEWLOOPNAME=' LOOP1'
```

Components are added to a loop using the command with **OPTION=LOOPADD**. The loop to which the component is to be added is specified by the **LOOPNAME** parameter and the component to be added is given by the **ELEMENTNAME** parameter. The component is positioned in the loop immediately after the element given by **LOOPELEMENT**. This should be preceded by a command with **OPTION-N=CHOOSEELEMENT**, to identify the existing element which is to be added and load its properties; otherwise **OPTION=LOOPADD** would redefine them.

For example, to add the resistor R1 at the start of the loop LOOP1,

```
CIRCUIT OPTION=CHOOSEELEMENT ELEMENTNAME=' R1'  
CIRCUIT OPTION=LOOPADD ELEMENTNAME=' R1' ,  
LOOPNAME=' Loop1' LOOPELEMENT=
```

A component can be removed from the loop using **OPTION=LOOPDELETE**, as is shown in the following example,

```
CIRCUIT OPTION=LOOPDELETE LOOPNAME=' Loop1' ,  
LOOPELEMENT=' R1'
```

If a loop is no longer required it can be removed by leaving the **LOOPELEMENT** parameter empty in the line above.

## The **CLEAR** Command

---

### Summary

Clears the model of all data or reloads the open file.

### Toolbutton



### Command line parameters

Command	<b>CLEAR</b>		
Parameter	Default	Function	
<b>REVERT</b>	<b>NO</b>	<b>NO</b>	Clears and re-initializes all data within the Modeller.
		<b>YES</b>	Clears all data and reloads the previously open file.
<b>USERVARIABLES</b>	<b>YES</b>	Clears all numeric user variables and user defined functions: <b>YES</b> or <b>NO</b> .	
<b>STRINGVARIABLES</b>	<b>YES</b>	Clears all string variables: <b>YES</b> or <b>NO</b> .	
<b>LOGFILES</b>	<b>NO</b>	Restarts <b>Ip</b> and <b>log</b> files: <b>YES</b> or <b>NO</b> .	
<b>PYTHON</b>	<b>YES</b>	Reset the Python interpreter: <b>YES</b> or <b>NO</b> .	
<b>OBJECTSTORE</b>	<b>YES</b>	Clear all <b>OperaObjects</b> from the object store: <b>YES</b> or <b>NO</b> .	

### Notes

This command is used to reset the Modeller to its initial state. All history states are cleared, so this operation cannot be undone. The parameters can be used from the console or in command files to adjust the behaviour.

- **REVERT=YES**: the open file is re-loaded to remove changes from the model that are not required.
- **USERVARIABLES=NO**: numeric user variables are not deleted. This setting allows **CLEAR** to be used in a command loop. Without this setting a **CLEAR** command in a command loop will cause the command input to be aborted, i.e. any pending commands from the command loop or from any other loops or any command input file containing the loop will be ignored and command input will return to the user.  
If numeric user variables are not deleted then user defined functions are kept as well.
- **STRINGVARIABLES=NO**: string variables created by the user are not deleted.

- **LOGFILES=YES**: the contents of the ***Ip*** and ***log*** files (see [Output Files \[page 25\]](#)) will be deleted and only subsequent input and output will be recorded.
- **PYTHON=YES**: all variables and functions in the Python memory will be cleared.
- **OBJECTSTORE=YES**: OperaObjects (used for transfer of data between Opera and Python "Python Arrays in Post-Processing" on page 891) will be cleared.

Following a **CLEAR** command (unless the **CLEAR** command is in a command input file), the program looks for a file called ***modeller.comi***, first in the current project folder and then in the user's home folder, and if it exists, it is opened and read as a **\$ COMINPUT** file (see [Command Input Files \[page 61\]](#)) before control is passed back to the user. This allows the user to supply an individual choice of default values for commands or define frequently used **\$ CONSTANTs** and **\$ PARAMETERs** (see [User Variable Commands \[page 55\]](#)).

The **CLEAR** command is run automatically as part of [The LOAD Command \[page 243\]](#) with **OPTION-  
N=NEW**.

## The COLOUR Command

---

### Summary

Sets the colour attached to a display label.

### Toolbutton



### Command line parameters

Command	COLOUR		
Parameter	Default	Function	
OPTION	SET	Option:	
		EXPORT	Save the current set of colours as a command input file.
		LOAD	Load current values as defaults.
		RESTORE	Restore the default set of colours.
		SET	Set new values.
CODE		Code number for the colour to be altered	
PROPERTY		Property type for item to be set	
LABEL		Individual label of property	
RED		Value of red component (0 - 255)	
GREEN		Value of green component (0 - 255)	
BLUE		Value of blue component (0 - 255)	
TRANSLUCENT	NO	NO	Switch off colour translucency
		YES	Set colour property to be translucent

### Notes

This command sets the COLOURs used for the display of the model.

Properties displayed within the Modeller are assigned an integer colour CODE. If this code number has been specified as a parameter this will be used. If not, a property and label can be specified, and the code for this property will be found and used, e.g. PROPERTY=MATERIAL and LABEL=AIR will change the colour of the display of items with material label air.

With **OPTION=SET**, the colour is changed to the values of **RED**, **GREEN**, **BLUE** and **TRANSLUCENT** supplied. If any value is not given, this component remains unchanged.

Making a material colour translucent (**TRANSLUCENT=YES**) makes other materials inside it visible. Only one material (or all the contour colours) can be translucent at a time. All other materials will be opaque (**TRANSLUCENT=NO**).

Colours can be saved to a command input file called ***modeller\_colours.comi*** with **OPTION=N=EXPORT**. The default set of colours can be restored using **OPTION=RESTORE**.

## The **COMBINE** Command

---

### Summary

Combines picked bodies using a boolean operation.

### Command line parameters

Command	COMBINE		
Parameter	Default	Function	
<b>OPERATION</b>	<i>none</i>	<b>CUTAWAY</b>	Leaves the first body unchanged and changes each of the other picked bodies by removing its intersection with the first body.
		<b>INTERSECT</b>	Merge the bodies so that the resulting body has only the components that existed in <b><i>all</i></b> of the original bodies.
		<b>SUBTRACT</b>	Changes the first picked body by removing from it the intersection with each of the other picked bodies in turn. The other picked bodies are then deleted.
		<b>TRIM</b>	As subtraction, but the other picked bodies are not deleted.
		<b>UNION</b>	Merge the bodies so that the resulting body has all the components that existed in <b><i>any</i></b> of the original bodies.
<b>REGULAR</b>	<i>YES</i>	<b>YES</b>	Regularize the result of the boolean operation
		<b>NO</b>	Do not regularize the result

### Notes

This command combines all picked bodies using the specified operation. At least 2 bodies must be picked for these operations. The resulting body will have the name of the first picked body.

**UNION** will form the sum of the picked bodies in a single body. After the operation the original bodies no longer exist within the model.

**INTERSECT** keeps the common volume that exists within all picked bodies. After the operation the original bodies no longer exist within the model.

**SUBTRACT** returns a body containing cells occupying the volume that exists in the first body picked, and which does not exist in any of the subsequently picked bodies. After the operation the original bodies no longer exist within the model.

**TRIM** leaves all picked bodies except the first untouched, and trims the first body so that it has no overlap with the other picked bodies.

**CUTAWAY** leaves only the first picked body untouched, and cuts away the first body from each of the others, so that it has no overlap with the other picked bodies.

If **REGULAR=YES**, faces, edges and vertices that are affected by the boolean operation are checked to see if they are needed to support the geometry. If not, they are removed and the body simplified. Internal faces formed by the operation are not needed to support the geometry, and so are removed. Faces bounding cells that have no volume, i.e. sheets, are also removed. Edges that bound two faces with the same underlying geometry are removed.

If the bodies do not overlap, the **INTERSECT** operation will have the effect of removing all the picked bodies.

If the first picked body is completely surrounded by the other picked bodies, the **SUBTRACT** operation will also remove all picked bodies.

It should be noted that the non-regular operations may leave sheet faces. Such sheet faces can be picked and deleted using [The DELETE Command \[page 200\]](#).

## The **COMMENT** Command

### Summary

Adds comments into the database when creating a new simulation for analysis.

### Command line parameters

Command	COMMENT	
Parameter	Default	Function
TEXT	<i>none</i>	Line of text to be added to the stored comment text
CLEAR	NO	NO
		YES
TYPE	DBTITLE	DBTITLE
		Type of comment

### Notes

This command adds text as a comment for storing with a database simulation. These comments can be used to add information about the simulation for later reference. The Post-Processor stores the first line of the title from the loaded database in the string variable **TITLE** so that it can easily be used, for example, in the **TITLE** command.

The **COMMENT** command can be run any number of times. Lines of text are added into the string for storing in the database. This stored string can be cleared using **CLEAR=YES**.

## The COMPATIBILITY Command

---

### Summary

Allows superseded features to be used in the latest version of the program.

### Command line parameters

Command	COMPATIBILITY	
Parameter	Default	Function
LOCALPICKING	YES	Use local numbering in PICK and SELECT:
		YES Use local numbering
		NO Use global numbering.
IDVERSION	19	Sets the version numbering scheme to be used.

### Notes

The IDVERSION parameter is used to control the setting of identifiers on entities within a model.

- IDVERSION=19 was first implemented in Opera 15R3.
- IDVERSION=18 was first implemented in Opera 15.
- IDVERSION=17 was first implemented in Opera 14.
- IDVERSION=16 was first implemented in Opera 13.
- IDVERSION=9 was first implemented in Opera 12.
- IDVERSION=0 is the version implemented in Opera 11 and earlier.

The change to IDVERSION was implemented to improve the reliability of command history replay, where replay of an operation may alter the number of entities in a body. Key changes include:

- Use of a unique identifier attached to each cell, face, edge and vertex of a body, and picking of an entity by that identifier. During an operation, only those entities affected by the operation will be changed. This improves the reliability of picking entities in a body, where replaying an operation in the command history may change the topology of the body.
- When picking by label, entities are returned in alphabetic order: with unique body name used to order the set of bodies used, and identifier used to order the set of entities in each body.
- When creating new entities by using the TRANSFORM, LOAD or EXTRACTCELLS commands, the unique body name of entities created is updated from the original by means of a COPYIDENTIFIER. This is generated automatically if undefined, but is explicitly kept in the command history to ensure consistent naming of bodies created.

The **IDVERSION** should remain at the current version unless existing scripts defined in a previous version require the setting to be changed.

The use of **IDVERSION** is intended to improve the reliability of scripting and **REPLAY** between different versions of Opera, but with changes to the ACIS kernel, differences in the numbering scheme may still exist.

Local body numbering was introduced in Opera version 10.5. It uses numbers which are unique to each body for graphical picking in the **SELECT** and **PICK** commands. A body is referred to by its **UNIQUENAME** in local numbering.

To use a command script from an earlier version of the Modeller, it might be necessary to use

**COMPATIBILITY LOCALPICKING=NO**

For this case, **IDVERSION** will be ignored.

**WARNING:** whichever numbering scheme is used, the method in which the numbering is applied cannot be guaranteed to be the same on subsequent runs, so using numbers may mean that the wrong items are selected if re-running a command script. Using **IDVERSION=16** will be the most reliable.

## The CONDUCTOR Command

---

### Summary

Modifies data common to all picked conductors of any type.

### Command line parameters

Command	CONDUCTOR		
Parameter	Default	Function	
OPTION	<i>none</i>	MODIFY	Modifies properties of the picked conductors
		LOAD	Loads defaults from picked conductors
DRIVELABEL		Name for the conductor drive label	
LCNAME		Name for Local Coordinate System for coordinate system 1	
SYMMETRY		Rotational symmetry about global Z axis	
XCEN2		Origin of coordinate system 2	
YCEN2			
ZCEN2			
THETA2		Euler angles defining orientation of coordinate system 2	
PHI2			
PSI2			
RXY		Reflection symmetries in XY, YZ and ZX planes.	
RYZ			
RZX			
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
TOLERANCE		Field calculation tolerance	
INCIRCUIT		Is the conductor part of a circuit:	
		NO	The conductor has defined current density.
		YES	The current in the conductor is determined by a circuit.

Command	CONDUCTOR		
Parameter	Default	Function	
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.	
CIRCUITELEMENT		The name of circuit element this conductor is part of.	
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation	
MODELCOMPONENT		NO	Do not convert to meshable cells.
		REGULAR	Convert to meshable cells which will be regularly meshed.
		YES	Convert to meshable cells which meet the mesh size criteria.
MESHSIZE		The size of the mesh to be used when meshing	
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command operates on the list of picked conductors of any type, and can be used to change common properties such as coordinate systems, symmetries, current density, tolerance etc.

This command cannot be used to change the parameters defining the geometry of the conductors. To modify the geometry of e.g. a solenoid, the **SOLENOID** command should be used.

The default values of the command parameters must be updated to match common values shared by all of the picked conductors. To do this **CONDUCTOR OPTION=LOAD** must be used after picking conductors (see [The PICK Command \[page 277\]](#)) to be modified and before **CONDUCTOR OPTION=MODIFY**.

If the picked conductors do not have a common value for a parameter, that parameter is left unset. **OPTION=MODIFY** will change the conductor data of all of the picked conductors to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on conductor parameters, see [Conductors \[page 512\]](#).

## Conductors in machines

The **GROUPLABEL** parameter is used to set the group to which the conductor is added in a CARMEN analysis.

## Meshed conductors

Meshed conductors can be used in circuits, inductance calculations or QUENCH analysis. Meshable cells can be created from the conductor geometry at the same time as the model body is formed by [The MODEL Command \[page 263\]](#). By default the cells created will be given a material label of the circuit element name. If this is not specified they will be modelled as air.

Three parameters control the meshing of conductors.

- **MODELCOMPONENT=NO:** this conductor will not be meshed.
- **MODELCOMPONENT=YES:** each section of the conductor will be split into one or more cells no larger than **MESHSIZE** in the current flow direction. The maximum cell size in the cross-section of the conductor is further limited to **MESHSIZE\*MESHLFACTOR**. **MESHLFACTOR** must be a positive value greater than 0 and not greater than 1. Small values for **MESHSIZE** and **MESHLFACTOR** may slow the model create stage significantly as many more cells will be formed.
- **MODELCOMPONENT=REGULAR:** one mesh cell will be formed from each section of the conductor. The mesh generator will create regular sized elements using the given **MESHSIZE** in the current flow direction and **MESHSIZE\*MESHLFACTOR** in the cross-section of the conductor. **MESHLFACTOR** must be a positive value greater than 0 and not greater than 1.

For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The CONTOUR Command

---

### Summary

Displays contours of volume properties and boundary conditions in the model.

### Toolbutton



### Command line parameters

Command	CONTOUR		
Parameter	Default	Function	
COMPONENT	NONE	NONE	No contours.
		<i>boundary condition values</i>	Display contours of boundary condition values.
		<i>volume properties</i>	Display contours of volume properties.

### Notes

This command displays contours of the **COMPONENT** on surfaces of objects in the model. Components are boundary conditions and volume properties. If an object has data containing the component, then the value of the data will be contoured over it. Other displayed objects will be displayed normally. The contour command options are based on data that can be set on cells and faces only.

### Boundary conditions

The following contours of boundary condition values:

- **AX, AY, AZ, AMOD:** Total magnetic vector potential
- **CMPOTENTIAL:** Constant magnetic potential
- **CVOLTAGE:** Constant voltage
- **DISPLACEMENTX, DISPLACEMENTY, DISPLACEMENTZ:** Displacements
- **DMPOTENTIAL:** Normal derivative of magnetic potential
- **DVOLTAGE:** Normal derivative of voltage
- **EX, EY, EZ, EMOD:** Total electric field
- **INAX, INAY, INAZ, INAMOD:** Incident magnetic vector potential

- **INEX, INEY, INEZ, INEMOD:** Incident electric field
- **MPOTENTIAL:** Magnetic scalar potential
- **NORMALDISPLACEMENT:** Normal displacement
- **PRESSURE:** Pressure
- **TEMPERATURE:** Temperature
- **TRACTIONX, TRACTIONY, TRACTIONZ:** Surface tractions
- **VOLTAGE:** Voltage

## Volume properties

The following options for **COMPONENT** display volume property values:

- **BODYFORCEX, BODYFORCEY, BODYFORCEZ:** Body forces
- **CHARGE:** Charge density
- **HEATVALUE:** Applied volume heat source
- **JX, JY, JZ, JMOD:** Applied source current density
- **PACKING:** Packing factor
- **ROTATION:** Applied rotational velocity
- **VX, VY, VZ, VMOD:** Applied linear velocity

If the **CONTOUR** command is used to display functional components, the picture on the screen depends on the existence of a surface mesh.

With the surface mesh generated, the display will be based on the mesh that has been formed, and it will show the distribution of the component which is passed on to the analysis module.

If the surface mesh is not formed yet, the display is based on the surface display facets, which may not represent the final mesh very well. In such a case, a large nonlinearity in functional definitions of these values may not be seen over the surface.

Note that the boundary condition components are displayed if they exist, even if the boundary condition is not currently selected. For example, if a user swaps between an **IVECTOR** (incident vector potential) boundary condition with its 3 components and an **IEVECTOR** (incident electric field) boundary condition with its components, all **INA** and **INE** components will be available to the **CONTOUR** command, irrespective of which boundary condition is currently applied to the surface.

## The **COVER** Command

---

### Summary

Creates a sheet face enclosed by edges that have been picked.

### Toolbutton



### Command line parameters

Command	<b>COVER</b>	
Parameter	Default	Function
<b>NAME</b>	Cover	Attaches this name to the body formed
<b>UNIQUE NAME</b>		Force a specific unique name for the block created.

### Notes

The **COVER** command creates sheet faces from a single picked body that contains only wire edges, or from a set of picked edges.

When covering a picked body, the **NAME** and **UNIQUE NAME** parameters are unused, and existing information from the picked body is maintained.

When the sheet face body is created from a set of picked edges, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUE NAME** parameter, this name is used instead.

A picked body and picked edges cannot be covered simultaneously.

The picked edges, or the edges of the body, must form one or more closed loops. If a cover operation is successful, but there are edges that do not form part of the sheet faces created, these edges will be removed.

## The CYLINDER Command

---

### Summary

Creates a cylinder, cone or n-sided prism with either circular or elliptic base.

### Toolbutton



### Command line parameters

Command	<b>CYLINDER</b>		
Parameter	Default	Function	
<b>NAME</b>		Attaches this name to the body formed	
<b>X0</b>		X coordinate of central point on the base	
<b>Y0</b>		Y coordinate of central point on the base	
<b>Z0</b>		Z coordinate of central point on the base	
<b>X1</b>		X coordinate of central point on the top	
<b>Y1</b>		Y coordinate of central point at the top	
<b>Z1</b>		Z coordinate of central point at the top	
<b>MAJORRADIUS</b>		Major radius of the base of the cylinder	
<b>MINORRADIUS</b>		Minor radius of the base of the cylinder	
<b>TOPRADIUS</b>		Radius at the top of the cylinder	
<b>TUBE</b>		<b>NO</b>	Create a solid cylinder
		<b>YES</b>	Create a tube of the specified thickness
<b>THICKNESS</b>		Thickness of the cylinder	
<b>SIDES</b>	2	Create an N-sided prism if >2	
<b>SHAPECONTROL</b>	<b>SIMPLE</b>	<b>CONE</b> <b>GENERAL</b> <b>SIMPLE</b> <b>TUBE</b>	Interface use only. Sets options available in the dialog for defining the variations of cylinder available.

Command	<b>CYLINDER</b>
<b>MATERIALLABEL</b>	Set the material label of the new cylinder.
<b>LEVEL</b>	Set the data storage level of the new cylinder
<b>UNIQUENAME</b>	Force a specific unique name for the cylinder created.

## Notes

The axis of a cylindrical or prismatic body is formed between points **(X0,Y0,Z0)** and **(X1,Y1,Z1)** and the normals of the base and top planes are parallel to this. If these 2 points are the same, a circular sheet face disc is created, normal to the Z axis. The two base radii given must be greater than zero.

If the **TOPRADIUS** is given as zero, the body formed is a cone.

If the **MAJORRADIUS** is not the same as the **MINORRADIUS**, the body formed has an elliptic base.

Additionally, a hollow tube of specified thickness can be given. If a positive thickness is specified, the value of **MAJORRADIUS** represents the outer radius of the tube. If a negative thickness is given, the inner radius of the tube is given by **MAJORRADIUS**. A thickness of zero will create a cylindrical sheet face.

The number of sides can be used to create an N-sided prism. All points on the prism surface will lie at or inside the radius specified by **MAJORRADIUS**. When used with a specified thickness, the prismatic tube will have a maximum thickness of the value specified. If **SIDES=2**, the sides of the cylindrical faces will be split.

The coordinates specified are in the Working Coordinate System.

The single **CELL** created by the **CYLINDER** command is given a material label and data storage level if specified in the **MATERIALLABEL** and **LEVEL** parameters. See "The CELldata Command" on page 168. If the dimensions of the cylinder create a sheet face, the properties specified are attached to the **FACE** created. See "The FACEDATA Command" on page 213.

When a cylinder is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

## The **DBCASEDATA** Command

---

### Summary

Sets the list of multiple simulation values: scaling factors for statics, frequencies for steady-state AC and output times for transient analyses.

### Toolbutton

#### Settings

### Command line parameters

Command	<b>DBCASEDATA</b>	
Parameter	Default	Function
<b>PROGRAM</b>	<i>none</i>	Identifies the solver and the type of data required:
		<b>CARMEN</b> Output times for Motional EM
		<b>DEMAG</b> Output times for Magnetization
		<b>ELEKTRASS</b> Frequencies for Harmonic EM
		<b>ELEKTRATR</b> Output times for Transient EM
		<b>MULTIPHYSICSACTIVESTAGE</b> Multiple simulation values for <b>STAGE</b> of a multiphysics analysis
		<b>QUENCH</b> Output times for Quench
		<b>SCALA</b> Scaling factors for Charged Particle
		<b>SOPRANOS</b> Frequencies for Harmonic HF
		<b>TEMPOTR</b> Output times for Tansient Thermal
		<b>TOSCAELEC</b> Scaling factors for Electrostatics
		<b>TOSCAMAGN</b> Scaling factors for Magnetostatics

<b>OPTION</b>	<b>ADD</b>	<b>ADD</b>	Add a new scaling factor / frequency / output time to the list
		<b>CLEAR</b>	Clear the list of scaling factors / frequencies / output times
		<b>DELETE</b>	Delete an item from the list
		<b>INSERT</b>	Insert a scaling factor / frequency / output time at a position in the list
		<b>LIST</b>	Lists the set of scaling factors / frequencies / output times.
		<b>REPLACE</b>	Replace an item in the list
<b>INDEX</b>	1	The position at which to insert, delete or replace	
<b>VALUE</b>		The new value for add, insert or replace	
<b>STAGE</b>	1	Stage number in a multiphysics analysis.	

## Notes

This command creates a list of scaling factors, frequencies or output times for the different analysis programs. Each program has a separate list of values.

- Magnetostatic, Electrostatic and Current Flow and Charged Particle simulations will be analysed for each scaling factor in the list. Scaling factors will be applied to a single named drive or all named drives associated with current densities and assigned potential and normal field boundary conditions. The choice of the named drive can be made using [The ANALYSISDATA Command \[page 122\]](#).
- Harmonic simulations will be analysed for each of the frequencies given in the list.
- Transient solvers will output a solution at each of the times in the list. The output time list is automatically chronologically ordered, so that insert and add have the same effect. An output time at t=0 is also added.

If **SIMPLE** fixed time-stepping has been selected in [The ANALYSISDATA Command \[page 122\]](#), output times which are not an integer multiple of the initial time-step will be ignored. Additional output times can be defined and the time-step adjusted using a command file (see [Command File Control \[page 629\]](#)).

**VALUE** can be used to **ADD** single values or multiple equally spaced values over a range. Multiple values should be specified using the syntax **value=start;finish;increment**, i.e. by specifying 3 values or expressions separated by ";".

Duplicate entries are automatically removed from the list.

## Multiphysics

In a multiphysics analysis (see [The MULTIPHYSICS Command \[page 269\]](#)), several analysis stages can be defined before any of them is run. The **STAGE** parameter should be used to identify the stage number for which the parameters are being set. There are some restrictions on the use of simulation values in multiphysics:

- Multiple values can only be defined for the first stage of a multiphysics analysis. In this case, multiple values imply a loop over all subsequent analyses in the sequence. For example, a sequence of Harmonic EM with  $n$  multiple frequencies followed by Static Thermal and Static Stress would result in  $3n$  simulations; each Steady-State EM analysis at a new frequency would be followed by Static Thermal and Static Stress analyses.
- Transient analyses can only be used for statics analysis with one output at time zero.
- Single value can be defined for other stages but only for the first analysis of any type in the sequence.

## The **DELETE** Command

---

### Summary

Deletes the picked entities.

### Toolbutton



### Command line parameters

Command	<b>DELETE</b>		
Parameter	Default	Function	
<b>REGULARIZE</b>	<b>YES</b>	<b>YES</b>	Regularize the result. Removes unnecessary edges after the deletion of faces
		<b>NO</b>	Do not regularize the result
<b>EXTERNAL</b>	<b>NO</b>	<b>YES</b>	Allow external faces to be deleted. De-features the geometry and extends the neighbouring faces to fill the gap
		<b>NO</b>	Do not delete external faces

### Notes

This command tries to **DELETE** all picked objects, including Local Coordinate Systems and conductors.

If bodies have been picked, these will always be deleted. If cells have been picked, the delete command will remove the cells from the body and delete them. If a body only has a single cell, deleting the cell has the same effect as deleting the body.

If a face to be deleted is internal to a body, or if the face does not form part of a cell, i.e. both sides are external, the face can be deleted. If the face is internal, the two cells to either side of the face will be merged. If regularization is requested, the edges and vertices that formed the boundary of the face are tested. If they are not required to support the geometry, they are deleted as well.

Faces forming the external boundaries of a body can be deleted only if the gap formed can be patched by the neighbouring faces. Therefore, one side of a cube could not be deleted because the neighbouring faces would be unable to patch it. The parameter **EXTERNAL=YES** must be used to allow these faces to be deleted.

Edges and vertices may be deleted if they are not required to support the topology of the body, e.g. if the face to both sides of an edge has the same underlying geometric surface.

## The **DRIVE** Command

---

### Summary

Sets the properties associated with a drive label.

### Toolbutton



### Command line parameters

Command	<b>DRIVE</b>	
Parameter	Default	Function
<b>OPTION</b>		PICK Adds a drive label to a list to be set
		UNPICK Clears the list of picked drive labels
		RESET Clears the data from the picked drive labels
		MODIFY Sets the data for the picked drive labels
		DELETE Deletes the picked labels
<b>DRIVELABEL</b>		Drive label to be picked
<b>TYPE</b>		Drive function associated with the label
		COSINE Cos drive
		DC Constant for all time
		FUNCTIONAL Function specified by <b>FUNCTION</b> .
		PEAK Rise to peak value before decaying
		RAMP Ramp drive
		RISE Exponential rise function
		SINE Sine drive
		STEP Step at time zero
		TABLE Switch on table
		TOFF Switch off table
<b>SSPHASE</b>		Phase lag used during harmonic analyses

Command	<b>DRIVE</b>	
Parameter	Default	Function
<b>SINFREQUENCY</b>		Frequency for Sine drives
<b>SINPHASE</b>		Phase for Sine drives
<b>COSFREQUENCY</b>		Frequency for Cosine drives
<b>COSPHASE</b>		Phase for Cosine drives
<b>STEPTIME</b>		This parameter is not used
<b>RAMPTIME</b>		Time taken to ramp
<b>PEAKTIME</b>		Time at which peak occurs
<b>RISETIME</b>		Time constant for the rise
<b>TABLEFILE</b>		File for switch on timetable information
<b>TABLEOFFFILE</b>		File for switch off timetable information
<b>ROTATING</b>		Not used at present
<b>SCALE</b>		Drive scaling factor
<b>FUNCTION</b>		Algebraic expression for functional drive

## Notes

This command defines the drive functions for use by the analysis programs:

- transient drives (see [Transient Driving Functions \[page 624\]](#));
- phase angles (the drive function is  $\cos(2\pi ft - \phi)$ , where  $f$  is the frequency set in [The DBCASEDATA Command \[page 197\]](#) and  $\phi$  the phase angle in degrees);
- scaling factors.

A set of drive labels is picked using the command repeatedly, with **OPTION=PICK** and a **DRIVELABEL** specified. A drive label can be removed from the set using **OPTION=UNPICK**. If no **DRIVELABEL** is given, the set is emptied.

Issuing the command with **OPTION=MODIFY** will modify the properties of the set of picked drive labels to the new values given in the parameters. The value of a property associated with the drive labels is unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked volume labels. If the data of one of these parameters is unset, or the picked drive labels do not share the same value, then the parameter value is left clear.

**OPTION=RESET** will clear the properties associated with all of the picked drive labels.

The properties of all drive labels can be listed using **OPTION=LIST**.

Drive labels that are not used, i.e. have no boundary condition or conductor referencing them, can be deleted using **OPTION=DELETE**. Deleting a drive label that is in use will reset its properties.

Harmonic EM and Harmonic HF use the **SSPHASE** parameter information. If this is unset, it is assumed to be 0.

Magnetostatic, Harmonic EM, Velocity EM and Harmonic HF use the **SCALE** parameter. If this is unset, it is assumed to be 1. Scaling is applied after the fields have been calculated. The conductor tolerance should, therefore, relate to the unscaled field values.

Transient EM, Motional EM and Magnetization drives are mainly controlled by the **TYPE** parameter. Which of the other parameters are used is determined from the option assigned to type. If unset, a **TYPE=DC** is assumed. The drive value is calculated at each time during the transient analysis and is multiplied by the **SCALE** factor. If unset, the **SCALE** factor has a value of 1.

See [Labels \[page 238\]](#) for more information on valid labels.

## The EDGEDATA Command

---

### Summary

Sets properties of picked edges.

### Toolbutton



### Command line parameters

Command	EDGEDATA		
Parameter	Default	Function	
<b>OPTION</b>	<b>MODIFY</b>	<b>MODIFY</b>	Applies new values to the picked edges
		<b>RESET</b>	Clears all data from the picked edges
<b>SIZE</b>	See notes	Mesh control size	
<b>LEVEL</b>	See notes	Data storage level for the edge data	

### Notes

This command is used to set or clear the properties of all picked edges.

Edges initially have no data assigned to them.

If issuing the command with **OPTION=MODIFY**, new values of parameters that have been set replace the existing values of data on the edges. The data on the edges, associated with any unset parameters, is left unchanged.

The default value of each of the parameters is updated to be the common value of all the picked edges. If the data of one of these parameters is unset, or the picked edges do not share the same value, then the parameter value is left clear.

**SIZE** is the maximum length of an element side along the edge.

The **LEVEL** parameter controls the storage of data when there is conflict during the merging of multiple edges. The data set with the greater level will be maintained. The result of merging 2 edges with the same level is indeterminate.

Upon issuing the command with **OPTION=MODIFY**, all picked items are deselected. Issuing the command with **OPTION=RESET** keeps the same set of picked objects.

## The **EMITTER** Command

---

### Summary

Sets up properties for an emitter surface or volume.

### Toolbutton



### Command line parameters

Command	EMITTER	
Parameter	Function	
OPTION	ADD	Adds a new emitter to the model.
	MODIFY	Modifies the emitter properties.
	DELETE	Deletes the emitter from the model.
NAME	The name of the emitter to ADD, MODIFY or DELETE.	
LABEL	The name of the boundary or volume to hold the emitter. Required for ADD.	
LABELTYPE	BOUNDARY	LABEL is a boundary.
	VOLUME	LABEL is a volume.
ACTIVE	YES	Emitter is active.
	NO	Emitter is inactive.
TYPE	Sets the type of the emitter.	
	PRIMARY	Emitter will be primary.
	SECONDARY	Emitter will be secondary.

Command	<b>EMITTER</b>	
Parameter	Function	
<b>ETYPE</b>	The primary emitter type. See <a href="#">Charged Particle Solver [page 548]</a> for details on the different types.	
AUTOMATIC		Automatic selection <b>FOWLER</b> , <b>SCHOTTKY</b> or <b>EXTSCHOTTKY</b> .
CHILD'S		Child's law current limit.
CURRENTDENSITY		Specified current density.
EXTSCHOTTKY		Extended Schottky field effect emission.
FOWLER		Fowler Nordheim field effect emission.
LANGFRY		Lamgmuir-Fry current limit.
LANGFRY_VC		Lamgmuir-Fry current limit (virtual cathode).
MAXWELL		Maxwell velocity sampling.
PLASMA103		Plasma free surface model.
SCHOTTKY		Schottky field effect emission.
THERMAL		Thermal saturation limit.
USERDEFINED		User defined beamlets (position, velocity and current).
<b>STYPE</b>	The secondary emitter type. See <a href="#">Charged Particle Solver [page 548]</a> for details on the different types.	
ADVBACKSCATTERED		Use advanced back-scattered emission.
ADVTRUESECONDARY		Use advanced true secondary emission.
BACKSCATTERED		Use back-scattered emission.
PLASMA		Use back-scattered emission.
TRUESECONDARY		Use true secondary emission.
<b>PARTICLE</b>	The emission particle type:	
ELECTRON		Electrons.
OTHER		Specified mass and charge.
PROTON		Protons.
<b>MASS</b>	The mass of the <b>OTHER</b> particle.	
<b>CHARGE</b>	The charge of the <b>OTHER</b> particle.	
<b>LENGTH</b>	The maximum distance between neighbouring rays in cm.	



Command	<b>EMITTER</b>
Parameter	Function
<b>SECMINENERGY</b>	Lower energy limit on secondary emission particles.
<b>SECMAXENERGY</b>	Upper energy limit on secondary emission particles.
<b>SECMINTHETA</b>	Lower $\theta$ limit on secondary emission particles.
<b>SECMAXTHETA</b>	Upper $\theta$ limit on secondary emission particles.
<b>SECMINPHI</b>	Lower $\phi$ limit on secondary emission particles.
<b>SECMAXPHI</b>	Upper $\phi$ limit on secondary emission particles.
<b>IONIZATIONENERGY</b>	Ionization energy of plasma atoms.
<b>IONIZATIONFRACT</b>	Fraction of loss that produces ionization.
<b>PLASMACURRENT</b>	Total plasma current.
<b>SAMPLES</b>	Number of velocity samples for <b>ETYPE=LANGFRY_VC</b> .

## Notes

This command defines the emitter characteristics for use by SCALA.

Emitters are attached to existing labelled boundaries or volumes<sup>1</sup>. When using **OPTION=ADD**, the emitter called **NAME** is attached to an entity of type **LABELTYPE** with the given **LABEL**.

If **LABELTYPE=BOUNDARY**, the **LABEL** identifies one of more faces in the model. The emission direction is defined as follows.

- Faces on the outside of the mesh: particles will be emitted towards the interior of the mesh.
- Interior faces of the mesh: particles will be emitted towards volumes of **AIR** or volumes with the **BEAMPASS** label. If both directions are possible, the direction chosen will be the normal direction of the face. The normal direction of surfaces can be "toggled" to ensure that all surfaces with a particular label are emitting in the same direction. See [The VECTOR Command \[page 339\]](#).)

Not all parameters are required for each emitter type. For details on which parameters are required for each emitter type, see [Charged Particle Solver \[page 548\]](#). If any unrequired parameters are given, they will not be used.

Several backscattering and secondary emission parameters (**SCATYIELD**, **SCATLOSS**, **SECYIELD**) are dimensionless when applied to surface emitters and have units of 1/cm with volume emitters.

Many parameters do not have a default value and a suitable value should be given (if required for the selected emitter type).

Emitters can be removed from the model with **OPTION=DELETE**. The emitter properties will not be retained.

---

<sup>1</sup>The use of volume emitters is a separately licensed feature.

## The **END** Command

---

### Summary

Ends the current Modeller session.

### Command line parameters

Command	<b>END</b>
No parameters	

### Notes

This command ends the current session of the Modeller. If there have been changes since the last **SAVE OPTION=ALL** or **SAVE OPTION=NEW** command, then confirmation of the end command will be required.

## The **EXPORT** Command

### Summary

Exports conductor data to a file for use by the Modeller, Pre or Post-Processor.

### Toolbutton



### Command line parameters

Command	<b>EXPORT</b>	
Parameter	Default	Function
<b>FILE</b>		Conductor file to be created

### Notes

The **EXPORT** command writes a file containing all conductors in the model. This file can be read by the Pre-Processor using [The READ Command \[page 481\]](#), the Post-Processor using [The CONDUCTOR Command \[page 727\]](#), and the Modeller using [The IMPORT Command \[page 235\]](#).

The conductor data in the file contains the values of the origin position and Euler angles defining coordinate system 1, not the name of the Local Coordinate System with which the conductor is associated.

## The **EXTRACTCELLS** Command

### Summary

Extracts picked cells or cells from picked bodies.

### Toolbutton



### Command line parameters

Command	<b>EXTRACTCELLS</b>	
Parameter	Default	Function
<b>COPYIDENTIFIER</b>		Allows adjustment of the unique name of bodies.

### Notes

This command will extract picked cells from the owning body. The cells will be deleted from the existing body and a new body will be created for each cell.

Where a body has been picked, each cell in the body will be extracted into a new single celled body.

Where a body has only one cell, the operation will have no effect.

Each body created will have the name of the parent body, and a unique name created from the unique name of the parent body and the **COPYIDENTIFIER** specified.

The **COPYIDENTIFIER** parameter allows bodies created during the operation to be given a new unique body name that is unlikely to conflict with other bodies present. If unset, a new identifier is generated from existing bodies to prevent any possible conflict.

## The **FACEDATA** Command

---

### Summary

Sets properties of picked faces.

### Toolbutton



### Command line parameters

Command	<b>FACEDATA</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>MODIFY</b>	<b>MODIFY</b>	Applies new values to the picked faces
		<b>RESET</b>	Clears all data from the picked faces
<b>BOUNDARYLABEL</b>	See notes	Boundary label on the face	
<b>ELEMENTTYPE</b>	See notes	<b>LINEAR</b>	Elements touching the face will be linear
		<b>QUADRATIC</b>	Elements touching this face will be quadratic
<b>SIZE</b>	See notes	Mesh control size	
<b>NORMALTOL</b>	See notes	Maximum normal angle between mesh nodes	
<b>SURFACETOL</b>	See notes	Maximum deviation of the mesh from the surface	
<b>LEVEL</b>	See notes	Sets the data storage level for the face data	
<b>FORMETHOD</b>	<b>NONE</b>	Layering method for forward layers from the face:	
		<b>GEOMETRY</b>	Create additional faces and copy the mesh.
		<b>MESH</b>	Form layers of elements in a hexahedral or prism mesh.
		<b>NONE</b>	No layering.
<b>FORLAYERS</b>	0	Number of forward layers from the face	
<b>FOROFFSET</b>		Distance between each forward layer	

<b>BACKMETHOD</b>	<b>NONE</b>	Layering method for backward layers from the face:
		<b>GEOMETRY</b> Create additional faces and copy the mesh.
		<b>MESH</b> Form layers of elements in a hexahedral or prism mesh.
		<b>NONE</b> No layering.
<b>BACKLAYERS</b>	0	Number of backwards layers from the face
<b>BACKOFFSET</b>		Distance between each backward layer

## Notes

This command is used to set or clear the properties of all picked faces.

Faces initially have no data assigned to them, and **OPTION=RESET** will clear all data of the picked faces.

Issuing the command with **OPTION=MODIFY** will modify the properties of the picked faces to the values given in the parameters. The value of properties on the picked faces are unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked faces. If the data of one of these parameters is unset, or the picked faces do not share the same value, then the parameter value is left clear.

Upon issuing the command with **OPTION=MODIFY**, the list of picked items is cleared. Issuing the command with **OPTION=RESET** keeps the same set of picked objects.

The **BOUNDARYLABEL** parameter allows a boundary label to be attached to a face. The properties of such a label are set or modified using the **BOUNDARY** command.

The **ELEMENTTYPE** parameter allows the user to force the use of quadratic elements when creating the database. Any elements that touch the face will become quadratic if the database is created with mixed surface elements. See [The SOLVERS Command \[page 309\]](#) for greater detail.

The **SIZE**, **NORMALTOL** and **SURFACETOL** parameters control the mesh size of elements on and near the face, when generating the surface and volume mesh.

The **LEVEL** parameter controls the storage of data when there is conflict during the merging of multiple faces. The data set with the greater level will be maintained. The result of merging 2 faces with the same level is indeterminate.

## Layering

Layering should help improve the quality of the mesh generated in very thin layers, and should make the modelling of thin surface regions easier. Layering affects the distribution of elements adjacent to a face. Two methods are available:

- **GEOMETRY** creates copies of the face, where possible<sup>1</sup> using an identical mesh on each layer. Any planar, spherical or cylindrical face may be layered. Cylindrical or spherical faces will be scaled by radius. The additional faces are introduced when the model body is created (see [The MODEL Command \[page 263\]](#)). Consequently, **GEOMETRY** layering should only be used where absolutely necessary; the additional faces required will increase the time and complexity of the operations needed to form the model body.  
Where interaction of the layers with other parts of the model occur, the original face will be modified to ensure that the face and its layered copies have a matching topology. Layers that extend outside the model body will be removed.
- **MESH** can be applied to any face of a cell which can be meshed using hexahedra or to the "end" faces of a cell which can be meshed with prisms. It affects the working of the mesh generator to form layers of elements without introducing additional faces.

For both methods, the parameters set

- the number of layers in front of and behind the original face (**FORLAYERS** and **BACKLAYERS**); and
- the offset between each layer (**FOROFFSET** and **BACKOFFSET**). The values may be a function of the system variable **LAYER**, e.g.

**forlayers=3, foroffset=2^layer**

would produce 3 copies of the face with exponentially increasing gap thickness (2, 4, 8) between each layer.

---

<sup>1</sup>It is not possible to use an identical mesh on the layers when the cells either side of the face have different element types.

## The **FILL** Command

---

### Summary

Fills the model with a mesh of volume elements.

### Toolbutton



### Command line parameters

Command	<b>FILL</b>	
Parameter	Default	Function
<b>TOLERANCE</b>		Tolerance used for checking for equivalent points

### Notes

This command generates the volume mesh and must be used after the **MESH** command and before creating the database. See [The MESH Command \[page 257\]](#) for a discussion of the different element types.

The **TOLERANCE** parameter should not need to be adjusted in most cases, and can control the tolerance used for matching equivalent points.

The cells inside the model body are meshed sequentially. On execution of the **FILL** command, the outline frame of the volume which is being processed is high-lighted. A progress bar is displayed at the bottom of the Modeller window.

This command may take a long time for complex models or large meshes.

### System variables

On successful completion of the volume mesh, the program sets the following system variables:

- **MESH** to 2
- **NODES** to the number of nodes
- **ELEMENTS** and **VOLUMEELEMENTS** to the number of volume elements

### Volume mesh errors

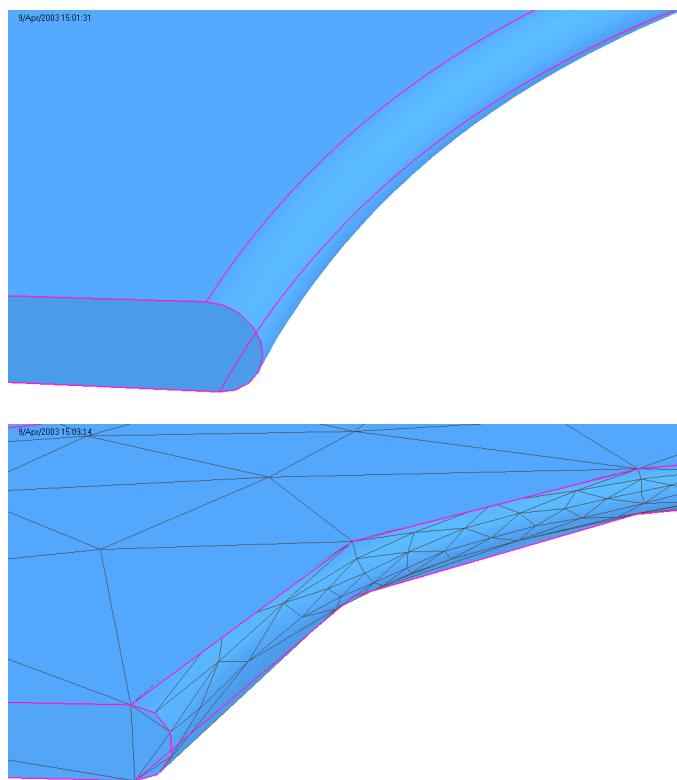
There are several types of failure that may be reported by the volume mesh generator. Unfortunately the error number and its associated message cannot in general be related to the source of

the problem.

### Typical source of the problem

An inappropriate or poor surface mesh is often the reason why the volume mesh generation fails. When the program fails to mesh a cell, the cell will be coloured and given the label **VOLUME MESH ERROR**, which can then be selected for viewing. The following checklist should be used to identify likely problems.

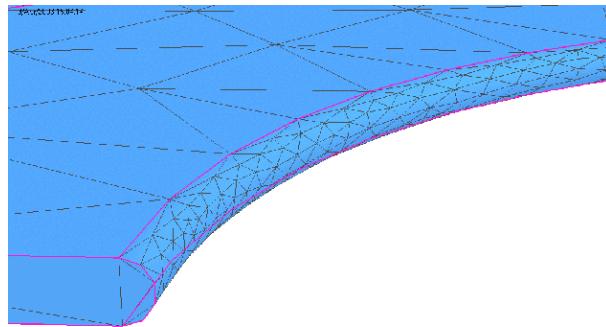
- View the surface mesh on the cell that failed to mesh
  - Look for badly shaped triangles and consider changing the **SIZE** control parameters to improve the surface mesh.
  - Pay particular attention to geometric details on the surface of the cell that failed to mesh. A common cause of problems is a detail with a tight curvature, attached to an adjoining face by a curved edge, shown in [Figure 3.4](#).



*Figure 3.4 The top picture shows a curved blended edge on a thin plate. The surface mesh for this detail is shown in the lower picture.*

This is a typical problem created by using only the normal tolerance to control mesh size, where the blend surface is being discretized with much smaller elements compared to the

curved edge of the plate. The problem is easily avoided by using surface fitting tolerance to control the mesh size on the blend, see [Figure 3.5](#).



*Figure 3.5 Improved mesh using surface fitting tolerance to control the mesh size on the blend surface*

- Look for areas with unexpectedly small elements. The surface mesh generator will create a dense mesh around very small area faces. When this happens it is often a result of small errors in the dimensions of the model.
- Thin shells with curved faces can cause problems because the meshes on two faces intersect each other. The surface fitting tolerance can be used to avoid this problem, its value should be of the order of the thin dimension of the cell.
- Check the size of the cell compared to the surface mesh size.
  - Numerical precision becomes a problem if the element edge sizes vary by more than two orders of magnitude over the surface of the cell. Cut the cell into smaller volumes to improve the performance of the mesh generator.

## Specific problems

The following error message:

Unable to allocate memory required

means that a mesh with a large number of elements was created and no more memory was available for storage. The mesh size controls applied to the model are probably not realistic. The maximum number of elements that can be created with a 2GByte addressing limit is approximately 8 million.

## The **FILTER** Command

### Summary

Sets a filter for graphical selection.

### Toolbuttons



### Command line parameters

Command	<b>FILTER</b>	
Parameter	Default	Function

<b>TYPE</b>	<b>NONE</b>	Type of geometric entity to be selected:
		<b>BODY</b> Bodies
		<b>CELL</b> Cells
		<b>CONDUCTOR</b> Conductors
		<b>EDGE</b> Edges
		<b>FACE</b> Faces
		<b>LCS</b> Local coordinate systems
		<b>NONE</b> No graphical selection
<b>COMMAND</b>	<b>PICK</b>	The effect of selecting by cursor:
		<b>PICK</b> Select the entity
		<b>CURSORHIDE</b> Hide an entity selected by the cursor
		<b>HIDE</b> Hide an entity.
		<b>LIST</b> List the properties of the entity.
<b>DISPLAY</b>	<b>VISIBLE</b>	Which vertices are selectable:
		<b>VISIBLE</b> Any visible vertex can be selected.
		<b>EXPLICIT</b> Only vertices explicitly selected for display (e.g. by the size property) can be selected.

## Notes

The filter command controls the graphical interaction with the Modeller. Setting the **TYPE** parameter controls the type of entity that can be selected using the cursor.

The **COMMAND** parameter controls the effect of double-clicking when an object is high-lighted. The cursor shape also indicates which **COMMAND** selected. The **DISPLAY** parameter can be used for extra control of the selection of items.

- **COMMAND=PICK** allows objects to be added to the list of picked entities by issuing a **PICK** command. Cursor: +
- **COMMAND=HIDE** hides the object from the display by issuing a **SELECT OPTION=HIDE** command.
- **COMMAND=CURSORHIDE** hides the object from the display by issuing a **SELECT OPTION=N=CURSORHIDE** command. Cursor: ↗
- **COMMAND=LIST** generates a list showing details of the object and attached data. Cursor: ↘

The **FILTER** command is ignored when either of the **SELECT** options **AUTOUPDATE=NO** or **AUTOUPDATE=CLOSEWINDOW** has been set. See "The **SELECT** Command" on page 299.

This command does not generate an entry in the history stream, and hence produces no **UNDO** or **REDO** command state.

When picking is active and the outline of an entity is high-lighted, single characters can be typed on the keyboard to change the picking filter type and command. The characters can be upper or lower case. A complete list of these and other keyboard shortcuts for use in the Modeller can be found in the section "Keyboard Shortcuts" on page 120.

## The **FITTEDCPE** Command

---

### Summary

Create or modify fitted constant perimeter end (CPE) conductors.

### Toolbuttons



### Command line parameters

Command	<b>FITTEDCPE</b>		
Parameter	Default	Function	
OPTION	NEW	NEW	Creates a new fitted CPE conductor
		MODIFY	Modifies properties of the picked fitted CPE conductors
		LOAD	Loads defaults from picked conductors
DRIVELABEL		Name for the drive label	
LCNAME		Name for Local Coordinate System for coordinate system 1	
SYMMETRY		Rotational symmetry about global Z axis	
XCEN2		Origin of coordinate system 2	
YCEN2			
ZCEN2			
THETA2		Euler angles defining orientation of coordinate system 2	
PHI2			
PSI2			
RXY		Reflection symmetries in XY, YZ and ZX planes	
RYZ			
RZX			
ALPHA		Azimuthal angular position of the straight	
BETA		Cutter angle	

Command	<b>FITTEDCPE</b>		
Parameter	Default	Function	
<b>WIDTH</b>		Radial cross-sectional size	
<b>THICKNESS</b>		Azimuthal cross-sectional size	
<b>H1</b>		Half-length of the straight	
<b>R1</b>		Radius of forming cylinder	
<b>R2</b>		Radius of cross-over arc	
<b>CURD</b>		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
<b>TOLERANCE</b>		Field calculation tolerance	
<b>INCIRCUIT</b>		Is the conductor part of a circuit:  NO      The conductor has defined current density.  YES     The current in the conductor is determined by a circuit.	
<b>REVERSE</b>		Reverse the connections to this conductor in its circuit: YES or NO.	
<b>CIRCUITELEMENT</b>		The name of circuit element this conductor is part of.	
<b>GROUPLABEL</b>		The group of which the conductor is a part for a Motional EM simulation	
<b>MODELCOMPONENT</b>		NO	Do not convert to meshable cells.
		REGULAR	Convert to meshable cells which will be regularly meshed.
		YES	Convert to meshable cells which meet the mesh size criteria.
<b>MESHSIZE</b>		The size of the mesh to be used when meshing	
<b>MESHLFACTOR</b>		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
<b>KEEP</b>	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new fitted CPE conductor or modifies existing fitted CPEs.

- **OPTION=NEW** creates a new fitted CPE conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked fitted CPEs. If the picked fitted CPEs do not have a common value for a

parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking fitted CPEs (see [The PICK Command \[page 277\]](#)) to be modified and before **FITTEDCPE OPTION=MODIFY**.

- **OPTION=MODIFY** changes the conductor data of all of the picked fitted CPEs to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Constant Perimeter Ends \[page 520\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **GUIOPTIONS** Command

---

### Summary

Set console visibility and other window options.

### Command line parameters

Command	GUIOPTIONS	
Parameter	Function	
<b>OPTION</b>	Action of command	
	<b>LOAD</b>	Used by the GUI.
	<b>SET</b>	Update the values.
<b>CONSOLEVIEW</b>	Console visible: <b>YES</b> or <b>NO</b>	
<b>CONSOLEDOCK</b>	Console docked: <b>YES</b> or <b>NO</b>	
<b>CONSOLEBUFFER</b>	Number of lines saved in the console.	
<b>PRINT3DCARD</b>	Print from graphics card: <b>YES</b> or <b>NO</b>	
<b>WINDOWWIDTH</b>	Width of window in pixels.	
<b>WINDOWHEIGHT</b>	Height of window in pixels.	

### Notes

The **GUIOPTIONS** command can be used to change the options which affect the appearance and behaviour of the GUI window. Default settings are loaded from and changes are saved in the registry<sup>1</sup>. Saving the window size and position is optional - see below.

- Console - the command input and text output console can be visible or hidden (**CONSOLEVIEW**) and when it is visible it can be docked (at the bottom of the main window) or undocked (a separate window) (**CONSOLEDOCK**). The scrolling buffer is limited to a number of lines (**CONSOLEBUFFER**).
- Printing - the option to print the image stored in the graphics card is preferable with most graphics card (**PRINT3DCARD**) but can be changed if necessary. The alternative copies the image from the screen display.

---

<sup>1</sup>On Linux systems, registry values are saved in files in the directory  
**~/config/VectorFields**.

- The **WINDOWWIDTH** and **WINDOWHEIGHT** can be set to ensure a consistent size when creating pictures (see [The PICTURE Command \[page 281\]](#) and [The PRINT Command \[page 286\]](#)). These parameters are only available from the command line.

The console window, showing the commands issued to the software, may be displayed by right clicking anywhere in the tab area. On the resulting context sensitive menu, select **Show/Hide** items and click on the tick box for **Console**.

## The HEATTRANSFER Command

---

### Summary

Calculate an approximate heat transfer coefficient for the cylindrical surface of the rotor of a machine.

### Command line parameters

Command	HEATTRANSFER	
Parameter	Default	Function
<b>OPTION</b>		Action of the command:
		<b>LOAD</b> Load default values for the parameters of the command. (Modeller only)
		<b>SET</b> Calculate the heat transfer coefficient from the parameter values.
		<b>UNITS</b> Set the <b>UNITS</b> for the command to one unit sets listed below.
<b>LABEL</b>		The boundary condition label.
<b>RADIUSIN</b>		Inner radius of the machine air-gap.
<b>RADIUSOUT</b>		Outer radius of machine air-gap.
<b>RPM</b>		Rotation speed [rpm].
<b>MASSFLOW</b>		Coolant flow rate.
<b>PRESSURE1</b>	101325	Pressure at coolant inlet.
<b>TEMPERATURE1</b>	273	Coolant inlet temperature [K].
<b>TEMPERATURE2</b>	273	Coolant outlet temperature [K].
<b>HCAPACITY</b>	1.005	Specific heat capacity of coolant at constant pressure.
<b>KAPPA</b>	0.0243	Thermal conductivity of coolant.
<b>GASCONSTANT</b>	287.058	Specific gas constant of coolant.
<b>VISCOSITY</b>	1.983E-05	Dynamic viscosity of coolant.
<b>MDENSITY</b>	1.205	Mass density of coolant.

Command	HEATTRANSFER	
Parameter	Default	Function
SCENARIO		Cooling scenario:  CylinderNaturalCooling   Natural cooling.  CylinderForcedCooling   Coolant forced through the air-gap.
UNITS	METRE	Unit set to be used for the command:  CGS   cm, g, joule, watt, pascal INCH   inch, kg, joule, watt, pascal METRE   m, kg, joule, watt, pascal MICRON   μm, kg, joule, watt, pascal MM   mm, kg, joule, watt, pascal TRUECGS   cm, g, erg, erg/s, dyne/cm <sup>2</sup>

## Notes

The HEATTRANSFER command calculates the heat transfer coefficient for a cylindrical surface of a machine rotor, rotating in a larger cylindrical space. The formulae used for the calculation have been collated from theory and experiment over different regimes of dimensions, rotation speeds and coolant properties. While there is no guarantee of accuracy, the coefficients given provide good initial estimates and avoid the need for computational fluid dynamics before thermal analysis can proceed.

Two SCENARIOS have been provided: CylinderNaturalCooling and CylinderForcedCooling. Both scenarios use:

- the radii of the rotating cylinder (RADIUSIN) and of the outer surface of the gap between rotor and stator (RADIUSOUT);
- the rotation speed; and
- the thermal conductivity (KAPPA), dynamic VISCOSITY and mass density (MDENSITY) of the coolant.

In addition, the CylinderForcedCooling scenario uses:

- the mass flow rate (MASSFLOW);
- the inlet pressure (PRESSURE1);
- the inlet and outlet temperatures (TEMPERATURE1, TEMPERATURE2); and
- the specific heat capacity (HCAPACITY) and specific gas constant (GASCONSTANT) of the coolant.

All parameters must be given in the units indicated in the parameter table above. The default values describe dry air at standard temperature and pressure.

The result of the calculation becomes the default value for the HEATTRANSFER parameter of "The BOUNDARY Command" on page 150 and is saved as system variable HTCOEF.

## Units

The **HEATTRANSFER** command can operate in 6 different unit sets. When the **OPTION** to change the units is used (e.g. **OPTION=UNITS**, **UNITS=TRUECGS**), the values of the other parameters will be scaled so they represent the same quantities in the new unit set.

## The **HELICALEND** Command

---

### Summary

Create or modify helical end conductors.

### Icon



### Toolbuttons



### Command line parameters

Command	<b>HELICALEND</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>NEW</b>	<b>NEW</b>	Creates a new helical end conductor
		<b>MODIFY</b>	Modifies properties of the picked helical end conductors
		<b>LOAD</b>	Loads defaults from picked conductors
<b>DRIVELABEL</b>		Name for the drive label	
<b>LCNAME</b>		Name for Local Coordinate System for coordinate system 1	
<b>SYMMETRY</b>		Rotational symmetry about global Z axis	
<b>XCEN2</b>		Origin of coordinate system 2	
<b>YCEN2</b>			
<b>ZCEN2</b>			
<b>THETA2</b>			
<b>PHI2</b>		Euler angles defining orientation of coordinate system 2	
<b>PSI2</b>			

Command	HELICALEND	
Parameter	Default	Function
RXY		Reflection symmetries in XY, YZ and ZX planes
RYZ		
RZX		
ALPHA		Azimuthal angular position of the straight
BETA		Cutter angle
WIDTH		Radial cross-sectional size
THICKNESS		Azimuthal cross-sectional size
H1		Half-length of the straight
H2		Length of conductor
R1		Radius of forming cylinder
R2		Radius of cross-over arc
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>
TOLERANCE		Field calculation tolerance
INCIRCUIT		Is the conductor part of a circuit:
		NO      The conductor has defined current density.
		YES     The current in the conductor is determined by a circuit.
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.
CIRCUITELEMENT		The name of circuit element this conductor is part of.
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation
MODELCOMPONENT		NO      Do not convert to meshable cells.
		REGULAR   Convert to meshable cells which will be regularly meshed.
		YES     Convert to meshable cells which meet the mesh size criteria.
MESHSIZE		The size of the mesh to be used when meshing
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.

Command	<b>HELICALEND</b>		
Parameter	Default	Function	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new fitted helical-end conductor or modifies existing fitted helical-ends.

- **OPTION=NEW** creates a new fitted helical-end conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked fitted helical-ends. If the picked fitted helical-ends do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking fitted helical-ends (see [The PICK Command \[page 277\]](#)) to be modified and before **HELICALEND OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked fitted helical-ends to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Helical Ends \[page 519\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **HIDE** Command

---

### Summary

Hides picked entities.

### Command line parameters

Command	HIDE
No parameters	

### Notes

The **HIDE** command hides all picked objects from display, and unpicks them. Objects can be temporarily hidden (without removing them from the list of picked objects) by using **SELECT OPTION=PICKEDREMOVE**.

## The **HISTORY** Command

### Summary

Controls the size of the history stream.

### Command line parameters

Command	<b>HISTORY</b>	
Parameter	Default	Function
<b>STATES</b>	0	Number of history states to keep in the history stream
<b>GUIINIT</b>		NO
		Interface option to allow the history dialog to be repopulated with the current states available.
<b>YES</b>		

### Notes

The **HISTORY** command allows the size of the history stream to be limited. This may help reduce the memory usage of the Modeller. The **STATES** parameter determines how many states are kept. If zero, there is no limit. If greater than zero, only this number of operations will be retained, and hence the **UNDO** command can only be called a limited number of times.

## The **IMPORT** Command

### Summary

Imports conductor data from a conductor file generated by the Modeller, Pre or Post-Processor.

### Toolbutton



### Command line parameters

Command	<b>IMPORT</b>	
Parameter	Default	Function
<b>FILE</b>	Conductor file to be imported	
<b>COPYIDENTIFIER</b>	Allows adjustment of the unique names of local coordinate systems	

### Notes

The **IMPORT** command reads a file containing conductor data. The conductor file can be generated by

- the Pre-Processor using [The CONDUCTOR Sub-command WRITE \[page 391\]](#),
- the Post-Processor [The CONDUCTOR Command \[page 727\]](#) with **ACTION=EXPORT**,
- and by the Modeller using [The EXPORT Command \[page 211\]](#).

The data in the file may contain more than one conductor.

The conductor data in the file contains the Local Coordinate System values, not the names of Local Coordinate Systems. If an existing Local Coordinate System is found that matches the coordinate system values of a conductor, this will be used, otherwise a new Local Coordinate System will be created and set as coordinate system 1 for the conductor. The new Local Coordinate System will be created with name based on the **COPYIDENTIFIER**. This parameter allows local coordinate systems created during **IMPORT** to be given a new unique name that is unlikely to conflict with other local coordinate systems present. This copy identifier is set in the command history, so naming of local coordinate systems during replay is consistent.

For hints on how to write compatible command scripts for importing conductors into the Modeller, Pre-Processor and Post-Processor see [Conditional Commands \[page 52\]](#).

## The **INSERTOP2FILE** Command

---

### Summary

Imports an Opera-2d data file.

### Command line parameters

Command	INSERTOP2FILE		
Parameter	Default	Function	
<b>OPTION</b>	<b>LOAD</b>	<b>LOAD</b>	Import the file specified by the <b>FILE</b> parameter
		<b>ANALYSE</b>	Open the data file and determine if any region groups have been defined
		<b>CLEAR</b>	Initialize the dialog
<b>FILE</b>		Name of the data file to be imported	
<b>IGNOREAIR</b>	<b>YES</b>	<b>YES</b>	Regions with the property of air will not be imported
		<b>NO</b>	Air regions in the data file will be imported
<b>GROUP</b>		Name of a particular region group to be imported from the file	
<b>USEGROUP</b>	<b>NO</b>	<b>NO</b>	Do not look for a particular group
		<b>YES</b>	Import regions with a particular group name specified by <b>GROUP</b>
<b>GROUPLABEL</b>	<b>YES</b>	<b>YES</b>	Add a label to the face if it belongs to a group. The label will be based on the group name
		<b>NO</b>	
<b>MATERIALLABEL</b>	<b>YES</b>	<b>YES</b>	Add a material label to the cells or sheet face bodies created when the file is imported
		<b>NO</b>	
<b>REGIONLABEL</b>	<b>NO</b>	<b>NO</b>	Add a region label to the bodies created when the file is imported
		<b>YES</b>	
<b>SWEEPDISTANCE</b>	<b>1</b>	Defines the extrusion distance or the angle of rotation for axisymmetric models	
<b>COPYIDENTIFIER</b>		Allows adjustment of the unique name of bodies	

### Notes

This command allows Opera-2d geometries to be imported into the Modeller.

The imported 2d geometry is represented in the 3d Modeller as a set of bodies. If **SWEETDISTANCE** is set to zero the Opera-2d regions will be imported as a set of sheet face bodies. If **SWEETDISTANCE** is non zero the imported regions are extruded by the distance, or for axisymmetric models rotated by the angle, specified by the value of the parameter.

All bodies created from the import geometry are assigned a label based on the name of the Opera-2d data file.

Additional labels can be added to the imported geometry using the parameters, **GROUPLABEL** and **REGIONLABEL**. These labels are always attached to the bodies created from the imported file. The parameter **MATERIALLABEL** adds a material label to the cells created during the extrusion process. If no extrusion is performed, **SWEETDISTANCE=0**, then the material label is added to the sheet face bodies.

The **COPYIDENTIFIER** parameter allows bodies created during the operation to be given a new unique body name that is unlikely to conflict with other bodies present. If unset, a new identifier is generated from existing bodies to prevent any possible conflict.

## The **LABEL** Command

---

### Summary

Controls setting additional labels on entities.

### Toolbutton



### Command line parameters

Command	<b>LABEL</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>ADD</b>	<b>ADD</b>	Adds a label to picked items
		<b>REMOVE</b>	Removes the named label from the picked items
		<b>CLEAR</b>	Removes all labels from the picked items
<b>LABEL</b>		Label to be added or removed	

### Notes

The **LABEL** command can be used to add labels to any entity within the model. This label can then be used as the basis for grouping items for selection for visual display, or for picking for modification.

Using **OPTION=ADD** will add the named label to all picked items. **OPTION=REMOVE** will remove the named label from all items that are picked, if they have the label.

**OPTION=CLEAR** will remove all labels from the picked items.

The label defined on a face can also be used as the name in the labelled face SCALA emitters.

During boolean operations, the labels attached use the data storage level of any other data attached to the entity when deciding on what labels are kept. If the data is at the same level, the labels from both entities are merged.

### Labels

Labels should be character strings including letters and numbers. The label will be "mangled" to replace spaces and invalid characters ([ ] + - \) with "\_", lowercase with uppercase and to make the name unique among labels of the same type. The mangled names are used by the Post-Processor.

If a label is the name of a labelled face SCALA emitter, with emitter data defined in an **emit** file, the name in the file must match the mangled name.

## The **LCS** Command

---

### Summary

Creates a new named Local Coordinate System (LCS), or changes the location of an existing Local Coordinate System of this name.

### Command line parameters

Command	<b>LCS</b>	
Parameter	Default	Function
<b>LCNAME</b>		Name of Local Coordinate System to be created or modified
<b>X0</b>		Origin of the Local Coordinate System
<b>Y0</b>		
<b>Z0</b>		
<b>THETA</b>		Euler angles defining the local orientation of the Local Coordinate System
<b>PHI</b>		
<b>PSI</b>		

### Notes

All values must be given and are specified in the Working Coordinate System. If a Local Coordinate System of the given name already exists, its position and orientation are reset to the new values.

Each Local Coordinate System is uniquely identified by its **LCNAME**. A Local Coordinate System can be set to be the current working coordinate system using the **WCS** command. A Local Coordinate System is also used within conductors to define the Local Coordinate System 1 of the conductor.

A Local Coordinate System can be picked from the display by using [The FILTER Command \[page 219\]](#) with **TYPE=LCS**. If a single Local Coordinate System has been picked it can be set to be the Working Coordinate System and can also be renamed. If one or more Local Coordinate System is picked they can be transformed, copied or deleted. Copying a Local Coordinate System will generate a new name from the original to keep the name unique.

## The **LIST** Command

---

### Summary

Shows information about entities within the model.

### Toolbutton



### Command line parameters

Command	<b>LIST</b>	
Parameter	Default	Function
TYPE	<i>none</i>	BODY
		CELL
		FACE
		EDGE
		VERTEX
		LCS
		CONDUCTOR
NUMBER		Entity number
IDENTIFIER		Entity identifier
UNIQUEBODYNAME		The unique name of the body used when picking BODY, CELL, FACE, EDGE or VERTEX
LCSNAME		The name of a local coordinate system.

### Notes

The list command shows information about

- an **LCS** by specifying **TYPE=LCS** and the **LCSNAME**.
- a **BODY** by specifying **TYPE=BODY** and its **UNIQUEBODYNAME**.
- a **CELL, FACE, EDGE** or **VERTEX** by specifying the containing **UNIQUEBODYNAME**, the **TYPE** and the entity **IDENTIFIER** or **NUMBER**.

If no entity corresponds to this information, data on all picked entities is given.

This command can be activated graphically by using **FILTER COMMAND=LIST**. Subsequent selections using the cursor will show information on that entity, rather than picking it. Alternatively the properties of a single entity can be listed by typing **I** on the keyboard while the outline of an entity is high-lighted during picking. For more information, see [The FILTER Command \[page 219\]](#).

After the properties of an entity have been listed, the numerical data is also available through the following system variables:

<b>Entity</b>	<b>Variables</b>	<b>Meaning</b>
<b>BODY</b>	<b>BODYFACES</b>	number of cells
	<b>BODYVOLUME</b>	volume
<b>CELL</b>	<b>CELLFACES</b>	number of faces
	<b>CELLVOLUME</b>	volume
<b>FACE</b>	<b>FACEEDGES</b>	number of edges
	<b>FACEAREA</b>	area
<b>EDGE</b>	<b>EDGELENGTH</b>	length
	<b>XSTART, YSTART, ZSTART</b>	cartesian coordinates at start
	<b>RSTART, TSTART, ZSTART</b>	polar coordinates at start
	<b>XEND, YEND, ZEND</b>	cartesian coordinates at end
	<b>REND, TEND, ZEND</b>	polar coordinates at end
	<b>XCENTRE<sup>a</sup>, YCENTRE, ZCENTRE</b>	cartesian coordinates of centre of circular or elliptical edge
	<b>RCENTRE, TCENTRE, ZCENTRE</b>	polar coordinates of centre of circular or elliptical edge
	<b>RADIUS</b>	radius of a circular edge
	<b>ANGLE</b>	angle subtended by a circular edge
	<b>MAJORRADIUS</b>	major radius of an elliptical edge
	<b>MINORRADIUS</b>	minor radius of an elliptical edge
	<b>X, Y, Z</b>	cartesian coordinates
<b>VERTEX</b>	<b>R, TH, Z</b>	polar coordinates
	<b>U, V, W</b>	origin
<b>LCS</b>	<b>THETA, PHI, PSI.</b>	Euler angles

<sup>a</sup>Variables like **XCENTRE** can be used with the alternative spellings, **XCENTER**, etc.

## The **LOAD** Command

---

### Summary

Loads data from a model file.

### Toolbutton



### Command line parameters

Command	<b>LOAD</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>NEW</b>	<b>NEW</b>	Clears all existing model data and loads a new model file.
		<b>INSERT</b>	Inserts new data from the file into the existing model.
		<b>REPLACE</b>	Deletes existing bodies, conductors and LCS and inserts the specified file.
<b>FILE</b>		File to be opened.	
<b>COPYIDENTIFIER</b>		Identifier used for generating identifiers on entities when inserting a file.	
<b>CADREPAIR</b>	<b>DEFAULT</b>	Options to improve imported data:	
		<b>DEFAULT</b>	For IGES data, this is equivalent to <b>STITCH</b> ; for other data types, <b>NONE</b> .
		<b>HEAL</b>	Attempt to remove inconsistencies and errors.
		<b>STITCH</b>	Form bodies from surfaces.
		<b>NONE</b>	Accept data as it is.

### Notes

The **LOAD** command reads data from a native Modeller data file or a non-native CAD file. A filename must always be given. The name of the loaded file is saved in string variable **LOADFILENAME**.

LOAD operates in 3 ways: **NEW**, **INSERT** or **REPLACE**.

- If opening a new file with **OPTION=NEW**, any existing model data is cleared using an implicit **CLEAR** command (see [The CLEAR Command \[page 180\]](#))<sup>1</sup>.

The user will be asked to confirm the operation if data has been changed.

**LOAD OPTION=NEW** sets the currently open file. This filename will be changed by the **SAVE OPTION=NEW** command and cleared by the **CLEAR REVERT=NO** commands.

- **OPTION=INSERT** is used to append additional parts to the model, by loading them from file, into the working coordinate system. If the model currently has no bodies or conductors, all bodies in the file will be loaded, including a model body and its mesh. If bodies or conductors do exist however, a model body in the file being inserted will not be loaded. All component bodies, conductors and LCS will be added. User variables, functions and strings are not loaded.
- **OPTION=REPLACE** deletes all existing bodies, conductors and LCS before inserting geometry from the data file.

With **OPTION=INSERT** and **OPTION=REPLACE**, only previously undefined data attached to any of these items is loaded. For example, if the current model has a material label **Iron** defined, the material properties associated with **Iron** will not be updated from those in the data file. Additionally, user variables, periodicity, model symmetry and analysis data will not be loaded from an inserted model.

The **COPYIDENTIFIER** is used to specify part of the identifier applied to all entities inserted from the file, and used by e.g. [The PICK Command \[page 277\]](#). By default, it does not need to be specified and the system will automatically provide a suitable value to ensure there are no conflicts.

## Native formats

The **LOAD** command will operate on "native" **opc** files and binary **opcb** files. The binary form of the data file can be loaded by explicitly giving the **opcb** extension. The loading of these files may be slightly faster, but the files may not be portable between different platforms. The **opcb** files should not be renamed to a different file type, since the file type is used to determine the format.

## Non-native formats

Data from other geometric modelling software can be loaded. The format will be determined from the file name and if necessary the contents of the file<sup>2</sup>:

- CATIA V4 (\*.model, \*.exp, \*.session)
- CATIA V5 (\*.CATPart, \*.CATProduct)
- IGES (\*.igs, \*.iges),
- NX (\*.prt)
- ParaSolid (\*.x\_t, \*.xmt\_txt, \*.x\_b, \*.xmt\_bin)

<sup>1</sup>The restriction that the **CLEAR** command cannot be used in a command loop applies to this form of the **LOAD** command as well. If it is necessary to use the **LOAD** command in a loop, the following commands should be used:  
**CLEAR USERVARIABLES=NO**

**LOAD OPTION=INSERT filename**

<sup>2</sup>Licenses must be obtained from Cobham Technical Services in order to use formats other than SAT and IGES.

- Pro/E (\*.prt, \*.prt.\* , \*.asm, \*.asm.\* , \*.xpr, \*.xas)
- SAT (\*.sat),
- SolidWorks (\*.sldprt, \*.sldasm)
- STEP (\*.stp, \*.step)

The data within the file will be loaded and manipulated to try to generate a valid solid body geometry, but the data should be checked to ensure that it has no errors. Files in these formats might not contain a valid solid body geometry. When loading files from other software packages, it should be noted that the model may not be well suited to finite element mesh generation and analysis and may cause problems. The **CADREPAIR** options can be used to improve the data.

**Linux:** note that CATIA V5, NX, Parasolid and SolidWorks formats are not available in Linux versions.

## Material Names

If possible, material names used in CAD files are transferred into the Modeller. Any "illegal" characters will be replaced with underscores. Material names are often used as part of a variable name, so the characters used in names must conform to the rules for variables (see "[User Variable Commands](#)" on page 55).

## The **LOFT** Command

---

### Summary

Fills the volume between two faces.

### Toolbutton



### Command line parameters

Command	<b>LOFT</b>		
Parameter	Default	Function	
<b>FACE1TANGENT</b>	1.0	Tangent continuity parameter of face 1, the first face selected	
<b>FACE2TANGENT</b>	1.0	Tangent continuity parameter of face 2, the second face selected	
<b>KEEP</b>	<b>NO</b>	<b>YES</b>	The original lofting faces are kept in the final model. A new cell will be created between the two faces
		<b>NO</b>	The original lofting faces are removed from the model.

### Notes

The **LOFT** command can be used to fill the volume between two faces.

It is necessary to select the two lofting faces before the command can be executed.

The way in which the space between the two lofting faces is filled is governed by the face tangent parameters, **FACE1TANGENT** and **FACE2TANGENT**, the **KEEP** parameter and any model edges that already connect the two faces.

The tangent continuity parameters control the smoothness of transition between the faces connected to the lofting face and the new faces created to bound the volume between the two faces.

**FACE1TANGENT** sets the smoothness for the face selected first and **FACE2TANGENT** for the face selected second. In every case the edges joining the selected faces will be curved but as the face tangent parameters tend towards zero, the edge shape will tend towards straight. Values greater than 0 will result in a smoothing effect and faces with underlying spline geometry will be created.

If **KEEP=YES** is specified then the lofting faces are retained and the volume between the two faces is filled with a new cell. The properties of this new cell will be the same as one of the cells adjacent to

the initial lofting faces. If **KEEP=NO** is defined the initial lofting faces are removed. Any cells adjacent to the lofting faces are removed to be replaced by one new cell. The new cell will be defined with the properties of the cell with the greater data storage level.

If any edges connect the two lofting faces these edges will be used as guides to help create the new geometry.

## The MATERIALS Command

---

### Summary

Sets the material properties associated with a material label.

### Toolbutton



### Command line parameters

Command	<b>MATERIALS</b>	
Parameter	Function	
OPTION	<b>CGS</b>	Work in mixed CGS units
	<b>DELETE</b>	Deletes the picked materials
	<b>INCH</b>	Work in SI units with inches
	<b>LIST</b>	Lists the material properties of the picked materials
	<b>METRE</b>	Work in SI units
	<b>MICRON</b>	Work in SI units with microns
	<b>MM</b>	Work in SI units with MM
	<b>MODIFY</b>	Sets the data for the picked materials
	<b>PICK</b>	Adds a material label to a list to be set
	<b>RESET</b>	Sets picked materials to have the properties of air
	<b>TRUECGS</b>	Work in true CGS units
<b>MATERIALLABEL</b>	Material label used with <b>OPTION=PICK</b>	
ANISOTROPY	Sets the anisotropy of all permeability, permittivity and conductivity	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>PACKED</b>	Set to packed
	<b>MULTIPLE</b>	Set to anisotropic

Command	<b>MATERIALS</b>	
Parameter	Function	
<b>LINEARITY</b>	Sets the properties to be linear or nonlinear	
	<b>LINEAR</b>	Set to linear
	<b>NONLINEAR</b>	Set to nonlinear
<b>MUANISOTROPY</b>	Sets the anisotropy of permeability	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>PACKED</b>	Set to packed
	<b>MULTIPLE</b>	Set to anisotropic
<b>MULINEARITY</b>	<b>HYSERETICMODEL</b>	Nonlinear analysis of a hysteretic material
	<b>LINEAR</b>	Set to linear
	<b>NONLINEAR</b>	Set to nonlinear
<b>MU</b>	Isotropic linear permeability	
<b>HC</b>	Isotropic coercivity	
<b>BH</b>	Isotropic nonlinear BH curve label	
<b>MPHASE</b>	Phase lag for isotropic permeability	
<b>MUXX</b>	Anisotropic components of linear permeability	
<b>MUYY</b>		
<b>MUZZ</b>		
<b>HCX</b>	Anisotropic components of linear coercivity	
<b>HCY</b>		
<b>HCZ</b>		
<b>BHX</b>	BH data labels defining the anisotropic components of nonlinear permeability.	
<b>BHY</b>		
<b>BHZ</b>		
<b>MAPHASE</b>	Complex phase lag for anisotropic permeability	
<b>SIGANISOTROPY</b>	Sets the anisotropy of conductivity	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>MULTIPLE</b>	Set to anisotropic
<b>SIGMA</b>	Isotropic conductivity	
<b>SPHASE</b>	Phase lag for isotropic conductivity	

Command	<b>MATERIALS</b>	
Parameter	Function	
<b>SIGXX</b>	Anisotropic components of conductivity	
<b>SIGYY</b>		
<b>SIGZZ</b>		
<b>SAPHASE</b>	Complex phase lag for anisotropic conductivity	
<b>EPSANISOTROPY</b>	Sets the anisotropy of permittivity	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>MULTIPLE</b>	Set to anisotropic
<b>EPSILON</b>	Isotropic permittivity	
<b>EPHASE</b>	Phase lag for isotropic permittivity	
<b>EPSXX</b>	Anisotropic components of permittivity	
<b>EPSYY</b>		
<b>EPSZZ</b>		
<b>EAPHASE</b>	Complex phase lag for anisotropic conductivity	
<b>KAPANISOTROPY</b>	Sets the anisotropy of thermal conductivity	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>MULTIPLE</b>	Set to anisotropic
<b>KAPPA</b>	Isotropic thermal conductivity	
<b>KAPXX</b>	Anisotropic components of thermal conductivity	
<b>KAPYY</b>		
<b>KAPZZ</b>		
<b>HCAP</b>	Heat capacity	
<b>MDEN</b>	Mass density	
<b>MAGNETIZATION</b>	Magnetization curve for Magnetization solver	
<b>MAGX</b>	Magnetization curves for anisotropic material properties for Magnetization solver	
<b>MAGY</b>		
<b>MAGZ</b>		
<b>SIGWIRE</b>	Electrical conductivity along wire	
<b>TURNSDENSITY</b>	Turns density. Unused because defined by total turns and area of cross section	
<b>AREAWIRE</b>	Area of the wire cross-section	

Command	<b>MATERIALS</b>	
Parameter	Function	
<b>BASENAME</b>	Base name for logging variables. Unused because same as Material Label	
<b>IC</b>	Critical current of the wire for Quench analysis	
<b>IW</b>	Defines the current flowing in the wire in a Quench analysis. Unused because it is linked to current of related circuit winding	
<b>JOULEHEAT</b>	<b>YES</b>	Does this material generate joule heat losses during a Quench analysis.
	<b>NO</b>	Unused because it should always be YES
<b>USESIBC</b>	<b>YES</b>	Use Surface Impedance Boundary Condition in Harmonic, Motional and Transient EM, Magnetization and Harmonic HF
	<b>NO</b>	
<b>HYSTSWITCHTOL</b>	Hysteresis switching tolerance	
<b>MECHANICALANISOTROPY</b>	Sets the anisotropy of mechanical properties	
	<b>ISOTROPIC</b>	Set to isotropic
	<b>ANISOTROPIC</b>	Set to anisotropic (orthotropic)
	<b>FULLANISOTROPIC</b>	Set to fully anisotropic
<b>YOUNGS</b>	Young's modulus of isotropic material	
<b>POISSON</b>	Poisson's ratio modulus of isotropic material	
<b>THERMALEXPANSION</b>	Thermal expansion coefficient of isotropic material	
<b>GENERALEXPANSION</b>	General expansion integral of isotropic material	
<b>YOUNGSXX</b>	Components of the Young's modulus tensor of an anisotropic material	
<b>YOUNGSYY</b>		
<b>YOUNGSZZ</b>		
<b>POISSONXY</b>	Components of the Poisson's ratio tensor of an anisotropic material	
<b>POISSONYZ</b>		
<b>POISSONZX</b>		
<b>SHEARMODULUSXY</b>	Components of the shear modulus tensor of an anisotropic material	
<b>SHEARMODULUSYZ</b>		
<b>SHEARMODULUSZX</b>		

Command	<b>MATERIALS</b>
Parameter	Function
<b>THERMAEXPANSIONX</b>	Coefficients of Thermal expansion of an anisotropic material
<b>THERMAEXPANSIONY</b>	
<b>THERMAEXPANSIONZ</b>	
<b>GENERALEXPANSIONX</b>	General expansion integrals of an anisotropic material
<b>GENERALEXPANSIONY</b>	
<b>GENERALEXPANSIONZ</b>	
<b>ELASTICMijkl</b>	Fully anisotropic elasticity matrix, where <b>ijkl</b> be <b>XXXX, XXYY, XXZZ, XXXY, XXXZ, XXZX, YYYY, YYZZ, YYXY, YYYZ, YYZX, ZZZZ, ZZXY, ZZYZ, ZZZX, XYXY, XYYZ, XYZX, YXYZ, YZZX or ZXZX.</b>
<b>SIBCFREQ</b>	The characteristic frequency experienced by a conducting material modelled using SIBC
<b>SIBCORDER</b>	The number of harmonics modelled in transient SIBC
<b>BOPERATING</b>	The flux density level used to calculate the skin depth for transient SIBC

## Notes

This command defines the material characteristics for use by the analysis programs.

A set of material labels is picked using the command repeatedly, with **OPTION=PICK** and a **MATERIALLABEL** specified. A material label can be removed from the set using **OPTION=UNPICK**. If no **MATERIALLABEL** is given, the set is emptied.

Issuing the command with **OPTION=MODIFY** will modify the properties of the picked material labels to the new values given in the parameters. The value of properties associated with the material labels are unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked material labels. If the data of one of these parameters is unset, or the picked material labels do not share the same value, then the parameter value is left clear.

New material labels are assumed to be linear and isotropic. **OPTION=RESET** will clear the properties associated with all of the picked material labels.

The properties of all material labels can be listed using **OPTION=LIST**.

Material labels that are not used, i.e. have no cell referencing them can be deleted using **OPTION=N=DELETE**. Deleting a material label that is in use will reset its properties to those of air.

All parameters can be specified. Which of these values will be used by the analysis will depend upon the analysis module being used, e.g. the phase lag properties are not used by the Magnetostatic solver.

See [Labels \[page 238\]](#) for more information on valid labels.

## Functional materials

All material properties can be defined as functional properties. These properties can be defined as functional properties of position ( $X$ ,  $Y$ ,  $Z$ ) or be recovered from element or nodal vectors added into the analysis database using the post-processor **TABLE** command.

Alternatively, functions can be nonlinear, i.e. dependent upon the solution that is currently being calculated. The variables available for nonlinear functional analysis are therefore different for the different analysis programs, and are listed below.

Magnetostatic	H
Current Flow and Electrostatic	E
Dynamic Electromagnetic, Motional Electromagnetic and Magnetization	B, E
Thermal and Quench	T

Each of these variables can be used directly or in expressions using them. Additional variables are also defined for the local component of each solution vector i.e. H components are **HLX**, **HLY** and **HLZ**.

Functional materials should have **LINEARITY=LINEAR**.

## Anisotropic materials

Anisotropic materials have different values of a property in different directions. The orientation of a cell is set by assigning a local coordinate system to the volume property label of a cell using [The VOLUME Command \[page 342\]](#). This orientation will then apply to all anisotropic material properties assigned to the material of the cell.

The anisotropic material properties refer to the X, Y and Z directions of the local coordinate system of the cell.

## Packed materials

Packed materials can be modelled in 2 ways:

- isotropic, using a single **BH** label with **ANISOTROPY=PACKED**. The packing factor and material orientation are supplied using [The VOLUME Command \[page 342\]](#).
- anisotropic, using functional material properties.

- Create user functions to return permeability orthogonal and tangential to the rolling direction or grain of the material as functions of  $H$ , e.g. **MUOH (H)** and **MUTH (H)** (see [User Function Command \[page 60\]](#)). The functions should cover the full range of expected values of  $H$ , extending beyond saturation if necessary.
- Define the packing factor as a user variable, e.g.:

```
$constant #pack 0.98
```

- The effective field normal to the laminations is a nonlinear function, **#hlz** which is given by the solution of:

```
$equation #hlz ((1-#pack) *muoh (#h) +#pack) *#hlz-hlz
```

For information on equation variables, see [User Variable Commands \[page 55\]](#).

- Define the effective field in the laminations, e.g.:

```
$parameter #h sqrt(hlx^2+hly^2+#hlz^2)
```

- The permeability tangential to the laminations in the rolling or grain direction, **MUXX**, can be defined as

```
$parameter #muxx muth (#h) *#pack+(1-#pack)
```

- The permeability tangential to the laminations orthogonal to the rolling or grain direction, **MUYY**, can be defined as

```
$parameter #muyy muoh (#h) *#pack+(1-#pack)
```

- The normal permeability, **MUZZ**, can be defined using, e.g.

```
$parameter #muzz muoh (#h) / (#pack+(1-#pack)*muoh (#h))
```

If **MUOH (H)** and **MUTH (H)** are the same function, this method is equivalent to the packed option.

## Magnetization

In the Magnetization solver, the hard magnetic properties of isotropic materials are defined using the following parameters.

- **MAGNETIZATION** defines a function which can provide values of  $M$  as a function of  $B$ ,  $B_{max}$  and  $T$ . Functions are defined using tables of values imported into the Modeller using the [User Function Command \[page 60\]](#).
- **MU** defines the linear recoil permeability which is used if the applied fields remagnetize the material. A constant value or an expression in terms of field values can be used.
- **HYSTSWITCHTOL** provides a tolerance (in flux density units) on switching between magnetization and demagnetization. The change in flux density must reverse by more than this value before a switch will be implemented. If no value is give,  $10^{-4}$  tesla will be used.

Multiple parameters (**BHX**, **BHY**, **BHZ**, **MAGX**, **MAGY**, **MAGZ**, **MUXX**, **MUYY**, **MUZZ**) are available for anisotropic materials.

Examples and further instructions are given in the ***Opera-3d User Guide***.

## Hysteretic materials

Hysteretic materials can be modelled by any of the transient and motional electromagnetic analysis solvers if Magnetization is licensed. The material should have **MULINEARITY=HYSTERETICMODEL** and the corresponding BH data should obey the rules given in [Hysteretic materials \[page 146\]](#).

During analysis, the magnetic history of each element must be stored in terms of the turning points of the B-H trajectory. The parameter **HYSTSWITCHTOL** is a tolerance (in flux density units) that is used for numerical efficiency. Two turning points are identified as the same point if their **B** values differ by less than the tolerance. **HYSTSWITCHTOL** must be small compared with the typical variation of **B** in the model but large enough to avoid numerical noise.

## SIBC

The [Surface Impedance Boundary Condition \[page 620\]](#) can be used in Harmonic, Transient and Motional EM, Magnetization and Harmonic HF to avoid very small elements which would otherwise be needed to model materials with very small skin depths. SIBC can only be applied to materials which touch non-conducting materials because the method cannot model currents which cross the material surface.

- Harmonic EM and HF: the materials can only have linear material properties.
- Transient and Motional EM and Magnetization: materials can be linear or non-linear.

Three parameters can be used to adjust the behaviour of the SIBC algorithm.

- **SIBCFREQ**: Transient analysis can model variations in time which are much more complex than a single frequency oscillation. However, the skin depth of a material is defined as a function of frequency. A typical fundamental frequency must be provided for each material. For a non-moving transient analysis, this will be the principal frequency of the drive; for motional analysis, the speed of rotation of magnetic poles will determine the frequency of the applied field experienced by the parts of a model which use SIBC.
- **SIBCORDER**: The order specifies the number of harmonics of the frequency which will be used. Only odd harmonics are included, so an order of 3 will use the fundamental and the 3rd and 5th harmonics. Increasing the order enables a larger spread of frequencies to be modelled but also increases the solution time. Available options are 1 to 5; 1 giving the fundamental only (used for purely sinusoidal fields/current and linear materials) through to 5 giving fundamental, 3rd, 5th, 7th and 9th harmonics.
- **BOPERATING**: Skin depth is also a function of permeability so for a non-linear material, the skin depth varies with the magnitude of the applied field. The SIBC algorithm needs to know a typical value of flux density in order to calculate a typical skin depth.

The values of frequency and B are not critical but supplying values in the right range will enable the SIBC algorithm to obtain the best solution.

## Floating conductor

To model a conductor with an unknown voltage in an electrostatic field, a non-zero value of conductivity (**SIGMA**) should be assigned.

## Mechanical properties

Mechanical properties of materials for stress analysis can describe ISOTROPIC, ANISOTROPIC (orthotropic) or FULLANISOTROPIC materials.

Isotropic and orthotropic materials are defined using single values or diagonal tensors of

- Young's modulus (**YOUNG**, **YOUNGSXX**, etc.);
- Poisson's ratio (**POISSON**, **POISSONXY**, etc.) and
- the shear modulus (**SHEARMODULUSXY**, etc.) (orthotropic only).

For fully anisotropic materials, these properties are combined into the elasticity matrix,  $C$ . This is a 6x6 matrix relating the 6 values of the stress tensor,  $\sigma$  (xx, yy, zz, xy, yz and zx) to the corresponding values of the strain tensor,  $\epsilon$ .

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \quad (3.2)$$

The matrix is symmetric so only the upper triangle is needed (**ELASTICMijkl**, where  $ijkl$  can be **XXXX**, **XXYY**, **XXZZ**, **XXXY**, **XXYZ**, **XXZX**, **YYYY**, **YYZZ**, **YYXY**, **YYYZ**, **YYZX**, **ZZZZ**, **ZZXY**, **ZZYZ**, **ZZZX**, **XYXY**, **XYZZ**, **YZYZ**, **YZZX** or **ZXZX**).

Static stress analysis can also use:

- thermal expansion coefficients (**THERMALEXPANSION**, **THERMALEXPANSIONX**, etc.) to impose deformations from a change in temperature;
- general expansion integrals (**GENERALEXPANSION**, **GENERALEXPANSIONX**, etc.) to impose magnetostrictive, electrostrictive or piezo strains calculated from magnetic or electric fields in the database;
- mass density (**MDEN**) when analysing the effects of gravitational or rotational forces.

## Units

The working material unit set can be changed using **OPTION=METRE**, **OPTION=CGS** or one of the other unit sets. If the units are changed, the values in the parameters are converted to the new unit set, and values entered will be interpreted in this unit set. Properties given as expressions are not converted until the database unit set has been set by [The SOLVERS Command \[page 309\]](#). The unit sets are as follows:

Unit Set	Length	Conductivity	Coercivity	Thermal Conductivity	Specific Heat Capacity	Mass Density
<b>SI</b>	metre	siemens/m	amp/m	W/(m K)	J/(kg K)	kg/m <sup>3</sup>
<b>Mixed CGS</b>	cm	siemens/cm	oersted	W/(cm K)	J/(g K)	g/cm <sup>3</sup>
<b>True CGS</b>	cm	siemens/cm	oersted	erg/(s cm K)	erg/(g K)	g cm <sup>-3</sup>
<b>SI (mm)</b>	mm	siemens/mm	amp/m	W/(mm K)	J/(kg K)	kg/mm <sup>3</sup>
<b>SI (inch)</b>	inch	siemens/inch	amp/m	W/(inch K)	J/(kg K)	kg/inch
<b>SI (micron)</b>	μm	siemens/μm	amp/m	W/(μm K)	J/(kg K)	kg/μm

## The **MESH** Command

---

### Summary

Generates the surface mesh of the model body.

### Toolbutton



### Command line parameters

Command	<b>MESH</b>	
Parameter	Default	Function
<b>GENERATOR</b>	<b>AUTOMATIC</b>	Choice of mesh generator:
		<b>AUTOMATIC</b> Let the program decide, based on details of the model.
		<b>ONE</b> Type I: tetrahedral and mosaic meshing.
		<b>TWO</b> Type II: tetrahedral meshing
<b>SIZE</b>	1	Maximum element size in the mesh.
<b>NORMALTOL</b>	30	Maximum angle between mesh facets normals (degrees).
<b>SURFACTOL</b>	0.0	Absolute tolerance to which the mesh must fit a curved surface.
<b>TOLERANCE</b>	1.0e-6	Absolute tolerance used to check point coincidence.
<b>TYPE</b>	<b>PREFERTETRA</b>	Type of volume mesh:
		<b>MOSAIC</b> Use hexahedral/prismatic elements wherever possible, otherwise use tetrahedra.
		<b>PREFERMOSAIC</b> Use hexahedral/prismatic elements wherever possible and tetrahedral not specified, otherwise use tetrahedra.
		<b>PREFERTETRA</b> Use tetrahedra except in cells where hexahedral/prismatic elements are specified and possible.
		<b>TETRAHEDRAL</b> Use tetrahedra everywhere.

## Notes

The **MESH** command can only be used after the model body has been created using the **MODEL** command.

The operation creates the surface mesh from the information provided. If cells, faces, edges or vertices have their own mesh data attached, the smaller value is used.

The mesh element size on a face will be determined from the distribution of elements along the edges. The mesh element size along an edge is determined from the smallest values from:

- the information from the **MESH** command parameters
- mesh information on any cells that contain the edge
- mesh information on any faces that are bounded by the edge
- mesh information on the edge itself
- mesh size information on the vertices at the ends of the edge<sup>1</sup>

The faces of all cells are meshed sequentially. A progress bar showing the number of the facet being processed and the total number of surface elements created so far is displayed at the bottom of the Modeller window frame.

On completion the status bar at the bottom of the window displays the message "Surface Mesh".

After the surface mesh has been created, the volume mesh can be generated using [The FILL Command \[page 216\]](#).

## Choice of Mesh Generator

The Modeller includes two mesh generators. The choice of generator can be left to the software (**AUTOMATIC**) or a specific generator can be chosen (**ONE** or **TWO**).

- Type I (**GENERATOR=ONE**) can create all supported element types in any model. This is the automatic choice when cell preferences indicate that hexahedra or prism elements are required.
- Type II (**GENERATOR=TWO**) can create tetrahedral meshes in most models. This is the automatic choice when only tetrahedral elements are required.

There are some special cases which Type II mesh generator cannot handle and Type I must be used: filamentary conductors in circuits; multiple periodicities.

The mesh generator chosen for the **MESH** command is also used for [The FILL Command \[page 216\]](#).

## Mosaic meshing

Mosaic meshing creates volume meshes using 4 types of element:

- hexahedra: 8-noded elements with 6 quadrilateral facets. Hexahedra will be created in cells which also have 8 vertices and 6 quadrilateral faces and have the same numbers of segments

---

<sup>1</sup>Vertex size information is ignored when creating a quadrilateral mesh on quadrilateral face.

on opposite edges (see [Transition ratio \[page 259\]](#)).

- prisms: 6-noded elements with 2 triangular facets and 3 quadrilateral facets. Prisms will be created in cells which have only one pair of similar faces (i.e. the same topology and subdivisions) connected by a number of quadrilateral faces which have the same numbers of segments on opposite edges (see [Transition ratio \[page 259\]](#)).
- tetrahedra: 4-noded elements with 4 triangular facets. Tetrahedra can be generated in any cell.
- pyramids: 5-noded elements with one quadrilateral facet and 4 triangular facets. Pyramids are used on quadrilateral faces of cells otherwise meshed with tetrahedra, when the neighbouring cell is meshed with prisms or hexahedra.

Mosaic meshing allows a mixture of hexahedra, prisms and tetrahedra, with pyramids forming an interface between elements with quadrilateral facets and elements with triangular facets as the first layer of elements in a cell otherwise meshed with tetrahedra.

The [SELECT Command \[page 299\]](#) can be used to display which cells are meshable with prisms or hexahedra using [LABELs Prismatic meshable](#) and [Hexahedral meshable](#).

## Transition ratio

Before meshing a cell, the Mosaic mesh generator will attempt to force opposite edges of quadrilateral facets to have matching subdivisions by increasing the subdivision on any edge which has fewer subdivisions than the maximum for the set of edges which should match. The transition ratio is the maximum value of

$$(matching\_subdivision / original\_subdivision)$$

for all the edges in the cell.

The program has upper limits on transition ratio:

- a lower limit is used for cells with no element shape preference.
- a higher limit is used for cells with preference of [HEXOPRISM](#). The user will be told if such a cell cannot be meshed with hexahedra or prisms because the transition ratio is higher than the limit.
- no limit is used for all cells when [TYPE=MOSAIC](#).

Cells with transition ratios above the limit are meshed with tetrahedra.

Note that the process of forcing the subdivisions to match can propagate a larger number of subdivisions through the mesh from cell to neighbouring cell.

## Mesh type

There are four types of mesh created by the [MESH](#) and [FILL](#) commands:

- [TETRAHEDRAL](#): use tetrahedral elements everywhere, ignoring cell preferences.
- [PREFERTETRA](#): takes the preference from each cell (see [The CELldata Command \[page 168\]](#)). If the cell preference is [HEXOPRISM](#), hexahedral or prismatic meshing will be used if possible; otherwise tetrahedral meshing will be used.

- **TYPE=PREFERTETRA** is the default. If no cells have a preference, this is equivalent to **TYPE=E=TETRAHEDRAL**.
- **PREFERMOSAIC**: takes the preference from each cell (see [The CELLDATA Command \[page 168\]](#)). If the cell preference is **TETRAHEDRA**, or hexahedral or prismatic meshing is not possible, tetrahedral meshing will be used; otherwise hexahedral or prismatic meshing will be used.
- **MOSAIC**: use Mosaic meshing, ignoring cell preferences: hexahedra in hexahedral cells, prisms in prismatic cells and tetrahedra elsewhere with pyramids forming the interface between elements with triangular and quadrilateral facets.

The **MESH** command determines the element type for each cell and creates a surface mesh of quadrilaterals or triangles on each face as appropriate.

## Curved surface refinement

Additional elements can be included in the mesh to model curved surfaces more accurately. There are two controls:

- **NORMALTOL** specifies the largest angle allowed between the normal vectors to two adjacent elements on the same surface.
- If **SURFACETOL** is greater than zero, it specifies the largest distance allowed between the centroid of a surface element and the real surface.

If the mesh breaks either of these criteria, the elements will be subdivided to reduce the angles between normals or the distances between element centroids and the real surface.

On quadrilateral surfaces which are to be meshed with quadrilateral facets, refinement only affects the edge subdivision. However, if the number of segments along a curved edge is increased according to the above criteria, the subdivision on the opposite edge of the face will also be increased to match.

## System variables

On successful completion of the surface mesh, the program sets the following system variables:

- **MESH** to 1
- **NODES** to the number of surface nodes
- **ELEMENTS** and **SURFACEFACETS** to the number of surface triangles or quadrilaterals

## Surface mesh errors

Simple topology tests are used to check the initial surface mesh created on the model. The program may report a number of different types of error, these are explained in the following:

- Surface mesh quality: Each triangle is measured and a quality factor is calculated from the normalized ratio of the radius of the inscribed circle to the perimeter length. This gives an equilateral triangle a quality factor of 1 and a triangle with one or two long edges which are about 50 times the height of the triangle a quality factor of 0.05.

Triangles are identified as of poor quality if their quality falls below 0.05. This value can be adjusted using the user variable, #VF\_PoorTriangleTol.

The label **Surface mesh error** is added to any faces containing poor quality triangles so that they can be selected for display using [The SELECT Command \[page 299\]](#).

- Surface mesh failure: CHECK the bodies for errors

This means that errors have been found in the surface mesh that are probably caused by errors in the model geometry. [The CHECK Command \[page 171\]](#) should be used to find these errors and possibly to repair them. Repairing is not always possible, in some cases it may be necessary to find a better way of constructing the model - for example avoiding glancing contacts between two curved surfaces.

- ACIS surface mesh contains Faces with no mesh - CHECK body for errors

This again means that errors have been found in the surface mesh that are probably caused by errors in the model geometry. The CHECK command should be used to find these errors and possibly to repair them as above. This error may also arise if a vanishingly small face has been created by small errors in the dimensions of the model.

- ACIS surface mesh contains holes - CHECK body for errors

or

ACIS surface mesh is degenerate - change mesh size

This means that holes have been found in the surface mesh that may be caused by errors in the model geometry or by the surface meshing of the geometry. The **CHECK** command should be used first to see if there are errors in the geometry, and these should be repaired. If no errors are found, a more rigorous surface meshing procedure can be used that tests for loss of internal edges and points. The string variable **ACISMESHCHECK** can be used to control the level of testing applied to the surface meshing process, the default is:

**\$STRING ACISMESHCHECK NONE**

The following options may be used

String Variable	<b>ACISMESHCHECK</b>
Option	Description
<b>NONE</b>	No additional tests performed
<b>PLANARVERTEX</b>	Check for duplicate vertices only on planar faces
<b>PLANAREdge</b>	Check for duplicate edges only on planar faces
<b>PLANARALL</b>	Check for duplicate edges and vertices on planar faces
<b>ANYVERTEX</b>	Check for duplicate vertices on all faces
<b>ANYEDGE</b>	Check for duplicate edges on all faces
<b>ANYALL</b>	Check for duplicate vertices and edges on all faces

This checking can slow the surface meshing significantly, so should only be used where problems occur. **PLANARVERTEX** is the fastest option, and **ANYALL** is the slowest.

- Unable to allocate memory required

This means that a gigantic surface mesh was created and no more memory was available for storage. The mesh size controls applied to the model are probably not realistic.

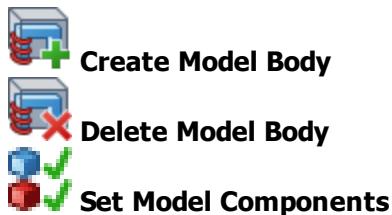
## The **MODEL** Command

---

### Summary

Creates a single body model suitable for meshing and analysis.

### Toolbuttons



### Command line parameters

Command	MODEL		
Parameter	Default	Function	
<b>OPTION</b>	<b>CREATE</b>	<b>CREATE</b>	Creates a model body for meshing and analysis
		<b>DELETE</b>	Deletes the model body
		<b>DEFAULT</b>	Sets the status of picked bodies for inclusion during create model body.
		<b>INCLUDE</b>	
		<b>EXCLUDE</b>	
	<b>LOAD</b>		Loads the status of the picked bodies

### Notes

The **MODEL OPTION=CREATE** command is used to generate a single body model. This is a necessary step before meshing. It must be used before a surface or volume mesh can be generated. For more information, see [Building a Model for Mesh Generation \[page 103\]](#).

The status of picked component bodies can be set by using:

- **OPTION=DEFAULT** so that bodies will be included in the model body unless they are formed from wire edges only.
- **OPTION=INCLUDE** so that bodies will always be included
- **OPTION=EXCLUDE** so that bodies will always be excluded.

During the operation, if no bodies are picked, the union without regularization of all bodies, whose status indicates that they are to be included, is created to form the model body. If bodies are picked, their union is formed without regularization to form a single *model body* (and the status of the bodies is ignored)

If any of the bodies have the name **BACKGROUND**, the *model body* is trimmed to the boundary of the union of these bodies, before being unioned with them. This is useful when only a section of the model is needed, but when the geometry is easier to build completely, e.g. for axisymmetric models where the construction is best done using cylinders.

Any cell within the model body, that has material label name **NULL** is deleted from the model body.

After forming the model body, any geometric layering data, applied using the **FACEDATA** command, is used to generate thin regions from these faces.

Once created, the model geometry cannot be changed. Properties associated with parts of the model can be adjusted, but these changes are not applied to the components used to form the model, and hence changes will be lost when reverting to component view.

When the model body has been created, this is the only body that can be seen. Component bodies used to form the model body still exist, but may not be manipulated or viewed until the model body has been deleted using **MODEL OPTION=DELETE** or with the **DELETE** command.

The model body can be saved, and the component bodies will be saved when using **SAVE OPTION=N=ALL** or **OPTION=NEW**. When loading, the view will be of the model body, rather than the components. If loading with **OPTION=INSERT**, the model body is not loaded.

Conductors and Local Coordinate Systems are unaffected by the **MODEL** command and can still be created and modified.

## The **MORPH** Command

---

### Summary

Morphs the geometry of bodies by applying a functional transformation.

### Toolbutton



### Command line parameters

Command	<b>MORPH</b>	
Parameter	Default	Function
<b>UMAPPING</b>	<b>U</b>	Expression for the new U coordinates.
<b>VMAPPING</b>	<b>V</b>	Expression for the new V coordinates.
<b>WMAPPING</b>	<b>W</b>	Expression for the new W coordinates.
<b>KEEP</b>	<b>NO</b>	Keep the picked items for further operations: <b>YES</b> or <b>NO</b> .

### Notes

The **MORPH** command allows bodies to be transformed to a new shape. The expressions specified for **UMAPPING**, **VMAPPING** and **WMAPPING** must be given in terms of the original **U**, **V** and **W** coordinates of the bodies.

For example the command

```
morph umapping=u vmapping=v+sin(w) wmapping=w
```

would transform the body by corrugating it.

Before the **MORPH** operation starts, the expressions given for **UMAPPING**, **VMAPPING** and **WMAPPING** are converted from the Opera command language syntax to that used by the underlying ACIS software. The changes in syntax are handled automatically and function names are replaced if necessary with the equivalent ACIS functions.

- Most common mathematical functions can be used:
  - `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()`;
  - `sind()`, `cosd()`, `tand()`, `asind()`, `acosd()`, `atand()`;
  - `abs()`, `exp()`, `log()`, `sqrt()`.

- Note that there are some mathematical functions that cannot be used in expressions:
  - functions with multiple arguments such as `atan2()` and `min()`;
  - `nint();`
  - `delta()`, `if()`, `range()`, etc., which are specific to Opera;
  - user defined functions.

The morph command will not change the underlying topology of a body (i.e. it will still have the same number and connectivity between faces, edges etc.). Morphing can cause e.g. self-intersection which will result in bodies with topological errors. These are very likely to cause problems in later operations.

Note that any morph that does not make use of all 3 values of **U**, **V** and **W** will be transformed into a plane (only 2 values used) or an edge (only one value given). The body formed from these operations will not be valid as the topology will not be updated to match the new dimensionality.

## The **MOUSE** Command

---

### Summary

Sets the functionality of the mouse buttons.

### Toolbutton



### Command line parameters

Command	<b>MOUSE</b>
Parameter	Function
<b>BUTTON</b>	Which button?
<code>LEFT</code>	Left button
<code>MIDDLE</code>	Middle button or wheel
<code>RIGHT</code>	Right button
<b>MODIFIER</b>	Additional controls:
<code>CONTROL</code>	Simultaneously press the <code>control</code> key
<code>DOUBLE</code>	Press the mouse button twice in quick succession (only available with the <code>LEFT</code> button).
<code>NONE</code>	No additional controls
<code>SHIFT</code>	Simultaneously press the <code>shift</code> key
<b>ACTION</b>	Action performed:
<code>CHECK</code>	Check that all necessary actions have been assigned
<code>CLEAR</code>	No action
<code>CONTEXTMENU</code>	Pop-up the context menu
<code>DEFAULT</code>	Restore default settings
<code>PAN</code>	Pan the picture
<code>PICK</code>	Pick the high-lighted entity
<code>ROTATE</code>	Rotate the picture
<code>ZOOM</code>	Zoom the picture

## Notes

The **MOUSE** command allows the 5 actions performed by the mouse in the 3d graphics window to be assigned to the 3 mouse buttons. The current settings are saved when the program ends for subsequent runs.

The default mouse functions are:

Action	Button and Modifier
CONTEXTMENU	RIGHT
PAN	RIGHT with SHIFT
PICK	DOUBLE click LEFT
ROTATE	LEFT
ZOOM	MIDDLE. If the middle button is a wheel, turning the wheel can also be used for ZOOM (Microsoft Windows only)

While the view of the picture is being changed with rotate, pan or zoom, the program hides some features to improve responsiveness. The whole picture can be seen during these operations by using the shift or control modifiers as well as the appropriate mouse button, unless that combination has been assigned to another function.

There are 3 additional **ACTIONS**:

- **CHECK** can be used to ask the program whether all necessary actions are provided by the current settings. **MOUSE ACTION=CHECK** is run automatically when the **Mouse buttons** dialog is closed.
- **CLEAR** can be used to remove any action from a button/modifier combination.
- **DEFAULT** can be used to restore all the settings to the above defaults.

N.B. If a Microsoft Windows user has a wheel mouse, it is possible to have two functions applied to the middle button: one when it is pressed (e.g. **PAN**) and one when it is turned (always **ZOOM**). This avoids the need to use the shift key.

## The **MULTIPHYSICS** Command

---

### Summary

Sets the list of simulations for a multiphysics analysis.

### Command line parameters

Command	<b>MULTIPHYSICS</b>	
Parameter	Default	Function
<b>OPTION</b>	<i>none</i>	<b>ADD</b> Add a new analysis stage given by <b>TYPE</b> .
		<b>DELETE</b> Delete an analysis <b>STAGE</b> .
		<b>DOWN</b> Move analysis <b>STAGE</b> down the list.
		<b>RESET</b> Remove all analysis <b>STAGEs</b> from the list.
		<b>UP</b> Move analysis <b>STAGE</b> up the list.
<b>TYPE</b>	<i>none</i>	Analysis type to be added as new stage:
		<b>ELEKTRARO</b> Velocity EM
		<b>ELEKTRASS</b> Harmonic EM
		<b>ELEKTRATR</b> Transient EM
		<b>ELEKTRAVL</b> Velocity EM
		<b>SCALA</b> Charged Particle
		<b>SOPRANOEV</b> Modal HF
		<b>SOPRANOSS</b> Harmonic HF
		<b>STRESSST</b> Static Stress
		<b>TEMPOST</b> Static Thermal
		<b>TOSCACURRENT</b> Current Flow
		<b>TOSCAELEC</b> Electrostatic
		<b>TOSCAMAGN</b> Magnetostatic
<b>STAGE</b>	<i>none</i>	Stage number in a multiphysics analysis.

### Notes

This command creates a set of analyses to be solved in sequence in a single database. Analyses are added by their **TYPE**. Once in the list, they are identified by the **STAGE** number.

The parameter **OPTION** specifies the action of the command:

- **ADD**: add a simulation given by **TYPE** to the list
- **DELETE**: remove the simulation given by the **STAGE** number
- **DOWN**: move the simulation at the **STAGE** number down the list
- **RESET**: remove all simulation **STAGES** from the list
- **UP**: move the simulation at the **STAGE** number up the list

When a new database is created using [The SOLVERS Command \[page 309\]](#), all the stages will be added to the database. After creating a new database, more stages can be added to the list. The **SOLVERS** command can then be used to add any new stages to the database.

## Simulation types

The following simulation **TYPEs** can be included in a multiphysics database. Any restrictions which apply to multiphysics simultaneous are described here.

- **ELEKTRARO**: Fixed Velocity Electromagnetic.
- **ELEKTRASS**: Harmonic Electromagnetic.
- **ELEKTRATR**: Transient Electromagnetic. This can only be used as a static analysis (i.e. at time zero). It does enable magnetostatic analysis with circuits and meshed conductors.
- **ELEKTRAVL**: Fixed Velocity Electromagnetic.
- **SCALA**: Charged Particle.
- **SOPRANOEV**: Modal High Frequency.
- **SOPRANOSS**: Harmonic High Frequency.
- **STRESSST**: Static Stress.
- **TEMPOST**: Static Thermal.
- **TOSCACURRENT**: Current Flow.
- **TOSCAELEC**: Electrostatic.
- **TOSCAMAGN**: Magnetostatic.

## Multiple cases

There are some additional features for analyses which can have multiple cases:

- multiple frequencies: Harmonic EM and Harmonic HF;
- multiple scaling factors: Magnetostatic, Electrostatic, Current Flow and Space Charge.

Frequencies or scaling factors can only be set for the first simulation of the type in the sequence. Subsequent simulations of the same type will be run with the same frequency or scaling factor.

Multiple frequencies or scaling factors imply a loop over all subsequent simulations in the sequence. For example, a sequence of simulations as follows:

1. Harmonic EM with  $n$  frequencies
2. Static Thermal
3. Static Stress

will be expanded to  $3n$  simulations when the database is created:

1. Harmonic EM at the first frequency
2. Static Thermal
3. Static Stress
4. Harmonic EM at the second frequency
5. Static Thermal
6. Static Stress
7. Harmonic EM at the third frequency
8. Static Thermal
9. Static Stress
10. etc.

Simulation values for single and multiple cases can be set using [The DBCASEDATA Command \[page 197\]](#).

## Data transfer

The results of each analysis can be used as input to subsequent analyses. In addition to the standard solution vectors (fields, temperatures, etc.), which can be used in expressions for material and volume properties, other quantities are calculated to transfer information:

- **JHD** (joule heat density):  $I^2R$  losses in conductors.
- **LOSS**: heat density calculated from **J.E** including materials modelled using the Surface Impedance Boundary Condition.
- **BEAMPOWER**: from particle trajectories.
- **EFDS**<sup>1</sup>: element force densities, calculated by Lorentz force or Maxwell stress, depending on the material properties.

In thermal analysis, **JHD+LOSS** will be applied to all cells with thermal conductivity unless any cell in the model has a volume property label which specifies a value of heat. If any cell volume property label has a value of heat, then all values of heat must be provided using volume property labels of the cells.

Similarly, in stress analysis, **EFDS** will be applied to all cells with mechanical properties unless any cell in the model has a volume property label which specifies values of body force density. If any cell volume property label has values of body force density, then all values of body force density must be provided using the volume property labels of the cells.

For more information on cell volume property labels, see [The CELldata Command \[page 168\]](#) and [The VOLUME Command \[page 342\]](#).

The use of displacements is optional (see the **USEDEFORMEDMESH** parameter of [The ANALYSISDATA Command \[page 122\]](#)). If selected, this will adjust the nodal coordinates by the displacement vector calculated from an earlier stress analysis.

---

<sup>1</sup>**EFD**, which is calculated by the Post-Processor, gives the same values as **EFDS**, if the default option of the automatic choice of Lorentz force or Maxwell stress is selected.

## Material properties

Material properties cannot change during a multiphysics analysis sequence except any which are defined by expressions in terms of fields, displacements, etc. which change as the analysis progresses.

When multiphysics is selected, the dialogs of [The MATERIALS Command \[page 248\]](#) give access to all material properties: electromagnetic, thermal and mechanical.

## The **OFFSET** Command

### Summary

Makes bodies grow or shrink by offsetting picked faces.

### Toolbutton



### Command line parameters

Command	OFFSET	
Parameter	Default	Function
DISTANCE		The distance to offset the face.
KEEP	NO	Keeps the face picked after offsetting.
	YES	
	NO	

### Notes

This command operates on a set of picked faces. They do not have to be planar faces.

If a sheet face is picked then a solid body is created. The face is offset by half of the **DISTANCE** each side of the initial sheet face.

If the face is part of a body then a positive value for **DISTANCE** will offset away from the body, creating a larger body. If the value for **DISTANCE** is negative then the face is offset into the body creating a smaller body.

## The PERIODICITY Command

---

### Summary

Defines the set of transformations for creating periodic symmetry boundary pairings within the model.

### Command line parameters

Command	PERIODICITY		
Parameter	Default	Function	
NSETS	0	Defines the number of periodicity transforms defined.	
DX1	0	Defines the x, y, z displacement of the first periodicity transform.	
DY1	0		
DZ1	0		
ROTX1	0	Defines an axis, based on the Global Coordinate System, about which a rotational component of the first periodicity transform can be defined	
ROTY1	0		
ROTZ1	0		
ANGLE1	0	The angle of rotation about the axis.	
TYPE1	POSITIVE	POSITIVE	Set the potentials on paired faces to be the same.
		NEGATIVE	Set the potentials on paired faces to have the opposite sign.
OPTION1	MATCH	MATCH	Force the geometry of the model to match the specified periodicity transform.
		EXTERNAL	Consider the faces on the external surface of the model when pairing faces.
		SYMMETRY	Consider faces which have been assigned the symmetry boundary condition.
DX2	0	Defines the x, y, z displacement of the second periodicity transform.	
DY2	0		
DZ2	0		
ROTX2	0	Defines an axis, based on the Global Coordinate System, about which a rotational component of the second periodicity transform can be defined	
ROTY2	0		
ROTZ2	0		

Command	PERIODICITY	
Parameter	Default	Function
ANGLE2	0	The angle of rotation about the axis.
TYPE2	POSITIVE	See TYPE1
OPTION2	MATCH	See OPTION1
DX3	0	Defines the x, y, z displacement of the third periodicity transform.
DY3	0	
DZ3	0	
ROTX3	0	Defines an axis, based on the Global Coordinate System, about which a rotational component of the third periodicity transform can be defined
ROTY3	0	
ROTZ3	0	
ANGLE3	0	The angle of rotation about the axis.
TYPE3	POSITIVE	See TYPE1
OPTION3	MATCH	See OPTION1

## Notes

The **PERIODICITY** command is used to define periodic or symmetry boundaries. Periodic boundaries are recognized by all solvers.

Periodic boundaries link the potential values, and hence the fields, over one surface to the potential values over another surface within the model. More precisely, the potential values on one surface are set equal to the values on the other surface (with or without a change of sign).

The **PERIODICITY** command is used to define which faces are linked or paired within the model. This pairing is achieved by specifying a periodicity transform which maps one face, defined as the master face, onto its paired face, the slave face. Edges and vertices are also paired by the periodicity transform. A maximum of 3 separate periodicity transforms can be specified. The actual number of sets used is controlled by the **NSETS** parameter.

Each transform is defined by a displacement and a rotation about the origin. For the first periodicity transform the displacement is defined by the parameters **DX1**, **DY1** and **DZ1**, which correspond to the x, y and z components of the displacement. The rotational component of the transform is specified by the vector **ROTX1**, **ROTY1**, **ROTZ1** and the parameter **ANGLE1**. The vector defines the axis of rotation and the parameter **ANGLE1** determines the angle of rotation about this axis.

The parameter **TYPE1** specifies whether the potentials over paired faces have the same (**TYPE1=POSITIVE**) or opposite (**TYPE1=NEGATIVE**) sign. The parameter **OPTION1** defines the method by which vertices, edges and faces are paired.

- To crop the geometry of the model to match the periodicity conditions defined, **OPTION1=MATCH** can be specified.

- To pair only the external faces, **OPTION1=EXTERNAL**, can be specified. When an exact match is not possible between faces, faces are subdivided to ensure that the paired faces have the same geometry
- To pair faces with the **SYMMETRY** boundary condition applied **OPTION1=SYMMETRY** can be used. This option can be selected when only certain faces on the external surface of the model have to be paired. Both faces to be paired need to have the **SYMMETRY** boundary condition applied.

The second and third periodicity transforms can be defined in a similar way.

The pairing of faces is performed when the model body is created. Once the model body has been formed, paired vertices, edges and faces are assigned labels which can be displayed. A label is also given to faces with positive symmetry and to faces with negative symmetry.

When the surface mesh is generated, the mesh on paired faces will be identical. A consequence of this is that any surface mesh refinements specified on one vertex, edge or face of a pair will also be applied to the mesh refinement on the second vertex, edge or face of the pair.

Warnings are given if:

- A defined periodicity transform does not result in the pairing of any faces within the model.
- A face with the **SYMMETRY** boundary condition applied is located within the interior of the model. Periodic boundaries can only be defined on the external surfaces of the model.
- A face with the **SYMMETRY** boundary condition is not paired by the periodicity transforms specified.

Periodic boundaries can also be applied using [The BACKGROUND Command \[page 135\]](#).

## The **PICK** Command

---

### Summary

Picks items for modification.

### Toolbuttons



### Command line parameters

Command	PICK		
Parameter	Default	Function	
OPTION	ADD	RESET	Clears all picked items
		ADD	Adds items into the list of picked objects
		REMOVE	Removes items from the list
		TOGGLE	Adds if the items are not picked, otherwise removes them
		FILTER	Removes any picked entities that are not of the current filter type
		CHANGE	Replaces picked entities with those of the current filter type
		ALL	Picks all entities of the current filter type
PROPERTY		Property of items including MATERIAL, BOUNDARY, ELEMENT, POTENTIAL etc.	
LABEL		Label associated with the specified property type	

Command	PICK			
Parameter	Default	Function		
TYPE	<i>none</i>	BODY	Type of entity whose data is to be shown	
		CELL		
		FACE		
		EDGE		
		VERTEX		
		LCS		
		CONDUCTOR		
NUMBER		Number for the item of the entity type		
IDENTIFIER		Identifier for the entity to be picked.		
PTU		Coordinate in or near to an entity to be picked		
PTV				
PTW				
WAIT	NO	Wait until after another command is issued before updating the display.		
WILDCARD	NO	YES	Wildcard used in labels (* is any number of characters and ? is one)	
		NO	Wildcards are not used	
UNIQUEBODYNAME		The unique name of the body, used when picking CELL, FACE, EDGE or VERTEX.		

## Notes

The **PICK** command allows objects to be picked for modification. Objects picked are stored in an ordered list and can then be used in many operations.

The list can be cleared using **OPTION=RESET**.

**OPTION=ADD** adds new entities to the list.

**OPTION=REMOVE** removes entities from the list

**OPTION=TOGGLE** adds an entity if it is not in the list, or removes it if it is in the list.

With **OPTION=ADD**, **OPTION=REMOVE** and **OPTION=TOGGLE** there are 3 ways of specifying items:

The first uses an entity **TYPE** with an **IDENTIFIER** or **NUMBER** and the **UNIQUEBODYNAME** of the containing body. This method is used during graphical selection of objects. The entity type can be **CELL**, **FACE**, **EDGE**, **VERTEX**, **LCS** or **CONDUCTOR**. Within a **BODY**, each entity has a unique identifier

and this is specified in the **IDENTIFIER** parameter. Additionally each entity has a unique integer identifier that can be specified in the **NUMBER** parameter.

The use of the **IDENTIFIER** flag should be used in preference to picking by **NUMBER**. The numbering scheme used might change between versions of the software. If scripting is used to vary values of parameters which define a model, boolean operations might lead to different numbers of cells, faces, etc. and hence affect the numbering of all entities in the body. Using the **IDENTIFIER** will provide a more consistent identification scheme. During **REPLAY** or scripting, changes to identifiers are more predictable and will not affect entities that are not directly involved in the operation.

Even when using identifiers, there is a risk that entities formed during boolean operations may be identified differently if parametric changes affect the resulting body created.

- If no identifier or number is specified, an entity of the given type belonging to the body specified by the **UNIQUEBODYNAME** can be picked by providing a coordinate (**PTU**, **PTV**, **PTW**), specified in the Working Coordinate System. If the type is cell, this method will pick a cell that contains the point. If the type is face, edge or vertex, the nearest entity will be found. Local Coordinate Systems and conductors cannot be picked using this method. If the pick is ambiguous, no item is picked.
- The third method uses the data attached to each of the entities. The **PROPERTY** parameter can be used to select a type of data item that is stored. For example **PROPERTY=MATERIAL** will pick objects that have a material label attached. This will be all cells, as they all have this information. By specifying a label the **PICK** command can be more refined, e.g. **PROPERTY=MATERIAL LABEL=AIR**, would pick only cells with a material label of air. A body should always be picked with its **UNIQUEBODYNAME** property.

If more than one item is picked in a single **PICK** command, the order in which items are added to the list is undefined. In most cases this is unimportant. For **COMBINE OPERATION=SUBTRACT**, **OPERATION=TRIM** or **OPERATION=CUTAWAY**, the order is critical to the topology of the resulting body and care must be taken in picking the objects. For any **COMBINE** operation, the order affects the name of the resulting body.

Subsequent **PICK** commands always add picked items to the end of the list of picked items.

Picked faces can be displayed with direction arrows showing the normal direction of the face. This normal direction is used by various commands, e.g. **SWEEP OPTION=DISTANCE**. The direction of the normal can be reversed by double clicking over one of the direction vectors (with **FILTER TYPE=E=FACE**). This has the effect of generating a **PICK** command with a negative entity **IDENTIFIER** or **NUMBER**.

**OPTION=FILTER** filters the list of picked items with the current filter entity type, and removes any that do not match.

**OPTION=CHANGE** changes the list of picked items to those of the current filter entity type. If this type is **CELL**, this option will pick all cells in all picked bodies. If type is **FACE**, this option will pick all faces in all picked bodies and cells. The filter option is then applied to remove any entities not of this type.

The **TYPE** parameter of the **PICK** command will be used. If it is unset, the **TYPE** parameter of the **FILTER** command will be used instead.

- **TYPE=CELL** converts all picked bodies to picked cells, and unpicks any other entity type (such as faces, edges and vertices).
- **TYPE=FACE** converts all picked bodies and cells to picked faces, and unpicks any other entity type.
- **TYPE=EDGE** converts all picked bodies, cells and faces to picked edges, and unpicks any other entity type.
- **TYPE=VERTEX** converts all picked bodies, cells, faces and edges to picked vertices, and unpicks any other entity type.

As explained above, the process only works in one direction (from edges to vertices) not the other way.

**OPTION=ALL** selects all items of the filter entity type.

**WAIT=YES** can be used if multiple pick operations are to be specified, to avoid unnecessary screen updates of the model.

## System variables

The following system variables hold the numbers of picked entities: **PICKEDBODIES**, **PICKEDCELLS**, **PICKEDFACES**, **PICKEDEDGES**, **PICKEDVERTICES** and **PICKEDCONDUCTORS**.

System variables relating to the entity picked are not updated. For this, use [The LIST Command \[page 241\]](#).

## The **PICTURE** Command

---

### Summary

Saves the current display in a file or on the clipboard.

### Toolbuttons



### Command line parameters

Command	<b>PICTURE</b>		
Parameter	Default	Function	
<b>SAVE</b>	<b>NO</b>	<b>NO</b>	Copy file to clipboard.
		<b>YES</b>	Save picture in a file.
<b>FILENAME</b>		The name of the file to be saved..	
<b>TYPE</b>	<b>PNG</b>	Type of image file format used to save the picture.	
		<b>BMP</b>	Window Bitmap
		<b>JPG</b> or <b>JPEG</b>	Joint Photographic Experts Group
		<b>PNG</b>	Portable Network Graphics
		<b>PPM</b>	Portable Pixmap
		<b>TIFF</b>	Tagged Image File Format
		<b>XBM</b>	X11 Bitmap
		<b>XPM</b>	X11Pixmap

### Notes

This command allows the current display to be stored on the clipboard, or in a file.

If **SAVE=NO** the image is placed on the clipboard and can then be pasted into another application. Under Linux, the image is placed on the X-Selection.

With **SAVE=YES**, the image is stored in a file. The filename can be specified in the **FILENAME** parameter, and the format can be selected with the **TYPE** parameter.

The size of the window can be set using [The GUIOPTIONS Command \[page 225\]](#) to ensure a consistent size for pictures.

An image can be printed directly using [The PRINT Command \[page 286\]](#).

## The PRECISIONDATA Command

### Summary

Sets tolerances for the Modeller.

### Toolbutton



### Command line parameters

Command	PRECISIONDATA	
Parameter	Default	Function
ABSTOL	1.0e-6	The minimum distance between geometric objects within the model
SMALLFACE	1.0e-6	The area of a face that is considered to be of negligible area

### Notes

This command controls the tolerance of operations within the Modeller. These tolerances should be set according to the size of the model being created. In general, these parameters should be set before building the model. Changing the minimum dimension when parts of the model have already been defined (especially if these are now outside the defined range) may cause some operations to fail.

The **ABSTOL** parameter sets the minimum distance at which points are treated as being independent. The **ABSTOL** parameter also defines the maximum dimension of the model. This is set to allow a dynamic range of  $10^{10}$  for double precision data (with allowances for calculation tolerances), giving a maximum dimension of  $10^{10} * \text{ABSTOL}$ , i.e.  $10^4$  by default. If the model does not sit comfortably within this dimension range, this tolerance should be adjusted accordingly.

During some boolean operations, the tolerancing may be such that very small faces are created. The **SMALLFACE** parameter defines the area considered by the Modeller to be negligible. Any face of area less than this value is unwanted, and the Modeller tries to remove it. In some cases, this may fail and the user may need to take steps to improve the model. Small faces might cause problems during mesh generation, or the mesh could be of lower quality near such a face.

When changing the value of **SMALLFACE**, the entire model is tested and faces deemed as small are removed if possible.

## The **PREVIEW** Command

---

### Summary

Allows preview of commands

### Command line parameters

Command	<b>PREVIEW</b>		
Parameter	Default	Function	
<b>OPTION</b>		<b>ACCEPT</b>	Accepts changes introduced by <b>PICK</b> , but cancels the operations previewed since <b>START</b>
		<b>END</b>	Cancels any operations since <b>START</b>
		<b>PICK</b>	Starts preview before a <b>PICK</b> command
		<b>START</b>	Start a preview of an operation
		<b>UNPICK</b>	Cancels changes introduced during <b>PICK</b> previewing
<b>REDISPLAY</b>		Redisplay changes after this command	
		<b>NO</b>	Do not redisplay
		<b>YES</b>	Update the display of the model

### Notes

This command allows previewing of subsequent commands. It creates markers in the history stream so that commands can be cancelled by rewinding the modelling history to the marked location. It is used by the Modeller interface when picking an item with the context menu, and in preview operations of specific commands.

There are 2 markers that can be introduced into the history stream by the preview command. The **PICK** marker is introduced with **OPTION=PICK**, the **START** marker by **OPTION=START**. The following assumptions and guides control the use of these 2 markers.

- It is assumed that a **PICK** marker will always be placed before the **START** marker.
- Any operations following a **START** marker will always be discarded when any other **PREVIEW** command is issued. This applies to the creation of user variables, except for model dimensions, during **PREVIEW**; model dimension variables not be lost.
- Operations following a **PICK** marker (excluding those after a **START** marker) can be kept by using **OPTION=ACCEPT**. **OPTION=PICK** or **UNPICK** will discard these changes.

Following a preview command, the option exists to redisplay. In some cases, it is not useful to redisplay immediately as another command will be issued immediately that would require further updates to the display.

## The **PRINT** Command

---

### Summary

Prints the current display.

### Toolbutton



### Command line parameters

Command	<b>PRINT</b>
No parameters	

### Notes

This command allows the current display to be printed as a bitmap. A dialog of the available printers and their options for printing is opened. The current display can then be printed on the selected printer.

The printers available are determined by the operating system.

The size of the window can be set using [The GUIOPTIONS Command \[page 225\]](#) to ensure a consistent size for pictures.

An image can be saved to file or copied to the clipboard using [The PICTURE Command \[page 281\]](#).

## The PRISM Command

---

### Summary

Creates an *n*-sided prism or pyramid, with points evenly distributed around a circular or elliptic base.

### Command line parameters

Command	PRISM	
Parameter	Default	Function
NAME		Attaches this name to the body formed
SIDES		Number of sides on the base of the prism
HEIGHT		Height along the axis of the prism
MAJORRADIUS		Major radius of the base of the prism
MINORRADIUS		Minor radius of the base of the prism
TOPRADIUS		Radius at the top of the prism
MATERIALLABEL		Set the material label of the new prism
LEVEL		Set the data storage level of the new prism
UNIQUENAME		Force a specific unique name for the prism created

### Notes

At least 3 sides must be specified. The HEIGHT must be greater than zero.

If the TOPRADIUS is given as zero, the body formed is a pyramid.

The points needed to form the *n*-sided polygon are regularly positioned around a base ellipse given by the MAJORRADIUS and MINORRADIUS, and the sides of the polygon are connected by straight edges. The prism is formed by sweeping along the z-axis of the working coordinate system, a distance of HEIGHT. If TOPRADIUS is different from the major radius of the base, the prism is tapered. If the TOPRADIUS is zero, an *n*-sided pyramid is formed.

The body formed is centred on the origin of the working coordinate system (it extends from -height/2 to +height/2 in the z-direction).

The single CELL created by the PRISM command is given a material label and data storage level if specified in the MATERIALLABEL and LEVEL parameters. See "The CELldata Command" on page 168. If the height of the prism is zero, a sheet face is created and the properties specified are attached to the FACE created. See "The FACEDATA Command" on page 213.

When a block is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

See also [The CYLINDER Command \[page 195\]](#), as this can also be used to create n-sided prisms and tubes.

## The **RACETRACK** Command

---

### Summary

Create or modify racetracks.

### Toolbuttons



### Command line parameters

Command	<b>RACETRACK</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>NEW</b>	<b>NEW</b>	Creates a new racetrack conductor
		<b>MODIFY</b>	Modifies properties of the picked racetrack conductors
		<b>LOAD</b>	Loads defaults from picked conductors
<b>DRIVELABEL</b>		Name for the racetrack drive label	
<b>LCNAME</b>		Name for Local Coordinate System for coordinate system 1	
<b>SYMMETRY</b>		Rotational symmetry about global Z axis	
<b>XCEN2</b>		Origin of coordinate system 2	
<b>YCEN2</b>			
<b>ZCEN2</b>			
<b>THETA2</b>		Euler angles defining orientation of coordinate system 2	
<b>PHI2</b>			
<b>PSI2</b>			
<b>RXY</b>		Reflection symmetries in XY, YZ and ZX planes	
<b>RYZ</b>			
<b>RZX</b>			
<b>XP1</b>		Inside lower on the racetrack cross-section	
<b>YP1</b>			

Command	<b>RACETRACK</b>		
Parameter	Default	Function	
<b>WIDTH</b>		Local x cross-sectional size	
<b>THICKNESS</b>		Local y cross-sectional size	
<b>H1</b>		Half-length of the straight	
<b>R1</b>		Radius of the arc	
<b>CURD</b>		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
<b>TOLERANCE</b>		Field calculation tolerance	
<b>INCIRCUIT</b>		Is the conductor part of a circuit:	
	<b>NO</b>	The conductor has defined current density.	
	<b>YES</b>	The current in the conductor is determined by a circuit.	
<b>REVERSE</b>		Reverse the connections to this conductor in its circuit: <b>YES</b> or <b>NO</b> .	
<b>CIRCUITELEMENT</b>		The name of circuit element this conductor is part of.	
<b>GROUPLABEL</b>		The group of which the conductor is a part for a Motional EM simulation	
<b>MODELCOMPONENT</b>		<b>NO</b>	Do not convert to meshable cells.
		<b>REGULAR</b>	Convert to meshable cells which will be regularly meshed.
		<b>YES</b>	Convert to meshable cells which meet the mesh size criteria.
<b>MESHSIZE</b>		The size of the mesh to be used when meshing	
<b>MESHLFACTOR</b>		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
<b>KEEP</b>	<b>NO</b>	<b>NO</b>	Clear the list of picked items
		<b>YES</b>	Keep the list of picked items for further modification

## Notes

This command creates a new racetrack conductor or modifies existing racetracks.

- **OPTION=NEW** creates a new racetrack conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked racetracks. If the picked racetracks do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking racetracks

(see [The PICK Command \[page 277\]](#)) to be modified and before **RACETRACK OPTION=MODIFY**.

- **OPTION=MODIFY** changes the conductor data of all of the picked racetracks to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Racetracks \[page 515\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The REDO Command

---

### Summary

Redo previously undone commands

### Toolbutton



### Short cut

<Ctrl>+Y

### Command line parameters

Command	REDO		
Parameter	Default	Function	
STATE	none	Name of the state to which the history is to be undone	
WAITDISPLAY	NO	NO	Refresh the display immediately
		YES	Wait until the next command before refreshing the display.

### Notes

This command allows an **UNDO** command to be reversed, by forwarding the state of the model to a former position.

If no value for **STATE** is given, the operation forwards one state.

The **REDO** facility is available to go to future positions in the history stream after the **UNDO** command. The **REDO** command is available until a new entry is added into the history stream, i.e. a new command changes the current state of the model.

Note: If generating command scripts from log files generated during sessions, it is advisable to remove any **UNDO** or **REDO** commands that specify a state name, as the state name is dependent upon the number of commands issued. This makes it *very sensitive* to changes earlier in the script.

The **WAITDISPLAY** parameter allows refreshing of the model to be delayed in case further operations are to be implemented.

## The **RENAME** Command

### Summary

Changes the name on a picked body or Local Coordinate System.

### Toolbutton



### Command line parameters

Command	<b>RENAME</b>	
Parameter	Default	Function
<b>NAME</b>		New name to be used to replace the existing name
<b>UNIQUE NAME</b>		New unique name to be used to replace the existing name (bodies only)

### Notes

The **RENAME** command changes the name associated with a single picked body or Local Coordinate System.

If more than one entity selected, the operation has no effect. Two Local Coordinate Systems may not have the same name. Renaming a Local Coordinate System will not affect any conductors that use this Local Coordinate System for describing coordinate system 1.

Bodies with the name **BACKGROUND** define the extent of the model space when creating the model body with the **MODEL** command.

## The REPLAY Command

---

### Summary

Roll back to an earlier command, so that the input values can be adjusted, before subsequent commands are replayed to rebuild a modified version of the model.

### Toolbutton



### Command line parameters

Command	REPLAY	
Parameter	Default	Function
<b>OPTION</b>	<b>PREPARE</b>	Prepare replay dialog with available states
	<b>REBUILDPREPARE</b>	Prepare rebuild dialog with available states
	<b>EDIT</b>	Start replay and start editing of the command associated with the history state
	<b>ACTION</b>	Re-issues the command associated with the history state
	<b>END</b>	Completes editing of the command and actions all downstream commands
	<b>CANCEL</b>	Aborts replay and reverts to the initial model state
<b>STATE</b>	<i>none</i>	Name of the history state to be rewound to and edited

### Notes

This command allows rewind of the history stream to an earlier command. The user input of this command can be edited and the command replayed. All further downstream operations are then actioned to rebuild the adjusted model.

It is recommended that **REPLAY** is used through the menus to ensure correct sequencing of the state editing. There are 3 methods of accessing the replay facility.

- The **Edit** menu entry shows all editable commands in the sequential order in which they were created.
- The **Operations** menu entry is active when one or more bodies are picked. The display in the dialog is a tree style view showing commands associated with the construction of the body and component bodies used in its construction.

- The variable definition dialog allows replay of all commands used to build the model. This will allow a change of a model dimension value to be reflected in its use when creating the model.

When specifying **OPTION=EDIT**, the value for **STATE** must refer to a valid history state. Valid history states are modelling operation commands with parameters that can be readily adjusted to recreate a variation of the current model.

Additionally, **STATE=ROOT** can be specified to repeat all commands, with no editing of any specific command available. This can be used after the redefinition of a model dimension variable to a new value to generate a new model with the newly specified dimensions.

When editing a command for replay, **OPTION=ACTION** can be issued to replay the original command input. This can be necessary to ensure that all command parameters and user defined constants are correctly defined.

When editing of the command is complete, the **OPTION=END** command is issued and this forces replay of the downstream operations to rebuild the model given the subsequent changes. When replaying these commands, a minimal set of commands is used to rebuild the current model state. This will mean that some commands generating history states will not be replayed, hence the history will not be the same as before the replay was started.

The command can be cancelled or aborted using the **OPTION=CANCEL** command. This will revert the state of the model to the original state of the model, saved when the **OPTION=EDIT** command was issued.

## Saving replay commands

The replay commands are automatically stored in the Modeller data file. Upon loading the replay commands are available for replay or rebuild.

When a model is created by an older version of the modeller that did not support replay, replay of the commands used to create the model will not be available. Additionally, as the set of commands needed to rebuild the full model will be incomplete, no replay commands will be saved to any modified version of the data file.

In some cases, particularly for models created from command files containing \$ control blocks, replay may become invalid. In such cases the replay commands will not be saved with the file.

## Replay limitations

- In some circumstances, the commands being replayed may no longer be valid. Where such an error occurs, the replay will be aborted and the model reverted to its original state. Typical causes for such problems are:
  - Changes to the user input means that a command is no longer valid, e.g. new dimensions for a block give zero volume.
  - A downstream pick operation is no longer valid, invalidating a subsequent command. Such cases might occur if a cell or face created during a boolean operation is picked. Changes to the

model, result in a different outcome from the boolean operation and this face / cell is no longer created or identifiable.

- The replay command works using a streamlined subset of the users input. The commands being used can be recovered to file by opening the Command File Editor, and selecting **File -> Open Command History**.

These commands can then be saved as a COMI file and adjusted or used as input.

- Where more complex changes to the history need to be made than those allowed using replay, this provides a very good base from which to make the necessary adjustments.
- Where a problem occurs during replay, this command script can be used to locate the source of the problem and to generate a model that replays successfully.
- Where command files have been used, the commands from the command file are inlined, so the command file is not needed for replay to work.
- When inserting other files, replay will rely on these files still being available and may no longer be portable if path names have been specified.
- Care must be taken if using the **REPLAY OPTION=EDIT** command using entry from the console where input of new user input is required. The commands issued from the menus are structured to allow editing and replay of the commands that are available. Incorrect or incomplete use of the command could cause the model to fail to rebuild correctly with changes that cannot be undone.
- The **REPLAY** command cannot be issued from within any \$ control blocks.

## The **SAVE** Command

---

### Summary

Saves model data to file.

### Toolbutton



### Command line parameters

Command	<b>SAVE</b>		
Parameter	Default	Function	
OPTION	ALL	ALL	Saves all model data to the current file
		NEW	Creates a new file containing all the model data
		PICKED	Saves all picked bodies, Local Coordinate Systems and conductors
		EXPORT	Exports picked bodies to SAT format file
		IGES	Exports the picked bodies to IGES format file
		MESH	Saves all model data to a binary file including the mesh
		STEP	Exports the picked bodies to STEP format file
		CATIA4	Exports the picked bodies to CATIA V4 format file
		CATIA5	Exports the picked bodies to CATIA V5 format file
FILE		Name of file to be created	
FILEVERSION		OPC, SAT, CATIA or STEP file version	

### Notes

**SAVE** will save a file containing all the model data. A saved file can be opened with [The LOAD Command \[page 243\]](#).

To save all the model data in a new **opc** file, use **SAVE OPTION=NEW**. To overwrite a previously saved or loaded **opc** file, use **SAVE OPTION=ALL**. Note that **OPTION=ALL** can only be used once the open file has been established by **LOAD OPTION=NEW** or **SAVE OPTION=NEW** commands.

The **opc** file format includes the command history so that the model can subsequently be rebuilt using [The REPLAY Command \[page 294\]](#). **FILEVERSION=20** can be used to create **opc** files compatible with versions of the Modeller earlier than 18.

To save a subset of the model, pick the bodies, Local Coordinate System or conductors and use **OPTION=PICKED** to save them to file.

Models can also be exported in formats for other geometric modelling software<sup>1</sup>:

- CATIA V4 format (**\*.model**): use **SAVE OPTION=CATIA4**. Supported **FILEVERSIONs** are 4.1.9 (the default) to 4.2.4;
- CATIA V5 format (**\*.CATPart**): use **SAVE OPTION=CATIA5**. Supported **FILEVERSIONs** are CATIAV5R6 to CATIAV5R23 (the default);
- IGES format (**\*.igs, \*.iges**): use **SAVE OPTION=IGES**;
- SAT format (**\*.sat**): use **SAVE OPTION=EXPORT**. Supported **FILEVERSIONs** are 12 to 25 (the default);
- STEP format (**\*.stp, \*.step**): use **SAVE OPTION=STEP**. Supported **FILEVERSIONs** are AP203 (the default) and AP214.

To save the whole model and the finite element mesh in a binary (**opcb**) file, use **SAVE OPTION=MESH**. This does not change the open file name used by **SAVE OPTION=ALL**.

A model data file is also created by [The SOLVERS Command \[page 309\]](#) when creating an analysis database.

**Linux:** note that CATIA V5 format is not available in Linux versions.

---

<sup>1</sup>Licenses must be obtained from Cobham Technical Services in order to use IGES, STEP or CATIA formats.

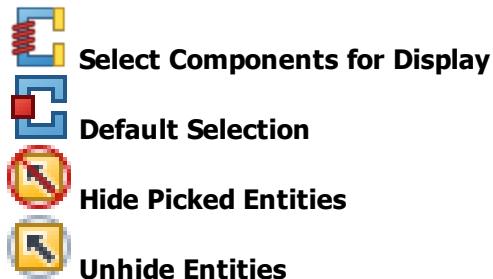
## The **SELECT** Command

---

### Summary

Selects items for display.

### Toolbuttons



### Command line parameters

Command	SELECT	
Parameter	Default	Function
<b>OPTION</b>	<b>ADD</b>	<b>ADD</b> Selects an item for display
		<b>DEFAULT</b> Sets the default view options
		<b>CURSORHIDE</b> Hides an item from the display when selected by the cursor
		<b>HIDE</b> Explicitly hides an item from the display
		<b>PICKEDREMOVE</b> Temporarily hides all picked entities
		<b>REMOVE</b> Removes the hidden or displayed flag from an item
		<b>RESET</b> Clears all selected items
		<b>TOGGLEDISPLAY</b> Toggles the selection of an item for display.
		<b>UNHIDE</b> Unhides all entities that have been hidden using <b>CURSORHIDE</b>
<b>PROPERTY</b>		Property of data for selection
<b>LABEL</b>		Label in the data of the specified type

Command	<b>SELECT</b>		
Parameter	Default	Function	
TYPE	<i>none</i>	BODY	Type of entity whose data is to be shown
		CELL	
		FACE	
		EDGE	
		VERTEX	
		LCS	
		CONDUCTOR	
IDENTIFIER		Identifier for the item of the entity type	
NUMBER		Unique number for the item	
PTU		Point nearest to or contained within an entity of the selection TYPE	
PTV			
PTW			
WAIT		YES	Do not update the display from the selection change
		NO	Update the display from this selection
AUTOUPDATE	YES	YES	Refresh the display after every change
		NO	Never update the display
		CLOSEWINDOW	Closes the display window
SELECTLEVEL	1	Set the options from which display items can be chosen	
WILDCARD	NO	YES	Wildcard used in labels (* is any number of characters and ? is one)
		NO	Wildcards are not used
UNIQUEBODYNAME		The unique name of the containing body used when picking CELL, FACE, EDGE or VERTEX	

## Notes

The **SELECT** command allows a list of objects to be selected for display and a list of objects to be explicitly hidden. It provides facilities for viewing the different data attached to entities, differentiating between different data with different colours. It also allows parts of the model to be hidden. This is a stronger option than not being displayed, as it forces items to be hidden from the display.

**OPTION=DEFAULT** will clear all display items and reset the data to display any entities with properties of *Material*, *Coiltype*, *LCName* or *System*.

The selected and hidden items are completely cleared using **OPTION=RESET**.

**OPTION=ADD** selects an item or data characteristic for display.

**OPTION=HIDE** selects an item or data characteristic to be explicitly hidden from the display.

**OPTION=REMOVE** removes the display or hide selection from an item or data characteristic.

With **OPTION=ADD**, **OPTION=REMOVE** or **OPTION=HIDE**, there are three ways of specifying the selection:

- Data is attached to each of the entities. The **TYPE** parameter can be used to select a type of data item that is stored. For example **PROPERTY=BOUNDARY** will select objects that have a boundary label attached.

By specifying a label the **SELECT** command can be refined, e.g. adding **LABEL=FACE1** in the above example would restrict the selection to entities with attached boundary condition label **FACE1**.

When using **OPTION=REMOVE** with only the **PROPERTY** parameter set, the selection of all individually selected items in that property list will also be removed.

It should be noted that different entity types will be selected by different properties. For example the *Material* property is attached to cells, so selecting **PROPERTY=MATERIAL** will select cells.

The *boundary* property is attached to faces, so **PROPERTY=BOUNDARY** will select faces.

The colour with which each object is displayed will indicate the specific label of the property.

Where the colour for the display of a section is ambiguous, the part is displayed with the specific colour indicating the ambiguity. Colours can be changed using **The COLOUR Command** [page 182].

- The entity **TYPE** is used, together with an entity **IDENTIFIER** or **NUMBER** and the **UNIQUEBODYNAME** of the containing body. This method is used during graphical selection of objects. The entity type can be **CELL**, **FACE**, **EDGE**, **VERTEX**, **LCS** or **CONDUCTOR**. Within a **BODY**, each entity has a unique identifier and this is specified in the **IDENTIFIER** parameter.
- A third method uses an entity **TYPE** and a point coordinate (**PTU**, **PTV**, **PTW**), defined in the current Working Coordinate System. Only entities of type body, cell, face, edge or vertex can be selected by this method. For cells and bodies, the entity that contains the point is selected. For others, the entity nearest to the supplied point is selected. If there is more than one possible selection, the result is undefined.

**OPTION=UNHIDE** unhides all objects that have been hidden by the second and third methods described above. Objects hidden by **PROPERTY** or **LABEL** will not be affected.

The different selections are stored and can be used in any combination. For example,

```
SELECT OPTION=ADD TYPE=MATERIAL
```

will display all entities with material data (i.e. all cells). A second command

```
SELECT OPTION=HIDE TYPE=MATERIAL LABEL=AIR
```

will mean that all materials are displayed except those with material label **AIR**.A further command

```
SELECT OPTION=HIDE TYPE=POTENTIAL LABEL=REDUCED
```

will mean that any reduced potential regions are also excluded from the display.

The faces of objects are used to display the objects, i.e. bodies, cells and faces are displayed by drawing their faces. This means that the selection of the display is quite complex, as the choice of selected and hidden items will often conflict. The following rules are used to choose the colour and faces to be shown:

1. High-light any picked item.
2. Determine any other parts of the model that have been hidden and flag these for exclusion from the display.
3. Include faces with associated data that have been selected for display, which are not hidden and which are not already displayed.
4. Include bodies with associated data that have been selected for display, which are not hidden and which are not already displayed.
5. Include cells with associated data that have been selected for display, which are not hidden and which are not already displayed.

It can be seen from the above order that the display will always include all picked items. **OPTION-N=PICKEDREMOVE** can be used to override this, so that any hidden object is temporarily removed from the display. Subsequent displays after issuing this command will revert to the normal display mode with picked items included.

A further option, **SELECTLEVEL=1** restricts the options to the basic information attached to the different entities. More options, e.g. the boundary condition type that is associated to a boundary label, are available with **SELECTLEVEL=2**.

By default any change to the model will cause the display to be refreshed. This behaviour can be changed using **AUTOUPDATE=NO** or **AUTOUPDATE=CLOSEWINDOW**. In the first case, the **THREED** command must be called to refresh the display to show changes. This should not normally be used during an interactive session as it will mean there is no visual feedback of any actions performed. The second option (**AUTOUPDATE=CLOSEWINDOW**) closes the display window, and releases any memory being used by the display (for example to allow larger model files to be processed).

When making multiple selections the **WAIT** parameter allows the **AUTOUPDATE** to be overridden until the next change. The **WAIT** parameter is always reset to **NO**, so must be explicitly declared with any **SELECT** command when needed.

## The **SHELL** Command

### Summary

Hollows out a body to leave a shell. Picked faces are removed and become openings into the resulting body.

### Toolbutton



### Command line parameters

Command	SHELL		
Parameter	Default	Function	
THICKNESS		Thickness of the sides of the shell.	
OUTWARDS	YES	YES	Shell is outside or inside the original body.
		NO	

### Notes

This command operates on a set of picked faces. The body will be replaced with a hollow body with the **THICKNESS** giving the thickness of the resulting shell. The picked faces will be removed and become openings into the body.

If **OUTWARDS=YES** then the resulting shell will be on the outside of the original body, such that the original body could fit inside. If **OUTWARDS=NO** then the resulting shell comes inwards from the original body. In this case the exterior dimensions remain the same.

## The **SKETCH** Command

---

### Summary

Displays a two-dimensional sketching grid on the xy-plane, and allows sketching of the primitive objects.

### Toolbutton



### Command line parameters

Command	<b>SKETCH</b>	
Parameter	Default	Function
<b>OBJECT</b>		The primitive to be sketched:
		<b>BLOCK</b> Defined by 3 points: 2 opposite corners on the base and the third defining the height
		<b>CYLINDER</b> Defined by 3 points: centre of base, circumference of base, height
		<b>NONE</b> Define no primitive, only set parameters
		<b>SPHERE</b> Defined by 2 points, centre and circumference
		<b>TORUS</b> Defined by 3 points, 1 <sup>st</sup> and 2 <sup>nd</sup> define the major radius, 2 <sup>nd</sup> and 3 <sup>rd</sup> the minor radius
		<b>WIREEDGE</b> Defined by 2 points
<b>ACTIVE</b>	<b>YES</b>	Define primitives by sketching: <b>YES</b> or <b>NO</b>
<b>SNAP</b>	<b>NO</b>	Include points on the snap grid in the set of snapping points: <b>YES</b> or <b>NO</b>
<b>SNAPVISIBLE</b>	<b>YES</b>	Include points on the visible grid in the set of snapping points: <b>YES</b> or <b>NO</b>
<b>AUTOGRID</b>	<b>YES</b>	Which grid should be displayed if <b>GRID=YES</b> :
		<b>NO</b> Display the user defined grid
		<b>YES</b> Display the automatic grid
<b>GRIDX</b>	10	Set the visible grid x size
<b>GRIDY</b>	10	Set the visible grid y size

Command	<b>SKETCH</b>	
<b>SNAPX</b>	0.5	Snap grid x coordinate
<b>SNAPY</b>	0.5	Snap grid y coordinate
<b>SNAPZ</b>	0.5	Snap grid z coordinate
<b>GRIDDX</b>	0.5	Set the visible grid x increment
<b>GRIDDY</b>	0.5	Set the visible grid y increment
<b>MINOR</b>	4	Number of minor grid lines for each major grid line
<b>COMPLETE</b>		Only for use by the GUI
<b>GRID</b>	<b>YES</b>	Display the grid: <b>YES</b> or <b>NO</b>

## Notes

The **SKETCH** command defines primitives by sketching on the graphics window. The command offers several choices of how sketching will operate:

- **ACTIVE** switches sketching on and off. When sketching is on, the graphical cursor will show as a dot. Points can be selected using the mouse pick operation (see [The MOUSE Command \[page 267\]](#)).
- **GRID** switches the display of a 2d grid on and off. The grid is displayed on the Z=0 plane of the working coordinate system. **AUTOGGRID** specifies which grid should be displayed:
  - **YES**: an automatically generated grid.
  - **NO**: a user specified grid, defined by the grid spacing (**GRIDDX**, **GRIDDY**) and the distance the grid extends from the local origin (**GRIDX**, **GRIDY**).
- Points can be entered precisely at the cursor position or the program can use the closest snapping coordinates. The set of snapping coordinates can include either or both of the following:
  - **SNAPVISIBLE** includes the lines of the visible grid in the set of snapping coordinates.
  - **SNAP** includes a set of coordinates at spacing **SNAPX** in X, **SNAPY** in Y, and **SNAPZ** in Z, the set of snapping points. If any of **SNAPX**, **SNAPY** or **SNAPZ** is less than or equal to zero, there will be no snapping coordinates in that direction.

The coordinates specified are in the Working Coordinate System.

**SKETCH** is only intended to be used with the GUI. Initial values for the visible or snapping grid could be set from the command line or in a command input file.

The following commands can be used while an object is being sketched:

- [The PICTURE Command \[page 281\]](#) to save the current display as a picture file.
- [The THREED Command \[page 324\]](#) to change the view of the model (including the corresponding mouse operations).
- [The WINDOW Command \[page 351\]](#) to hide or reveal parts of the display.

Any other command will cancel an incomplete sketch.

## The SOLENOID Command

---

### Summary

Create or modify solenoids.

### Toolbuttons



### Command line parameters

Command	SOLENOID		
Parameter	Default	Function	
OPTION	NEW	NEW	Creates a new solenoid conductor
		MODIFY	Modifies properties of the picked solenoid conductors
		LOAD	Loads defaults from picked conductors
DRIVELABEL		Name for the solenoid drive label	
LCNAME		Name for Local Coordinate System for coordinate system 1	
SYMMETRY		Rotational symmetry about global Z axis	
XCEN2		Origin of coordinate system 2	
YCEN2			
ZCEN2			
THETA2		Euler angles defining orientation of coordinate system 2	
PHI2			
PSI2			
RXY		Reflection symmetries in XY, YZ and ZX planes	
RYZ			
RZX			
XP1		First point on the solenoid cross-section	
YP1			

Command	SOLENOID	
Parameter	Default	Function
XP2		Second point on the solenoid cross-section
YP2		
XP3		Third point on the solenoid cross-section
YP3		
XP4		Fourth point on the solenoid cross-section
YP4		
CU1	0	Curvatures of the 4 sides of the cross-section
CU2	0	
CU3	0	
CU4	0	
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>
TOLERANCE	0	Field calculation tolerance
INCIRCUIT	NO	Is the conductor part of a circuit: NO      The conductor has defined current density. YES     The current in the conductor is determined by a circuit.
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.
CIRCUITELEMENT		The name of circuit element this conductor is part of.
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation
MODELCOMPONENT		NO      Do not convert to meshable cells. REGULAR   Convert to meshable cells which will be regularly meshed. YES      Convert to meshable cells which meet the mesh size criteria.
MESHSIZE		The size of the mesh to be used when meshing
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.
KEEP	NO	NO      Clear the list of picked items YES     Keep the list of picked items for further modification

## Notes

This command creates a new solenoid or modifies existing solenoids.

- **OPTION=NEW** creates a new solenoid.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked solenoids. If the picked solenoids do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking solenoids (see [The PICK Command \[page 277\]](#)) to be modified and before **SOLENOID OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked solenoids to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Solenoids \[page 514\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **SOLVERS** Command

---

### Summary

Creates a new database, or adds a new simulation to an existing database.

### Toolbutton



### Command line parameters

Command	<b>SOLVERS</b>	
Parameter	Default	Function
<b>FILE</b>		Name of the database file.
<b>OPTION</b>	<b>NEW</b>	Command option:
		<b>ADD</b> Add a new simulation to an existing database file.
		<b>NEW</b> Create a new database file.
		<b>OVERWRITE</b> Create a new database, overwrites existing files.
		<b>TEST</b> Test the existence of database and model files (GUI use only).
<b>UNITS</b>	<b>CGS</b>	Specify the unit set to be used to write the database:
		<b>CGS</b> Mixed CGS units.
		<b>INCH</b> SI units with inches.
		<b>METRE</b> SI units.
		<b>MICRON</b> SI units with microns.
		<b>MM</b> SI units with mm.
		<b>TRUECGS</b> True CGS units.
<b>ELEMENT</b>	<b>MIXED</b>	Type of elements to be created:
		<b>MIXED</b> Use element type defined by cell property.
		<b>LINEAR</b> Use all linear elements.
		<b>QUADRATIC</b> Use all quadratic elements.

Command	<b>SOLVERS</b>	
Parameter	Default	Function
<b>SURFACE</b>	<b>CURVED</b>	Type of elements to be created in all elements that touch surfaces:
		<b>CURVED</b> As mixed, but curved surfaces default to quadratic.
		<b>LINEAR</b> Use all linear elements.
		<b>QUADRATIC</b> Use all quadratic elements.
<b>SOLVENOW</b>	<b>NO</b>	Start analysis option:
		<b>NO</b> Only write the database.
		<b>YES</b> Write the database and add the analysis job to the Opera Manager batch queue.
<b>SAVEMODEL</b>	<b>YES</b>	<b>NO</b> Do not save an OPC file.
		<b>YES</b> Save an OPC file of the same name as the OP3 database.
<b>PROGRAM</b>		Obsolete.

## Notes

The **SOLVERS** command creates a new database or adds a new simulation to an existing database (**op3**) file.

There are 4 **OPTIONS**:

- **NEW** creates a new database. It is an error if the database already exists. If multiphysics has been selected, the number of simulations added to the database will correspond to the number of multiphysics stages (see [The MULTIPHYSICS Command \[page 269\]](#)); otherwise one simulation will be added.
- **OVERWRITE** replaces an existing database.
- **ADD** creates one or more new simulations to an existing database.
  - multiphysics simulations: the new stages must first be added to the end of the set of analyses defined by the [The MULTIPHYSICS Command \[page 269\]](#). Any stages in the multiphysics set not already in the database will be added to the database.
  - other simulations: one simulation will be added.
- **TEST** is used by the GUI to check the existence of the database and if necessary ask the user if it should be overwritten. If the user's response is that the file should not be overwritten, the sequence of commands issued by the GUI will terminate. **TEST** cannot be used from the command line.

The **SAVEMODEL** parameter allows a geometry model file of the same name as the **op3** database to be created as well.

The type of analysis to be used is determined from the active analysis type. See "The ANALYSISDATA Command" on page 122.

When creating a new database, the type of elements are controlled by the ELEMENT and SURFACE parameters. The ELEMENT parameter allows the type of volume elements in cells to be controlled. The SURFACE parameter allows the type of volume elements that touch a face to be controlled. The analysis programs which use edge elements (Dynamic Electromagnetic, Motional Electromagnetic, Magnetization and High Frequency) are limited to linear elements. However quadratic elements can be used to improve the representation of curved surfaces in the Post-Processor.

When adding a simulation to a database, the UNITS, ELEMENT and SURFACE parameters are not used.

The SOLVENOW option creates the database and adds the analysis job to the Opera Manager batch queue. See the ***Opera Manager User Guide*** for details of the batch queue.

## Units

The unit sets are defined as follows:

Unit Set	True CGS	Mixed CGS	SI	SI (mm)	SI (micron)	SI (inch)
length	cm		m	mm	μm	inch
magnetic flux density	gauss			tesla		
magnetic field strength	oersted			amp/m		
magnetic scalar potential	oersted cm			amp		
magnetic vector potential	gauss cm			weber/m		
conductivity	siemens/cm	siemens/m	siemens/mm	siemens/μm	siemens/inch	
current density	amp/cm <sup>2</sup>	amp/m <sup>2</sup>	amp/mm <sup>2</sup>	amp/μm <sup>2</sup>	amp/inch <sup>2</sup>	
power	erg/s		watt			
force	dyne					
energy	erg					
electric field strength	volt/cm		volt/m			
electric flux density	coulomb/cm <sup>2</sup>		coulomb/m <sup>2</sup>			

Unit Set	True CGS	Mixed CGS	SI	SI (mm)	SI (micron)	SI (inch)
<b>mass</b>	gram		kg			
<b>pressure</b>	dyne/cm <sup>2</sup>		pascal			
<b>temperature</b>	Any unit can be used for temperature (kelvin, celsius or fahrenheit) as long as every quantity in the model which involves temperature uses the same unit.					

## The SPHERE Command

---

### Summary

Creates a sphere.

### Toolbutton



### Command line parameters

Command	<b>SPHERE</b>	
Parameter	Default	Function
<b>NAME</b>		Attaches this name to the body formed
<b>X0</b>		X coordinate of centre
<b>Y0</b>		Y coordinate of centre
<b>Z0</b>		Z coordinate of centre
<b>RADIUS</b>		Radius of the sphere
<b>MATERIALLABEL</b>		Set the material label of the new sphere
<b>LEVEL</b>		Set the data storage level of the new sphere
<b>UNIQUENAME</b>		Force a specific unique name for the sphere created

### Notes

A sphere of the given **RADIUS** is formed, centred on the coordinate (**Z0**, **Y0**, **Z0**).

The coordinates specified are in the Working Coordinate System.

The single **CELL** created by the **SPHERE** command is given a material label and data storage level if specified in the **MATERIALLABEL** and **LEVEL** parameters. See "The CELldata Command" on page 168.

When a sphere is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

## The **STRAIGHT** Command

---

### Summary

Create or modify straights.

### Toolbuttons



### Command line parameters

Command	<b>STRAIGHT</b>		
Parameter	Default	Function	
OPTION	NEW	NEW	Creates a new straight conductor
		MODIFY	Modifies properties of the picked straight conductors
		LOAD	Loads defaults from picked conductors
DRIVELABEL		Name for the drive label	
LCNAME		Name for Local Coordinate System for coordinate system 1	
SYMMETRY		Rotational symmetry about global Z axis	
XCEN2		Origin of coordinate system 2	
YCEN2			
ZCEN2			
THETA2		Euler angles defining orientation of coordinate system 2	
PHI2			
PSI2			
RXY		Reflection symmetries in XY, YZ and ZX planes	
RYZ			
RZX			
WIDTH		Local x cross-sectional width	
THICKNESS		Local y cross-sectional height	

Command	<b>STRAIGHT</b>		
Parameter	Default	Function	
H1		Length of the straight	
CURD		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
TOLERANCE		Field calculation tolerance	
INCIRCUIT		Is the conductor part of a circuit:	
	NO	The conductor has defined current density.	
	YES	The current in the conductor is determined by a circuit.	
REVERSE		Reverse the connections to this conductor in its circuit: YES or NO.	
CIRCUITELEMENT		The name of circuit element this conductor is part of.	
GROUPLABEL		The group of which the conductor is a part for a Motional EM simulation	
MODELCOMPONENT		NO	Do not convert to meshable cells.
	REGULAR	Convert to meshable cells which will be regularly meshed.	
	YES	Convert to meshable cells which meet the mesh size criteria.	
MESHSIZE		The size of the mesh to be used when meshing	
MESHLFACTOR		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
KEEP	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new straight conductor or modifies existing straights.

- **OPTION=NEW** creates a new straight conductor.
- **OPTION=LOAD** updates the default values of the command parameters to match common values shared by all of the picked straights. If the picked straights do not have a common value for a parameter, that parameter is left unset. **OPTION=LOAD** must be used after picking straights (see [The PICK Command \[page 277\]](#)) to be modified and before **STRAIGHT OPTION=MODIFY**.
- **OPTION=MODIFY** changes the conductor data of all of the picked straights to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Straight Bars \[page 522\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

## The **STRETCH** Command

### Summary

Stretches the geometry of bodies.

### Toolbutton



### Command line parameters

Command	STRETCH		
Parameter	Default	Function	
U1		Position of the starting point of the stretch	
V1			
W1			
DISP1		Displacement of the starting point	
U2		Position of the end point of the stretch	
V2			
W2			
DISP2		Displacement of the end point	
KEEP	NO	NO	Keep the picked body for further operations
		YES	

### Notes

The **STRETCH** command allows bodies to be transformed to a new shape by stretching space. The stretch is defined by 2 coordinates and a displacement at each. A positive displacement indicates a shift along the vector from point 1 to point 2.

Any part of picked bodies that lie between these 2 coordinates will be stretched.

## The **SWEEP** Command

---

### Summary

Sweeps a picked face through space to form a new body, or to extend the existing body owning the face.

### Toolbutton



### Command line parameters

Command	<b>SWEEP</b>		
Parameter	Default	Function	
<b>TYPE</b>	<b>DISTANCE</b>	<b>DISTANCE</b>	Sweep a fixed distance along the face normal
		<b>VECTOR</b>	Sweep along a specified vector
		<b>ROTATE</b>	Sweep a fixed angle about an axis
		<b>PATH</b>	Sweep along a path given by edge or wire body.
<b>DISTANCE</b>		Distance to be swept through, <b>TYPE=DISTANCE</b> only	
<b>DU</b>		Vector to sweep along, <b>TYPE=VECTOR</b> only	
<b>DV</b>			
<b>DW</b>			
<b>ROTU</b>		Axis of rotation, <b>TYPE=ROTATE</b> only	
<b>ROTV</b>			
<b>ROTW</b>			
<b>POSU</b>		Point on the axis of rotation, <b>TYPE=ROTATE</b> only	
<b>POSV</b>			
<b>POSW</b>			
<b>ANGLE</b>		Angle of rotation, <b>TYPE=ROTATE</b> only	
<b>REGULARIZE</b>	<b>NO</b>	<b>NO</b>	Leaves the starting face
		<b>YES</b>	Removes the original face if part of a body

Command	SWEEP		
Parameter	Default	Function	
DRAFTTYPE	NONE	ANGLE	The draft angle for the hole is equal and opposite to that specified for the face periphery
		HOLEANDANGLE	The draft angles for the hole and periphery may vary independently
		HOLEWITHANGLE	Use if there are no holes in the sweep face and also if there are but they should taper in the same way as the face periphery
		NONE	No draft during the sweep
DRAFHOLEANGLE		The hole draft angle	
DRAFTANGLE		The periphery draft angle	
TWISTTYPE	NONE	NONE	No twist
		ANGLE	Twist through a fixed angle
		PARAMETRIC	Twist through a angle specified as a function of position along the edge
TWISTANGLE		Fixed angle of twist	
TWISTEXP		Parametric expression defining twist	
RIGID	NO	NO	Keep the face normal fixed during the sweep.
		YES	
KEEP	NO	NO	Keep the swept face picked.
		YES	

## Notes

This command will sweep picked faces to extend a solid volume, or to create a solid if sweeping a sheet face. Multiple faces can be swept, but each face must belong to a different body.

The sweep is controlled by the **TYPE** parameter.

- **TYPE=DISTANCE** will sweep the face along its normal. Vectors representing the normal direction can be viewed on picked faces with the **VECTOR** command. This normal direction can be toggled by picking one of these vectors. Only planar faces can be swept using this option, as it is necessary to identify a unique direction vector.
- **TYPE=VECTOR** will sweep along the specified vector.
- **TYPE=ROTATE** allows sweeping about an angle around an axis passing through a point.
- **TYPE=PATH** allows sweeping along an edge or wire body (i.e. a body comprised of only wire edges). To perform this operation both a face and edge or face and wire body should be picked.

For **TYPE=VECTOR**, **TYPE=ROTATE** and **TYPE=PATH**, care should be taken to avoid sweeping a face where the normal to the face is almost perpendicular to the direction.

For **TYPE=PATH**, if the path being swept does not start on the plane of the face, the sweep will occur in both directions up to the start and finish of the path. If the path does not intersect the plane of the face, unpredictable results may occur.

A twist may be specified for **TYPE=PATH**. This twist uses the path as the centre of twist, and may be specified as an angle (in degrees) or as a parametric expression of the distance along the edge, **U**. It is not possible to describe the underlying algorithm used by the geometric modelling kernel when sweeping with parametric twists, so this facility should be used with care.

Care should also be taken to avoid sweeping a face into another part of the same body. This can cause errors in the body formed. Some of these difficulties may be resolved, but other operations may progress and form inconsistent bodies. Such bodies will be high-lighted using the **CHECK** command, but cannot be easily corrected.

When sweeping, the area of the face can be changed using the draft angle.

The coordinates specified are in the Working Coordinate System.

## The **TANGENTIALCPE** Command

---

### Summary

Create or modify tangential constant perimeter end (CPE) conductors.

### Toolbuttons



### Command line parameters

Command	<b>TANGENTIALCPE</b>			
Parameter	Default	Function		
OPTION	NEW	NEW	Creates a new tangential CPE conductor	
		MODIFY	Modifies properties of the picked tangential CPE conductors	
		LOAD	Loads defaults from picked conductors	
DRIVELABEL		Name for the drive label		
LCNAME		Name for Local Coordinate System for coordinate system 1		
SYMMETRY		Rotational symmetry about global Z axis		
XCEN2		Origin of coordinate system 2		
YCEN2				
ZCEN2				
THETA2		Euler angles defining orientation of coordinate system 2		
PHI2				
PSI2				
RXY		Reflection symmetries in XY, YZ and ZX planes		
RYZ				
RZX				
ALPHA		Azimuthal angular position of the straight		
BETA		Cutter angle		

Command	<b>TANGENTIALCPE</b>		
Parameter	Default	Function	
<b>WIDTH</b>		Radial cross-sectional size	
<b>THICKNESS</b>		Azimuthal cross-sectional size	
<b>H1</b>		Half-length of the straight	
<b>R1</b>		Radius of forming cylinder	
<b>R2</b>		Radius of cross-over arc	
<b>CURD</b>		Current density in the conductor. This can be defined in terms of the current as <b>current/AREA</b>	
<b>TOLERANCE</b>		Field calculation tolerance	
<b>INCIRCUIT</b>		Is the conductor part of a circuit:  NO      The conductor has defined current density.  YES     The current in the conductor is determined by a circuit.	
<b>REVERSE</b>		Reverse the connections to this conductor in its circuit: YES or NO.	
<b>CIRCUITELEMENT</b>		The name of circuit element this conductor is part of.	
<b>GROUPLABEL</b>		The group of which the conductor is a part for a Motional EM simulation	
<b>MODELCOMPONENT</b>		NO	Do not convert to meshable cells.
		REGULAR	Convert to meshable cells which will be regularly meshed.
		YES	Convert to meshable cells which meet the mesh size criteria.
<b>MESHSIZE</b>		The size of the mesh to be used when meshing	
<b>MESHLFACTOR</b>		Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	
<b>KEEP</b>	NO	NO	Clear the list of picked items
		YES	Keep the list of picked items for further modification

## Notes

This command creates a new tangential CPE conductor when using **OPTION=NEW**.

**OPTION=MODIFY** it is used to operate on the list of picked tangential CPE conductors, changing geometry, coordinate systems, symmetries, current density, tolerance etc. The current common values of the picked tangential CPE conductors are available by calling **CONDUCTOR OPTION=LOAD**.

The default values of the command parameters are updated to match common values shared by all of the picked tangential CPE conductors. If the picked tangential CPE conductors do not have a common value for a parameter, that parameter is left unset. Using **OPTION=MODIFY** will change the conductor data of all of the picked tangential CPE conductors to the new values given in the parameters, but will not affect the conductor data of any unset parameter.

For more information on the parameters, see [The CONDUCTOR Command \[page 189\]](#) and [Constant Perimeter Ends \[page 520\]](#). For more information on connecting conductors to circuits, see [The CIRCUIT Command \[page 173\]](#).

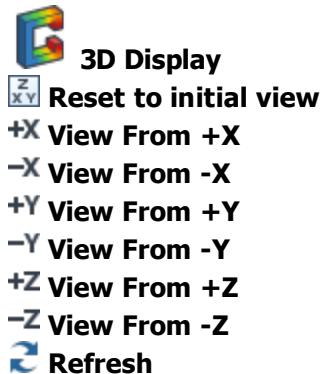
## The **THREED** Command

---

### Summary

Controls the display of the geometry.

### Toolbuttons



### Command line parameters

Command	<b>THREED</b>	
Parameter	Default	Function
<b>OPTION</b>		Command option:  GETVIEW    Retrieve view parameters after mouse interaction. INITIALIZE    Refresh picture without changing the view. SETVIEW    Refresh picture using the view parameters.
<b>SIZE</b>	10	Display extends from the origin by <b>SIZE</b> in each direction. <b>SIZE=0</b> requests the initial view of the model
<b>ROTX</b>	20	Rotation of model around X axis.
<b>ROTY</b>	20	Rotation of model around Y axis.
<b>ROTZ</b>	0	Rotation of model around Z axis.
<b>XORIGIN</b>	0	X coordinate at centre of picture
<b>YORIGIN</b>	0	Y coordinate at centre of picture
<b>ZORIGIN</b>	0	Z coordinate at centre of picture

Command	THREED		
Parameter	Default	Function	
<b>PERSPECTIVE</b>	<b>YES</b>	Perspective switch:	
		<b>YES</b>	Perspective view.
		<b>NO</b>	Orthographic view.
<b>LINECOLOUR</b>	<b>YES</b>	Colour used for outlines:	
		<b>YES</b>	Material colour.
		<b>NO</b>	Edge colour.
<b>XASPECT</b>	1	Scaling for X coordinates	
<b>YASPECT</b>	1	Scaling for Y coordinates	
<b>ZASPECT</b>	1	Scaling for Z coordinates	
<b>FACETANGLE</b>	10	Largest angle subtended by a curved facet.	

## Notes

The **THREED** command updates the 3D picture of the model. The pictures consists of the three dimensional geometry of the model and conductors. The view can be adjusted using the mouse buttons or by setting explicitly using this command.

- **OPTION=GETVIEW**: updates the values of the parameters **SIZE**, **ROTX**, **ROTY**, **ROTZ**, **XORIGIN**, **YORIGIN**, **ZORIGIN**, **XASPECT**, **YASPECT** and **ZASPECT**.
- **OPTION=SETVIEW**: uses the current values of **SIZE**, **ROTX**, **ROTY**, **ROTZ**, **XORIGIN**, **YORIGIN**, **ZORIGIN**, **XASPECT**, **YASPECT** and **ZASPECT**. The "initial view" of a model can be obtained using **SIZE=0**.

When a local coordinate system is active (see [The LCS Command \[page 240\]](#)), these parameters are interpreted with respect to the local coordinate system.

- **OPTION=INIT**: updates the picture without changing the view. If automatic update is switched off using the command **SELECT AUTOUPDATE=NO**, this command option must be called to update the display.

Perspective view can be switched off using **PERSPECTIVE=NO**.

The **FACETANGLE** parameter allows the user to request more precise display of curved surfaces. The Modeller subdivides each curved face into triangular facets limiting the angles between the normals of adjacent facets to **FACETANGLE**. Reducing **FACETANGLE** will increase the number of facets so that they better match the curved surface. This will also slow down the display. **FACETANGLE** is ignored if the surface mesh has been generated using [The MESH Command \[page 257\]](#); the finite element facets will be used for the display.

The parameters **XASPECT**, **YASPECT** and **ZASPECT** can be adjusted to obtain views of models with high aspect ratios. The values should be between 0 and 1. For example, a pipe of radius 1 unit and

length 100 units could be displayed with **XASPECT** and **YASPECT** left at the default value of 1 and **ZASPECT=0.01**.

## The **TITLE** Command

---

### Summary

Adds title, date and time to the display.

### Command line parameters

Command	<b>TITLE</b>	
Parameter	Default	Function
<b>STRING</b>	<i>none</i>	A graphics window title.
<b>POSITION</b>	<b>NONE</b>	Title position:  BOTMOMCENTRE Bottom centre BOTMOMLEFT Bottom left BOTMOMRIGHT Bottom right NONE No title TOPCENTRE Top centre TOPLEFT Top left TOPRIGHT Top right
<b>DATE</b>	<b>TOPLEFT</b>	Time/date position:  BOTMOMCENTRE Bottom centre BOTMOMLEFT Bottom left BOTMOMRIGHT Bottom right NONE No date and time TOPCENTRE Top centre TOPLEFT Top left TOPRIGHT Top right

### Notes

The **TITLE** command controls the display of a title and the date and time. There is a choice of 6 positions for each, or **NONE** to omit that item. If the same position is chosen for both the title and the date, the title appears above the date and time.

The version number of the software can be included in the title using the string and system variables, **VERSION** using:

```
title string='Version: &version&'
```

or

```
title string='Version: %real(version,5)'
```

## The **TORUS** Command

---

### Summary

Creates a torus.

### Toolbutton



### Command line parameters

Command	<b>TORUS</b>	
Parameter	Default	Function
<b>NAME</b>		Attaches this name to the body formed
<b>X0</b>		X coordinate of centre
<b>Y0</b>		Y coordinate of centre
<b>Z0</b>		Z coordinate of centre
<b>MAJORRADIUS</b>		Major radius of the torus
<b>MINORRADIUS</b>		Minor radius of the torus
<b>MATERIALLABEL</b>		Set the material label of the new torus.
<b>LEVEL</b>		Set the data storage level of the new torus
<b>UNIQUENAME</b>		Force a specific unique name for the torus created.

### Notes

A torus with the given radii is formed, centred on the coordinate (**X0,Y0,Z0**).

The coordinates specified are in the Working Coordinate System.

The single **CELL** created by the **TORUS** command is given a material label and data storage level if specified in the **MATERIALLABEL** and **LEVEL** parameters. See "The CELldata Command" on page 168.

When a torus is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

## The **TRANSFORM** Command

---

### Summary

Transforms or copies bodies, cells, faces, edges, local coordinate systems and conductors.

### Toolbutton



### Command line parameters

Command	<b>TRANSFORM</b>		
Parameter	Default	Function	
<b>OPTION</b>	<b>APPLY</b>	<b>APPLY</b>	Apply the transformation to the picked items
		<b>COPY</b>	Creates a copy of the new item and applies the transformation to it
		<b>WCS</b>	Transform the current WCS.
<b>TYPE</b>	<b>DISPLACE</b>	<b>DISPLACE</b>	Use a displacement transformation
		<b>ROTATE</b>	Use rotation about an axis
		<b>REFLECT</b>	Reflect in a plane
		<b>SCALE</b>	Apply anisotropic scaling
		<b>EULER</b>	Rotate by Euler angle set
<b>DU</b>	0	Translation component in the working coordinate system	
<b>DV</b>	0		
<b>DW</b>	0		
<b>ROTU</b>	0	Rotation axis in the working coordinate system	
<b>ROTV</b>	0		
<b>ROTW</b>	1		
<b>ANGLE</b>	0	Angle of rotation about the given axis	

Command	<b>TRANSFORM</b>		
Parameter	Default	Function	
NU	0	Normal vector to the plane of reflection	
NV	0		
NW	1		
SCU	1	Scaling factors in the working coordinate system	
SCV	1		
SCW	1		
THETA	0	Values of Euler angle rotations	
PHI	0		
PSI	0		
LABEL		Adds a label to each item transformed or copied.	
COUNT	1	Number of copies to be created	
KEEP	NO	NO	Reset the list of picked objects
		YES	Keep the list of picked objects
COPYIDENTIFIER		Allows adjustment of the unique name of bodies	

## Notes

The **TRANSFORM** command is used to transform the geometry of existing items, or to create copies and transform them.

The **TYPE** of transformation determines which of the parameter values are used. All others are ignored. The **COUNT** parameter is only used when copying, and allows multiple copies to be produced from a single command.

Type	Parameters
DISPLACE	DU, DV, DW
ROTATE	ROTU, ROTV, ROTW, ANGLE
REFLECT	NU, NV, NW
SCALE	SCU, SCV, SCW
EULER	THETA, PHI, PSI

## Transform

**OPTION=APPLY**, transforms the picked entities (bodies or faces).

When the face of a body is transformed, the underlying surface is changed. Other faces of the body are then extended or reduced in order to meet the new surface and form the edges of the new face. Not all transformations will work (e.g. reflect or rotate by 90° or more) because they would invalidate the topology of the body.

### Copying bodies, cells faces, edges

**OPTION=COPY** creates copies of entities (bodies, cells, faces, edges) and then transforms the copies.

If only cells have been picked, the **OPTION=COPY** operation can be used to create a new body for each cell that is copied.

If only faces have been picked, the **OPTION=COPY** operation can be used to create a new body for each face. This body then contains only a copy of this face. The body formed will enclose no volume, but the face may be selected and used in sweep or other operations.

If only edges have been picked, the **OPTION=COPY** operation can be used to create a new body for each edge. This body then contains only a copy of this edge. The body formed will enclose no volume, but the edge may be selected and used in sweep or other operations or used as a wire edge.

### Working Coordinate System

**OPTION=WCS** allows the existing working coordinate system to be transformed to a new location. Where the WCS is the global coordinate system, a new local coordinate system is created and made the working coordinate system.

If any Local Coordinate Systems or conductors are picked or when **OPTION=WCS**, the **SCALE** and **REFLECT** transformations will not have any effect on these objects, as they must remain with the same handedness and size.

### Conductors

When transforming or copying conductors, the operator modifies either Local Coordinate System 1 or Local Coordinate System 2 of the conductor:

- If Local Coordinate System 1 is the current Working Coordinate System, Local Coordinate System 2 is changed.
- If Local Coordinate System 1 is not the Working Coordinate System, a copy of Local Coordinate System 1 is created and transformed, and the conductor's Local Coordinate System 1 is changed to point to the new Local Coordinate System.

### Other parameters

The transformed items will all have the **LABEL** attached as a user label. If copying, the original items will be given the label. All new items created by copying will be given the label suffixed by a count

from 1 to **COUNT**, i.e. labeltext1, labeltext2 etc. This allows the items created by copying to be easily grouped according to the number of times that the transformation was applied to the original items.

The **KEEP** parameter can be used if more than one transformation is to be applied to the picked set of objects, to avoid needing to re-pick them.

The **COPYIDENTIFIER** parameter allows bodies created during the operation to be given a new unique body name that is unlikely to conflict with other bodies present. If unset, a new identifier is generated from existing bodies to prevent any possible conflict. This copy identifier is set in the command history so naming of bodies during replay is consistent.

## The **TWIST** Command

---

### Summary

Twists the geometry of bodies.

### Toolbutton



### Command line parameters

Command	<b>TWIST</b>		
Parameter	Default	Function	
<b>U1</b>		Position of the starting point of the twist	
<b>V1</b>			
<b>W1</b>			
<b>ANGLE1</b>		Displacement of the starting point	
<b>U2</b>		Position of the end point of the twist	
<b>V2</b>			
<b>W2</b>			
<b>ANGLE2</b>		Displacement of the end point	
<b>CONTINUITY</b>		Level of continuity at the ends of the twist	
<b>KEEP</b>		<b>NO</b>	Keep the picked body for further operations
		<b>YES</b>	

### Notes

The **TWIST** command allows bodies to be transformed to a new shape by twisting space. The twist is defined by 2 coordinates and a twist at each of them. The twist takes place around an axis directly between the 2 points given.

Any part of picked bodies that lie between these 2 coordinates will be twisted. A continuity level (0, 1 or 2) can be specified to control the level of surface continuity if the twist region does not enclose all of the body. Level 0 will have a sharp join, while level 2 will enforce continuity of the second derivatives at the interface.

## The UNDO Command

### Summary

Rewinds the history to a specified position.

### Toolbutton



### Short cut

<Ctrl>+Z

### Command line parameters

Command	UNDO	
Parameter	Default	Function
STATE		Name of the state to which the history is to be undone
WAITDISPLAY	NO	NO
		YES
		Refresh the display immediately
		Wait until the next command before refreshing the display.

### Notes

This command returns the state of the model to a former position.

If no value for **STATE** is given, the operation rewinds one previous state.

Most operations add an entry to the history stream. This entry is given a unique **STATE** name (based upon the command name and the number of commands so far). Any state name within the list can be specified in the **STATE** parameter. This allows the model to be returned to any previous state by a single command.

After an **UNDO** command, [The REDO Command \[page 292\]](#) is available to go back to future positions in the history stream. The **REDO** command is available until a new entry is added into the history stream.

Note: If generating command scripts from log files generated during sessions, it is advisable to remove any **UNDO** or **REDO** commands that specify a state name, as the state name is dependent

upon the number of commands issued. This makes it *very sensitive* to modifications earlier in the script.

The **WAITDISPLAY** parameter allows refreshing of the model to be delayed in case further operations are to be implemented.

## The **VARIABLE** Command

---

### Summary

Sets a user defined variable.

### Toolbutton



### Command line parameters

Command	<b>VARIABLE</b>	
Parameter	Default	Function
<b>OPTION</b>	<i>none</i>	<b>MODELDIMENSION</b>
		<b>CONSTANT</b>
		<b>PARAMETER</b>
		<b>DELETE</b>
		<b>LIST</b>
<b>NAME</b>	<i>none</i>	Variable name
<b>VALUE</b>	<i>none</i>	Value or expression assigned to the variable
<b>DESCRIPTION</b>	<i>none</i>	Description of the variable

### Notes

This command creates and changes the values of user defined variables. Such variables must begin with the '#' character and be a total length of not more than 64 characters.

There are 3 types of variable available:

- **CONSTANT**: Holds a constant value. The creation of this variable is stored in the history stream. Undo to an earlier history state will mean that the variable is no longer available.
- **PARAMETER**: Holds an expression that is re-evaluated every time it is used. As with constants, parameters are stored in the history stream.
- **MODELDIMENSION**: A model dimension is a constant valued variable. It is different from a constant because it is not stored in the history stream. Model dimensions should be used if defining or changing the value of variables that are to be used during replay. Undo or replay of an earlier history state will not affect the current value or existence of the variable. In contrast, constants or

parameters will be reverted back to earlier values or may no longer exist for the command that is being replayed. For more information on command replay, See "The REPLAY Command" on page 294.

A constant or parameter variable can be redefined as a model dimension variable, but model dimension variables cannot be redefined to a different variable type.

The **DESCRIPTION** can be any character string of up to 80 characters. It can be used to annotate variables, especially model dimensions.

The command replaces the use of the **\$ MODELDIMENSION**, **\$ CONSTANT** and **\$ PARAMETER** commands, and allows history facilities for **UNDO/REDO**, as well as storage of the variables when loading and saving models.

**OPTION=DELETE** removes a stored variable, although this may not be possible if the variable forms part of an expression which is in use in the Modeller. Care should be taken when deleting a model dimension: if the model dimension is used in the definition of a body or an operation, [The REPLAY Command \[page 294\]](#) will fail unless the model dimension is redefined first.

**OPTION=LIST** shows the current set of variables and their values.

The **\$ MODELDIMENSION**, **\$ CONSTANT** and **\$ PARAMETER** commands can be used, but the effects are independent of the history, so cannot be explicitly undone. However, these commands can be significantly faster than the **VARIABLE** command and should be used if lots of variables are being defined or redefined.

These variables can be used as part of any expression, and so can help to automate command scripts or to assist in model definition by allowing common values to be defined.

## The **VECTOR** Command

---

### Summary

Displays vectors of boundary conditions and face and volume properties on the model.

### Toolbutton



### Command line parameters

Command	<b>VECTOR</b>	
Parameter	Default	Function
<b>COMPONENT</b>	<b>NONE</b> Vectors to be displayed. <b>NONE</b> No vectors Volume properties: <b>BODYFORCE</b> Body force density <b>CONDUCTOR</b> Conductor current <b>J</b> Current density set in the volume property of a cell <b>ORIENTATION</b> Local orientation set in the volume property of a cell <b>V</b> Linear velocity set in the volume property of a cell Face properties and boundary conditions: <b>A</b> Magnetic vector potential of a face boundary condition <b>DISPLACEMENT</b> Displacement <b>E</b> Electric field <b>FACEDIRECTION</b> Direction of faces for use with sweeping and emitters <b>INA</b> Incident magnetic vector potential <b>INE</b> Incident electric field <b>TRACTION</b> Surface traction	
<b>SCALE</b>	1	Scale factor for sizing the vectors
<b>MAGNITUDE</b>	<b>YES</b>	<b>NO</b> Display all vectors with <b>LENGTH=SCALE</b> <b>YES</b> Display vectors with their magnitude set by <b>COMPONENT*SCALE</b>

## Notes

This command displays vectors of the **COMPONENT** on displayed objects in the model. The component can be a face property, a boundary condition or a volume property. If an object has data containing the component, then the vectors of the data will be displayed on it. Other objects will be displayed without vectors.

**COMPONENT=FACEDIRECTION** vectors will only be displayed on selected faces. The direction of faces involved in boolean operations will be determined from the data storage level of data attached to the faces. The direction of faces can be toggled by double clicking on a displayed vector.

The vectors displayed can be scaled in size by the value of **SCALE**. Those vectors that have a magnitude can be displayed either with size given by the magnitude of the **COMPONENT** multiplied by **SCALE** (**MAGNITUDE=YES**), or with a constant size of **SCALE** (**MAGNITUDE=NO**).

## Vectors

The vectors are displayed as cones which point in the appropriate direction. In a few cases, e.g. **ORIENTATION**, the direction is the z direction of a local coordinate system. To identify the other coordinate directions, a small fin on the cone points in the local x direction.

Figure 3.6 below shows a vector with the local z corresponding to global y and the local direction X parallel to global Z.

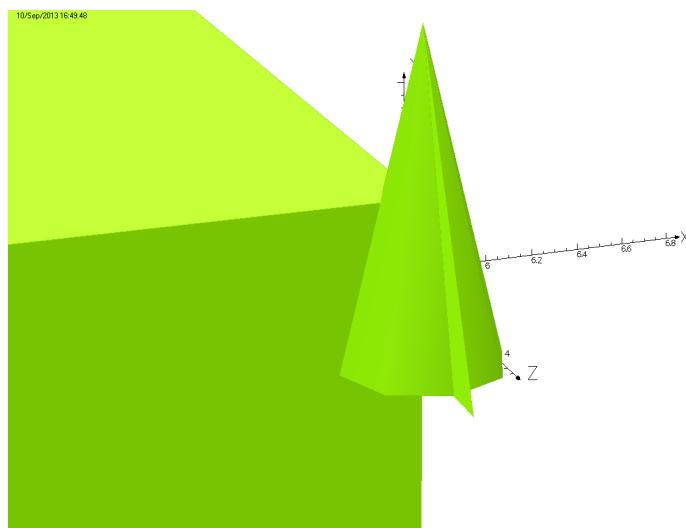


Figure 3.6 A Vector showing the local Z direction parallel to global Y and the local direction X parallel to global Z

## The VERTEXDATA Command

---

### Summary

Sets properties of picked vertices.

### Toolbutton



### Command line parameters

Command	VERTEXDATA		
Parameter	Default	Function	
<b>OPTION</b>	<b>MODIFY</b>	<b>MODIFY</b>	Applies new values to the picked vertices
		<b>RESET</b>	Clears all data from the picked vertices
<b>SIZE</b>	See notes	Mesh control size	
<b>LEVEL</b>	See notes	Data storage level for the vertex data	

### Notes

This command is used to set or clear the properties of all picked vertices. Vertices initially have no data assigned to them.

If issuing the command with **OPTION=MODIFY**, the new value of parameters that have been set replace the existing values of data on the vertices. The value of data, associated with any of the unset parameters, is left unchanged.

The default value of each of the parameters is updated to be the common value of all the picked vertices. If the data of one of these parameters is unset, or the picked vertices do not share the same value, then the parameter value is left clear.

**SIZE** sets the maximum element side length of any element touching that vertex.

Upon issuing the command with **OPTION=MODIFY**, all picked items are deselected. Issuing the command with **OPTION=RESET** keeps the same set of picked objects.

The **LEVEL** parameter controls the storage of data when there is a conflict during the merging of multiple cells. The data set with the greater level will be maintained. The result of merging 2 cells with the same level is indeterminate.

## The **VOLUME** Command

---

### Summary

Sets the properties associated with a volume label.

### Toolbutton



### Command line parameters

Command	<b>VOLUME</b>	
Parameter	Function	
<b>OPTION</b>	<b>PICK</b>	Adds a volume label to a list of labels to be set.
	<b>UNPICK</b>	Clears the list of volume labels to be set.
	<b>RESET</b>	Clears the data from the picked labels.
	<b>MODIFY</b>	Sets the data for the picked labels.
	<b>DELETE</b>	Deletes the picked labels.
<b>VOLUMELABEL</b>	Volume label to be picked for modification.	
<b>VX</b>	Components of linear velocity for Velocity EM.	
<b>VY</b>		
<b>VZ</b>		
<b>JX</b>	Components of volume source current density.	
<b>JY</b>		
<b>JZ</b>		
<b>THETA</b>	Components of orientation for anisotropic material properties and easy direction of magnetization in permanent magnets.	
<b>PHI</b>		
<b>PSI</b>		
<b>PACKING</b>	Packing factor for packed materials.	
<b>ROTATION</b>	Rotational velocity [rpm].	
<b>CHARGE</b>	Volume charge density [ $C/length\_unit^3$ ].	

Command	<b>VOLUME</b>
Parameter	Function
<b>HEATTYPE</b>	Source of heat for Thermal analysis:  NONE No heat source. VALUE Specified functional heat source. NODALTABLE Heat source is to be read from a table of values in the <b>op3</b> database, <b>RNODALHEAT</b> . ELEMENTTABLE Heat source is to be read from a table of values in the <b>op3</b> database, <b>RELEMENTHEAT</b> .
<b>HEATVALUE</b>	The functional specification of the heat source.
<b>BODYFORCEX</b>	Components of body force density for Static Stress.
<b>BODYFORCEY</b>	
<b>BODYFORCEZ</b>	

## Notes

This command defines the volume characteristics for use by the analysis programs.

A set of volume labels is picked using the command repeatedly, with **OPTION=PICK** and a **VOLUMELABEL** specified. A volume label can be removed from the set using **OPTION=UNPICK**. If no **VOLUMELABEL** is given, the set is emptied.

Issuing the command with **OPTION=MODIFY** will modify the properties of the set of picked volume labels to the new values given in the parameters. The value of a property associated with the volume labels is unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked volume labels. If the data of one of these parameters is unset, or the picked volume labels do not share the same value, then the parameter value is left clear.

**OPTION=RESET** will clear the properties associated with all of the picked volume labels.

The properties of all volumes with the label **VOLUMELABEL** can be listed using **OPTION=LIST**.

Volume labels that are not used, i.e. have no cell referencing them can be deleted using **OPTION-N=DELETE**. Deleting a volume label that is in use will reset its properties.

All parameters can be specified. Some only apply to specific analysis modules.

See [Labels \[page 238\]](#) for more information on valid labels.

## Velocity

The Fixed Velocity Electromagnetic solver can solve linear or rotational velocities.

- The parameters **VX**, **VY** and **VZ** specify the linear velocity in the local coordinate system.
- **ROTATION** specifies the rotational velocity in rpm around the global Z axis.

## Current density

**JX**, **JY** and **JZ** specify current density in the local coordinate system for use in Electromagnetic and Magnetization solvers.

## Packing factor

**PACKING** specifies a packing factor which will be used if the material properties defined using [The MATERIALS Command \[page 248\]](#) specify **ANISOTROPIC=PACKED**. The laminations will be in parallel to the XY plane of the local coordinate system (except for thin plate laminations which are always parallel to the boundary condition surface).

## Local coordinate system

**THETA**, **PHI** and **PSI** specify Euler angles which define the local coordinate system used for velocity, current density and laminations. Note that this local coordinate system also applies to anisotropic material properties defined in [The MATERIALS Command \[page 248\]](#). See [Euler Angles \[page 83\]](#) for the definition of Euler angles.

## Charge

**CHARGE** specifies a volume charge density for use in Electrostatics and Charged Particle solvers. The units are in  $C/length\_unit^3$ , where *length\_unit* is the length unit used when creating the database.

## Heat

**HEATTYPE** and **HEATVALUE** specify heat source for Thermal solvers.

When **HEATTYPE=NODALTABLE**, the Thermal analysis will require that a vector of heat density values at every node in the model is added into the database before the analysis. Similarly, if **HEATTYPE=ELEMENTTABLE**, the value of heat density at the centroid of each element must be added into the database. Values of heat can be calculated by an earlier stage of a multiphysics analysis (see [The MULTIPHYSICS Command \[page 269\]](#)) or added to the database using the Post-Processor (see [The TABLE Command \[page 817\]](#)).

## Body force

**BODYFORCEX**, **BODYFORCEY** and **BODYFORCEZ** specify body force densities for Static Stress. They can be specified using functions of system variables. In many cases the values will be element force densities from an electromagnetic field analysis calculated by an earlier stage of a multiphysics analysis (see [The MULTIPHYSICS Command \[page 269\]](#)) or added to the database using the Post-Processor (see [The TABLE Command \[page 817\]](#)).

## The WCS Command

---

### Summary

Sets the Working Coordinate System (WCS).

### Command line parameters

Command	WCS	
Parameter	Default	Function
<b>OPTION</b>		<b>SET</b> Set Working Coordinate System to Local Coordinate System given by <b>LCNAME</b>
		<b>PICKED</b> Set Working Coordinate System by the entities which have been picked
		<b>UNSET</b> Use the Global Coordinate System as the Working Coordinate System
<b>LCNAME</b>		Name of the Local Coordinate System to be set as Working Coordinate System
<b>COPYIDENTIFIER</b>		Allows adjustment of the unique name of local coordinate systems.

### Notes

This command controls the Working Coordinate System. The Working Coordinate System is a coordinate system in which many commands operate, and it can be used to assist in the definition of geometries that do not lie neatly in the Global Coordinate System. The commands affected are those that create new objects, i.e. **BLOCK**, **CYLINDER**, **PRISM**, **SPHERE**, **TORUS** and **LCS**. It also operates with the **TRANSFORM** and **SWEET** commands (when specifying positions, vectors and axes of rotation) and the morphing commands **MORPH**, **TWIST** and **STRETCH**.

- **OPTION=SET**: changes the Working Coordinate System to the Local Coordinate System given by **LCNAME**. Local Coordinate Systems can be defined using [The LCS Command \[page 240\]](#).
- **OPTION=UNSET**: unsets the Working Coordinate System, so that the Global Coordinate System will be used.
- **OPTION=PICKED**: operates on picked entities. If a single Local Coordinate System has been picked, this will become the current Working Coordinate System.

If other entities (face, edge or vertex) have been picked, the Working Coordinate System will be transformed to the picked entities as show in the table:

Picked Entities	New Origin	New Orientation
Vertex	At the vertex.	Unchanged
Edge	At the centre of the edge.	U parallel to the edge.
Face	At the centre of the face.	W normal to the face.
Face followed by edge	At centre of the edge.	U parallel to the edge. W normal to the face.
Edge followed by face	At the centre of the face.	U parallel to the edge. W normal to the face.
Face, edge and vertex in any order	At the vertex.	U parallel to the edge. W normal to the face.

If the Working Coordinate System is the Global Coordinate System, a new Local Coordinate System will be created with name based on the **COPYIDENTIFIER**. This parameter allows local coordinate systems created by the **OPTION=PICKED** to be given a new unique name that is unlikely to conflict with other local coordinate systems present. This copy identifier is set in the command history so naming of local coordinate systems during replay is consistent.

## The **WINDING** Command

---

### Summary

Control the Winding Tool.

### Command line parameters

Command	<b>WINDING</b>	
Parameter	Default	Function
<b>OPTION</b>		Action of the command:
		<b>CALCULATE</b> Calculate the layout, winding factors and MMF.
		<b>CLOSE</b> Close the Winding Tool.
		<b>LOAD</b> Load a parameter set.
		<b>REMOVE</b> Remove a parameter set.
		<b>START</b> Start the Winding Tool.
<b>SET</b>	0	Parameter set identifier.
<b>POLES</b>	4	Number of poles in the winding.
<b>PHASES</b>	3	Number of phases in the winding.
<b>SLOTS</b>	12	Number of slots in the stator.
<b>SKEW</b>	0	Skew angle of the winding.
<b>LAYERS</b>	2	Number of layers required.
<b>HARMONIC</b>	16	Maximum harmonic order required.
<b>COILPITCH</b>	0	Coil pitch required.

### Notes

The **WINDING** command controls the Winding Tool which can be used to design the winding of a rotating machine. The **OPTIONs** are:

- **START** to open the Winding Tool.
- **CLOSE** to close the Winding Tool.
- **CALCULATE** to calculate a results **SET**, updating values of **SLOTS** and **POLES**.
- **LOAD** to return to a previously **CALCULATEd** set of results. Only the parameter **SET** needs to be provided to identify the set of parameters required. A command with this option is issued by the graphical Winding Tool using the current value of the set number when the **Load** button is pressed.

- If the results for a set of parameters with the requested **SET** value has previously been **CALCULATED** and has not been **REMOVEd**, the graphical Winding Tool will reload the existing values of all parameters and update the graphical displays accordingly. If called with a value of **SET**, for which a previous winding has not been calculated, a new default parameter set will be created with the requested set number and the graphical Winding Tool will be refreshed with these default values.
- **REMOVE** to remove a previously calculated parameter set. Only the parameter **SET** needs to be provided. A command with this option is issued by the graphical Winding Tool using the current value of the set number when the **Remove** button is pressed.
- If the provided value of **SET** is 0, the default parameter set will be returned to its default state. Otherwise the requested parameter set will be removed if it had previously been calculated. The graphical Winding Tool will be refreshed with the current values default parameter set, identified by **SET=0**.

## Parameters

All of the parameters for the **WINDING** command have associated bounds. Should the command be issued with an out of bounds parameter, an error will be reported. If the command is issued successfully, that does not necessarily imply that a valid winding can be calculated using the provided parameters; the associated system variables or OperaObject should be tested to verify validity. The bounds for the parameters are as follows:

- **SET** must be greater than or equal to 0.
- **POLES** must be greater than or equal to 2.
- **PHASES** must be greater than or equal to 1.
- **SLOTS** must be greater than or equal to 2.
- **SKEW** must be greater than or equal to 0 and less than or equal to 360/**SLOTS**. If a value above this upper bound is explicitly requested, an error will be reported. If a value above the upper bound is implicitly requested via the **WINDING** command zero will be used instead. This may happen if the number of **SLOTS** is increased thus reducing the maximum value of **SKEW**. Note this behaviour differs from that of the graphical Winding Tool where, if the number of slots is increased by such an amount as to make the current skew angle illegal, the current skew angle will be set to the maximum allowable value.
- **LAYERS** must be 1 or 2. When the **WINDING** command is issued from the command line, the supplied value of **LAYERS** is always used, regardless of single layer feasibility. The graphical Winding Tool will only use 1 layer when **Calculate** is pressed if the single layer tick box is checked and the currently visible parameters allow a feasible single layer winding. Otherwise, 2 layers will be used.
- **HARMONIC** must be greater than or equal to 0.
- **COILPITCH** must be greater than or equal to 0. For double layer windings, if a value of 0 is supplied, the average pitch,  $|SLOTS/POLES|$ , will be used. Otherwise, the supplied **COILPITCH** will be used when possible. For single layer windings, the actual coil pitch is computed automatically and this parameter has no effect.

## System Variables

After the **WINDING** command has successfully calculated results for set *m*, the following system variables will be available:

- **WINDING\_m\_DOUBLELAYERFEASIBLE** set to 1 if the winding is double layer feasible or 0 otherwise.
- **WINDING\_m\_SINGLELAYERFEASIBLE** set to 1 if the winding is single layer feasible or 0 otherwise.
- **WINDING\_m\_ACTUALCOILPITCH** set to the actual coil pitch used in the winding.

## OperaObject

After the **WINDING** command has successfully calculated results an **OperaObject** will be produced containing the results of the winding. The result object and its contents may be accessed using the Python **OperaObject** interface, for example,

```
WINDING OPTION=CALCULATE SET=0 PHASES=3 SLOTS=12 POLES=4
$PYTHON COMMAND='windingObject = operafea.lastCreatedObject() '
$PYTHON COMMAND='print(windingObject.getValue("VALID")) '
```

will assign the results object to the Python variable **windingObject** and print the value of its "**VALID**" variable.

The results object will contain the following:

- The value of the parameter **SET**, **SLOTS**, **PHASES**, **HARMONIC**, **SKEW**, **POLES**, **LAYERS** and **COILPITCH** in variables "**SET**", "**SLOT**", etc
- "**ACTUAL\_COIL\_PITCH**": the actual coil pitch used in the winding
- "**MAXIMUM\_SKEW**": the maximum skew angle permitted, defined as  $360/\text{SLOTS}$
- "**MACHINE\_PERIODICITY**": the number of pattern repetitions in the winding, providing the winding is completely constructed from a repeating pattern
- "**VALID**": either 1 if the winding is feasible, or 0 if the winding is not feasible

In addition, if the winding is feasible, the following harmonic and slot based information will be included in **NumPy** arrays inside the object.

### Harmonic information:

Harmonic orders and MMF harmonics are signed positive and negative appropriately, although when displayed in the graphical Winding Tool, only the absolute value is used.

- "**HARMONIC\_ORDERS**": the orders of the computed harmonics.
- "**WINDING\_FACTORS**": the computed winding factors for the computed harmonic orders.
- "**MMF**": the MMF harmonics for the computed harmonic orders.

### Slot based information:

Slots are numbered from 1 to the number of slots, although Python uses zero based array indexing and hence information for Slot N will be found at index N-1 of the arrays. Here the 1st entry refers to

the entry of the array at index 0, the second entry at index 1 and so on.

- "**LAYER1**": the phases of slots in the first layer.
- "**LAYER2**": the phases of slots in the second layer. All entries will be zero for single layer windings.
- "**CONNECT\_TO**": the  $N^{\text{th}}$  entry gives the number of the slot which slot N connects to.
- "**CONNECT\_FROM**": the  $N^{\text{th}}$  entry gives the number of the slot which slot N connects from.

As an example of connectivity, consider the following.

If the 4<sup>th</sup> entry of "**CONNECT\_TO**" is 10, slot 4 issues a coil to slot 10 and the 10<sup>th</sup> entry of "**CONNECT\_FROM**" will be 4. For double layer windings, this implies slot 4 of layer 1 connects to slot 10 of layer 2. For single layer windings, this implies a coil leaves slot 4 and returns in slot 10. A value of 0 in an entry of "**CONNECT\_TO**" indicates no coil leaves that slot and similarly a value of 0 in "**CONNECT\_FROM**" implies no coil returns to that slot. For a given slot in a single layer winding, exactly one of the corresponding entries in "**CONNECT\_TO**" and "**CONNECT\_FROM**" will be non-zero.

## More Information

Full details of how to interact with the winding tool and the different ways the results calculated can be displayed and used are given in the ***Opera-3d User Guide***.

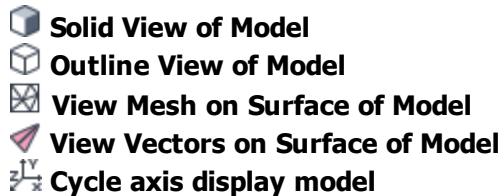
## The **WINDOW** Command

---

### Summary

Show or hide parts of the display.

### Toolbuttons



### Command line parameters

Command	<b>WINDOW</b>	
Parameter	Default	Function
<b>AXES</b>	<b>YES</b>	Show coordinate axes: <b>MAJOR</b> , <b>TRIAD</b> , <b>ALL</b> or <b>NONE</b> or <b>CYCLE</b> around options.
<b>SOLID</b>	<b>YES</b>	Show solid view of model: <b>YES</b> or <b>NO</b> .
<b>OUTLINE</b>	<b>YES</b>	Show outline view of model: <b>YES</b> or <b>NO</b> .
<b>VECTORS</b>	<b>YES</b>	Show vectors on the surface of the model: <b>YES</b> or <b>NO</b> .
<b>MESH</b>	<b>YES</b>	Show surface mesh <b>YES</b> or <b>NO</b> .

### Notes

The **WINDOW** command can be used to hide or show again parts of the display which exist. For example, if a surface mesh exists, it can be hidden using **WINDOW MESH=NO** or shown again using **WINDOW MESH=YES**.

**AXES=CYCLE** cycles between the options:

- **MAJOR** to show the major axes for the active coordinate system;
- **TRIAD** to show the triads at the origins of the global and local coordinate systems;
- **ALL** to show the major axes and the triads;
- **NONE** to hide all axes.

## The **WIREEDGE** Command

---

### Summary

Creates one or more straight or arc wire-edges from picked vertices and the data supplied.

### Toolbutton



### Command line parameters

Command	<b>WIREEDGE</b>	
Parameter	Default	Function
<b>NAME</b>		Attaches this name to the body formed
<b>EDGETYPE</b>	<b>STRAIGHT</b>	Type of wire edge to be created
	<b>STRAIGHT</b>	A straight wire edge
	<b>ARC</b>	A curved arc passing through the position specified
	<b>CENTRE</b>	A curved edge centred on the position given, using an arc length subtending less than 180°
	<b>OBTUSECENTRE</b>	A curved edge centred on the position given, using an arc length subtending more than 180°
<b>X0</b>		Coordinates of start of wire edge
<b>Y0</b>		
<b>Z0</b>		
<b>X1</b>		Coordinates of end of wire edge
<b>Y1</b>		
<b>Z1</b>		
<b>X2</b>		Coordinates of centre point or mid-side point of an arc
<b>Y2</b>		
<b>Z2</b>		
<b>UNIQUENAME</b>		Force a specific unique name for the wire edge created.

Command	WIREEDGE	
Parameter	Default	Function
<b>CONTINUE</b>	<b>NO</b>	Continue definition of wire edges:
	<b>NO</b>	Do not continue.
	<b>YES</b>	Pick the final vertex so it can be used as the first point of another edge.
	<b>CLOSE</b>	Force closure of the polygon to the start of the wire body.
	<b>COVER</b>	As <b>CLOSE</b> , but covers the wire body.
	<b>AUTOCOVER</b>	As <b>YES</b> , but covers a closed wire polygon if one has been created.
	<b>TESTCLOSE</b>	As <b>NO</b> , but identifies closure of a closed wire polygon has been created.

## Notes

The **WIREEDGE** command creates one or more wire edges.

- A wire edge can be used as the sweep direction in [The SWEEP Command \[page 318\]](#).
- A set of wire edges can be used to create a sheet face (see [The COVER Command \[page 194\]](#)).

The data for a wire edge can be provided using the parameters of the command. Picked vertices can also be used to supply the coordinates.

When a wire edge is created, it is given a unique name used for unique identification within the model, e.g. when picking. By default, this name is created from the supplied **NAME**. If necessary, the name is adjusted by appending a number to ensure that it is unique. If a name is given in the **UNIQUENAME** parameter, this name is used instead.

## Using Parameters

The coordinates of the start (**X0,Y0,Z0**) and end (**X1,Y1,Z1**) points and optional third point (**X2,Y2,Z2**) defining the wire edge can be given using parameters. The coordinates are specified in the current Working Coordinate System.

## Curved Edges

The type of edge formed is set by the **EDGETYPE** parameter, and can be straight or form a circular arc.

- For **EDGETYPE=STRAIGHT**, no additional data is required.

- For **EDGETYPE=ARC**, the point ( $X2,Y2,Z2$ ) specifies a point on the arc between the end points. The 3 points specified must not be co-linear, and a circular arc is formed passing through the position ( $X2,Y2,Z2$ ).
- For **EDGETYPE=CENTRE**, the point ( $X2,Y2,Z2$ ) represents the centre of the circular arc. The arc created will be the shortest arc between the start and end point using this centre position. If the start and end point are not at a fixed radius, a circular arc will still be created. The arc will have one end at either the specified start or end position, but the other end will not be at the specified position.
- **EDGETYPE=OBTUSECENTRE** behaves as the centre option, but selects the longest arc.

### Single picked vertex

If a single vertex has been picked before the **WIREEDGE** command, this will be assumed to be the starting position of the wire edge. The wire edge will be united into the body from which the vertex is picked, so the values in parameters ( $X0,Y0,Z0$ ), **NAME** and **UNIQUENAME** are not used.

### More than one picked vertex

If more than one vertex has been picked before the **WIREEDGE** command, a straight-sided polygon will be created through the picked vertices. The wire edges will be united into a body with the given **NAME**. When used in this way, the **EDGETYPE** must be **STRAIGHT**.

### Continuing and Closing

The **CONTINUE** parameter allows continuation of wire edge definition to form a polygon.

- **CONTINUE=YES**  
The end point of the edge or polygon that has been created will be picked in preparation for the next wire edge to be defined and further wire edge input will be expected.
- **CONTINUE=CLOSE**  
This can be used when adding a wire edge to an existing picked vertex of a body that contains only wire edges. If a suitable starting vertex of the wire body can be identified, the values in parameters ( $X1,Y1,Z1$ ) will be ignored and an edge to this starting vertex created. When creating a polygon from 3 or more picked vertices, this option will close the polygon.
- **CONTINUE=COVER**  
This is the same as **CONTINUE=CLOSE**, but the resulting polygon will be covered to form a sheet face body. See "The **COVER** Command" on page 194.
- **CONTINUE=TESTCLOSE** and **CONTINUE=AUTOCOVER** are used in commands issued by the GUI to test if a closed polygon has been created, and to take appropriate actions accordingly. Identification of the closure of a polygon will mean that during **REPLAY**, repositioning the start point of the polygon will correctly adjust the end point of the polygon to match.

## System Variables in the Modeller

The following tables are lists of [System Variables \[page 355\]](#) and [String Variables \[page 360\]](#) which are defined and updated by the Modeller.

Where the description of a variable refers to a command, the documentation of that command should be read for more information about the variable.

<b>System Variables</b>	
<b>Name</b>	<b>Usage</b>
<code>ANALYSIS</code>	Id of the analysis type of the current simulation.
<code>ANALYSIS_CHARGEDPARTICLE</code> , <code>ANALYSIS_SCALA</code>	Unique id for Charged Particle analysis
<code>ANALYSIS_CURRENTFLOW</code> , <code>ANALYSIS_TOSCACURR</code>	Unique id for Current Flow analysis
<code>ANALYSIS_ELECTROSTATIC</code> , <code>ANALYSIS_TOSCAELEC</code>	Unique id for Electrostatic analysis
<code>ANALYSIS_HARMONICEM</code> , <code>ANALYSIS_ELEKTRASS</code>	Unique id for Harmonic Electromagnetic analysis
<code>ANALYSIS_HARMONICHF</code> , <code>ANALYSIS_SOPRANOSS</code>	Unique id for Harmonic High Frequency analysis
<code>ANALYSIS_MAGNETIZATION</code> , <code>ANALYSIS_DEMAG</code>	Unique id for Magnetization analysis
<code>ANALYSIS_MAGNETOSTATIC</code> , <code>ANALYSIS_TOSCAMAGN</code>	Unique id for Magnetostatic analysis
<code>ANALYSIS_MOTIONALEM</code> , <code>ANALYSIS_CARMEN</code>	Unique id for Motional Electromagnetic analysis
<code>ANALYSIS_MODALHF</code> , <code>ANALYSIS_SOPRANOEV</code>	Unique id for Modal High Frequency analysis
<code>ANALYSIS_MODALSTRESS</code> , <code>ANALYSIS_STRESSEV</code>	Unique id for Modal Stress analysis
<code>ANALYSIS_QUENCH</code>	Unique id for Quench analysis
<code>ANALYSIS_STATICSTRESS</code> , <code>ANALYSIS_STRESSST</code>	Unique id for Static Stress analysis
<code>ANALYSIS_STATICTHERMAL</code> , <code>ANALYSIS_TEMPOST</code>	Unique id for Static Thermal analysis

<b>System Variables</b>	
<b>Name</b>	<b>Usage</b>
<b>ANALYSIS_TRANSIENTEM, ANALYSIS_ELEKTRATR</b>	Unique id for Transient Electromagnetic analysis
<b>ANALYSIS_ TRANSIENTTHERMAL, ANALYSIS_TEMPOTR</b>	Unique id for Transient Thermal analysis
<b>ANALYSIS_VELOCITYEM, ANALYSIS_ELEKTRAVL</b>	Unique id for Fixed Velocity Electromagnetic analysis
<b>ANGLE</b>	Angle subtended by the circular edge just <b>LISTed</b> .
<b>AREA</b>	Cross sectional area of conductor during <b>CONDUCTOR MODIFY</b> .
<b>BODIES</b>	The number of bodies in the model.
<b>BODYCELLS</b>	The number of cells in the body just <b>LISTed</b> .
<b>BODYVOLUME</b>	The volume of cells in the body just <b>LISTed</b> .
<b>BOLTZMANN</b>	Boltzmann constant in J/K.
<b>C</b>	Speed of light in m/s.
<b>CASE</b>	The simulation number.
<b>CELLFACES</b>	The number of faces bounding the cell just <b>LISTed</b> .
<b>CELLS</b>	The number of cells in the model.
<b>CELLVOLUME</b>	The volume of the cell just <b>LISTed</b> .
<b>CONDU</b>	Unit factor for conductivity.
<b>CONDUCTORS</b>	Number of conductors.
<b>CPSECOND</b>	The number of processor seconds since the start of the job.
<b>CURDU</b>	Unit factor for current density.
<b>DAY</b>	The day of the month.
<b>DISPU</b>	Unit factor for displacement current.
<b>EDGELENGTH</b>	The length of the edge just <b>LISTed</b> .
<b>EDGES</b>	The number of edges in the model.
<b>ELECTRONCHARGE</b>	Charge of electron, elementary charge in C.
<b>ELECTRONENERGY</b>	Energy of electron in eV.
<b>ELECTRONMASS</b>	Mass of electron in kg.
<b>ELECU</b>	Unit factor for electric field strength.

<b>System Variables</b>	
<b>Name</b>	<b>Usage</b>
<b>ELEMENTS</b>	The number of elements in the finite element mesh. If the volume mesh exists, this is the number of volume elements. If only the surface mesh exists, this is the number of surface elements.
<b>ENERU</b>	Unit factor for energy.
<b>EPSILON0</b>	Permittivity of free space in F/m.
<b>FACEAREA</b>	The area of the face just LISTed.
<b>FACEEDGES</b>	The number of edges bounding the face just LISTed.
<b>FACES</b>	The number of faces in the model.
<b>FIELU</b>	Unit factor for magnetic field strength.
<b>FILEEXISTS</b>	Set by \$ EXIST command: 1 file exists 0 file does not exist
<b>FILETYPE</b>	Set by \$ EXIST command: 1 file 2 folder or directory
<b>FLUXU</b>	Unit factor for magnetic flux density
<b>FORCU</b>	Unit factor for force.
<b>FREQ or FREQUENCY</b>	The frequency of a harmonic simulation.
<b>GRAVITY</b>	The acceleration due to gravity in m/s <sup>2</sup> .
<b>HOUR</b>	The number of whole hours since midnight.
<b>JDAY</b>	The Julian day number (e.g. 1 for 1 <sup>st</sup> January and 366 for 31 <sup>st</sup> December in a leap year).
<b>LAYER</b>	The layer number when creating layered surfaces.
<b>LENGU</b>	Unit factor for length.
<b>MAJOR RADIUS</b>	The major radius of the elliptical edge just LISTed.
<b>MESH</b>	The state of the finite element mesh: 0 no mesh 1 surface mesh 2 volume mesh
<b>MINORRADIUS</b>	The minor radius of the elliptical edge just LISTed.
<b>MINUTE</b>	The number of whole minutes since the start of the hour.

System Variables	
Name	Usage
MODELBODY	The state of the model body:
	0 There is no model body.
	1 The model body has been created successfully.
MODELLER	2, see PROGRAM [page 358].
MONTH	The month number.
MU0	Permeability of free space in H/m.
NODES	The number of nodes in the mesh. If the volume mesh exists, this is the number of nodes in the volume mesh. If only the surface mesh exists, this is number of nodes in the surface mesh.
OPENSTREAM	The number of last stream opened by \$ OPEN.
OPERA2DPP	4, see PROGRAM [page 358].
PI	$\pi$
PICKEDBODIES	The number of picked bodies.
PICKEDCELLS	The number of picked cells.
PICKEDCONDUCTORS	The number of picked conductors.
PICKEDEDGES	The number of picked edges.
PICKEDFACES	The number of picked faces.
PICKEDVERTICES	The number of picked vertices.
PLANCK	Planck constant in J s.
POSTPROCESSOR	3, see PROGRAM [page 358].
POWEU	Unit factor for power.
PREPROCESSOR	1, see PROGRAM [page 358].
PROGRAM	2. In a command input file, PROGRAM can be compared against MODELLER, OPERA2DPP, PREPROCESSOR and POSTPROCESSOR to determine which program is being run.
PROTONMASS	Proton mass in kg.
PROTONMASSRATIO	Proton to electron mass ratio.
RADIUS	The radius of the circular edge just LISTed.
READVALUE00	The number of items on a line read from a file by \$ READ.

<b>System Variables</b>	
<b>Name</b>	<b>Usage</b>
READVALUE01, READVALUE02, ...	The values of items read from a file by \$ READ.
RCENTER, TCENTER, ZCENTER RCENTRE, TCENTRE, ZCENTRE	Polar coordinates at the centre of a circular or elliptical line just LISTed.
REND, TEND, ZEND	Polar coordinates at the end of the line just LISTed.
RSTART, TSTART, ZSTART	Polar coordinates at the start of the line just LISTed.
SCALU	Unit factor of magnetic scalar potential.
SECOND	The number of seconds since the start of the minute (to nearest millisecond).
SLICETHICK	The thickness of a 2d-slice model.
STEFANBOLTZMANN STEPHANBOLTZMANN	Stefan-Boltzmann constant in W/(m <sup>2</sup> K <sup>4</sup> ).
SURFACEELEMENTS	The number of surface elements in the finite element mesh.
THETA, PHI, PSI	Euler angles of the local coordinate system just LISTed.
U, V, W	Origin of the local coordinate system just LISTed.
VECTU	Unit factor for magnetic vector potential.
VERSION	Software version as a number.
VERTICES	The number of vertices in the model.
VOLUMEELEMENTS	The number of volume elements in the finite element mesh.
X, Y, Z	Point coordinates. Coordinates of the vertex just LISTed.
XCENTER, YCENTER, ZCENTER XCENTRE, YCENTRE, ZCENTRE	Cartesian coordinates at the centre of a circular or elliptical line just LISTed.
XEND, YEND, ZEND	Cartesian coordinates at the end of the line just LISTed.
XSTART, YSTART, ZSTART	Cartesian coordinates at the start of the line just LISTed.
YEAR	The year number.
Z0FREE	Impedance of free space, MU0*C.

<b>String Variables</b>									
<b>Name</b>	<b>Usage</b>								
ANALYSIS	Physics name of the current solver: <b>CHARGEDPARTICLE, CURRENTFLOW, ELECTROSTATIC, HARMONICEM, HARMONICHF, MAGNETIZATION, MAGNETOSTATIC, MODALHF, MODALSTRESS, MOTIONALEM, QUENCH, STATICSTRESS, STATICTHERMAL, TRANSIENTEM, TRANSIENTTHERMAL or VELOCITYEM</b>								
ANALYSIS_PROGRAM	Old name of the current solver (in the same order as above): <b>SCALA, TOSCACURR, TOSCAELEC, ELEKTRASS, SOPRANOSS, DEMAG, TOSCAMAGN, SOPRANOEV, STRESSEV, CARMEN, QUENCH, STRESSST, TEMPOST, ELEKTRATR, TEMPOTR or ELEKTRAVL</b>								
COMIBASENAME	The name of a command input file while it is open (without <b>.comi</b> ).								
COMIPATH	The name of the folder containing a command input file, while it is open.								
LOADFILENAME	The name of the last file <b>LOADED</b> .								
NOW	Time in <b>hh:mm:ss</b> format.								
PROJECTFOLDER	The project folder.								
READSTRING00	The complete line read from a file by <b>\$ READ</b> .								
READSTRING01, READSTRING02, ...	The items read from a file by <b>\$ READ</b> .								
TODAY	Date in <b>dd/mmm/yyyy</b> format.								
VERSION	Version number as a character string.								
VF_ANALYSISTYPE	The selected analysis program set by <b>The ANALYSISDATA Command [page 122]</b> .								
VF_LASTANLFILE	The name of the database from the last <b>Start analysis</b> .								
VF_MATERIALUNITS	The unit set used to define material properties: <b>CGS, INCH, METRE, MICRON, MM or TRUECGS</b> .								
VF_VOLORIENT	Current settings of volume orientation: <table border="1"> <tr> <td>XYZ</td> <td>0, 0, 0</td> </tr> <tr> <td>YZX</td> <td>90, 0, 90</td> </tr> <tr> <td>ZXY</td> <td>90, 90, 180</td> </tr> <tr> <td>OTHER</td> <td>any other angles</td> </tr> </table>	XYZ	0, 0, 0	YZX	90, 0, 90	ZXY	90, 90, 180	OTHER	any other angles
XYZ	0, 0, 0								
YZX	90, 0, 90								
ZXY	90, 90, 180								
OTHER	any other angles								

WORKINGFOLDER	The working folder.
YESORNO	Pre-answer "yes-no" questions with YES or NO.

# **Chapter 4**

# **Opera-3d Pre-Processor**

## **Introduction**

---

The Opera-3d Pre-Processor can prepare data for the Opera-3d analysis programs. The analysis programs use finite elements to model three dimensional electromagnetic devices, with the added facility for direct evaluation of fields from conductors carrying prescribed currents. The Pre-Processor supports the major features of the analysis programs.

The Opera-3d Pre-Processor is an interactive program that is used to create and edit three dimensional finite element models, define material characteristics for nonlinear magnetic or dielectric components, assign boundary conditions, specify complicated conductor geometries with prescribed excitation, show the models using wire frame and hidden surface displays and output data files in the formats accepted by the analysis programs.

When the Pre-Processor starts, on Linux operating systems, the first input the user must give is 'device nomination'. This tells the program what graphics options should be used. This is described with [The DEVICE Command \[page 434\]](#) which can be used at any time to reset the device, or change graphics options.

After device nomination, or when the Opera-3d Pre-Processor is restarted with the **CLEAR** command, the program looks for a file called *oppre.comi*, first in the current file directory and then in the user's home directory. If such a file exists it is read into the program as a **\$ COMINPUT** file (see [Command Input Files \[page 61\]](#)). This allows the user to reset the default values of certain commands, e.g. [The COLOUR Command \[page 381\]](#) or define frequently used **\$ CONSTANTS** and **\$ PARAMETERS** (see [User Variable Commands \[page 55\]](#)) each time the program is started.

Two methods of command and data entry are available:

1. Menu system or GUI (Graphical User Interface) - command selection and data specification are carried out under mouse control.
2. Command line input - command selection and data specification are carried out from the keyboard.

Under normal operation, the Pre-Processor starts in the GUI mode.

## The Pre-Processor Quick Reference Guide

The following is a complete list of the ‘top-level’ keyboard commands (which can be entered in response to the prompt ‘Opera >’) and menu items. Following sections contain complete descriptions of all the keyboard commands, sub-commands and modes of the Opera-3d Pre-Processor. The commands are described in alphabetical order. The modes of the **DEFINE** command are described in the order they occur when using the program.

### Commands for Keyboard Entry – Command Line

Command line entry is carried out in the text window.

- Help Command:

<b>HELP</b>	Obtain System overview, help on command interpreter, Euler angles, units and new features in this version.
-------------	--

- Mesh definition and editing commands:

<b>DEFINE</b>	Enter the define sequence to input points, facets, subdivisions, extrusions, materials and boundary conditions of the finite element mesh. <b>DEFINE</b> can also be used to define geometry of 20 node ‘brick’ conductor elements.
<b>REDEFINE</b>	Remove all extrusions and enter <b>DEFINE</b> sequence to modify base-plane and redefine extrusions.
<b>EXTEND</b>	Add more extrusion layers to a mesh.
<b>MODIFY</b>	Modify mesh data: points, subdivisions, materials and boundary conditions.
<b>LABEL</b>	Add extra labels to points, lines, facets and volumes.
<b>MATERIALS</b>	Define material properties.
<b>TRANSFORM</b>	Transform a labelled set of points
<b>SLIP</b>	Add or remove a slip surface for CARMEN models.
<b>CHECK</b>	Examine the shape of the finite element volumes, adding label <b>DEBUG</b> . Find the exterior facets of the finite element volumes, adding label <b>EXTERNAL</b> . Report whether hexahedral meshing is possible.

- Conductor definition and editing command:

<b>CONDUCTOR</b>	Enter conductor sub-command mode to define, modify and list conductor data.
------------------	---

- Finite element mesh commands:

<b>MESH</b>	Generate a mesh on the surfaces of the volumes.
<b>FILL</b>	Generate a mesh inside each volume.

- Display command:

<b>DISPLAY</b>	Display some or all of the mesh and conductors in 3D with optional hidden surface removal.
<b>THREED</b>	Start or update the 3D Viewer

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

<b>BHDATA</b>	Enter the BH data definition and editing mode.
---------------	--

- Commands to read and write data files:

<b>READ</b>	Read a file of Opera-3d Pre-Processor data.
<b>EDIT</b>	Read a file of Opera-3d Pre-Processor data allowing data to be edited at break points.
<b>WRITE</b>	Write a file of Opera-3d Pre-Processor data for use with <b>READ</b> and <b>EDIT</b> commands.
<b>TABLE</b>	Write a table file containing all the nodes of the model.
<b>SOLVERS</b>	Prepare an analysis database.
<b>IDEAS</b>	Read a finite element mesh from an I-DEAS universal file.

- Program management commands:

<b>DEVICE</b>	Reset or change graphics device.
<b>COLOUR</b>	Enquire and set colours for the display.
<b>TITLE</b>	Control screen titles.
<b>CLEAR</b>	Clear program data and re-initialize all commands.
<b>DUMP</b>	Write a picture file containing the current display.
<b>END</b>	End Opera-3d Pre-Processor.

## Menu System – GUI

The following is an overview of the first level menu structure. A brief description of the functionality of each item is also given and the page number of the section of this chapter which describes it in detail.

FILE	OPTIONS	DISPLAY	HELP	MESH	MODIFY	DEFINE	MENU_OFF
------	---------	---------	------	------	--------	--------	----------

### FILE

File Options		
Read Pre-Processor file	read a Pre-Processor command file	<a href="#">page 481</a>
Write Pre-Processor file	write a Pre-Processor command file	<a href="#">page 510</a>
I-deas universal file	read a finite element mesh from an I-DEAS universal file	<a href="#">page 452</a>
Analysis	Create/edit an analysis database	<a href="#">page 485</a>
... create new database	Create an analysis database	
... use existing database	Edit an analysis database	
Start analysis now	Start an analysis program using <b>OPERAANL</b>	<a href="#">page 79</a>
Write node table file	Write a file in table format containing all the node coordinates	<a href="#">page 502</a>
Commands in	read a command input file	<a href="#">page 61</a>
Dump picture	send current graphics display to a file	<a href="#">page 442</a>
Change directory	change the current directory	<a href="#">page 77</a>
System command	run an operating system command	<a href="#">page 77</a>
Return	close menu	
End Opera-3d/Pre	end Pre-processing session	<a href="#">page 447</a>

## OPTIONS

Options		
Calculator...		
Parameters	set a user defined parameter	<a href="#">page 55</a>
Constants	set a user defined constant	<a href="#">page 55</a>
List variables	list user defined variables	<a href="#">page 55</a>
Colour settings	modify colours in graphics palette	<a href="#">page 381</a>
Clear and Reset	clear all mesh data and restart	<a href="#">page 380</a>
Dump picture	send current graphics display to a file	<a href="#">page 442</a>
Graphics output	select type of graphics output	<a href="#">page 434</a>
Title options	specify title options	<a href="#">page 507</a>
Title position	specify title position in window	<a href="#">page 507</a>
Display title	enter text of title	<a href="#">page 507</a>
Return	close menu	

## DISPLAY

3D Viewer...		
... style	specify wire frame, solid etc. display of model	<a href="#">page 503</a>
... select parts	specify components to be displayed	<a href="#">page 503</a>
... refresh display	start or update 3D Viewer	<a href="#">page 503</a>
Display Command ...		
... view	specify display size and orientation of model	<a href="#">page 436</a>
... style	specify wire frame, solid etc. display of model	<a href="#">page 436</a>
... select parts	specify components to be displayed	<a href="#">page 436</a>
... refresh display	clear and re-draw graphics window	<a href="#">page 436</a>
... copy 3D view	copy view parameters from 3D Viewer	<a href="#">page 436</a>
Return	close menu	

**HELP**

Help		
System Overview	general description of Pre-Processor	<a href="#">page 451</a>
The GUI	assistance with menu system	
Command line	assistance with command line entry	
Return	close menu	

**MESH**

Mesh Generator		
Surface mesh ...		
... options	set options for surface mesh generator	<a href="#">page 467</a>
... triangles	generate a surface mesh of triangles	<a href="#">page 467</a>
... quadrilaterals	generate a surface mesh of quadrilaterals	<a href="#">page 467</a>
Volume mesh ...		
... options	set volume mesh generator options	<a href="#">page 450</a>
Mesh	generate a volume mesh	<a href="#">page 450</a>
Return	close menu	

**MODIFY**

Data Modification		
Mesh number	specify mesh to be modified	<a href="#">page 469</a>

No 3D Viewer	choose whether to use 3D Viewer	<a href="#">page 469</a>
Point coordinates	modify point coordinates	<a href="#">page 470</a>
Subdivisions	modify subdivisions	<a href="#">page 474</a>
Material properties	modify element/potential/material types	<a href="#">page 479</a>
Boundary conditions	modify boundary conditions	<a href="#">page 480</a>
Add slip surface	add slip surface to CARMEN models	<a href="#">page 484</a>
Remove slip surface	remove slip surface	<a href="#">page 484</a>
Labels	modify entity labels	<a href="#">page 458</a>
Transform labelled points	apply transformation to a group of points	<a href="#">page 509</a>
Conductors	modify conductor definitions	<a href="#">page 387</a>
Return	close menu	

## DEFINE

Data Definition		
Define new mesh	start creating a new mesh	<a href="#">page 393</a>
Extend existing mesh	continue extending an existing mesh	<a href="#">page 448</a>
Redefine a mesh	remove extrusions, modify baseplane and re-extrude	<a href="#">page 483</a>
Check a mesh	check existing mesh geometry	<a href="#">page 378</a>
Conductors	create a conductor	<a href="#">page 383</a>
Read conductor data	read a file of conductor definition commands	<a href="#">page 481</a>
Material properties	define material properties	<a href="#">page 462</a>
BH tables	define/edit a B-H curve	<a href="#">page 375</a>
Return	close menu	

## The Graphical User Interface

---

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

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

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

### Menus

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

Alternatively, menu items can be chosen using the keyboard arrow keys. When the required item is high-lighted, it can be selected using the <Enter> or <Return> keys.

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

Symbol	Action
down arrow	Drop Down: this activates a sub-menu. It only exists in the top-level horizontal menu.
	Pull Right: activates a sub-menu.
	Pick and Pull: activates a sub-menu after a selection from the displayed model (see Pick below).
	Return: returns to higher-level menu.
	Toggle: swaps between 2 options and the symbol changes between  and . The current state of the program is displayed.

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

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

Not all menu items can be used at all times. For example, it is not possible to create a mesh before any volumes have been defined. Unavailable menu items are displayed in pale-blue rather than white until they become available as the result of other commands.

## Parameter Boxes

ParameterBoxes (Figure 4.1)



Figure 4.1 A Typical ParameterBox

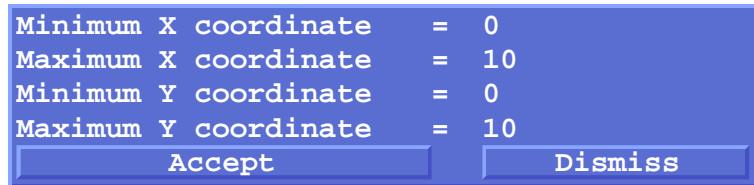


Figure 4.2 A Typical ParameterBox

are used to input information from the keyboard. Default values (if they exist) are displayed and are initially high-lighted. When a value is high-lighted it can be replaced by the first characters typed. A value can be edited by moving the text cursor before typing any characters keys.

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

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

- Up and down arrows and <Tab> can be used to move between the parameters.
- Left and right arrows move the text cursor within the value being edited.
- <Enter> or <Return> move to next parameter. If **Accept** is high-highlighted or there is only one parameter, the command is executed.
- <Esc> escapes from the command.
- <Back-space> or <Delete> delete characters.

To toggle insert mode: <Insert> (Windows) or function keys <F2> or <PF2> (X-windows)

- To move to start of the value <Home> (Windows) or function keys <F3> or <PF3> (X-windows)
- To move to end of the value: <End> (Windows) or function keys <F4> or <PF4> (X-windows)

## FileBoxes and CDBoxes

FileBoxes (Figure 4.3)

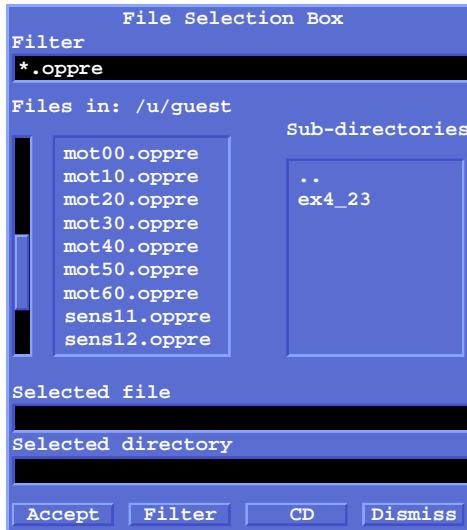


Figure 4.3 A File Selection Box

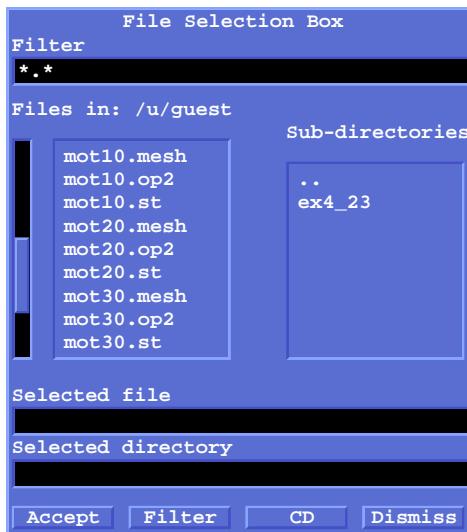


Figure 4.4 A File Selection Box

are used for selecting a file name for reading or writing. The FileBox contains a filter string which specifies a subset of all files. If the filter is edited, it can be acted on by typing <Enter> or <Return> or by selecting the **Filter** button.

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

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

One file should be selected from the list of files. Double-clicking (selecting the file twice in quick succession) confirms the selection. Alternatively, the selection can be confirmed by selecting the

**Accept** button. The required file name can also be typed into the selection box and accepted by typing <Enter> or <Return>.

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

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

<Esc> or by selecting the **Quit** button.

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

## Dialog Boxes

Dialog Boxes are used to input information using a combination of keyboard and mouse operations. Within a Dialog Box there can be:

- Text-inputs: black rectangles. Initially the first text-input is selected and any characters typed will appear there. Any text-input can be selected with the mouse before typing. The <Enter>, <Return>, down arrow or <Tab> keys can be used to move to the next text-input. The up arrow key can be used to move to the previous text input. Within a text-input the value can be edited using the editing and function keys defined for [Parameter Boxes \[page 369\]](#). Selecting a down arrow button  to the right of a text input activates a FileBox to supply a file name for the text-input (see [FileBoxes and CDBoxes \[page 370\]](#)).
- Switches: small squares or 'radio-buttons'. The switches can be on  (red) or off  (blue). The state of a switch can be changed by selecting with the mouse pointer. Turning on one switch might turn others off if the options are mutually exclusive.

- Buttons: labelled rectangles. These are used to action the selected options, or exit without issuing a command. A button can be selected using the mouse or, if it is high-lighted, with the <Enter> or <Return> key. The <Esc> can also be used to exit without issuing a command.
- Scrolling lists: a list of items from which one or many can be selected using the mouse. If the list is long or wide, scroll-bars can be used to view other parts of the list in the same was as for FileBoxes (see [FileBoxes and CDBoxes \[page 370\]](#)).

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

## MessageBoxes

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

- Information: the results of commands, warnings etc. These boxes are labelled with a large i. Information MessageBoxes can be dismissed by typing any key on the keyboard (except <F1>) or with the left mouse button.
- If the quantity of information exceeds the size of the window, a scroll-bar is displayed to enable the whole message to be viewed. The window will show the top of the message as the scroll is generated, but other parts of the message can be viewed by dragging the scroll bar up or down.
- Errors: these include Pick operations outside the model space and bad values in ParameterBoxes. The program gives the user another chance to perform the input if an error occurs. Error MessageBoxes are labelled with a large !. They can be dismissed by typing any key on the keyboard (except <F1>) or with the left mouse button.
- Questions: these always require a choice between **YES** and **NO**. QuestionBoxes are labelled with a large ?. They can be dismissed by selecting either the **YES** box or the **NO** box.
- Input: these always require additional information to be given by the user via a ParameterBox, a DialogBox or a FileBox which appears below the MessageBox. The boxes disappear when the information has been supplied.
- Timer: these indicate how much of an operation has been completed. Timers are only displayed for operations for which the estimated elapsed time is greater than 5 seconds. Timer boxes cannot be dismissed, but will automatically disappear when the operation is complete.

## ColourBoxes

The ColourBox ([Figure 4.5](#)) is used to redefine colours used on the display. It

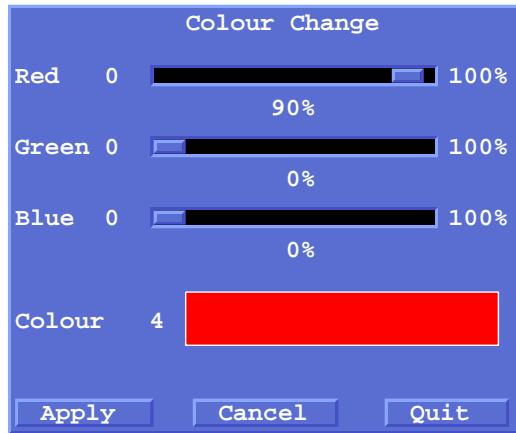


Figure 4.5 A ColourBox

consist of 3 horizontal slider bars, one each for red, green and blue, a square showing the colour as it is changed and three buttons, **Accept**, **Cancel** and **Quit**.

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

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

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

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

## Hiding and Leaving the Menus

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

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

## The BHDATA Command

---

### Menu route

**DEFINE-> BH tables**

### Command line parameters

Command	BHDATA	
Parameter	Default	Function
<b>UNITS</b>	<b>CGS</b>	The units of B and H
		<b>CGS</b> gauss and oersted
		<b>SI</b> tesla and ampere/metre

The TOSCA, ELEKTRA and CARMEN programs use material characteristics to relate flux density and field intensity of all materials (except for **AIR** and **NULL**). 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 magnetization; the program extrapolates correctly.

Hysteresis can be modelled in CARMEN, ELEKTRA/TR and DEMAG. To enable hysteretic material properties for a material a user variable must be defined for the material before the analysis database is created. The variable name must have the form **#material\_sethysteretic** where **material** is the name of the material. The value of the variable must be 1.

Hysteresis can also be modelled in a limited form, particularly applicable to ferrites, in ELEKTRA/SS using complex permeability in [The MATERIALS Command \[page 462\]](#) or **SOLVERS** command, **MATERIAL** sub-command [\[page 495\]](#).

Anisotropic materials can be treated as a stack of laminations or by use of multiple BH curves. For hard magnetic materials the demagnetization curve in the second and third quadrant is used. The orientation of laminations, anisotropic materials and permanent magnets is set by the **VECTOR** property, set in the **DEFINE** command, [Material Definition Mode \[page 420\]](#). The **PACKING** factor of laminations is set there also.

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

More information on BH curves for different types of material can be found in [Material Types \[page 146\]](#).)

The **BHDATA** command is used to create, edit or check tables of pairs of values that define the non-linear BH, DE or JE characteristics of magnetic, dielectric or conducting materials. There must be at least 5 and not more than 50 entries in each table. Material characteristics are not stored with the

models in Pre-Processor data files. After a table has been defined or edited the new data should be stored in a materials library file. A separate system directory could be created to contain all such files. Such a directory is supplied with the software and can be accessed using the directory name **\$VFDIR/bh** (Linux) or **%VFDIR%|bh** (Windows) in the FileBox Selected directory input box.

Names of BH files are requested by the **SOLVERS** command, **MATERIAL** sub-command [page 495] for all material names used.

The **BHDATA** command has one parameter, **UNITS**, which allows the user to define the data in either **CGS** units (gauss and oersted) or **SI** units (tesla and ampere/metre). Files of BH data stored in one set of units can be loaded and edited in either set of units.

On entering the **BHDATA** command, the user is presented with a default BH curve. In general it will be necessary to delete this curve and replace it with the desired characteristic. In order to use the default curve it must be also stored in a file.

The **BHDATA** command has 8 sub-commands allowing addition of points on the current curve, modification of existing points, access to files, and data checking. The commands do not have named parameters but rely on keywords and values being supplied in the correct order. In the description below the keywords are given in upper case and the variable values in lower case. Keywords can be abbreviated to single letters. In any case only the first 4 characters are decoded.

In the **REPLACE** sub-command, the system variables **B** and **H**, the values of the current point, can be used in expressions. For example, an existing curve can be degraded in a command loop such as:

```
OP-BH > $ do #i 1 10
OP-BH > repl #i h+(b-h)*0.9 h -redr
OP-BH > $ end do
```

The boolean **-REDRAW** prevents the graph of **B** against **H** being re-drawn each time round the loop.

## BHDATA Sub-commands

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

Sub-Commands	Function
<b>ADD b h</b>	Add a new point to the end of the table. <b>b</b> and <b>h</b> are numeric values of <b>B</b> and <b>H</b> .
<b>CHECK m n</b>	Check the data and display the interpolations of the data used in analysis. <b>m</b> and <b>n</b> specify the first and last point displayed.
<b>DELETE m n</b>	Deletes the points <b>m</b> to <b>n</b> of the curve. <b>n</b> can have the value * to indicate the last point.
<b>INSERT number b h</b>	Inserts a new point after the point <b>number</b> of the curve. <b>b</b> and <b>h</b> are numeric values of <b>B</b> and <b>H</b> .

Sub-commands	Function
<b>LOAD</b> <i>file</i>	Loads a curve from a file. The file name extension <b>bh</b> is added to the name if no extension is given. Any points already defined are deleted.
<b>QUIT</b>	Leave the <b>BHDATA</b> command.
<b>REPLACE</b> <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>B</b> and <b>H</b> . <b>± REDRAW</b> can be used to control whether the curve is re-displayed.
<b>STORE</b> <i>file</i>	Stores the curve in a file. The file name extension <b>bh</b> is added to the name if no extension is given.

## The **CHECK** Command

---

### Menu route

**DEFINE -> Check a mesh**

### Command line parameters

Command	<b>CHECK</b>	
Parameter	Default	Function
<b>MESH</b>	1	Number of the finite element mesh to be checked.
<b>VERBOSE</b>	NO	Verbose reporting of errors:
		NO      Summary of errors.
		YES     Detail of error given for each volume.

The **CHECK** command examines the shape of the volumes of the **MESH** number given, adding the label **DEBUG** to any which may cause errors in the mesh generator or analysis programs. It also gives the label **EXTERNAL** to all facets which are not shared by two volumes. There is no checking between facets of different meshes.

The type of checking depends on whether the model is capable of being meshed with hexahedra or can only be meshed with tetrahedra. Hexahedral meshes can be generated in models which obey the following rules:

- All volume facets must be triangles or quadrilaterals.
- The subdivisions on opposite lines of quadrilateral facets must be the same.
- The subdivisions of two lines of triangular facets must be the same.

The **CHECK** command first examines the model to see whether it can be meshed with hexahedra. It then checks each volume as described below.

In verbose mode, the unique labels assigned to volumes have the form **Vnnn**, where **nnn** is the internal volume number. This allows individual volumes to be displayed.

If the **CHECK** command is used again, after correcting the volumes in error, the label **DEBUG** is removed from any volumes previously in error and the unique labels, **Vnnn** are also removed.

- Example: to display volumes with label **DEBUG** in layer 3:

```
Opera > check
Opera > disp type=volume label=debug 11=3 3
```

## Checks on Hexahedral Models

If the model can be meshed with hexahedra, the following checks are performed.

The program calculates the determinant of the Jacobian which defines the isoparametric mapping from a hexahedron to a unit cube. For the mapping to be unique this must be greater than zero everywhere within the hexahedron. The Jacobian determinant is evaluated at every point defining the volume. In some circumstances the checks may be more strict than necessary. For example, a quadratic volume is checked using quadratic shape functions, but in analysis, the volume may be discretized into linear elements in which the errors do not show up.

If the Jacobian determinant changes sign within a volume, this may be due to any of the errors listed below:

- faces which cut through each other.
- mid-side points which are too close to the corners (closer than 0.25 times length of side).
- excessive curvature or distortion.

The Jacobian can also be used to evaluate the size of each volume. If the Jacobian goes to zero or is negative throughout a volume, this is also reported.

The number of volumes in error is reported for each layer and since they have the label **DEBUG** they can be displayed separately using the **DISPLAY** command.

If **+VERBOSE** is selected, the **CHECK** command reports an error message and assigns a unique label to each **DEBUG** volume. The errors reported are:

- Jacobian changes sign.
- Very small volume.
- Negative volume.
- Mid-side node too close to corner.

The **CHECK** command also counts the total numbers of nodes and elements in the mesh. These numbers are approximate if degenerate volumes (with facets with less than 4 sides) or **NULL** materials are used.

## Checks on other Models

For models with facets with more than 4 sides or irregular subdivision, the following checks are performed.

The equation of the plane of each facet with more than 4 sides is calculated. An error is reported if any points defining the facet do not lie in the plane.

## The **CLEAR** Command

---

### Menu route

**OPTIONS -> Clear and Reset**

### Command line parameters

Command	<b>CLEAR</b>
No Parameters	

The **CLEAR** command puts Opera-3d back to the state it was in when it first started. It deletes all the data including conductors and construction lines, re-initializes all variables and sets all parameters back to their default values.

Note that the **CLEAR** command should not be used in a command input (**comi**) file because it will stop the execution of the file.

The program looks for a file called **oppre.comi**, first in the current project folder and then in the user's home folder, and if it exists, it is opened and read as a **\$ COMINPUT** file (see [Command Input Files \[page 61\]](#)) before control is passed back to the user. This allows the user to supply an individual choice of default values for commands or define frequently used **\$ CONSTANTs** and **\$ PARAMETERS** (see [User Variable Commands \[page 55\]](#)).

## The COLOUR Command

---

### Menu route

**OPTIONS -> Colour settings**

### Command line parameters

Command	COLOUR	
Parameter	Default	Function
<b>OPTION</b>	<b>ENQUIRE</b>	Option: <b>CONFIGURE</b> , <b>ENQUIRE</b> or <b>SET</b> .
		<b>CONF</b> Re-configure the colour map for a different number of distinct <b>MATERIALS</b> .
		<b>ENQU</b> Enquire which colour numbers are used for each material and part of the display.
		<b>SET</b> Reset colour number <b>CODE</b> to new values of <b>RED</b> , <b>GREEN</b> and <b>BLUE</b> .
<b>CODE</b>	<i>none</i>	Colour map number to be redefined.
<b>RED</b>	<i>none</i>	Amount of red for colour <b>CODE</b> .
<b>GREEN</b>	<i>none</i>	Amount of green for colour <b>CODE</b> .
<b>BLUE</b>	<i>none</i>	Amount of blue for colour <b>CODE</b> .
<b>MATERIALS</b>	5	Number of distinct material colours.

The **COLOUR** command informs the user which colours are used for each part of the display and enables the definitions of the colours to be changed.

There are three **OPTIONS**: **CONFIGURE**, **ENQUIRE** and **SET**.

- **CONFIGURE** sets the number of distinct material colours to be made available. The operation of this option depends on the total number of colours available on the display being used. A minimum of 5 and a maximum of 100 distinct material colours are allowed. If there are more materials than colours, the colours are re-used cyclically for higher material numbers. Increasing the number of material colours reduces the number of light-source shading levels.
- **ENQUIRE** lists the colour numbers used for each part of the display.
- **SET** displays the definition of the colour number given by **CODE** in terms of its red, green and blue components. The colour can be changed by giving values to **RED**, **GREEN** and **BLUE** which should each be in the range 0 to 1. The current values of **RED**, **GREEN** and **BLUE** can be used in expressions to define the new values.

If the colour **CODE** to be reset is the main colour for air, conductors or a material, etc., the corresponding light-source shading colours are reset to different shades of the new colour.

On some types of display, the new colours will not be shown until the picture is refreshed.

## The CONDUCTOR Command

---

### Menu route

**DEFINE -> Conductors**

### Command line parameters

Command	<b>CONDUCTOR</b>
No Parameters	

This command enters the conductor sub-command mode of the Pre-Processor. In this mode new conductors may be defined, existing conductors edited or erased, the parameters of the current set of conductors may be listed or they may be stored in a **cond** data file. The sub-commands are:

<b>CONDUCTOR Sub-commands</b>	
Sub-command	Function
<b>DEFINE</b>	Define a new conductor
<b>ERASE</b>	Erase one or several conductors
<b>EXTERNAL</b>	Define uniform external field
<b>MODIFY</b>	Modify one or several conductors
<b>PRINT</b>	Print details of one or several conductors to the text window and the log file.
<b>QUIT</b>	Leave the CONDUCTOR command.
<b>WRITE</b>	Write a data file with commands to define all the conductors.

The conductor geometries and all the parameters are described in detail in [Conductors \[page 512\]](#).

## The CONDUCTOR Sub-command **DEFINE**

### Menu route

**DEFINE -> Conductors -> Define a conductor**

## Command line parameters

Sub-command	DEFINE		
Parameter	Default	Function	
<b>END</b>	<i>none</i>	Conductor type	
		<b>SOLENOID</b>	Simple solenoid
		<b>GSOLENOID</b>	Generally orientated solenoid
		<b>RACETRACK</b>	Simple racetrack
		<b>GRACETRACK</b>	Generally orientated racetrack
		<b>BEDSTEAD</b>	Simple bedstead
		<b>GBEDSTEAD</b>	Generally orientated bedstead
		<b>HELIX</b>	Helical end racetrack
		<b>CPEND</b>	Constant perimeter end racetrack
		<b>STRAIGHT or GSTRAIGHT</b>	Straight bar
		<b>ARC or GARC</b>	Circular arc
		<b>BR8</b>	8-noded brick
		<b>BR20</b>	20-noded brick
<b>DEFAULT</b>	<i>max</i>	Number of an existing conductor used to reference default values. If no number is given, then the highest numbered conductor of the same type is used.	

The program prompts the user to provide values for all the parameters appropriate for the conductor type. The input lines are decoded in the same way as top-level commands without a command name. This means that the help character, !, can be used to get more information about the parameters and can be used twice, !!, to get individual parameter prompting.

The default value is given in parentheses following each parameter name. If no value is given, there is no default and a value must be supplied.

The full parameter lists were given above with the descriptions of the conductors, in the order in which they are requested from the user.

## Examples

In the example which follows various forms of input are used during the definition of a solenoid.

Opera > **cond**

Define and list conductor data

Conductor definition and editing.

```

Sub-commands: Define, Modify, Erase, Print, Write and Quit; ! for help.
COND > define solenoid
Define conductors.
Specify new values by:
giving values positionally
OR assigning PARAMETER=value
OR hitting <return> to accept all defaults.

Conductor 1: Solenoid around Y' axis.
Local coordinate system 1
XCENTRE(0), YCENTRE(0), ZCENTRE(0)
COND > xcen=5 ycen=3 zcen=2
Solenoid cross section in X'Y' plane.
X1(), Y1(), X2(), Y2()
COND > !
Parameter Value Meaning
X1           Local X coordinate of bottom inside
              corner
Y1           Local Y coordinate of bottom inside
              corner
X2           Local X coordinate of top inside
              corner
Y2           Local Y coordinate of top inside
              corner
COND > 3 3 3 5
Solenoid cross section in X'Y' plane.
X3(), Y3(), X4(), Y4()
COND > 5 5 5 3
Solenoid cross section curvatures.
CU1(0), CU2(0), CU3(0), CU4(0)
COND > cu3=0.05
Current density, symmetry and drive label
CURD(), SYMMETRY(1), PHASE(ONE)
COND > !!
There are 3 Parameters
For each parameter:
hit return to accept default
OR enter new value
OR type $H for help
OR type $S to skip remaining parameters and execute command
OR type $A to skip remaining parameters and abort command
NO.   Name      Value Meaning
1     CURD      Current density
!! > 10

```

```

2      SYMMETRY 1      Symmetry code (1 or 2)
!! > 2
3      PHASE    ONE     Drive label
!! >
Type return to obey command or $A to abort
!! >

Sub-commands: Define, Modify, Erase, Print, Write and Quit; ! for help.
COND >

```

## The CONDUCTOR Sub-command ERASE

### Menu route

**DEFINE -> Conductors -> Erase conductors**

### Command line parameters

Sub-command	<b>ERASE</b>	
Parameter	Default	Function
<b>C1</b>	<i>none</i>	Number of first conductor to be erased.
<b>C2</b>	<i>none</i>	Number of last conductor to be erased. <b>C2</b> can be set to * to indicate the highest numbered conductor.

The **ERASE** sub-command removes a range of conductors from the conductor database. For example, if **C1** is set to 3 and **C2** to 6 then conductors with numbers 3, 4, 5 and 6 will be erased. **C2** can be set to \* to indicate the highest numbered conductor.

The remaining conductors with numbers greater than **C2** are renumbered to give a contiguous set.

## The CONDUCTOR Sub-command EXTERNAL

### Menu route

**DEFINE -> Conductors -> Define external field**

### Command line parameters

Sub-command	<b>EXTERNAL</b>	
Parameter	Default	Function
<b>HXEXT</b>	0	X-component of external field.
<b>HYEXT</b>	0	Y-component of external field.
<b>HZEXT</b>	0	Z-component of external field.
<b>DRIVELABEL</b>		Drive label.

The **EXTERNAL** sub-command defines a uniform external field which can be used in addition to field from conductors in TOSCA (magnetostatics), SCALA, CARMEN and ELEKTRA analyses. A drive label must also be given so that multiple case scaling factors or a time variation of the external field can be defined (see [SOLVERS Command Prompts \[page 491\]](#)).

## The CONDUCTOR Sub-command MODIFY

### Menu routes

**DEFINE -> Conductors -> Modify conductors**  
**MODIFY -> Conductors**

### Command line parameters

Sub-command	<b>MODIFY</b>	
Parameter	Default	Function
<b>C1</b>	<i>none</i>	Number of first conductor to be modified.
<b>C2</b>	<i>none</i>	Number of last conductor to be modified. <b>C2</b> can be set to * to indicate the highest numbered conductor.
<b>END</b>	<i>none</i>	Conductor type: <b>SOLENOID</b> , <b>GSOLENOID</b> , <b>RACETRACK</b> , <b>GRACETRACK</b> , <b>BEDSTEAD</b> , <b>GBEDSTEAD</b> , <b>HELIX</b> , <b>CPEND</b> , <b>STRAIGHT</b> , <b>ARC</b> , <b>BR8</b> or <b>BR20</b> .
<b>SYMMETRY</b>	<i>none</i>	Symmetry code.
<b>XCENTRE</b>	<i>none</i>	X coordinate of origin of local system 1.
<b>YCENTRE</b>	<i>none</i>	Y coordinate of origin of local system 1.
<b>ZCENTRE</b>	<i>none</i>	Z coordinate of origin of local system 1.

Sub-command	<b>MODIFY</b>	
Parameter	Default	Function
<b>CURD</b>	<i>none</i>	Current density.
<b>PHI1</b>	<i>none</i>	Euler angle phi (local system 1).
<b>TOLERANCE</b>	<i>none</i>	Tolerance on flux density.
<b>X0</b>	<i>none</i>	X coordinate of origin of local system 2.
<b>Y0</b>	<i>none</i>	Y coordinate of origin of local system 2.
<b>Z0</b>	<i>none</i>	Z coordinate of origin of local system 2.
<b>T</b>	<i>none</i>	Euler angle theta (local system 2).
<b>P</b>	<i>none</i>	Euler angle phi (local system 2).
<b>S</b>	<i>none</i>	Euler angle psi (local system 2).
<b>IRXY</b>	<i>none</i>	Reflection code in xy plane of local system 1.
<b>IRYZ</b>	<i>none</i>	Reflection code in yz plane of local system 1.
<b>IRZX</b>	<i>none</i>	Reflection code in zx plane of local system 1.
<b>X1</b>	<i>none</i>	X coordinate of corner of conductor cross section.
<b>Y1</b>	<i>none</i>	Y coordinate of corner of conductor cross section.
<b>X2</b>	<i>none</i>	X coordinate of corner of solenoid cross section.
<b>Y2</b>	<i>none</i>	Y coordinate of corner of solenoid cross section.
<b>X3</b>	<i>none</i>	X coordinate of corner of solenoid cross section.
<b>Y3</b>	<i>none</i>	Y coordinate of corner of solenoid cross section.
<b>X4</b>	<i>none</i>	X coordinate of corner of solenoid cross section.
<b>Y4</b>	<i>none</i>	Y coordinate of corner of solenoid cross section.
<b>WIDTH</b>	<i>none</i>	Width of conductor in x or radial direction.
<b>THICKNESS</b>	<i>none</i>	Thickness of conductor in y or azimuthal direction.
<b>H1</b>	<i>none</i>	Length of straight section
<b>H2</b>	<i>none</i>	Length of upright ( <b>BEDSTEAD</b> ) Local Z coordinate of midpoint of cross-over ( <b>HELIX</b> ).
<b>R1</b>	<i>none</i>	Radius: inner radius of arc ( <b>RACETRACK</b> , <b>BEDSTEAD</b> , <b>ARC</b> ). Radius of cylinder ( <b>HELIX</b> , <b>CPEND</b> ).

Sub-command	MODIFY	
Parameter	Default	Function
R2	<i>none</i>	Radius: inner radius of arc ( <b>BEDSTEAD</b> ).
		Width of cross-over ( <b>HELIX</b> ).
		Radius of generating cylinder ( <b>CPEND</b> ).
PHI	<i>none</i>	Angle of <b>ARC</b> .
ALPHA	<i>none</i>	Angle of straight from mid plane of cylinder ( <b>HELIX</b> and <b>CPEND</b> ).
BETA	<i>none</i>	Angle of end of helix ( <b>HELIX</b> ), or cutter ( <b>CPEND</b> ).
CU1	<i>none</i>	Curvature of cross section of solenoid (points 1 to 2).
CU2	<i>none</i>	Curvature of cross section of solenoid (points 2 to 3).
CU3	<i>none</i>	Curvature of cross section of solenoid (points 3 to 4).
CU4	<i>none</i>	Curvature of cross section of solenoid (points 4 to 1).
FIT	<i>none</i>	Fit of straight section to cylinder: <b>TANGENTIAL</b> or <b>FITTING (CPEND)</b> .
PHASE	<i>none</i>	Drive label.
THETA1	<i>none</i>	Euler angle theta (local system 1).
PSI1	<i>none</i>	Euler angle psi (local system 1).

The **MODIFY** command can be used in two ways. If only the range of conductors is specified on the **MODIFY** sub-command then the program issues the same prompts as are used by **The CONDUCTOR Sub-command DEFINE [page 383]**. The displayed values are taken from the first conductor in the range and any changes made by the user are applied to all the conductors within the range which are of the same type.

The second way of using **MODIFY** makes use of the other parameters of the **MODIFY** command. The values of the parameters used on the command are applied to all the conductors in the range. This way of using **MODIFY** should be used with care, especially if more than one conductor is specified by **C1** and **C2**, and the range includes conductors of more than one type. It is possible to create conductors with invalid data, since some of the parameters have different meanings for different conductor types. However it is very powerful for changing parameters which apply to all conductors, such as the current density (**CURD**).

**N.B.** Only the second way of using **MODIFY** is available from the GUI.

Expressions can also be used to good effect in this second way of using **MODIFY**. The expressions are evaluated for each conductor; within each evaluation any parameters which are referenced have

the correct values for the conductor concerned. However, the original values are used in each expression. Thus setting **A=A\*2**, **CURD=CURD/A/B** would set the current density using the original value of **A**, not the updated value.

- Example: **MODIFY**ing the conductor defined above.

Sub-commands: Define, Modify, Erase, Print, Write and Quit; ! for help.  
**COND > modi 1 1**

Modify conductors.

Modifying conductor 1.

Specify new values by:

giving values positionally

OR assigning **PARAMETER=value**

OR hitting <return> to accept all defaults.

Conductor 1: Solenoid around Y' axis.

Local coordinate system 1

**XCENTRE(5), YCENTRE(3), ZCENTRE(2)**

**COND >**

Solenoid cross section in X'Y' plane.

**X1(3), Y1(3), X2(3), Y2(5)**

**COND > x1=2 x2=2**

Solenoid cross section in X'Y' plane.

**X3(5), Y3(5), X4(5), Y4(3)**

**COND > x3=4,,4**

Solenoid cross section curvatures.

**CU1(0), CU2(0), CU3(0.05000), CU4(0)**

**COND >**

Current density, symmetry and drive label

**CURD(10), SYMMETRY(1), PHASE(ONE)**

**COND >**

Sub-commands: Define, Modify, Erase, Print, Write and Quit; ! for help.

**COND > modi 1 \* curd=curd/2**

Modify conductors.

## The CONDUCTOR Sub-command PRINT

### Menu route

**DEFINE -> Conductors -> Print data**

### Command line parameters

Sub-command	<b>PRINT</b>	
Parameter	Default	Function
C1	1	Number of first conductor to be printed.
C2	*	Number of last conductor to be printed. C2 can be set to * to indicate the highest numbered conductor.

The **PRINT** sub-command lists the parameters of the selected range of conductors to the terminal and to the log file. It uses the same format for the prompts in **DEFINE** and **MODIFY**.

## The CONDUCTOR Sub-command **QUIT**

### Menu route

**DEFINE** -> **Conductors** -> **Return**

### Command line parameters

Sub-command	<b>QUIT</b>
No Parameters	

The **QUIT** sub-command leaves the **CONDUCTOR** command and returns to the top-level commands.

## The CONDUCTOR Sub-command **WRITE**

### Menu route

**DEFINE** -> **Conductors** -> **Write data to a file**

### Command line parameters

Sub-command	<b>WRITE</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of file.

The **WRITE** sub-command creates a command input **FILE** containing the **CONDUCTOR** command, the set of **DEFINE** sub-commands for all the conductors currently defined and the **QUIT** sub-command.

In the Pre-Processor, conductor files are useful for copying the conductors from one data set to another, or for transferring the data into the Post-Processor. The actual commands used to define the conductors are also included in any file created by the Pre-Processor top level command **WRITE**.

If no file name extension is given, the program adds the extension *cond*.

Conductor definition (**cond**) files can be read:

- into the Pre-Processor using [The READ Command \[page 481\]](#) or the 'built-in' command, **\$ COMINPUT** (see [Command Input Files \[page 61\]](#)),
- into the Modeller using [The IMPORT Command \[page 235\]](#),
- and into the Post-Processor
- using [The CONDUCTOR Command \[page 727\]](#) with **ACTION=IMPORT**.

For hints on how to write compatible command scripts for all 3 programs see [Conditional Commands \[page 52\]](#).

## The **DEFINE** Command

---

### Menu route

**DEFINE -> Define new mesh -> Finite element mesh**  
**DEFINE -> Define new mesh -> 8 or 20 node conductors**

### Command line parameters

Command	<b>DEFINE</b>	
Parameter	Default	Function
<b>TYPE</b>	<b>MESH</b>	Type of elements to be defined.
		<b>CONDUCTOR</b> 8 or 20 node conductor elements
		<b>MESH</b> Finite elements mesh
<b>KEEP</b>	<b>YES</b>	Keep existing construction lines
		<b>NO</b> Construction lines removed at start of <b>DEFINE</b> sequence.
		<b>YES</b> Construction lines kept from any previous <b>DEFINE</b> commands.
<b>THREED</b>	<b>NO</b>	Use 3D Viewer
		<b>NO</b> No 3D Viewer
		<b>YES</b> 3D View starts when first layer is complete.

**DEFINE** is the command which puts the Opera-3d Pre-Processor into its finite element mesh creation mode. The Pre-Processor only supports one type of finite element mesh construction: a surface defined by a set of facets is specified and is then extruded or swept through space to create a volume discretization. The user input is structured by the program. Points used to define the geometry are defined first, followed by the surface facets and the element subdivision of these facets. It is possible to define more points while facets are being defined, and more facets while subdivisions are being defined. The surface of facets is then extruded or swept through space thus creating a series of layers of volumes. The topology of the initial surface is maintained during the extrusion operations, but the point coordinates can be changed in the new surface created by each extrusion or sweep. Once the complete model has been created, the volumes in the layers of the mesh can be assigned material attributes and boundary conditions can be assigned to the facets of the volumes.

The **TYPE** parameter allows definition of the finite element mesh for the solution of the model (**TYPE=MESH**) or 8 or 20 node brick conductor elements (**TYPE=CONDUCTOR**). After definition conductor elements are copied into the conductor storage area as **BR8** or **BR20** conductors with total current of

1 Amp and can only be modified by use of the **CONDUCTOR** sub-commands. (N.B. The cross sections of conductors can be only triangles or quadrilaterals.) The finite element mesh can be modified using **MODIFY** and have extra layers added using **EXTEND**.

The **KEEP** parameter affects construction lines defined during a previous use of the **DEFINE** command. If **KEEP=NO**, any construction lines which already exist are removed before any new data is defined.

The **THREED** parameter can be used to request the 3D Viewer to be used during the **DEFINE** command to display the outline of the model. When **THREED=YES** is selected, the viewer is started when the first layer has been created and is updated whenever the two dimensional display is refreshed.

Labels are assigned to each entity (i.e. point, line, facet or volume) in the mesh. All entities have the label **ALL**, facets have boundary condition names and volumes have material names, potential types and element types. Additional labels can be added or removed with [The LABEL Command \[page 458\]](#). These labels can be used to select subsets of the entities in [The DISPLAY Command \[page 436\]](#).

Each time the **DEFINE** command is used to create a finite element mesh, a new mesh is started. Up to 100 meshes can be defined, each one forming part of any data set prepared for analysis. The user is responsible for ensuring that the meshes are contiguous at their interfaces.

## Finite Element Meshing

The elements used by the analysis programs are created by subdividing volumes created by the Pre-Processor. These volumes are formed by extruding facets on the base plane through space. There are two types of mesh available.

- If all the facets are made up from 3 or 4 lines with regular subdivisions, the volumes can be meshed using hexahedra (which can be degenerate). Regular subdivision means that for 4 sided facets, opposite sides must have the same subdivision and for 3 sided facets, two sides must have the same subdivision.
- Any model can be meshed with tetrahedra. Tetrahedral meshes can be generated in extruded polygonal volumes with no restrictions on the subdivision of the edges. The points defining any polygonal facet must be coplanar. Polygonal facets can have up to 100 edges.

In either type of mesh, the lines can be straight (2 points) or quadratic curves (3 points). The position of the mid-side point of quadratic curves affects the subdivision of the volumes. If it is not central there will be smaller elements at the end of the side to which it is closer. Any combination of straight and curved lines may be used. Thus, facets can be anything from 3 noded triangles to many-noded curved sided polygons. In the same way extrusions can be linear or quadratic. Quadratic extrusions have mid-side points which can be moved independently, allowing definition of curves or non-uniform subdivision in the extrusion direction.

## DEFINE start up sequence and modes

### Coordinate system

After the **DEFINE** command has been issued, the user will be asked to select the default coordinate system to be used for input and display of the initial surface (base plane). There are 3 pre-defined systems

Pre-defined Coordinate Systems		
Keyword	Menu item	Coordinate system
<b>XY</b>	XY plane, extrude in Z	Same as global, extrusion direction Z.
<b>YZ</b>	YZ plane, extrude in X	U=Y, V=Z, W=X, extrusion direction X.
<b>ZX</b>	ZX plane, extrude in Y	U=Z, V=X, W=Y, extrusion direction Y.
<b>NEW</b>	(keyboard only)	Define a new coordinate system.

However any other right-handed system can be defined to give the orientation of the base plane and the default extrusion direction. The coordinate system is defined by its origin, and [Euler Angles \[page 83\]](#). (This coordinate system can be changed during **DEFINE** using **Change View** menu item or the **V** cursor hit.) The most appropriate system should be selected. The default extrusions will be normal to the plane selected and unless a coordinate triple is specified for a point, the point will be given the default coordinate of the plane. Once a plane has been selected, subsequent input of point coordinates is specified in a local U, V, W coordinate system, where U and V are in the plane and W is out of the plane.

### W coordinate

The user is next asked to input the default W coordinate of the plane, for points whose W coordinate is not explicitly defined.

### Window size

Finally, the user is asked to input window size to be used for subsequent graphical display of the point and surface data. This is only an initial specification and can be changed at any time subsequently. The input request is for **umin**, **umax**, **vmin** and **vmax** which may be supplied in free format. The screen is cleared at this point and the program enters Point Definition Mode.

The graphics display indicates, at the top, the mesh number and the plane or extrusion layer number currently being worked on, and, at the bottom, the definition or name of the local (UVW) coordinate system.

## Aspect Ratio Searching

When the user selects a point or line on the screen, the program searches for the point or line nearest to the cursor position in real coordinate space. If the horizontal and vertical axes limits are very different, i.e. if

$$\alpha = \frac{u_{max} - u_{min}}{v_{max} - v_{min}} \quad (4.1)$$

and  $\alpha \gg 1$  or  $\alpha \ll 1$  it might be difficult to select the intended object.

To make it possible to select correctly in such circumstances, aspect ratio searching can be switched on or off using cursor command **z** or menu item **Aspect ratio search**. When it is switched on, the program makes use of the window aspect ratio ( $\alpha$ ), to find the nearest object.

## Defining Meshes with the GUI

When the **DEFINE** command is used with the GUI, the Base Plane Definition Modes are presented on one menu:

Define Baseplane Menu	
Fast Polygon Input	To define points line and facets using grids or construction lines (see <a href="#">Fast Polygon Input Menu and Cursor Hits [page 403]</a> ).
Point Input	To define points using keyboard input of coordinates, grids and construction lines (see <a href="#">The Point Mode Menu and Cursor Hits [page 399]</a> ).
Facet Input	To select points to form facets (see <a href="#">Facet Mode Menu and Cursor Hits [page 406]</a> ).
Subdivision	To subdivide lines (facet sides) to set the finite element mesh size (see <a href="#">Facet Subdivision Menu and Cursor Hits [page 411]</a> ).
Extrude	To create and edit the first layer (coordinates, materials and boundary conditions (see <a href="#">Extrusion DialogBox [page 415]</a> ).
Escape from baseplane	To create the first layer of size 1 with no editing (if there is at least one facet).

The two input methods, **Fast Polygon Input** and **Point Input** with **Facet Input** can be used separately or together to create the base plane facets.

There are several additional points which should be noted:

- All the data for the first layer should be defined first. The **Define new mesh** option should be used to define the first layer and the **Extend existing mesh** option for one or more additional layers (see [The EXTEND Command \[page 448\]](#)).

- Not all options are available from the menus. For example, the coordinate system menu only allows the 3 pre-defined systems.
- It is not possible to abort the definition; the layer must be completed. However, as soon as there is at least one facet, the menu option **Escape from baseplane** is available. This completes all remaining modes of the baseplane definition and extrudes by one unit. The model can be subsequently edited using [The REDEFINE Command \[page 483\]](#).

## Defining Meshes with the Keyboard

In keyboard mode, the program steps through the modes of the **DEFINE** command sequentially, each one being terminated with a **Q** cursor hit. During Point Definition Mode, Fast Polygon Input can be accessed; during Facet Definition Mode, Group Operations can be accessed. There are cursor and text commands which can be used to move between the various modes in almost any order.

## Construction Lines and Grids

Whenever point coordinates can be defined, they can be supplied graphically using construction lines or grids.

### Construction lines

Construction lines are straight lines (**LINE**) or arcs (**ARC**). The arc is in fact a linear interpolation in the cylindrical polar coordinate system. It will only be an arc of a circle if the radial coordinates of the end points are the same. The definitions can be given numerically or by choosing points which have already been defined: **LINEs** are defined by two **<space>** cursor hits; **ARCs** are defined by either a **c** and two **<space>**s to specify the centre and two end points or by 3 **<space>** cursor hits to specify points on a circle. Cursor defined **LINEs** extend beyond the points which define them. Cursor defined **ARCs** are minor arcs if defined by centre and end points or complete circles if defined by 3 points on the circumference.

The menu item **Enter C\_lines** or cursor hit **x** can be used for define construction lines.

- *In keyboard mode* this produces a request to specify data defining construction lines, together with the format of the specification.
- *In menu mode* the type of construction line can be selected from a menu and the values supplied via a ParameterBox:

Construction Line Sub-commands	
Command	Parameters and Function
<b>ARC</b>	$uc\ vc\ r1\ t1\ r2\ t2$ An arc centred on ( <b>uc</b> , <b>vc</b> ) starting at polar coordinates( <b>r1</b> , <b>t1</b> ) and ending at ( <b>r2</b> , <b>t2</b> ). Both <b>r</b> and <b>t</b> vary linearly between the end points.

Construction Line Sub-commands	
ARC	<b>CURSOR</b> An arc defined by cursor hits: <b>Either:</b> select points for centre with c and end points with <space>. <b>or:</b> select 3 points on circumference of circle with <space>.
	<b>u1 v1 u2 v2 angle</b> A straight line from ( <b>u1, v1</b> ) to ( <b>u2, v2</b> ) rotated by <b>angle</b> around ( <b>u1, v1</b> ).
	<b>CURSOR</b> A straight line through 2 existing points chosen by <space> cursor hits.
	<b>QUIT</b> End the definition of construction lines.

Points can be defined on the nearest construction line using menu item **On nearest C\_line** or cursor hit **n**, or at the nearest intersection of construction lines using menu item **At C\_line intersection** or cursor hit **x**. The set of intersections also includes the end points of the construction lines. In each case the cursor hit only specifies the U and V coordinates. The W coordinate used is the default coordinate of the plane.

The nearest construction line can be erased using menu item **Remove C\_lines** or cursor hit **E** and the lines can be listed at the terminal using menu item **List C\_lines** or cursor hit **L**.

## Grids

Grids are two-dimensional arrays of points in cartesian or polar coordinates. A grid can be switched on or off following menu item **Grid** or **G** cursor hit.

Grid Sub-commands	
Command	Parameters and Function
<b>CARTESIAN</b>	<i>du dv</i> Define grid with spacing <i>du</i> in the horizontal direction and <i>dv</i> in the vertical.
<b>POLAR</b>	<i>dr dθ</i> Define grid with spacing <i>dr</i> in the radial direction and <i>dθ</i> in the azimuthal direction.
<b>NONE</b>	Remove any existing grid.

When a grid exists, and menu item **At the mouse** or cursor hit <space> selects the closest grid point.

## Point Definition Mode

The user must specify points to define the corners and mid-side points of the facets in the base plane. Additional points can be entered by returning to Point Definition Mode from Facet Definition Mode. It is also possible to copy points and facets in the Facet Group Operations Mode.

Points are defined using the graphics cursor or numerically using the keyboard. The cursor can be used to position points at the cross-hairs position, or at points on construction lines or grids. Keyboard input can be in cartesian or cylindrical polar coordinates with respect to the current local coordinate system origin.

When enough points have been defined, the user should leave the Point Definition Mode and move on to Facet Definition Mode.

Full details of construction lines and grids are give above and the keyboard input and the cursor commands are described in the following sections.

### Keyboard input

Keyboard input mode can be used to specify coordinates of points in

- cartesian coordinates: menu item **Give U, V, W** or cursor hit **c**
- cylindrical polar coordinates: menu item **Give R, Theta, W** or cursor hit **p**. The origin of the polar coordinate system is the same as the UVW system, with the axis in the W direction and zero azimuthal angle (T) in the U direction.

For each point 1, 2 or 3 values or expressions should be specified. The first defines the U or R coordinate. If it is omitted by using a comma at the start of the command line, the value of U or R will be the last value given for U or R. The second value defines the V or T coordinate. If it is omitted the value of V or T will be the last value used for V or T. When keyboard input mode is entered, U and V have default values taken from the cursor position, and R and T are both zero. The third value defines the W coordinate and always defaults to the default value for the plane.

To return to cursor mode type the keyword **CURSOR** or select the **Quit** button in the DialogBox.

### The Point Mode Menu and Cursor Hits

Point Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<space>	At mouse	Define a point at the cursor cross hair position or closest grid point.
<b>A</b>		Abort the <b>DEFINE</b> command. If Point Definition Mode has been re-entered from Facet Definition Mode then <b>A</b> acts like <b>Q</b> .

<b>Point Mode Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
C	Give U, V, W	Switch to input from keyboard in cartesian coordinates UVW.
D	Delete point	Delete point closest to cursor cross hair.
E	Remove C_line	Erase construction line closest to cursor cross hair.
F		Enter Fast Polygon Input Mode (see <a href="#">Fast Polygon Input Menu and Cursor Hits [page 403]</a> ).
G	Grid	Define or remove a grid (see <a href="#">Grids [page 398]</a> ).
H		Display menu help message explaining all the cursor options.
I	Enter C_lines	Input construction line specifications (see <a href="#">Construction lines [page 397]</a> ).
L	List C_lines	List construction line specifications.
M	Move point	Move the nearest point. The point can be repositioned using cursor hits which have the same meanings as for point definition. See <a href="#">The Point Repositioning Mode Menu and Cursor Hits [page 401]</a> .
N	On nearest C_line	Define a point on the nearest construction line with minimum normal distance.
P	Give R, Theta, W	Switch to input from keyboard in local cylindrical polar coordinates RθW.
Q	Go to Facet Definition	Leave Point Definition Mode and move on to Facet Definition Mode.

Point Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
		4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings.
		<b>CURSOR</b> Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.
		<b>RESTORE</b> Return to previous screen size.
		<b>BOUND</b> Use bounding rectangle of geometry.
		<b>&lt;return&gt;</b> Reconstruct at the same size.
<b>T</b>	Show coordinates	Type the coordinates of the point closest to the cross-hairs. The coordinates are given in cartesian (UVW) and polar (RθW) coordinates.
<b>U</b>	Undo	Undo the last move.
<b>V</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
<b>X</b>	At C_line intersection	Define a point at the closest construction line intersection or end.
<b>Z</b>	Aspect-ratio search	Switch aspect ratio searching on or off ( <a href="#">Aspect Ratio Searching [page 396]</a> ).

## The Point Repositioning Mode Menu and Cursor Hits

<b>Point Repositioning Mode Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	At mouse	Reposition the point at the cursor cross hair position or closest grid point.
<b>C</b>	Give U, V, W	Switch to input from keyboard in cartesian coordinates UVW. Coordinates should be entered in free format. Default values of U, V and W are the values prior to the move. They can be accessed via the variables #1, #2 and #3. Type Q to leave point in its present position.
<b>E</b>	Remove C_line	Erase construction line closest to cursor cross hair.
<b>G</b>	Grid	Define or remove a grid (see <a href="#">Grids [page 398]</a> ).
<b>H</b>		Display menu help message explaining all the cursor options.
<b>I</b>	Enter C_lines	Input construction line specifications.
<b>L</b>	List C_lines	List construction line specifications.
<b>N</b>	On nearest C_line	Reposition the point on the nearest construction line with minimum normal distance. This only affects the U and V coordinates; W remains unchanged.
<b>P</b>	Give R, Theta, W	Switch to input from keyboard in local polar coordinates RθW. Coordinates should be entered in free format. Default values of R, θ and W are the values prior to the move. They can be accessed via the variables #1, #2 and #3. Type Q to leave point in its present position.
<b>Q</b>	Return without moving	Leave the point at its previous position.

Point Repositioning Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
		4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings.
		<b>CURSOR</b> Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.
		<b>RESTORE</b> Return to previous screen size.
		<b>BOUND</b> Use bounding rectangle of geometry.
<b>T</b>	Show coordinates	<b>&lt;return&gt;</b> Reconstruct at the same size.
		Type the coordinates of the point closest to the cross-hairs, and select it for repositioning instead of the point already selected. The coordinates are given in cartesian (UVW) and polar (RθW) coordinates.
<b>x</b>	At C_line intersection	Reposition the point at the closest construction line intersection or end. This only affects the U and V coordinates; W remains unchanged.
<b>z</b>	Aspect-ratio search	Switch aspect ratio searching on or off ( <a href="#">Aspect Ratio Searching [page 396]</a> ).

### Fast Polygon Input Menu and Cursor Hits

In Fast Polygon Input Mode, points, lines and facets are all defined together. Points can be positioned at the cursor position and can be accurately positioned using construction lines or grids. Duplicate points are removed and new polygons can be defined using the points of existing polygons.

Mid-side points can be used to define polygon sides which are curved or have non-uniform element sizes. The mid-side points must be between the  $\frac{1}{4}$  and  $\frac{3}{4}$  points along the length of the side (this is verified by [The CHECK Command \[page 378\]](#)). If the mid-side point is not half way along the side the discretization will also vary quadratically, with smaller elements near the corner which is closer to the mid-side point. This can be used to grade the mesh even for straight sided facets.

A mid-side point is defined after menu item **Next point is mid-side** or cursor hit **M** and can be defined using the cursor position, construction lines, grids or points of existing polygons. Mid-point definition can be cancelled using menu item **Cancel** or repeating cursor hit **M**. Points which have been used as corners cannot subsequently be used as mid-side points, and vice-versa.

Although the name *Polygon* is used, 3 or 4 sided facets can be defined for quadrilateral meshing.

Fast Polygon Input Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	At mouse	Define a polygon corner at the cursor cross hair position or closest grid point.
<b>A</b>		Abort the <b>DEFINE</b> command.
<b>B</b>	Backtrack	Remove the last corner from the current polygon.
<b>D</b>	Delete facet	Delete the facet containing the cursor cross hair.
<b>E</b>	Remove C_line	Erase construction line closest to cursor cross hair.
<b>F</b>	Close polygon	Close an open polygon.
<b>G</b>	Grid	Define or remove a grid (see <a href="#">Grids [page 398]</a> ).
<b>H</b>		Display menu help message explaining all the cursor options.
<b>I</b>	Enter C_lines	Input construction line specifications (see <a href="#">Construction lines [page 397]</a> ).
<b>L</b>	List C_lines	List construction line specifications.
<b>M</b>	Next point is mid-side	The next point will be a mid-side point. It can be defined using <b>&lt;space&gt;</b> , <b>O</b> , <b>N</b> or <b>X</b> .
<b>N</b>	On nearest C_line	Define the next corner on the nearest construction line with minimum normal distance.
<b>O</b>	At nearest old point	Define next corner of polygon to be at the nearest point which has already been defined.
<b>Q</b>	Return	Return to Point Definition Mode.

Fast Polygon Input Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.
	<b>CURSOR</b>	Select diagonally opposite corners of the display area with <space> cursor hits.
	<b>RESTORE</b>	Return to previous screen size.
	<b>BOUND</b>	Use bounding rectangle of geometry.
	<return>	Reconstruct at the same size.
<b>x</b>	At C_line intersect	Define the next corner at the nearest construction line intersection or end.

## Facet Definition Mode

Facets are defined by connecting the points on the base plane. Facets can be triangles, quadrilaterals or higher-order polygons with straight or curved edges. Points are selected in sequence around each facet:

- **corners:** menu items **Corner ... no auto-close** or **polygon corner** (cursor hits **c** or **p**). These are equivalent except that if a polygon corner is defined, the facet is not limited to four sides.
- **mid-side points:** menu item **Mid-side** (cursor hit **m**)

**Close** or cursor hit **f** can be used to close the facet. To extend a polygon to more than 4 sides, **polygon corner** or cursor hit **p** must be used for one of the first 4 corners.

Facets with 4 straight sides can be defined more quickly by selecting the 4 corner points with **Corner ... auto-close after 4** or cursor hit <space>.

Sides with mid-side points are quadratic. The mid-side points must be between the  $\frac{1}{4}$  and  $\frac{3}{4}$  points along the length of the side (this is verified by [The CHECK Command \[page 378\]](#)). If the mid-side point is not half way along the side the discretization will also vary quadratically, with smaller elements near the corner which is closer to the mid-side point. This can be used to grade the mesh even for straight sided facets.

Points which have been used as corners cannot subsequently be used as mid-side points, and vice-versa.

Additional points can be defined by returning to Point Definition Modes (menu item **Return to points** or cursor hit **N**).

Facets can be copied and points transformed using the Facet Group Operations Mode which can be entered using menu item **Go to Group Ops.** or cursor hit **G**.

Care must be taken to ensure that the entire base plane is covered with facets. When the base plane is complete, menu item **Go to Subdivisions** or cursor hit **Q** leaves Facet Definition Mode and moves on to Base Plane Subdivision Mode. It is possible to return to Facet Definition Mode from Base Plane Subdivision Mode in order to define more facets.

Full details of the cursor commands are given in the following section.

## Facet Mode Menu and Cursor Hits

Facet Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	auto-close after 4	Select nearest point as a corner. Facet closes automatically after 4 sequential <b>&lt;space&gt;</b> hits.
<b>A</b>		Abort <b>DEFINE</b> command.
<b>C</b>	no auto-close	Select nearest point as a corner.
<b>D</b>	Delete facet	Delete the facet containing the cross-hairs.
<b>E</b>	Erase last point	Forget the last point selected for the current face.
<b>F</b>	Close	Close the current face (after at least 3 corners).
<b>G</b>	Go to Group Ops.	Enter <b>Facet Group Operations Mode</b> [page 407] to group points for transformations or facets for copying.
<b>H</b>		Display menu help message explaining all the cursor options.
<b>M</b>	Mid-side	Select nearest point as a mid-side point. This cannot be used for the first point of a facet. The facet is closed automatically if the mid-side point is on the fourth side unless <b>P</b> has been used to define a corner.
<b>N</b>	Return to points	Enter Point Definition Mode to define or move points.

Facet Mode Menu and Cursor Hits												
Cursor hit	Menu item	Function										
P	polygon corner	Select nearest point as a corner. Using this, rather than <space> or C identifies the facet as a polygon which can have more than 4 sides.										
Q	Go to Subdivisions	Leave the facet definition mode and move on to Base Plane Subdivision Mode.										
R	Re-draw picture	<p>Reconstruct the display. The program requests a new size. Valid replies are:</p> <table> <tr> <td>4 numeric values</td> <td><i>umin, umax, vmin, vmax</i> Default values are the current settings.</td> </tr> <tr> <td>CURSOR</td> <td>Select diagonally opposite corners of the display area with &lt;space&gt; cursor hits.</td> </tr> <tr> <td>RESTORE</td> <td>Return to previous screen size.</td> </tr> <tr> <td>BOUND</td> <td>Use bounding rectangle of geometry.</td> </tr> <tr> <td>&lt;return&gt;</td> <td>Reconstruct at the same size.</td> </tr> </table>	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.	CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.	RESTORE	Return to previous screen size.	BOUND	Use bounding rectangle of geometry.	<return>	Reconstruct at the same size.
4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.											
CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.											
RESTORE	Return to previous screen size.											
BOUND	Use bounding rectangle of geometry.											
<return>	Reconstruct at the same size.											
V	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.										
Z	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.										

## Facet Group Operations Mode

There are two Group Operations which can be performed on the base plane: the coordinates of groups of points can be transformed and groups of facets can be copied:

- A point is added into the point group using menu item **Select/de-select point** or cursor hit **N**. This selects the nearest point or if it is repeated for a selected point it removes the point from the group.
- The coordinates of points in the point group can be transformed using menu item **Transform points** or cursor hit **T**. If there is no point group then all the points are transformed.
- A facet is added to the facet group using menu item **Select/de-select** or cursor hit **F** cursor hit. This selects the facet containing the cross-hairs or if it is repeated for a selected facet it removes the facet from the group.
- The facets in the facet group can be copied using menu item **Copy facets** or cursor hit **c**. If there is no facet group then all the facets are copied. Multiple copies can be made. The program

prompts the user for the number of new copies, e.g. to end up with 8-fold symmetry it is necessary to create 7 new copies.

- The copy operation creates new facets, lines and points, and then transforms the coordinates of the points. For multiple copies the transformation applies between the original points and the first copy, between the first and second copies, between the second and third, etc. The index number of the copy (#**COPY**) can be used in transformation options **CARTESIAN** and **POLAR**.

At any one time there can be either a point group **or** a facet group, not both.

The transformation options are: **CARTESIAN**, **DISPLACE**, **MANGLE**, **MIRROR**, **POLAR**, **PROJECT**, **ROTATE** and **SCALE**.

When all transformations and copies are complete, menu item **Return to facets** or cursor hit **Q** is used to return to Facet Definition Mode. The program prompts for a tolerance which is used to coalesce coincident points. Points are coalesced if the differences in U, V and W coordinates are all less than the tolerance.

Full details of the menu items, cursor commands and the transformation options are given in the following sections.

## Group Operation Menu and Cursor Hits

Group Operations Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>C</b>	Copy facets	Copy the facet group, or all the facets if there is no group.
<b>F</b>	Select/de-select facet	Select or de-select the facet containing the cross-hairs as a member of the facet group.
<b>H</b>		Display menu help message explaining all the cursor options.
<b>N</b>	Select/de-select point	Select or de-select the nearest point as a member of the point group.
<b>Q</b>	Return to Facets	Leave the Facet Group Operations Mode and return to Facet Definition Mode.

Group Operations Menu and Cursor Hits			
Cursor hit	Menu item	Function	
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:	
		4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.
		<b>CURSOR</b>	Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.
		<b>RESTORE</b>	Return to previous screen size.
		<b>BOUND</b>	Use bounding rectangle of geometry.
		<b>&lt;return&gt;</b>	Reconstruct at the same size.
<b>T</b>	Transform points	Transform the point group, or all the points if there is no group.	
<b>U</b>	Undo transform	Undo a point transformation operation.	
<b>Z</b>	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off	

## Group Transformation Options

The transformations can be any combination of the following:

Base Plane Transformation Commands	
Command	Parameters and Function
<b>CARTESIAN</b>	<i>exp_u exp_v exp_w</i> The points are moved to new positions defined by expressions for their ( <i>u</i> , <i>v</i> and <i>w</i> ) coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, ( <b>#COPY</b> ), can also be used.
<b>DISPLACE</b>	<i>du dv dw</i> Displace points in the current viewing local coordinate system by adding ( <i>du</i> , <i>dv</i> , <i>dw</i> ) to their coordinates ( <i>u</i> , <i>v</i> , <i>w</i> ).
<b>MANGLE</b>	<i>u1 v1 angle</i> Reflect point coordinates in line specified by one point ( <i>u1</i> , <i>v1</i> ) and an <i>angle</i> . The <i>W</i> coordinate of the points is not affected.

Base Plane Transformation Commands	
Command	Parameters and Function
<b>MIRROR</b>	$u1\ v1\ u2\ v2$ Reflect point coordinates in line specified by its end points $(u1, v1)$ , $(u2, v2)$ . The W coordinate of the points is not affected.
<b>POLAR</b>	$exp\_r\ exp\_θ\ exp\_w$ The points are moved to new positions defined by expressions for their $(r, \theta$ and $w)$ coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, ( <b>#COPY</b> ), can also be used.
<b>PROJECT</b>	$du\ dv\ dw\ ucentre\ vcentre\ wcentre\ uangle\ vangle\ wangle$ Project the points in the direction $(du, dv, dw)$ until they intersect the XY plane of a coordinate system specified by its origin $(ucentre, vcentre, wcentre)$ and axis rotation angles $(uangle, vangle, wangle)$ .
<b>QUIT</b>	End the sequence of transformations.
<b>ROTATE</b>	$ucentre\ vcentre\ wcentre\ uangle\ vangle\ wangle$ Rotate points by angles $(uangle, vangle, wangle)$ around axes parallel to the local coordinate system and passing through the point $(ucentre, vcentre, wcentre)$ .
<b>SCALE</b>	$ucentre\ vcentre\ factor$ Scale points by multiplying the distance from local coordinate point $(ucentre, vcentre)$ by <i>factor</i> . The W coordinate of the points is not affected.

- Example: to rotate about the origin by angle 30 degrees about the W axis:  
OP-TRANS > **r 0 0 0 0 0 30**
- Example: to scale the model to half its size:  
OP-TRANS > **s 0 0 1/2**
- Example: to move all points onto a circle, radius 10, leaving the azimuthal coordinates the same:  
OP-TRANS > **polar 10 atan2(v;u)\*180/pi w**

## Base Plane Subdivision Mode

This mode is omitted if **DEFINE** is being used for conductor elements.

The volumes created by extruding the initial surface plane (base plane) are treated as super-elements. They are normally divided into smaller elements that are used for the actual finite element approximation. The division of the volume into elements is determined by the subdivision defined for the facet edges and the subdivision specified for each edge created by an extrusion operation.

A default subdivision of 1 is initially set for all facet edges. When all the edges have been updated, the program will reply that the subdivision is complete.

- **If there are no polygonal facets,** regular subdivision is assumed. In this case, when the subdivision is given for a single edge and that edge is one side of a quadrilateral facet, then the subdivision of the opposite edge will be set to the same value. The subdivision will be carried through the mesh until a triangular facet, the edge of the mesh or a facet already set in this operation is reached.
- In models with regular subdivision, triangular facets are meshed by mapping onto a quadrilateral with 2 corners coincident. Therefore triangles must have 2 sides with the same number of subdivisions. The program can check whether this rule has been satisfied using menu item **Check for hex meshing** or cursor hit **c**. The check is also made if Subdivision Mode is left using cursor hit **Q**.
- **If there are polygonal facets,** the subdivision of each edge must be set individually unless all edges are set to the same subdivision.

The subdivision of each edge is uniform, unless the edge is a quadratic line with the mid-point not at the geometric mid-point of the line. In that case the elements at the end of the edge closer to the mid-point will be smaller than those at the other end.

- *In keyboard mode* a facet is selected for subdivision by positioning the cross-hairs just inside a facet, close to the edge to be divided and pressing the <space> bar. The program prompts for the number of subdivisions. Subdivisions can also be set globally, i.e. all edges set to the same subdivision number, using the **G** cursor hit.
- *In menu mode* the subdivision can be set first using menu item **Set subdivision** and can then be applied to a single edge or globally to the entire mesh.

Subdivisions of edges can be changed any number of times, until the required pattern is achieved.

Additional facets can be defined by returning to Facet Definition Mode with menu item **Return to Facets** or menu hit **F**. If this facility is used it is essential to ensure that the mesh subdivision is complete before finally leaving Facet Subdivision Mode. Menu item **Go to Extrusions** or cursor hit **Q** leaves Facet Subdivision Mode and proceeds to Extrusion Mode. It is necessary to make sure that the extrusion direction required has been set using menu item **Change view** or cursor hit **v** before leaving the Subdivision Mode.

Full details of the cursor commands are given in the following section.

## Facet Subdivision Menu and Cursor Hits

Facet Subdivision Menu and Cursor Hits												
Cursor hit	Menu item	Function										
<space>	Apply to line	Select closest edge for its subdivision to be set.										
A		Abort the <b>DEFINE</b> command.										
C	Check for hex meshing	If there are no polygons, check that the subdivisions obey the rules for meshing with quadrilaterals/hexahedra.										
F	Return to Facets	Return to Facet Definition Mode to define additional facets.										
G	Apply globally	Select all edges for their subdivisions to be set to the same value.										
H		Display menu help message explaining all the cursor options.										
Q	Go to Extrusions	Leave the Facet Subdivision Mode and proceed to the Extrusions Mode. If there are no polygons, the subdivisions are checked for quadrilateral/hexahedral meshing.										
R	Re-draw picture	<p>Reconstruct the display. The program requests a new size. Valid replies are:</p> <table> <tr> <td>4 numeric values</td> <td><i>umin, umax, vmin, vmax</i> Default values are the current settings.</td> </tr> <tr> <td><b>CURSOR</b></td> <td>Select diagonally opposite corners of the display area with &lt;space&gt; cursor hits.</td> </tr> <tr> <td><b>RESTORE</b></td> <td>Return to previous screen size.</td> </tr> <tr> <td><b>BOUND</b></td> <td>Use bounding rectangle of geometry.</td> </tr> <tr> <td>&lt;return&gt;</td> <td>Reconstruct at the same size.</td> </tr> </table>	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.	<b>CURSOR</b>	Select diagonally opposite corners of the display area with <space> cursor hits.	<b>RESTORE</b>	Return to previous screen size.	<b>BOUND</b>	Use bounding rectangle of geometry.	<return>	Reconstruct at the same size.
4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.											
<b>CURSOR</b>	Select diagonally opposite corners of the display area with <space> cursor hits.											
<b>RESTORE</b>	Return to previous screen size.											
<b>BOUND</b>	Use bounding rectangle of geometry.											
<return>	Reconstruct at the same size.											
V	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.										
X		Leave the Facet Subdivision Mode and proceed to the Extrusions Mode. No checking is done.										
Z	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.										

## Extrusions Mode

Once the initial surface set of facets has been defined and subdivisions assigned to the edges, the program moves on to applying extrusion operations to the set of facets. There must be at least one extrusion operation applied to the set of facets, but many others may be needed to define the complete model. It is also possible to add more extrusions to a completed mesh using the **EXTEND** command. The **EXTEND** Command [page 448] must be used to add the second and subsequent extrusions if the data is defined in menu mode.

- In keyboard mode a single line command defines each extrusion. It consists of two compulsory numeric values with optional keywords. The first numeric value is interpreted as the W-coordinate for the points at the top of the new extrusion layer. The second numeric value gives the number of subdivisions between the planes. The keywords **LINEAR**, **QUADRATIC**, **GLOBAL** and **RELATIVE** specify the type of extrusion. The keyword **NOEDIT** indicates that no editing is required for points on the new mesh plane(s). The keyword **SUBDIVISION** request the program to return to Facet Subdivision Mode and ignore all other items on the command line. **SUBDIVISION** is only available before the first extrusion has been created.
- In menu mode a **LINEAR** or **QUADRATIC** extrusion can be chosen from a menu. The DialogBox shown in the section **Extrusion DialogBox** [page 415] allows the user to give the coordinate of the next plane, the number of subdivisions and to select **GLOBAL** or **RELATIVE**.

The values and options are interpreted as follows:

- **Coordinate of next plane.** The W-coordinate can be interpreted as **GLOBAL** or **RELATIVE**. When a **GLOBAL** coordinate is given for the new plane, the extrusion is formed by projecting the current facet set in the W direction until the W-directed lines intersect with the plane defined by the global value of W. The W-coordinate can also be interpreted as a **RELATIVE** move in the W direction, in which case the coordinate value is added to the W-coordinates of the points in the current plane.
- **Subdivision.** This specifies the number of layers of elements there will be between the two planes. This can be adjusted for the whole layer or for individual W-directed lines using the **MODIFY** command (see **Subdivision Modification Mode** [page 474]).  
The subdivision is ignored when **DEFINE** is being used for conductor elements.
- **Linear or Quadratic.** The W-directed lines can be **LINEAR** or **QUADRATIC**. The lines are created initially linear, i.e. straight, but if **QUADRATIC** is selected, a mid-extrusion plane of points is also created, at the geometric mid-points of the lines. This means that **QUADRATIC** lines can be changed to be curved, or to have non-uniform subdivision by moving the points on the mid-extrusion plane away from the geometric mid-points of the lines. The mid-points should be between the 1/4 and 3/4 points along the length of the extrusion.

Unless **NOEDIT** has been requested, the coordinate values of points in the new plane, and in the mid-extrusion plane if **QUADRATIC** is selected, can be changed after the layer has been created. In some cases it is better to make the new plane(s) exactly the same as the current plane by using a **RELATIVE** coordinate of zero, and subsequently modifying the coordinates to the points using the Point Movement cursor options which are presented next. In other cases, it can be better to request **NOEDIT** and subsequently use the **MODIFY** command to move the points (see **Point Modification Mode** [page 470]).

For editing the point coordinates, the program draws a picture of the new plane and invites points to be moved individually or in groups. Individual points are selected using menu item **Move point** or cursor hit <space> and can be repositioned using the same options as offered for the base plane. Points can be grouped using menu item **Select/de-select point** or cursor hit **x** and transformed using menu item **Transform points** or cursor hit **T**. If no points have been grouped all the points of the plane will be transformed. The transformations are similar to those available in [Facet Group Operations Mode \[page 407\]](#).

The original coordinates are stored for all points moved or transformed so that they can be put back using the menu item **Undo move or transform** or cursor hit **U**.

- In keyboard mode, after the extrusion has been created and all necessary points have been moved, cursor hit **Q** moves on to the next extrusion. The user is given the option of creating another extrusion or not (valid replies are **YES** or **NO**). After a **NO** response, the program moves on to Material Definition Mode, or, when **DEFINE** is being used for conductor elements, copies the data into the conductor database for modification with [The CONDUCTOR Sub-command MODIFY \[page 387\]](#).
- In menu mode, after the extrusion has been created and all necessary points have been moved, menu item **Finish editing** moves on to Material Definition Mode, or, when **DEFINE** is being used for conductor elements, copies the data into the conductor database for modification with [The CONDUCTOR Sub-command MODIFY \[page 387\]](#). Further extrusions can be defined using [The EXTEND Command \[page 448\]](#).

Full details of the extrusion command line, the point movement cursor commands and the transformation options are given in the following sections.

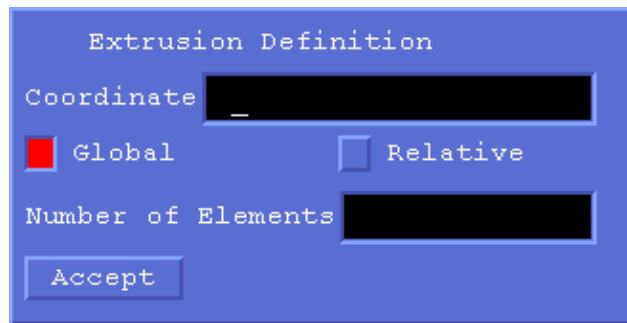
## Extrusion Command Lines.

Extrusion Command Lines	
Two compulsory numeric values:	
<i>first_value</i>	W coordinate ( <b>GLOBAL</b> or <b>RELATIVE</b> ) of next plane.
<i>second_value</i>	Number of subdivisions in the layer.
Optional keywords:	
<b>LINEAR</b> (default) or <b>QUADRATIC</b>	W-directed lines <b>LINEAR</b> (no 'Mid-extrusion Plane'), or <b>QUADRATIC</b> (with 'Mid-extrusion plane').
<b>GLOBAL</b> (default) or <b>RELATIVE</b>	W coordinate is <b>GLOBAL</b> or <b>RELATIVE</b> to previous plane.
<b>NOEDIT</b>	No point movements are required on the new plane (s).
<b>SUBDIVISION</b>	Return to Facet Subdivision Mode (only available before the first extrusion).

- Example: to extrude to W=10 with 3 subdivisions and quadratic W-directed lines:  
**OP-EXTRUDE > 10 3 global quadratic**
- Example: to extrude by adding 5 to the W coordinates of the current plane with 4 subdivisions and linear W-directed lines; no point movements are necessary on the new plane:  
**OP-EXTRUDE > relative 5 4 linear noedit**

## Extrusion DialogBox

In the DialogBox, the values of coordinate and number of elements must be supplied. **Global** and **Relative** are options.



## Point Selection Menu and Cursor Hits

Point Selection Menu and Cursor Hits		
Cursor hit	Menu Item	Function
<space>	Move point	Select point nearest cursor to be moved. It can be repositioned using <b>The Point Repositioning Mode Menu and Cursor Hits [page 416]</b> .
H		Display menu help message explaining all the cursor options. (More help available after point selection.)
K	Select/de-select point	Select point nearest the cursor to be transformed. Repeating K for a selected point de-selects it.

Point Selection Menu and Cursor Hits												
Cursor hit	Menu Item	Function										
<b>Q</b>	Finish Editing	Leave this plane. If on a mid-extrusion plane, move on to the top plane of the new layer. Otherwise the program asks about the next extrusion ( <i>keyboard mode</i> ) or moves on to material definitions ( <i>menu mode</i> ).										
<b>R</b>	Re-draw picture	<p>Reconstruct the display. The program requests a new size. Valid replies are:</p> <table> <tr> <td>4 numeric values</td> <td><i>umin, umax, vmin, vmax</i> Default values are the current settings.</td> </tr> <tr> <td><b>CURSOR</b></td> <td>Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.</td> </tr> <tr> <td><b>RESTORE</b></td> <td>Return to previous screen size.</td> </tr> <tr> <td><b>BOUND</b></td> <td>Use bounding rectangle of geometry.</td> </tr> <tr> <td><b>&lt;return&gt;</b></td> <td>Reconstruct at the same size.</td> </tr> </table>	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.	<b>CURSOR</b>	Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.	<b>RESTORE</b>	Return to previous screen size.	<b>BOUND</b>	Use bounding rectangle of geometry.	<b>&lt;return&gt;</b>	Reconstruct at the same size.
4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.											
<b>CURSOR</b>	Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.											
<b>RESTORE</b>	Return to previous screen size.											
<b>BOUND</b>	Use bounding rectangle of geometry.											
<b>&lt;return&gt;</b>	Reconstruct at the same size.											
<b>T</b>	Transform points	Define general transformations for the points on the plane. (Use <b>K</b> to select points, or all points will be transformed.) The transformation options are given in <a href="#">Plane Transformation Options [page 418]</a>										
<b>U</b>	Undo move or transform	Undo the last move or transform operation.										
<b>V</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.										
<b>Z</b>	Aspect-ratio search	Switch <a href="#">Aspect Ratio Searching [page 396]</a> on or off.										

## The Point Repositioning Mode Menu and Cursor Hits

<b>Point Repositioning Mode Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	At mouse	Reposition the point at the cursor cross hair position.
<b>C</b>	Give U, V, W	Switch to input from keyboard in cartesian coordinates UVW. Coordinates should be entered in free format. Default values of U, V and W are the values prior to the move. They can be accessed via the variables #1, #2 and #3. Type Q to leave point in its present position.
<b>E</b>	Remove C_line	Erase construction line closest to cursor cross hair.
<b>H</b>		Display menu help message explaining all the cursor options.
<b>I</b>	Enter C_lines	Input construction line specifications.
<b>L</b>	List C_lines	List construction line specifications.
<b>N</b>	On nearest C_line	Reposition the point on the nearest construction line with minimum normal distance. This only affects the U and V coordinates; W remains unchanged.
<b>P</b>	Give R, Theta, W	Switch to input from keyboard in local polar coordinates RθW. Coordinates should be entered in free format. Default values of R, θ and W are the values prior to the move. They can be accessed via the variables #1 , #2and #3. Type Q to leave point in its present position.
<b>Q</b>	Return without moving	Leave the point at its previous position.

Point Repositioning Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
		4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings.
		<b>CURSOR</b> Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.
		<b>RESTORE</b> Return to previous screen size.
		<b>BOUND</b> Use bounding rectangle of geometry.
		<b>&lt;return&gt;</b> Reconstruct at the same size.
<b>T</b>	Show coordinates	Type the coordinates of the point closest to the cross-hairs, and select it for repositioning instead of the point already selected. The coordinates are given in cartesian (UVW) and polar (RθW) coordinates.
<b>x</b>	At C_line intersection	Reposition the point at the closest construction line intersection or end. This only affects the U and V coordinates; W remains unchanged.
<b>z</b>	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.

## Plane Transformation Options

The transformations can be any combination of the following:

Plane Transformation Commands	
Command	Parameters and Function
<b>CARTESIAN</b>	$exp\_u \ exp\_v \ exp\_w$ The points are moved to new positions defined by expressions for their ( $u$ , $v$ and $w$ ) coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, (# <b>COPY</b> ), can also be used.
<b>DISPLACE</b>	$du \ dv \ dw$ Displace points in the current viewing local coordinate system by adding ( $du$ , $dv$ , $dw$ ) to their coordinates ( <b>U</b> , <b>V</b> , <b>W</b> ).
<b>POLAR</b>	$exp\_r \ exp\_\theta \ exp\_w$ The points are moved to new positions defined by expressions for their ( $r$ , $\theta$ and $w$ ) coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, (# <b>COPY</b> ), can also be used.
<b>PROJECT</b>	$du \ dv \ dw \ ucentre \ vcentre \ wcentre \ uangle \ vangle \ wangle$ Project the points in the direction ( $du$ , $dv$ , $dw$ ) until they intersect the XY plane of a coordinate system specified by its origin ( <b>ucentre</b> , <b>vcentre</b> , <b>wcentre</b> ) and axis rotation angles ( <b>uangle</b> , <b>vangle</b> , <b>wangle</b> ).
<b>QUIT</b>	End the sequence of transformations.
<b>ROTATE</b>	$ucentre \ vcentre \ wcentre \ uangle \ vangle \ wangle$ Rotate points by angles ( <b>uangle</b> , <b>vangle</b> , <b>wangle</b> ) around axes parallel to the local coordinate system and passing through the point ( <b>ucentre</b> , <b>vcentre</b> , <b>wcentre</b> ).
<b>SCALE</b>	$ucentre \ vcentre \ factor$ Scale points by multiplying the distance from local coordinate point ( <b>ucentre</b> , <b>vcentre</b> ) by <b>factor</b> . The W coordinate of the points is not affected.

In menu mode the transformation commands are offered as menu items and the parameters are supplied using parameter boxes.

- Example: to shift the selected points sideways by 5 units in the U direction:  
OP-TRANS > **displace 5 0 0**
- Example: to rotate the plane by -30 degrees around a line parallel to the V axis and passing through W=10:  
OP-TRANS > **rotate 0 0 10 0 -30 0**
- Example: to double the U and halve the V coordinates of the points:  
OP-TRANS > **cartesian u\*2 v/2 w**

## Material Definition Mode

The extrusion operations create a discretization of space. It is now necessary to set the material and mesh properties within each volume. To do this, the program presents the user with each layer in turn and the user sets values which over-ride the default or current settings within each volume.

- In keyboard mode volumes are selected using the cursor. Single volumes are selected using cursor hit <space>. Multiple volumes are selected using **x** in each volume except the last. The last volume should be selected using <space>. After a <space> cursor hit the program prompts for a one line material property definition. A summary of the volume properties can be displayed in each volume using **s**. This gives the name, potential code and element type. A full list of all the properties of an individual volume can be obtained using **l**. Cursor hit **q** moves on to the next layer and **f** finishes definition in all layers. After material definition the program moves on to Boundary Condition Definition Mode.
- In menu mode the menu item **Select/de-select volume** allows volumes to be added or removed from a list. Menu item **Select and define** adds one last volume to the list and causes the program to display a DialogBox into which the material properties can be entered. Other menu items allow the properties to be summarized in all the volumes (**Show**) or to be **Listed** for an individual volume.

Material property definitions consist of two compulsory keywords and several options which add special properties or control setting of several volumes simultaneously.

The first compulsory keyword is the **material name**. This can be (almost) any character string of up to 8 characters, beginning with a letter. There are two pre-defined material names, **AIR** and **NULL**. **AIR** is for any non-conducting volume with a relative permeability and relative permittivity both equal to 1. Volumes with name **NULL** are omitted from the final mesh, enabling the creation of holes in the mesh to represent e.g. electrodes. (The second keyword must be omitted for material name **NULL**.) The definition of each material in terms of its permeability and, if necessary, conductivity is supplied by [The MATERIALS Command \[page 462\]](#), or during the commands which create the analysis data files.

The second compulsory keyword is the **potential type**. The following rules must be followed, depending on which analysis program will be used:

- **TOSCA (magnetostatics)**

**REDUCED** scalar potential *must* be used in a space where source currents are flowing. It is often easiest to make all the **AIR** use reduced scalar potential but a model should not have only **REDUCED** potential volumes which would result in a zero solution.

**TOTAL** scalar potential *should* be used in non-**AIR** volumes. It is also possible to use reduced scalar potential in non-**AIR** volumes but this should only be used as a last resort, if it is not possible to specify a region where the total scalar potential would be single valued.

The total scalar potential becomes multi-valued if there exists a closed path, entirely in total scalar potential volumes, through which a net current flows. TOSCA breaks such a path by the use of automatic cuts, i.e. additional volumes which can have multiple values of potential on the surface. TOSCA gives these volumes additional labels, **POTENTIAL\_CUT***n*, so that they can be selected and displayed in the Post-Processor.

If automatic cuts are disabled (see [SETTINGS sub-command \[page 499\]](#)), it is necessary to use reduced scalar potential volumes or two different total scalar potential boundary conditions with one value inside the coil and another outside (see [Boundary Condition Definition Mode \[page 426\]](#)). In this case the potential difference between the boundaries should exactly balance the current enclosed in the mesh (SI units):

$$\Delta\psi = - \int H \cdot dl = I \quad (4.2)$$

**TOTAL** scalar potential can also be used in any **AIR** volumes which do not contain source currents. This reduces the number of nodes at which the coil field has to be calculated and also improves the accuracy of the total field in situations where the fields from the coils and the magnetic materials almost cancel (shielding models).

To achieve good answers, it is important that coil fields can be represented well by the finite element mesh on the interface between the reduced and total potentials. This implies that the best choice of reduced potential region is a simple shaped region that contains the coils and which has a surface as far as possible from the coil surface. It is not always possible to achieve this, especially when for example the coil is wrapped on an iron surface since in this case the interface must correspond to the coil surface. In such cases the discretization of the interface should be increased to maintain accuracy and the choice of adaptive integration is essential.

Any **VECTOR** potential volumes in a TOSCA model will be treated as **TOTAL** scalar potential.

- **ELEKTRA**

**REDUCED** vector potential must be used in a space where source currents are flowing. It is often easiest to make all the **AIR** use reduced vector potential but a model should not have only **REDUCED** potential volumes which would result in a zero solution.

Total vector potential (keywords **TOTAL** or **VECTOR**) must be used in conducting or non-conducting non-**AIR** volumes and can also be used in any **AIR** volumes which do not contain source currents. This reduces the number of nodes at which the coil field has to be calculated and also improves the accuracy of the total field in situations where the fields from the coils and the magnetic or conducting materials almost cancel.

To achieve good answers, it is important that coil fields can be represented well by the finite element mesh on the interface between the reduced and total vector potentials. This implies that the best choice of reduced potential region is a simple shaped region that contains the coils and which has a surface as far as possible from the coil surface. It is not always possible to achieve this, especially when for example the coil is wrapped on a material surface since in this case the interface must correspond to the coil surface. In such cases the discretization of the interface should be increased to maintain accuracy and the choice of adaptive integration is essential.

- **CARMEN**

**REDUCED** vector potential must be used in a space where source currents are flowing. It is often easiest to make all the **AIR** use reduced vector potential but a model should not have only **REDUCED** potential volumes which would result in a zero solution.

Total vector potential (keywords **TOTAL** or **VECTOR**) must be used in conducting or non-conducting non-**AIR** volumes and the **GAP**. It can also be used in any other **AIR** volumes which do not contain source currents. This reduces the number of nodes at which the coil field has to be calculated and also improves the accuracy of the total field in situations where the fields from the coils and the magnetic or conducting materials almost cancel.

To achieve good answers, it is important that coil fields can be represented well by the finite element mesh on the interface between the reduced and other potentials. This implies that the best choice of reduced potential region is a simple shaped region that contains the coils and which has a surface as far as possible from the coil surface. It is not always possible to achieve this, especially when for example the coil is wrapped on an iron surface since in this case the interface must correspond to the coil surface. In such cases the discretization of the interface should be increased to maintain accuracy and the choice of adaptive integration is essential.

- **SCALA and TOSCA (current flow and electrostatics)**

All volumes are treated as if they use **TOTAL** scalar potential.

- **SOPRANO**

All volumes must use magnetic **VECTOR** potential. The program overrides the assignment of other potential types.

- **TEMPO (steady-state and transient)**

Any potential type can be used.

More information on the use of the different potentials is given in the user guides for the analysis programs.

Optional keywords defining properties are element types (**LINEAR**, **QUADRATIC**), one scalar property (**SCALAR**), three vector properties (**VECTOR**, **CURRENT**, **VELOCITY**) and a lamination **PACKINGFACTOR**. The vector properties can be defined as expressions in terms of the coordinates (**X**, **Y** and **Z**) to allow spatial variation.

- **LINEAR** and **QUADRATIC** define the **element types** to be used. **LINEAR** elements are 8-noded hexahedra or 4 noded tetrahedra; **QUADRATIC** elements are 20-noded isoparametric hexahedra or 10-noded isoparametric tetrahedra. Both element types can be used in one model. (Only linear elements can be used in ELEKTRA and SOPRANO.)
- In **electrostatic analysis** (SCALA and TOSCA), the **volume electric charge density** is specified by **SCALAR** followed by one numeric value.
- In models with **permanent magnets** the coercive force is specified by the material properties (linear materials) or the BH curve (nonlinear materials), but the direction is specified here by **VECTOR** followed by 3 **Euler Angles** [page 83] which define a local coordinate system for the volume. The permanent magnetic field is in the local negative Z direction.
- In models with **anisotropic materials** the local coordinate system for the material is defined by **VECTOR** followed by 3 expressions representing the **Euler Angles** [page 83]. If the material is laminated, (TOSCA only) the **packing factor** is specified by **PACKFACTOR** followed by one numeric value which modifies the permeability in the directions normal and parallel to the laminations (the local XY plane). If the material is to be specified by **multiple properties**, only the **VECTOR** property is required here to define the local coordinate system for the material. In either case, anisotropy must be switched on in the **SOLVERS MATERIAL** sub-command [page 495].
- In **eddy current analyses** (CARMEN and ELEKTRA), an assigned current density can be specified by **CURRENT** followed by 3 expressions representing the x, y and z-components of the current density. This only applies to non-conducting **VECTOR** potential volumes.

- In **motion induced eddy current analyses** (ELEKTRA/VL), the linear **velocity** is specified by **VELOCITY** followed by 3 expression representing the x, y and z-components of velocity in units of length/s. This can only be defined for **VECTOR** potential volumes. Rotational motion is given by **SCALAR** followed by the angular velocity in rpm. The rotation is always around the global Z axis.
- In **thermal analysis** (TEMPO/ST and TEMPO/TR) a uniform heat source in the volume can be specified by **SCALAR** followed by the value of heat density.

The three values associated with **CURRENT**, **VECTOR** and **VELOCITY** can be given as expressions in terms of X, Y and Z so that vectors and properties which vary throughout the volumes can be specified. The computed directions can be viewed using [The DISPLAY Command \[page 436\]](#).

Optional keywords for setting additional volumes with the same definition are **ALL**, **FROM**, **TO** and **KEEP**. **ALL** indicates that all the volumes in the layer(s) should be set the same. Both **FROM** and **TO** should be followed by numeric values specifying a layer number to specify a range of layers to be set the same. If either is omitted, the current layer number is assumed. The numeric value after **TO** can be specified as \* to indicate the top layer. **KEEP** specifies that the current selection of volumes should be kept, following the definition of materials, so that another definition can be given. This can be used when setting corresponding volumes in a non-contiguous set of layers.

The material names, potential types and element types are stored as labels on each volume and can be used to select parts of the model in [The DISPLAY Command \[page 436\]](#).

Full details of the cursor commands and material definitions are given in the following sections.

## Material Definition Mode Menu and Cursor Hits

Material Definition Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<space>	Select and define	Select the volume containing the cross-hairs to give a material definition for this volume and any selected with <b>K</b> .
<b>F</b>	Finish	End the volume material definition mode and move on to Boundary Condition Mode.
<b>H</b>		Display menu help message explaining all the cursor options. (More help available after volume selection.)
<b>K</b>	Select/de-select volume	Keep the volume containing the cross-hairs until a <space> is used. Repeating <b>K</b> for a selected volume, de-selects the volume.

Material Definition Menu and Cursor Hits		
Cursor hit	Menu item	Function
L	List properties	List all the properties of the volume containing the cross-hairs.
Q		Move on to the next extrusion layer or, if at the last layer move on to Boundary Condition Mode.
R	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.
	CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.
	RESTORE	Return to previous screen size.
	BOUND	Use bounding rectangle of geometry.
	<return>	Reconstruct at the same size.
S	Show volumes	Display the material names, potential codes and element types.
V	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
Z	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.

## Material Definition Command Line

The single line definitions of material and other volume properties consist of the following items:

Material Definitions	
Two compulsory keywords:	
<b>first_keyword</b> or <b>QUIT</b>	Material name. Use <b>AIR</b> for air, <b>NULL</b> to omit volume from mesh or a material name. <b>QUIT</b> abandons the currently selected volumes.

Material Definitions	
<b>second_keyword</b>	Potential type: REDUCED, VECTOR or TOTAL. It should not be given for material NULL.
Optional keywords – element types:	
LINEAR (default) or QUADRATIC	LINEAR elements, or QUADRATIC elements.
Optional keywords – volume properties:	
PACK <b>value</b>	Sets packing factor for volume.
SCALAR <b>value</b>	Sets charge density (electrostatics only). Sets angular velocity in rotating volumes [rpm] (ELEKTRA/VL). Sets heat density (TEMPO/ST and TEMPO/TR)
CURRENT <b>jx jy jz</b>	Sets assigned current density.
VECTOR <b>t p s</b>	Sets local coordinate system Euler angles of volume for permanent magnets or anisotropic materials.
VELOCITY <b>vx vy vz</b>	Sets linear velocity [length/s] in moving volumes (ELEKTRA/VL).
Optional keywords – setting multiple volumes:	
ALL	Sets all volumes in layer(s).
FROM <b>value</b>	Sets all layers from layer number given though to the current layer or the layer number given with TO.
KEEP	Keeps current selection of volumes for another material definition.
TO <b>value</b>	Sets all layers from current layer or the layer number given with FROM though to the layer number given. The value may be * to indicate the top layer.

- Example: to set selected volumes to material name IRON in layers 3, 4,..., 7 and 8:  
OP-MATERIALS > iron total from 3 to 8
- Example: to set all volumes in the layer to material name ALCOMAX with vector direction set:  
OP-MATERIALS > alcomax total all vect 90 90 0

## Material Definition DialogBox

In the DialogBox, the default properties correspond to the last volume selected. A material name must be supplied. Unless the material is NULL, one of the potential types must be selected. All the other items are optional and should only be assigned values if needed. The Local XYZ options allow the VECTOR property to be set to appropriate Euler angles to swap the coordinate axes. If another direction is needed, the Euler angles should be typed into the Other vector box.

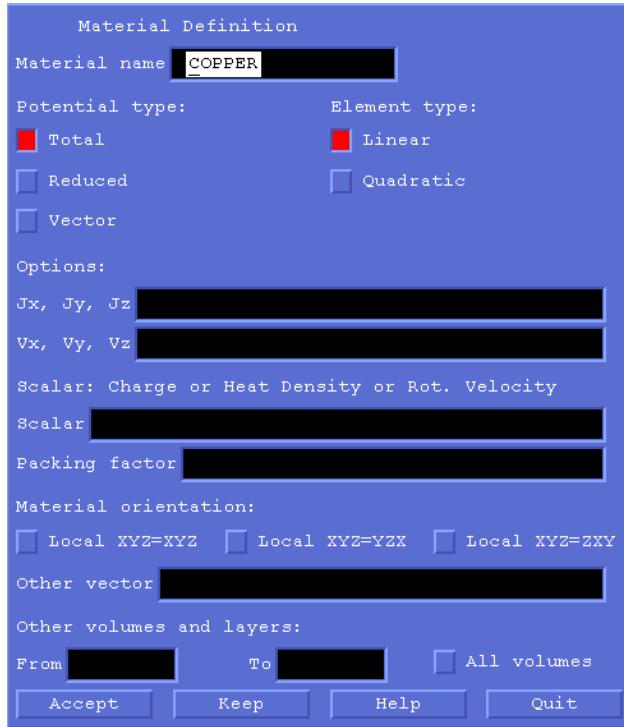


Figure 4.6 The Material Definition DialogBox

## Boundary Condition Definition Mode

Boundary conditions can be set on any surface of any volume in the mesh. In order to achieve this the program presents the facets in separate groups:

*In keyboard mode* there are 4 groups of facets: first the facets on the base plane; second the planes between the first and the last; third the final plane of the mesh; and fourth the extrusion facets normal to the planes, one layer at a time. Facets are selected using the cursor. Single facets are selected using **<space>**. Multiple facets are selected using **K** for each facet except the last. The last facet should be selected using **<space>**. After a **<space>** cursor hit the program prompts for a one line boundary condition definition. Alternatively all the free surfaces of the mesh (all planes and layers) can be set to the same boundary condition using **G**. Free surfaces are those which are on the outside of the mesh or adjacent to a volume with material name **NULL**. A full list of all the conditions on an individual facet can be obtained using **L**. Cursor hit **Q** moves on to the next plane or layer and **S** or **F** finishes definition in all planes or layers. After boundary condition definitions the **DEFINE** command is complete and the program expects another top-level command.

*In menu mode* there are 3 groups of facets: first the facets on the base plane; second the top plane of the mesh; and third the extrusion facets normal to the planes. The menu item **Select/de-select**

**facet** allows facets to be added or removed from a list. Menu item **Select and define** adds one last facet to the list and causes the program to display a DialogBox into which the boundary conditions can be entered. The menu item **List conditions** allows the boundary conditions to be listed for an individual facet.

Boundary conditions can be imposed in eight ways:

- by restricting the **MAGNETIC** or **ELECTRIC** field to be **NORMAL** or **TANGENTIAL**. This sets combinations of the potentials and derivatives as appropriate for the type of analysis and volume concerned:

<b>Normal and Tangential Boundary Conditions</b>			
<b>Normal Magnetic</b>	<b>Tangential Magnetic</b>	<b>Normal Electric</b>	<b>Tangential Electric</b>
<b>TOSCA Magnetic Field Analysis</b>			
$\psi = 0$	$\frac{\partial \psi}{\partial n} = 0$	not allowed	not allowed
<b>SCALA or TOSCA Electric Field Analysis</b>			
not allowed	not allowed	$V = 0$	$\frac{\partial V}{\partial n} = 0$
<b>CARMEN/ELEKTRA Vector potential Volumes</b>			
$A \cdot n = 0, \frac{\partial V}{\partial n} = 0$	$A \times n = 0$ $V = 0$	$A \times n = 0$ $V = 0$	$A \cdot n = 0, \frac{\partial V}{\partial n} = 0$
<b>SOPRANO</b>			
$A \cdot n = 0,$	$A \times n = 0$	$A \times n = 0$	$A \cdot n = 0,$

In the above table  $\psi$  indicates the total or reduced magnetic scalar potential,  $A$  the magnetic vector potential and  $V$  the voltage.

If necessary a potential boundary condition can be assigned in addition to the normal or tangential conditions in order to override the zero values of scalar potential.

- by setting values of magnetic scalar **POTENTIAL**, the **VOLTAGE** or components of the magnetic vector potential (**AX**, **AY** and **AZ**). For SOPRANO, the incident values of magnetic vector potential and voltage can be specified instead (**INAX**, **INAY**, **INAZ** and **INVO**).
- by setting the normal derivative of the magnetic scalar potential (**DERIVATIVE**) or a mixed magnetic scalar potential condition in TOSCA (**PMIX**).
- by imposing a **RADIATION** condition (SOPRANO only).
- by imposing a **PEC** (perfect conductor) boundary condition (CARMEN, ELEKTRA and SOPRANO only). This is functionally equivalent to **NORMAL ELECTRIC** but allows boundaries to be distinguished in the Post-Processor.
- by imposing a **SLIP** condition (CARMEN only) to identify the interface between the stator and rotor.
- by setting a **SYMMETRY** or periodicity condition (ELEKTRA, TOSCA and SCALA only).

- by setting thermal boundary conditions on a TEMPO model: perfect insulator (**INSULATOR**), fixed temperature (**TEMPERATURE**), constant heat flux (**HEATFLUX**) or heat transfer (**HEATTRANSFER** and **TEMPERATURE**).

If potential or derivative boundary conditions are used, a facet can have up to 4 potential boundary conditions – one for the scalar potential and one for each component of the vector potential.

Boundary condition definitions consist of a compulsory keyword and up to two values followed by several options which control setting of several facets simultaneously or clearing boundary conditions.

The compulsory keyword is the boundary condition name. This can be one of **NORMAL**, **TANGENTIAL**, **POTENTIAL**, **VOLTAGE**, **AX**, **AY**, **AZ**, **INAX**, **INAY**, **INAZ**, **INVO**, **DERIVATIVE**, **PMIX**, **RADIATION**, **PEC**, **SLIP**, **SYMMETRY**, **INSULATOR**, **TEMPERATURE**, **HEATFLUX** or **HEATTRANSFER**. The boundary condition name is assigned to the facet as a label. If normal or tangential boundary conditions are applied, the following labels are assigned to the facet as appropriate: **ELECTRIC**, **MAGNETIC**, **NORMELEC**, **NORMMAGN**, **TANGELEC**, **TANGMAGN**.

**CLEAR** followed by a boundary condition name clears that condition from the facet(s).

Boundary conditions have the following effects:

- In **magnetic scalar potential volumes**, setting a constant **POTENTIAL** condition specifies that the tangential components of the field are to be zero. Non-zero total scalar **POTENTIAL** conditions can be used to impose an external field (i.e. an m.m.f. across the model) or to balance enclosed currents in multiply connected geometries. Reduced scalar **POTENTIAL** boundary conditions can only have the value zero.

A constant **DERIVATIVE** specifies the value of the normal field (TOSCA only). The zero **DERIVATIVE** condition is the natural condition of the finite element mesh and need not be assigned explicitly.

- In **electric scalar potential volumes** (electrostatics or current flow), setting a constant **VOLTAGE** condition specifies that the tangential components of the field are to be zero. Non-zero total **VOLTAGE** conditions can be used to impose an external field (i.e. a potential difference across the model).

A constant **DERIVATIVE** specifies the value of the normal field. The zero **DERIVATIVE** condition is the natural condition of the finite element mesh and need not be assigned explicitly.

- In **vector potential volumes**, setting **AX**, **AY** or **AZ** to zero implies that the electric field in that direction is also zero. Only components tangential to the surface are affected. In CARMEN and ELEKTRA, the constant electric **VOLTAGE** boundary condition should also be set on surfaces where the tangential components of electric field are zero. This constant should in general be zero, unless the voltage is providing the driving field.

A second set of boundary conditions (**INAX**, **INAY**, **INAZ**) set the characteristics of the incident field for SOPRANO and also apply the **RADIATION** condition

- **Periodic or SYMMETRY boundaries** (ELEKTRA, TOSCA and SCALA only) specify that the potentials on one surface are equal to the values on another surface (with or without a change of sign). The facets forming such surfaces are labelled here with the boundary condition **SYMMETRY** and are joined together in the **SOLVERS PERIODICITY sub-command [page 497]** by the specification of transformations which map one set of boundary nodes onto the other.

- **SLIP boundaries** (CARMEN only) identify the interface between the stationary and rotating parts of a rotating machine. The volumes on both sides of a slip surface should be modelled using **TOTAL** scalar potential. It is often more convenient to apply the **SLIP** boundary condition using [The SLIP Command \[page 484\]](#).
- **INSULATOR** boundaries (TEMPO only) allow no heat to pass.
- **TEMPERATURE** boundaries (TEMPO only) fix the temperature or specify the ambient temperature if a **HEATTRANSFER** condition also exists.
- **HEATFLUX** boundaries (TEMPO only) specify a constant flow of heat out of the model. The heat flux value can be an expression in terms of coordinates or temperature (**T**). If a function of temperature is given, nonlinear analysis must be selected when creating the analysis database (see the **SOLVE SETTINGS** sub-command [\[page 499\]](#)).
- **HEATTRANSFER** boundaries (TEMPO only) specify a transfer of heat proportional to the temperature difference between the body and the ambient temperature given by a **TEMPERATURE** boundary condition on the same surface. The heat transfer coefficient can be an expression in terms of coordinates or temperature (**T**). If a function of temperature is given, nonlinear analysis must be selected when creating the analysis database (see the **SOLVE SETTINGS** sub-command [\[page 499\]](#)).

In some circumstances it may be necessary to specify the boundary condition values as functions of the node coordinates. The values of conditions **POTENTIAL**, **VOLTAGE**, **AX**, **AY**, **AZ**, **TEMPERATURE**, **HEATFLUX** and **HEATTRANSFER** can be specified as expressions in terms of **X**, **Y** and **Z**. User variables and system constants (**PI**, **MU0**, **EPSILON0** and **C**) can be used in the expressions. The program assigns names for boundary condition expressions and these names can be used as labels in [The DISPLAY Command \[page 436\]](#).

For analysis with CARMEN, ELEKTRA and SOPRANO/SS, non-zero potential boundary conditions (**POTENTIAL**, **VOLTAGE**, **AX**, **AY**, **AZ**, **INAX**, **INAY** or **INAZ**) can be given a drive label so that the associated time-function or phase angle can be assigned by [SOLVERS Command Prompts \[page 491\]](#).

Further information on boundary conditions is given in [Boundary Conditions \[page 615\]](#) and in the Opera-3d Training Course.

The boundary condition names are stored as labels on each facet and can be used to select parts to the model in [The DISPLAY Command \[page 436\]](#).

Optional keywords for setting additional facets with the same definition are **ALL**, **FROM** and **TO**. **ALL** indicates that all the facets in the plane(s) or layer(s) should be set the same. Both **FROM** and **TO** should be followed by a numeric value specifying a plane or layer number to specify a range of planes or layers to be set the same. If either is omitted, the current plane or layer number is assumed. The numeric value after **TO** can be specified as **\*** to indicate the highest numbered plane or layer.

Full details of the menu items and cursor commands and boundary condition definitions are given in the following sections.

## Boundary Condition Definition Mode Menu and Cursor Hits

Boundary Condition Mode Menu and Cursor Hits		
Cursor hit	Menu item	Function
<space>	Select and define	Select the facet nearest the cross-hairs to give a boundary condition definition for this facet and any selected with <b>K</b> .
<b>F</b>		End the boundary condition definition mode (extrusion layers only).
<b>H</b>		Display menu help message explaining all the cursor options. (More help available after facet selection.)
<b>K</b>	Select/de-select facet	Keep the facet containing the cross-hairs until a <space> is used. Repeating <b>K</b> for a selected facet, de-selects the facet.
<b>L</b>	List conditions	List all the boundary condition on the facet containing the cross-hairs.
<b>Q</b>	Finish	Move on to the next plane or layer or, if at the last layer, leave Boundary Condition Mode.
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
		4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings.
		<b>CURSOR</b> Select diagonally opposite corners of the display area with <space> cursor hits.
		<b>RESTORE</b> Return to previous screen size.
		<b>BOUND</b> Use bounding rectangle of geometry.
		<return>    Reconstruct at the same size.
<b>S</b>		Skip to the last plane (planes only).
<b>V</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
<b>Z</b>	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.

## Boundary Condition Command Line Definitions

The single line commands for boundary condition definition consist of the following items:

Boundary Condition Definitions	
One compulsory keyword and values - one of the following:	
<b>AX <i>expression label</i></b>	The X component of the vector potential.
<b>AY <i>expression label</i></b>	The Y component of the vector potential.
<b>AZ <i>expression label</i></b>	The Z component of the vector potential.
<b>CLEAR <i>comp</i></b>	Clear boundary conditions with <i>comp</i> equal to <b>POTENTIAL</b> , <b>AX</b> , <b>AY</b> or <b>AZ</b> . If <b>comp</b> is omitted all boundary conditions are cleared.
<b>DERIVATIVE <i>value</i></b>	The normal derivative of scalar potential.
<b>HEATFLUX <i>expression</i></b>	Constant flow of heat (TEMPO)
<b>HEATTRANSFER <i>expression</i></b>	Heat transfer condition (TEMPO). The normal heat flow is given by $q.n = \text{HEATTRANSFER} * (T - \text{TEMPERATURE})$
<b>INAX <i>value label</i></b>	The incident vector potential, X component (SOPRANO).
<b>INAY <i>value label</i></b>	The incident vector potential, Y component (SOPRANO).
<b>INAZ <i>value label</i></b>	The incident vector potential, Z component (SOPRANO).
<b>INSULATOR</b>	Perfect insulator (TEMPO)
<b>NORMAL <i>field</i></b>	Boundary conditions which restrict <i>field</i> to be normal to the facet. <i>Field</i> can be <b>MAGNETIC</b> or <b>ELECTRIC</b> .
<b>PEC</b>	Perfect conductor boundary condition (CARMEN, ELEKTRA and SOPRANO only).
<b>PMIX <i>value1 value2</i></b>	Mixed scalar potential boundary condition $\frac{\partial \phi}{\partial n} + \text{value1} = \text{value2}$ in TOSCA
<b>POTENTIAL <i>expression label</i></b>	The magnetic scalar potential.
<b>QUIT</b>	Abandons the current selection of faces.
<b>RADIATION</b>	Radiation boundary condition (SOPRANO).

<b>Boundary Condition Definitions</b>	
<b>SLIP</b>	Slip surface periodic boundary condition for a rotating machine.
<b>SYMMETRY</b>	Periodic boundary condition. Facets are connected by rotation and displacement operations defined in the <b>SOLVERS PERIODICITY</b> sub-command [page 497].
<b>TANGENTIAL <i>field</i></b>	Boundary conditions which restrict <i>field</i> to be tangential to the facet. <i>Field</i> can be <b>MAGNETIC</b> or <b>ELECTRIC</b> .
<b>TEMPERATURE <i>expression</i></b>	A fixed temperature or (if <b>HEATTRANSFER</b> also exists) an ambient temperature (TEMPO).
<b>VOLTAGE <i>expression label</i></b>	The electric scalar potential.
Optional keywords - setting multiple facets:	
<b>ALL</b>	Sets all facets in plane(s) or layer(s).
<b>FROM <i>value</i></b>	Sets all planes or layers from number given though to the current plane or layer or the number give with <b>TO</b> .
<b>KEEP</b>	Keeps current selection of facets for another boundary condition definition.
<b>TO <i>value</i></b>	Sets all plane or layers from current plane or layer or the number given with FROM, through to the number given. The value may be * to indicate the top plane or layer.

- Example: to set selected facets to zero scalar potential:

OP-B/C > **pote 0**

## Boundary Condition DialogBox

The default settings correspond to the last facet selected.

The boundary condition can be specified in one of the following ways:

- Potentials: the option should be selected and a value given. For steady-state ac or transient analysis, a label can also be given to enable drive information to be associated with the boundary condition.
- Other boundary condition option buttons: no value is necessary.
- **CLEAR**: the condition to be cleared can be given in the value box.

Other types of boundary condition can be given by not selecting any of the options but giving a keyboard style boundary condition command line in the value box.

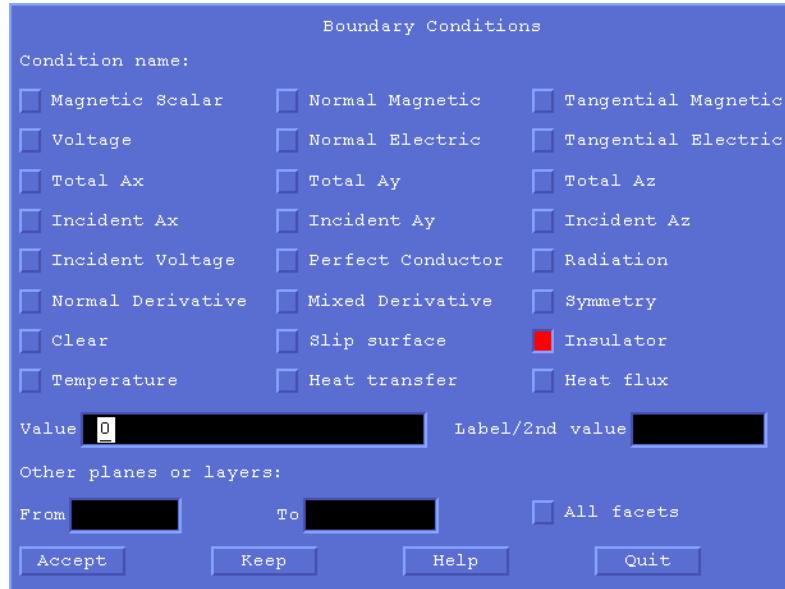


Figure 4.7 The Boundary Condition DialogBox

## The **DEVICE** Command

---

### Menu route

**OPTIONS -> Graphics output**

### Command line parameters

Command	<b>DEVICE</b>
No Parameters	

There are two different graphics implementations of the software:

- **Windows** (available on Microsoft Windows operating systems)
- **X-lib** (available on Linux 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	<b>DEVICE command</b>	Meaning
<b>SCREEN</b>	*	*	graphics displayed on the screen
<b>FILE</b>	*		all graphics commands stored in one file
<b>BOTH</b>	*	*	graphics on the screen and in a file
<b>NONE</b>	*		no graphics, except that the <b>DUMP</b> 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 (Linux) called **VGRAPHICS**. A valid value of **VGRAPHICS** eliminates the initialization prompt.

Other environment variables which affect the software on Linux are:

Variable	Meaning
<b>VFWINDOWW</b>	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 → Preferences → Graphics Window Size** menu item in the Opera Manager.

## The **DISPLAY** Command

---

### Menu route

**DISPLAY -> Display command ... refresh display**

### Command line parameters

Command	<b>DISPLAY</b>	
Parameter	Default	Function
<b>SIZE</b>	10	Size of coordinate space to be displayed. The space extends from ( <b>XORIGIN</b> , <b>YORIGIN</b> , <b>ZORIGIN</b> ) by <b>SIZE</b> in each direction.
<b>XEYE</b>	0	X-coordinate of eye position.
<b>YEYE</b>	0	Y-coordinate of eye position.
<b>ZEYE</b>	100	Z-coordinate of eye position.
<b>PERSPECTIVE</b>	<b>NO</b>	Perspective view switch.  <b>NO</b> Orthogonal projection.
		<b>YES</b> Perspective projection.
<b>ROTX</b>	0	Rotation angle about X-axis to change eye position.
<b>ROTY</b>	0	Rotation angle about Y-axis to change eye position.
<b>ROTATE</b>	0	Rotation angle about viewing direction.
<b>ELEMENT</b>	<b>NO</b>	Element display switch.  <b>NO</b> No element subdivision.  <b>SURFACE</b> Subdivision on volume surfaces.  <b>VOLUME</b> Subdivision within volumes.
<b>MESH</b>	<b>ALL</b>	Mesh number, <b>ALL</b> or <b>NONE</b> .

Command	<b>DISPLAY</b>	
Parameter	Default	Function
<b>TYPE</b>	<b>VOLUME</b>	Type(s) of entities to be displayed. <b>VOLUME, FACET, LINE, POINT, ALL or SAME.</b>
<b>LABEL</b>	<b>NOTAIR</b>	Label(s) on entities.
<b>L1</b>	1	First layer to be displayed.
<b>L2</b>	*	Last layer to be displayed. * means top layer.
<b>COIL</b>	<b>YES</b>	Conductor display switch.
	<b>NO</b>	Conductors not displayed.
	<b>YES</b>	Conductors displayed.
<b>C1</b>	1	First conductor to be displayed.
<b>C2</b>	*	Last conductor to be displayed. * means highest numbered conductor.
<b>XORIGIN</b>	0	X-coordinate at centre of display.
<b>YORIGIN</b>	0	Y-coordinate at centre of display.
<b>ZORIGIN</b>	0	Z-coordinate at centre of display.
<b>HIDDEN</b>	<b>NO</b>	Hidden surfaces removed switch.
	<b>FULL</b>	Colour-fill display of visible surfaces (slower but more reliable algorithm).
	<b>NO</b>	Wire frame display of all surfaces.
	<b>YES</b>	Colour-fill display of visible surfaces

Command	DISPLAY	
Parameter	Default	Function
VECTORS	CONDUCTORS	Vector display switch.
		CONDUCTORS Vectors show current direction on conductors.
		CURRENT Vectors show current density direction.
		MATERIAL Vectors show material orientation.
		NO Vectors not displayed.
		VELOCITY Vectors show velocity direction.
ERASE	YES	Picture erase switch.
		NO Old picture not erased.
		YES Old picture erased.
THREED	NO	Copy view from 3D Viewer.
		NO Use DISPLAY command parameters.
		YES Set DISPLAY command parameters to match 3D Viewer.
AXES	YES	Draw axes switch:
		NO No axes drawn.
		YES Display coordinate axes.

The **DISPLAY** command draws pictures of the three dimensional geometry of the finite element mesh and conductors. Pictures can be line-drawings or coloured surfaces with hidden surfaces obscured. The discretization can be included and vectors can be drawn to indicate the vector properties of the volumes and the current directions in the conductors. Colours are used to differentiate between volumes with different material names or to show values of assigned potentials. The parameters control the coordinate limits of the volume of three dimensional space included, the direction of the view, the parts of the model included and other options.

Pictures can also be displayed using [The THREED Command \[page 503\]](#).

## The Coordinate Limits and the Viewing Direction

### Menu route

**DISPLAY -> Display Command ... view**

**DISPLAY -> Display Command ... copy 3d view**

The **DISPLAY** command draws a picture including parts of the model which are inside a volume which is a cube of dimension **2\*SIZE**. The centre of the cube is at coordinates (**XORIGIN**, **YORIGIN**, **ZORIGIN**).

The view obtained is controlled by the eye position (**XEYE**, **YEYE**, **ZEYE**), the origin (**XORIGIN**, **YORIGIN**, **ZORIGIN**) and whether perspective is selected (+**PERSPECTIVE** or -**PERSPECTIVE**).

For non-perspective (orthogonal) views the view direction is set by **XEYE**, **YEYE** and **ZEYE**. For perspective views the view direction is set by the eye position and the origin. The distance between the origin and the eye position is important; parts of the model behind the eye will be omitted from the display.

The viewing direction can also be altered by the rotation angle parameters (**ROTX** and **ROTY**), which have the effect of altering the values of **XEYE**, **YEYE** and **ZEYE**, the eye position. The values of **ROTX** and **ROTY** are not remembered. **ROTATE** specifies a rotation angle about the viewing direction.

### Copying 3D View

The above parameters (**SIZE**, **XEYE**, **YEYE**, **ZEYE**, **PERSPECTIVE**, **ROTX**, **ROTY**, **ROTATE**, **XORIGIN**, **YORIGIN** and **ZORIGIN**) can also be set so that the view of the model corresponds to that of the 3D Viewer (+**THREED**). The **THREED** parameter is set back to **NO** after each time it is used so that these parameters can be adjusted for the next **DISPLAY**.

The view parameters are only effective if the previous picture is erased with +**ERASE**. If the previous picture is not erased (-**ERASE**) then the view remains as before.

The coordinate axes can be optionally displayed to show the scale of the model (+**AXES**)

## Selecting Parts of the Finite Element Model

### Menu route

**DISPLAY -> Display Command ... select parts**

The **DISPLAY** command draws all entities, volumes, facets, lines and points, created with [The DEFINE Command \[page 393\]](#) and [The EXTEND Command \[page 448\]](#). By use of the **TYPE** and **LABEL** parameters, sub-sets of the entities can be selected in order to aid visualization of the model. The parameter **TYPE** can be set to any individual entity type or any combination of entity types separated by + or -. For example, to display lines and points, **TYPE=LINE+POINT** should be used; for everything except points, **TYPE=ALL-POINT**. Similarly, **LABEL** can be set to individual labels or

combinations of labels. The additional label, **NOTAIR** can be used to select all material name labels except **AIR**. Abbreviated label names can be used where the abbreviation is not ambiguous. Abbreviations can be followed by \* to indicate that all labels which match should be selected. An entity is drawn if it has any of the labels selected. For example, to display all facets with vector potential boundary conditions, **LABEL=A\*-ALL** could be used.

Labels are assigned automatically to parts of the model by the commands **DEFINE**, **EXTEND**, **MODIFY** and **SLIP**. Automatically assigned labels include **ALL**, material names, element types potential types and boundary condition types and functions. Additional labels can be given in two ways: **The LABEL Command [page 458]** can be used to give any label to any part of the model and **The CHECK Command [page 378]** can be used to assign the labels **DEBUG** to volumes with bad shapes and **EXTERNAL** to facets which are not shared by two volumes.

At the end of the **DISPLAY** command, **LABEL** is set to **SAME** indicating that the same labels will be used for the next picture, unless **LABEL** is reset. **SAME** can be used in further combinations of labels to add or remove labels from the previously selected list.

The part of the finite element mesh displayed can also be restricted by the parameters **MESH**, **L1** and **L2**. The **MESH** parameter is used to select **ALL** meshes or one particular mesh (each **DEFINE** command creates a mesh with a new number). **MESH=NONE** can be used to omit the finite element mesh completely. **L1** and **L2** are used to select a subset of the layers of the mesh. **L2** can be set to \* to indicate the top layer.

## Selecting Conductors

### Menu route

**DISPLAY -> Display command ... select parts -> Conductors**  
**DISPLAY -> Display command ... select parts -> Conductor numbers**

The **DISPLAY** command draws the conductors including any symmetry copies. This can be controlled by the parameters **COIL**, **C1** and **C2**. **+COIL** and **-COIL** switch the display of the conductors on and off, and **C1** and **C2** select a range of conductors for display. **C2** can be set to \* to indicate the highest numbered conductor.

## Other parameters

### Menu route

**DISPLAY -> Display command ... style**

Two types of picture can be produced: wire-frame line drawings or coloured surface pictures, with hidden surfaces obscured. **-HIDDEN** produces a wire-frame picture; **+HIDDEN** uses a fast but unsophisticated algorithm to produce the "hidden surface" pictures. This does not always achieve a

perfect picture. It is usually possible to obtain a satisfactory picture with a suitable choice of view point or **HIDD=FULL** which orders the displayed facets by a slower but more reliable algorithm could be used. With **TYPE=VOLUME +HIDDEN**, facets which are shared by volumes with the same labels are omitted.

Hidden surface pictures can only be displayed after the surface mesh has been calculated (see [The MESH Command \[page 467\]](#)).

The discretization of the surfaces and volumes can also be displayed using the **ELEMENT** parameter. If the volume or surface meshes have not been calculated, the program displays using the nearest options it can.

Hidden surface views of **FACTS**, with **LABEL** set to a single potential name (**AX, AY, AX, POTENTIAL, VOLTAGE** or the incident potentials) are displayed as contour plots to show the values of the potential on the facets.

Arrows to show the direction of the volume vector properties can be drawn inside each volume. By default, the arrows are only drawn inside the conductors to show the current direction (**VECTOR-R=CONDUCTORS**) but they can also show the **MATERIAL** orientation, the **CURRENT** density direction or the **VELOCITY** direction. The arrow display can be switched off completely (**VECTOR=NO**).

## Examples

The following example **DISPLAY** commands assume program default values, and that the commands are issued in sequence. The model is assumed to occupy a unit cube in the positive **X, Y** and **Z** octant of the coordinate system.

- Example: to display a wire-frame picture without conductors, orthogonal projection from the **Z** direction (note the use of positional parameters):

```
Opera > display 0.5 xorigin=0.5 0.5 0.5 -coil
```

- Example: to display the conductors, and check that the reduced potential volumes enclose them:

```
Opera > xeye=3 4 5 +coil label=reduced
```

- Example: to obtain a "hidden-surface" picture of the conductors and one material, checking the finite element discretization on the surfaces of the volumes, and using perspective (note that the eye position has to be moved further away to reduce the effect of the perspective):

```
Opera > xeye=9 12 15 +perspective label=iron,
```

```
Opera > +hidden, element=surface
```

- Example: to display the surfaces with the **NORMAL MAGNETIC** boundary condition:

```
Opera > type=face label=normmagn
```

## The **DUMP** Command

---

### Menu route

**OPTIONS -> Dump picture**  
**File ↓**  
**Print → PostScript to file**

### Command line parameters

Command	<b>DUMP</b>	
Parameter	Default	Function
<b>FILE</b>	none	Name of file to contain the picture.
<b>TYPE</b>	<b>POSTSCRIPT</b>	Graphics language:
		<b>HPGL</b> Hewlett-Packard Graphics Language
		<b>POSTSCRIPT</b> Adobe PostScript
<b>SIZE</b>	<b>A4</b>	Paper sizes: HPGL can use <b>A</b> , <b>A3</b> , <b>A4</b> , <b>B</b> , <b>A0D</b> , <b>A1D</b> , <b>A2D</b> , <b>A3D</b> or <b>A4D</b> ; PostScript can use <b>A4</b> , <b>A</b> or <b>USER</b> .
<b>LLX</b>	0	X-coordinate of lower-left corner in mm ( <b>TYPE=POSTSCRIPT</b> , <b>SIZE=USER</b> ).
<b>LLY</b>	0	Y-coordinate of lower-left corner in mm ( <b>TYPE=POSTSCRIPT</b> , <b>SIZE=USER</b> ).
<b>URX</b>	150	X-coordinate of upper-right corner in mm ( <b>TYPE=POSTSCRIPT</b> , <b>SIZE=USER</b> ).
<b>URY</b>	106	Y-coordinate of upper-right corner in mm ( <b>TYPE=POSTSCRIPT</b> , <b>SIZE=USER</b> ).
<b>COLOUR</b>	<b>YES</b>	Colour PostScript: <b>NO</b> implies grey-scale.
<b>FILL</b>	<b>NO</b>	Filled polygons in HPGL: <b>YES</b> or <b>NO</b> .
<b>ORIENT</b>	<b>LANDSCAPE</b>	PostScript paper orientation: <b>LANDSCAPE</b> or <b>PORTRAIT</b> .

Command	DUMP	
Parameter	Default	Function
SWAP	YES	Swap black and white in PostScript:
		NO Colours appear as on screen.
		YES Black and white are swapped.

## Notes

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. Note that the 3D Viewer window cannot be copied to a file using the **DUMP** command. This window must be saved as a bitmap directly.

- Adobe PostScript (**TYPE=POSTSCRIPT**): 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+YES** 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=R=NO**. 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 ( $\pm$  **FILL**). If selected, polygons are filled using shading with parallel horizontal or vertical lines.

## The **EDIT** Command

---

**Not available from Menus**

### Command line parameters

Command	<b>EDIT</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of Opera-3d Pre-Processor data file.

The **EDIT** command reads an Opera-3d Pre-Processor data file into the program in edit mode. This allows major changes to be made to the data, including changes to the topology of the base plane. The data already stored in the Pre-Processor can also be changed using [The EXTEND Command \[page 448\]](#), [The MODIFY Command \[page 469\]](#) and [The REDEFINE Command \[page 483\]](#).

The **EDIT** command has one parameter which defines the name of the **FILE**. If no file name extension is given, the extension *oppre* is assumed.

As the file is being read a certain amount of editing can be done. Each top-level command can be **eXECUTED**, **IGNORED** or replaced; and at breakpoints within the **DEFINE** command the user can add additional information or change the data which has been read. At any point it is possible to **FINISH** editing and read the rest of the file or **SKIP** over the rest of the file and return immediately to normal control.

All the commands which the **EDIT** or executes are decoded, so that the parameter values are stored but only certain commands are obeyed. These are the commands which define finite element data (**DEFINE**, **EXTEND** and **MODIFY**), conductor data (**CONDUCTOR**) and built-in commands (**\$ OS** and **\$ CD** are not executed).

More details of reading Opera-3d Pre-Processor data files are given with the **READ** command. When the editing is complete, the **WRITE** command can be used to store the edited data in a file.

The **EDIT** command should not be used in a **\$ COMINPUT** file.

### Editing Top-level Commands

As the program reads each top-level command from the file, the command is displayed on the terminal with the message

Your next input was:

and the user has the opportunity to **eXECUTE**, **IGNORE** or replace it, to **SKIP** over all of the remaining commands in the file or to **FINISH** reading and executing until the bottom of the file is reached.

Replacement commands should not introduce extra interactions with the program, since it is not possible to insert new command lines.

Top-level Edits	
Keyword	Meaning
<b>FINISH</b>	Continue reading to end of file.
<b>IGNORE</b>	Ignore the displayed command.
<b>SKIP</b>	Ignore the rest of the file.
<b>XECUTE</b>	Execute the displayed command.
<i>replacement command</i>	Execute the replacement command.

## Editing the **DEFINE** Command

As the sub-commands and cursor-hits within the **DEFINE** command are being read, some of the keyboard command lines can be edited in the same way as top-level commands above. It is not usually sensible to use the **IGNORE** option except to remove an invalid command line from the file otherwise the sequence of commands would be destroyed. Similarly, a replacement command should not invoke extra interaction with the program, since it is impossible to add extra command lines.

Edits to <b>DEFINE</b> sub-commands	
Keyword	Meaning
<b>FINISH</b>	Continue reading to end of file.
<b>IGNORE</b>	Ignore the displayed command.
<b>SKIP</b>	Ignore the rest of the file.
<b>XECUTE</b>	Execute the displayed command.
<i>replacement command</i>	Execute the replacement command.

Additional break points are included at the ends of each of the modes of the **DEFINE** command. At these break points a question requiring the answer **YES** or **NO** is given enabling the user to define more points or facets, redefine facet subdivisions, move points on subsequent planes or redefine materials and boundary conditions.

Note that there is only a break point at the ends of Point Definition Mode and Facet Definition Mode when those modes are left for the first time. There is no break point when Point Definition Mode is re-entered from Facet Definition Mode or when Facet Definition Mode is re-entered from Subdivision Mode. However it is possible to re-enter Point Definition Mode from the break point at the end of Facet Definition Mode and to re-enter Facet Definition Mode from the break point at the end of Subdivision Mode.

- Example - at the end of Base Plane Facet Definition Mode, the message given is:

End of facet definition mode. Do you need to define more facets? (Y or N)

In response to a **YES** reply, the program enters the appropriate **DEFINE** command mode (see [The DEFINE Command \[page 393\]](#)). It is usually necessary to use the **R** cursor hit first so that the current state of the data can be displayed. When a **Q** cursor hit is used to leave the mode, the program resumes reading the file, until the next break point is reached. In the case of the above example, a **YES** response makes the Facet Definition Mode cursor commands available, including **N** for returning to [Point Definition Mode \[page 399\]](#).

**FINISH** and **SKIP** can be used as replies instead of **YES** or **NO** to end the editing by reading and executing the rest of the file or returning immediately to normal input.

Edits at <b>DEFINE</b> break-points	
Keyword	Meaning
<b>FINISH</b>	Continue reading to end of file.
<b>NO</b>	Continue reading to the next break point.
<b>SKIP</b>	Ignore the rest of the file.
<b>YES</b>	Make cursor input available to edit the data.

## The END Command

---

### Menu route

FILE -> End Opera-3d/Pre

### Command line parameters

Command	END
No Parameters	

The END command stops the Opera-3d Pre-Processor. All data files are closed.

Note that it is important to create a Pre-Processor data file using [The WRITE Command \[page 510\]](#) before ending the program so that all the commands and data are recorded.

If the program is ended without a WRITE command having been issued to store the Pre-Processor data, then the file *Opera3d\_Pre\_n.backup* can be renamed to have a file name extension *oppre*. This file is equivalent to a Pre-Processor data file.

## The EXTEND Command

---

### Menu route

**DEFINE -> Extend existing mesh**

### Command line parameters

Command	EXTEND	
Parameter	Default	Function
<b>MESH</b>	1	Number of finite element mesh to be extended.
<b>EDIT</b>	<b>YES</b>	Material and boundary condition editing switch:
		<b>NO</b> Materials left as <b>AIR TOTAL LINEAR</b> , no boundary conditions set.
	<b>YES</b>	Material properties and boundary conditions can be set after extensions.

The **EXTEND** command puts the Pre-Processor into its finite element mesh creation mode but starts from the top plane of an existing mesh. The user input is tightly structured by the program. The top mesh plane surface is extruded or swept through space thus creating new layers of volumes. The topology is maintained during the extrusion operations, but the point coordinates can be changed in the new surface created by each extrusion or sweep. If editing has been selected (+**EDIT**), once all the new layers have been created, the volumes in the new layers of the mesh can be assigned material attributes and boundary conditions can be assigned to the faces of the volumes.

*In menu mode* the **Extend with editing** option with editing adds one additional extrusion at a time and allows coordinates, material properties and boundary conditions to be edited. **Extend without editing** can be used to add several extrusions, but any variation in coordinate positions, material properties and boundary conditions must be applied later using **MODIFY**.

The **MESH** parameter specifies the number of the finite element mesh to be extended.

### Extrusions Mode

The **EXTEND** command applies extrusion operations to the set of facets of the top mesh plane of previously created mesh. There must be at least one extrusion operation, but many others may be needed to define the complete model. It is also possible to add more extrusions to a completed mesh using the **EXTEND** command again. Point movements and transformations can be performed in the same way as for **DEFINE**. The details of the command lines to define the extrusions are given with the **DEFINE** command [Extrusions Mode \[page 413\]](#).

After the extrusion has been created and all necessary points have been moved, the program moves on to the next extrusion. The user is given the option of creating another extrusion or not (valid replies are **YES** or **NO**). After a **NO** response, the program moves on to Material Definition Mode for the new layers, if **EDITing** is selected.

## Material Definition Mode

It is now necessary to set the material and mesh properties within each new volume. This is done by presenting the user with each layer in turn and the user setting values which over-ride the default or current settings within each volume. Only the new layers are presented to the user, but lower numbered layers can be modified by using the **FROM** keyword on the material definition command line.

Volumes are selected and materials and properties defined in the same ways as for the **DEFINE** command [Material Definition Mode \[page 420\]](#). When all necessary changes have been made the program moves on to Boundary Condition Definition Mode for the new facets, starting with the mesh plane which was previously the top plane.

## Boundary Condition Definition Mode

Boundary conditions can be set on any surface of any of the new volumes in the mesh. In order to achieve this the program presents the facets in 3 separate groups: first the facets on the planes from the previous top plane to the one below the new top plane, second the final plane of the mesh, and last the facets normal to the planes, for each of the new layers, one layer at a time. Facets of lower numbered planes or layers can be modified by using the **FROM** keyword on the boundary condition definition command line.

Facets are selected and boundary conditions defined in the same ways as for the **DEFINE** command [Boundary Condition Definition Mode \[page 426\]](#). When all necessary changes have been made the program ends the **EXTEND** command and waits for another top-level command.

## The **FILL** Command

---

### Menu routes

**MESH -> Volume mesh ... options**  
**MESH -> Mesh**

### Command line parameters

Command	<b>FILL</b>	
Parameter	Default	Function
<b>TOLERANCE</b>	1.E-5	Geometric Tolerance
<b>PRINT</b>	0	Diagnostic printing level (0 or 1)

The **FILL** command generates the volume finite element mesh in all the Pre-Processor volumes. The type of mesh (tetrahedra or hexahedra) is determined by which type of surface mesh has been created by [The MESH Command \[page 467\]](#).

### Tetrahedral meshes

Each region of the model is meshed independently, given the required triangular element mesh on the surfaces that has been created by the **MESH** command. If the volume has only 3 or 4 sided facets with regular subdivisions, the internal nodes are regularly positioned. Otherwise, the meshing is based on Delaunay point insertion, followed by maximizing the minimum angle of the elements.

The **TOLERANCE** parameter may need to be adjusted if the software fails to create a mesh successfully. Problems can occur if the element size varies too much between the surfaces of a region; the tolerance should be increased to overcome this. In other cases, for example, a complicated region with a wide range of feature sizes, the tolerance may need to be reduced.

If the mesh generation process fails, and the **TOLERANCE** parameter does not correct the problem, then it is likely that the element size is changing too much between adjoining surfaces. Regions that cannot be meshed are flagged using the **DEBUG** label and can be selected for **DISPLAY**. Adjust the edge subdivisions to reduce the element size variations in such regions.

### Hexahedral meshes

Hexahedral elements can only be generated if the model uses regions that are hexahedra or degenerate hexahedra. A regular finite element mesh is generated in all regions, by subdivision of the region to similar shaped elements. The **TOLERANCE** parameter is used to test for coincident points in degenerate shapes.

## The **HELP** Command

---

### Menu route

**HELP -> Help**

### Command line parameters

Command	<b>HELP</b>
No Parameters	

The **HELP** command gives help to remind users of several of the features of the program. The topics are:

- **System Overview:** this gives a flow-chart of the Opera-3d Pre-Processor top level commands indicating the sequence in which they should be used to prepare a data set for analysis.
- **Command Interpreter:** this summarizes the syntax and built-in help features of the command decoder, including details of sub-commands and cursor commands. More information is given in chapter "["Command Language" on page 23](#)
- **Euler Angles:** the definitions of the Euler Angles used within Opera-3d is given, and information on how to use the escape function **\$EULER** to specify them. For more information see [Euler Angles \[page 83\]](#).
- **New Features:** this summarizes the features of the program which have been added since the previous version.
- **Units:** this lists the units used in the systems allowed by the analysis programs.

The program prompts for the name of a topic.

Keyword	Meaning
<b>COMMAND</b>	Command interpreter.
<b>EULER</b>	Euler angles.
<b>NEW</b>	New features.
<b>SYSTEM</b>	System overview.
<b>UNITS</b>	Unit systems.
<b>QUIT</b>	Leave the <b>HELP</b> command.

When accessed from the menus, the **HELP** command has different topics explaining the use of the program and the menu interface.

## The IDEAS Command

---

### Menu route

**FILE -> Read I-DEAS universal file**

### Command line parameters

Command	<b>IDEAS</b>
No Parameters	

The **IDEAS** command introduces a set of sub-commands which can be used to read a finite element mesh from an I-DEAS Universal File and associate Opera-3d material properties and boundary conditions with the elements and nodes it contains.

The **READ** sub-command is used to read the data. Additional information about materials and boundary conditions can be supplied using the sub-commands **BOUNDARY** and **MATERIAL** after the file has been read. Conductors can be added and the data can be **DISPLAYed** (see [The DISPLAY Command \[page 436\]](#)) and written to an Opera-3d analysis database (see [The SOLVERS Command \[page 485\]](#)), but the finite element data cannot be modified.

### GUI mode

When the **IDEAS** command is used from the GUI, the program issues the **READ** sub-command and then creates a sequence of dialogs which leads the user through the **MATERIAL** and **BOUNDARY** sub-commands needed to assign Opera-3d properties to the geometric data.

## Universal Data File for Input to Opera-3d

The Universal Data File created by I-DEAS must contain the following Datasets (Level 6 or Master Series):

Dataset Name	Dataset Number
Header	151
Nodes	15, 781 or 2411
Material Properties	91, 747, 773, 1710 or 1714
Elements	71, 780 or 2412
Restraints or Loads	755, 756, 782, 791 or 792

The order of the datasets is not important except that any data which is referenced by other datasets must be defined before it is referenced; e.g. nodes must be defined before elements. However, if an element refers to an undefined material number, the program will invent a material name. Other datasets may also be included in the input file, but will be ignored.

The I-DEAS Universal Data File Level 6 and Master Series formats are defined in the I-DEAS documentation published by Siemens PLM, and are not repeated here. The following information describes how the data is used in interface to Opera-3d.

## Material Properties

Up to 100 materials may be used in the universal file. The properties in the universal file are ignored; only the material number and name are used by the interface. A different material should be used for each different combination of Opera-3d material and volume properties. For example, in the universal file data, the volume representing free space might be divided into two parts with different material names, so that in Opera-3d part can use **REDUCED** potential and the other part **TOTAL** potential. Each material defined in the universal file will initially be given the properties of **AIR** with **TOTAL** potential and **LINEAR** elements. The Opera-3d material name, potential type and element type can be changed and other volume properties added using [The IDEAS Sub-command MATERIAL \[page 456\]](#).

Material names are invented for any material number which is not defined before it is referenced by an element.

## Element Topologies

The interface is able to process elements of the following types:

111	Linear tetrahedron
115	Linear hexahedron
116	Quadratic hexahedron
118 or 119	Quadratic tetrahedron

Mixed element types are allowed, but in this case tetrahedra will be represented by degenerate hexahedra.

All quadratic elements in the universal file are converted to linear elements by the interface. Quadratic elements can be selected either by specifying that a particular universal file material should use quadratic elements (see [The IDEAS Sub-command MATERIAL \[page 456\]](#)) or that all elements should be quadratic when the analysis database is created (see [The SOLVERS Command \[page 485\]](#)).

## Restraints and Loads

Restraint and Load datasets identify the sets of nodes which should have the same boundary condition. The types of restraint must be nodal temperature but the values of temperature in the

universal file are ignored. Only the node numbers are used. The **BOUNDARY** sub-command can be used to associate one or more Opera-3d boundary conditions with element faces which have all their nodes in the a particular restraint set.

## Displaying Universal File Data

Universal File data can be displayed using [The DISPLAY Command \[page 436\]](#). Line drawings are not available, but displays without hidden surfaces can show surface or volume elements.

As the universal file data is being read, the elements are allocated to volumes, one for each universal file material. A label **MATE***n* (where *n* is the universal file material number) is added to each volume so that they can be selected for display. After Opera-3d properties have been added, the material names and other property labels can be used as well.

Similarly, the element facets identified by the nodes in each boundary condition set are grouped and given facet labels **BCSET***n* (where *n* is a counter of how many boundary condition sets have been read). The boundary condition facets can be displayed using these labels. After the Opera-3d boundary conditions have been applied using the **BOUNDARY**, the Opera-3d boundary condition names can be used as well.

## The IDEAS Sub-command **BOUNDARY**

### Menu route

**FILE -> I-DEAS universal file -> Define Boundary Conditions**

### Command line parameters

Sub-command	<b>BOUNDARY</b>	
Parameter	Default	Function
<b>NUMBER</b>	<i>none</i>	Boundary condition set number
<b>OPTION</b>	<b>ENQUIRE</b>	Option:
	<b>ENQUIRE</b>	List boundary conditions defined for this set.
	<b>SET</b>	Define or clear boundary conditions for this set.

Sub-command	<b>BOUNDARY</b>	
Parameter	Default	Function
<b>CONDITION</b>	<i>none</i>	Boundary condition name or <b>NONE</b> : AX      AX boundary condition. AY      AY boundary condition. AZ      AZ boundary condition. DAX     DAX boundary condition. DAY     DAY boundary condition. DAZ     DAZ boundary condition. DERIVATIVE    DERIVATIVE boundary condition. INAX    INAX boundary condition. INAY    INAY boundary condition. INAZ    INAZ boundary condition. NONE    Clear all boundary conditions. NORMELEC    NORMAL ELECTRIC boundary condition. NORMMAGN    NORMAL MAGNETIC boundary condition. PEC      PEC boundary condition. POTENTIAL    POTENTIAL boundary condition. RADIATION    RADIATION boundary condition. SLIP     SLIP boundary condition. SYMMETRY    SYMMETRY boundary condition. TANGELEC    TANGENTIAL ELECTRIC boundary condition. TANGMAGN    TANGENTIAL MAGNETIC boundary condition. VOLTAGE    VOLTAGE boundary condition.
<b>VALUE</b>	<i>none</i>	Potential value.
<b>LABEL</b>	<i>none</i>	Drive label.

The **BOUNDARY** sub-command can be used to list existing boundary conditions on the set of nodes given by **NUMBER**, (**OPTION=ENQUIRE**) or add new ones (**OPTION=SET**, **CONDITION=name**). **OPTION=SET**, **CONDITION=NONE** clears all existing conditions on the set of nodes. The Opera-3d boundary conditions are described in the section [Boundary Condition Definition Mode \[page 426\]](#).

## The IDEAS Sub-command MATERIAL

### Menu route

**FILE -> I-DEAS universal file -> Define Materials**

### Command line parameters

Sub-command	<b>MATERIAL</b>	
Parameter	Default	Function
<b>NUMBER</b>	<i>none</i>	Universal file material number.
<b>OPTION</b>	<b>ENQUIRE</b>	Option:  ENQUIRE List material properties. SET Define material properties.
<b>NAME</b>	<i>none</i>	Opera-3d material name.
<b>POTENTIAL</b>	<i>none</i>	Potential type:  REDUCED Reduced potential. TOTAL Total potential. VECTOR Magnetic vector potential.
<b>ELEMENT</b>	<i>none</i>	Element type:  LINEAR First order elements. QUADRATIC Second order elements.
<b>PACK</b>	<i>none</i>	Packing factor.
<b>SCALAR</b>	<i>none</i>	Scalar property: charge density or rotational velocity.
<b>CURX</b>	<i>none</i>	X-component of current density.
<b>CURY</b>	<i>none</i>	Y-component of current density.
<b>CURZ</b>	<i>none</i>	Z-component of current density.
<b>VELX</b>	<i>none</i>	X-component of velocity.
<b>VELY</b>	<i>none</i>	Y-component of velocity.
<b>VELZ</b>	<i>none</i>	Z-component of velocity.
<b>THETA</b>	<i>none</i>	Material orientation Euler Angle $\theta$ .
<b>PHI</b>	<i>none</i>	Material orientation Euler Angle $\phi$ .
<b>PSI</b>	<i>none</i>	Material orientation Euler Angle $\psi$ .

The **MATERIAL** sub-command can be used to list existing material properties (**OPTION=ENQUIRE**) or define new ones (**OPTION=SET**). **OPTION=ENQUIRE** also sets the default values of the other parameters to the current values for the material.

The current density, velocity and material orientation vectors can be expressed as functions of **X**, **Y** and **Z**.

The Opera-3d material properties are described in the section [Material Definition Mode \[page 420\]](#).

## The IDEAS Sub-command QUIT

### Menu route

**FILE -> I-DEAS universal file -> Return**

### Command line parameters

Sub-command	<b>QUIT</b>
No Parameters	

The **QUIT** sub-command leaves the **IDEAS** command and returns to the top-level commands.

## The IDEAS Sub-command READ

### Menu route

**FILE -> I-DEAS universal file -> Read File**

### Command line parameters

Sub-command	<b>READ</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of file.

The **READ** sub-command reads a Universal File to input materials, nodes, elements and boundary conditions. If no file name extension is given, the extension *unv* is assumed.

## The **LABEL** Command

---

### Menu route

**MODIFY -> Labels**

### Command line parameters

Command	<b>LABEL</b>	
Parameter	Default	Function
<b>MESH</b>	1	Number of finite element mesh to be labelled.
<b>START</b>	1	Number of the first plane or layer. Planes are numbered: 1,2, 3, etc.; mid-extrusion planes and layers are numbered: 1.5, 2.5, 3.5, etc.

The **LABEL** command allows labels to be added or removed from the entities (points, lines, facets and volumes) of the finite element mesh. Some labels are added automatically by [The DEFINE Command \[page 393\]](#): every entity has label **ALL**, facets have the boundary condition types, and volumes have the material names, potential types and element types. Such labels can only be changed by the **MODIFY** command. Other labels are added to parts of the model by [The CHECK Command \[page 378\]](#), [The FILL Command \[page 450\]](#), [The MESH Command \[page 467\]](#) and [The SLIP Command \[page 484\]](#). Additional labelling by the **LABEL** command allows entities to be grouped for [The DISPLAY Command \[page 436\]](#) and points to be grouped for [The TRANSFORM Command \[page 509\]](#). To enable multiple points to be labelled more easily, any labelling operation on volumes, facets or lines is also applied to the points which define them.

Facet and volume labels are stored in analysis databases and can be used in the Post-Processor.

The data of the finite element mesh is separated into 4 sections, corresponding to the modes of the **DEFINE** command. The user is asked first to choose between points, lines, facets and volumes.

<b>LABEL Modes</b>	
Keyword	Meaning
<b>FACETS</b>	Add or remove labels on facets.
<b>LINES</b>	Add or remove labels on lines.
<b>POINTS</b>	Add or remove labels on points.
<b>QUIT</b>	Leave the <b>LABEL</b> command.
<b>VOLUMES</b>	Add or remove labels on volumes.

Valid replies are: **FACETS**, **LINES**, **POINTS**, **QUIT** or **VOLUMES**.

*In menu mode* labels can be set on one plane or layer at a time. The program prompts for a value of **Plane number** or **Layer number**. The menu item **Select/de-select entity** allows entities to be added or removed from a list. Menu item **Select and define** adds one last entity to the list and causes the program to display a DialogBox into which the labels can be entered. The menu item **List conditions** allows the labels to be listed for an individual entity.

*In keyboard mode*, each mesh plane or extrusion layer of the mesh specified by the **MESH** parameter, starting with the plane or layer given by the **START** parameter, is presented to the user. Mesh planes are numbered 1, 2, 3, etc., and mid-extrusion planes and layers 1.5, 2.5, 3.5, etc. Layer 1.5 is between plane 1 and plane 2. The facets are presented in two sections: first those lying in the mesh planes, then those connecting the extruded planes. Entities can be selected using the cursor, singly with a <space>, or in groups with **K** for all but the last and <space> for the last.

A new label can be **ADDED** to or an existing label **MOVED** from the selected entities and corresponding entities in other planes or layers. For each set of entities the selection cursor hits and labelling sub-commands are the same.

## Entity Selection

Entity Selection Menu and Cursor Hits		
Cursor hit	Menu item	Function
<space>	Select and define	Select entity nearest cursor and prompt for a labelling command.
<b>F</b>	Finish	Finish labelling this type of entity.
<b>H</b>		Display menu help message explaining all the cursor options. (More help is available after entity selection.)
<b>K</b>	Select/de-select entity	Select the entity nearest the cursor. Repeating <b>K</b> for a selected entity, de-selects it. A labelling command for all the selected entities can be given after a <space> cursor hit.
<b>L</b>	List labels	List all the labels on the nearest entity.
<b>Q</b>		Leave this plane or layer.

Entity Selection Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are:
	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.
	<b>CURSOR</b>	Select diagonally opposite corners of the display area with <space> cursor hits.
	<b>RESTORE</b>	Return to previous screen size.
	<b>BOUND</b>	Use bounding rectangle of geometry.
	<return>	Reconstruct at the same size.
<b>v</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
<b>z</b>	Aspect-ratio search	Switch <a href="#">Aspect Ratio Searching [page 396]</a> on or off.

## Labelling Sub-commands in Keyboard Mode

Following a <space> cursor hit, one of the following sub-commands should be given. The commands apply to the entity selected with the <space> and to any previously selected with **R**.

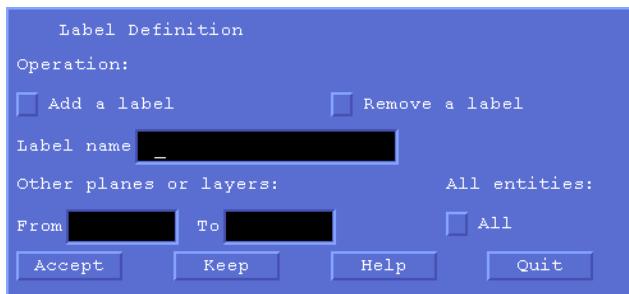
The labelling sub-commands consist of the following items:

Labelling Sub-commands	
Command and parameter:	
<b>ADD label</b>	Add a label to the selected entities.
<b>HELP</b>	Obtain help on all sub-commands and options.
<b>QUIT</b>	Abandon the currently selected entities.
<b>REMOVE label</b>	Remove a label to the selected entities.
Optional keywords - setting multiple entities:	
<b>ALL</b>	Sets all entities in layer(s) or plane(s).

Labelling Sub-commands	
<b>FROM value</b>	Sets all layers/planes from layer/plane number given through to the current layer/plane or the layer/plane number given with <b>TO</b> .
<b>KEEP</b>	Keeps current selection of entities for another <b>ADD</b> or <b>REMOVE</b> command.
<b>TO value</b>	Sets all layers/planes from current layer/plane or the layer/plane number given with <b>FROM</b> through to the layer/plane number given. The value may be * to indicate the top layer/plane.

## Labelling DialogBox in Menu Mode

In the DialogBox, **ADD** or **REMOVE** must be selected and a label name given.



## The MATERIALS Command

---

### Menu route

**DEFINE -> Material Properties**

### Command line parameters

Command	MATERIALS		
Parameter	Default	Function	
<b>OPTION</b>		<b>PICK</b>	Adds a material label to a list to be set.
		<b>UNPICK</b>	Clears the list of material labels to be set.
		<b>RESET</b>	Sets picked materials to have the properties of air.
		<b>MODIFY</b>	Sets the data for the picked materials.
		<b>METRE</b>	Work in SI units.
		<b>CGS</b>	Work in CGS units.
		<b>INCH</b>	Work in SI units with inches.
		<b>MM</b>	Work in SI units with MM.
		<b>MICRON</b>	Work in SI units with microns.
		<b>LIST</b>	Lists the material properties of the picked materials.
		<b>DELETE</b>	Deletes the picked materials.
<b>MATERIALLABEL</b>		Material label used with <b>OPTION=PICK</b>	
<b>ANISOTROPY</b>		Sets the anisotropy of all permeability, permittivity and conductivity	
		<b>ISOTROPIC</b>	Set to isotropic.
		<b>PACKED</b>	Set to packed.
		<b>MULTIPLE</b>	Set to anisotropic.

Command	MATERIALS	
Parameter	Default	Function
LINEARITY		Sets the properties to be linear or nonlinear.
		LINEAR Set to linear.
		NONLINEAR Set to nonlinear.
MUANISOTROPY		Sets the anisotropy of permeability.
		ISOTROPIC Set to isotropic.
		PACKED Set to packed.
		MULTIPLE Set to anisotropic.
MULINEARITY		LINEAR Set to linear.
		NONLINEAR Set to nonlinear.
MU		Isotropic linear permeability.
HC		Isotropic coercivity.
BH		Isotropic nonlinear BH curve.
MPHASE		Phase lag for isotropic permeability.
MUXX		Anisotropic components of linear permeability.
MUYY		
MUZZ		
HCX		
HCY		Anisotropic components of linear coercivity.
HCZ		
BHX		
BHY		File names for anisotropic BH characteristics.
BHZ		
MAPHASE		Complex phase lag for anisotropic permeability.
SIGANISOTROPY		Sets the anisotropy of conductivity.
		ISOTROPIC Set to isotropic.
		MULTIPLE Set to anisotropic.
SIGMA		Isotropic conductivity.
SPHASE		Phase lag for isotropic conductivity.

Command	MATERIALS	
Parameter	Default	Function
SIGXX		Anisotropic components of conductivity.
SIGYY		
SIGZZ		
SAPHASE		Complex phase lag for anisotropic permittivity.
EPSANISOTROPY		Sets the anisotropy of permittivity.
		ISOTROPIC Set to isotropic.
		MULTIPLE Set to anisotropic.
EPSILON		Phase lag for isotropic permittivity.
EPHASE		Isotropic permittivity.
EPSXX		Anisotropic components of permittivity.
EPSYY		
EPSZZ		
EAPHASE		Complex phase lag for anisotropic permittivity.
KAPANISOTROPY		Sets the anisotropy of thermal conductivity
		ISOTROPIC Set to isotropic
		MULTIPLE Set to anisotropic
KAPPA		Isotropic thermal conductivity
KAPXX		Anisotropic components of thermal conductivity
KAPYY		
KAPZZ		
HCAP		Heat capacity
MDEN		Mass density

This command defines the material characteristics for use by the analysis programs.

A set of material labels is picked using the command repeatedly, with **OPTION=PICK** and a **MATERIALLABEL** specified. A material label can be removed from the set using **OPTION=UNPICK**. If no **MATERIALLABEL** is given, the set is emptied.

Issuing the command with **OPTION=MODIFY** will modify the properties of the picked material labels to the new values given in the parameters. The value of properties associated with the material labels are unchanged if the parameter is unset.

The default value of each of the parameters is updated to be the common value of all the picked material labels. If the data of one of these parameters is unset, or the picked material labels do not share the same value, then the parameter value is left clear.

New material labels are assumed to be linear and isotropic. **OPTION=RESET** will clear the properties associated with all of the picked material labels.

The working material unit set can be changed using **OPTION=METRE**, **OPTION=CGS** or one of the other sets. If the units are changed, the values in the parameters are converted to the new unit set, and values entered will be interpreted in this unit set (note however that in **CGS** units, the coercive force  $\mathbf{H}_c$  is in units Oersted, and for all other options,  $\mathbf{H}_c$  is in units A/m).

'The properties of all material labels can be listed using **OPTION=LIST**.

Material labels that are not used, i.e. have no cell referencing them can be deleted using **OPTION-N=DELETE**. Deleting a material label that is in use will reset its properties to those of air.

All parameters can be specified. Which of these values will be used by the analysis will depend upon the analysis module being used, e.g. the phase lag properties are not used by **TOSCA**.

This command defines the **MATERIAL** properties for use by the analysis programs. All parameters can be specified for any model. Each analysis program will use the appropriate subset of properties which it needs. The values can be adjusted during creation of the database using the **SOLVERS MATERIAL sub-command [page 495]** (this is not available for thermal material properties).

All materials start with properties of air.

- **option=pick**: A set of material labels is selected using the command repeatedly with **option-n=pick** and a material label.
- **option=modify**: After a set of material labels has been picked, the default values of each parameter will be set to the data that is common to the picked material labels. This data can be modified using **option=modify**, with the new values supplied for each of the parameters. If a value is not supplied for a parameter value, that data stored with each material label is unaffected. There are two sets of parameters, one for **anisotropy=isotropic** or **packed** and the other for **anisotropy=multiple**. Only isotropic properties can be set using the GUI.
- **option=reset** will reset the values of all the data of all picked material labels to the properties of air.
- **option=METRE, option=CGS**: The values can be specified in CGS units or SI with various length units by using Material **option=CGS**, **option=METRE** etc. If the units are changed, the values in the parameters are converted to the new unit set, and values entered will be interpreted in this unit set.
- **option=list**: The properties of all materials can be listed using **option=list**.
- **option=delete**: Material labels that are not used, i.e. have no volume referencing them can be deleted using **option=delete**. Deleting a material that is in use will reset its properties to those of air.

All parameters can be specified. Which of these values will be used by the analysis will depend upon the analysis module being used, e.g. the phase lag properties are not used by **TOSCA**.

The definition of the thermal conductivity, **KAPPA** or **KAPXX**, **KAPYY** and **KAPZZ**, may be defined as functional properties of position (**X**, **Y**, **Z**) or as a nonlinear function of temperature, (**T**).

## The **MESH** Command

---

### Menu routes

**MESH -> Surface mesh ...**  
**MESH -> ... options**  
**MESH -> ... triangles**  
**MESH -> ... quadrilaterals**

### Command line parameters

Command	<b>MESH</b>					
Parameter	Default	Function				
<b>NORMAL</b>	0.1	Curved surface fitting tolerance.				
<b>PARAMETER</b>	<b>NO</b>	Triangulate in surface parameter space: <b>YES</b> or <b>NO</b> .				
<b>UPDATE</b>	<b>YES</b>	Check that the triangulation matches the Delaunay criterion after each point is added: <b>YES</b> or <b>NO</b> .				
<b>ELEMENT</b>	<b>QUADRILATERAL</b>	Surface element type: <table border="1"> <tr> <td><b>QUADRILATERAL</b></td> <td>4-sided surface facets leading to a hexahedral volume mesh.</td> </tr> <tr> <td><b>TRIANGLE</b></td> <td>3-sided surface facets leading to a tetrahedral volume mesh.</td> </tr> </table>	<b>QUADRILATERAL</b>	4-sided surface facets leading to a hexahedral volume mesh.	<b>TRIANGLE</b>	3-sided surface facets leading to a tetrahedral volume mesh.
<b>QUADRILATERAL</b>	4-sided surface facets leading to a hexahedral volume mesh.					
<b>TRIANGLE</b>	3-sided surface facets leading to a tetrahedral volume mesh.					
<b>TOL</b>	1.0E-5	Coincident point tolerance.				

The finite element mesh is created by first meshing the region surfaces using the **MESH** command. The **FILL** Command [page 450] can then be used to generate the volume mesh.

The **MESH** command generates the surface mesh on all the Pre-Processor surface facets from the subdivision the user has specified for the edges. The two types of elements (**TRIANGLES** and **QUADRILATERALS**) that may be selected, determine which type of volume element will be created by the **FILL** command (tetrahedra or hexahedra).

If the **MESH** command fails on any surface facet because the variation of element size is too extreme or the discretization does not obey the rules for quadrilateral meshes, that facet is given the label

**DEBUG** so that it can be identified using [The DISPLAY Command \[page 436\]](#) or [The THREED Command \[page 503\]](#).

## Triangular meshes

Triangular meshes can be generated on any polygonal surface. The only restriction is that surfaces with more than 4 sides should be planar.

Points are automatically added to the surface to give a smooth variation of element size, given the edge subdivision. A regular triangular mesh will be created on 3 and 4 sided surfaces, if the edge subdivisions are equal on opposite sides. Additional points may be added to the surface mesh if the surface is curved. The difference between the real surface shape and the finite element approximation is checked against the **NORMAL** tolerance and points may be added to improve the finite element model.

In general the surface mesh should be created in the real coordinate system used for the model. There may be exceptional cases where the surface mesh is required to be created in the **PARAMETER** space of the surface, for example, to achieve a rapid variation in element size.

## Quadrilateral meshes

Quadrilateral meshes can only be generated if the model consists of regions which are all hexahedra, or degenerate hexahedra. The following conditions must be fulfilled in order to generate a mesh of quadrilateral surface elements:

- there are no facets with more than 4 sides
- all 4 sided facets have equal subdivisions on opposite sides
- all 3 sided facets have two sides with the same subdivision

## The **MODIFY** Command

---

### Menu route

**MODIFY -> Mesh number**  
**MODIFY -> Point coordinate**  
**MODIFY -> Subdivisions**  
**MODIFY -> Material properties**  
**MODIFY -> Boundary conditions**

### Command line parameters

Command	<b>MODIFY</b>	
Parameter	Default	Function
<b>MESH</b>	1	Number of finite element mesh to be modified.
<b>START</b>	1	Number of the first plane or layer. Planes are numbered: 1,2, 3, etc.; mid-extrusion planes and layers are numbered: 1.5, 2.5, 3.5, etc.
<b>THREED</b>	<b>NO</b>	Use 3D Viewer.
		<b>NO</b> No 3D Viewer.
		<b>YES</b> Use 3D Viewer to show the model during <b>MODIFY</b> .

The **MODIFY** command allows much of the finite element mesh data created by [The DEFINE Command \[page 393\]](#) and [The EXTEND Command \[page 448\]](#) to be changed. The interaction is very similar to that used in **DEFINE** and **EXTEND**, and therefore only the differences will be highlighted.

Only the topology of the base plane cannot be modified. [The EDIT Command \[page 444\]](#) and [The REDEFINE Command \[page 483\]](#) can be used to change this topology.

The data of the finite element mesh is separated into 4 sections, corresponding to the modes of the **DEFINE** command. The user is asked first to choose between points, subdivisions of base plane facets and extrusion layers, materials and volume properties or boundary conditions.

<b>MODIFY Modes</b>	
Keyword	Meaning
<b>BOUNDARY</b>	Boundary conditions can be changed.
<b>MATERIALS</b>	Material names, potential types, element types and properties can be changed.

MODIFY Modes	
Keyword	Meaning
POINTS	Points can be moved and mesh planes transformed.
QUIT	Leave the MODIFY command.
SUBDIVISIONS	Subdivision of the planes and the layers can be changed. Layers can be changed between LINEAR and QUADRATIC extrusions.

Valid replies are: **BOUNDARY**, **MATERIAL**, **POINTS**, **QUIT** or **SUBDIVISIONS**.

*In menu mode* modifications can be made to one plane or layer at a time. The program prompts for a value of **PLANE NUMBER** or **LAYER NUMBER**.

*In keyboard mode*, for each section, each mesh plane or extrusion layer of the mesh specified by the **MESH** parameter, starting with the plane or layer given by the **START** parameter, is presented to the user. Mesh planes are numbered 1, 2, 3, etc., and mid-extrusion planes and layers 1.5, 2.5, 3.5, etc. Layer 1.5 is between plane 1 and plane 2.

## Point Modification Mode

### Menu route

#### MODIFY -> Point coordinates

*In menu mode*, the plane selected is presented to the user. Points can be moved individually or in groups. Individual points are selected using menu item **Move Point** and can be repositioned graphically including the use of **Construction Lines and Grids [page 397]** or numerically. Points can be grouped using menu item **Select/de-select point** and transformed using **Transform points**. If no points have been grouped, all the points of the plane are transformed. If mistakes are made, the point or points can be returned to their original locations using **Undo**. This must be done before leaving the **MODIFY** command.

*In keyboard mode* each plane, starting from the plane number given by the **START** parameter and ending with the top plane is presented to the user. Points can be moved individually or in groups. Individual points are selected using the **<space>** cursor hit and can be repositioned using the cursor including the use of **Construction Lines and Grids [page 397]** or the keyboard. Points can be grouped using **K** and transformed using **T**. If no points have been grouped, then **T** will transform all the points of the plane. If mistakes are made, the point or points can be returned to their original locations using **U** cursor hit. This must be done before finishing modifications to the plane.

Full details of the cursor commands and the transformation options are given in the following sections.

## Point Selection Menu and Cursor Hits

Point Selection Menu and Cursor Hits												
Cursor hit	Menu Item	Function										
<space>	Move point	Select point nearest cursor to be moved. It can be repositioned using <a href="#">The Point Repositioning Mode Menu and Cursor Hits [page 472]</a> .										
H		Display menu help message explaining all the cursor options. (More help available after point selection.)										
K	Select/de-select point	Select point nearest the cursor to be transformed. Repeating K for a selected point de-selects it.										
Q	Finish Editing	Leave this plane. If on a mid-extrusion plane, move on to the top plane of the new layer. Otherwise the program asks about the next extrusion ( <i>keyboard mode</i> ) or moves on to material definitions ( <i>menu mode</i> ).										
R	Re-draw picture	<p>Reconstruct the display. The program requests a new size. Valid replies are:</p> <table> <tr> <td>4 numeric values</td> <td><i>umin, umax, vmin, vmax</i> Default values are the current settings.</td> </tr> <tr> <td>CURSOR</td> <td>Select diagonally opposite corners of the display area with &lt;space&gt; cursor hits.</td> </tr> <tr> <td>RESTORE</td> <td>Return to previous screen size.</td> </tr> <tr> <td>BOUND</td> <td>Use bounding rectangle of geometry.</td> </tr> <tr> <td>&lt;return&gt;</td> <td>Reconstruct at the same size.</td> </tr> </table>	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.	CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.	RESTORE	Return to previous screen size.	BOUND	Use bounding rectangle of geometry.	<return>	Reconstruct at the same size.
4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.											
CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.											
RESTORE	Return to previous screen size.											
BOUND	Use bounding rectangle of geometry.											
<return>	Reconstruct at the same size.											
T	Transform points	Define general transformations for the points on the plane. (Use K to select points, or all points will be transformed.) See <a href="#">Plane Transformation Options [page 418]</a> .										

<b>Point Selection Menu and Cursor Hits</b>		
Cursor hit	Menu Item	Function
<b>U</b>	Undo move or transform	Undo the last move or transform operation on this plane or layer.
<b>V</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
<b>Z</b>	Aspect-ratio search	Switch <a href="#">Aspect Ratio Searching [page 396]</a> on or off.

## The Point Repositioning Mode Menu and Cursor Hits

<b>Point Repositioning Mode Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	At mouse	Reposition the point at the cursor cross hair position or the nearest grid point.
<b>C</b>	Give U, V, W	Switch to input from keyboard in cartesian coordinates UVW. Coordinates should be entered in free format. Default values of U, V and W are the values prior to the move. They can be accessed via the variables #1, #2 and #3. Type Q to leave point in its present position.
<b>E</b>	Remove C_line	Erase construction line closest to cursor cross hair.
<b>G</b>	Grid	Define or remove a grid.
<b>H</b>		Display menu help message explaining all the cursor options.
<b>I</b>	Enter C_lines	Input construction line specifications.
<b>L</b>	List C_lines	List construction line specifications.
<b>N</b>	On nearest C_line	Reposition the point on the nearest construction line with minimum normal distance. This only affects the U and V coordinates; W remains unchanged.

Point Repositioning Mode Menu and Cursor Hits												
Cursor hit	Menu item	Function										
P	Give R, Theta, W	Switch to input from keyboard in local polar coordinates RθW. Coordinates should be entered in free format. Default values of R, θ and W are the values prior to the move. They can be accessed via the variables #1, #2 and #3. Type Q to leave point in its present position.										
Q	Return without moving	Leave the point at its previous position.										
R	Re-draw picture	<p>Reconstruct the display. The program requests a new size. Valid replies are:</p> <table> <tr> <td>4 numeric values</td> <td><i>umin, umax, vmin, vmax</i> Default values are the current settings.</td> </tr> <tr> <td>CURSOR</td> <td>Select diagonally opposite corners of the display area with &lt;space&gt; cursor hits.</td> </tr> <tr> <td>RESTORE</td> <td>Return to previous screen size.</td> </tr> <tr> <td>BOUND</td> <td>Use bounding rectangle of geometry.</td> </tr> <tr> <td>&lt;return&gt;</td> <td>Reconstruct at the same size.</td> </tr> </table>	4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.	CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.	RESTORE	Return to previous screen size.	BOUND	Use bounding rectangle of geometry.	<return>	Reconstruct at the same size.
4 numeric values	<i>umin, umax, vmin, vmax</i> Default values are the current settings.											
CURSOR	Select diagonally opposite corners of the display area with <space> cursor hits.											
RESTORE	Return to previous screen size.											
BOUND	Use bounding rectangle of geometry.											
<return>	Reconstruct at the same size.											
T	Show coordinates	Type the coordinates of the point closest to the cross-hairs, and select it for repositioning instead of the point already selected. The coordinates are given in cartesian (UVW) and polar (RθW) coordinates.										
x	At C_line intersection	Reposition the point at the closest construction line intersection or end. This only affects the U and V coordinates; W remains unchanged.										
z	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.										

## Plane Transformation Options

The transformations can be any combination of the following:

Plane Transformation Commands	
Command	Parameters and Function
CARTESIAN	$exp\_u \ exp\_v \ exp\_w$ The points are moved to new positions defined by expressions for their ( $u$ , $v$ and $w$ ) coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, (#COPY), can also be used.
DISPLACE	$du \ dv \ dw$ Displace points in the current viewing local coordinate system by adding ( $du$ , $dv$ , $dw$ ) to their coordinates (U, V, W).
POLAR	$exp\_r \ exp\_\theta \ exp\_w$ The points are moved to new positions defined by expressions for their ( $r$ , $\theta$ and $w$ ) coordinates in terms of their old coordinates: <b>U</b> , <b>V</b> , <b>R</b> , <b>TH</b> and <b>W</b> . The copy number, (#COPY), can also be used.
PROJECT	$du \ dv \ dw \ ucentre \ vcentre \ wcentre \ uangle \ vangle \ wangle$ Project the points in the direction ( $du$ , $dv$ , $dw$ ) until they intersect the XY plane of a coordinate system specified by its origin ( $ucentre$ , $vcentre$ , $wcentre$ ) and axis rotation angles ( $uangle$ , $vangle$ , $wangle$ ).
QUIT	End the sequence of transformations.
ROTATE	$ucentre \ vcentre \ wcentre \ uangle \ vangle \ wangle$ Rotate points by angles ( $uangle$ , $vangle$ , $wangle$ ) around axes parallel to the local coordinate system and passing through the point ( $ucentre$ , $vcentre$ , $wcentre$ ).
SCALE	$ucentre \ vcentre \ factor$ Scale points by multiplying the distance from local coordinate point ( $ucentre$ , $vcentre$ ) by <i>factor</i> . The W coordinate of the points is not affected.

In menu mode the transformation commands are offered as menu items and the parameters are supplied using parameter boxes.

## Subdivision Modification Mode

The two mesh generators (for tetrahedra and hexahedra) have different requirements for the subdivision of mesh volumes. Hexahedral elements can only be generated in hexahedral volumes or degenerate hexahedral volumes with regular subdivision (i.e. opposite lines of each facet must have the same numbers of elements). Tetrahedral elements can be generated in any volume with any subdivision.

To support the hexahedral mesh generator, the subdivision of the facets in the mesh plane should be modified use the method described in [Subdivision Modification Mode \[page 474\]](#), accessed by using **START=1**. The subdivision of extrusion layers should be modified as described in [Modifying Subdivisions of the Extrusion Layers \[page 476\]](#), avoiding the **EDIT** option.

The **EDIT** option ([Modifying Subdivisions for Tetrahedral Meshes \[page 477\]](#)) provides facilities for modifying the subdivision of lines in a more general way to support the flexibility of the tetrahedral mesh generator.

## Modifying the Subdivisions of All the Planes

### **MODIFY -> Subdivision -> Uniform subdivision -> In-plane**

*In keyboard mode* a facet is selected for subdivision by positioning the cross-hairs just inside a facet, close to the edge to be divided and pressing the **<space>** bar. The program prompts for the number of subdivisions. Subdivisions can also be set globally, i.e. all edges set to the same subdivision number, using the **G** cursor hit.

*In menu mode* the subdivision can be set first using menu item **Set subdivision** and can then be applied to a single edge or globally to all the in-plane lines in the entire mesh.

If the subdivision is given for a single edge and that edge is one side of a quadrilateral facet, then the subdivision of the opposite edge will be set to the same value. The subdivision will be carried through the mesh until a triangular facet, the edge of the mesh or a facet already set in this operation is reached. (This does not happen if a polygon has been defined on the base plane.)

When the subdivision of an edge is set, the corresponding edges in all other layers are also set to the same subdivision.

Subdivisions of edges can be changed any number of times, until the required pattern is achieved.

## Facet Subdivision Menu and Cursor Hits

<b>Facet Subdivision Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
<b>&lt;space&gt;</b>	Apply to line	Select closest edge for its subdivision to be set.
<b>A</b>	Finish ... valid hex mesh?	Leave Facet Subdivision Modification, performing checks.
<b>G</b>	Apply globally	Select all edges for their subdivisions to be set to the same value.
<b>H</b>		Display menu help message explaining all the cursor options.

Facet Subdivision Menu and Cursor Hits		
Cursor hit	Menu item	Function
<b>Q</b>		Leave Facet Subdivision Modification and proceed to Extrusion Layer Modification.
<b>R</b>	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are: 4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings.
	<b>CURSOR</b>	Select diagonally opposite corners of the display area with <b>&lt;space&gt;</b> cursor hits.
	<b>RESTORE</b>	Return to previous screen size.
	<b>BOUND</b>	Use bounding rectangle of geometry.
	<b>&lt;return&gt;</b>	Reconstruct at the same size.
<b>x</b>	Finish ... no checking	Leave Facet Subdivision Modification, without checking.
<b>v</b>	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
<b>z</b>	Aspect-ratio search	Switch <b>Aspect Ratio Searching</b> [page 396] on or off.

## Modifying Subdivisions of the Extrusion Layers

### MODIFY -> Subdivision -> Uniform subdivision -> Extrusions

The program presents a list of the current settings of the extrusion subdivisions and requests that any necessary changes are made. The list gives the layer number, the number of extrusions and the layer type. The layer type is either **LINEAR** or **QUADRATIC**; a quadratic extrusion layer has a mid-extrusion plane and the W directed lines are quadratic.

The subdivision and layer type of each layer can be changed. Changing a linear layer to a quadratic layer causes a new mid-extrusion plane of points to be added. The positions of these points is initially at the geometric mid-point of the layer, but can be changed in Point Modification Mode. Changing a quadratic extrusion to linear removes the mid-extrusion plane.

Single line commands specify the changes to be made. These consist of the layer number, the number of subdivisions and the keyword **LINEAR** or **QUADRATIC**. \* can be used for the layer number to indicate that all layers should be set the same. Three keywords are available:

- **LIST** to list the current settings
  - **EDIT PLANE *n*** to switch to variable subdivision editing on plane *n*.
  - **EDIT LAYER *n*** to switch to variable subdivision editing in layer *n*.
  - **QUIT** to leave Extrusion Layer Modification Mode.
- Example - to change layer 3 to have 5 subdivisions and be quadratic, to list the settings and then leave Extrusion Layer Modification Mode:

```
OP-SUBDIV > 3 5 q
OP-SUBDIV > list
OP-SUBDIV > quit
```

## Modifying Subdivisions for Tetrahedral Meshes

### **MODIFY -> Subdivision -> Variable subdivision**

To support the flexibility of the tetrahedral mesh generator, the subdivision of the lines in mesh planes or extrusion layers can be modified using the Variable Subdivision option.

*In keyboard mode* this is selected using the **EDIT LAYER *n*** or **EDIT PLANE *n*** options in the extrusion subdivision modification mode.

The selected plane or layer is displayed so that one or more lines can be selected. Note that, for layers, the lines to be selected appear as the corners of the facets. A new number of subdivisions can be applied to the lines and also to the corresponding lines in other planes or layers using the **FROM** and **TO** keywords. The number of subdivisions can be given as an expression in terms of **N**, the existing number of subdivisions and **SIDE**, the length of the line.

## Line Selection Menu and Cursor Hits

<b>Line Selection Menu and Cursor Hits</b>		
Cursor hit	Menu item	Function
<space>	Select and define	Select line nearest cursor and prompt for subdivision.
H		Display menu help message explaining all the cursor options. (More help is available after entity selection.)

Line Selection Menu and Cursor Hits		
Cursor hit	Menu item	Function
K	Select/de-select line	Select the line nearest the cursor. Repeating K for a selected line, de-selects it. The subdivision for all the selected lines can be given after a <space> cursor hit.
Q	Finish	Leave this plane or layer.
R	Re-draw picture	Reconstruct the display. The program requests a new size. Valid replies are: 4 numeric values <i>umin, umax, vmin, vmax</i> Default values are the current settings. <b>CURSOR</b> Select diagonally opposite corners of the display area with <space> cursor hits. <b>RESTORE</b> Return to previous screen size. <b>BOUND</b> Use bounding rectangle of geometry. <return>      Reconstruct at the same size.
V	Change view	Change the UVW coordinate system. This affects the view and the extrusion direction.
Z	Aspect-ratio search	Switch <b>Aspect Ratio Searching [page 396]</b> on or off.

## Subdivision Sub-commands in Keyboard Mode

Following a <space> cursor hit, one of the following sub-commands should be given. The commands apply to the line selected with the <space> and to any previously selected with K.

The subdivision sub-commands consist of the following items:

Subdivision Sub-commands	
Command and parameter:	
number	Divide the line into <i>number</i> elements. <i>number</i> can be an expression in terms of N, the existing subdivision and SIDE, the length of the line.
HELP	Obtain help on all sub-commands and options.
QUIT	Abandon the currently selected lines.

<b>Subdivision Sub-commands</b>	
Optional keywords - setting multiple entities:	
<b>ALL</b>	Sets all lines in layer(s) or plane(s).
<b>FROM value</b>	Sets all layers/planes from layer/plane number given through to the current layer/plane or the layer/plane number given with <b>TO</b> .
<b>KEEP</b>	Keeps current selection of lines for another subdivision.
<b>TO value</b>	Sets all layers/planes from current layer/plane or the layer/plane number given with <b>FROM</b> through to the layer/plane number given. The value may be * to indicate the top layer/plane.

## Subdivision DialogBox in Menu Mode

Following **Select and define**, a value for the subdivision must be supplied in the DialogBox.



The value can be a number or an expression in terms **N**, the existing number of subdivisions or **SIDE**, the length of the line. Other inputs and buttons give access to other layers and to setting all lines in the layer or plane.

## Material Modification Mode

### Menu route

#### MODIFY -> Material properties

The material and mesh properties within each volume from the layer specified with the **START** parameter can be changed.

- *In keyboard mode*: Each layer is presented to the user in turn and values may be set which override the current settings within each volume.
- *In menu mode*: Only the layer selected by **START** is presented. Other layers can be modified by setting values for **From** and **To**.

The method for selecting volumes and defining materials is exactly the same as in the **DEFINE** command [Material Definition Mode \[page 420\]](#).

## Boundary Condition Modification Mode

### Menu route

#### **MODIFY -> Boundary conditions**

Boundary conditions can be set on any surface of any volume in the mesh. In order to achieve this the program presents the facets in 4 separate groups: first the facets on the base plane, second the planes between the first and the last, third the final plane of the mesh, and last the facets normal to the planes, one layer at a time. However, if the **START** parameter is greater than 1, then some of the planes are omitted. If **START** points to an extrusion layer then only the extrusion facets are presented.

Facets are selected and boundary conditions defined in exactly the same ways as in the **DEFINE** command [Boundary Condition Definition Mode \[page 426\]](#).

## The **READ** Command

---

### Menu routes

**FILE -> Read Pre-Processor file**  
**DEFINE -> Read conductor data**

### Command line parameters

Command	<b>READ</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of Opera-3d Pre-Processor data file.

The **READ** command reads a Pre-Processor data file into the program. There is one parameter which defines the name of the **FILE**. If no file name extension is given, the extension *oppre* is assumed.

Opera-3d Pre-Processor data files consist of all the commands issued to the program including cursor commands. Thus reading a file recreates the data in the same way as when commands are typed at the terminal except that fewer text messages appear on the screen and only the picture of the base plane is shown. All messages are written to the dialogue file *Pre\_n.lp*.

All the commands are decoded, so that the parameter values are stored but only certain commands are executed. These are the commands which define finite element data (**DEFINE**, **EXTEND** and **MODIFY**), the conductor data (**CONDUCTOR**) and built-in commands (**\$ OS** and **\$ CD** are not executed).

If any errors are encountered while reading the file, the number of the line last read is given. This may enable correction of the file. The file has comments in it to aid the user to follow the sequence of commands. These occur in place of the cursor commands which are introduced by the character string **CURS**. Comments have the character string **\*\*\*\*** followed by the number of following comment lines.

If the file does not contain all the commands necessary to complete a **DEFINE** command, normal terminal input is resumed when the end of the file is reached. It is advisable to use the **R** cursor hit first to obtain a picture of the current position within the **DEFINE** sequence.

The **READ** command should only be used to input files created using the **WRITE** command or the **WRITE** sub-command of **CONDUCTORS** (conductor files written by the Modeller and Post-Processor can also be read) or by renaming *Opera3d\_Pre\_nn.backup*. Other command input files, such as *Opera3d\_Pre\_nn.log* should be input using **\$ COMINPUT** (see [Command Input Files \[page 61\]](#)).

Pre-Processor data files can be edited to contain **\$ COMINPUT** commands and a command input file can also contain a **READ** command. However, if a command input file contains a **READ** command, the file to be read must not contain a **\$ COMINPUT** command.

The **READ** command in menu mode should only be used to read complete data files, i.e. files which contain complete **DEFINE** commands. Incomplete data files can only be read in keyboard mode.

## The **REDEFINE** Command

### Menu route

**DEFINE -> Redefine a mesh**

### Command line Parameters

Command	REDEFINE	
Parameter	Default	Function
MESH	1	Number of finite element mesh to be redefined.

The **REDEFINE** command removes all the volumes from a mesh, leaving only the baseplane points, lines and facets. It then enters the same command or menu sequence as [The \*\*DEFINE\*\* Command \[page 393\]](#) to add, remove or edit point, facets and subdivisions in the baseplane and then extrude. The only way this differs from the **DEFINE** command is that the baseplane facets from the mesh are already there when [Point Definition Mode \[page 399\]](#) is entered.

This implies that all the points on the baseplane will retain their existing **u**, **v** and **w** coordinates. Any new points added will use the specified **w** coordinate.

The parameter, **MESH**, allows any mesh in a multiple mesh model to be redefined.

## The **SLIP** Command

---

### Menu routes

**MODIFY -> Add slip surface**  
**MODIFY -> Remove slip surface**

### Command line parameters

Command	<b>SLIP</b>	
Parameter	Default	Function
<b>OPTION</b>	<b>ADD</b>	Command option:
		<b>ADD</b> Add slip surface.
		<b>REMOVE</b> Remove all SLIP boundary conditions.
<b>RADIUS</b>	1	Radius of slip surface.
<b>RTOLERANCE</b>	0.001	Tolerance on <b>RADIUS</b> .

The **SLIP** command is the most convenient way to apply the **SLIP** boundary condition to the interface between the stator and rotor of CARMEN rotating machine analysis models. The slip surface should be cylindrical, extend the full length of the model, and only touch volumes modelled with **AIR** and **TOTAL** scalar potential.

- The **SLIP OPTION=ADD** command adds a slip surface by applying the **SLIP** boundary condition to all facets whose points are at the given **RADIUS** within the specified **RTOLERANCE**. The label **SLIP** is also added to the points for use in [The TRANSFORM Command \[page 509\]](#).
- The **SLIP OPTION=REMOVE** command removes the **SLIP** boundary condition from all facets which have it and also removes the label **SLIP** from their points.

## The **SOLVERS** Command

---

### Menu route

**FILE -> Analysis**

### Command line parameters

Command	<b>SOLVERS</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Database file name.
<b>PROGRAM</b>	<i>none</i>	Analysis program: <b>CARMEN</b> Rotating machines. <b>ELEKTRA</b> Low frequency eddy currents. <b>SCALA</b> Space charge beam. <b>SOPRANO</b> High frequency. <b>TEMPO</b> Thermal analysis <b>TOSCA</b> Statics.
<b>TYPE</b>	<i>none</i>	Analysis type: <b>CURRENT</b> Current flow (TOSCA). <b>ELECTROSTATICS</b> Electrostatics (TOSCA). <b>EV</b> Eigenvalues (SOPRANO). <b>MAGNETOSTATICS</b> Magnetostatics (TOSCA). <b>ROTATIONAL</b> Rotational velocity (ELEKTRA). <b>SSAC</b> Steady-state AC (ELEKTRA or SOPRANO). <b>STEADYSTATE</b> Steady-state (TEMPO) <b>TRANSIENT</b> Transient (ELEKTRA and TEMPO). <b>VELOCITY</b> Linear velocity (ELEKTRA).

Command	<b>SOLVERS</b>	
Parameter	Default	Function
<b>OPTION</b>	<b>NEW</b>	Database option:
		<b>ADD</b> Add a new simulation to an existing database.
		<b>COPY</b> Copy a simulation within an existing database.
		<b>DEFAULT</b> Equivalent to <b>COPY</b> or <b>ADD</b> depending whether there is a volume mesh or not.
		<b>EDIT</b> Edit a pending simulation in an existing database.
		<b>NEW</b> Create a new database.
		<b>RESTART</b> Restart a completed analysis in an existing database.
<b>CASE</b>	1	Simulation number to be copied, edited or restarted.
<b>ELEMENT</b>	<b>MIXED</b>	Element-type over-ride.
		<b>LINEAR</b> All elements linear.
		<b>MIXED</b> Elements as previously set.
		<b>QUADRATIC</b> All elements quadratic.
<b>UNITS</b>	<b>CGS</b>	Units of the data.
		<b>CGS</b> Practical CGS units.
		<b>INCH</b> SI units with lengths in inches.
		<b>METRE</b> SI units.
		<b>MICRON</b> SI units with lengths in $\mu\text{m}$ .
		<b>MM</b> SI units with lengths in mm.
<b>DRSCALE</b>	<b>NO</b>	Enable drive scaling in TOSCA: <b>YES</b> or <b>NO</b>

This command creates or modifies a database file for analysis by one or more of the Opera-3d analysis programs. The command prompts for some additional information and has a set of sub-commands which should be used to set other options and material properties. The **SOLVERS** command has been designed to be used in menu mode and the options displayed in the menu boxes are restricted to those appropriate for the type of analysis selected.

## Analysis types

The parameters **PROGRAM** and **TYPE** together select the analysis type. The following are valid combinations.

Opera-3d Analysis Types			
PROGRAM	TYPE	Menu item	Meaning
CARMEN		Rotating Machine	Eddy currents induced by rotational motion in a transient field including nonlinear materials and permanent magnets.
ELEKTRA	SSAC	Steady-state AC	Eddy currents induced by low-frequency AC fields, including quasi-nonlinear materials.
	TRANSIENT	Transient	Eddy currents induced by transient fields, including nonlinear materials and permanent magnets.
	VELOCITY	Linear velocity	Eddy currents induced by linear motion of smooth components in a DC field, including nonlinear materials and permanent magnets.
	ROTATION	Rotation velocity	Eddy currents induced by rotational motion of smooth rotors in a DC field, including nonlinear materials and permanent magnets.
SCALA		Space charge beam	Electrostatics, including particle emitters and tracking and space charge effects.
SOPRANO	SSAC	Steady-state AC	High-frequency analysis data defined frequency.
	EV	Eigenvalues	Eigenvalue analysis of cavities.
TEMPO	STEADYSTATE	Steady-state	Steady-state thermal analysis.
	TRANSIENT	Transient	Transient thermal analysis

Opera-3d Analysis Types			
PROGRAM	TYPE	Menu item	Meaning
TOSCA	MAGNETIC	Magnetostatics	Nonlinear magnetostatics.
	ELECTRIC	Electrostatics	Nonlinear electrostatics.
	CURRENT	Current Flow	Nonlinear current flow.

## Database files and file names

Opera-3d database files are binary files. The Pre-Processor adds a file name extension *op3*.

A database file contains one finite element mesh and one or more simulations. The finite element mesh consists of:

- the nodes
- the elements, including material names and potential types and references to Pre-Processor facets and volume numbers
- the unit set

A simulation consists of:

- the analysis type
- the analysis options
- the boundary conditions
- the material properties
- the coils and drive specifications
- an optional title

When a database is created, the finite element data is stored in it, together with a simulation, which is initially marked as 'pending'. Additional simulations can be added to a database. During analysis, the Opera-3d analysis programs will solve any pending simulations of the appropriate type found in the database.

## Creating a new database

### Menu route

**FILE -> Analysis ... create new database**

## Command line parameters

**SOLVE OPTION=NEW,**

Before a new database can be created, the surface and volume mesh must be generated (see [The MESH Command \[page 467\]](#) and [The FILL Command \[page 450\]](#)). The program writes the database with the finite element data and creates a new simulation of the type selected.

Two sets of options are available when creating a new database: the units being used and the element type.

## Units

The Opera-3d databases and analysis programs have one set of units. The Pre-Processor data is converted to the internal units when the database is created. The Pre-Processor data is interpreted in any one of five unit systems. These are:

Units					
Menu Item	CGS	SI (metres)	SI (mm)	SI (Microns)	SI (Inches)
Keyword	CGS	SI	MM	MICRON	INCH
length	cm	m	mm	μm	inch
magnetic flux den.	gauss	tesla	tesla	tesla	tesla
magnetic field str.	oersted	amp/m	amp/m	amp/m	amp/m
magnetic scalar pot.	oersted cm	amp	amp	amp	amp
magnetic vector pot.	gauss cm	weber/m	weber/m	weber/m	weber/m
electric flux den.	coulomb/cm <sup>2</sup>	coulomb/m <sup>2</sup>	coulomb/m <sup>2</sup>	coulomb/m <sup>2</sup>	coulomb/m <sup>2</sup>
electric field str.	volt/cm	volt/m	volt/m	volt/m	volt/m
electric potential	volt	volt	volt	volt	volt
conductivity	siemens/cm	siemens/m	siemens/mm	siemens/μm	siemens/inch
current density	amp/cm <sup>2</sup>	amp/m <sup>2</sup>	amp/mm <sup>2</sup>	amp/μm <sup>2</sup>	amp/inch <sup>2</sup>
mass	g	kg	kg	kg	kg

The unit set is stored in the database and loaded into the Post-Processor.

## ELEMENT over-ride

The element types ([LINEAR](#) or [QUADRATIC](#)) are set by

- the [DEFINE command Material Definition Mode \[page 420\]](#),
- the [EXTEND command Material Definition Mode \[page 449\]](#)
- and the [MODIFY command Material Modification Mode \[page 479\]](#).

It is possible to over-ride these settings to set all elements to one type (**LINEAR** or **QUADRATIC**), or to leave them as previously set (**MIXED**).

Element Types	
Keyword	Meaning
<b>LINEAR</b>	All elements linear (8-noded hexahedra or 4-noded tetrahedra).
<b>MIXED</b>	Element types as set in <b>DEFINE</b> , <b>MODIFY</b> or <b>EXTEND</b> commands.
<b>QUADRATIC</b>	All elements quadratic (20-noded hexahedra or 10-noded tetrahedra).

## Editing an existing database

### Menu route

**FILE -> Analysis ... use existing database**

### Command line parameters

```
SOLVE OPTION=ADD,  
SOLVE OPTION=COPY,CASE=n,  
SOLVE OPTION=EDIT,CASE=n,  
SOLVE OPTION=RESTART,CASE=n,
```

### Adding a new simulation

To add a new simulation to an existing database, it is necessary that the Pre-Processor contains the corresponding model and that the surface and volume mesh are generated.

Before meshing, the **MODIFY** command **Boundary Condition Modification Mode** [page 480] can be used to change the boundary conditions. Material names, potential types, point coordinates and subdivisions should not be changed. The coils can also be changed using **The CONDUCTOR Sub-command MODIFY** [page 387] and the external field using **The CONDUCTOR Sub-command EXTERNAL** [page 386].

As the simulation is added, the analysis options and material properties can be set.

**NB:** Potential types which are not part of the formulation of an analysis program are treated as a valid type. TOSCA will treat **VECTOR** potential volumes as **TOTAL** scalar potential; SOPRANO will treat all volumes as **VECTOR** potential; TEMPO ignores potential types.

## Copying a simulation

Copying a simulation does not need the Pre-Processor model or finite element mesh. A new simulation of the same analysis type is created. Only the material properties and analysis options can be changed.

## Editing a simulation

Editing a simulation does not create a new simulation. It allows the material properties and analysis options of a pending simulation to be changed.

## Restarting a simulation

Restarting a simulation does the same as copying, allowing the same changes, except that within the analysis program, the old solution is used as an initial solution.

# SOLVERS Command Prompts

The **SOLVERS** command prompts the user for some additional information:

- a title (all analysis types)
- drive information (CARMEN, ELEKTRA, SOPRANO/SS and TOSCA)
- eigenvalues (SOPRANO/EV)

## Title

Simulations can be annotated with a title. This can have as many lines of up to 80 characters as required. It is delimited by a \* as the first character of the last line. Titles can be displayed by the Post-Processor using [The SHOW Command \[page 804\]](#) and the first line of the title from the currently loaded database is available in the string variable, **TITLE**.

## Drive Information

Each conductor in an Opera-3d model has a drive label; non-zero potential boundary conditions and external fields can also have drive labels. The **SOLVERS** command prompts the user for drive information for each drive label.

- **Statics (ELEKTRA/VL and TOSCA):** the drive information is a scaling factor. If the analysis is the second or subsequent case with the conductor fields and integrals already in the database, the fields and integrals will be scaled from a previous case to reduce the analysis time.  
Note that the scaling is applied after the coil fields have been calculated. The conductor tolerance should therefore relate to the unscaled fields.
- **Steady-State AC:** the drive information required is the phase angle in degrees for each drive (zero corresponds to a cosine drive).

- **Transient:** the drive information specifies the shape in time of each drive using the following options:

Transient Drives		
Keyword	Meaning	Data
DC	$F = 1$ for all time.	none
STEP	$t < 0, F = 0$ $t \geq 0, F = 1$	none
RAMP	$t < 0, F = 0$ $t \geq 0, t \leq t_c : F = \frac{t}{t_c}$ $t \geq t_c, F = 1$	$t_c$
SINE	$t < 0, F = F(0)$ $t \geq 0, F = \sin((2\pi ft - \phi))$	$f$ and $\phi$
COSINE	$t < 0, F = F(0)$ $t \geq 0, F = \cos((2\pi ft - \phi))$	$f$ and $\phi$
PEAK	$t < 0, F = 0$ $t \geq 0 : F = ate^{(-t^2/b)}$ . a and b are chosen such that $F = 1$ at $t = t_c$ . $a = \frac{e^{1/2}}{t_c}, b = 2t_c^2$	$t_c$
RISE	$t < 0, F = 0$ $t \geq 0 : F = 1 - e^{(-t/t_c)}$ .	$t_c$
TABLE (switch on)	$t \leq 0, F = 0$ $t > 0 : F =$ cubic splines from tabulated data in a time-table file.	<b>filename</b>
TOFF (table switch off)	$t < 0, F = F(0)$ $t \geq 0 : F =$ cubic splines from tabulated data in a time-table file. $F(0)$ is the value of the function in the table file at time $t=0$ .	<b>filename</b>

The drive function is then used to scale the corresponding 'driving' field.

The program prompts for the additional information needed for each drive type. In addition, the keyword ROTATING should be specified for any drives which are in the rotating part of a CARMEN model.

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. Time-table files can be displayed using [The GRAPH Command \[page 751\]](#) of the Post-Processor.

## Eigenvalues

The data for the SOPRANO Eigenvalue analysis program includes the number of eigenvalues required. They can be found in a specified range of frequencies or above a specified frequency.

## SOLVERS Sub-commands

Seven sub-commands allow material properties and analysis options to set, edited and checked.

- **MATERIAL**: to define material properties (except for thermal properties)
- **SETTINGS**: to set the analysis options
- **PERIODICITY**: to define periodicity data (ELEKTRA, SCALA and TOSCA)
- **DRIVE**: to add drive fields from a table file
- **CASE**: to specify drive frequencies (ELEKTRA/SS and SOPRANO/SS), output times (ELEKTRA/TR) or output angles (CARMEN)
- **CHECK**: to check the database and display a summary of the materials and options selected
- **QUIT**: to leave the **SOLVERS** command.

## CASE sub-command

### Menu route

**FILE -> Analysis ... create new database -> Drive frequencies**  
**FILE -> Analysis... use existing database -> Output times**  
**FILE -> Analysis... use existing database -> Output angles**

### Command line parameters

Sub-Command	CASE	
Parameter	Default	Function
COMMAND		Command options:  ADD Add a new <b>VALUE</b> to the end of the list. DELETE Delete the item given by <b>POSITION</b> from the list. INSERT Insert a new value after the item given by <b>POSITION</b> . LIST List all the values. REPLACE Replace the <b>value</b> given by <b>POSITION</b> .
POSITION		Position in the list for <b>DELETE</b> , <b>INSERT</b> or <b>REPLACE</b> .
VALUE		Value for <b>ADD</b> , <b>INSERT</b> or <b>REPLACE</b> .

The **CASE** sub-command should be used to create and edit a list of multiple case data:

- ELEKTRA/SS and SOPRANO/SS: multiple frequencies. At least one frequency must be supplied.
- ELEKTRA/TR and TEMPO/TR: output times. The program automatically inserts zero in the first position in the list.
- CARMEN: output angles, unless the rotation speed is zero, in which case output times must be given.

## CHECK sub-command

### Menu route

**FILE -> Analysis ... create new database -> Check**  
**FILE -> Analysis ... use existing database -> Check**

### Command line parameters

Sub-Command	CHECK
No Parameters	

The **CHECK** sub-command should be used to ensure that all the relevant analysis options have been specified and that all materials have been given properties.

## DRIVE sub-command

### Menu route

**FILE -> Analysis ... create new database -> Add drive fields**  
**FILE -> Analysis ... use existing database -> Add drive fields**

### Command line parameters

Sub-Command	<b>DRIVE</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of table file containing fields.
<b>LABEL</b>	<i>none</i>	Label for drive.

The **DRIVE** sub-command adds drive fields to the analysis database from a table file. The fields replace the fields from coils for a TOSCA magnetostatics analysis or form an additional drive for a CARMEN or ELEKTRA analysis.

For the format of the table file, see [TABLE Files \[page 674\]](#). The first three columns should hold the node coordinates and the second three columns the magnetic fields.

If this command is used, the adaptive RHS integrals must not be used, since only nodal values of the drive field are available (see [SETTINGS sub-command \[page 499\]](#)).

The program prompts the user to provide the scaling factor (statics), phase angle (steady-state ac) or the drive shape (transient) which should be used with the additional drive. The prompts and replies take the same format as for coil and boundary condition drives (section [Drive Information \[page 491\]](#)).

## MATERIAL sub-command

### Menu route

**FILE -> Analysis ... create new database -> Materials**  
**FILE -> Analysis ... use existing database -> Materials**

### Command line parameters

Sub-Command	<b>MATERIAL</b>	
Parameter	Default	Function
<b>NAME</b>	<i>none</i>	Material name. ! for list.
<b>TYPE</b>	<b>LINEAR</b>	Material type:
		<b>LINEAR</b> Constant properties.
		<b>NONLINEAR</b> Properties depend on field strength.
<b>ANISOTROPY</b>	<b>ISOTROPIC</b>	Material structure:
		<b>ISOTROPIC</b> Same properties in all directions.
		<b>PACKED</b> Laminated material (TOSCA only).
		<b>MULTIPLE</b> Tensor properties.
<b>CANCEL</b>	<b>NO</b>	Cancel the material command to clear defaults. <b>CANCEL</b> always reverts to <b>NO</b> after use.
<b>MUX</b>	1	Relative permeability (isotropic or $\mu_{xx}$ ).
<b>MUY</b>		Relative permeability $\mu_{yy}$ .
<b>MUZ</b>		Relative permeability $\mu_{zz}$ .
<b>MPHA</b>	0	Complex phase lag for permeability.
<b>HCX</b>	0	X component of coercive force, $H_c$ .
<b>HCY</b>	0	Y component of coercive force, $H_c$ .
<b>HCZ</b>	0	Z component of coercive force, $H_c$ .
<b>BHX</b>		BH file (isotropic or X-direction).
<b>BHY</b>		BH file (Y-direction).
<b>BHZ</b>		BH file (Z-direction).
<b>SIGX</b>	0	Conductivity (isotropic or $\sigma_{xx}$ ).
<b>SIGY</b>		Conductivity $\sigma_{yy}$ .
<b>SIGZ</b>		Conductivity $\sigma_{zz}$ .
<b>SPHA</b>	0	Complex phase lag for conductivity.
<b>JEX</b>		JE file (isotropic or X-direction).
<b>JEY</b>		JE file (Y-direction).
<b>JEZ</b>		JE file (Z-direction).

Sub-Command	<b>MATERIAL</b>	
Parameter	Default	Function
<b>EPSX</b>	1	Relative permittivity (isotropic or $\epsilon_{xx}$ ).
<b>EPSY</b>		Relative permittivity $\epsilon_w$ .
<b>EPSZ</b>		Relative permittivity $\epsilon_{zz}$ .
<b>EPHA</b>	0	Complex phase lag for permittivity.
<b>DEX</b>		DE file (isotropic or X-direction).
<b>DEY</b>		DE file (Y-direction).
<b>DEZ</b>		DE file (Z-direction).

When a database is created, all materials have the properties of air or the values given by [The MATERIALS Command \[page 462\]](#). The MATERIAL sub-command can be used as many times as necessary to edit the material properties to the ones required.

Thermal material properties used by TEMPO can only be set using [The MATERIALS Command \[page 462\]](#).

Only the BH characteristic can be nonlinear; the parameters to specify JE or DE curves are ignored. See also [page 501](#) for a description of how the material TYPE and properties are interpreted in linear and nonlinear analyses.

The BH curves can be specified as `$default` to request the default BH curve which is displayed by the first use of [The BHDATA Command \[page 375\]](#).

If only the material name is given, then the default values for all the parameters are set as appropriate for that material. The MATE +CANC command clears the default values.

*In menu mode* only those properties relevant to the analysis type of the current simulation are presented for editing.

## PERIODICITY sub-command

### Menu route

**FILE -> Analysis ... create new database -> Periodicity conditions**  
**FILE -> Analysis ... use existing database -> Periodicity conditions**

### Command line parameters

Sub-Command	PERIODICITY		
Parameter	Default	Function	
<b>OPTION</b>	<i>none</i>	Option:	
		<b>ADD</b>	Add a new periodicity condition.
		<b>DELETE</b>	Delete a periodicity condition.
		<b>EDIT</b>	Edit a periodicity condition.
		<b>LIST</b>	List all periodicity conditions.
<b>NUMBER</b>	<i>none</i>	Condition number for <b>DELETE</b> or <b>EDIT</b> .	
<b>XBC</b>	<i>none</i>	Translation in x-direction.	
<b>YBC</b>	<i>none</i>	Translation in y-direction.	
<b>ZBC</b>	<i>none</i>	Translation in z-direction.	
<b>TBC</b>	<i>none</i>	Euler angle $\theta$ .	
<b>PBC</b>	<i>none</i>	Euler angle $\Phi$ .	
<b>SBC</b>	<i>none</i>	Euler angle $\Psi$ .	
<b>TYPE</b>	<i>none</i>	Connection type:	
		<b>NEGATIVE</b>	Potential changes sign
		<b>POSITIVE</b>	Potential has same sign

The **PERIODICITY** sub-command connects together nodes on facets which have the **SYMMETRY** boundary condition (see [Boundary Condition Definition Mode \[page 426\]](#)). The connection is made by use of a coordinate transformation. The points are first off-set by (**XBC**, **YBC**, **ZBC**) and are then rotated using the [Euler Angles \[page 83\]](#) (**TBC**, **PBC**, **SBC**). In most cases either the translations or the rotations will be zero. If a transformed (slave) node is coincident with an untransformed (master) node, then a periodicity condition is established between them, such that the potential of the slave is the same as (**TYPE=POSITIVE**) or the negative of (**TYPE=NEGATIVE**) the potential of the master.

Here are some examples of Euler angles:

Coordinate Transformation	$\theta$	$\Phi$	$\Psi$
rotation around X-axis	0	90	-90
rotation around Y-axis	0	0	0
rotation around Z-axis	0	$\Phi$	0

## QUIT sub-command

### Menu route

**FILE -> Analysis ... create new database -> Return**  
**FILE -> Analysis ... use existing database -> Return**

### Command line parameters

Sub-Command	<b>QUIT</b>
No Parameters	

The **QUIT** sub-command closes the database and leaves the **SOLVERS** command. **QUIT** also runs **CHECK** to display the final properties of the simulation.

It is good practice to follow the **QUIT** sub-command with [The WRITE Command \[page 510\]](#), so that all the **SOLVERS** sub-commands are recorded in the Pre-Processor data file.

## SETTINGS sub-command

### Menu routes

The routes

**FILE -> Analysis ... create new database ->**  
**FILE -> Analysis ... use existing database ->**

contain some of the following entries. The contents of the menus are adjusted for the analysis type selected for the simulation.

- Linear solution**
- Nonlinear solution**
- Simple RHS Integrals**
- Adaptive RHS Integrals**
- External fields**
- Scala iteration data**
- Timestep options**
- Rotation speed**
- Use Upwinding**
- No Upwinding**
- Automatic potential cuts**
- No potential cuts**

## Command line parameters

Sub-Command	SETTINGS	
Parameter	Default	Function
LINEAR	NO	Linear solution: YES or NO.
TYPE	NEWTON	Iteration type (ELEKTRA and TOSCA):
		NEWTON Newton-Raphson iterations.
		SIMPLE Simple updates.
NITERATIONS	21	Number of nonlinear iterations.
TOLERANCE	0.001	Nonlinear convergence tolerance.
RELAXATION	0.1	Under-relaxation factor (SCALA only).
RHS	ADAPTIVE	RHS integral type:
		SIMPLE Trapezium rule on each element edge.
		ADAPTIVE Up to 9 gauss points on each edge.
HXEXT	0	X component of the external field.
HYEXT	0	Y component of the external field.
HZEXT	0	Z component of the external field.
UPWINDING	NO	Use upwinding for ELEKTRA/VL analysis: YES or NO.
ITPTSTEP	21	Nonlinear iterations per time-step (ELEKTRA/TR).
UPDATE	SIMPLE	Time-stepping method (ELEKTRA/TR and TEMPO/TR).
		ADAPTIVE Adaptive Crank-Nicholson.
		SIMPLE Fixed time-step Crank-Nicholson
MAXADERR	1	Maximum % error in adaptive time-stepping.
DELTAT	0.001	Fixed or initial time-step.
RPM	3000	Rotation speed (CARMEN)
POTENTIALCUT	YES	Use automatic cuts in TOSCA: YES or NO.
INITIALTEMP	0	Initial temperature for TEMPO/TR

The **SETTINGS** command sets various analysis options.

*In menu mode* only those options relevant to the current simulation are available.

- **Linear or Nonlinear:**

CARMEN, ELEKTRA and TOSCA: Linear analyses use constant material properties; nonlinear analyses update the material properties, depending on the solution and re-solve. Nonlinear analyses can include some linear materials and linear analyses can include nonlinear materials. In this latter case, the material property characteristic (e.g. BH curve) is used to supply the value of the material property (permeability). The slope of the curve at zero flux-density is used.

TEMPO: Linear analyses use material properties and boundary conditions which are constant in time; nonlinear analyses can update material properties and boundary conditions as the local value of the temperature ( $T$ ) changes.

- **RHS Integrals:**

The line and surface integrals of coil fields which are part of TOSCA magnetostatics, CARMEN and ELEKTRA analyses can be done using simple integrals, which only use the values of coil fields at the nodes or using adaptive integrals which use additional field points in order to reduce the errors.

- **External Fields** can be added to TOSCA (magnetostatics), SCALA, ELEKTRA and CARMEN analyses. For ELEKTRA/SS, ELEKTRA/TR and CARMEN, the external field and associated drive label should be first defined using [The CONDUCTOR Sub-command EXTERNAL \[page 386\]](#). The values of the field vector can be modified using this command. Also see the Opera-3d User Guide for more information on External Fields.
  - **Automatic Potential cuts** can be used in TOSCA magnetostatic analysis to automatically insert potential cuts to avoid having multiply connected volumes, where a loop of total potential volume encloses a non-zero net current.
  - **SCALA Iterations** converge to a consistent set of particle trajectories and fields. The number of iterations and under-relaxation factor can be set.
  - **Upwinding** is a technique to improve analysis of moving systems (ELEKTRA/VL). The analysis program reports whether upwinding is required or not.
  - **Time-stepping** in ELEKTRA/TR and TEMPO/TR can use fixed time-steps or can automatically adjust the time-step to achieve a given accuracy.
- In ELEKTRA/TR the initial value of the solution is set by the drive functions. In TEMPO/TR, the initial temperature can be set to a constant value, an expression in terms of coordinates or  $T$  to use nodal values of temperature already stored in the database from an earlier analysis or added by the Post-Processor using [The TABLE Command \[page 817\]](#).
- **Rotation speed** in CARMEN.

## The **TABLE** Command

### Menu route

**FILE -> Write node table file**

### Command line parameters

Command	TABLE	
Parameter	Default	Function
FILE	<i>none</i>	Name of table file.
UNIT	CGS	Name of length unit.

The **TABLE** command is provided to facilitate an interface to other programs, primarily so that coil fields from other sources can be added to an Opera-3d database (see [DRIVE sub-command \[page 495\]](#)).

The **TABLE** command can only be used after the volume mesh has been generated using [The FILL Command \[page 450\]](#).

The coordinates of the points in the table are always stored as centimetres, and are scaled as necessary depending on the value of **UNIT**.

The format of table files is described in [TABLE Files \[page 674\]](#).

## The **THREED** Command

---

### Menu route

**DISPLAY -> 3d Viewer ... refresh display**

### Command line parameters

Command	THREED	
Parameter	Default	Function
<b>ELEMENT</b>	<b>NO</b>	Element display switch.
		<b>NO</b> No element subdivision.
		<b>SURFACE</b> Subdivision on volume surfaces.
		<b>VOLUME</b> Subdivision within volumes.
<b>MESH</b>	<b>ALL</b>	Mesh number, <b>ALL</b> or <b>NONE</b> .
<b>TYPE</b>	<b>VOLUME</b>	Type(s) of entities to be displayed. <b>VOLUME</b> , <b>FACET</b> , <b>LINE</b> , <b>POINT</b> , <b>ALL</b> or <b>SAME</b> .
<b>LABEL</b>	<b>NOTAIR</b>	Label(s) on entities.
<b>L1</b>	1	First layer to be displayed.
<b>L2</b>	*	Last layer to be displayed. * means top layer.
<b>COIL</b>	<b>YES</b>	Conductor display switch.
		<b>NO</b> Conductors not displayed.
		<b>YES</b> Conductors displayed.
<b>C1</b>	1	First conductor to be displayed.
<b>C2</b>	*	Last conductor to be displayed. * means highest numbered conductor.

Command	THREED	
Parameter	Default	Function
VECTORS	CONDUCTORS	Vector display switch.
	CONDUCTORS	Vectors show current direction on conductors.
	CURRENT	Vectors show current density direction.
	MATERIAL	Vectors show material orientation.
	NO	Vectors not displayed.
	VELOCITY	Vectors show velocity direction.

The **THREED** command starts or updates the picture in the 3D Viewer. Pictures show the three dimensional geometry of the finite element mesh and conductors using line-drawings or coloured surfaces with hidden surfaces obscured. The discretization can be included and vectors can be drawn to indicate the vector properties of the volumes and the current directions in the conductors. Colours are used to differentiate between volumes with different material names or to show values of assigned potentials. The parameters control the parts of the model included and other options.

The orientation, position and size of the model in the 3D Viewer and the contents of the picture can be adjusted dynamically using the mouse.

## Selecting Parts of the Finite Element Model

### Menu route

#### DISPLAY -> 3d Viewer ... select parts

The **THREED** command draws volumes and facets created with [The DEFINE Command \[page 393\]](#). By use of the **TYPE** and **LABEL** parameters, sub-sets of the entities can be selected in order to aid visualization of the model. The parameter **TYPE** can be set to **FACET** or **VOLUME**; the parameter **LABEL** can be set to individual labels or combinations of labels. The additional label, **NOTAIR** can be used to select all material name labels except **AIR**. Abbreviated label names can be used where the abbreviation is not ambiguous. Abbreviations can be followed by \* to indicate that all labels which match should be selected. An entity is drawn if it has any of the labels selected. For example, to display all facets with vector potential boundary conditions, **LABEL=A\*-ALL** could be used.

Labels are assigned automatically to parts of the model by the commands **DEFINE**, **EXTEND**, **MODIFY** and **SLIP**. Automatically assigned labels include **ALL**, material names, element types potential types and boundary condition types and functions. Additional labels can be given in two ways: The [LABEL Command \[page 458\]](#) can be used to give any label to any part of the model and [The](#)

**CHECK** Command [page 378] can be used to assign the labels **DEBUG** to volumes with bad shapes and **EXTERNAL** to facets which are not shared by two volumes.

The part of the finite element mesh displayed can also be restricted by the parameters **MESH**, **L1** and **L2**. The **MESH** parameter is used to select **ALL** meshes or one particular mesh (each **DEFINE** command creates a mesh with a new number). **MESH=NONE** can be used to omit the finite element mesh completely. **L1** and **L2** are used to select a subset of the layers of the mesh. **L2** can be set to **\*** to indicate the top layer.

## Selecting Conductors

### Menu route

**DISPLAY -> 3d Viewer ... select parts -> Conductors**  
**DISPLAY -> 3d Viewer ... select parts -> Conductor numbers**

The **THREED** command draws the conductors including any symmetry copies. This can be controlled by the parameters **COIL**, **C1** and **C2**. **+COIL** and **-COIL** switch the display of the conductors on and off, and **C1** and **C2** select a range of conductors for display. **C2** can be set to **\*** to indicate the highest numbered conductor.

## Other parameters

### Menu route

**DISPLAY -> 3d Viewer ... command style**

Three different types of picture can be displayed:

- **ELEMENT=NO** gives a wire-frame view without discretization. This is always available.
- **ELEMENT=SURFACE** gives a solid colour view showing the selected surfaces (by facet label or volume label) and the surface discretization. This option can only be used after **The MESH Command** [page 467] has been run successfully.
- Views of facets, with **LABEL** set to a single potential name (**AX**, **AY**, **AX**, **POTENTIAL**, **VOLTAGE** or the incident potentials) are displayed as contour plots to show the values of the potential on the facets.
- **ELEMENT=VOLUME**, **TYPE=VOLUME** gives a solid colour view of each individual element. The elements are shown shrunk in size so that internal elements are also visible. This option can only be used after **The FILL Command** [page 450] has been run successfully.

Arrows to show the direction of the volume vector properties can be drawn inside each volume. By default, the arrows are only drawn inside the conductors to show the current direction (**VECTOR=R=CONDUCTORS**) but they can also show the **MATERIAL** orientation, the **CURRENT** density direction or the **VELOCITY** direction. The arrow display can be switched off completely (**VECTOR=NO**).

## The 3D Viewer Menus

The picture in the 3D Viewer can be manipulated using the mouse. The main control is the left mouse button. Moving the mouse with the left button pressed changes the view of the model. The way in which the view is changed and other options can be selected from a menu which pops up when the right mouse button is clicked. The menu options can be selected using the left mouse button.

3D Viewer Menu	
<b>Translate</b>	The model can be moved in any direction by dragging with the left mouse button pressed. The cursor shape is  .
<b>Rotate</b>	The model can be rotated by dragging with the left mouse button pressed. Movement up or down rotates the model around the horizontal axis. Movement from side to side rotates around the vertical axis. This is the default state when the 3D Viewer is started. The cursor shape is  .
<b>Zoom</b>	The model can be moved closer or further away by dragging with the left mouse button pressed. Movement up the window brings the model closer; movement down the window pushes the model further away. The cursor shape is  .
<b>Views</b>	The options in the <b>Views</b> sub-menu can be used to re-initialize the view or to rotate the model so it is being viewed from either the positive or negative ends of the major coordinate axes.
<b>Options</b>	The <b>Options</b> sub-menu can be used to switch on or off components of the display, e.g. the axes.
<b>Quit</b>	The <b>Quit</b> option closes the 3D Viewer. It can be restarted using the <b>THREED</b> command.

## The **TITLE** Command

---

### Menu route

**View -> Title**  
**File ↓**  
**Print → PostScript to file**

### Command line parameters

Command	<b>TITLE</b>	
Parameter	Default	Function
<b>STRING</b>	<i>none</i>	A graphics window title.
<b>POSITION</b>	<b>TL</b>	Graphics window title position:
		<b>BC</b> Bottom centre.
		<b>BL</b> Bottom left.
		<b>BR</b> Bottom right.
		<b>TC</b> Top centre.
		<b>TL</b> Top left.
		<b>TR</b> Top right.
<b>KEEP</b>	<b>YES</b>	Title preservation switch:
		<b>NO</b> Only display title once.
		<b>YES</b> Display title on subsequent pictures as well.
<b>NOW</b>	<b>YES</b>	First appearance switch:
		<b>NO</b> Display after next graphics window clear.
		<b>YES</b> Display immediately.
<b>DATE</b>	<b>NO</b>	Date, time and page number switch:
		<b>NO</b> Date, time and page number not displayed.
		<b>YES</b> Date, time and page number displayed.
<b>EXTRA</b>	<b>YES</b>	Display units and problem data:
		<b>NO</b> Units and Problem Data not displayed.
		<b>YES</b> Units and Problem Data displayed.

## Notes

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 **POSITIONed** 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 subsequent pictures (**+KEEP**). **STRINGs** which contain spaces or commas or start with a non-alphabetic character must be enclosed in quotation marks ('').

The version number of the software can be included in the title using the string and system variables, **VERSION** using:

```
title string='Version: &version&'  
or  
title string='Version: %real(version,5)'
```

- The display of the **DATE**, time and page number can be switched on or off on subsequent pictures with **+DATE** and **-DATE**.
- The display of unit settings and information about the model can be switched on or off on subsequent pictures with **+EXTRA** and **-EXTRA**.

## The TRANSFORM Command

---

### Menu route

**MODIFY -> Transform labelled points**

### Command line parameters

Command	TRANSFORM	
Parameter	Default	Function
LABEL	ALL	Point label.
NEWX	X	Expression for new X-coordinates.
NEWY	Y	Expression for new Y-coordinates.
NEWZ	Z	Expression for new Z-coordinates.

The **TRANSFORM** command can be used to move a labelled set of points to new coordinates. The points must first be labelled using [The LABEL Command \[page 458\]](#). Then the new coordinates for the points (**NEWX**, **NEWY**, **NEWZ**) can be given as values or as expressions in terms of their existing cartesian (**X**, **Y**, **Z**) or cylindrical polar (**R**, **TH**, **Z**) coordinates

The following should be noted:

- Whenever [The LABEL Command \[page 458\]](#) is used to add a label to or remove a label from a volume, facet or line, the same operation is also applied to the points which define the volume, facet or line. This enables groups of points to be labelled together.
- [The SLIP Command \[page 484\]](#) adds the label **SLIP** to all the points on the slip surface.
- The **TRANSFORM** command cannot be undone unless the inverse transformation exists.

## The **WRITE** Command

---

### Menu route

**FILE -> Write Pre-Processor file**

### Command line parameters

Command	<b>WRITE</b>	
Parameter	Default	Function
<b>FILE</b>	<i>none</i>	Name of Opera-3d Pre-Processor data file.

The **WRITE** command writes an Opera-3d Pre-Processor data file. There is one parameter which defines the name of the **FILE**. If no file name extension is given, and extension *oppre* is assumed.

Pre-Processor data files consist of all the commands issued to the program including cursor commands, except that some erroneous cursor hits are omitted. Comments are included among the cursor commands to aid the user to follow the sequence of commands. Cursor commands are introduced by the character string **CURS**. Comments have the character string **\*\*\*\*** followed by the number of following comment lines.

- Example - the following is the beginning of a Opera-3d Pre-Processor data file:

```

DEFI
XY
0 100 0 100
**** 1
Start of point definition
CURS I 11 0.2461406E+03 0.1195938E+03 0.9931790E+02 0.3576628E+02
A 0 0 10 0 10 60
A 0 0 55 0 55 60
A 0 0 100 0 100 60
L 0 0 100 0 0
L 0 0 100 0 60
Q
CURS X 15 0.2475313E+03 0.5979688E+02 0.1001073E+03 0.1821565E+01
CURS X 15 0.1654844E+03 0.2058125E+03 0.5353201E+02 0.8470982E+02
CURS X 15 0.1293281E+03 0.1348906E+03 0.3300729E+02 0.4444981E+02
CURS X 15 0.8343750E+02 0.7787500E+02 0.6956694E+01 0.1208392E+02
CURS X 15 0.6953125E+02 0.5423438E+02-0.9374268E+00-0.1336082E+01
CURS X 15 0.1001250E+03 0.5423438E+02 0.1642964E+02-0.1336082E+01
CURS X 15 0.1682656E+03 0.5423438E+02 0.5511083E+02-0.1336082E+01
CURS I 11 0.1682656E+03 0.5423438E+02 0.5511083E+02-0.1336082E+01
L 0 0 10 0 30
L 0 0 100 0 30
Q
CURS X 15 0.8900000E+02 0.6675000E+02 0.1011434E+02 0.5768625E+01
CURS X 15 0.1724375E+03 0.9734375E+02 0.5747906E+02 0.2313569E+02
CURS X 15 0.2308438E+03 0.1279375E+03 0.9063436E+02 0.4050275E+02
CURS Q 6 0.2308438E+03 0.1279375E+03 0.9063436E+02 0.4050275E+02

```

```
***** 1
Start of face definition
CURS C 9 0.2308438E+03 0.5006250E+02 0.9063436E+02-0.3704319E+01
CURS M 10 0.2308438E+03 0.1251563E+03 0.9063436E+02 0.3892393E+02
CURS C 9 0.1613125E+03 0.1835625E+03 0.5116376E+02 0.7207923E+02
CURS C 9 0.1265469E+03 0.1585313E+03 0.3142846E+02 0.5786982E+02
CURS M 10 0.1501875E+03 0.1029063E+03 0.4484847E+02 0.2629334E+02
CURS C 9 0.1613125E+03 0.5006250E+02 0.5116376E+02-0.3704319E+01
CURS F 11 0.1515781E+03 0.5006250E+02 0.4563788E+02-0.3704319E+01
***** 1
End of face
CURS C 9 0.1515781E+03 0.5006250E+02 0.4563788E+02-0.3704319E+01
CURS M 10 0.1515781E+03 0.1112500E+03 0.4563788E+02 0.3102981E+02
CURS C 9 0.1223750E+03 0.1404531E+03 0.2906023E+02 0.4760746E+02
CURS C 9 0.7648438E+02 0.7787500E+02 0.3009633E+01 0.1208392E+02
CURS M 10 0.8621875E+02 0.6814063E+02 0.8535518E+01 0.6558037E+01
CURS C 9 0.8621875E+02 0.5284375E+02 0.8535518E+01-0.2125494E+01
CURS F 11 0.8621875E+02 0.5284375E+02 0.8535518E+01-0.2125494E+01
***** 1
End of face
CURS C 9 0.8621875E+02 0.5284375E+02 0.8535518E+01-0.2125494E+01
CURS M 10 0.8621875E+02 0.6396875E+02 0.8535518E+01 0.4189801E+01
CURS C 9 0.7370313E+02 0.7092188E+02 0.1430809E+01 0.8136862E+01
CURS C 9 0.7370313E+02 0.5423438E+02 0.1430809E+01-0.1336082E+01
CURS F 11 0.7370313E+02 0.5423438E+02 0.1430809E+01-0.1336082E+01
***** 1
End of face
```

# **Chapter 5**

# **Conductors**

## **Introduction**

---

Conductors are used throughout Opera-3d to provide magnetic fields from electric currents. The simplest use is in the Post-Processor, which can calculate fields from defined currents flowing in pre-defined conductor shapes without any finite element mesh.

In the Magnetostatic and Charged Particle solvers, defined current densities are used as the sources for the magnetic field. Similarly, Harmonic EM can be used with defined sinusoidally alternating current densities and the other Electromagnetic solvers (including Magnetization and Quench) with current densities which have a defined variation in time. In all these simulations, the current density is pre-defined. Used in this way, conductors are referred to as Biot-Savart conductors.

Biot-Savart conductors are not part of the finite element mesh. They are defined separately from the mesh using the [Conductor Commands \[page 536\]](#). The fields are calculated using the Biot-Savart law. It is necessary that Biot-Savart conductors are inside reduced potential volumes. The mesh need not match the conductor shape. If symmetry allows only part of a model to be meshed, all symmetries of Biot-Savart conductors must be specified.

In general in time-varying analyses, conductors form parts of more complex electric circuits with additional components, such as current or voltage sources, resistors, capacitors, inductors, diodes and switches. In such models, the currents flowing in the conductors are calculated from the circuit equations. Additional information about the resistance per unit length of the wire and number of turns is needed so that the correct resistance and inductance of the conductors can be included.

Conductors are not used in Current Flow, Electrostatics, High Frequency, Stress or Thermal solvers.

## Conductor Shapes

The programs have a wide range of pre-defined conductor geometries, ranging from simple solenoids to bedsteads and racetracks wound on the surface of cylinders. There are also primitive conductor elements which can be joined together to build up conductor circuits.

In the Pre-Processor, there are 2 forms of some conductor shapes, a simpler form and a more general form with a "G" prefix, e.g. **SOLENOID** and **GSOLENOID**. In other programs the name without the "G" prefix is used for the more general form.

The following conductors are available:

Conductor Shapes	
<b>SOLENOID</b>	Solenoid around Y' axis.
<b>GSOLENOID</b>	Generally orientated solenoid.
<b>RACETRACK</b>	Racetrack around Y' axis.
<b>GRACETRACK</b>	Generally orientated racetrack.
<b>BEDSTEAD</b>	Bedstead around Y' axis.
<b>GBEDSTEAD</b>	Generally orientated bedstead.
<b>HELIX</b>	Helical end racetrack.
<b>CPEND</b>	Constant perimeter end racetrack.
<b>ARC</b>	Circular arc element.
<b>GARC</b>	Circular arc element with a flux density tolerance.
<b>STRAIGHT</b>	Straight bar element.
<b>GSTRAIGHT</b>	Straight bar element with a flux density tolerance.
<b>BR8</b>	8 noded brick element.
<b>BR20</b>	20 noded brick element.

First the different shapes are introduced with the geometric parameters specific to each one. Later sections describe some of the common parameters in more detail:

- Local Coordinate Systems [page 529]
- Current Density, Drive Label and Tolerance [page 531]
- Symmetries and Reflections [page 532]

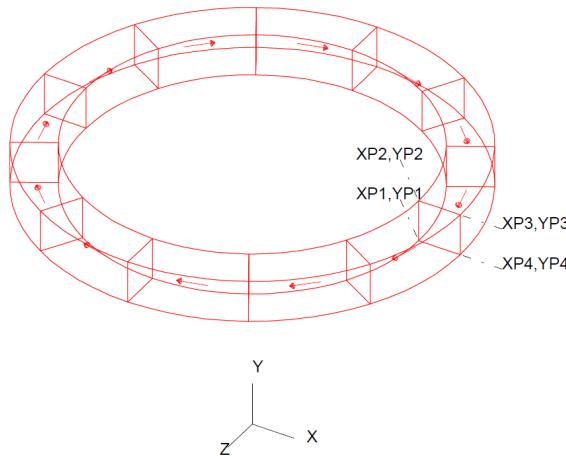
## Parameter Names

The parameter names given in the descriptions and pictures below correspond to the names used by the Modeller, except for local coordinate system 1 which is supplied by its name (**LCNAME**). In some

case, different parameter names are used in [Pre-Processor \[page 536\]](#) and [Post-Processor \[page 536\]](#).

## Solenoids

Two types of solenoid are available ([Figure 5.1](#)). The first (**SOLENOID**) has a restricted set of symmetries and can only be orientated with local coordinate system 1 without the Euler angles (**PHI1**, **THETA1**, **PSI1**). The second (**GSOLENOID**) can be generally orientated and replicated with any of the symmetries and reflections.



*Figure 5.1 A Solenoid*

The quadrilateral cross section of the solenoids in the local XY plane is defined by the coordinates of the 4 corners. (**XP1**, **YP1**, ..., **XP4**, **YP4**) and the curvatures of the sides (**CU1**, ..., **CU4**). If the 4 corners are entered in a clockwise sense when viewed from the positive Z axis, then positive curvature gives an increase in the area of the cross section.

Positive currents flow in the positive Z direction across the positive X half of the XY plane, assuming that the vertices of the cross section have been defined in a clock-wise sense; otherwise the direction of current flow is reversed.

### Parameters for solenoids

<b>SOLENOID</b>	
<b>XCENTRE, YCENTRE, ZCENTRE</b>	Local coordinate system 1
<b>XP1, YP1, XP2, YP2</b>	Solenoid cross section in X'Y' plane

<b>XP3, YP3, XP4, YP4</b>	Solenoid cross section in X'Y' plane
<b>CU1, CU2, CU3, CU4</b>	Solenoid cross section curvatures
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label

### Parameters for general solenoids

<b>GSOLENOID</b>	
<b>XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1</b>	Local coordinate system 1
<b>XCEN2, YCEN2, ZCEN2</b>	Local coordinate system 2 (origin)
<b>THETA2, PHI2, PSI2</b>	Local coordinate system 2 (Euler angles)
<b>XP1, YP1, XP2, YP2</b>	Solenoid cross section in X'Y' plane
<b>XP3, YP3, XP4, YP4</b>	Solenoid cross section in X'Y' plane
<b>CU1, CU2, CU3, CU4</b>	Solenoid cross section curvatures
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label
<b>IRXY, IRYZ, IRZX</b>	Reflections in local coordinate system 1 coordinate planes
<b>TOLERANCE</b>	Flux density tolerance

### Racetracks

Two types of racetrack are available (Figure 5.2). The first (

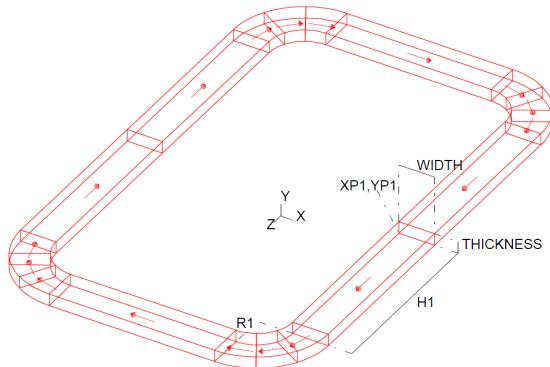


Figure 5.2 A Racetrack

**RACETRACK**) has the restricted set of symmetries and can only be orientated with local coordinate system 1 without the Euler angles (**PHI1, THETA1, PSI1**). The second (**GRACETRACK**) can be generally orientated and replicated with any of the symmetries and reflections.

The racetrack is made up of 4 straight sections and four 90 degree arcs. The cross section is rectangular, defined by its **WIDTH** in local X direction and **THICKNESS** in local Y direction. The coordinates of the bottom inside edge as it crosses the XY plane are given by **XP1** and **YP1**. The half-length of the z-directed straight is **H1** and the inside radius of the corners is **R1**. **R1** must be greater than zero.

Positive currents flow in the positive Z direction across the positive X half of the XY plane (local coordinates).

## Parameters for racetracks

<b>RACETRACK</b>	
<b>XCENTRE, YCENTRE, ZCENTRE</b>	Local coordinate system 1
<b>WIDTH, THICKNESS</b> ( <i>aliases A, B</i> )	Conductor cross section
<b>XP1, YP1</b>	Local coordinates of bottom inside corner
<b>H1, R1</b>	Half-length and corner radius
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label

## Parameters for general racetracks

GRACETRACK	
XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1	Local coordinate system 1
XCEN2, YCEN2, ZCEN2	Local coordinate system 2 (origin)
THETA2, PHI2, PSI2	Local coordinate system 2 (Euler angles)
WIDTH, THICKNESS ( <i>aliases A, B</i> )	Conductor cross section
XP1, YP1	Local coordinates of bottom inside corner
H1, R1	Half-length and corner radius
CURD, SYMMETRY, DRIVELABEL	Current density, symmetry and drive label
IRXY, IRYZ, IRZX	Reflections in local coordinate system 1 coordinate planes
TOLERANCE	Flux density tolerance

## Bedsteads

Two types of bedstead are available (Figure 5.3). The first (

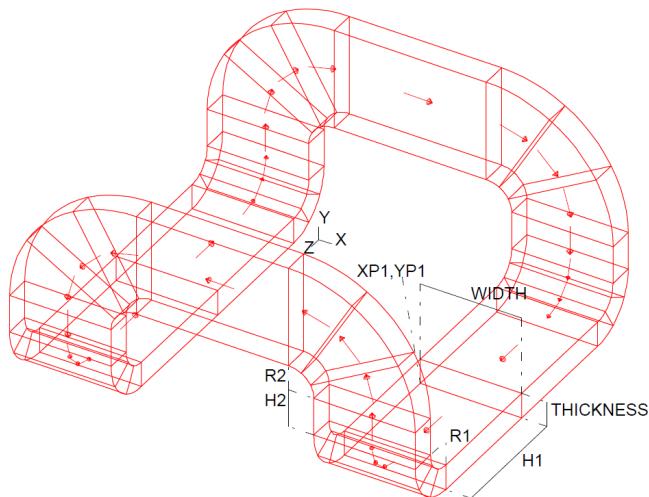


Figure 5.3 A Bedstead

**BEDSTEAD**) has the restricted set of symmetries and can only be orientated with local coordinate system 1 without the Euler angles (**PHI1, THETA1, PSI1**). The second (**GBEDSTEAD**) can be generally orientated and replicated with any of the symmetries and reflections.

The bedstead is made up of 8 straight sections and eight 90 degree arcs. The cross section is rectangular, defined by its **WIDTH** in local X direction and **THICKNESS** in local Y direction. The coordinates of the bottom inside edge as it crosses the XY plane are given by **XP1** and **YP1**. The half-length of the z-directed straight is **H1** and the inside radius of the corners at the ends of the z-directed straights is **R1**. The length of the straights in the local y direction is **H2** and the inside radius of the arcs connecting these straights to the straight bars which cross the YZ plane is **R2**. **R1** and **R2** must be greater than zero.

Positive currents flow in the positive Z direction across the positive X half of the XY plane (local coordinates).

### Parameters for bedsteads

<b>BEDSTEAD</b>	
<b>XCENTRE, YCENTRE, ZCENTRE</b>	Local coordinate system 1
<b>WIDTH, THICKNESS</b> ( <i>aliases A, B</i> )	Conductor cross section
<b>XP1, YP1</b>	Local coordinates of bottom inside corner
<b>H1, H2</b>	Lengths of straight sections
<b>R1, R2</b>	Inside radii
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label

### Parameters for general bedsteads

<b>GBEDSTEAD</b>	
<b>XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1</b>	Local coordinate system 1
<b>XCEN2, YCEN2, ZCEN2</b>	Local coordinate system 2 (origin)
<b>THETA2, PHI2, PSI2</b>	Local coordinate system 2 (Euler angles)
<b>WIDTH, THICKNESS</b> ( <i>aliases A, B</i> )	Conductor cross section

<b>GBEDSTEAD</b>	
<b>XP1, YP1</b>	Local coordinates of bottom inside corner
<b>H1, H2</b>	Lengths of straight sections
<b>R1, R2</b>	Inside radii
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label
<b>IRXY, IRYZ, IRZX</b>	Reflections in local coordinate system 1 coordinate planes
<b>TOLERANCE</b>	Flux density tolerance

## Helical Ends

The helical end (**HELIX**, Figure 5.4) is the first of two conductors which are

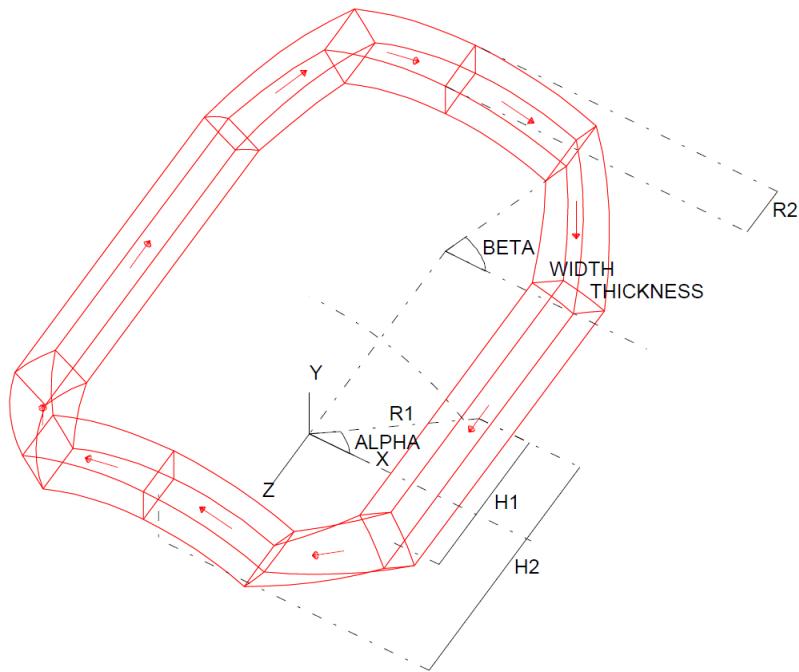


Figure 5.4 A Helical End

wound on the surface of a cylinder. It is made up of two straight sections parallel to the axis of the cylinder (the local Z axis). The half-length of the central filament of the straights is **H1**. The cross section is rectangular, defined by the **WIDTH** in the radial direction and **THICKNESS** in the azimuthal dir-

ection. The azimuthal position of the straights on the cylinder is defined by the angle from the mid i.e ZX, plane (**ALPHA**).

Each end of the conductor consists of two helices which extend to azimuthal angle **BETA** measured from the mid-plane and Z coordinate **H2**. The width can change along the helices so that the arc joining the helices together has a width **R2**. **R2** is normally set the same as **THICKNESS**.

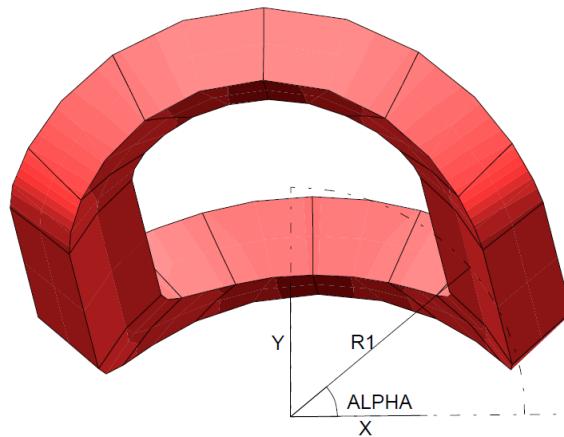
Positive currents flow in the positive Z direction across the positive X half of the XY plane (local coordinates).

### Parameters for helical ends

HELIX	
XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1	Local coordinate system 1
XCEN2, YCEN2, ZCEN2	Local coordinate system 2 (origin)
THETA2, PHI2, PSI2	Local coordinate system 2 (Euler angles)
WIDTH, THICKNESS ( <i>aliases A, B</i> )	Conductor cross section
H1, H2	Half-lengths of conductor
R1, R2	Radius of forming cylinder; thickness of cross-over arc
ALPHA, BETA	Angles to central filament of straight and end of helix
CURD, SYMMETRY, DRIVELABEL	Current density, symmetry and drive label
IRXY, IRYZ, IRZX	Reflections in local coordinate system 1 coordinate planes
TOLERANCE	Flux density tolerance

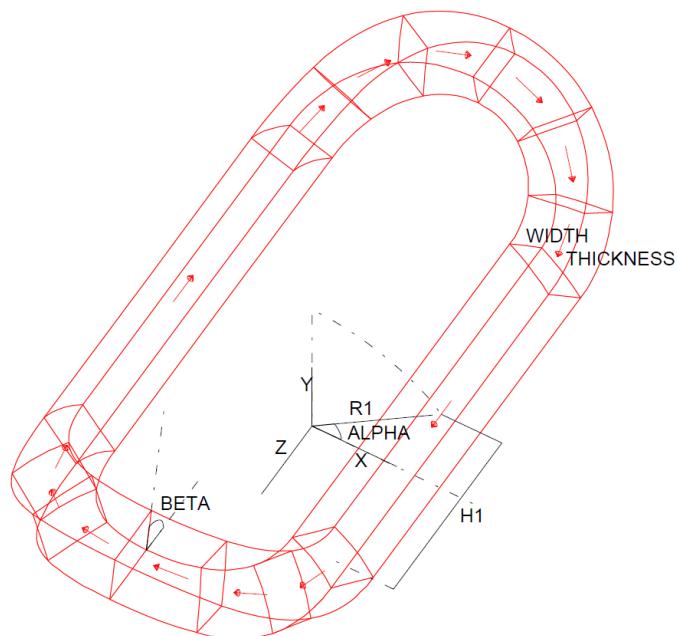
### Constant Perimeter Ends

The constant perimeter end (**CPEND**, Figure 5.5 and Figure 5.6)



*Figure 5.5 A Constant Perimeter End*

is also



*Figure 5.6 Another view of a Constant Perimeter End*

wound on the surface of a cylinder. It is made up of two straight sections parallel to the axis of the cylinder (the local Z axis). The half-length of the central filament of the straights is H1. The cross section is rectangular, defined by the **WIDTH** in the radial direction and **THICKNESS** in the azimuthal dir-

ection. The azimuthal position of the straights on the cylinder is defined by the angle from the mid. i.e ZX, plane (**ALPHA**).

There are two types of CPEND conductors, determined by the shape of the surface which touches the cylinder. It can be

- flat and **TANGENTIAL** to the cylinder or
- curved and **FITTING** the cylinder.

In the Modeller these are treated as separate shapes. The **FITTING** conductors are more accurate in geometry and field than the **TANGENTIAL**.

The ends of the conductor form a smooth curve over the cylinder. In manufacture they are machined by a cutter which has an angle **BETA** to the local Z axis of the cylinder and traverses a circular path on a cylinder of radius **R2**.

Positive currents flow in the positive Z direction across the positive X half of the XY plane (local coordinates).

### Parameters for constant perimeter ends

CPEND	
<b>XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1</b>	Local coordinate system 1
<b>XCEN2, YCEN2, ZCEN2</b>	Local coordinate system 2 (origin)
<b>THETA2, PHI2, PSI2</b>	Local coordinate system 2 (Euler angles)
<b>WIDTH, THICKNESS</b> ( <i>aliases A, B</i> )	Conductor cross section
<b>ALPHA, BETA</b>	Angles to central filament of straight and of cutting tool
<b>H1</b>	Half-length of straight
<b>R1, R2</b>	Radii of construction cylinders
<b>FIT</b>	Conductor fit to mandrel
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label
<b>IRXY, IRYZ, IRZX</b>	Reflections in local coordinate system 1 coordinate planes
<b>TOLERANCE</b>	Flux density tolerance

### Straight Bars

The **STRAIGHT** bar (or **GSTRAIGHT**), Figure 5.7 is a simple rectangular

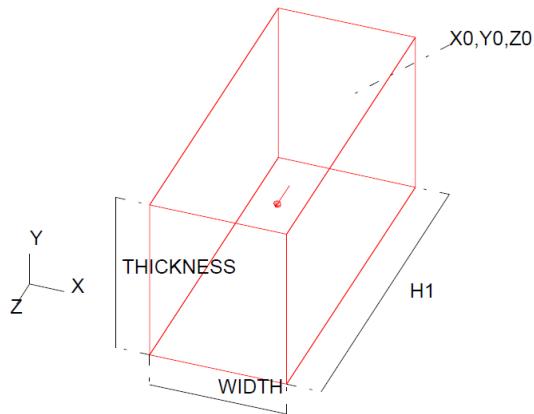


Figure 5.7 A Straight Bar

cross section conductor. Its cross section is **WIDTH** in the local X direction and **THICKNESS** in the local Y direction. Its central filament starts at the local origin and extends **H1** in the local Z direction. Positive currents flow in the positive local Z direction.

### Parameters for straight bars

<b>STRAIGHT or GSTRAIGHT</b>	
<b>XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1</b>	Local coordinate system 1
<b>XCENTRE, YCENTRE, ZCENTRE</b>	Local coordinate system 2 (origin)
<b>THETA2, PHI2, PSI2</b>	Local coordinate system 2 (Euler angles)
<b>WIDTH, THICKNESS (aliases A, B)</b>	Conductor cross section
<b>H1</b>	Length of straight
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label
<b>IRXY, IRYZ, IRZX</b>	Reflections in local coordinate system 1 coordinate planes
<b>TOLERANCE</b>	Flux density tolerance ( <b>GSTRAIGHT</b> only)

## Circular Arcs

The circular ARC (or GARC), Figure 5.8 is similar to the straight bar. Its

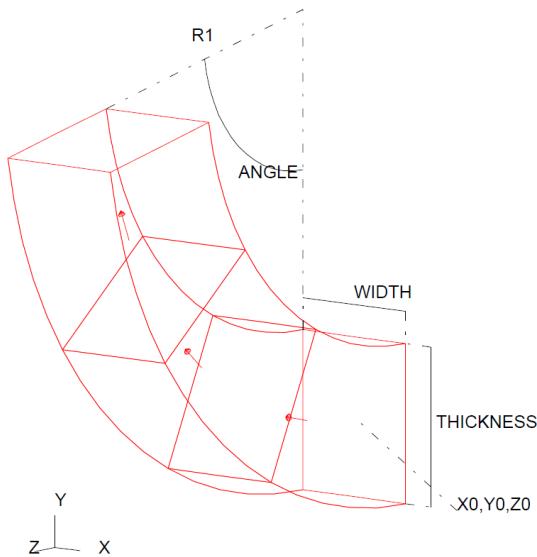


Figure 5.8 An Arc

cross section is **WIDTH** in the local X direction and **THICKNESS** in the local Y direction. Its central filament starts at the local origin and moves in the local Z direction. It bends through an angle **PHI** towards the positive local Y direction. Positive currents flow from the starting coordinates in the positive local Z direction.

### Parameters for circular arcs

ARC or GARC	
<b>XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1</b>	Local coordinate system 1
<b>XCEN2, YCEN2, ZCEN2</b>	Local coordinate system 2 (origin)
<b>THETA2, PHI2, PSI2</b>	Local coordinate system 2 (Euler angles)
<b>WIDTH, THICKNESS (aliases A, B)</b>	Conductor cross section
<b>R1, ANGLE</b>	Inside radius and angle

ARC or GARC	
CURD, SYMMETRY, DRIVELABEL	Current density, symmetry and drive label
IRXY, IRYZ, IRZX	Reflections in local coordinate system 1 coordinate planes
TOLERANCE	Flux density tolerance (GARC only)

## Bricks

Two brick elements can be used to defined more complex shapes, especially those which involve changes in cross section. They have 8 nodes (BR8) or 20 nodes (BR20, Figure 5.9).

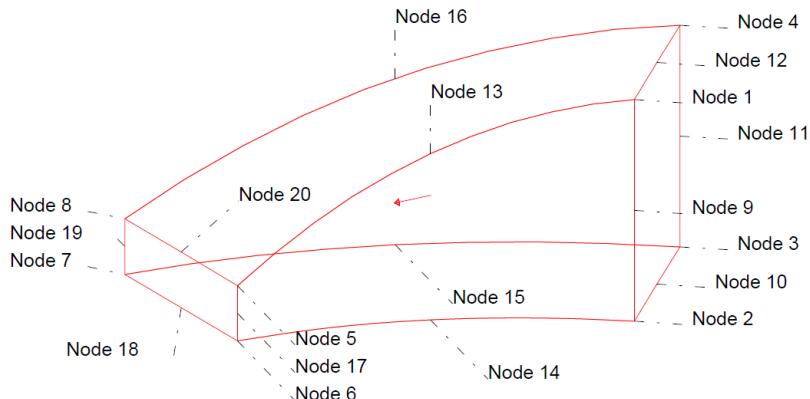


Figure 5.9 A 20-Node Brick

The bricks are defined by the coordinates of the nodes (XP1, YP1, ZP1, ..., XP20, YP20, ZP20). The current in a brick flows from face 1 (nodes 1, 2, 3, 4) to face 2 (nodes 5, 6, 7, 8). The total current in the brick is calculated from the current density (CURD) multiplied by the area of face 1. The 20 node brick has mid-edge nodes:

node 9	between nodes 1 and 2
node 10	between nodes 2 and 3
node 11	between nodes 3 and 4
node 12	between nodes 4 and 1
node 13	between nodes 1 and 5

node 14	between nodes 2 and 6
node 15	between nodes 3 and 7
node 16	between nodes 4 and 8
node 17	between nodes 5 and 6
node 18	between nodes 6 and 7
node 19	between nodes 7 and 8
node 20	between nodes 8 and 5

If the mid-edge nodes are not co-linear with the corner nodes, then the surfaces are parabolic. If the mid-edge nodes are not at the centre of the edges, the current density will vary over the cross section. The mid-edge nodes should not be displaced outside the  $\frac{1}{4}$  and  $\frac{3}{4}$  points.

The [DEFINE Command \[page 393\]](#) can also be used to define the geometry of brick conductors. This allows triangular or quadratic cross sections, which may be curved, to be extruded in the third direction. When the extrusions are complete the geometry is copied into the conductor storage area so that other conductor parameters, local coordinate systems, replications, current density etc. can be set to their correct values with the [MODIFY](#) sub-command.

## Parameters for 8-noded bricks

BR8	
XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1	Local coordinate system 1
XCEN2, YCEN2, ZCEN2	Local coordinate system 2 (origin)
THETA2, PHI2, PSI2	Local coordinate system 2 (Euler angles)
XP1, YP1, ZP1	Bottom right corner of front face
XP2, YP2, ZP2	Top right corner of front face
XP3, YP3, ZP3	Top left corner of front face
XP4, YP4, ZP4	Bottom left corner of front face
XP5, YP5, ZP5	Bottom right corner of back face
XP6, YP6, ZP6	Top right corner of back face
XP7, YP7, ZP7	Top left corner of back face
XP8, YP8, ZP8	Bottom left corner of back face
CURD, SYMMETRY, DRIVELABEL	Current density, symmetry and drive label

BR8	
IRXY, IRYZ, IRZX	Reflections in local coordinate system 1 coordinate planes
TOLERANCE	Flux density tolerance

### Parameters for 20-noded bricks

BR20	
XCENTRE, YCENTRE, ZCENTRE, PHI1, THETA1, PSI1	Local coordinate system 1
XCEN2, YCEN2, ZCEN2	Local coordinate system 2 (origin)
THETA2, PHI2, PSI2	Local coordinate system 2 (Euler angles)
XP1, YP1, ZP1	Bottom right corner of front face
XP2, YP2, ZP2	Top right corner of front face
XP3, YP3, ZP3	Top left corner of front face
XP4, YP4, ZP4	Bottom left corner of front face
XP5, YP5, ZP5	Bottom right corner of back face
XP6, YP6, ZP6	Top right corner of back face
XP7, YP7, ZP7	Top left corner of back face
XP8, YP8, ZP8	Bottom left corner of back face
XP9, YP9, ZP9	Mid-point of right edge of front face
XP10, YP10, ZP10	Mid-point of top edge of front face
XP11, YP11, ZP11	Mid-point of left edge of front face
XP12, YP12, ZP12	Mid-point of bottom edge of front face
XP13, YP13, ZP13	Mid-point of bottom right edge
XP14, YP14, ZP14	Mid-point of top right edge
XP15, YP15, ZP15	Mid-point of top left edge
XP16, YP16, ZP16	Mid-point of bottom left edge
XP17, YP17, ZP17	Mid-point of right edge of back face
XP18, YP18, ZP18	Mid-point of top edge of back face
XP19, YP19, ZP19	Mid-point of left edge of back face

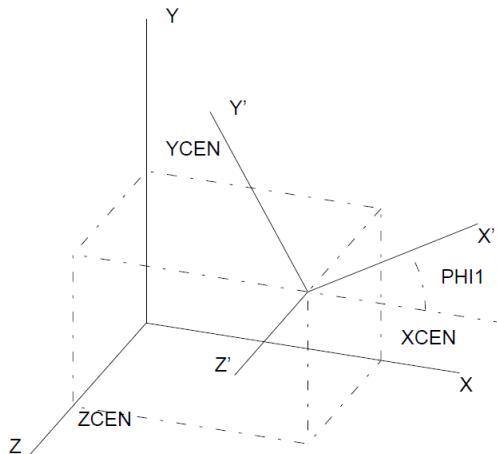
<b>BR20</b>	
<b>XP20, YP20, ZP20</b>	Mid-point of bottom edge of back face
<b>CURD, SYMMETRY, DRIVELABEL</b>	Current density, symmetry and drive label
<b>IRXY, IRYZ, IRZX</b>	Reflections in local coordinate system 1 coordinate planes
<b>TOLERANCE</b>	Flux density tolerance

## Local Coordinate Systems

To enable conductors to be oriented in space correctly, local coordinate systems can be defined. To reduce the amount of data necessary, symmetry and reflection codes can be used to replicate a basic shape.

### Local Coordinate System 1

The local coordinate system 1 is formed by displacing the origin with respect to the global origin to coordinates (**XCENTRE**, **YCEN**, **ZCEN**) and rotating by Euler angles (**PHI1**, **THETA1**, **PSI1**) (see [Euler Angles \[page 83\]](#) and [Figure 5.10](#)). **ANGLE** is an alias for **PHI1**.



*Figure 5.10 Local Coordinate System 1*

### Modeller LCS1

In the Modeller, the coordinate system to be used as local coordinate system 1 for a conductor is given by the name of a coordinate system.

### Local Coordinate System 2

The local coordinate system 2 is formed by displacing the origin with respect to local coordinate system 1 by (**XCEN2**, **YCEN2**, **ZCEN2**), and rotating by Euler angles (**THETA2**, **PHI2**, **PSI2**) (see [Euler Angles \[page 83\]](#) and [Figure 5.11](#)).

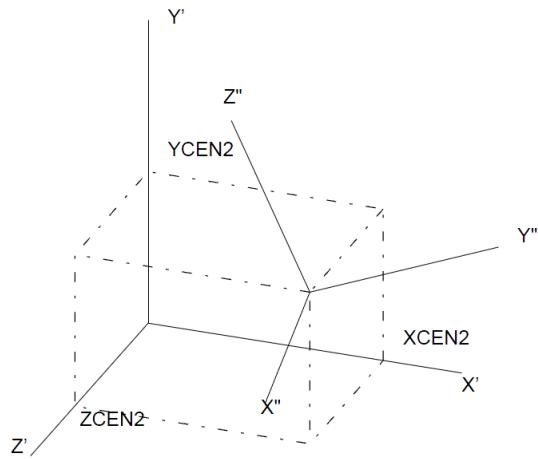


Figure 5.11 Local Coordinate System 2

# Current Density, Drive Label and Tolerance

## Current density

The current flowing in a Biot-Savart conductor is defined by the current density (**CURD**). For conductors with changing cross section the current density applies to the first face of the conductor. The system variable **AREA** holds the correct value of area so that the current density can be given in terms of the total current as **current/AREA**.

The current density distribution is uniform over the cross-section of Biot-Savart conductors, with the exception of 8-node and 20-node bricks. For these, the current density will be uniform only if the cross-section is rectangular.

## Drive label

The **DRIVELABEL** parameter is used to specify a drive label. Each drive label can be associated with a scaling factor (statics), phase angle (steady-state ac) or a drive function (transient).

## Tolerance

The fields from some conductors are calculated using an adaptive integration method which requires the user to supply a **TOLERANCE**. This specifies the error tolerance on the flux density in the units system being used. The flux density from conductors without a **TOLERANCE** parameter, or with **TOLERANCE** set equal to zero is calculated to a tolerance of 1 gauss or  $10^{-4}$  tesla. This is also the maximum allowed value of tolerance.

If the coil fields are multiplied by a scaling factor during analysis, note that the tolerance relates to the unscaled field values. In QUENCH analysis, the fields from conductors are calculated with the current density set to 1 and are scaled by the real values of current density. In this case as well, the tolerance should be set for the unscaled fields.

## Single filament conductors

A negative value of **TOLERANCE** can be used to request a single filament approximation to the conductors. The absolute value of the **TOLERANCE** is then used to control numerical integration along the trajectory of the current filament (fields are evaluated just by a line integral with no integration over the cross-section).

## Symmetries and Reflections

Conductors can have symmetries so that many copies of a defined shape can be made from a single definition.

### Rotational symmetry

**SYMMETRY=n** gives **n** copies of the basic shape each rotated by  $360/n$  degrees about the global Z axis with respect to the previous copy. If **n** is negative then the sign of the current alternates from one copy to the next allowing definition of quadrupoles, hexapoles, etc.

### Reflection symmetry

Conductors can also have reflections in major coordinate planes (xy, yz and zx) of local coordinate system 1. The reflection parameters can have values

- **0** for no reflection,
- **1** for a reflection with the same sign of current, or
- **-1** for a reflection with reverse sign of current.

### Pre-Processor conductors

The old style **SOLENOIDS**, **RACETRACKS** and **BEDSTEADs** can be reflected in the global ZX plane, with the currents in the reflection flowing in the same sense around the local Y' axis (**SYMM=2** or **-2**). Any other values of **SYMMETRY** give no reflection.

### Symmetries and circuits

When conductors with rotated or reflected copies are included in circuits, the copies are wound in series. For more information, see [The CIRCUIT Command \[page 173\]](#).

## Conductors in Circuits

Conductor defined using the Modeller can be included as windings in circuits using [The CEDITOR Command \[page 162\]](#) and [The CIRCUIT Command \[page 173\]](#). Additional parameters on the conductor definition and editing commands in the Modeller control the way in which such conductors are used.

<b>INCIRCUIT</b>	Is the conductor part of a circuit:	
	<b>NO</b>	The conductor has defined current density.
	<b>YES</b>	The current in the conductor is determined by a circuit.
<b>REVERSE</b>	Reverse the connections to this conductor in its circuit: YES or NO.	
<b>CIRCUITELEMENT</b>	The name of circuit element this conductor is part of.	
<b>MODELCOMPONENT</b>	<b>NO</b>	Do not convert to meshable cells.
	<b>REGULAR</b>	Convert to meshable cells which will be regularly meshed.
	<b>YES</b>	Convert to meshable cells which meet the mesh size criteria.
<b>MESHSIZE</b>	The size of the mesh to be used when meshing.	
<b>MESHLFACTOR</b>	Mesh factor to help split the conductor into multiple cells or control the mesh size through the cross-section.	

### Including a Conductor in a Circuit

To include as part of a circuit, set **INCIRCUIT=YES** and specify **CIRCUITELEMENT=name**, where **name** is the name which will be used to refer to the component when it is used in a circuit.

Multiple conductors can have the same name (this is typically used where a conductor is made up of a combination of individual arcs, straights and bricks). All conductors with the same **CIRCUITELEMENT** name form part of a series-connected set of windings. The number of turns and wire properties are defined with the circuit element and must be the same in each conductor section.

The definition of the conductor geometry defines an orientation of assumed positive current flow. If necessary, this can be reversed by changing the value of **REVERSE**. The direction of positive current flow within a conductor can be viewed using [The VECTOR Command \[page 339\]](#).

## Modelling the Conductor

When a conductor is included as part of a coupling between a circuit and the finite element model, the conductor must have a physical representation within the finite element mesh. This can be done using filaments or a volume mesh.

### Filamentary

By default (**MODELCOMPONENT=NO**), the physical representation of a conductor within the finite element mesh is achieved by creating a set of filament wire edges in the model body. These are then used as part of the finite element model as the current paths through the conductors. This representation is a good approximation where the conductors are of small cross-section, or where the magnetic flux is diverted around the conductors, such as in slots of electrical machines.

Filaments which intersect or are coincident are allowed but care should be taken to avoid filaments which lie in symmetry planes. This can happen if the symmetry plane cuts the conductor in half parallel to the current direction and the conductor is modelled with an odd number of filaments.

### Volume mesh

For other cases, using a meshed representation of the conductor may provide a better coupling of the inductive behaviour of the conductor. The geometry of the conductor is duplicated within the component geometry as cells with the material name set to match the **CIRCUITELEMENT** name. The mesh-size of the cells is controlled by the **MESHSIZE** and **MESHLFACTOR** parameters.

There are two options:

- **MODELCOMPONENT=YES**.<sup>1</sup> The maximum cell size in the current flow direction is set by **MESHSIZE**. If **MESHLFACTOR** is less than 1, the maximum cell size in the cross-section will be reduced to **MESHSIZE\*MESHLFACTOR**. This allows an anisotropic cell structure to be created within the geometry. The time taken to generate the geometry of the cross-section can become significant if the conductor needs to be subdivided into many cells.
- **MODELCOMPONENT=REGULAR**. The geometry of the conductor will be created as a single cell through the cross-section. However, additional meshing controls will be applied to the cells such that hexahedral mesh elements will be preferred. The **MESHLFACTOR** will be taken into account to provide additional layering in order to generate a mesh with elements which are smaller in the cross-section than in the current flow direction. The generation of the hexahedral mesh elements is subject to the normal geometry constraints, so may not be possible if the conductor intersects other geometry within the model.

The anisotropic mesh or geometry created using **MESHLFACTOR** can be particularly useful for modelling anisotropic material properties within the conductor for Quench analysis, or for improving post-processor calculations involving conductors.

---

<sup>1</sup>In 8 or 20-node brick conductors, **MESHLFACTOR** is ignored and assumed to have a value of 1 with **MODELCOMPONENT=YES**.

## Volume Mesh for Biot-Savart Conductors

---

Volume meshing of conductors, described in [Conductors in Circuits \[page 533\]](#) can also be used with conductors which are not in circuits. The options **MODELCOMPONENT=REGULAR** and **MODELCOMPONENT=YES** are available with Biot-Savart conductors, for improving the mesh in the space occupied by the conductors to improve post-processing. The behaviour is the same when creating the model body, except that the created cells will have a material name based on the drive label (*drive\_MATERIAL*) and potential set to **REDUCED**.

## Conductor Commands

---

### Modeller

The Modeller has one command for each conductor shape. These commands can be used to define a new conductor or to modify picked conductors of that shape. There is also a **CONDUCTOR** command which can be used to modify the common parameters of any conductor.

- The **ARC** Command [page 132]
- The **BEDSTEAD** Command [page 140]
- The **BRICK8** Command [page 156]
- The **BRICK20** Command [page 159]
- The **CONDUCTOR** Command [page 189]
- The **FITTEDCPE** Command [page 222]
- The **HELICALEND** Command [page 230]
- The **RACETRACK** Command [page 289]
- The **SOLENOID** Command [page 306]
- The **STRAIGHT** Command [page 314]
- The **TANGENTIALCPE** Command [page 321]

### Pre-Processor

In the Pre-Processor, all the commands for defining and modifying conductors are sub-commands of **The CONDUCTOR Command** [page 383].

Some parameters have names which are different from those described above:

- Corners of **SOLENOID**: **X1, Y1, ..., X4, Y4**
- Corner of **RACETRACK** and **BEDSTEAD**: **X1, Y1**
- Angle of an **ARC**: **PHI**
- Local coordinate system 2; **X0, Y0, Z0, T, P, S**
- Drive label: **PHASE**

### Post-Processor

The Post-Processor has one command for all operations on conductors: **The CONDUCTOR Command** [page 727].

Some parameters have names which are different from those described above:

- Corners of **SOLENOID**: **X1, Y1, ..., X4, Y4**

- Corner of **RACETRACK** and **BEDSTEAD**: **X1, Y1**
- Angle of an **ARC**: **PHI**
- Corners of **BRICK8** and **BRICK20**: **XB1, YB1, ZB1, ..., XB20, YB20, ZB20**
- Local coordinate system 2; **X0, Y0, Z0, T, P, S**
- Drive label: **PHASE**

## Conductor Files

---

Conductor files are useful for copying the conductors from one dataset to another, or for transferring the data between Modeller, Pre-Processor and Post-Processor. Only data which is common to all 3 programs is contained in conductor files; information about filamentary or meshed conductors is not included.

If no file name extension is given, the programs add the extension *cond*.

For hints on how to write compatible command scripts for all 3 programs see [Conditional Commands \[page 52\]](#).

### Writing Conductor Files

Conductor files can be written by:

- the Modeller, using [The EXPORT Command \[page 211\]](#);
- the Pre-Processor, using [The CONDUCTOR Sub-command WRITE \[page 391\]](#); and
- the Post-Processor, using [The CONDUCTOR Command \[page 727\]](#) with **ACTION=EXPORT**.

### Reading Conductor Files

Conductor definition (*cond*) files can be read:

- into the Pre-Processor using [The READ Command \[page 481\]](#) or the 'built-in' command, **\$ COMINPUT** (see [Command Input Files \[page 61\]](#)),
- into the Modeller using [The IMPORT Command \[page 235\]](#),
- and into the Post-Processor using [The CONDUCTOR Command \[page 727\]](#) with **ACTION=N=IMPORT**.

# **Chapter 6**

# **Analysis Programs**

## **Introduction**

---

This chapter describes the electromagnetic field and thermal analysis programs within the Opera-3d environment.

- [Motional Electromagnetic Solver \[page 543\]](#)
- [Magnetization Solver \[page 589\]](#)
- [Dynamic Electromagnetic Solvers \[page 542\]](#)
- [Quench Analysis \[page 597\]](#)
- [Charged Particle Solver \[page 548\]](#)
- [High Frequency Solvers \[page 585\]](#)
- [Stress Analysis \[page 599\]](#)
- [Static Thermal Solver \[page 591\]](#)
- [Transient Thermal Analysis \[page 594\]](#)
- [Static Electromagnetic Field Solvers \[page 541\]](#)

The analysis programs incorporate state of the art algorithms for the calculation of electromagnetic fields and advanced finite element numerical analysis procedures. A brief introduction to the use of finite elements is given in section [The Finite Element Method \[page 609\]](#) as an aid to application engineers who need to understand the limitations of the technique and evaluate the validity of their results. In sections [Codes of Practice \[page 612\]](#) and [Solution Errors \[page 613\]](#), some further discussions on good codes of practice, and techniques for reducing errors are presented.

Finite element discretization forms the basis of the methods used in the analysis programs. This widely applicable technique for the solution of partial differential equations requires special enhancements to make it applicable to electromagnetic field calculations. Access to these features is supported by the Opera-3d Modeller and Pre-Processor which provide facilities for the creation of finite element models, specification of complicated conductor geometry, definition of material characteristics including for example, nonlinearity and anisotropy and graphical displays for examination of the data. Full details are given in

- [The Geometric Modeller \[page 85\]](#)

- [Opera-3d Pre-Processor \[page 362\]](#)

## Static Electromagnetic Field Solvers

---

### Introduction

Opera-3d solvers can perform three types of static electromagnetic field analysis:

- Current Flow: electric currents driven through conducting materials by fixed potential on the boundaries;
- Magnetostatic fields from defined current sources or permanent magnets;
- Electrostatic fields from fixed potential boundaries, including lossy dielectrics for which current flow analysis provides the potential boundary conditions for electrostatic field analysis.

All analysis types can use nonlinear anisotropic material properties.

### The Magnetostatic, Electrostatic and Current Flow Algorithm

The Magnetostatic, Electrostatic and Current Flow algorithm is based on [Scalar Potential Formulation \[page 602\]](#).

In electric field analyses (Electrostatic and Current Flow) a single electric potential is used.

In Magnetostatic analysis two potentials are used. The total magnetic scalar potential *should* be used in magnetic materials and the reduced magnetic scalar potential *must* be used in regions which contain source currents. In most cases the **DEFAULT** choice given by [The CELLDATA Command \[page 168\]](#) is satisfactory. However, the choice of potential in regions where either potential is valid can affect the accuracy of the solution (see [Accurate Fields \[page 634\]](#) and [Total and Reduced Potential at the Open Boundary \[page 618\]](#)).

## Dynamic Electromagnetic Solvers

---

### Introduction

The Opera-3d Dynamic Electromagnetic solvers can be used to compute time varying electromagnetic fields in three dimensions including the effects of eddy currents. They do not model displacement current effects. Practically, this means that the largest dimension of the device being modelled must be less than 10% of the wavelength of the electromagnetic field in free space. The program incorporates state of the art algorithms for the calculation of electromagnetic fields and advanced finite element numerical analysis procedures. In the following sections the algorithm used in the Electromagnetic solvers described so that users are able to relate the finite element method to their model.

### Time Variation in the Dynamic Electromagnetic Solvers

There are three Dynamic Electromagnetic solvers, each having a different form of time variation.

- Harmonic calculates steady-state ac currents where all fields and potentials are oscillating at the same frequency.
- Transient calculates transient eddy currents induced by the fields of driving currents, boundary conditions and external fields which change in time in a predetermined way.
- Fixed Velocity calculates eddy currents induced by motion which does not change the geometry of the model. The source fields and driving conditions are all time invariant.

In each case the analysis uses edge variable elements and the [Vector Potential Formulation \[page 605\]](#). Harmonic and Transient EM can be used with the [Surface Impedance Boundary Condition \[page 620\]](#) to model materials with very small skin depths.

## Motional Electromagnetic Solver

### Introduction

The Motional Electromagnetic solver can be used to compute electromagnetic fields including the effects of eddy currents in moving systems in three dimensions. Parts of the mesh are allowed to move independently at speeds controlled by the user or calculated as the analysis proceeds. The analysis is a transient analysis, with eddy currents being induced in conducting media both through the effects of the moving magnetic fields, and through the time variation of the model sources.

Three types of motion are allowed:

- rotational motion: rotation about the z-axis;
- linear motion: displacement in x, y and z;
- general motion: rotation about z and displacement in z.

The analysis relies upon a remeshing technique. During the transient analysis, before the solution at the next time-step, the position of each moving part is updated. The model must be created in such a way that an "air gap" exists between any moving parts. The air gap regions are remeshed to form a continuous volume mesh coupling the different moving parts of the model.

For more details on the algorithms used see [Vector Potential Formulation \[page 605\]](#) and the time-stepping algorithm is described in [Time-Stepping \[page 624\]](#).

### Models for the Motional EM Solver

The definition of the moving parts and air gap makes use of the group label, attached to cells, to group them into sections (see [The CELLDATA Command \[page 168\]](#)). All cells in the gap region must have the label **GAP**, other parts can be named as required, with all cells having the same group label being in the same moving part.

The gap region must have properties of air and, in the case of linear motion, must be a single cell. The cells of other groups which touch the cells of the gap should also be air thus providing at least 3 layers of elements between the non-air materials.

Biot-Savart and circuit winding conductors can be in the stationary or moving parts. Any conductor in the model must also have a group label set to specify the part to which it belongs. As the part moves, the conductor will also move with that group. Conductors may not be part of the gap region.

### Pre-Processor Models for the Motional EM Solver

Modelling in the pre-processor for the Motional EM solver requires the use of a hexahedral mesh and definition of a slip surface. The model also requires a regular subdivision on the slip surface. The only movement supported is rotation about the z axis.

The slip surface is defined as a boundary condition, and runs the entire length of the device (z-direction). It should have a constant radius the entire length, and is labelled **SLIP** in the boundary definition stage of the Pre-Processor. As an alternative to selecting the slip surface in the boundary definition stage, the **SLIP** boundary condition can be applied using [The SLIP Command \[page 484\]](#). The user specifies the radius of the slip surface and the tolerance for inclusion of points not precisely at this radius.

During analysis, the solver will split the model into 3 sections labelled ROTOR, GAP and STATOR. The GAP group is a single layer of elements just outside the slip surface. The ROTOR contains all elements inside the SLIP surface. The STATOR group contains all other elements. Any Biot-savart conductor with a drive label marked as rotating will be considered as part of the rotor, all other conductors will be assigned to the stator.

During update the rotation of the rotor is checked, and if necessary, the elements in the GAP region are reformed by reconnecting. This reconnection requires that at any rotation angle of the rotor, the mesh can be rebuilt. This forces the requirement that the subdivision on the slip surface be regular.

## Motion Control

The control of motion and time-stepping the Motional EM solver is very flexible. Details of the controls available for time-stepping control can be found in [Command File Control \[page 629\]](#).

The Motional EM solver sets system variables that give information on the rotation angle, shift, speed, acceleration, force and torque. These variables can be used for logging and as part of the motion control **comi** file to alter the model behaviour, e.g. to change drive function when the moving part passes a position. To make these variables available for use in post-processing, they must be stored as user variables using commands in the time-stepping control file, e.g.

```
$constant #ro_torquez ro_torquez
```

In addition to these controls, the command file is also used to control the motion of the mesh sections in the model. Each of the rigid body sections can move independently and this file provides the method for this control.

## System variables

Each section of the model is identified by a group name. The first 2 characters of this group name determines the names of the control variables that are used, e.g.

- all variables for the group name **ROTOR** will begin **RO\_**,
- all variables for the group name **MOVING** will begin **MO\_**,
- for the name **STATOR** will begin **ST\_**, etc.

No variables are available for the GAP region.

In the following sections the letters **xx** are used as the first two characters of the group name. These letters should be replaced according to the actual names of the groups.

## Motion control

The type of motion control is achieved by setting the string variable `xx_MOTIONCONTROL`. With each different control type, there are sets of input variables which should be set by the motion command file and output variables which are set by the solver to describe the current state of the parts of the model. The following table shows the input and output variables.

Note that rotational motion output variables (with names including `TORQUE`, `THETA`, `OMEGA` or `ALPHA`) are only available for rotating and general motion; similarly, linear motion output variables (with names including `FORCE`, `SHIFT`, `SPEED` or `ACCEL`) are only available for linear and general motion.

<b>Control Type</b>	<b>Input variables set in the <i>comi</i> file</b>	<b>Output variables calculated by the program</b>
all control types		Torque and Forces <sup>a</sup> : <code>xx_TORQUEZ</code> <code>xx_FORCEX</code> <code>xx_FORCEY</code> <code>xx_FORCEZ</code>
<b>STATIONARY</b>	No further motion will occur.	
	none	none
<b>POSITION</b>	Directly set the position.	
	Angle and position: <code># xx_THETAZ,</code> <code># xx_SHIFTX,</code> <code># xx_SHIFTY,</code> <code># xx_SHIFTZ</code>	none
<b>SPEED</b>	Control the speed.	
	Speed: <code># xx_OMEGAZ,</code> <code># xx_SPEEDX,</code> <code># xx_SPEEDY,</code> <code># xx_SPEEDZ</code>	Angle and position <code>xx_THETAZ,</code> <code>xx_SHIFTX,</code> <code>xx_SHIFTY,</code> <code>xx_SHIFTZ</code>

---

<sup>a</sup>The forces and torques are calculated for the complete model including any rotational or mirror image copies of the section being analysed.

<b>ACCELERATION</b>	Control the acceleration.	
	Acceleration: <code>#xx_ALPHAZ,</code> <code>#xx_ACCELX,</code> <code>#xx_ACCELY,</code> <code>#xx_ACCELZ</code>	Angle and position <code>xx_THETAZ,</code> <code>xx_SHIFTX,</code> <code>xx_SHIFTY,</code> <code>xx_SHIFTZ</code> Speed: <code>xx_OMEGAZ,</code> <code>xx_SPEEDX,</code> <code>xx_SPEEDY,</code> <code>xx_SPEEDZ</code>

Limits on motion can also be imposed by the command script. For example, in acceleration control the minimum and maximum allowable positions can be set using variables such as `#xx_SHIFTZMIN` and `#xx_SHIFTZMAX`. When the moving part reaches a limit, the corresponding speed is set to zero. If the direction of the acceleration is away from the limit, motion can continue. When developing a model, it is a good idea to set limits to prevent the moving part going outside the gap region.

## Units

The units used in motion control are as follows:

- in linear motion, the length unit in values of position, speed and acceleration is the length unit chosen for when the database was created;
- in rotational motion, the angular unit in values of position, speed and acceleration is radian;
- the unit of time is always second.

The system variable PI is available for use in expressions, for example to convert speeds in RPM to radians/s.

## Examples

- Simple linear motion with position control. The group with name MO... oscillates at 120Hz:
 

```
$STRING MO_MOTIONCONTROL POSITION
$constant #MO_SHIFTX 0
$constant #MO_SHIFTY 0
$constant #MO_SHIFTZ 0.001*sind(time*360*120)
```
- Linear motion with acceleration control. The acceleration is calculated from the force and the user defined mass of the complete model:
 

```
$STRING MO_MOTIONCONTROL ACCELERATION
$constant #MO_MASS 7.764571*8.96
$constant #MO_ACCELX 0
$constant #MO_ACCELY 0
$constant #MO_ACCELZ MO_FORCEZ/#MO_MASS
```

## Additional integrands

During the time integration process, the program can integrate other quantities specified in the **comi** file. If any user variables exist with names of the form **#DDT#xxx** (where **xxx** can be anything), the program will use the values of **#DDT#xxx** as time derivatives of **#xxx** and will integrate them to calculate **#xxx** in the same way as it integrates acceleration to calculate speed.

# Charged Particle Solver

## Introduction

The Charged Particle solver can be used to compute electrostatic fields in three dimensions, including the effects caused by space charge in beams of charged particles. Secondary particles produced as a result of collisions can be included in the calculation as well as the effects of the magnetic fields created by the charged particle beam. The program incorporates state of the art algorithms for the calculation of electromagnetic fields, advanced finite element and nonlinear equation numerical analysis procedures. In the following sections the algorithms used in the Charged Particle solver are described so that users are able to relate the finite element method to their model.

## The Charged Particle Algorithm

The Charged Particle solver uses the finite element method to solve the electrostatic Poisson's equation, and calculate the electric scalar potential (see [Scalar Potential Formulation \[page 602\]](#)).

### Electric and magnetic fields

The space charge density, included in the Poisson's equation solution, is found by calculating the trajectories of a set of charged particles from the emitters under the influence of the electrostatic field and magnetic fields. Magnetic fields can be supplied in several ways:

1. Values of magnetic flux density,  $\mathbf{B}$ , can be added to the database using [The TABLE Command \[page 817\]](#) in the Post-Processor.  $\mathbf{B}$  will always be used if it exists.
2. Values of magnetic field strength,  $\mathbf{H}$ , can be added to the database using [The TABLE Command \[page 817\]](#) in the Post-Processor.  $\mathbf{H}$  will only be used if  $\mathbf{B}$  does not exist.
3. Values of magnetic field strength,  $\mathbf{H}_s$ , can be calculated from any conductors and/or external field defined in the model. Calculated values of  $\mathbf{H}_s$  will only be used if neither  $\mathbf{B}$  nor  $\mathbf{H}$  exists in the database.
4. Values of magnetic field strength,  $\mathbf{H}_s$ , can be added to the database using [The TABLE Command \[page 817\]](#) in the Post-Processor.  $\mathbf{H}_s$  will only be used if neither  $\mathbf{B}$  nor  $\mathbf{H}$  exists and there are no conductors or external field defined.

Optionally, the magnetic field from the beam itself (see [Magnetic Field of Particle Beam \[page 568\]](#)) will be added to  $\mathbf{B}$ ,  $\mathbf{H}$  or  $\mathbf{H}_s$ .

### Particle trajectories

The particle trajectory calculations include full relativistic correction. The characteristics of the emitter are used to determine a current that is associated with the trajectory from a point on the emitter surface, the trajectory and its current are referred to as a beamlet. These are intersected

with the finite element mesh in order to allocate the space charge in the beamlet to the nodes of the mesh.

The charged particle trajectory calculation is by default performed in free space (**AIR**) volumes and the particles are stopped if they touch non-**AIR** volumes. The additional label **BEAMSTOP** can be applied to **AIR** volumes through which the trajectories should not pass, similarly the additional label **BEAMPASS** can be applied to material volumes through which trajectories should pass.

## Space charge iterations

The beamlet space charge affects the electric field distribution and may also affect the current in the beamlets. A consistent voltage, current and space charge distribution is required and this is calculated by an iteration that updates the space charge distribution, recalculates the electric fields and then the beamlet trajectories. The total electric field can be represented as the superposition of the electric fields from the electrodes and the space charge; these fields tend to cancel close to an emitter surface and the iteration that updates the space charge distribution is therefore liable to oscillate rather than converge to a consistent solution. The solver determines the sensitivity of the electric fields to the beamlet currents at the beginning of the second update iteration and uses this information to calculate an under relaxation factor that will avoid oscillations. This factor is applied to the change in space charge density at subsequent update iterations. The relaxation factor can also be specified, but the program's calculated value will be used if it is smaller.

## Emission models

Different types of particle emission models can be used to create the primary beams of charged particles. These include Child's law and Langmuir/Fry relationships for the calculation of the space charge limited current from a thermionic emitter, field effect emission relationships, emission from the surfaces of plasmas and beams with defined current densities and initial energy. Several emitter surfaces can be specified in a model and any number of emission models can be added to each surface. The emission model defines the type of particle and emission characteristics that are required.

## Thermal Emission Models

### Thermal saturation limit - type 0

The thermal saturation limit assumes that the electron current emitted by an electrode is independent of the applied voltage, and that it depends only on the temperature, work function and emission constant of the material. The current density is given by the Richardson-Dushman law

$$j_e = AT^2 e^{-\frac{q\phi_w}{kT}} \quad (6.1)$$

where  $A$  is the emission constant for the surface in amp cm<sup>-2</sup> K<sup>-2</sup>,  $\phi_w$  is the work function of the cathode material in volts,  $q_e$  is the electronic charge in coulombs,  $k$  is the Boltzmann constant ("Constants" on page 669) and  $T$  is the temperature of the cathode in kelvin.

The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution.

### Child's law current limit - type 2

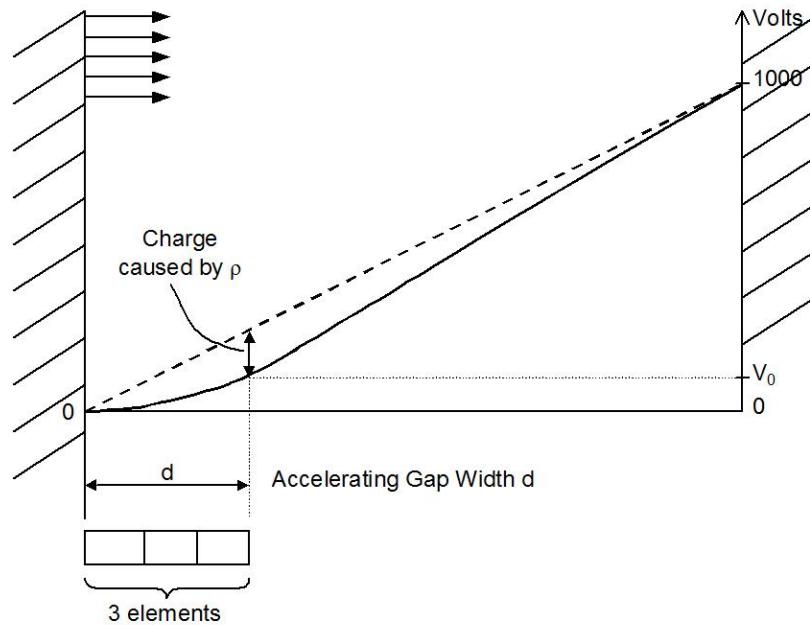
Child's law gives the maximum current density that can be carried in a beam of charged particles across a one dimensional accelerating gap. The equation is derived by requiring equilibrium of the charged particles with a self-consistent space charge field.

In order to apply this equation within the program, an accelerating gap width  $d$  must be supplied, see figure Figure 6.1. The equation only applies to infinite planar emitters. It is assumed that the radius of curvature of the emitting surface is large compared to the dimension  $d$  and that therefore the one dimensional solution can be used.

The equation for the space charge limited current is

$$j_e = \frac{4\epsilon_0}{9} \sqrt{\frac{2q}{m_0}} \frac{V_0^{3/2}}{d^2} \quad (6.2)$$

where  $\epsilon_0$  is the permittivity of free space,  $q$  is the charge on the particle in coulombs,  $m_0$  is the particle rest mass in kilograms and  $V_0$  is the accelerating voltage applied to the accelerating gap  $d$ .



*Figure 6.1 Potential distribution between two electrodes for a Child's Law emitter. The emitter on the left-hand side has a potential of 0 volt. The other electrode on the right-hand side has a potential of 1000 volt.*

When this model of the emitter is used in the space charge solution program, the accelerating voltage is determined by calculating the voltage at a distance  $d$  normal to emitter surface, from the finite element solution.

The distance  $d$  should be such that two or three elements are included between the sample point and the surface; it should also be small compared to the radius of curvature of the surface.

The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution. The emission constant, work function and temperature must be given, and if the Child's law current density exceeds the thermal saturation limit, the current density will be limited to the saturation value.

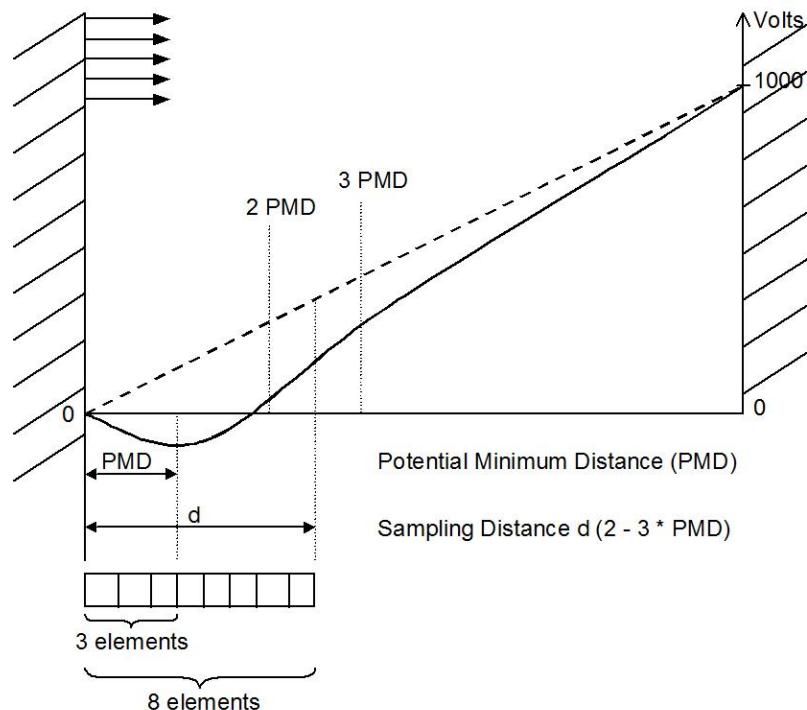
For both Child's law and Langmuir/Fry emitters (types 1, 2 and 8) the current from the emitter is equal to the current actually leaving the emitter. Any test beamlet (particle) that is returned to the emitter by the potential barrier is not included in the total current from the emitter that the program prints out in the **res** file.

### Langmuir/Fry current limit - types 1 and 8

Child's law assumes zero initial energy particles. A more realistic one dimensional solution can be found taking into account the velocity distribution of particles in a thermionic emitter.

In many cases thermionic emitters are operated in a space charge limited mode. This produces a uniform current distribution because the flow is insensitive to local variations of the surface emissivity. The initial particle energy distribution is important in this case.

A non-linear differential equation must be solved in order to calculate the space charge limited current (Reference: Kirstein, Kino & Waters, Space Charge Flow, McGraw-Hill, pp265-276), the program solves the non-linear equation using a shooting method.



*Figure 6.2 Potential distribution between two electrodes. The emitter on the left-hand side has a potential of 0 volt. The other electrode on the right-hand side has a potential of 1000 volt. The space charge in front of the emitter is generating a potential barrier.*

As in the Child's law model, a sample distance  $d$  must be specified, see figure Figure 6.2. The sampling distance should be 2 to 3 times the value of the potential minimum distance.

The one dimensional Langmuir/Fry model is then solved using the voltage computed at the sample point.

Two options are supported with the Langmuir/Fry emission model. The particles are either tracked from the specified emission surface (type 8) or from a virtual cathode (type 1).

The virtual cathode is defined by a set of positions on the normals from points on the emission surface, where the voltage is equal to the value at the emission surface. This assumes that the space

charge is creating a voltage minimum in front of the emitter, if the voltage minimum does not exist then particles are tracked from the emitter surface.

It should be noted that the sample distance  $d$  must be larger than the spacing between the emission surface and the virtual cathode.

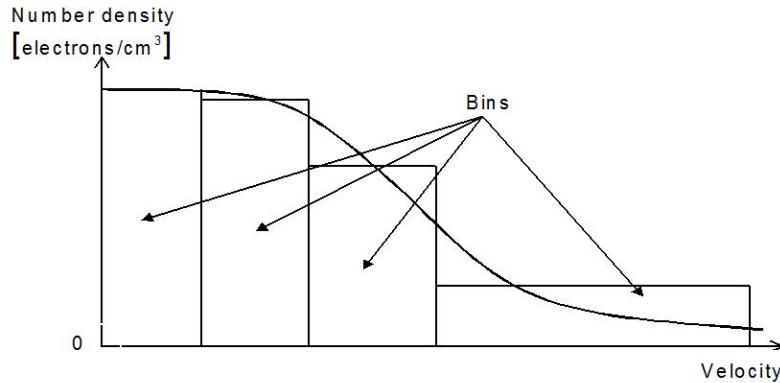
The initial velocity of the particles is computed from the displaced Maxwellian distribution of the particles that escape the voltage minimum that exists in front of the emitter. With the virtual cathode the number of sample beamlets can be specified. A single sample will have the mean velocity of the Maxwellian distribution; multiple samples will randomly sample the normal and tangential velocity distributions.

For both Child's law and Langmuir/Fry emitters (types 1, 2 and 8) the current from the emitter is equal to the current actually leaving the emitter. Any test beamlet (particle) that is returned to the emitter by the potential barrier is not included in the total current from the emitter that the program prints out in the **res** file.

### Maxwell velocity distribution sampling – type 10, 11 and 12

These options sample the velocity distribution of the electrons escaping from a thermionic emitter. Normal and tangential velocities may be sampled in the current release of the software.

With emitter types 10 and 11, the Maxwellian velocity distribution is divided into a set of ranges (also called bins) such that each range contains the same number of particles. Figure [Figure 6.3](#) shows an example of a velocity distribution with 4 bins. A test ray is tracked from each velocity bin, its velocity is determined by the average kinetic energy of the particles in the bin.



*Figure 6.3 Maxwell velocity distribution with 4 ranges*

Emitter type 10 uses only a normal velocity sampling; emitter type 11 uses both normal and tangential velocity sampling (the Maxwell distributions for the normal and tangential directions are independent).

Emitter type 12 uses a randomly selected set of test rays. The set is divided into two equal groups: one group can escape any potential minimum in front of the emitter; the second group cannot escape. Both the directions and energies of the sample rays are randomly selected.

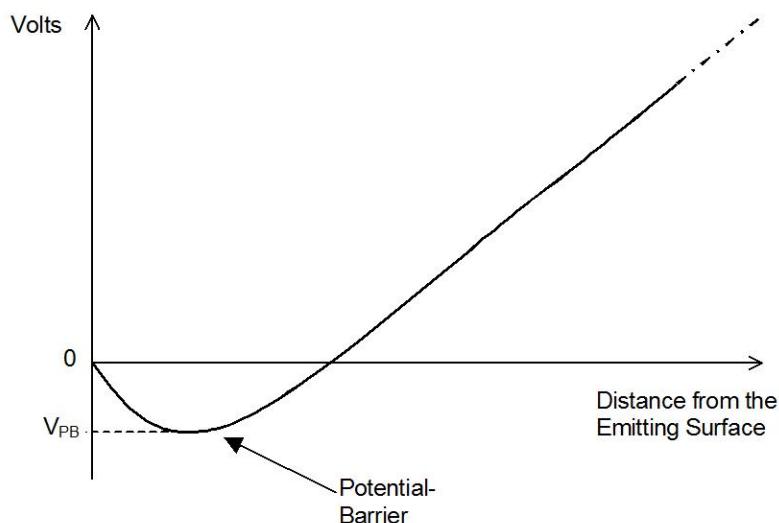
The current density of electrons at a particular velocity ( $v$ ) is  $j_e n(v)$  where

$$j_e = AT^2 e^{-\frac{q\phi_w}{kT}} \quad (6.3)$$

and the normalised distribution function in terms of velocity is

$$n(v) = 2 \sqrt{\frac{m}{2\pi kT}} e^{-\frac{mv^2}{2kT}} \quad (6.4)$$

Depending on the specific geometry of a finite element model some space charge may build up in front of an emitter. In this case the cathode's current is "space charge limited"; there will be a potential barrier in front of the cathode created by the space charge, see figure [Figure 6.4](#).

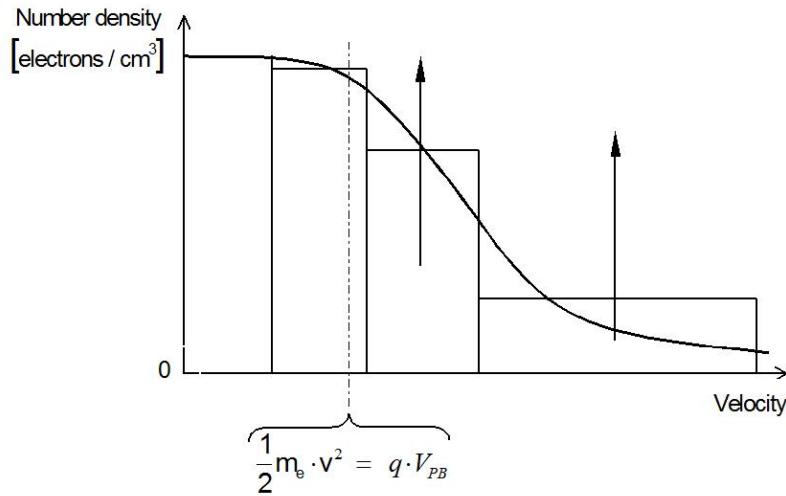


*Figure 6.4 Potential barrier in front of an emitter*

In order to leave the emitter surface, an electron needs an initial velocity (kinetic energy) greater than that required to pass over the potential barrier.

If the current is space charge limited, the finite element mesh close to the emitter surface must be capable of modelling the voltage minimum that may occur.

With a given value of the potential barrier  $V_{PB}$  the escape-velocity can be worked out, see figure [Figure 6.5](#). In the given example four test trajectories are being emitted per emitting point, but only two of these are capable of passing the potential barrier.



*Figure 6.5 Two trajectories with energy greater than  $V_{PB}$  are leaving the example-emitter of figure Figure 6.3*

A test ray from a discrete normal velocity bin can escape the potential barrier only if

$$v_{normal} > \frac{2 \cdot q \cdot V_{PB}}{m_e} \quad (6.5)$$

Only the normal component of the velocity determines whether a particle escapes. Similarly, the potential barrier has no affect on the tangential velocity component for emitter type 11.

### Example for emitter type 11

If the user requests a normal sampling = 4 and a tangential sampling = 2, the Maxwell velocity distribution will give  $4 \cdot 2 \cdot 2 \cdot 4 = 64$  test-rays per emitting point, but only 32 of these can escape the potential barrier in figure Figure 6.5.

The factors above are based on

- 4 normal samplings for the example in figure Figure 6.3,
- 2 beams tangential (in x direction),
- 2 beams tangential (in y direction) and
- 4 combinations (x+y, x-y, -x+y, -x-y).

The resolution of the emission current is defined by the equation

$$Resolution = \frac{I_{Saturation}}{\text{number of } v_{normal} \text{ bins}} \quad (6.6)$$

For a full nonlinear solution a sampling of the normal velocity should be expected to require more than 10 bins.

## Field Effect Emission Models

### Fowler Nordheim field emission - type 4

This option calculates the current that will be extracted by very high electric fields applied to the surface of the cathode. Significant tunnelling of electrons close to the surface of a cold metallic cathode will occur if a high electric field is applied to materials with a low thermionic work function. Fowler and Nordheim (1928) derived a rigorous solution for this tunnelling current:

$$J_e = 6.2 \times 10^2 \left( \frac{E_f}{\phi_w} \right)^{1/2} \frac{E^2}{E_f + \phi_w} e^{-\frac{6.83 \times 10^7 \phi_w^{3/2}}{E}} \quad (6.7)$$

where  $J_e$  is the current density in  $\text{A cm}^{-2}$ ,  $E_f$  is the Fermi energy of electrons in the metal and  $E$  is the electric field in  $\text{V m}^{-1}$  applied to the metallic surface. Extensions to the basic theory to include the change in barrier voltage produced by the escaping electrons give the following result:

$$J_e = A \left( \frac{hq_e E}{4\pi k} \right)^2 \cdot \frac{1}{2q_e m_0 \phi_w} e^{\frac{4\pi b(2q_e m_0)^{1/2} \phi_w^{3/2}}{Eh}} \quad (6.8)$$

where  $A$  is the emission constant for the surface in  $\text{amp cm}^{-2} \text{K}^{-2}$ ,  $h$  is the Planck constant,  $k$  the Boltzmann constant, and:

$$b = -\frac{2(1-\chi^{1.69})}{3(1+0.1107\chi^{1.33})} \quad (6.9)$$

and

$$\chi = \frac{(qE)^{1/2}}{\phi_w (4\pi\epsilon_0)^{1/2}} \quad (6.10)$$

The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution.

### Schottky field emission - type 5

This option calculates the current that will be emitted from a cathode, at a known temperature, for the lowest field strengths at which field emission occurs. The current density ( $J_s$  in  $\text{Amps/cm}^2$ ) is given by

$$J_s = AT^2 e^{-\frac{q_e \phi_w + (q_e^3 E)^{1/2}}{kT}} \quad (6.11)$$

where  $A$  is the emission constant for the surface in  $\text{amp cm}^{-2} \text{K}^{-2}$ . The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution.

## Extended Schottky field emission - type 6

At intermediate fields, the extended Schottky model predicts higher and more realistic currents. In this case the current density is given by

$$J_e = J_s \frac{F}{\sin(F)} \quad (6.12)$$

where

$$F = \frac{hq_e^{1/4}E^{3/4}}{(2m)^{1/2}kT} \quad (6.13)$$

and where  $J_s$  is given by equation (6.11)

The initial velocity of the particles is assumed to be the mean of the thermal velocity Maxwellian distribution.

## Automatic field emission selection - type 7

Three regimes of field emission are available in the software, the choice depends on the magnitude of the electric field at the start point of each particle. Users may either select a particular model, as shown above, or allow the program to determine which model is appropriate (type 7). The program will select the model that gives the largest current density.

### Field enhancement factor

The value of electric field,  $E$ , computed and used in the expressions for the field effect emissions (emitter types 4, 5, 6 and 7) may be multiplied by a user defined field enhancement factor. This allows the user to simply modify the emission model without re-specifying all the emission surface parameters. The default value is 1.0 if not specified. See "Record 3" on page 570.

## User Defined Emission Models

### User defined - type 9

This option allows users to define their own emission model using the electric field calculated at the surface of an emitter. User defined parameter variables are specified for the emission current density and initial particle velocity. These quantities must be evaluated in the unit set selected for the analysis (for example, if the model is defined in SI-micron units the user's current density parameter must be evaluated in Amps/micron<sup>2</sup> and the initial velocity in microns/second).

The following system variables are created for the point on the emitter surface:

<b>System variable</b>	<b>Units</b>	<b>Description</b>
<b>REX</b>	electric field units	Electric Field X component
<b>REY</b>	electric field units	Electric Field Y component
<b>REZ</b>	electric field units	Electric Field Z component
<b>NX</b>	None	Surface normal X component
<b>NY</b>	None	Surface normal Y component
<b>NZ</b>	None	Surface normal Z component

Emitter data from the emitter file is also made available as system variables:

<b>System variable</b>	<b>Units</b>	<b>Description</b>
<b>PMASS</b>	kg	Particle mass
<b>PCHARGE</b>	electronic charge units	Particle charge (e.g. proton = +1)
<b>PWORK</b>	electron volt	Emitter work function
<b>PTEMP</b>	kelvin	Emitter temperature
<b>PCONST</b>	current density units	Emission constant of the surface

The default unit set for an OPERA model is determined from its length unit. The units of the system variables described above are related to the length unit as follows:

<b>Model Unit</b>	<b>Electric Field</b>	<b>Current density</b>	<b>Velocity</b>
<b>SI (Metre)</b>	volt metre <sup>-1</sup>	amp metre <sup>-2</sup>	metre s <sup>-1</sup>
<b>CGS (Centimetre)</b>	volt cm <sup>-1</sup>	amp cm <sup>-2</sup>	cm s <sup>-1</sup>
<b>SI (Millimetre)</b>	volt metre <sup>-1</sup>	amp mm <sup>-2</sup>	mm s <sup>-1</sup>
<b>SI (Microns)</b>	volt metre <sup>-1</sup>	amp micron <sup>-2</sup>	micron s <sup>-1</sup>
<b>SI (Inch)</b>	volt metre <sup>-1</sup>	amp inch <sup>-2</sup>	inch s <sup>-1</sup>

## Specified current and density - types 3 and 20

The program has two options:

- **type 3:** The current density and initial particle energy must be specified. The sign of the current density should match the sign of the charge on the particles.
- **type 20:** The total current, initial particle position and initial particle velocity to be given.

## Plasma Emission

### Plasma free-surface (specified position) - type 102

The extraction of charged particle beams from plasmas can be modelled. The emission boundary of the plasma is defined as the surface on which the normal component of the electric field strength is constant. A single point on the emitter surface is kept fixed and the emission surface shape is adjusted to achieve the constant field condition. The fixed point on the surface is either specified (for emitters defined using a surface label, [See "Record 8.4" on page 579.](#)) or it is the first point defined on the emitter surface. Child's law is used to compute the current density that can be extracted from each point on the plasma emission boundary, a current saturation limit is calculated from the temperature and emission constant of the surface.

### Plasma free-surface (specified current) - type 103

In this model a current density, relative meniscus voltage and a temperature are specified; the program finds the position of the plasma meniscus (the surface bounding the quasi-neutral plasma). Space charge associated with the ion current leaving the plasma determines the meniscus position. A self-consistent space charge limited current flow forms beyond the meniscus and the program adjusts the meniscus position until the plasma's specified current density equals the self-consistent space charge limited current density.

This Plasma free-surface emitter should be specified on a surface inside the plasma chamber, emitting towards the extraction aperture. The surface should have an assigned voltage equal to the expected plasma potential. The plasma meniscus will be formed in front of this surface (in the emission direction) and therefore the emitter surface must not be placed too close to the extraction aperture.

The plasma's Bohm current density is typically specified for this emitter.

The Bohm current density of a plasma is derived by considering a bulk neutral plasma surrounded by a quasi-neutral pre-sheath. Plasma electrons are assumed to be in thermal equilibrium with a Maxwell-Boltzmann velocity distribution; the temperature of the ions is assumed to be low, and therefore their thermal velocity is neglected.

The model assumes that the plasma potential falls slightly in the pre-sheath region and ions in the pre-sheath are accelerated towards the meniscus. The region outside the meniscus is called the

sheath; here the voltage begins to change quickly as the ions are accelerated and the electron density decreases.

A 1-dimensional model of the plasma boundary is considered, in which collisions and ionization in the sheath are neglected. The ion current density ( $J_i$ ) must be constant throughout the sheath

$$J_i = q_i n_i(s) v_i(s) = q_i n_i(0) v_i(0), \quad (6.14)$$

where  $q_i$  is the charge of the ion,  $n_i(s)$  is the ion number density as a function of position  $s$  measured from the meniscus. Using this equation, the ion number density can be expressed as a function of the voltage  $\varphi(s)$  at a position in the sheath:

$$n_i(s) = \frac{n_i(0) v_i(0)}{\sqrt{v_i(0)^2 + 2q_i \frac{(\varphi(0) - \varphi(s))}{m_i}}}. \quad (6.15)$$

The electron density in the sheath will vary with the voltage:

$$n_e(s) = n_e(0) e^{\frac{q_e(\varphi(0) - \varphi(s))}{kT_e}}, \quad (6.16)$$

where  $n_e(s)$  is the electron density as a function of position,  $q_e$  is the electronic charge and  $k$  is the Boltzmann constant.

The voltage as a function of position in the sheath can be found from Poisson's equation, based on the electron and ion number densities from above:

$$\frac{d^2\varphi}{ds^2} = \frac{I}{\epsilon_0} \left( q_i n_i(s) + q_e n_e(s) \right). \quad (6.17)$$

For this equation to have a real solution,  $\varphi(s)$  must be monotonically decreasing through the sheath; this can only be true if

$$\frac{kT_e}{m_i v_i(0)^2} \leq I, \quad (6.18)$$

or rewriting this in terms of the ion velocity at the meniscus,

$$v_i(0)^2 \geq \frac{kT_e}{m_i}. \quad (6.19)$$

Equation (6.19) is known as the Bohm sheath criterion.

Based on this analysis of the plasma boundary, the minimum magnitude of the meniscus voltage,  $\varphi(0)$ , relative to the bulk plasma potential, is therefore

$$\varphi(0) = \frac{kT_e}{2q_e}. \quad (6.20)$$

The assumption that the pre-sheath is quasi-neutral implies that the electron and ion charge densities at the meniscus are approximately the same. Since the electrons are described by Maxwell-Boltzmann statistics, their density at the meniscus will be

$$n_e \begin{pmatrix} 0 \\ 0 \end{pmatrix} = n_e^{plasma} e^{-\frac{q_e \phi(0)}{kT_e}} = n_e^{plasma} e^{-\frac{I}{2}} . \quad (6.21)$$

The ion current density predicted by this model is therefore

$$J_i = q_i n_i^{plasma} e^{-\frac{I}{2}} \sqrt{\frac{kT_e}{m_i}} , \quad (6.22)$$

where  $n_e^{plasma}$  and  $n_i^{plasma}$  are the electron and ion densities in the neutral plasma.

Note that 3 quantities must be specified for this type of emitter:

- the ion current density (the sign of the current density must match the sign of the charge on the particles),
- the meniscus voltage relative to the plasma, and
- the ion temperature.

The ion velocity at the meniscus is determined by this temperature, rather than from the Bohm sheath criterion.

The Bohm analysis is approximate, although it has been found to agree well with measurements. The additional degree of freedom provided by being able to specify the temperature can be used to calibrate the model as required, if measurements are available.

To define a free surface plasma emitter that is exactly equivalent to the Bohm analysis, the emitter's temperature should be set to:

$$T = \frac{q_e T_e}{2 |q_e|} . \quad (6.23)$$

## Emitter Current

### Emitter types 1, 2 and 8

The current from the emitter is equal to the current actually leaving the emitter. Any test beamlet (particle) that is returned to the emitter by the potential barrier is not included in the total current from the emitter that the program reports.

### Emitter types 10, 11 and 12

The total emitted current is printed without subtracting particles that return to the emitter. It is necessary to use [The VIEW Command \[page 840\]](#) in the Post-Processor, with **PLOT=INTERSECTION**, to calculate the current in the beam.

## Secondary Emission

Secondary emission properties can be applied to labelled surfaces of the model. Collisions of the particle beamlets with these labelled surfaces are detected and secondary particles are introduced. These secondaries may also collide to produce further new secondary particles; the maximum number of generations of secondary particles can be limited. The space charge effects created by secondary particles can be excluded from the calculation.

The net power arriving at a secondary emitter surface (i.e the difference between the incident power and the power in secondary beams) is calculated and is available in the Post-Processor as system variable **BEAMPOWER**. The incident beam power can be determined by using a backscattered secondary with the energy loss factor set to a very small number, e.g.  $10^{-6}$  (see [Backscattered \[page 562\]](#)).

Many secondary emission properties can be applied to a labelled surface. Each secondary property is linked to the type of input particle and it defines the type of secondary particle that will be produced. For example, if a beam of protons collides with a carbon material surface three secondary emitters could be applied to input protons:

- elastic backscattered protons,
- true secondary electrons and
- secondary carbon ions (of a given charge state).

The secondary emitters all produce outputs that depend on the input particle flux. The energy and orientation of the input particle is available so that advanced secondary emission characteristics can be defined using Opera tabulated functions or algebraic expressions.

Four basic types of secondary emitter are provided, two basic options that enable a minimum specification and two advanced options that allow complete flexibility.

### Backscattered

The input particles are almost elastically scattered (reflected) from the secondary emission surface. The yield,  $\gamma$ , of backscattered particles and an energy loss factor,  $\eta$ , are specified for this model and one output beamlet is created for each input beamlet.

The **Yield** and **Energy loss factor** are given by

$$\gamma = \frac{I_o}{I_i} \quad (6.24)$$

and

$$\eta = \frac{E_o}{E_i} \quad (6.25)$$

where  $I$  is current,  $E$  energy and the subscripts  $i$  and  $o$  indicate input and output respectively. The values of  $\gamma$  and  $\eta$  will typically be in the range 0 to 1.

## True secondary

The scattering fraction of the output particles varies as the cosine of the polar angle measured from the surface normal. The input particle direction has no effect on the distribution of secondaries and only the average energy of secondaries is defined.

## Advanced backscattered

This emitter allows control of the energy and angular distribution of backscattered particles. The scattering fraction  $\gamma$  [1/(eV sr)] can be specified as a function of output energy and direction of the backscattered particles, together with the energy and angle of incidence of the primary particle. The total yield of backscattered particles is given by

$$\delta_s = \int_{E_I}^{E_2} \int_{\theta_I}^{\theta_2} \int_{\phi_I}^{\phi_2} \gamma(E, \theta, \phi) \sin(\theta) dE d\theta d\phi \quad (6.26)$$

The default limits for the integration are

$$\begin{aligned} 0 < E < E_{in} \\ 0 < \theta < \frac{\pi}{2} \\ 0 < \phi < 2\pi \end{aligned} \quad (6.27)$$

Other limits can be given in order to produce an accurate sampling of distributions that are narrowly directed. A randomly generated set of particles is produced that samples the distribution specified, within the limits specified.

## Advanced secondary

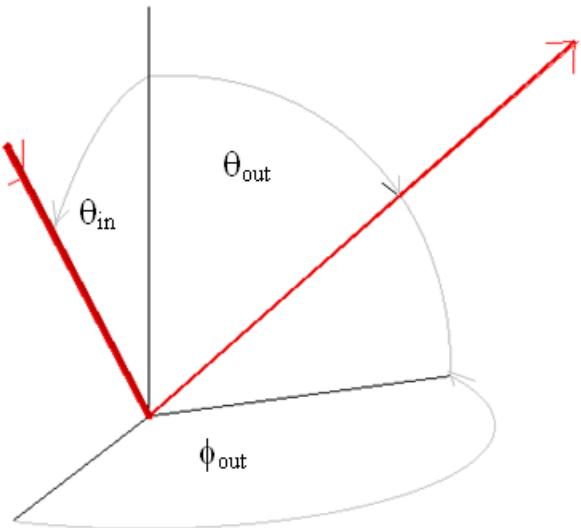
The scattering fraction as a function of output energy, polar angle and azimuthal angle can be defined with this type of secondary emission. This function is integrated with respect to output energy ( $E$ ), polar angle ( $\theta$ ) and azimuthal angle ( $\phi$ ) to determine the total yield ( $\delta_s$ ) (6.26).

A set of randomly generated particles are distributed over the ranges of  $E$ ,  $\theta$  and  $\phi$ . Upper and lower limits can be applied to each of these ranges.

## Secondary Emission System Variables

System variables are maintained by the solver so that they can be used for calculating the parameters required to describe the secondary emission. These variables include information about the incident particles, for example their energy and direction, together with other with similar data for the output particles.

- [Figure 6.6](#) shows the local coordinate system and variables that are created at the point where a beamlet hits a surface.
- [Table 61 on the next page](#) shows the incident particle system variables.
- [Table 62 on page 565](#) shows the secondary particle system variables.
- [Table 63 on page 565](#) shows the variables that describe the point of incidence.



*Figure 6.6 Scala track hitting a surface. The thick red line shows the input beamlet and a typical output beamlet is shown as a thin red line*

**Table 61: Incident particle system variables**

Name	Units	Description
INMASS	electron mass units	Incident particle mass
INCHARGE	electronic charge units	Incident particle charge
INENERGY	electron volt	Incident particle energy
INCURRENT	amp	Incident particle current
INTHETA	degrees	Incident particle angle to the outward normal from the surface
INVELX	cm/s	Global X component of the incident particle velocity
INVELY	cm/s	Global Y component of the incident particle velocity
INVELZ	cm/s	Global Z component of the incident particle velocity

**Table 62: Secondary particle system variables**

Name	Units	Description
OUTMASS	electron mass units	Secondary particle mass
OUTCHARGE	electronic charge units	Secondary particle charge
OUTENERGY	electron volt	Secondary particle energy
OUTTHETA	degrees	Secondary particle polar angle - from the outward normal to the surface
OUTPHI	degrees	Secondary particle azimuthal angle - measured from the incident direction

**Table 63: Incident point system variables**

Name	Units	Description
NX	None	X component of surface normal at point of incidence
NY	None	Y component of surface normal
NZ	None	Z component of surface normal

## Beam Loss and Volume Interactions

In the Charged Particle solver, a beamlet consists of the trajectory of a representative particle and a current. The current is usually calculated from the part of an emitting surface that the beamlet is associated with. Beamlets can interact with volumes they pass through in two ways, using volume emitters<sup>1</sup> that can be defined for backscattering and true secondary emission.

- The intensity and energy of the beamlets is affected by the parameters specified for a volume **Backscattered** emitter, but the particles are not deflected by the scattering events.
- Secondary particles can be produced by both **True Secondary** and **Advanced True Secondary** emitters attached to volumes.

Cell properties include a **Volume data label**. Volume emitters can be attached to these labels (the labels available are shown in the left hand panel of the **Set Emitter Properties** dialog).

### Backscattered emission

The rate of change of the current and energy of the particles, in beamlets containing particles of the specified type, is determined by the **Yield** and **Energy loss factor**. The rate of change of current is given in amp/cm by

---

<sup>1</sup>The use of volume emitters is a separately licensed feature.

$$\frac{dI}{ds} = I\gamma \quad (6.28)$$

where  $\gamma$  is the specified yield, and the rate of change of energy in eV/cm by

$$\frac{dE}{ds} = E\eta \quad (6.29)$$

where  $\eta$  is the specified energy loss factor. The values of current,  $I$ , and energy,  $E$ , are those of a particular position in the beam and  $s$  is path length along the trajectory. The values of  $\gamma$  and  $\eta$  will typically be small and negative.

### True secondary emission

Volume **True Secondary** emission produces a number of particles of a specified energy, randomly distributed in direction. A true secondary emitter attached to a volume will create a secondary particle yield per cm length of the primary particle trajectory as the primary particle passes through the emitter volumes. The yield is given in 1/cm by:

$$Y_v = N\gamma, \quad (6.30)$$

where  $N$  is the linear density of particles in the primary beam and  $\gamma$  is the specified yield per cm of secondary particles. Secondary events are randomly generated along the primary trajectory, with a characteristic spacing related to the **Maximum trajectory step size**.

### Advanced true secondary emission

Volume **Advanced True Secondary** emission allows an expression to be specified that is used to determine the energy and direction distribution for the randomly generated secondary particles. Note that in this case the secondary particles are distributed over a range of energies and directions.

The total yield of secondary particles is determined by equation (6.26), where the polar angle is measured from the primary beam direction vector. Again, these secondary events are randomly generated along the primary trajectory, with a characteristic spacing related to the specified

**Maximum trajectory step size**. At each event position the specified number of secondary particle beamlets will be generated in randomly determined directions and with random energies, in a discrete approximation of the defined scattering fraction function.

## Volume Plasma Emitter

The volume plasma emitter uses a self-consistent plasma boundary method to model low density magnetised plasmas. The method is described as self-consistent because it determines the plasma boundary surface shape, together with the ion and electron currents that are consistent with the potential distribution caused by the space charge in the particle flows outside the plasma volume. Ions are not explicitly represented inside the plasma volume and the plasma is assumed to be quasi neutral.

It is assumed that high energy electrons accelerated through the plasma potential are the dominant source of gas ionisation that forms the plasma. The high energy electrons may start as secondary

electrons created by plasma ion collisions with a target surface, they are then accelerated and enter the plasma, causing ionisation and excitation of the gas that forms the plasma.

The volume distribution of the rate of gas ionisation is found by tracking the secondary electrons and evaluating their ionisation yield as a function of position. An approximate diffusion solution is then used to calculate the position dependent ion density at the plasma meniscus and thus determine the local Bohm current density.

The volume plasma emitter is normally used as part of closed cycle of linked secondary emitters. For example, a model will contain

- a volume plasma emitter that represents ionisation of argon gas by energetic electrons;
- a secondary surface emitter that creates electrons from colliding argon ions;
- a Volume backscatter emitter to model loss of energetic electrons.

The first and second emitters form the closed cycle of secondary emission:

- argon ions flow out from the plasma that is being formed by gas being ionised by energetic electrons;
- secondary electrons are created outside the plasma, as a result of a surface being bombarded by the argon ion current flowing from the plasma. The electrons are accelerated into the plasma where they produce ionisation.

The third emitter is also essential part of the cycle, it determines the volume distribution of the rate of ionisation and represents loss of high energy electrons. This rate of ionisation is used in the plasma emitter model to calculate the plasma density at the meniscus. More complicated physics can be included by adding more emitters, for example loss of current from the high energy group could be balanced by a new current of scattered lower energy particles.

The model is defined by the gas that is ionised to form the plasma and its ionisation state, together with the type of particle that causes ionisation. The model assumes that the quasi neutrality of the plasma is maintained by a low energy population of ionising particles (typically electrons) that are in thermal equilibrium, their temperature is required.

The operating voltage of a device using a volume plasma emitter will be determined by the voltage boundary conditions that have been applied to electrodes. In addition, the total plasma current flow must be specified for the model. A stable solution is achieved over a wide range of input parameters by using the total current to control the simulation. The model scales the total rate of ionisation in the plasma to be equal to the specified current and the scaling factor that is used is reported. The plasma will not be created in practise if the scaling factor is greater than one, a higher voltage or gas pressure would be needed. The current flow through the device would need to be limited by the external supply if the reported scaling factor is less than one. The voltage that the plasma rises to will be related to this scaling factor and if the reported scaling factor is much less than one the plasma voltage will become impractically high.

## Magnetic Field of Particle Beam

For applications which have intense beams of very fast charged particles, the forces from the magnetic fields, created by the motion particles themselves, becomes significant in the trajectory calculations. This option can be switched on using the **BEAMMAGFIELDS** parameter of [The ANALYSISDATA Command \[page 122\]](#). The solver also needs to know the symmetry of the model so that the fields from symmetry copies of the beams, implied by boundary conditions, are included in the calculations. In order for this to happen correctly, the model symmetry and boundary conditions must be supplied using [The BACKGROUND Command \[page 135\]](#) rather than by applying boundary conditions to labelled faces with [The BOUNDARY Command \[page 150\]](#).

## Emitter Data Files

The emitter surfaces can be defined using [The EMITTER Command \[page 206\]](#) or can be described in a data file with a filename extension **emit**, and the same filename as the model geometry file. The contents of an emitter data file are as follows:

- [Record 1 \[page 569\]](#): model symmetry
- [Record 2 \[page 569\]](#): global parameters, including number of emitters
- [Record 3 \[page 570\]](#): emitter characteristics, followed by:
  - [Record 3.1 \[page 572\]](#): only for emitter type 9
  - [Record 4 \[page 572\]](#): particle type
  - [Record 5 \[page 572\]](#): sample rays
  - [Record 6 \[page 573\]](#): number of face segments or beamlets
  - [Record 7 \[page 573\]](#): face geometry type
  - One or more of one of the following:
    - [Record 8.0 \[page 574\]](#): extruded line
    - [Record 8.1 \[page 575\]](#): 4 straight sided facet
    - [Record 8.2 \[page 576\]](#): 4 curved sided facet (8 points)
    - [Record 8.3 \[page 579\]](#): beamlet
    - [Record 8.4 \[page 579\]](#): labelled face

Records 3 to 8 are repeated as a group, for each emitter.

Records 7 and 8 are repeated as a group, for each facet or beamlet.

**N.B.** CGS units must be used in the **emit** data file.

The **emit** data file format is as follows (note that free format input is used and all data items must therefore be entered).

## Record 1

### Model symmetry.

This data specifies how a complete model should be created from the section that has been discretized. Where the model and field exhibit some symmetry it would be costly and unnecessary to discretize the complete space. The discretization should model the minimal symmetric part of the complete object, but the field must also share the same symmetry. For example, if the geometry and field have a plane of mirror image symmetry, then only half the complete model needs to be specified. The method used to add the space charge from the particle beams into the finite element model also takes advantage of symmetry, only the sections of the emitters inside the discretized space should be described in the *emit* data file. N.B. This definition of model symmetry must match the model symmetry defined in the database.

Field Number	Type	Units	Description
1	Integer	None	Order of rotational symmetry about the Z axis. If the sign is negative then the voltage changes sign in each section.
2	Integer	None	Reflection flag for XY plane. 0 = no reflection 1 = positive mirror image -1= inverted mirror image
3	Integer	None	Reflection flag for YZ plane. 0 = no reflection 1 = positive mirror image -1= inverted mirror image
4	Integer	None	Reflection flag for ZX plane. 0 = no reflection 1 = positive mirror image -1= inverted mirror image

## Record 2

### Global parameters.

A number of independent emitting surfaces may be defined in a model. Each surface has its own emission characteristics. Some parameters apply to all aspects of the calculation, for example, the maximum distance between the points used to represent each trajectory, the accuracy of the trajectory calculation and the distance used to evaluate the current limit models.

Field Number	Type	Units	Description
1	Integer	None	Number of Emitters

2	Real	cm	Maximum step length allowed in the trajectory calculation. Note that there is a limit of 200,000 steps per trajectory.
3	Real		Absolute tolerance for the trajectory calculation.
4	Real	cm	Normal sampling distance. The distance from the emitter surface used to sample the voltage for the Child's and Langmuir/Fry equations and the plasma free surface model.

Records 3 to 8 are then repeated as a group, Number of Emitter times.

### Record 3

#### **Emitter characteristics.**

The emission model is specified for each emitter, together with the data required to characterize the emitter.

Field Number	Type	Units	Description
1	Integer	None	Emitter type: 0 Thermal saturation limit 1 1D Langmuir/Fry limit (virtual cathode) 2 1D Child's law limit 3 Specified current density 4 Fowler Nordheim Field emission 5 Schottky Field emission 6 Extended Schottky field emission 7 Automatic selection between types 4, 5 and 6 8 1D Langmuir/Fry limit (specified surface) 9 User defined 10 Maxwell Normal Velocity sampling 11 Maxwell Normal and Tangential Velocity sampling 12 Maxwell Random Velocity sampling 20 Beamlets with defined position, velocity and current 102 Plasma free-surface (specified position) 103 Plasma free-surface (specified current)
2 <sup>a</sup>	Real	K	Emitter temperature
3 <sup>a</sup>	Real	volt	Emitter Work function (particle energy for emitter type 3)

<sup>a</sup>The value given for Field 2 has no effect with emitter type 3. The values given for Fields 2, 3 and 4 have no effect with emitter type 20. However, values (for example 0.0) must be provided.

Field Number	Type	Units	Description
4 <sup>a</sup>	Real	amp/(cm <sup>2</sup> K <sup>2</sup> )	Material's Emission constant ( $A$ in equation (6.1)) for types 0, 1, 2, 4, 5, 6, 10, 11 and 12. Current density for type 3 emitter.
5	Real	None	Field enhancement factor for field effect emitters types 4 through 7. If omitted, the default value 1.0 is used.

## Record 3.1

User defined emission - only required for emitter type 9.

Field Number	Type	Units	Description
1	Character	current density	Current density user variable name
2	Character	length/s	Initial velocity user variable name

## Record 4

Particle type.

Field Number	Type	Units	Description
1	Real	None	Particle rest mass in electron rest mass units.
2	Integer	None	Number of charge quanta carried by the particle (-1 for an electron).

## Record 5

**Sample rays.**

Enhanced models for the emitters have been developed. These include sampling of the velocity distribution of the input particles. Trajectories are calculated for a set of particles. In the curved face emitter model at least one particle is started from each subdivision panel of the face. The maximum distance parameter can be used to increase the number used.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Integer	None	Number of sample bins used to sample tangential velocity (only used with emitter type 11).
2	Integer	None	Number of sample bins used to sample normal velocity (only used with emitter types 10 and 11). Number of sample bins for random velocity sampling (emitter type 12).
3	Real	cm	Maximum distance (tangential to the emitter surface) between sample rays (not relevant for emitter type 20).

## Record 6

### **Number of face segments and local coordinate system.**

Each emitter can be modelled as a set of faces. The emitter is defined with respect to a local coordinate system (selected in order to orientate the face correctly).

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Integer	None	The number of Faces or Beamlets in the model for the emitter.
2	Real	cm	The global X coordinate of the origin of the local system
3	Real	cm	The global Y coordinate of the origin of the local system
4	Real	cm	The global Z coordinate of the origin of the local system
5	Real	degrees	The THETA Euler angle
6	Real	degrees	The PHI Euler angle
7	Real	degrees	The PSI Euler angle

Records 7 and 8 are repeated as a group, Number of Faces/beamlets times.

## Record 7

### **Face segment geometry type.**

Field Number	Type	Units	Description
1	Integer	None	Geometry type for the face segment 0 = Extruded curved line segment 1 = 4 point ruled quadrilateral facet 2 = 8 point curved quadrilateral facet 3 = A set of points and beamlet data (only valid for emitter type 20) 4 = A labelled face

A ruled surface is generated by a family of straight lines (for a description see, I D Faux and M J Pratt, *Computational Geometry for Design and Manufacture*, published by Ellis Horwood Ltd., 1979, ISBN 0-85312-114-1).

## Record 8.0

### Extruded curved line segment definition.

A line segment is defined by the X and Y coordinates of its end points in a local coordinate system XY plane. The line is extruded between the two local Z coordinates to create a face. The face is divided into facets as defined by the line and extrusion subdivision parameter. At least one sample particle is tracked from each facet, unless the facet size is greater than the maximum distance parameter. The subdivision and maximum distance parameters should be such that a reasonable number of sample rays are used, the spacing should be less than the element size.

The bias parameter on the line can be used to produce a non-uniform distribution of particles along the line. If the bias is less than 0.5 the particles will be biased towards the first point.

Field Number	Type	Units	Description
1	Real	cm	Starting X coordinate of the line.
2	Real	cm	Starting Y coordinate of the line.
3	Real	cm	Final X coordinate of the line.
4	Real	cm	Final Y coordinate of the line.
5	Real	1/cm	Curvature of the line (positive implies centre is to the right of the line from start to end).
6	Real	cm	Bias parameter (0.5 for uniform distribution).
7	Integer	None	Line subdivision, i.e. the number of segments the line will be divided into (for best results use line segments that correspond with region edges and have the same subdivision).

Field Number	Type	Units	Description
8	Real	cm	Starting Z coordinate of the extrusion.
9	Real	cm	Final Z coordinate of the extrusion.
10	Integer	None	Number of divisions in the extrusion direction.

## Record 8.1

### 4 point ruled facet definition.

The emitting surface may be modelled with an assembly of quadrilateral facets, defined by their four vertex coordinates. The edges of such facets are straight lines, and the surface is ruled. The facets may be degenerated to triangles by defining two consecutive, coincident points.

At least one sample particle is tracked from each sub-facet, unless the sub-facet size is greater than the maximum distance parameter ([Record 5 \[page 572\]](#)). The maximum distance parameters should be such that a reasonable number of sample rays are used. In general, with this type of emitter geometry, the sub-facets should be coincident with a facet of the finite element model, and the maximum distance parameter should be equal to or less than the finite element size. This will ensure that all finite elements at the emitter surface are intersected by particles that are used to sample the emission.

Field Number	Type	Units	Description
1	Integer	none	Division into sub-facets, along edges 1 and 3.
2	Integer	none	Division into sub-facets, along edges 2 and 4.

The facet is regularly subdivided into sub-facets, according to the above parameters. Edge 1 joins vertices 1 and 2; edge 2 joins vertices 2 and 3; edge 3 joins vertices 3 and 4; edge 4 joins vertices 4 and 1.

Note that the vertex ordering is important. Particles leave the surface in the direction of its outward normal, defined by a right-hand screw convention rotating in the direction of increasing vertex number.

Field Number	Type	Units	Description
1	Real	cm	X coordinate of the first vertex.
2	Real	cm	Y coordinate of the first vertex.
3	Real	cm	Z coordinate of the first vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the second vertex.
2	Real	cm	Y coordinate of the second vertex.
3	Real	cm	Z coordinate of the second vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the third vertex.
2	Real	cm	Y coordinate of the third vertex.
3	Real	cm	Z coordinate of the third vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the fourth vertex.
2	Real	cm	Y coordinate of the fourth vertex.
3	Real	cm	Z coordinate of the fourth vertex.

## Record 8.2

### 8 point ruled facet definition.

The emitting surface may be modelled with an assembly of quadrilateral facets, defined by their four vertex coordinates and four mid-side coordinates. The edges of such facets are curved, and the surface is curved. The shape of the facet is isoparametric, using a tri-quadratic mapping, equivalent to the surface shape obtained with a 20 node isoparametric brick finite element. The facets may be degenerated to triangles by defining two consecutive, coincident vertex points, that are also coincident with the mid-side point on the edge between them.

At least one sample particle is tracked from each sub-facet, unless the sub-facet size is greater than the maximum distance parameter ([Record 5 \[page 572\]](#)). The maximum distance parameters should be such that a reasonable number of sample rays are used. In general, with this type of emitter, the sub-facets should be coincident with facets of the finite element model, and the maximum distance parameter should be equal to or less than the finite element size. This will ensure that all finite elements at the emitter surface are intersected by particles that are used to sample the emission.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Integer	none	Division into sub-facets, along edges 1 and 3.
2	Integer	none	Division into sub-facets, along edges 2 and 4.

The facet is regularly subdivided into sub-facets, according to the above parameters. Edge 1 joins vertices 1 and 2; edge 2 joins vertices 2 and 3; edge 3 joins vertices 3 and 4; edge 4 joins vertices 4 and 1.

Note that the vertex ordering is important. Particles leave the surface in the direction of its outward normal, defined by a right-hand screw convention rotating in the direction of increasing vertex number.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the first vertex.
2	Real	cm	Y coordinate of the first vertex.
3	Real	cm	Z coordinate of the first vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the second vertex.
2	Real	cm	Y coordinate of the second vertex.
3	Real	cm	Z coordinate of the second vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the third vertex.
2	Real	cm	Y coordinate of the third vertex.
3	Real	cm	Z coordinate of the third vertex.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of the fourth vertex.

2	Real	cm	Y coordinate of the fourth vertex.
3	Real	cm	Z coordinate of the fourth vertex.

Field Number	Type	Units	Description
1	Real	cm	X coordinate of midside node between vertices 1 and 2.
2	Real	cm	Y coordinate of midside node between vertices 1 and 2.
3	Real	cm	Z coordinate of midside node between vertices 1 and 2.

Field Number	Type	Units	Description
1	Real	cm	X coordinate of midside node between vertices 2 and 3.
2	Real	cm	Y coordinate of midside node between vertices 2 and 3.
3	Real	cm	Z coordinate of midside node between vertices 2 and 3.

Field Number	Type	Units	Description
1	Real	cm	X coordinate of midside node between vertices 3 and 4.
2	Real	cm	Y coordinate of midside node between vertices 3 and 4.
3	Real	cm	Z coordinate of midside node between vertices 3 and 4.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	cm	X coordinate of midside node between vertices 4 and 1.
2	Real	cm	Y coordinate of midside node between vertices 4 and 1.
3	Real	cm	Z coordinate of midside node between vertices 4 and 1.

## Record 8.3

### Beamlet with defined direction and current.

This option can only be used with emitter type 20.

The data describes each beamlet that will be included in the calculation. The starting position, velocities and current in the beamlets all must be specified. Note that the beamlets are specified in the local coordinate system defined in [Record 6 \[page 573\]](#), this affects both the coordinates and the velocities.

<b>Field Number</b>	<b>Type</b>	<b>Units</b>	<b>Description</b>
1	Real	Amps	The current in the beamlet
2	Real	cm	The local X coordinate of the start of the beamlet.
3	Real	cm	The local Y coordinate of the start of the beamlet.
4	Real	cm	The local Z coordinate of the start of the beamlet.
5	Real	cm/sec	Initial local X velocity of the particles in the beamlet.
6	Real	cm/sec	Initial local Y velocity of the particles in the beamlet.
7	Real	cm/sec	Initial local Z velocity of the particles in the beamlet.

## Record 8.4

### Labelled face.

Additional labels may be added to a surface in the Modeller or Pre-Processor, these labelled surface (s) can be used as emitters. The solver will ensure that at least one trajectory leaves every element

on the surface, additional rays will be distributed over the element if the maximum distance between sample rays specified in [Record 5 \[page 572\]](#) is smaller than the element size.

- Faces on the outside of the mesh: particles will be emitted towards the interior of the mesh.
- Interior faces of the mesh: particles will be emitted towards volumes of **AIR** or volumes with the **BEAMPASS** label. If both directions are possible, the direction chosen will be the normal direction of the face. The direction can be reversed if the label is preceded by a minus (-). (The normal direction of surfaces can also be "toggled" in the Modeller to ensure that all surfaces with a particular label are emitting in the same direction. See [The VECTOR Command \[page 339\]](#).)

The fixed point on a Plasma free surface emitter (type 102) should be specified for the first labelled face of the emitter.

Field Number	Type	Units	Description
1	Character	none	Additional label on surface (Note: This type of emitter cannot be used for emitter type 20).
2	Real	cm	The global X coordinate of the fixed point on the surface of a plasma free surface emitter.
3	Real	cm	The global Y coordinate of the fixed point on the surface of a plasma free surface emitter.
4	Real	cm	The global Z coordinate of the fixed point on the surface of a plasma free surface emitter.

## Example Data Files

### Example 1: Extruded Line

An example *emit* data file defining a plane emitting surface (in an XY plane of the global coordinate system) which emits particles in the positive Z direction of the global coordinate system. The emitting surface is modelled using an extruded line.

1	0	1	0					
1	0.2	0.001	0.004					
1	1250.0	1.86	350.0					

1	0	1	0							
1.0	-1.0									
1	1	0.05								
1	0.0	5.15	0.001	90	-90	90				
0										
0.0	0.0	0.085	0.0	0.0	0.5	6	-0.5	0.5	8	

## Example 2: Curved Quadrilateral

An example *emit* data file defining an eight node curved quadrilateral facet as an emitter, the facet is triangular with points 1, 4 and 8 coincident.

4	0	0	1							
1	0.005	0.011	0.001E-5							
4	300.0	4.52	120.0							
1.0	-1.0									
1	1	0.4E-6								
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	2									
30	4									
0.0	0.0	5.01E-6								
2.505E-6	0.0	4.339E-6								
1.7713E-6	1.7713E-6	4.339E-6								
0.0	0.0	5.01E-6								
1.297E-6	0.0	4.8393E-6								
2.3144E-6	0.9586E-6	4.339E-6								
0.9171E-6	0.9171E-6	4.8393E-6								
0.0	0.0	5.01E-6								

### Example 3: Beamlets

An example **emit** data file defining a set of beamlets with specified current, position and velocity.

4	0	0	1				
1	0.005	0.011	0.001E-5				
20	0.0	0.0	0.0				
1.0	-1.0						
1	1	0.0					
4	0.0	0.0	0.0	0.0	0.0	0.0	
3							
1.0E-6	-0.1	-0.1	0.0	0.0	0.0	1.0E+6	
3							
1.0E-6	-0.1	0.1	0.0	0.0	0.0	1.0E+6	
3							
1.0E-6	0.1	0.1	0.0	0.0	0.0	1.0E+6	
3							
1.0E-6	0.1	-0.1	0.0	0.0	0.0	1.0E+6	

### Example 4: Labelled Surface

An example **emit** data file defining a labelled surface.

0	0	1	0				
1	0.05	0.05	0.01				
1	2600.0	4.25	120.0				
1.0	-1.0						
1	1	0.1					
1	0.0	0.0	0.0	0.0	0.0	0.0	
4							
LFEMIT							

## Example 5: User Defined Equation

The final example shows a user defined emission equation. The equations defining the emitted current density as a function of field intensity are specified as parameter functions in the Opera-3d Modeller or Pre-Processor. For example, the following command script could be specified to set up a Fowler Nordheim emitter as a user defined function:

```

/ Temporary set up of system variables
$sysvar rex 1 | $sysvar rey 1 | $sysvar rez 1
$sysvar nx 1 | $sysvar ny 0 | $sysvar nz 0
$sysvar pcharge -1
$sysvar pmass 1
$sysvar pwork 4
$sysvar pconst 120.0*1.0d-12
$sysvar ptemp 270
-----
/ Calculate Enormal in volts/m
$parameter #Enormal rex*nx+rey*ny+rez*nz
/ Pick up some essential physical constants
$CONST #E0 ELECTRONCHARGE*PCHARGE
$CONST #V0 ELECTRONEENERGY*PMASS
$CONST #M0 ELECTRONMASS*PMASS
$CONST #ETA -#E0/#M0

$parameter #PCONST PCONST*1.0E12
$parameter #F SQRT((#M0/BOLTZMANN) / (PTEMP*2))
$parameter #LAMDA PLANCK/SQRT(ABS(#E0)*2*#M0*PWORK)
$parameter #D ABS(#E0*#Enormal)*#LAMDA/(4*PI)
$parameter #Y SQRT(ABS(#E0*#Enormal)/(4*PI*EPSILON0))/PWORK
$parameter #BB (1-#Y^1.69)*2/((#Y^1.33*0.1107+1)*3)
$parameter #FNJ (#D/BOLTZMANN)^2*#PCONST*EXP(#BB*#E0*PWORK/#D)*1.0E4
/ #VELFN = Initial electron velocity in microns/second
$parameter #VELFN 2/(SQRT(PI)*#F)*1.0E6
/ #CURFN = Emitted current density in Amps/micron^2
$parameter #CURFN #FNJ*ABS(PCHARGE)/PCHARGE*1.0E-12

```

The user defined functions can then be referenced as shown in the following emit file:

1	0	1	1			
1	0.001	0.001	0.0			
9	300.0	4.27	120.0			
#CURFN	#VELFN					
1.0	-1.0					
1	1	0.1E-5				
1	0.0	0.0	0.0	0.0	0.0	0.0
	4					
FIELD						



# High Frequency Solvers

---

## Introduction

This section describes the Opera-3d solvers that can be used to compute high frequency electromagnetic fields in three dimensions. There are two solvers:

- Modal High Frequency calculates resonant modes of cavities;
- Harmonic High Frequency solves the wave equation at predefined frequencies.

The programs incorporate state of the art algorithms for the calculation of electromagnetic fields and advanced finite element numerical analysis procedures. In the following sections the algorithms used are described so that users are able to relate the finite element method to their model.

## The Modal HF Algorithm

The Modal High Frequency solver is designed to solve resonant cavity models containing lossless, isotropic dielectrics. The walls of a cavity are treated as perfect electrical conductors (**PEC**) which is a good approximation at high frequency.

The solver models the electric field directly using "edge" elements. In this approach the degrees of freedom associated with **E** are not its components directly, but the emf jump between connected nodes. The emf  $e_{\{ij\}}$  is defined as,

$$e_{(ij)} = \int_{I_{(ij)}} E \cdot dl \quad (6.31)$$

where  $i$  and  $j$  are two nodes in the mesh and  $I_{\{ij\}}$  is the *directed* edge connecting them. This is the origin of the term "edge" in edge elements. In some respects this approach is very similar to the method of Yee in Finite Difference Time Domain. The advantage of edge elements is that they model the important singular field behaviour around re-entrant corners accurately and in a very natural manner.

At high frequency the Electric field obeys the vector wave equation:

$$\nabla \times \frac{1}{\mu} \nabla \times E = -\frac{\partial^2(\epsilon E)}{\partial t^2} \quad (6.32)$$

with the following condition holding on **PEC** boundaries

$$E \times n = 0 \quad (6.33)$$

Further assuming all materials within the cavity are linear, ("High Frequency Solvers" above) reduces to the vector Helmholtz equation for each individual mode  $m$ ,

$$\nabla \times \frac{1}{\mu} \nabla \times E - \epsilon \omega_m^2 E = 0 \quad (6.34)$$

Applying the Galerkin method as described in [The Finite Element Method \[page 609\]](#), leads to a sparse generalized eigenvalue problem,

$$Ke = \omega_m^2 Me \quad (6.35)$$

from which the eigenvalues  $\omega_m$  and eigenvectors  $e$  can be determined.

The solver finds the eigenvalues above a given frequency, or within a specified range. In the Modeller or Pre-Processor the user selects which method is preferred, and supplies the required values.

The method to find a range of eigenvalues must be used with some care. As a rule of thumb the number of eigenvalues in a unit range about the  $n^{\text{th}}$  eigenvalue is proportional to  $n^3$  (this is a property of three dimensional systems). For example if the first two eigenvalues of a system are 1 and 2, then approximately 8 eigenvalues fall in the range 1-3 and 27 in the range 1-4.

The efficient calculation of eigenvalues depends on the use of direct solution methods, rather than the iterative solution methods used by default in other Opera-3d solvers. Optimal reordering methods are used to minimize the memory requirements of the direct solution method used.

It should be noted that the eigenvalue solver also stores the eigenvalues and vectors in memory, in addition to the finite element matrices and the matrix decomposition mentioned above. The memory required to store the eigenvectors depends on the number of equations multiplied by the number of eigenvalues.

The eigenvalue solver will typically be used to find relatively few eigenvalues (up to 10), it is efficient even for large numbers of eigenvalues but the additional storage required to store the eigenvectors may be limited by available memory. In this case multiple solutions may be used to obtain a large number of eigenvalues by specifying the frequency range for each solution.

## Boundary conditions

There are no sources included in a Modal HF model – it is not driven. Other than symmetry planes, the model must be enclosed by perfect electric walls, with no **RADIATION** conditions present. Symmetry may still be exploited in the usual way, as long as it implies that a closed volume is to be modelled.

## The Harmonic HF Algorithm

The Harmonic High Frequency solver uses edge elements to solve the wave equation in terms of the electric field, although for compatibility non-zero boundary conditions are still defined in terms of magnetic vector potential. The equation solved is identical to the equation for Modal HF except that the angular frequency  $\omega$  is defined by the user i.e. a deterministic solution of the equation is made.

$$\nabla \times \frac{i}{\mu} \nabla \times E - \epsilon \omega^2 E = 0 \quad (6.36)$$

Again, the assumption is that all materials are linear.

Under steady-state alternating excitation Harmonic HF uses a complex variable substitution. This carries the assumption that the materials have linear characteristics and that therefore the waveforms of the fields are precisely the same. The substitutions are:

$$\begin{aligned} E(t) &= E_c e^{i\omega t} \\ H(t) &= H_c e^{i\omega t} \end{aligned} \quad (6.37)$$

where  $\omega$  is the angular frequency of the alternating excitation.

For high frequency analyses, it is not usual to use conductivity, but instead to use a complex permittivity, where the imaginary part takes into account the losses of any material,  $\epsilon = \epsilon' - i\epsilon''$ . The conductivity is then equivalent to the term  $i\omega\epsilon''$ .

With these assumption equation 6.36 becomes

$$\nabla \times \frac{i}{\mu} \nabla \times E_c - \omega^2 \epsilon' E_c + i\omega^2 \epsilon'' E_c = 0 \quad (6.38)$$

or

$$\nabla \times \frac{i}{\mu} \nabla \times E_c - \omega^2 \epsilon' E_c + i\omega\sigma E_c = 0 \quad (6.39)$$

## Boundary Conditions

Boundary conditions are used in two ways. Firstly they can provide a way of reducing the size of the finite element representation of symmetrical models. Secondly they are used to approximate the magnetic field at large distances from the model (far-field boundaries).

### General

Model symmetry and the symmetry of the fields are implied by the electric field boundary conditions applied to the finite element model. The simplest types of boundary condition are:

Boundary Conditions		
	Field Symmetry	Electric Field
TANGENTIAL MAGNETIC	$H \cdot n = 0$	
NORMAL ELECTRIC or PEC	$E \times n = 0$	$E \times n = 0$
NORMAL MAGNETIC or TANGENTIAL ELECTRIC	$H \times n = 0$ $E \cdot n = 0$	$(\nabla \times E) \times n = 0$ $E \cdot n = 0$
RADIATION	See Absorbing boundaries [page 588].	
SIBC	See Surface Impedance Boundary Condition [page 620]	
CURRENTSOURCE	See Current Source Boundary Condition [page 622]	

where  $n$  is the normal unit vector to the surface being considered.

All exterior boundaries should have a boundary condition.

## Sources

Sources in Harmonic HF are defined by assigning non-zero values to components of the electric field, E, or magnetic vector potential, A. For example, specifying

$$E_x = E_z = 0, E_y = \cos\left(\left(\frac{\pi x}{0.086}\right)\right) \quad (6.40)$$

on the boundary representing a port of a microwave device 86 mm wide in X, centred on X = 0 on the XY plane applies the fundamental mode of the Y-directed electric field with peak value of 1 V/m.

Two forms of the source boundary conditions are available:

- total field: **EVECTOR** and **VECTOR** specify the total electric field and magnetic vector potential;
- incident field: **IEVECTOR** and **IVECTOR** specify the incident electric field and magnetic vector potential. Incident field boundaries imply a coincident absorbing boundary condition.

## Reflecting boundaries

Electromagnetic fields are frequently not contained within a finite volume. Except where the interaction with the far environment is of interest, the field from an isolated device is usually required. A simple finite element mesh obviously has a finite extent and applying either tangential or normal field boundary conditions on the open boundaries will perturb the true infinite domain solution, and will in fact reflect all energy incident upon it.

## Absorbing boundaries

There are techniques that accurately model the infinite domain, and one of these has been implemented in the Harmonic HF solver. It is only necessary to identify the outer boundary using the boundary condition **RADIATION**. This imposes an absorbing boundary condition, which effectively absorbs all energy incident upon it, and does not reflect the energy back onto the model being solved. The absorption can be decreased by specifying a complex factor (**ZMULTREALPART**, **ZMULTIMAGPART**) which multiplies the wave impedance of free-space.

It is recommended that the mesh be extended to a reasonable distance, typically about 1 wavelength, using at least 10 to 15 elements per wavelength. The outer boundary should be convex, without sharp corners. Better results will be obtained the closer this boundary is to spherical in shape.

Ideally the radiating source in the model should be at the approximate centre of the coordinate system. The absorbing boundary is then a sphere, centred on the origin of the coordinate system, giving the best accuracy. In general, take care not to place the boundary at or near the origin, as this can cause excessive errors.

# Magnetization Solver

## Introduction

The Opera-3d Magnetization solver can be used to compute the magnetization of permanent magnet materials by time varying electromagnetic fields in three dimensions including the effects of eddy currents. Like the Transient Electromagnetic solver, it does not model displacement current effects. Practically, this means that the largest dimension of the device being modelled must be less than 1/10 of the wavelength of the electromagnetic field in free space. The program incorporates state of the art algorithms for modelling magnetic materials, calculation of electromagnetic fields and advanced finite element numerical analysis procedures. In the following sections the algorithm is described so that users are able to relate the finite element method to their model.

## Material Models

The Magnetization solver calculates the magnetization and demagnetization of magnetic materials in a time varying magnetic field, including the effects of eddy currents. The electromagnetic fields are simulated using a [Vector Potential Formulation \[page 605\]](#) based on edge variable elements.

### Magnetization

During the magnetization process, the maximum value of the flux density in each element is monitored and stored in variables **MAGB**.

### Demagnetization

During demagnetization, the values of **MAGB** determine which demagnetization (second quadrant) curve each element follows and its direction of magnetization. Again the flux density in each element is monitored and the minimum values are stored in variables **MINB**.

The values of **MAGB** can then be transferred to the Transient or Motional EM solver. In such a simulation where the applied field from current sources etc. are opposing the magnet's field, **MINB** will show the operating point of the magnet. In a transient simulation, they will show the lowest operating point that was reached during the transient.

### Remagnetization

If the flux density in element moves into the irreversible part of the 2nd (or 3rd) quadrant curve, i.e. goes beyond the knee of the curve, and the demagnetizing field is then subsequently removed, the remagnetization curve is a straight line defined by its slope (the recoil permeability, **MU**) and the point at  $B=MINB$  on the demagnetization curve.

Demagnetization in service can be modelled. The minimum field will be tracked and updated during subsequent simulations, and the appropriate demagnetization curve or recoil permeability will be used. For more information, see the ***Opera-3d User Guide***.

## Static Thermal Solver

---

### Introduction

This section describes the Static Thermal solver. The program is one of the analysis programs of the Opera-3d Analysis Environment.

The Static Thermal solver can be used to compute the temperature, heat flux, and thermal gradient fields, including the effects of nonlinear materials and boundary conditions, in three dimensions. The program incorporates advanced finite element and nonlinear equation numerical analysis procedures. In the following sections the algorithms are described so that users are able to relate the finite element method to their model.

### The Static Thermal Algorithm

Three dimensional thermal fields can be represented using a single scalar potential ( $T$ ). Physically  $T$  is the usual temperature field. The heat flux density,  $q$ , is given by:

$$q = -k \nabla T \quad (6.41)$$

where  $\kappa$  is the thermal conductivity tensor. The divergence of the heat flux density is related to the heat source density  $Q$ :

$$\nabla \cdot q = Q \quad (6.42)$$

Combining equations 6.41 and 6.42 gives the usual Poisson's equation for the temperature distribution:

$$\nabla \cdot k \nabla T = -Q \quad (6.43)$$

The thermal conductivity tensor and heat source density<sup>1</sup> can be specified as a functions of  $x$ ,  $y$ ,  $z$  and  $T$  (nonlinear analysis), using Fortran-style expressions. The heat-source density can also be specified using element or node-based Heat Tables [page 591].

The various quantities are available in the Post-Processor for viewing the results. The extra system variables available are:  $T$  (temperature),  $DTX$ ,  $DTY$  and  $DTZ$  (temperature gradient, equivalent to  $dT/dx$ ,  $dT/dy$  and  $dT/dz$  respectively),  $QX$ ,  $QY$  and  $QZ$  (equivalent to  $q_x = -kdT/dx$ ,  $q_y = -kdT/dy$  and  $q_z = -kdT/dz$  respectively).

### Heat Tables

When using an element or node-based table to specify the heat source density, the actual table must be saved to the analysis database using the TABLE command in the Post-Processor. The first three

---

<sup>1</sup>In the Pre-Processor, the heat source density can only be a constant within a volume unless it is specified using a table file.

columns of the table must contain the coordinates of the element centroids (element-based) or nodes (node-based) of the finite-element mesh, and the fourth column must contain the heat values and be named **RELEMENTHEAT** (element-based) or **RNODALHEAT** (node-based). The most common use of this feature is to transfer heat values calculated from an electromagnetic analysis, into a thermal analysis.

- Modeller: the volumes in which the table values of heat are to be used should be identified as having the appropriate **HEATTYP**E in **The VOLUME Command [page 342]**.
- Pre-Processor: the volumes in which the table values of heat are to be used should have the label **ELEMENTHEAT** or **NODALHEAT** added by **The LABEL Command [page 458]** as appropriate.

## Boundary Conditions for the Static Thermal Solver

### General

Boundary conditions are primarily used to specify the coupling of the thermal system to its surroundings. A secondary function is to reduce the size of the finite element mesh in symmetric models. Some of the more common boundary conditions are: insulator, specified temperature, specified flux, heat transfer, and radiation. Any of the last 3 of these can be combined into a single boundary condition.

The boundary conditions implemented in the Static Thermal solver are described in the following table (where  $\mathbf{n}$  is the normal unit vector to the surface being considered and  $\sigma$  is the Stefan-Boltzmann constant).

Boundary Conditions	
INSULATOR	$k \frac{\partial T}{\partial n} = q \cdot \mathbf{n} = 0$
TEMPERATURE	$T = a$
HEATFLUX	$q \cdot \mathbf{n} = b$
HEATTRANSFER	$q \cdot \mathbf{n} = h(T - \gamma)$
THERMALCONTACT	$q \cdot \mathbf{n} = h\delta T$
RADIATION	$q \cdot \mathbf{n} = \epsilon\sigma(T^4 - \gamma^4)$
COMBINED	Combination of <b>HEATFLUX</b> , <b>HEATTRANSFER</b> and <b>RADIATION</b> .

The quantities  $\alpha$ ,  $\beta$ ,  $\gamma$  (bath temperature),  $h$  (heat-transfer coefficient or thermal contact conductance) and  $\epsilon$  (emissivity) can all be specified as expressions of  $x$ ,  $y$  and  $z$ . In addition the expressions for  $\beta$  and  $h$  can also contain  $T$ , resulting in a nonlinear analysis. The default condition that will be applied if no boundary condition is specified on an exterior surface is **INSULATOR**.

## Nodal fields

The default boundary condition is only weakly satisfied even when it is assigned to a surface. 'Weakly satisfied' means that it is applied as an integral over the surface patch for each shape function sub-domain. Examination of the heat flow or temperature gradient close to a surface with the default boundary condition will reveal that the normal component is not precisely zero. The magnitude of the normal component reflects the local accuracy of the solution unless it is the result of a model error. However, on the surface itself, the normal component is forced to be precisely zero by the Post-Processor.

The heat flow computed by taking derivatives of the finite element shape functions will be discontinuous but in each element the temperature boundary condition will be exactly specified. The temperature gradient smoothing processes (nodal averaging) must take this into account and hence the true boundary condition is forced at these surfaces to ensure that accuracy is maintained.

## Uniqueness

A potential boundary condition must be specified on at least one surface. This gauges the scalar potential and without it the solution will not be unique. This uniqueness is ensured by requiring at least one **RADIATION**, **TEMPERATURE** or **HEATTRANSFER** (equivalent to specifying the temperature at infinity) condition to be specified by the user.

## Transient Thermal Analysis

---

### Introduction

This section describes the Transient Thermal solver. The program is one of the analysis programs of the Opera-3d Analysis Environment.

Transient Thermal can be used to compute the temperature, heat-flux, and thermal-gradient fields, including the effects of nonlinear materials and boundary conditions, in three dimensions. The program incorporates advanced finite element and nonlinear equation numerical analysis procedures. In the following sections, the algorithms used are described so that users are able to relate the finite element method to their model.

### The Transient Thermal Algorithm

Three dimensional thermal fields can be represented using a single scalar potential  $T$ . Physically,  $T$  is the usual temperature field. The thermal-gradient intensity  $\mathbf{DT}$  is given by:

$$\mathbf{DT} = -\nabla T \quad (6.44)$$

and the heat flux density  $q$  by:

$$q = k\mathbf{DT} \quad (6.45)$$

where  $\kappa$  is the thermal conductivity. Energy conservation requires the time rate of change of internal thermal energy within a volume to equal the rate of heat generated within the volume minus the rate of heat flux through its surface; applying energy conservation to an infinitesimal volume yields:

$$\rho C \frac{\partial}{\partial t} T = Q - \nabla \cdot q \quad (6.46)$$

where  $Q$  is the heat source density,  $\rho$  the mass density, and  $C$  the heat capacity. Combining equations 6.44, 6.45 and 6.46 gives the well-known diffusion equation for the temperature distribution:

$$\rho C \frac{\partial}{\partial t} T - \nabla \cdot k \nabla T = Q \quad (6.47)$$

In the time integration of equation 6.47, the user can specify either fixed or adaptive time-stepping. In the case of adaptive time-stepping, the result from a full step is compared to the result using two half steps; if the results agree to within a specified tolerance, then the step size is maintained (or doubled if the difference is small enough), otherwise it is halved. For most cases adaptive stepping will be more efficient.

The quantities  $\kappa$ ,  $\rho$ ,  $C$  and  $Q$  can all be specified as a functions of  $X$ ,  $Y$ ,  $Z$ , and  $T$  (nonlinear analysis), using Fortran-style expressions<sup>1</sup>; it is also possible to use more general expressions, in terms of

---

<sup>1</sup>In the Pre-Processor, the heat source density,  $Q$ , can only be a constant within a volume unless it is specified using a table file.

other variables, provided that the values of those variables are assigned before running the analysis; this assignment is done using the **TABLE** command in the Post-Processor. The heat-source density can also be specified using element-based or node-based **Heat Tables** [page 595], so that results from an electromagnetic analysis can be used as input for a thermal analysis.

The various quantities are available in the Post-Processor for viewing the results. The extra system variables available are: **T** (temperature), **DTX**, **DTY** and **DTZ** (temperature gradient, equivalent to  $dT/dx$ ,  $dT/dy$  and  $dT/dz$  respectively), **QX**, **QY** and **QZ** (equivalent to  $q_x = -kdT/dx$ ,  $q_y = -kdT/dy$  and  $q_z = -kdT/dz$  respectively).

See [Transient Analysis Logging and Control \[page 627\]](#) for more information on transient analysis (logging is not available during Transient Thermal analysis).

## Heat Tables

When using an element-based or node-based table to specify the heat source density, the actual table must be saved to the analysis database using the **TABLE** command in the Post-Processor. The first three columns of the table must contain the coordinates of the element centroids (element-based) or nodes (node-based) of the finite-element mesh, and the fourth column must contain the heat values and be named **RELEMENTHEAT** (element-based) or **RNODALHEAT** (node-based). The most common use of this feature is to transfer heat values, calculated from an electromagnetic analysis, into a thermal analysis.

- Modeller: the volumes in which the table values of heat are to be used should be identified as having the appropriate **HEATTYP**E in [The VOLUME Command \[page 342\]](#).
- Pre-Processor: the volumes in which the table values of heat are to be used should have the label **ELEMENTHEAT** or **NODALHEAT** added by [The LABEL Command \[page 458\]](#) as appropriate.

## Boundary Conditions for the Transient Thermal Solver

### General

Boundary conditions are primarily used to specify the coupling of the thermal system to its surroundings. A secondary function is to reduce the size of the finite element mesh in symmetric models. Some of the more common boundary conditions are: insulator, specified temperature, specified flux, heat transfer, and radiation.

The boundary conditions implemented in the Transient Thermal solver are described in the following table (where **n** is the normal unit vector to the surface being considered and  $\sigma$  is the Stefan-Boltzmann constant).

Boundary Conditions	
INSULATOR	$k \frac{\partial T}{\partial n} = q \cdot n = 0$

TEMPERATURE	$T = a$
HEATFLUX	$q \cdot n = b$
HEATTRANSFER	$q \cdot n = h(T - \gamma)$
THERMALCONTACT	$q \cdot n = h\delta T$
RADIATION	$q \cdot n = \epsilon\sigma(T^4 - \gamma^4)$
COMBINED	Combination of HEATFLUX, HEATTRANSFER and RADIATION.

The quantities  $\alpha$ ,  $\beta$ ,  $\gamma$  (bath temperature),  $h$  (heat-transfer coefficient or thermal contact conductance) and  $\epsilon$  (emissivity) can all be specified as expressions of  $x$ ,  $y$  and  $z$ . In addition the expressions for  $\beta$  and  $h$  can also contain  $T$ , resulting in a nonlinear analysis. The default condition that will be applied if no boundary condition is specified on an exterior surface is **INSULATOR**.

### Nodal field extraction

The default boundary condition is only weakly satisfied even when it is assigned to a surface. 'Weakly satisfied' means that it is applied as an integral over the surface patch for each shape function sub-domain. Examination of the field solution close to a surface with the default boundary condition will reveal that the normal component is not zero. The magnitude of the normal component reflects the local accuracy of the solution unless it is the result of a model error. On the surface itself however, the normal component is forced to be precisely zero by the Post-Processor.

The fields computed by taking derivatives of the finite element shape functions will be discontinuous, but in each element the potential boundary condition will be exactly specified. The field smoothing processes (nodal averaging) must take this into account, and hence the true boundary condition is forced at these surfaces to ensure accuracy is maintained as much as possible.

# Quench Analysis

## Introduction

This section describes the Quench solver. The program is one of the analysis programs of the Opera-3d Analysis Environment.

The Quench solver is designed specifically for quench modelling of superconducting coils as the coils become resistive. It is based on the Transient Thermal solver, coupled with the analysis of electrical circuits and fields from Biot-Savart conductors.

## Modelling

Models for the Quench solver can only be created in the Opera-3d Geometric Modeller.

### Superconducting Coil Modelling

The superconducting coils can be modelled using the pre-defined conductor primitives. These conductors can be marked as circuit elements for use in circuit modelling. The thermal modelling requires a finite element mesh, and the conductor primitives can be marked for conversion to meshable volumes. This conversion takes place during create model body, and the cells created have a material label set to be the circuit element name, and an orientation set up from the conductor primitive, with the local Z axis being parallel to the superconducting wires.

### Material Modelling

The material modelling of the coils is set using [The MATERIALS Command \[page 248\]](#). Bulk thermal properties of the superconductors need to be defined. Additionally, wire material properties, including critical current, wire conductivity, wire cross-section need to be defined. The current that will be flowing through the wire is automatically assigned to be the current through the circuit element, but this can be adjusted if necessary.

All of these material properties can be defined as functional properties, with expressions based on local temperature and local field where necessary. Tabular functional properties can be included within these expressions allowing simpler definition of functions, particularly for multi-dimensional variations e.g. when specifying critical current as a function of temperature and field.

### Circuit Modelling

Circuits must be defined using [The CEDITOR Command \[page 162\]](#) or [The CIRCUIT Command \[page 173\]](#).

As the analysis does not include electromagnetic analysis, the inductance and inductive coupling between circuit windings needs to be included explicitly. This can be done by adding mutual coupling

terms between pairs of circuit winding elements, and where the same circuit element is given twice, a self-inductance will be set.

## Quench Thermal Analysis

The Quench solver incorporates the nonlinear transient thermal analysis as described in [Transient Thermal Analysis \[page 594\]](#), together with the circuit analysis. See also [Transient Analysis Logging and Control \[page 627\]](#) for more information on transient analysis and data logging during the analysis.

At each stage, material properties are checked and where the critical current is exceeded by the current through a circuit element, a local area of resistivity is introduced. This resistance has 2 effects.

- The resistance of the circuit element is altered, leading to a variation in the circuit solution.
- Local resistive losses cause local heating, and this is introduced into the thermal model. This additional heating propagates into neighbouring elements, increasing the temperature and in turn the resistive zone.

Fields and their time-derivatives are available locally within the thermal model, and are calculated using the Biot-Savart field calculation. The field from these conductors is scaled by the circuit current through the conductor, so that as the currents change, the local magnetic field is also updated.

## Quench Multiphysics Analysis

The Quench solver can be coupled with the Transient Electromagnetic solver to model transient electromagnetic fields and circuits. In this way transient eddy current in electrically conducting materials can be simulated accurately.

A coupled Quench model is built by choosing **Quench Multiphysics** in the Analysis Settings tool. Background air regions must be included in coupled models to correctly model the electromagnetic fields. All regions having no thermal conductivity will be excluded from thermal simulation.

The Modeller will create two database files, one for Quench and one for Transient EM. The database file for Transient EM has an extra suffix \_ETR in its file name. When using the Modeller or the Opera Manager, the analysis can be started using either database; the analysis of the other database will start automatically.

During the analysis, the temperatures, electromagnetic fields, circuit variables etc. are exchanged between the Quench and Transient EM solvers at every time-step.

# Stress Analysis

---

## Introduction

There are two Opera-3d solvers for Stress analysis:

- Static Stress calculates the deformation, stresses and strains of a body subject to external and internal forces. The forces calculated from electromagnetic fields can be used as input to the stress analysis.
- Modal Stress calculates the modes, i.e. the eigenvalues and eigenvectors, of a body without any internal or external forces.

The programs uses volume finite elements and therefore can be used with the same models as are used for electromagnetic analysis.

Data for stress analysis can only be created using the Modeller.

## Material Properties

Stress analysis can analyse isotropic and anisotropic (orthotropic) and fully anisotropic materials. It uses the following material properties:

- Young's modulus
- Poisson's ratio
- Shear modulus
- Elasticity matrix (fully anisotropic materials)
- Thermal expansion coefficients (Static Stress only)
- General expansion integrals (Static Stress only)
- Density (Static Stress only)

The material properties can be specified as numeric values or as expressions. For example, the thermal expansion coefficients could be functions of temperature or position.

## Internal Forces

In the Static Stress solver, the model can be deformed by internal forces from 3 sources:

- a functional force density, which can use values calculated by the Post-Processor from an electromagnetic analysis, e.g. EFD, the element force density (see [Element Force Densities \[page 803\]](#));
- rotational forces calculated from a constant rotational velocity about the global Z axis;
- gravity (any of the 6 major coordinate directions can be defined as "down").

## External Forces and Constraints

### Static Stress

In the Static Stress solver, external forces and constraints can be applied globally or as boundary conditions on the surfaces of the model. The available forces are:

- pressure, a force normal to the surface, and
- traction, a force in any direction.

The available constraints are:

- fixed:
  - fixed in all 3 directions;
  - fixed in tangential directions;
  - fixed in the direction normal to the surface;
- defined normal displacement;
- defined displacement in all 3 directions;
- thermal expansion, with values of reference and operating temperatures, which can be imported from thermal analysis, and
- general expansion, which can be used to model magnetostriction, electrostriction and piezo effects, with the expansion integrals expressed as functions of magnetic or electric field vectors imported from electromagnetic analysis.

### Modal Stress

In the Modal Stress solver, the available constraints are:

- fixed in all 3 directions;
- fixed in tangential directions, and
- fixed in the direction normal to the surface.

## Eigenvalue Frequency Range

The Modal Stress solver finds a specified number of eigenvalues above a given frequency, or within a specified range. Multiple simulations are created in the analysis database allowing post-processing of each mode.

## Post-Processing

The Stress solvers calculate displacement vectors and stress ( $\sigma$ ) and strain ( $\epsilon$ ) tensors for display by the Post-Processor. In addition, the Post-Processor calculates the principals and invariants of the

stress and strain tensors and the Von Mises stress,

$$\sigma_V = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}}, \quad (6.48)$$

where  $\sigma_1$ ,  $\sigma_2$ , and  $\sigma_3$  are the principal stresses.

All the above quantities can be calculated and displayed in the Post-Processor at points, along lines and on surfaces in the same way field quantities from other solvers. In addition, displacements can be visualised in the "The THREED Command" on page 823 command as a deformation of the solid or outline view of the model.

## Scalar Potential Formulation

---

Three dimensional stationary electromagnetic fields can be represented as the sum of a solenoidal field and a rotational field. In electrostatic fields there is never a rotational component, the field can therefore always be defined using the electrostatic potential ( $V$ ). The electric field intensity ( $\mathbf{E}$ ) is given by:

$$\mathbf{E} = -\nabla V \quad (6.49)$$

The divergence of the electric flux density ( $\mathbf{D}$ ) is related to the charge density ( $\rho$ ):

$$\nabla \cdot \mathbf{D} = \rho \quad (6.50)$$

Combining equations (6.49) and (6.50) and introducing the dielectric permittivity tensor ( $\epsilon$ ) gives the usual Poisson's equation description of the electrostatic potential:

$$\nabla \cdot \epsilon \nabla V = -\rho \quad (6.51)$$

where  $\mathbf{D} = \epsilon \mathbf{E}$ .

A similar equation arises for current flow analyses,

$$\nabla \cdot \sigma \nabla V = 0 \quad (6.52)$$

where  $\sigma$  is the conductivity, and  $\mathbf{J} = \sigma \mathbf{E}$ .

Stationary magnetic fields on the other hand in general consist of both solenoidal and rotational components. The field produced by electric currents has a rotational component inside the volumes where the currents flow. In the exterior space the field is solenoidal but the scalar potential is multivalued. The field produced by magnetized volumes is solenoidal. It is convenient to use a splitting of the total field into two parts in order to obtain a description of the field in terms of a simple scalar potential.

The total field intensity  $\mathbf{H}$  is defined using the reduced field intensity ( $\mathbf{H}_m$ ) and the conductor field intensity ( $\mathbf{H}_s$ ):

$$\mathbf{H} = \mathbf{H}_m + \mathbf{H}_s \quad (6.53)$$

The reduced field intensity can now be represented using the reduced scalar potential ( $\phi$ ):

$$\mathbf{H}_m = -\nabla \phi \quad (6.54)$$

and in the case of stationary magnetic fields where the exciting currents are prescribed, the conductor field intensity may always be directly evaluated by integration:

$$\mathbf{H}_s = \int_{\Omega_J} \frac{\mathbf{J} \times \mathbf{R}}{|\mathbf{R}|^3} d\Omega_J \quad (6.55)$$

The divergence of the flux density is always zero. Introducing the permeability tensor ( $\mu$ ) and combining equations (6.53) to (6.55) gives the partial differential equation for the reduced scalar potential ( $\phi$ ):

$$\nabla \cdot \mu \nabla \phi - \nabla \cdot \mu \left( \int_{\Omega_J} \frac{\mathbf{J} \times \mathbf{R}}{|\mathbf{R}|^3} d\Omega_J \right) = 0 \quad (6.56)$$

This equation, like the Poisson's equation for electrostatic fields, can easily be solved using the finite element method. However the reduced potential formulation for magnetic fields is not acceptable. Large errors are found in the total fields computed by this method.

The errors arise because the space variation of  $\mathbf{H}_m$  and  $\mathbf{H}_s$  will be quite different if one is represented using derivatives of the finite element shape functions (a low order polynomial in  $x$ ) and the other using direct evaluation of the integral in equation (6.55). This effect combined with the possibility that  $\mathbf{H}_m$  and  $\mathbf{H}_s$  will strongly cancel in some volumes of the space leads to magnification of the errors in the approximate solution for  $\mathbf{H}_m$ .

This cancellation is particularly critical in the interior of nonlinear magnetic materials, where the magnified errors destroy the accuracy of the Jacobian matrix used for Newton iterations. It is also very undesirable when magnetic shielding is being designed since the largest errors will occur in volumes where the shield is most effective.

The above difficulty can be completely avoided when currents are not flowing in the magnetic materials. Exterior to the volumes where currents flow the total field can be represented using the total magnetic scalar potential ( $\psi$ ):

$$\mathbf{H} = -\nabla\psi \quad (6.57)$$

where the total magnetic scalar potential satisfies:

$$\nabla \cdot \mu \nabla\psi = 0 \quad (6.58)$$

By combining the two representations (the total and reduced scalar potentials) cancellation difficulties can be completely avoided. The minimal combination consists of using the reduced potential only inside volumes where currents flow and the total potential everywhere else. This has practical limitations in that, unless the reduced potential volume has a simple shape, cutting surfaces are needed in the total potential space to maintain a single valued potential. For these reasons simple singly connected spaces are generally used for the reduced potential volumes but, where that is not possible, facilities exist in the software to create the cuts automatically.

In exceptional circumstances, where it is impossible to create singly connected total scalar potential volumes or to avoid the use of the automatic cuts, it is possible to use reduced scalar potential within magnetic materials. Because of the cancellation problems described above, such reduced scalar potential volumes should be as small as possible and away from regions where accuracy is critical. Reduced scalar potential must be used in magnetic material carrying source currents.

On the interface between the total and reduced potential spaces the two potentials can be exactly linked together by applying the conditions of normal  $\mathbf{B}$  and tangential  $\mathbf{H}$  continuity. This involves evaluation of the normal field produced by the conductors<sup>1</sup> and the scalar potential that could be used to represent the conductor field on the interface surface. Practically, this also makes the method more attractive than the straightforward reduced potential formulation. Evaluating the fields from conductors although well defined, is expensive. The reduced potential formulation requires that the conductor fields be evaluated throughout the nonlinear magnetic material volumes. The total potential

---

<sup>1</sup>External fields are treated in the same way as fields from conductors. Reduced potential must be used in the "far-field" region of the finite element mesh so that the external field can be calculated on the interface between reduced and total potentials.

on the other hand only requires the conductor field to be evaluated on the interface surface between the reduced and total potential volumes for the solution of the equations. However, the conductor field is needed throughout the reduced potential volumes to obtain the total field during post-processing. It is also needed within volumes defined as reduced potential within magnetic regions.

## Vector Potential Formulation

---

The Vector Potential Formulation is used by the Electromagnetic (including Motional EM and Magnetization) and High Frequency solvers. The finite element method they use represents the magnetic vector potential with linear edge elements. Quadratic elements are treated as linear elements.<sup>1</sup>

Low frequency electromagnetic fields are described by the quasi-static limit of Maxwell's equations which exclude displacement currents. In this limit Maxwell's equations have the form

$$\nabla \times H = J \quad (6.59)$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (6.60)$$

$$\nabla \cdot B = 0 \quad (6.61)$$

where

$$J = \sigma(E + u \times B) \quad (6.62)$$

$\sigma$  is the electrical conductivity and  $u$  is the velocity of media with respect to the field. The affect of non-zero velocity will be discussed in [Fixed Velocity Equations \[page 607\]](#).

From 6.61 it follows that the flux density  $\mathbf{B}$  can be derived from a vector potential

$$B = \nabla \times A \quad (6.63)$$

### TOTAL and REDUCED Magnetic Vector Potentials

The Electromagnetic solvers use a combination of total and reduced vector potentials to model time varying electromagnetic fields. The magnetic field produced by a known distribution of current in free space (for example, the field from a coil wound with fine wire carrying a specified current) can be calculated by integration of the Biot-Savart equation. The vector potential describing the magnetic field excluding the fields from these source conductors, is called a reduced vector potential ( $A_R$ ) and it is defined by

$$B = \mu_0 H_s + \nabla \times A_R \quad (6.64)$$

In regions where the field is only derived from a (total) vector potential, combining equations (6.59) to (6.63) gives the following equation for  $A$ :

$$\nabla \times \frac{I}{\mu} \nabla \times A = -\sigma \frac{\partial A}{\partial t} - \sigma \nabla V \quad (6.65)$$

The electric scalar potential ( $V$ ) emerges because of the non-uniqueness of the potential which arises during integration of equation 6.60. In free space the electric scalar potential can be set to zero without any loss of generality, however in conducting regions a secondary equation (derived from  $\nabla \cdot J = 0$ ) is introduced

$$\nabla \cdot \sigma \nabla V + \nabla \cdot \sigma \frac{\partial A}{\partial t} = 0 \quad (6.66)$$

---

<sup>1</sup>Quadratic elements can be used on curved surfaces, but only to improve the display. The solution is still performed with linear elements.

so that both the electric scalar potential and the vector potential can be determined.

In free space regions containing source currents, where the reduced vector potential is used, combining equations (6.59) and (6.64) gives the following equation for  $\mathbf{A}_R$ :

$$\nabla \times \frac{I}{\mu_0} \nabla \times \mathbf{A}_R = 0 \quad (6.67)$$

The total and reduced vector potential descriptions of the field are directly combined in the formulation. The normal flux and tangential field intensity interface conditions exactly determine the relationship between the two types of vector potential.

## Source Conductors and Circuits

In the previous section source conductors were discussed in connection with the use of reduced vector potentials. There are two classes of source conductors:

- Biot-Savart current source - these source conductors are current driven, their fields are calculated by integration. They are independent of the finite element mesh, but must be contained in reduced potential regions<sup>1</sup>.
- Circuit element - these conductors are incorporated into the finite element mesh, represented by winding filaments or a volume meshed conductor, and they are connected to circuits. The circuits can be either voltage or current driven and may also contain passive components such as inductors and capacitors. These conductors must be included in total potential regions if there are any Biot-Savart source conductors present.

A source conductor may be either a "Biot-Savart current source" or a "Circuit element". See [Conductors in Circuits \[page 533\]](#) for more details of the parameters which control the representation of conductors in circuits.

## The Time Harmonic Equations

Under steady-state alternating current excitation a complex potential substitution can be used. This carries the assumption that the materials have linear characteristics and that therefore the waveforms of the current and field are precisely the same. The substitutions are:

$$\begin{aligned} A(t) &= A_c e^{i\omega t} \\ V(t) &= V_c e^{i\omega t} \end{aligned} \quad (6.68)$$

where  $\omega$  is the angular frequency of the excitation. With this assumption equation (6.65) becomes

$$\nabla \times \frac{I}{\mu} \nabla \times \mathbf{A}_c + i\omega\sigma\mathbf{A}_c + \sigma\nabla V_c = 0 \quad (6.69)$$

and equation (6.66) becomes

---

<sup>1</sup>External fields are treated in the same way as fields from Biot-Savart conductors. Reduced potential must be used in the "far-field" region of the finite element mesh so that the external field can be calculated on the interface between reduced and total potentials.

$$\nabla \cdot \sigma \nabla V_c + i\omega \nabla \cdot \sigma A_c = 0 \quad (6.70)$$

The form of the equations for the reduced vector potential is not affected by this substitution.

## Quasi-nonlinear solutions

Although it has been assumed that the materials are linear, it is still possible to model nonlinear materials using a quasi nonlinear model.

The assumption is that  $\mathbf{B}$  and  $\mathbf{H}$  are both in phase, so that (6.68) is still valid. The supremum<sup>1</sup> of the field is used to determine the value of the permeability (based on the usual BH relationship). This is a nonlinear process, and a simple update scheme is used— having solved equation (6.69), the element permeabilities are updated based on the BH curve, and the equation re-solved, until a converged solution is obtained.

## The Transient Equations

Transient Electromagnetic solvers (including Modal EM and Magnetization) analyse eddy current models where the driving sources are changing in time in a predetermined way, but not necessarily with a time harmonic behaviour.

The time-stepping method is described in [Time-Stepping \[page 624\]](#).

## Fixed Velocity Equations

The Fixed Velocity Electromagnetic solver calculates static fields, including the effect of eddy currents induced by motion. The conducting parts of the model can be assigned a velocity. This can be a linear velocity, specified by speed and direction or a rotational velocity specified by the angular velocity around the global z-axis. In either case, in each element there is a velocity vector,  $\mathbf{u}$ . It is always assumed that the driving sources are stationary and the eddy current conductors in motion. The formulation assumes that, at every instant in time, the geometry of the model is identical. This means that the cross-section of the moving conductor orthogonal to the direction of motion does not change. In other words, for linear motion the moving conductor is "infinite" and for rotational motion the RZ cross-section is invariant. Examples of these might be a pipeline inspection vehicle and an eddy current disk brake respectively.

From equations (6.62) and (6.64), the current is related to the potentials  $\mathbf{A}$  and  $V$  by

$$J = \sigma(\mathbf{u} \times \nabla \times \mathbf{A}) - \sigma \nabla V \quad (6.71)$$

where  $\mathbf{u}$  is the velocity. The equation that is solved (based on (6.67)) is

$$\nabla \times \frac{I}{\mu} \nabla \times \mathbf{A} - \nabla \frac{I}{\mu} \nabla \cdot \mathbf{A} = \sigma(\mathbf{u} \times \nabla \times \mathbf{A}) - \sigma \nabla V \quad (6.72)$$

<sup>1</sup>The supremum is the maximum length of the vector at any time around the ac cycle. If the x, y and z components of the vector have the same phase angle, the supremum is the same as the dyadic modulus; if they are out of phase, the supremum is somewhat less.

and equation (6.66) becomes

$$\nabla \cdot \sigma \nabla V - \sigma \nabla \cdot (u \times \nabla \times A) = 0 \quad (6.73)$$

Source currents in Velocity EM are dc; circuits are not available.

## Upwinding in Velocity EM

The analysis of motion induced eddy current models (without upwinding) will probably present oscillations if the 'cell Peclet number'  $\mu\sigma u/h$  is much greater than one ( $h$  being the size of the element in the velocity direction  $u$ ). This still may occur after mesh refinement. Such a phenomenon is well understood and is typical of the numerical methods. It arises when first order derivatives become very large compared to second order ones (see motional equations). To overcome this problem, "upwinding" can be employed to eliminate non-physical "wiggles" from the solution although with a possible degradation in accuracy. The number of iterations required by the solver is usually smaller than that for non-upwinded solutions, since the system matrix is typically better conditioned.

## The Finite Element Method

---

The Finite Element method is used to obtain solutions to partial differential or integral equations that cannot be solved by analytic methods. Partial differential and integral equations describe the spatial and temporal variation of a field either directly in terms of the field variable, for example the magnetic flux density  $\mathbf{B}$ , but more often using a potential function that is related to the Field by a gradient ( $\nabla$ ) or curl ( $\nabla \times$ ) operation. The finite element method is generally applicable to any model with any type of nonlinearity. The method is based on division of the domain of the equation (volume of space in which the equation is satisfied) into small volumes (the finite elements). Within each finite element a simple polynomial is used to approximate the solution.

The concepts used in finite element analysis are independent of the number of space dimensions, however it is convenient to use a simple one space dimension model in order to make the algebra straightforward and explanatory diagrams easy to view. Consider a Poisson type equation describing a potential function  $\phi$  in one dimension:

$$\nabla \cdot \epsilon \nabla \phi = \rho \quad (6.74)$$

The potential function  $\phi$  might be an electrostatic potential, in which case  $\rho$  would be a line charge density. In order to define  $\phi$ , boundary conditions are required, these may be either assigned values of  $\phi$  or its derivative, for example:

$$\frac{\partial \phi}{\partial x} = 0 \quad (6.75)$$

In all electromagnetic field examples it is essential that the potential is defined at one point in the domain at least, otherwise an infinite number of solutions could be generated by adding an arbitrary constant to the solution.

To solve equation (6.74) using a finite element method the domain is divided into line elements. A typical first order line element would have two nodes numbered e.g. 1 and 2. Within this element the potential  $\phi$  will be approximated by a linear polynomial:

$$\phi(x) = a + bx \quad (6.76)$$

The electrostatic potential  $\phi$  will be continuous over the domain, although its derivatives may be discontinuous if the permittivity  $\epsilon$  changes discontinuously. The finite element model should be capable of representing this behaviour and it is therefore convenient to characterize the polynomial shown in equation (6.76) by the values of  $\phi$  at the nodes of the element and use the same nodal value to characterize the polynomials in other elements that meet at the node. A further simplification is introduced by rewriting equation (6.76) in terms of nodal shape functions  $N_i$  defined such that:

$$\begin{aligned} N_i(x) &= 1; x = x_i \\ N_i(x) &= 0; x = x_j, j \neq i \end{aligned} \quad (6.77)$$

where  $x_i$  is the x coordinate of node  $i$  etc. The shape functions have the same polynomial form as the  $\phi$  approximation, and equation (6.76) can be written as:

$$\phi(x) = N_1(x)\phi_1 + N_2(x)\phi_2 \quad (6.78)$$

The shape functions  $N_i$  are usually expressed in terms of a local coordinate system in the element. This can be used to simplify the expressions and furthermore avoids problems of numerical rounding errors. Using the local coordinate system  $\xi$  the shape functions can be written as:

$$\begin{aligned} N_1 &= \frac{1}{2}(1 - \xi) \\ N_2 &= \frac{1}{2}(1 + \xi) \\ -1 \leq \xi &\leq 1 \end{aligned} \quad (6.79)$$

The shape function for a particular node is only defined in the elements that use the node and is zero outside these elements. The approximation to  $\phi$  is described as having local support when nodal shape functions of this type are used.

The discrete method of approximating the potential  $\phi$  using characteristic nodal values and associated shape functions that determine the spatial variation of the approximation provides the basis on which several alternative procedures could be used to solve equation (6.74). Variational methods, least squares and weighted residual procedures are three of the most frequently used. Weighted residuals have wide application and they are used in the software to develop a numerical solution. An approximate solution  $\phi$  is determined by requesting that this function should satisfy:

$$\int W(\nabla \cdot \epsilon \nabla \phi - \rho) dx = 0 \quad (6.80)$$

The weighted residual method can be used with either global (defined over the whole domain) approximations to  $\phi$  or the local approximations discussed here.  $W$  is a weighting function from which the method gains its name. The Galerkin weighted residual method is the best choice for the types of equation arising in electromagnetism. In this case the basis functions approximating  $\phi$  are also used for the weights. Equation (6.80) is often referred to as a strong form because of the constraints it places on the functions that can be used in the approximation  $\phi$  (the first derivative would clearly have to be continuous over the domain). In general a weak form of equation (6.80) is used to remove the derivative continuity requirement. This weak form is obtained by integrating equation (6.80) by parts (in more than one dimension this involves application of Green's theorem). Integrating equation (6.80) by parts to reduce the order of differentiation applied to  $\phi$  gives:

$$\int_a^b (\nabla W \cdot \epsilon \nabla \phi + W \rho) dx - \left( W \epsilon \frac{\partial \phi}{\partial x} \right)_a^b \quad (6.81)$$

where  $a$  and  $b$  are the limits of the domain of the equation. The weak form has several advantages: the functions representing  $W$  and  $\phi$  do not need derivative continuity and the natural boundary condition on the surface of the domain,

$$\frac{\partial \phi}{\partial x} = 0 \quad (6.82)$$

has emerged. Equation (6.81) leads directly to a numerical solution method, using the discrete finite elements and shape functions discussed above. Discretization of the domain  $ab$  into line elements with their associated nodes gives a set of independent weighting functions (the shape functions of the nodes) from which a set of equations can be developed by requiring that equation (6.81) is satisfied independently for each weight function. The equation for weight function  $W_i$ , i.e. shape function  $N_i$ , is obtained from:

$$\sum_j \left( \int_a^b (\nabla N_i \cdot \epsilon \nabla N_j \phi_j + N_i \rho) dx \right) - \left( N_i \epsilon \frac{\partial \phi}{\partial x} \right)_a^b = 0 \quad (6.83)$$

for all elements containing node  $i$ . Taking all the equations for the different weight functions together gives a set of linear equations, which written in matrix form are:

$$K\Phi = S \quad (6.84)$$

where  $K$  is a coefficient matrix (often called a stiffness matrix because of the background of finite elements in mechanics),  $\Phi$  is a vector of unknown nodal potential values and  $S$  the known right-hand side vector derived from the prescribed line charge densities or assigned boundary conditions. An individual element of the stiffness matrix consists of terms of the form:

$$K_{ij} = \int_a^b \nabla N_i \cdot \epsilon \nabla N_j dx \quad (6.85)$$

Note that the local support of the shape functions means that although the integral in equation (6.85) is taken over the whole domain, only elements containing both nodes  $i$  and  $j$  actually contribute. In the equations arising in electromagnetism the matrix equation (6.84) is frequently nonlinear because the value of  $\epsilon$  (or more frequently  $\mu$  for magnetic fields) is dependent on the field intensity.

## Nonlinear materials

A Newton-Raphson method can be used to solve this type of nonlinear equation. Given an initial solution  $\Phi_n$  a new solution  $\Phi_{n+1}$  is found by solving the linearized Jacobian system<sup>1</sup>:

$$\Phi_{n+1} = \Phi_n - J_n^{-1} R_n \quad (6.86)$$

where the residual  $R$  is given by

$$R_n = K_n \Phi_n - S_n \quad (6.87)$$

and the Jacobian  $J$  by

$$J_n = \frac{\partial}{\partial \Phi_n} (K_n \Phi_n - S_n) \quad (6.88)$$

This method converges providing the initial approximation used to start the iteration is not too far from the real solution. As it approaches the solution its convergence becomes quadratic. In the context of nonlinear finite element solutions to the electromagnetic field equations the reliability of the Newton-Raphson method is strongly linked to the smoothness of the equation used to relate the permeability or permittivity to the field.

To aid convergence, when the material property curve is not smooth or when the initial solution is far from the final solution, a relaxation factor is used in equation (6.86), which becomes:

$$\Phi_{n+1} = \Phi_n - \alpha J_n^{-1} R_n \quad (6.89)$$

where  $\alpha$  is chosen, starting with 1 and multiplying by 2 (if the change in  $|R|$  is too small) or dividing by 2 (if the norm of the residual  $|R_{n+1}|$  would be greater than  $|R_n|$ ) to find the value which minimizes the norm of the residual  $|R_{n+1}|$  at the start of the next iteration.

<sup>1</sup>The subscript 'n' indicates the iteration number

## Finite Element Applications

The experience of the user has been a vital ingredient in the successful application of finite elements to predictive engineering design. Large finite element systems for mechanical design have a mystique associated with them, partly from their origins as stand alone programs with a text file user input interface and a box of paper as the output display, and partly as a result of the jargon that is used to describe the element types and procedures available within the programs. It is now impossible to consider the use of finite element analysis programs without interactive pre-processors for data input, although these bring their own problems if they do not interface well with the analysis programs and thus increase the amount of knowledge needed to perform a calculation.

In electromagnetic field calculations special finite elements are not needed to solve the equivalent of shell and plate geometries that are so common in mechanical design. However, electromagnetic fields must usually be computed to much higher accuracy than is needed in other disciplines, the geometry is frequently complicated with a wide range of dimensions and the actual result needed by the user is often derived from the field solution by integration or differentiation. In whatever form the results are required, the basic limitation of finite element solutions is that the accuracy of the solution is related to the size of the discrete elements. Recent research has resulted in the development of techniques that can be used to determine the error in a finite element solution but this is strictly only correct for models with linear materials. The adoption of these techniques in finite element programs will improve the reliability of results, but they do not help to check that the finite element model and the physical model are equivalent.

## Codes of Practice

Users of finite element programs must prove that the model is consistent with the physical model. With electromagnetic fields it is often possible to perform simple calculations that give *orders of magnitude* answers as an essential part of the analysis. Until the accuracy of the model has been established it is irrelevant to consider the discretization errors. A code of practice should be established that is followed whenever a new analysis is begun:

1. If it is appropriate solve a simplified two dimensional model of the system, apply tests 2,3 and 4 (below) to that solution and then use the two dimensional solution as an initial check of the full three dimensional solution.
2. Once the three dimensional model has been defined, solve the simplest possible model i.e. using linear materials either with unity or large relative permeability or permittivity.
3. Check that the solution has the symmetry that is expected. For example, examine the fields on the boundaries of the model to see if they are as expected.
4. Check that the solution agrees with simple line integral predictions or images if infinite permeability approximations are applicable.

Only when a degree of confidence has been established in the model is it worth beginning to consider the errors produced by the discrete finite element approximation. In many ways these errors are more straightforward to evaluate than the accuracy of the model.

## Solution Errors

The local error at a point within a finite element model is strongly linked to the size of the elements surrounding the point and weakly linked to the average element size over the whole space, although this second source of error becomes more important and less easily estimated in nonlinear solutions. The relationship between the local error in the solution and the surrounding elements' size is given by:

For linear shape functions

$$E(\Phi) = O(h^2) \quad (6.90)$$

and for quadratic shape functions

$$E(\Phi) = O(h^3) \quad (6.91)$$

where  $E$  is the error,  $O$  means 'of the order' and  $h$  is the linear dimension of the elements. This simple analysis is only true for square elements, but it is reasonable to assume the worst case and use the largest dimension for  $h$ . Unfortunately, these formulae only give the order of the error, the actual error is dependent on the solution, or more precisely the geometry of the model in the vicinity of the point. As an example consider a point close to the corner of a magnetized steel cube, the field will be weakly singular at the corner. Given the same size discretization over the whole space, the errors will be far larger close to the edges and corners of the cube. This is because the low order polynomials used in the finite elements are not good at approximating the singularity. Calculating the magnetic field from the potential solution generally results in larger errors in the field than there were in the potential.

Differentiation of the finite element shape functions to determine the field gives an error in the field that is worse by  $O(h^1)$ . In the case of linear shape functions this results in an error in the field  $O(h)$ . In the analysis modules special facilities have been included in order to reduce the errors in the fields that are computed from potential solutions. Two methods are available that increase the field precision; the best method depends on the model being solved (see sections on accuracy in the following chapters). Nodal weighted averaging improves the field accuracy to  $O(h^2)$ . The volume integration technique does not improve the order of the error, but it enables the variation of the field to be calculated very accurately remote from magnetic, dielectric or conducting regions.

The programs use error estimation techniques to produce local and global errors in the fields derived from potential solutions ([Solution Errors \[page 613\]](#)). These displays show where the finite element discretization needs refinement as well as showing the error. However, even with these features, it is important that the user of a finite element program carries out a number of analyses to examine the effect of element size on the solution. Using the ideas introduced above it is clear that the best approach is to solve the same model with two levels of finite element discretization or with the same discretization but using linear elements in one case and quadratic elements in the other. Taking as an example the use of two levels of discretization, such that the element dimensions are halved in the second case, the case with the larger number of elements will have solution errors that are 4 times smaller (the errors in the fields evaluated by differentiation of the shape functions will be halved). Examination of the changes between the two solutions will give a good estimate of the discretization errors, but not, as pointed out in the previous section, any indication of the accuracy of the model.

This approach is very good for two dimensional discretizations, but in three dimensions the 8 fold increase in the number of nodes quickly becomes prohibitive for all but the simplest geometries. When increasing the overall discretization becomes too expensive it is necessary to carry out more trial analyses, in each case choosing particular regions of increased discretization to determine the sensitivity of the solution to the change in element size. An experienced user will have learnt how to minimize the number of trials as a result of carrying out this type of experiment on a number of different geometries. Unfortunately a little knowledge may be a dangerous thing! Even experts cannot rely completely on their past experience. Results must always be critically examined on the assumption that they are incorrect.

The adoption of error analysis techniques in finite element programs does not reduce the user's responsibility for the quality of the results.

## Boundary Conditions

---

Boundary conditions are used in several ways.

- They can provide a way of reducing the size of the finite element representation of symmetrical models.
- They are used to approximate the magnetic field at large distances from the model (far-field boundaries).
- They can provide the drive for a model, e.g. a voltage or current source.
- They can represent the material inside a boundary, e.g. a conductor with a small skin depth in an alternating field.

### Scalar Potential Boundary Conditions

Model symmetry and the symmetry of the fields are implied by the potential boundary conditions applied to the finite element model. The simplest types of boundary condition are shown in the following table (where  $\mathbf{n}$  is the normal unit vector to the surface being considered).

Boundary Conditions		
Magnetic Fields	Field Symmetry	Scalar Potential
TANGMAGN	$H \cdot \mathbf{n} = 0$	$\frac{\partial \phi}{\partial n} = 0$
NORMMAGN	$H \times \mathbf{n} = 0$	$\phi = 0$
NORMMAGP	$H \times \mathbf{n} = 0$	$\phi = \text{constant}$
POTENTIAL		$\phi = \text{function}$
Electric Fields or Current Flow	Field Symmetry	Scalar Potential
TANGELEC	$E \cdot \mathbf{n} = 0$	$\frac{\partial V}{\partial n} = 0$
NORMELEC	$E \times \mathbf{n} = 0$	$V = 0$
NORMELEV	$E \times \mathbf{n} = 0$	$V = \text{constant}$
VOLTAGE		$V = \text{function}$

Note that in the table,  $\phi$  refers to either the reduced or total scalar potential. Boundary conditions on the reduced scalar potential only affect the reduced field intensity.

The boundary conditions shown in the above table should only be applied to the exterior surfaces of the finite element model. In particular, NORMMAGN, TANGMAGN, NORMELEC and TANGELEC are ignored on internal boundaries.

In electrostatic fields and current flow, electrode surfaces will normally have assigned potential boundary conditions (**VOLTAGE**:  $V=value$ ). In current flow analysis it may also be necessary to assign

- non-zero derivative boundary conditions (**DVOLTAGE**:  $\partial V / \partial n =\text{value}$ ) to define impressed currents; or
- **MIXED** boundary conditions defined by  $V + a \frac{dV}{dn} = \beta$  to model polarization potentials produced by corrosion of metal surfaces.

The default condition that will always be applied if no boundary condition is specified on an exterior surface is:

Default Boundary Conditions		
Magnetic Fields	Field Symmetry	Scalar Potential
TANGMAGN	$H \cdot n = 0$	$\frac{\partial \phi}{\partial n} = 0$
Electric Fields or Current Flow	Field Symmetry	Scalar Potential
TANGELEC	$E \cdot n = 0$	$\frac{\partial V}{\partial n} = 0$

## Vector Potential Boundary Conditions

Model symmetry and the symmetry of the fields are implied by the potential boundary conditions applied to the finite element model. The simplest types of boundary condition are:

Boundary Conditions		
	Field Symmetry	Vector Potential
TANGMAGN, NORMELEC or PEC	$H \cdot n = 0$ $E \times n = 0$	$A \times n = 0$ $V=0$
NORMMAGN or TANGELEC	$H \times n = 0$ $E \cdot n = 0$	$(\nabla \times A) \times n = 0$ $\nabla V \cdot n = 0$

where **n** is the normal unit vector to the surface being considered. A non-zero value for the electric scalar potential  $V$  on an external surface can be used to drive current into a model.

The boundary conditions shown in the table above should only be applied to the exterior surfaces of the finite element model. The default condition that will always be applied if no boundary condition is specified on an exterior surface is:

Default Boundary Condition	
Field Symmetry	Vector Potential
$H \times n = 0$	$(\nabla \times A) \times n = 0$

## Nodal fields

The default boundary condition is only weakly satisfied even when it is assigned to a surface. ‘Weakly satisfied’ means that it is applied as an integral over the surface patch for each shape function sub-domain.

For example, examination of the field solution close to a surface with the default boundary condition will reveal that the tangential component of magnetic field is not zero. The magnitude of this component reflects the local accuracy of the solution unless it is the result of a modelling error (for example, in a reduced potential region if the conductor’s field does not have the same symmetry as the finite element model). On the surface itself however, the normal component is forced to be precisely zero by the Post-Processor.

The  $H \cdot n$  condition with a vector potential solution is strongly imposed. The potentials are prescribed to the values specified. However, even then the field solution may not be exactly as expected. If the surface normal direction is discontinuous (the normal direction is ambiguous at an edge for example) this will produce a solution which implies at least two possible values for the field at the edge.

The fields computed by taking derivatives of the finite element shape functions will be discontinuous, but in each element the potential boundary condition will be exactly specified. The field smoothing processes (nodal averaging, see the notes on accurate field computation in section [Accurate Fields \[page 634\]](#)) must take this into account, and hence the true boundary condition is forced at these surfaces to ensure accuracy is maintained as far as possible.

## Uniqueness

The total and reduced vector potential formulation does not require the vector or electric scalar potentials to be specified unless they are needed to represent the field symmetry of the model. The magnitudes of the potentials are automatically gauged.

## Voltage driven eddy currents

It is possible to calculate the current flow through a conductor driven by an applied voltage. The faces where voltages are applied (where currents enter and exit) must be external mesh surfaces. Normal electric boundary condition should be applied to these surfaces and the electric scalar potential should be set to define the driving voltage.

In some practical applications the conductor may not cross the mesh boundary, an approximation is then required to represent the connection of the input and output to a power supply. To achieve this, the faces where voltages are applied should be close to each other, separated by a small gap in the mesh. Tangential magnetic boundary conditions are required on all boundary surfaces of the gap.

## Open Boundaries

Electromagnetic fields are frequently not contained within a finite volume. In practice, at long distances from the device producing the field, the distribution will be modified by the local environment, but this will not effect the field close to the device. Except where the interaction with the far environment is of interest, the field from an isolated device is usually required. A simple finite element mesh obviously has a finite extent and applying either potential or derivative boundary conditions on the open boundaries will perturb the true infinite domain solution.

There are techniques that accurately model the infinite domain. On a convex outer surface a series of rings of elements with increasing size may be recursively generated automatically (ballooning) or a boundary integral solution for the exterior domain may be coupled to the interior finite element solution. These techniques are relatively expensive in three dimensions and approximate methods may be used instead. The approximate methods include matching a convex outer surface to finite elements that extend from the surface to infinity and which have appropriate decay functions, and the standard approach of extending the finite element mesh to a distance where the field truncation has no effect on the regions of interest.

It is recommended that the mesh be extended to a reasonable distance with either potential or derivative boundary conditions applied to the outer surface, so that the truncation has an insignificant effect on the region of interest. The effect of truncation can be estimated by observing the tangential component of the field on an open boundary with a derivative boundary condition. Half the field observed on such a boundary is being reflected back from the exterior. Combining this with knowledge of the probable decay in the exterior space will give an order of magnitude for the effect of the truncation on the regions of interest. In a particularly sensitive application a further test should be applied. Two models should be analysed, one using potential boundary conditions on the open boundary, the other using derivative conditions. These analyses represent the model as an infinite array of similar models with either the opposite or the same sign of field in alternate images. In general the two solutions will bound the correct answer. It is often found that the derivative condition is closer to the real solution. It would in fact be exact if the open boundary surface was a constant flux surface. Use of this simple approach gives an estimate for the effect of the far field boundary truncation and this is why it is recommended.

## Total and Reduced Potential at the Open Boundary

It is worthwhile considering the significance of the potential type interior to the open boundary surface. With a total potential interior to the open boundary surface both the conductor and iron fields will be reflected at the surface. On the other hand, if a reduced potential is interior to the open surface only the iron and eddy current fields will be reflected at the surface. This effect may be used to minimize the perturbation caused by the truncation.

For example, if the model consists of a small volume of iron, interior to a large conductor system, the reduced potential at the open surface will clearly give much smaller errors produced by the approximate open (far field) boundary condition. The relatively small field from the iron will be the only component reflected in the false open boundary.

An alternative example is one where the conductors are contained inside an iron shield. In this case the total field at the open boundary will be much smaller than the conductors' field (if the shield is effective), a total potential next to the open boundary surface will therefore give much less perturbation in the regions of interest.

## Thin Plates

The **THINPLATE** boundary condition enables the Opera-3d Magnetostatics solver to analyse structures formed from thin plates. It is not strictly a boundary condition, but it uses the procedures in the Modeller that are used for defining boundary conditions. Thin plate models can only be created using the Modeller.

### Using Thin Plates

Instead of creating bodies to represent the material, the user should identify the faces of the material on either side of the thin plate object with a boundary condition label. The **THINPLATE** boundary condition can then be applied to that label. The data associated with the **THINPLATE** boundary condition consists of the material label and the plate thickness, which should be small compared to the element size in the surrounding cells. The material label must then be associated with a soft ferromagnetic material which can be linear or nonlinear. Laminated materials can also be used. In this case a volume property label must also be specified in order to supply the packing factor. The laminations will be parallel to the boundary condition surface.

It is also possible to model a thin gap using a thin plate made of air, surrounded by ferromagnetic material.

During analysis, additional elements are created on each element face on the thin plate boundaries. Different potential and field values are calculated for each side of the plate.

In post-processing, thin plates are included in the default selection for display. The field values displayed on the surface of the thin plates are the values in the plate material. Different values are displayed for each side of the plates. The **THREED** command is the only way to see the field strength (**H**) and flux density (**B**) in the plate material. In all other field calculation commands, the values in the surrounding materials are shown.

For more information, see the ***Opera-3d User Guide***.

### Restrictions

There are several restrictions on the use of thin plates:

1. The surrounding cells must be modelled using **TOTAL** scalar potential because the thin plate cannot coexist with a total-reduced potential interface.
2. Automatic potential cuts are disabled. It might be necessary to define non-zero potential boundary conditions if the thin plates surround a non-zero current.
3. Thin plate materials cannot be fully anisotropic or permanent magnets.

## Surface Impedance Boundary Condition

The surface impedance boundary condition (SIBC) is an approximation that simplifies the treatment of good electrical conductors in Harmonic EM and Harmonic HF. It can also be used in Transient EM solvers (including Motional EM and Magnetization) in situations where the electromagnetic fields experienced by the conducting material are sinusoidal. The approximation permits calculations that would otherwise be impractically large.

SIBC applies to good electrical conductors for which the electrical conductivity and magnetic permeability are linear and isotropic (for instance, a block of copper), or may be assumed so. Nonlinear materials also be modelled using SIBC in transient analyses.

### SIBC Theory

If a bulk volume of good electrical conductor is placed in an oscillating electromagnetic field, the field in the conductor decreases exponentially with the distance from its surface.

The characteristic length scale for the decrease is the skin depth  $\delta$ . In SI units, it is given by the expression

$$\delta = [2 / (\mu\sigma\omega)]^{1/2}, \quad (6.92)$$

in which  $\mu$  is the magnetic permeability,  $\sigma$  is the electrical conductivity, and  $\omega$  is the angular frequency ( $2\pi$  times the frequency).

If the skin depth is commensurate with the size of the conductor, the appropriate modelling approach is to use a conventional volume mesh with a mesh size of  $\delta/3$  or finer.

If the electrical conductivity and frequency are sufficiently large, the skin depth is a small quantity: indicative values are in mm in low-frequency machines, and in  $\mu\text{m}$  at radio frequencies. In this regime, the use of a conventional volume mesh can become impractical because of the very fine mesh that is needed.

SIBC overcomes this difficulty because it uses only the fields at the surface of the conductor, and there is no need for a fine mesh within the conductor. SIBC is valid if the skin depth is much smaller than all other relevant length scales, namely:

- the thickness of the conductor,
- the radii of curvature of the surface, and
- the tangential length scale of the variation of the field at the surface.

If these conditions are satisfied, SIBC gives good engineering accuracy (typical relative error of a few per cent).

### SIBC Limitations

SIBC does not apply at sharp edges and corners because the surface geometry is discontinuous. However, the error is limited to the vicinity of the discontinuity, and the solution elsewhere retains its accuracy.

The surface elements used by SIBC can only model currents which flow parallel to the surface of the conducting material. This means that it is not possible to model currents which cross the surface. An electrically insulating boundary condition (EIBC) can be used between a conducting meshed material and a conducting material using SIBC. The following table indicates the combinations of adjacent conducting materials which are allowed.

<b>Material 1</b>	<b>Material 2</b>	<b>Boundary condition</b>	<b>Allowed?</b>
meshed	meshed	none	yes
meshed	meshed	EIBC	yes
meshed	SIBC	none	warning
meshed	SIBC	EIBC	yes
SIBC	SIBC	none	no
SIBC	SIBC	EIBC	SS only

The user must be aware that the SIBC is valid only when the skin depth is sufficiently small. The software does not validate its applicability; the user must check that it is a reasonable approximation to use for the calculation in hand.

## SIBC Parameters

The SIBC requires minimal extra input from the user. The geometry of the conductor (including a target mesh size) and its material properties (linear permeability only) are set in the usual way.

However, the SIBC is applied in a different way from other boundary conditions. It is a material property, set using [The MATERIALS Command \[page 248\]](#) and it applies to all surfaces of a material.

If the SIBC is not used, the conductor is treated in the usual way, and the target mesh size must satisfy the  $\delta/3$  criterion. If the SIBC is used, the target mesh size must resolve the surface variation of the field (a much weaker requirement); the internal mesh is not relevant, and the internal fields are assumed to vanish.

The user may switch between `usesIBC=no` and `usesIBC=yes` in the same model, so that a single model may be used for a sweep over a range of frequencies for which the skin depth changes from large to small.

For transient analysis, 3 additional parameters must be set.

1. Frequency: a value must be provided.  
Transient analysis can model variations in time which are much more complex than a single frequency oscillation. However, as stated above, the skin depth of a material is defined as a function of frequency. A typical fundamental frequency must be provided for each material. For a non-moving transient analysis, this will be the principal frequency of the drive; for motional analysis, the speed of rotation of magnetic poles will determine the frequency of the applied field experienced by the parts of a model which use SIBC.
2. Order: the default value is 1.

The order specifies the number of harmonics of the frequency which will be used. Only odd harmonics are included, so an order of 3 will use the fundamental and the 3rd and 5th harmonics. Increasing the order enables a larger spread of frequencies to be modelled but also increases the solution time. A value of 3 or more is recommended for nonlinear materials.

3. Operating value of B: the default value is 0.

Skin depth is also a function of permeability so for a nonlinear material, the skin depth varies with the magnitude of the applied field. The SIBC algorithm needs to know a typical value of flux density in order to calculate a typical skin depth. A better estimate will help the nonlinear iterations converge well. The recommended value is the the flux density at which the material saturates.

The values of frequency and B are not critical but supplying values in the right range will enable the SIBC algorithm to obtain the best solution most efficiently.

## SIBC Post-Processing

The standard post-processing options for force, power and energy apply to conductors that use the SIBC. The fields on the surface of the conductor may be displayed in the post-processor in the usual way.

## Current Source Boundary Condition

The Current Source boundary condition is available in Current Flow, Electromagnetic (including Magnetization but not Fixed Velocity) and Harmonic High Frequency solvers. In the solution, the total current flowing across all boundaries with the same boundary condition label will be the specified value of current.

A positive value for the **CURRENTSOURCE** means that a positive current enters the volume. A negative value represents a current flowing out of the volume.

If a model has matching input and output current source boundary conditions and no normal electric or assigned voltage conditions, one of the current source boundaries should be changed to an assigned zero voltage boundary condition so that the value of voltage is gauged in the finite element model.

## Bulk Conductor Terminal

The **TERMINAL** boundary condition identifies the faces as a terminal of a bulk eddy current conductors. The data associated with the boundary condition are the polarization of the terminal and the conductor name.

Polarizations can be: **POSITIVE**, **NEGATIVE** or **BOTH**. Current flows from a **POSITIVE** terminal to a **NEGATIVE** terminal. The option of **BOTH** indicates that the positive and negative terminals are on the same faces. The current flows in the direction of the normal to the face.

The terminals are constant voltage surfaces and therefore the current flow is normal to the faces.

## Electric Insulator Boundary Condition

The Electric Insulator boundary condition (**ELECTRICINSULATOR**) is available in Electromagnetic, Magnetization and Current Flow solvers. It can be applied to an internal surface between two regions of conducting material. In the solution, there will be no current crossing such a boundary.

Care should be taken when another conducting material connects cells separated by an **ELECTRICINSULATOR** boundary. There is no problem if the other material has a similar conductivity to that of the cells separated by the boundary; however if the other conductivity is non-zero but significantly smaller, the insulating boundary should be modelled using a thin cell of non-conducting material.

## External Forces and Constraints

External forces and constraints can be applied to Static Stress analyses. The available forces (**STRESSLOAD**) are:

- **PRESSURE**, a force normal to the surface, and
- **TRACTION**, a force in any direction.

The available constraints (**STRESSCONDITION**) are:

- **FREE** to move;
- fixed:
  - **FIXED** in all 3 directions;
  - fixed in tangential directions (**TANGFIXED**);
  - fixed in the direction normal to the surface (**NORMFIXED**);
- defined **NORMALDISPLACEMENT**;
- defined **DISPLACEMENT** in all 3 directions.

## Thermal Boundary Conditions

Thermal boundary conditions are described in

- **Boundary Conditions for the Static Thermal Solver [page 592]** and
- **Boundary Conditions for the Transient Thermal Solver [page 595]**.

## Time-Stepping

The transient analysis programs (Transient EM, Motional EM, Magnetization, Quench and Transient Thermal) analyse fields which are time dependent and not necessarily time harmonic. Furthermore, in the case of Motional EM, parts of the model might also be moving. The resulting equations must be solved using a time-stepping algorithm.

### Time-Stepping Equations

Applying the Galerkin procedure to equation (6.65) produces a matrix equation of the form

$$RA + S \frac{\partial A}{\partial t} + b = 0 \quad (6.93)$$

where **A** is now a vector of unknown potentials and **b** is a vector of driving terms. Discretizing **A** and **b** as first order functions in time:

$$A(t) = (1 - \tau)a_n + \tau a_{n+1} \quad (6.94)$$

$$B(t) = (1 - \tau)b_n + \tau b_{n+1} \quad (6.95)$$

where

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \quad (6.96)$$

and  $a_n$  and  $b_n$  are values of **A** and **b** at time  $t_n$ . Using  $\tau$  as the weight in a Galerkin weighted residual solution of (6.93) leads to a recurrence relationship between  $a_{n+1}$  and  $a_n$ :

$$\left( R(1 - \theta) - \frac{S}{\Delta t} \right) a_n + \left( R\theta + \frac{S}{\Delta t} \right) a_{n+1} + b_n(1 - \theta) + b_{n+1}\theta = 0 \quad (6.97)$$

where the time-step  $\Delta t = t_{n+1} - t_n$ . The value of  $\theta$  is chosen as appropriate for each analysis type. In general,  $\theta=2/3$ , but for models with moving parts or circuits a value of 1 is used.

A similar formulation is used in the thermal solvers.

### Transient Driving Functions

The time varying drive is provided by

- source currents,  $J_s$  in Biot-Savart (defined current density) conductors;
- current or voltage sources in circuits<sup>1,2</sup>;
- a uniform external field;
- non-zero potential boundary conditions, or

<sup>1</sup>If a functional voltage or current is defined in a circuit component, DC is the appropriate choice for the transient drive.

<sup>2</sup>Circuits created by the Circuit Editor always use functional drives with sinusoidal drives being  $\cos(2\pi ft - \phi)$ .

- a table of heat values in elements.

In each case (except for the table of heat values), the value of the drive at any time is found by multiplying by a time function. The available drive functions are as follows:

Transient Drive Functions	
Name	Function
COSINE	$t < 0, F = F(0)$ $t \geq 0, F = \cos((2\pi ft - \phi))$
DC	Uniform in time for all time.
FUNCTIONAL	$t < 0, F = 0$ $t \geq 0, F$ calculated from algebraic expression.
PEAK	$t < 0, F = 0$ $t \geq 0 : F = ate^{(-t^2/b)}$ $a = \frac{e^{1/2}}{t_c}, b = 2t_c^2$
RAMP	$t < 0, F = 0$ $t \geq 0, t \leq t_c : F = \frac{t}{t_c}$ $t \geq t_c, F = 1$
RISE	$t < 0, F = 0$ $t \geq 0 : F = 1 - e^{(-t/t_c)}$ .
SINE	$t < 0, F = F(0)$ $t \geq 0, F = \sin((2\pi ft - \phi))$
STEP	$t < 0, F = 0$ $t \geq 0, F = 1$
TABLE (switch on)	$t \leq 0, F = 0$ $t > 0: F$ cubic splines from tabulated data in a time-table file.
TOFF (switch off)	$t < 0, F = F(0)$ $t \geq 0 F$ cubic splines from tabulated data in a time-table file. $F(0)$ is the value of the function in the table file at time $t=0$ .

The driving function can be applied to all the source conductors, external fields and boundary conditions, or the drive label can be used to specify different driving functions for different conductors, external fields and boundary conditions. A phase angle can also be used in the sine and cosine functions.

The transient time table option allows the user to define driving functions other than those programmed into the analysis code. The tables consist of files containing pairs of numbers in free format, one pair per line. The first number on each line specifies the time; the second gives the function value. To check the shape of the drive function, the **GRAPH** command in the Post-Processor can be used to display the file contents.

Beyond the last value of time in the table, the function continues with the last value computed.

For the time table **TABLE** "Switch On" option, the values of time should start at zero and increase through the file. Discontinuities in function value or first derivative can be forced by specifying two entries for the same value of time. For the time table **TOFF** "Switch Off" option, the value of the function is set to the value of the drive function in the time table at  $t=0$  for all time up to time zero.

Other functions (**DC**, **COSINE**, and **SINE**) will also have function values that are non-zero at time zero (depending on the phase angle for sine and cosine drives).

## Time-Step Control

The time-stepping is controlled by parameters of The ANALYSISDATA Command [page 122], which control the transient time **UPDATE** method:

- **SIMPLE**: this uses a fixed time-step set by **DELTAT**. In nonlinear analyses a smaller time-step will be used if the nonlinear iterations fail to converge within **ITPTSTEP** iterations.
- **ADAPTIVE**: the time-step will be adjusted to achieve an error which is less than **MAXADERR** (expressed as a percentage). The initial time-step is set by **DELTAT**.

Many aspects of the time-stepping can be controlled more precisely and adjusted as the analysis proceeds using the methods described in Transient Analysis Logging and Control [page 627].

## Transient Analysis Logging and Control

The Opera-3d transient electromagnetic solvers have additional facilities to log time varying data and to control functional properties, circuits and the time-stepping method. For Motional EM this also extends to control of the motion of the sections of the model.

### Data Logging

When running a transient analysis, it is possible to log specific data at each analysis time. This allows tracking of key characteristics as a function of time in addition to the full solution at output times. Any expression can be logged. Typically these expressions would involve the system variables set by the analysis for the current and voltage in circuit elements, or the mechanical state of a Motional EM model.

### Circuit system variables

The current in a circuit loop, and the current and voltage in a circuit element are available as system variables. The following naming convention is used:

Circuit values	
<i>loopname_I</i>	Instantaneous current in a loop
<i>loopname_DIDT</i>	Time derivative of current in a loop
<i>component_I</i>	Instantaneous current in a component
<i>component_DIDT</i>	Time derivative of current in a component
<i>component_V</i>	Instantaneous voltage across a circuit element
<i>component_R</i>	Resistance of a circuit element which is a resistor

The voltage across a current source element is not available.

These variables can be logged to a data file, or can be used for as part of the control of the analysis.

### Additional variables

Some additional system variables are set by Quench Multiphysics for use during coupled analysis. They are also available in Quench Thermal and Transient EM and Thermal Transient solvers:

Simulation variables	Calculated by
TMAX, TMIN	Maximum and minimum temperature
TMAX_name	Maximum temperature
R_name	Wire resistance
QV_name	Quenched (Resistive) volume
JH_name	Joule heat power
EJ_name	Joule heat energy
ET_name	Stored thermal energy

The suffix, *name*, is the material label of a volume meshed conductor, which is same as the circuit element name of the conductor.

Wire resistance and quenched volume are only set for the materials which have wire material properties defined. Joule energy is set for either wire materials or conductive materials.

The value of wire resistance is calculated for the complete model including symmetry; the values of volume, energy and power only include the part of the model represented by the finite element mesh.

## Hysteresis energy loss

When hysteretic materials are used (see [Hysteretic materials \[page 146\]](#)), the hysteresis energy loss is accumulated for each material. The values are stored in system variables that have the form HLOSS\_name where *name* is the material name.

## Date and time

The variables for date and time are updated with the current values whenever they are referenced.

YEAR	the year number
MONTH	the month number
DAY	the day of the month
JDAY	the day of the year
HOUR	the number of whole hours since midnight
MINUTE	the number of whole minutes since the start of the hour
SECOND	the number of seconds since the start of the minute (to the nearest millisecond)
CPSECOND	the number of processor seconds since the start of the job

## Command File Control

The control is achieved through use of a command script. A specific name for this file is derived from the name of the ***op3*** database being solved. If the specific file does not exist, a generic file will be used instead allowing for a common control file for all databases in the same folder.

For a database file, ***filename.op3***, the following table lists the names of the specific and generic control command files for the different analysis programs.

Program	Specific command file	Generic command file
Motional EM	<i>filename_carmen.comi</i>	<b><i>control_carmen.comi</i></b>
Magnetization	<i>filename_demag.comi</i>	<b><i>control_demag.comi</i></b>
Transient EM	<i>filename_elektratr.comi</i>	<b><i>control_elektratr.comi</i></b>
Quench	<i>filename_quench.comi</i>	<b><i>control_quench.comi</i></b>
Transient Thermal	<i>filename_tempotr.comi</i>	<b><i>control_tempotr.comi</i></b>

This command file is executed several times through the course of a time-step. Key variables can be set within the script to alter the current behaviour, e.g. to change time-steps, force an output point etc. Additionally, key state values are set up by the system as system variables. These values can be used for conditional execution of commands, allowing switches in behaviour.

Additional user variables can be created in the command file to store values in the database for use in post-processing.

The name of the analysis database (***op3***) file is available in string variable **SOLVEFILENAME**.

Note that comparisons of numerical data in control files should be used with care. See the advice given in [Equality expressions \[page 32\]](#).

## Time-Step Summary

Time-stepping has 2 forms: simple or adaptive. For simple time-stepping, the solution progresses with a single solution at each time-step. For adaptive time-stepping, the solution consists of a single full step. The solution then repeats the time-step using 2 half steps and the difference between the single full step and the solution from the second half step used to estimate the error in the time-step solution. If this error is too high, the time-step will be repeated with a smaller time-step.

### Time-step stage

A string variable **TIMESTEP\_STAGE** is set before each call to the command script and allows the user to determine the current position through the time-step. For a time-step which starts at  $T_0$  and ends at time  $T_0 + \Delta T$ , the possible values for **TIMESTEP\_STAGE** are

<b>INIT</b>	The very first call, before time-stepping has started. $\text{TIME}=0$
<b>START</b>	The time-step is about to begin. $\text{TIME}=\text{T}_0$
<b>FULL</b>	The solution is for the full time-step. $\text{T}_0 \leq \text{TIME} \leq \text{T}_0 + \Delta\text{T}$
<b>FIRSTHALF</b>	The solution is for the first half step. $\text{T}_0 \leq \text{TIME} \leq \text{T}_0 + \Delta\text{T}/2$
<b>SECONDHALF</b>	The solution is for the second half step. $\text{T}_0 + \Delta\text{T}/2 \leq \text{TIME} \leq \text{T}_0 + \Delta\text{T}$
<b>END</b>	The solution has been completed. $\text{TIME}=\text{T}_0 + \Delta\text{T}$

The stage is critical when controlling different aspects of the time-step. Typically information can only be changed by the scripts run at the start or end of the time-step. The other stages are mainly for information. CARMEN will run the command script multiple times during these stages to allow an accurate calculation of position over the time-step.

### Time-step control at the **START** of a step

At the start of a time-step, the time-step method and step length can be changed.

- To change the **time-step method** reset the string variable **TIMESTEP\_METHOD** to

<b>SIMPLE</b>	Use simple fixed time-steps
<b>ADAPTIVE</b>	Use adaptive time-stepping

- The **time-step** can be overwritten by setting:
  - string variable **TIMESTEP\_DELTAT** to **OVERRIDE** and
  - the user variable **#TIMESTEP\_DELTAT** to the new step length. The existing value of time-step can be obtained through the system variable **TIMESTEP\_DELTAT**.
- The **minimum and maximum limits for a time-step** in adaptive time-stepping can be changed by setting:
  - string variable **TIMESTEP\_MINMAX** to **OVERRIDE** and
  - user variables **#TIMESTEP\_MIN** and **#TIMESTEP\_MAX**. The existing values are set in system variables **TIMESTEP\_MIN** and **TIMESTEP\_MAX**.

Once set, these limits persists for all subsequent time-steps.

- User variable `#Timestep_Next` can be set to force the adaptive time-stepping to include a particular value of time. This can be used to ensure that an output time exists or to mark a singularity in a driving function. (Some singular points in the drive functions, such as the top of a ramp drive, are included automatically.)

### Time-step control at the END of a step

At the end of the time-step, the following string variables will be set to indicate the default behaviour that will happen. Any of these string variables can be changed to **YES** or **NO** to force a different behaviour of the time-stepping. The behaviour of the string variables is listed below

<code>Timestep_Output</code>	<b>YES</b>	Output a simulation case with the solution at the current time.
	<b>NO</b>	Do not generate a simulation case at this time.
<code>Timestep_Final</code>	<b>YES</b>	This is the last solution and the analysis will terminate. Note, that this also forces an output point.
	<b>NO</b>	Continue time-stepping. This can allow outputs beyond the last time originally defined for the analysis.
<code>Timestep_Repeat</code>	<b>YES</b>	Repeat the current time-step. Note, <code>Timestep_Output</code> and <code>Timestep_Final</code> flags will be ignored.
	<b>NO</b>	Continue to the next time-step.

### Example time-step control command file

The following script is for a linear Motional EM analysis. The moving part keeps moving until it reaches an end-stop at a minimum z coordinate. It might reach this point at any time during a time-step. Time-step controls are used to repeat this step with a reduced step length, so that the time at which the end-stop is reached can be obtained more precisely.

Note that when a time-step is repeated, all system and numerical user variables are reverted to the values they had when the step was run the first time. This is why *string* variables (`STOPPED` and `ADJUSTNEXT`) are used in the command file because they retain the values assigned by the commands in the file. For example, `ADJUSTNEXT` is used to record the fact that the time-step and time-step limits should be adjusted at the start of the repeated step.

Note also the use of `%COMPARE` [page 38] to check the value of a string variable: it returns 0 if the strings match.

```
/ Motion control
```

```

$STRING MO_MOTIONCONTROL ACCELERATION
$CONSTANT #MO_MASS 0.25
$CONSTANT #MO_ACCELX 0
$CONSTANT #MO_ACCELY 0
$CONSTANT #MO_ACCELZ MO_FORCEZ/#MO_MASS

/ Limits to motion
$CONSTANT #MO_SHIFTZMAX 0
$CONSTANT #MO_SHIFTZMIN -0.0065

/ Time-step controls
$IF !%COMPARE(&Timestep_Stage&,INIT)
/ At start of time-stepping: initialize string variables.
$STRING STOPPED NO
$STRING ADJUSTNEXT NO
$END IF

$IF MO_SHIFTZ<=#MO_SHIFTZMIN&&!%COMPARE (&Timestep_Stage&,END)
/ It's reached the end stop and we are at the end of the time-step.

$IF !%COMPARE (&STOPPED&,NO)
/ The first time here: retry with a smaller step.
$STRING STOPPED RETRY
$STRING ADJUSTNEXT YES
$STRING Timestep_Repeat YES

$ELIF !%COMPARE (&STOPPED&,RETRY)
/ The second time here: force an output.
$STRING STOPPED YES
$STRING Timestep_Output YES
$END IF
$END IF

$IF !%COMPARE (&Timestep_Stage&,START) &&!%COMPARE (&ADJUSTNEXT&,YES)
/ At the start of a step ready to adjust the time-step and
/ minimum and maximum step lengths.
$STRING ADJUSTNEXT NO
$STRING Timestep_DeltaT OVERRIDE
$STRING Timestep_MinMax OVERRIDE
$CONSTANT #Timestep_DeltaT Timestep_DeltaT/10
$CONSTANT #Timestep_Min Timestep_DeltaT/20
$CONSTANT #Timestep_Max Timestep_DeltaT/10
$END IF

$IF !%COMPARE (&STOPPED&,YES) &&!%COMPARE (&Timestep_Stage&,START)
/ Beyond the end-stop: release the maximum step length.
$STRING STOPPED END
$STRING Timestep_MinMax OVERRIDE
$CONSTANT #Timestep_Max 1.E5
$END IF

```

## Restarting Transient Analyses

If an initial run does not continue for long enough, a transient analysis can be restarted. A new simulation can be added using [The COPYCASE Command \[page 733\]](#) in the Post-Processor, which can

set a new final value of time. Additional output points can be included using [Command File Control \[page 629\]](#) and in particular, [Time-step control at the START of a step \[page 630\]](#).

For example, if the initial analysis ran to time 0.1s and it is necessary to restart it to run to 0.5s with outputs every 0.02s, the following steps will give the required restart analysis:

1. In the Post-Processor, open the database and select the simulation at 0.1s.

2. From the console, run the command

```
COPYCASE NEWCASE=0.5 CONTINUE=YES
```

3. Close the Post-Processor

4. Add the following lines to the control file:

```
$if !%COMPARE(&TIMESTEP_STAGE&,INIT)
    / Set string variable at first "start of time-stepping"
    $string Firstrun yes
$end if

/ #dt is the step between output points
$constant #dt 0.02
/ #tl is the tolerance on an output point
$constant #tl 0.0001
/ set the next output to be an integer
/ multiple of #dt
$constant #timestep_next (int(time/#dt)+1)*#dt

$if !%COMPARE(&Firstrun&,yes)
    $string Firstrun no
$else
    / if time is an integer multiple of #dt
    / output results
    $if abs(nint(time/#dt)-(time/#dt))
        $string timestep_output yes
    $else
        $string timestep_output no
    $end if
$end if
```

5. Run the analysis.

## Accurate Fields

The total and reduced potential formulation allows a magnetostatic model to be defined with many possible combinations of potentials. The only restriction is that the total potential should be used inside magnetic volumes and the reduced potential must be used inside volumes where currents are flowing. Apart from these restrictions, the distribution of potential types may be chosen to give optimum accuracy for particular applications. One aspect of this has been mentioned in section [Total and Reduced Potential at the Open Boundary \[page 618\]](#). The following table identifies particular classes of model, and shows the recommended potential type in a particular volume of space.

In the table: 'Nodal' refers to the nodally averaged field output option; 'Integral' to the option where the field from the magnetic volumes are computed by evaluating the magnetization from the potential solution and then computing the field by volume integration of the magnetization.

<b>Model Description</b>	<b>Potential Type</b>	<b>Field Recovery</b>	
		<b>from mesh</b>	<b>from coils</b>
<b>Homogeneous Field: Conductor field &gt;&gt; Iron field</b>			
Close to the iron	Reduced	Nodal	Integral
Far from the iron	Any	Integral	Integral
<b>Homogeneous Field: Conductor Field almost = Iron Field</b>			
Homogeneous conductor field	Any	Nodal	Integral
Inhomogeneous conductor field	Total	Nodal	Integral
If total cannot be used	Reduced	Nodal	Integral
<b>Homogeneous Field: Conductor Field almost = -Iron Field</b>			
Homogeneous conductor field	Total	Nodal	Any
Inhomogeneous conductor field	Total	Nodal	Any
If total cannot be used	Reduced	Nodal	Integral
<b>Homogeneous Field: Conductor Field &lt;&lt; Iron Field</b>			
All situations	Any	Nodal	Integral
Peak Field on Conductor	Reduced	Nodal	Integral
<b>Inhomogeneous Field</b>			
Conductor Field >> Iron field	Reduced	Nodal	Integral
Conductor Field << Iron field	Any	Nodal	Integral
Far from iron	Any	Integral	Integral
Conductor Field almost = Iron field	Any	Nodal	Integral
Conductor Field almost = -Iron field	Total	Nodal	Integral

The table does not cover all eventualities and it may help to explain why the combinations are recommended. First, the use of total potential in volumes where the conductor and iron fields cancel is the basis on which the Magnetostatic solver algorithm was designed. This takes precedence over all other considerations. Finite elements are good at representing uniform fields and by implication, not so good for varying fields. If the combined conductor and iron fields produce a uniform resultant field, then it is best to use the total potential in such volumes. However, if the conductor field is much greater than the iron field, it is best to use a reduced potential in this volume because the conductor fields are essentially exact.

When results are displayed using the Opera-3d Post-Processor, the effects of the various types of field evaluation can be tested. The error caused by using interpolated conductor fields can be immediately compared against the more accurate but expensive conductor integration procedure. It is recommended that when peak fields on conductors or body forces on conductors are required, then conductor fields should be calculated by integration.

The fields computed by magnetization integration are a very specialized option. The primary role of this option is in applications such as shielded superconducting NMR systems. Field accuracy of the order of parts in  $10^6$  are required at some distance from the shield and where the shield contributes less than 10% of the total field. Reduced potential could be used in these volumes, providing the conductor fields are uniform. This can achieve an accuracy of 3 in  $10^5$ . Comparison with analytical solutions has showed that the magnetization integration can improve this to 3 in  $10^6$ . However this procedure is very computationally expensive.

## Magnetic Shielding

Many of the recommendations in the sections on accuracy apply to the calculation of electromagnetic fields in applications where shielding is the primary objective. The most important recommendations for shielding are that in the volumes where the shield is effective it is essential that the total potential option is used and fields are evaluated from the nodally averaged field.

When the shield is screening the outside world from the field produced by a device the above recommendation means that a total potential will be used in the exterior space. This is also important from the point of the approximate far field boundary condition that will be imposed at the boundary of the mesh. The total potential gives minimal perturbation of the interior solution.

Thin magnetic screens are often used at large distances from the device to be isolated. Elements with large aspect ratios can be used in such screens and will not adversely affect the solution accuracy. However, if a detailed evaluation of the field at a corner or next to a small gap is required, then the elements' dimension must be small compared to the feature being modelled. The restriction is perfectly reasonable, in any direction where the field is changing rapidly the element discretization must be capable of modelling the changes.

## Advanced Features

---

Most of the parameters which control the Opera-3d analysis programs can be set using the commands and dialogs of the Opera-3d/Modeller and Pre-Processor. There are a few adjustments which can be made by setting user variables to specific values before creating an ***op3*** database. These are options which the majority of users need never know about but which might need to be adjusted in a few cases. The following table shows the names of the user variables and what effects they have on the analysis programs.

Variable	Meaning		Default
#BICGSTABL	Sets the number of stabilization iterations used by BiCGStab. Must be in the range 1 to 10.		4
#ITSOLTYPE	Sets the algorithm to be used for solving the matrix equations (linear algebra) in all analysis programs except modal solvers.		
1	ICCG (all except Fixed Velocity Electromagnetic solver)		
2	DCG		
3	BiCGStab (all analysis programs)	in Fixed Velocity Electromagnetic solver	
4	QMR (all analysis programs)	in all other solvers	
5	SLUCG (all except Fixed Velocity Electromagnetic and all harmonic solvers)		
9	SSORCG ((all except Fixed Velocity Electromagnetic and all harmonic solvers)	in Charged Particle, Quench, all Thermal and statics solvers	

Variable	Meaning		Default
#ITSOLTYPE_COILPOTJUMP	Sets the algorithm to be used for solving the matrix equations (linear algebra) which calculate the potential jump from the coil fields.		SSORCG
	1	ICCG	
	2	DCG	
	3	BiCGStab	
	4	QMR	
	5	SLUCG	
	9	SSORCG	
#ITSOLVMAXIT	Sets the maximum number of iterations for the linear algebra.		Number of equations
#ITSOLVNOIMPROVE	Sets the maximum number of iterations to continue if the linear algebra fails to improve the solution.		#ITSOLVMAXIT/factor where the factor is 1000 for BiCGSTAB and 200 to QMR. The factor is halved for harmonic solvers.
#ITSOLVRETRY	Sets the maximum number of times to restart the linear algebra if the number of iterations in which the solution does not improve reaches #ITSOLVNOIMPROVE.		2
#ITSOLVTOL	Sets the convergence tolerance for the linear algebra. Must be in the range $10^{-4}$ to $10^{-14}$ . In nonlinear analyses, the linear algebra calculates changes in the solution. The tolerance is automatically increased as the changes in solution get smaller so that the tolerance relative to the total solution remains the same.		TOLERANCE^2/100
#ITSOLVTOLLOOSE	Sets a factor defining a looser tolerance. If convergence to a tolerance of #ITSOLVTOL fails but convergence to #ITSOLVTOL*#ITSOLVTOLLOOSE succeeds the analysis will be allowed to continue but a warning will be issued in the output file (*.res).		100

Variable	Meaning	Default
#ITSOLVTOLNR	Sets the convergence tolerance for the linear algebra during nonlinear iterations. This can be used to set a larger tolerance than that given by the automatic adjustment of #ITSOLVTOL described above.	
#LIMITEDGEPJ	During the coil field calculations within the Electromagnetic solvers, a potential jump on each edge on the reduced/total interface surface is calculated. This calculation involves a matrix solution to find the set of potential jump values which minimizes the errors in the coil fields. The matrix is constructed from a set of terms for the faces of elements in the reduced potential region. Each of these terms is weighted by an estimate of its accuracy, so that some terms will have a greater influence on the error. #LIMITEDGEPJ defines the minimum weighting by which any term will be scaled. By lowering the weighting, the variation of magnitude of entries within the matrix increases, and this makes the iterative matrix solution slower and less stable. In most cases this does not need to be adjusted, and will have very little influence on the solution. The extreme case for which it was intended was where the flux through the interface surface was very small compared to fluxes elsewhere within the model, where it was changed down, in that case to 1e-7.	1e-6, range 1e-2 to 1e-10
#MAXEDGEHDLPTS	Sets the maximum number of additional field points used to calculate $\int H_s \cdot dI$ on the interface between TOTAL and REDUCED potentials. Must be an integer power of 2 in the range 8 to 1024. If it is necessary to increase this, there will be a message in the <b>res</b> file.	8

Variable	Meaning	Default
#MAXTRACKLEN	Reduces the maximum number of points stored in a SCALA trajectory. It cannot be used to increase the maximum number.	200000
#USEFULLMATRIX	If equal to 1, it makes solvers use a full asymmetric Jacobian matrix. This can help convergence of nonlinear models with anisotropic materials. Not available in Charged Particle, Fixed Velocity Electromagnetic and all stress solvers.	0

User variables can be set using the **\$ CONSTANT** command (see [Numerical variables \[page 56\]](#)).

- Example:  
`$ constant #itsoltype 3`

## Solution Fields and Field Averaging

---

During the solution in the Analysis Program a set of primary fields is computed from the matrix solution potentials as part of the final part of a simulation stage. From these primary fields a set of secondary fields is then derived. Which fields are available depends on the Analysis Program and these are detailed in the table below.

<b>Analysis</b>	<b>Primary Field Quantity</b>	<b>Secondary Field Quantity</b>
Magnetostatic	<b>H</b>	<b>B</b>
Electrostatic	<b>E</b>	<b>D, J<sup>1</sup></b>
Current Flow	<b>E</b>	<b>J, D<sup>1</sup></b>
Harmonic EM	<b>B, E, JM</b>	<b>H, J</b>
Transient EM	<b>B, E, JM</b>	<b>H, J</b>
Velocity EM	<b>B, E</b>	<b>H, J</b>
Motional EM	<b>B, E, JM</b>	<b>H, J</b>
Magnetization	<b>B, E</b>	<b>H, J</b>
Charged Particle	<b>E</b>	<b>D, J<sup>1</sup></b>
Quench Thermal	<b>DT</b>	<b>Q</b>
Harmonic HF	<b>E, B</b>	<b>D, H, J</b>
Modal HF	<b>E, B</b>	<b>D, H</b>
Static Stress	<b>Strain, Tstrain<sup>2</sup>, Gstrain<sup>3</sup></b>	<b>Stress</b>
Modal Stress	<b>Strain</b>	<b>Stress</b>
Static Thermal	<b>DT</b>	<b>Q</b>
Transient Thermal	<b>DT</b>	<b>Q</b>

---

<sup>1</sup>If Lossy Dielectric option is used

<sup>2</sup>If Thermal Expansion is included

<sup>3</sup>If General Expansion is included

# **Chapter 7**

# **Opera-3d Post-Processor**

## **Introduction**

---

The Opera-3d Post-Processor displays and performs further calculations on results the Opera-3d solvers. The analysis programs use finite elements to model three dimensional electromagnetic devices. The Post-Processor provides facilities to view the finite element data, with superimposed contours of results and to process and display the results calculated along lines or on two dimensional areas, in addition to specific functions such as body forces and trajectory calculations.

The program is used to display three dimensional finite element models from direct access database files created by the Modeller or Pre-Processor and analysed by the analysis programs. Alternatively, conductor only models can be evaluated by entering the conductor data into the Post-Processor directly using the **CONDUCTOR** command.

Many results databases can be available to the program (activated) at any one time. It is possible for the program to look for a field point in all the active files until it is found or to restrict its searching to one resident (loaded) file. This enables the calculation of particle trajectories which extend through a system of magnets which have been analysed separately. When activating a database a local coordinate system can be specified to enable the individual magnets to be orientated correctly with respect to each other. Reflection codes and symmetries can also be given to replicate the finite element mesh so that the complete model is available to Opera-3d even if only a small section of it was analysed and the rest was implied by boundary conditions.

Two methods of command and data entry are available:-

1. Menu system or GUI (Graphical User Interface) - command selection and data specification are carried out under mouse control. Refer to the User Guide for information on the GUI.
2. Command line input - command selection and data specification are carried out from the keyboard.

## Commands Issued at Startup

---

Commands can be run and files read as the Post-Processor starts. They can be specified by a command input file and as command line arguments. The sequence of operations is as follows:

1. Read the initial command file. The program looks for a file called opera.comi,, first in the current directory and then in the user's home directory, and if it exists, executes the commands it contains (see [Command Input Files \[page 61\]](#)).
2. Activate and load a **database** given on the command line followed by a default selection and refresh commands (see [The ACTIVATE Command \[page 679\]](#), [The LOAD Command \[page 770\]](#), [The SELECT Command \[page 793\]](#) and [The THREED Command \[page 823\]](#)).
3. Read a **command\_file** given on the command line.
4. Run other commands given on the command line.

See the "Opera Manager User Guide", *Running Outside the Opera Manager* for details of the command line syntax.

Operation 1 above is also run when the program is restarted with [The CLEAR Command \[page 720\]](#).

## Opera-3d Post-Processor Tabbed Menu Interface

The Opera-3d Post-Processor Tabbed Menu Interface has 2 Tabs of commands:

- **Work:** contains commands for database access, user defined variables, model display, global options and units.
- **Post-Processing:** contains commands for visualizing results and deriving further information.

**Display buttons** are displayed at the bottom of the window and include commands to manipulate the display and perform other common operations.

**Toolbar summary**

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
3d display of model and fields	Work	View			THREED
3d display of model and fields	Post Processing	Fields			THREED
3d display of model and fields	Display buttons				THREED
Add command files as toolbuttons	Work	Command Files	User Defined Toolbuttons		\$TOOLBUTTON
Apply cut plane to 3d display	Work	View			SELECT
Calculate and display particle current or flux density maps	Post Processing	Particle Beams	Trajectories		VIEW
Calculate cavity Q factor	Post Processing	Fields	Integrals		
Calculate element force density using Maxwell stress (all elements)	Work	Options	Field Options		SET
Calculate element force density using Maxwell stress (non-air elements)	Work	Options	Field Options		SET

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Calculate field quantities along a line	Post Processing	Buffers and Graphs	Buffer Generation		LINE, AXESVIEW, DATALINE
Calculate field quantities around a circle	Post Processing	Buffers and Graphs	Buffer Generation		CIRCLE, AXESVIEW, DATALINE
Calculate field quantities around a circular arc	Post Processing	Buffers and Graphs	Buffer Generation		ARC, AXESVIEW, DATALINE
Calculate field quantities at a point	Post Processing	Buffers and Graphs	Buffer Generation		POINT
Calculate field quantities on a cartesian patch	Post Processing	Fields	Patches		CARTESIAN, MAP
Calculate field quantities on a polar patch	Post Processing	Fields	Patches		POLAR, MAP
Calculate field quantities on a spherical surface	Post Processing	Fields	Patches		SPHERICAL, MAP
Calculate fields and save to a buffer	Post Processing	Tables	Process		TABLE
Calculate fields and save to another table file	Post Processing	Tables	Process		TABLE
Calculate flux linkage in a conductor	Post Processing	Fields	Integrals		FLUXLINKAGE
Calculate Lorentz element force density (non-air elements)	Work	Options	Field Options		SET
Calculate Lorentz force density (all elements)	Work	Options	Field Options		SET
Calculate Lorentz forces or voltages in conductors	Post Processing	Fields	Integrals		BODY
Calculate particle trajectories	Post Processing	Particle Beams	Trajectories		TRACK

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Calculate time-averaged force density	Work	Options	Field Options		SET
Change display colours	Display buttons				COLOUR
Change project folder	Work	File			\$PROJECTFOLDER
Change the current loaded case	Post Processing	Database			SIMULATION
Change the current loaded case	Work	File			SIMULATION
Change the function of the mouse buttons	Work	Options			MOUSE
Change the LCS of the model	Work	Options			SYMMETRY
Clear all data	Work	File			CLEAR
Close loaded database	Work	File			LOAD
Create a buffer from a text or csv file	Post Processing	Buffers and Graphs	Buffer Generation		LOGGEDDATAFILE
Create a new 8-node brick conductor section	Post Processing	Conductors	Create		CONDUCTOR
Create a new 20-node brick conductor section	Post Processing	Conductors	Create		CONDUCTOR
Create a new arc conductor	Post Processing	Conductors	Create		CONDUCTOR
Create a new bedstead conductor	Post Processing	Conductors	Create		CONDUCTOR
Create a new constant perimeter end conductor	Post Processing	Conductors	Create		CONDUCTOR
Create a new helical end conductor	Post Processing	Conductors	Create		CONDUCTOR
Create a new racetrack conductor	Post Processing	Conductors	Create		CONDUCTOR

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Create a new simulation in the database by copying the current one	Post Processing	Database			COPYCASE
Create a new solenoid conductor	Post Processing	Conductors	Create		CONDUCTOR
Create a new straight bar conductor	Post Processing	Conductors	Create		CONDUCTOR
Create and view animations	Work	Images			ANIMATION
Create flux tubes	Post Processing	Particle Beams	Trajectories		VIEW
Create table file for all element centroids	Post Processing	Tables	Create		TABLE
Create table file for all node coordinates	Post Processing	Tables	Create		TABLE
Create table file for element centroids in selected volumes	Post Processing	Tables	Create		TABLE
Create table file for fields at all element centroids	Post Processing	Tables	Create		TABLE
Create table file for fields at all nodes	Post Processing	Tables	Create		TABLE
Create table file for fields at coordinates in field buffer	Post Processing	Tables	Create		TABLE
Create table file for fields at element centroids in selected volumes	Post Processing	Tables	Create		TABLE
Create table file for fields at nodes in selection	Post Processing	Tables	Create		TABLE
Create table file for fields at points on selected surface	Post Processing	Tables	Create		TABLE

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Create table file for nodes in selection	Post Processing	Tables	Create		TABLE
Create table file for points on selected surface	Post Processing	Tables	Create		TABLE
Create table files	Post Processing	Tables	Create		TABLE
Create table of field quantities on a grid	Post Processing	Tables	Create		GRID
Cycle axis display mode	Display buttons				WINDOW
Deactivate loaded database	Work	File			LOAD
Default selection and refresh display	Work	View			SELECT
Default selection and refresh display	Display buttons				SELECT
Define and list tabular functions	Work	User Defined			\$FUNCTION
Define and list user strings	Work	User Defined			\$STRING
Define and list user variables	Work	User Defined			\$CONSTANT, \$MODELDIMENSION, \$PARAMETER
Display a surface with a constant field value	Post Processing	Fields			ISOSURFACE
Display fields on conductors	Work	Options	Field Options		SET
Display particle trajectories	Post Processing	Particle Beams	Trajectories		VIEW
Display patches or buffers on contour or vector maps	Post Processing	Fields			MAP

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Do not include element force density in field calculations.	Work	Options	Field Options		SET
Erase selected conductors	Post Processing	Conductors			CONDUCTOR
Export conductors to file	Post Processing	Conductors			CONDUCTOR
Export conductors to file (calculated current density)	Post Processing	Conductors			CONDUCTOR
Export conductors to file (defined current density)	Post Processing	Conductors			CONDUCTOR
Export conductors to file (instantaneous current density)	Post Processing	Conductors			CONDUCTOR
Export conductors to file (peak current density)	Post Processing	Conductors			CONDUCTOR
Export I-deas universal file of mesh and field quantities	Post Processing	Tables	Create		IDEAS
Field harmonics	Post Processing	Buffers and Graphs	Harmonics		FIT
Field Options	Work	Options	Field Options		SET
Fit Fourier series to values	Post Processing	Buffers and Graphs	Harmonics		FIT
Fit Legendre polynomials to values	Post Processing	Buffers and Graphs	Harmonics		FIT
Graph nonlinear magnetic material properties (BH)	Work	File			BHDATA
Graph particle trajectories	Post Processing	Particle Beams	Trajectories		VIEW

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Import conductors from file	Post Processing	Conductors			CONDUCTOR
Import fields at all nodes or elements	Post Processing	Tables	Process		TABLE
Import fields at subset of elements	Post Processing	Tables	Process		TABLE
Import fields at subset of nodes	Post Processing	Tables	Process		TABLE
Include coil current density ( $J_c$ ) in field calculations	Work	Options	Field Options		SET
Include element force density in field calculations	Work	Options	Field Options		SET
Integrate field quantities over selected surfaces	Post Processing	Fields	Integrals		SURFACE
Integrate field quantities over selected volumes	Post Processing	Fields	Integrals		VOLUME
Integrate forces on bodies enclosed by the selected surfaces	Post Processing	Fields	Integrals		INTEGRATE
Integrate volumes to compute energy, power and force	Post Processing	Fields	Integrals		ENERGY
Intersect particle trajectories with a patch	Post Processing	Particle Beams	Trajectories		VIEW
List active databases	Work	File			SHOW
List all simulations in loaded database	Work	File			SHOW
List currently loaded system variables	Work	Options			SYSVARIABLE
List data for selected conductors	Post Processing	Conductors			CONDUCTOR
List system variables in the database	Work	Options			SYSVARIABLE

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Load system variables from the database	Work	Options			SYSVARIABLE
Load the next case in the database	Work	File			SIMULATION
Load the next case in the database	Post Processing	Database			SIMULATION
Load the previous case in the database	Post Processing	Database			SIMULATION
Load the previous case in the database	Work	File			SIMULATION
Modify selected conductors	Post Processing	Conductors			CONDUCTOR
Open database (activate and load)	Work	File	Open		ACTIVATE, LOAD
Open Reference Manual	Work	File			
Open the Circuit Editor	Post Processing	Conductors			CEDITOR
Open the Command File Editor	Work	Command Files			
Perform arithmetic on table files	Post Processing	Tables	Process		ARITHMETIC
Plot a graph	Post Processing	Buffers and Graphs			BUFFER, LOGGEDDATAILE, AXESVIEW, DATALINE
Print	Work	Images	Screen Output		PRINT
Process table files	Post Processing	Tables	Process		TABLE
Refresh the view	Display buttons				THREED
Reload active database	Work	File			LOAD
Repeat selection and refresh display	Work	View			SELECT

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Run command file	Work	Command Files	Run Comi		\$COMINPUT
Run recent command files	Work	File	Open		ACTIVATE, LOAD
Run recent command files	Work	Command Files	Run Comi		\$COMINPUT
Save image to Clipboard	Work	Images	Screen Output		PICTURE
Save image to file	Work	Images	Screen Output		PICTURE
Select components for display and calculation	Work	View			SELECT
Select components for display and calculation	Display buttons				SELECT
Select conductors	Post Processing	Conductors			CONDUCTOR
Select the active field buffer, delete and rename buffers.	Work	Options			BUFFER
Set AC time (for steady-state solutions)	Work	Options			SET
Set field point options	Work	Options	Field Options		SET
Set field recovery method (nodal/integral)	Work	Options	Field Options		SET
Set model units	Work	Options			UNITS
Set particle trajectory options for magnetic and electric fields	Post Processing	Particle Beams			COMBINE
Set the title of the display	Work	View			TITLE
Show details of the loaded simulation	Work	File			SHOW

Tooltip	Tab	Group	Button Stack	Toolbutton	Command
Test format of file	Post Processing	Tables	Process		TABLE
Toggle conductor picking	Post Processing	Conductors			CONDUCTOR
Toggle outline view of model	Display buttons				WINDOW
Toggle solid view of model	Display buttons				WINDOW
Toggle visibility of conductors	Display buttons				CONDUCTOR
Toggle visibility of contour map	Display buttons				WINDOW
Toggle visibility of contour or trajectory scales	Display buttons				WINDOW
Toggle visibility of isosurface	Display buttons				WINDOW
Toggle visibility of trajectories	Display buttons				WINDOW
Toggle visibility of vector map	Display buttons				WINDOW
Toggle visibility of vectors	Display buttons				WINDOW
Update model symmetry	Work	Options			SYMMETRY
User defined toolbuttons	Work	Command Files	User Defined Toolbuttons		

## Commands for Keyboard Entry – Command Line

The following is a complete list of the commands which can be entered in response to the console prompt 'Opera-3d >'. Following sections contain complete descriptions of all the commands and sub-commands of the Opera-3d Post-Processor.

Command line entry is carried out in the console window.

- Commands to make database files available to Opera-3d Post-Processor:

<b>ACTIVATE</b>	Activate a result database file, specifying reflections, symmetries and local coordinate system.
<b>LOAD</b>	Make an active database file resident or close a loaded database.
<b>SIMULATION</b>	Load another simulation from the loaded database.
<b>SYMMETRY</b>	Change the symmetry of the loaded database.
<b>SHOW</b>	List the names and some details of the active database files.
<b>SYSVARIABLE</b>	Add, delete or list system variables (solution vectors).
<b>BHDATA</b>	List the BH tables used in the loaded file.

- Conductor definition and editing command:

<b>CONDUCTOR</b>	Define, modify, erase, list, import and export conductor data.
<b>CEDITOR</b>	View circuits, currents and voltages in the Circuit Editor.

- Units selection command:

<b>UNITS</b>	Select units for display and evaluation
--------------	---

- Display commands:

<b>SELECT</b>	Select objects to be displayed.
<b>THREED</b>	Displays <b>SELECT</b> ion of mesh and conductors with surface contours.
<b>WINDOW</b>	Show or hide parts of the display.

- Field evaluation and display commands:

<b>SET</b>	Set field calculation options and local coordinate system for definition of field points.
<b>POINT</b>	Evaluate expressions of field values at a point.

<b>LINE</b>	Evaluate expressions of field values along a line.
<b>ARC</b>	Evaluate expressions of field values along a circular arc defined by end points and centre.
<b>CIRCLE</b>	Evaluate expressions of field values along a circular arc defined by radius, angles and local coordinate system.
<b>PLOT</b>	Display the results from the <b>ARC</b> , <b>CIRCLE</b> or <b>LINE</b> commands. <b>PLOT</b> also calculates the line integral.
<b>FIT</b>	Fit Fourier series to <b>ARC</b> , <b>CIRCLE</b> or <b>LINE</b> results or evaluate fields on a sphere and fit Legendre polynomials.
<b>CARTESIAN</b>	Evaluate expressions of field values over a quadrilateral patch defined in cartesian (XYZ) coordinates.
<b>POLAR</b>	Evaluate expressions of field values over a quadrilateral patch defined in cylindrical polar (RθZ) coordinates.
<b>SPHERICAL</b>	Evaluate expressions of field values over a 4-sided patch defined in spherical polar (Rθφ) coordinates.
<b>MAP</b>	Display a contour map or histogram of fields or field vectors evaluated with <b>CARTESIAN</b> , <b>POLAR</b> or <b>SPHERICAL</b> commands. <b>MAP</b> also calculates the surface integral.
<b>BUFFER</b>	Select the active field buffer for <b>MAP</b> and <b>PLOT</b> . Rename or delete field buffers.
<b>ISOSURFACE</b>	Display a surface where the field component has a constant value.

- Graphing commands:

<b>AXESVIEW</b>	Create new axes or add a line to axes.
<b>DATALINE</b>	Create a line from a graphing buffer.
<b>DATAVECTOR</b>	Create a vector.
<b>DATAVECTORSET</b>	Create a set of vectors for display on a vector diagram.
<b>LOGGEDDATAFILE</b>	Create a graphing buffer from data in a text file or row by row.
<b>PROCESSLINE</b>	Fit a polynomial to a line.

- Field integration commands:

<b>INTEGRATE</b>	Integrate forces on <b>SELECTed</b> surface using Virtual Work or Maxwell Stress.
<b>SURFACE</b>	Integrate any field component on the <b>SELECTed</b> surface.

<b>BODY</b>	Integrate body forces and torques on conductors: $J \times B$ .
<b>FLUXLINGKAGE</b>	Calculate flux linked per turn in solenoid, racetrack and bedstead conductors.
<b>ENERGY</b>	Calculate the stored energy, Lorentz force and power loss.
<b>VOLUME</b>	Integrate any field component over the volume of a material.

- Particle tracking commands:

<b>TRACK</b>	Calculate charged particle trajectories.
<b>COMBINE</b>	Combine magnetic and electric fields for particle tracking.
<b>VIEW</b>	Display particle trajectory intercepts.

- Interface and utility commands:

<b>GRID</b>	Evaluate fields over a 3D grid and put answers in a file.
<b>ARITHMETIC</b>	Combine values from two table files.
<b>TABLE</b>	Input tables of coordinates, etc., and output tables of coordinates, fields, etc.
<b>IDEAS</b>	Write or append mesh and results to an I-deas Universal File
<b>GRAPH</b>	Draw graphs of data in text files.

- Program management commands:

<b>COLOUR</b>	Enquire and set colours for the display.
<b>TITLE</b>	Control screen titles.
<b>CLEAR</b>	Clear program data and re-initialize all commands.
<b>PICTURE</b>	Copy the picture to the clipboard or a bitmap file.
<b>ANIMATION</b>	Create or play back an animation file.
<b>PRINT</b>	Send the picture to a printer.
<b>MOUSE</b>	Set the action performed by each mouse button.
<b>GUIOPTIONS</b>	Set window preferences.

- Ending the program:

<b>END</b>	End Opera-3d Post-Processor.
------------	------------------------------

## Keyboard Shortcuts

The Tabbed Menu Interface (TMI) can be used with keyboard shortcuts which have been defined for some of the more commonly used functions. The full set of shortcuts is as follows.

A	Fields on an Arc
B	Integrals over Conductors
C	Cartesian Patch
D	3d Display
E	Energy, Power and Force
F	Force on Selected Surface
L	Fields on a Line
O	Fields on a Circle
P	Polar Patch
S	Select
T	Display Trajectories
Ctrl C	Copy to clipboard
Ctrl P	Print
Ctrl O	Open a file
Ctrl 0	Reset display to initial view
Ctrl 1	View from +X
Ctrl 2	View from -X
Ctrl 3	View from +Y
Ctrl 4	View from -Y
Ctrl 5	View from +Z
Ctrl 6	View from -Z
F5	Refresh
F6	Default selection and refresh display

## Field Component Evaluation

The command decoder used with Opera-3d Post-Processor is described fully in chapter "[Command Language](#)" on page 23. More details are given here concerning the way in which the Post-Processor makes use of the expression analyser to evaluate output field components.

Several commands, **THREED**, **FIT**, **MAP**, **POINT**, **PLOT**, etc., have parameters **COMPONENT**, **VX**, **VY**, and **VZ** to define scalar and vector field components to be displayed. Expressions can be used to define these output field quantities with the variables being System Variables, user constants, user parameters and the parameters of the commands.

The expression for **COMPONENT**, **VX**, **VY** and **VZ** used with any command becomes the default value for all the commands which use those parameters. The initial value for **COMPONENT** is **X** and **VX**, **VY** and **VZ** are initially set to **X**, **Y** and **Z**. (These are used as defaults because they are always available.)

The **\$ CONSTANT** and **\$ PARAMETER** commands (see [User Variable Commands \[page 55\]](#)) can be used to perform further calculations on the results of the commands.

## System Variables

System Variables hold the data which can be used for point values, line graphs, contours and further calculations in the Opera-3d Post-Processor. Some system variables, such as field point coordinates, are available for all analysis types but others only have values if they can be read from an analysis database. System variables of this second type are the results of analysis. For example, a Magnetostatic analysis returns the magnetic scalar potential, field strength, flux density and source field strength (a scalar and three vectors). Commands such as **THREED**, **MAP** and **PLOT** accessed via the GUI allow the user to select the commonly used system variables through pop-up menus. These are activated by the button on the right side of the **COMPONENT** entry box.

Most loading of system variables happens automatically within the program, variables being loaded when they are required. For example, for a Magnetostatic solution, the potential, field strength and flux density are loaded when a database is activated and loaded. The source field strength is loaded automatically if integral coil fields are selected in [The SET Command \[page 798\]](#).

However, the user might sometimes need a variable which is available in the database but has not yet been loaded, or, for reasons of efficiency, might want to delete some of the system variables which have been loaded. [The SYSVARIABLE Command \[page 814\]](#) provides this functionality as well as listing the variables in the database and program. The calculation of the values of the source current density vector for field points inside conductors is selected using [The SET Command \[page 798\]](#).

In post-processing results from the Harmonic solvers, the system variables represent the real and imaginary parts of complex quantities and additional expressions are defined to give the instantaneous values. For other types of analysis only the real parts are available.

When system variables are loaded the program follows this procedure:

- **Scalars**, for example the magnetic scalar potential, **POT**: The program reads the real part (**RPOT**). If the imaginary part (**IPOT**) is also available it will be read as well. The following expressions will be defined:

- Real scalars:

```
POT=RPOT
POT0=RPOT
POTP=0
```

- Complex scalars:

```
POT=RPOT*COST-IPOT*SINT
POT0=SQRT (RPOT^2+IPOT^2)
POTP=ATAN2D (-IPOT;RPOT)
```

where **COST** and **SINT** are calculated from the value of AC time defined by [The SET Command \[page 798\]](#). Phase angle is defined as the angle at which the quantity has its maximum value.

The following variables can be used by the user: **POT**, **RPOT** and (for steady-state ac analyses) **IPOT**, **POT0** and **POTP**.

- **Vectors**, for example the magnetic field strength, **H**: The program reads each component (real and imaginary parts if available) and defines the following expressions.

- Real vectors:

```
HX=RHX
HX0=RHX
HXP=0
```

- Complex vectors:

```
HX=RHX*COST-IHX*SINT
HX0=SQRT (RHX^2+IHX^2)
HXP=ATAN2D (-IHX;RHX)
```

and similarly for the Y and Z components. It also defines the magnitude, the dyadic modulus and the supremum<sup>1</sup>:

```
HMOD=SQRT (HX^2+HY^2+HZ^2)
H0=SQRT (RHX^2+RHY^2+RHZ^2+IHX^2+IHY^2+IHZ^2)
HSUP=SUP3 (RHX;IHX;RHY;IHY;RHZ;IHZ)
```

the cylindrical polar components:

```
R=SQRT (X^2+Y^2)
HR=(HX*X+HY*Y)/R
HT=(HY*X-HX*Y)/R
```

and spherical polar components:

```
RR=SQRT (X^2+Y^2+Z^2)
HRR=(HX*X+HY*Y+HZ*Z)/R
HTT=(HR*Z-HZ*R)/RR
HPP=(HY*X-HX*Y)/R
```

The full set of components of the complex vector **H** is:

<sup>1</sup>The supremum is the maximum length of the vector at any time around the ac cycle. If the x, y and z components of the vector have the same phase angle, the supremum is the same as the dyadic modulus; if they are out of phase, the supremum is somewhat less. In quasi-nonlinear ac analyses, the supremum of B is used to look up the permeability.

	real	imaginary	instantaneous	amplitude	phase angle
x	RHX	IHX	HX	HX0	HXP
y	RHY	IHY	HY	HY0	HYP
z	RHZ	IHZ	HZ	HZ0	HZP
cylindrical r	RHR	IHR	HR	HR0	HRP
cylindrical $\vartheta$	RHT	IHT	HT	HT0	HTP
spherical r	RHRR	IHRR	HRR	HRRO	HRRP
spherical $\vartheta$	RHTT	IHTT	HTT	HTTO	HTTP
spherical $\Phi$	RHPP	IHPP	HPP	HPPO	HPPP
modulus			H, HMOD		
dyadic modulus			H0		
supremum			HSUP		

- **Edge or Face Vectors,  $A_1$**  and  $E$ : The program interpolates the edge or face values to provide vector quantities, e.g.  $E_X$ ,  $E_Y$  and  $E_Z$  and  $E_MOD$ , at each field point. Real, imaginary and instantaneous values are available as for  $H$  above.
- **Tensors**, stresses and strains: The program stores 6 components of a real symmetric tensor, e.g. **RSTRAINXX**, **RSTRAINXY**, **RSTRAINXZ**, **RSTRAINYY**, **RSTRAINYZ**, **RSTRAINZZ**. Expressions are defined for the values of each component and the symmetric terms in the tensor, e.g:

STRAINXX=RSTRAINXX  
 STRAINXY=RSTRAINXY  
 STRAINYX=STRAINXY

The 3 invariants of the tensor are calculated, e.g. **STRAINI1**, **STRAINI2**, **STRAINI3**, as well as the mean, e.g. **MEANSTRAIN**.

The principal values (eigenvalues and eigenvectors) of the tensor are calculated, e.g. **STRAIN1**, **STRAIN1X**, **STRAIN1Y**, **STRAIN1Z**.

In the lists of variables below, only the stem names are given (e.g.  $H$ ); it should be understood that all the components are also implied.

When system variables are removed, they are marked as no longer being available within the expression analyser software.

**Unit conversion** is performed on all system variables so that they appear in the user's choice of units.

---

<sup>1</sup>Dyadic modulus of edge vector potential  $A_1$  has a non-standard name which is  $A_DYADIC$ , because the  $A_0$  conflicts with the Fourier series system variable  $A_0$  created by **The FIT Command** [page 747].

## System Variables Defined in the Software

Field point geometry	
X, Y, Z	Field point coordinates
R, TH, Z	Cylindrical polar field point coordinates
RR, TT, PP	Spherical polar field point coordinates
TX, TY, TZ	Tangential unit vector to lines
NX, NY, NZ	Normal unit vector to surfaces <sup>a</sup>
NODE	Number of the node nearest to the field point
ELEMENT	Element containing the field point
ELEMENTVOLUME	The volume of the element containing the field point
MATCODE	Material code
POTCODE	Potential code

Source Quantities		
JC	Source current density <sup>b</sup>	vector
HC	Coercive field	vector
VEL	Velocity	vector
CHARGE	Charge density	scalar

Element Force Densities		
EFD	Element force density <sup>c,d</sup>	vector

<sup>a</sup>On surfaces of the model this is the outward normal to the selected material surface. On field point patches, the direction depends on the ordering of the corner points.

<sup>b</sup>JC is only available if JCOIL has the value YES in the SET command.

<sup>c</sup>EFD is only available if ELEMENTFORCEDENSITY has a value other than NO in the SET command.

<sup>d</sup>Only instantaneous or time-average values are available, not real or imaginary.

## System Variables in Analysis Databases

### Multiphysics

In multiphysics analyses, the Post-Processor has access to fields of the currently loaded simulation and the fields of earlier simulations in the database.

### Charged Particle

<b>Charged Particle</b>		
V	Electric scalar potential	scalar
VINIT	Initial electric scalar potential without any space charge effects	scalar
E	Electric field strength	vector
D	Electric flux density	vector
HS	Source magnetic field strength	vector
RHO	Space charge density	scalar
BEAMB	Magnetic flux density from the particle beams (if selected in the Modeller)	vector
BEAMPOWER	Power density deposited by the particle beams in a secondary emitter	scalar
ERRD	Error in flux density	vector

### Current Flow

<b>Current Flow</b>		
V	Electric scalar potential	scalar
E	Electric field strength	vector
J	Current density	vector
ERRJ	Error in current density	scalar

In Current Flow post-processing, the magnetic field strength, H is also available if integral fields are selected ([The SET Command \[page 798\]](#))

## Electromagnetic: Harmonic, Motional and Transient

Harmonic, Transient and Motional Electromagnetic		
V	Electric scalar potential	scalar
A_	Magnetic vector potential	vector
DADT_	Time derivative of magnetic vector potential	vector
H	Magnetic field strength	vector
HS	Source magnetic field strength	vector
B	Magnetic flux density	vector
E	Electric field strength	vector
J	Current density (eddy currents and currents in meshed conductors)	vector
JM	Current density in meshed conductors	vector
ERRB	Error in flux density	scalar
ERRJ	Error in current density	scalar
HLOSS	Hysteresis energy loss (transient only)	scalar
ISSIBC	1 in elements of materials modelled with Surface Impedance Boundary Condition; zero elsewhere.	scalar
SIBCLOSS	The power loss density <sup>1</sup> in elements of materials modelled with Surface Impedance Boundary Condition.	scalar

## Electrostatic

Electrostatic		
V	Electric scalar potential	scalar
E	Electric field strength	vector
D	Electric flux density	vector
J	Current density in lossy dielectrics	vector
ERRD	Error in flux density	scalar

<sup>1</sup>The SIBC power loss density is calculated as a single value per element in elements on the surface of the material so that when intergrated over the volume, the best possible value for total loss is obtained.

## High Frequency: Harmonic and Modal

<b>Modal HF</b>		
A_	Edge values of magnetic vector potential	vector
B	Nodal values of magnetic flux density	vector
D	Electric flux density	vector
E	Nodal values of electric field strength	vector
H	Magnetic field strength	vector
<b>Harmonic HF</b>		
A_	Edge values of magnetic vector potential	vector
B	Magnetic flux density	vector
D	Electric flux density	vector
E	Electric field strength	vector
H	Magnetic field strength	vector
J	Current density	vector
ISSIBC	1 in elements of materials modelled with Surface Impedance Boundary Condition; zero elsewhere.	scalar
SIBCLOSS	The power loss density <sup>1</sup> in elements of materials modelled with Surface Impedance Boundary Condition.	scalar

## Magnetization

<b>Magnetization</b>		
V	Electric scalar potential	scalar
A_	Magnetic vector potential	vector
DADT_	Time derivative of magnetic vector potential	vector
H	Magnetic field strength	vector
HS	Source magnetic field strength	vector
B	Magnetic flux density	vector
MAGB	Maximum flux density over time	vector

<sup>1</sup>The SIBC power loss density is calculated as a single value per element in elements on the surface of the material so that when intergrated over the volume, the best possible value for total loss is obtained.

<b>MINB</b>	Minimum flux density over time	vector
<b>E</b>	Electric field strength	vector
<b>J</b>	Current density (eddy currents and currents in meshed conductors)	vector
<b>JM</b>	Current density in meshed conductors	vector
<b>ERRB</b>	Error in flux density	scalar
<b>ERRJ</b>	Error in current density	scalar
<b>HLOSS</b>	Hysteresis energy loss	scalar

## Magnetostatic

<b>Magnetostatic</b>		
<b>POT</b>	Magnetic scalar potential	scalar
<b>H</b>	Magnetic field strength	vector
<b>HS</b>	Source magnetic field strength	vector
<b>B</b>	Magnetic flux density	vector
<b>ERRB</b>	Error in flux density	scalar

## Quench

<b>Quench</b>		
<b>T</b>	Temperature	scalar
<b>DT</b>	Temperature gradient	vector
<b>Q</b>	Heat flux	vector
<b>JHD</b>	Joule heat density	scalar
<b>HS</b>	Magnetic field strength	vector
<b>DBDT</b>	Time derivative of magnetic flux density	vector
<b>ERRQ</b>	Error in heat flux	scalar

## Stress: Modal and Static

<b>Modal and Static Stress</b>		
<b>DISP</b>	Displacement	vector
<b>STRAIN</b>	Total strain	tensor

<b>TSTRAIN</b>	Thermal strain	tensor
<b>GSTRAIN</b>	General strain	tensor
<b>ESTRAIN</b>	Elastic strain, i.e. <b>STRAIN-TSTRAIN-GSTRAIN</b>	tensor
<b>STRESS</b>	Stress	tensor

## Thermal: Static and Transient

<b>Static and Transient Thermal</b>		
<b>T</b>	Temperature	scalar
<b>DT</b>	Temperature gradient	vector
<b>Q</b>	Heat flux	vector
<b>ERRQ</b>	Error in heat flux	scalar

## Additional Field Quantities

These are defined using algebraic expressions of system variables for specific simulation types. For time-harmonic simulations the values represent the time-average quantities.

<b>Additional field quantities</b>		
<b>LOSS</b>	Power loss density $J \cdot E$	Current Flow, Electromagnetics and Harmonic High Frequency solvers <sup>1</sup>
<b>S</b>	Poynting vector $E \times H$	High Frequency solvers
<b>VONMISES</b>	Von Mises stress	Static Stress solver

## Maxwell Stress integrands

---

<sup>1</sup>In models with Surface Impedance Boundary Conditions, the expression for **LOSS** includes **SIBCLOSS**.

<b>Maxwell Stress integrands</b>		
<b>BDOTN</b>	Normal magnetic flux density	Any simulation with values of <b>B</b> and <b>H</b> and Coil only models.
<b>HDOTN</b>	Normal magnetic field strength	The torque action point is the origin. See <a href="#">Parameters for Maxwell Stress [page 671]</a> for the formulae used. The only difference is that the expressions used for these system variables include unit conversion factors so give the correct answers for any choice of units.
<b>MSX</b>	X component of Maxwell Stress tensor.	
<b>MSY</b>	Y component of Maxwell Stress tensor.	
<b>MSZ</b>	Z component of Maxwell Stress tensor.	
<b>MSTX</b>	Maxwell Stress torque around X axis.	
<b>MSTY</b>	Maxwell Stress torque around Y axis.	
<b>MSTZ</b>	Maxwell Stress torque around Z axis.	

## Material properties at field point

Material properties, permeability, permittivity and conductivity can only be calculated from the appropriate field quantities. For example, the isotropic permeability is given by

`BMOD/HMOD`

## Solution values

These are set when eigenvalue, steady-state ac or transient results are activated.

<b>Solution values</b>		
<b>ANGLE</b>	Rotor angle	Motional EM models from Pre-Processor only
<b>FREQ</b>	Rotational frequency (rpm)	
<b>FREQ</b>	Frequency	
<b>TTIME</b>	Transient time	

The variable `CASES` holds the number of simulations in the active database.

## Circuit Values

The following system variables are set when a simulation with circuits is loaded. Variables are defined for each circuit loop and circuit element.

<b>Circuit values</b>	
<b>Harmonic Electromagnetic Solver</b>	
<i>loopname_I</i>	Instantaneous current in a loop
<i>loopname_RI</i>	Real part of current in a loop
<i>loopname_II</i>	Imaginary part of current in a loop
<i>loopname_IO</i>	Amplitude of current in a loop
<i>loopname_IP</i>	Phase angle of current in a loop
<i>elementname_I</i>	Instantaneous current in a circuit element
<i>elementname_RI</i>	Real part of current in a circuit element
<i>elementname_II</i>	Imaginary part of current in a circuit element
<i>elementname_IO</i>	Amplitude of current in a circuit element
<i>elementname_IP</i>	Phase angle of current in a circuit element
<i>elementname_V</i>	Instantaneous voltage across a circuit element <sup>a</sup>
<i>elementname_RV</i>	Real part of voltage across a circuit element
<i>elementname_IV</i>	Imaginary part of voltage across a circuit element <sup>a</sup>
<i>elementname_V0</i>	Amplitude of voltage across a circuit element <sup>a</sup>
<i>elementname_VP</i>	Phase angle of voltage across a circuit element <sup>a</sup>
<i>elementname_R</i>	Resistance of a circuit element which is a resistor
<b>Transient EM, Motion EM and Magnetization Solvers</b>	
<i>loopname_I</i>	Instantaneous current in a loop
<i>loopname_DIDT</i>	Time derivative of current in a loop
<i>elementname_I</i>	Instantaneous current in a circuit element
<i>elementname_DIDT</i>	Time derivative of current in a circuit element
<i>elementname_V</i>	Instantaneous voltage across a circuit element <sup>a</sup>
<i>elementname_R</i>	Resistance of a circuit element which is a resistor

<sup>a</sup>The voltage is not available for current sources. A high value resistor, parallel to the component, can be used to measure the voltage.

The name of the loop or element must not be greater than 40 characters long, and the voltage across a current source element is not available.

The current density in meshed conductors in circuits can also be displayed using the vector system variables **J** (which also includes eddy currents) and **JM** (which only includes the circuit currents).

## Local Coordinate System

The field vectors are defined with respect to the Global Coordinate System. The rotation matrix defined by the Euler angles of [The SET Command \[page 798\]](#) can also be used in the expressions. The local X component of the flux density is given by

**COMP=BX\*ROTL11+BY\*ROTL21+BZ\*ROTL31**, etc.

Local coordinate system	
<b>ROTL11, ROTL12, ..., ROTL33</b>	Rotation matrix to convert global field vectors to local coordinate system.

## Results of Commands

After the commands **BODY**, **ENERGY**, **INTEGRATE**, **MAP**, **PLOT**, **THREED** and **VIEW** system variables are updated with the results of the calculations.

Results of commands	Commands
<b>MINIMUM</b> , <b>MAXIMUM</b>	<b>MAP</b> , <b>PLOT</b> , <b>THREED</b>
<b>FX</b> , <b>FY</b> , <b>FZ</b>	<b>BODY</b> , <b>ENERGY</b> , <b>INTEGRATE</b>
<b>FMOD</b>	Magnitude of force: $\text{SQRT}(\text{FX}^2 + \text{FY}^2 + \text{FZ}^2)$
<b>TORQX</b> , <b>TORQY</b> , <b>TORQZ</b>	<b>BODY</b> , <b>INTEGRATE</b>
<b>TORQMOD</b>	Magnitude of torque: $\text{SQRT}(\text{TORQX}^2 + \text{TORQY}^2 + \text{TORQZ}^2)$

Results of commands	Commands
ENERGY, POWER, VOLUME, COENERGY, NLENERGY, ELECENER, HYSENERGY, HYSPOWER	ENERGY
INTEGRAL	MAP, PLOT, SURFACE, VOLUME
AREA	CONDUCTOR, SURFACE
VOLUME	VOLUME
J	Charged beam current density
PJ	Charged beam power density
TOF	Time of flight of particle
TXBEAM, TYBEAM, TZBEAM	Tangential unit vector to a trajectory
VELX, VELY, VELZ	Particle velocity
X0BEAM, Y0BEAM, Z0BEAM	Coordinates at centre of a beam
XSTART, YSTART, ZSTART	Coordinates at start of a trajectory

## Constants

Constants	
BOLTZMANN	Boltzmann constant, $k$
C	Speed of light, $c$

Constants	
ELECTRONCHARGE	Electron charge, $e$
ELECTRONENERGY	Electron energy, $m_e c^2$
ELECTRONMASS	Electron mass, $m_e$
EPSILON0	Permittivity of free space, $\epsilon_0 = 1/c^2 \mu_0$
GRAVITY	Acceleration due to gravity, $g$
MU0	Permeability of free space, $\mu_0$
PI	Pi, $\pi$
PLANCK	Planck's constant, $h$
PROTONMASS	Proton mass, $m_p$
PROTONMASSRATIO	Proton to electron mass ratio, $m_p/m_e$
STEFANBOLTZMANN STEPHANBOLTZMANN	Stefan-Boltzmann constant, $\sigma$
Z0FREE	Free space impedance, $\mu_0 c$

## User constants

User constants are defined and examined using the **\$ CONSTANT** command. They allow the current value of system variables or expressions to be stored for use in subsequent calculations (see [User Variable Commands \[page 55\]](#))

## User parameters

User parameters are defined and examined using the **\$ PARAMETER** command. They allow expressions to be stored. The value of user parameters is recalculated from the expression each time it is referenced using the current values of any other parameters or variables (see [User Variable Commands \[page 55\]](#))

## Expressions

Expressions cannot exceed 250 characters, since they cannot be continued on subsequent lines. Full details of the operators and functions allowed in expressions are described in [Expressions in Parameter Values \[page 30\]](#). The **\$ PARAMETER** command should be used to 'program' Opera-3d Post-Processor if more complicated expressions are needed.

## Examples

### Parameters for Maxwell Stress

The first example gives expressions to calculate the Maxwell stress on a surface using the integral given by [The MAP Command \[page 775\]](#) or [The SURFACE Command \[page 809\]](#). The force densities are given by #MSX, #MSY and #MSZ. The constants #X0, #Y0 and #Z0 should be set in advance to the action point for the torques, #MSTX, #MSTY and #MSTZ. The expressions given here assume that the units are set to SI (see [The UNITS Command \[page 835\]](#)).

```
$parameter hn hx*nx+hy*ny+hz*nz
$parameter bdoth bx*hx+by*hy+bz*hz
$parameter msx bx*#hn-nx*#bdoth/2
$parameter msy by*#hn-ny*#bdoth/2
$parameter msz bz*#hn-nz*#bdoth/2
$parameter mstx (y-#y0)*#msz-(z-#z0)*#msy
$parameter msty (z-#z0)*#msx-(x-#x0)*#msz
$parameter mstz (x-#x0)*#msy-(y-#y0)*#msx
```

The above commands are only valid if the field points used are all in air. If the integration surface passes through ferromagnetic material, the following set of commands should be used instead. These calculate the forces assuming an infinitely thin gap at the integration surface.

```
$parameter han (nx*bx+ny*by+nz*bz)/mu0
$parameter hanx nx*#han
$parameter hanx ny*#han
$parameter hanx nz*#han
$parameter hax hx-nx*(nx*hx+ny*hy+nz*hz)+#hanx
$parameter hay hy-ny*(nx*hx+ny*hy+nz*hz)+#hany
$parameter haz hz-nz*(nx*hx+ny*hy+nz*hz)+#hanz
$parameter hn #hax*nx+#hay*ny+#haz*nz
$parameter hm2 #hax*#hax+#hay*#hay+#haz*#haz
$parameter msx (2*#hax*#hn-nx*#hm2)*mu0/2.0
$parameter msy (2*#hay*#hn-ny*#hm2)*mu0/2.0
$parameter msz (2*#haz*#hn-nz*#hm2)*mu0/2.0
$parameter mstx (y-#y0)*#msz-(z-#z0)*#msy
$parameter msty (z-#z0)*#msx-(x-#x0)*#msz
$parameter mstz (x-#x0)*#msy-(y-#y0)*#msx
```

The program has system variables which can be used instead of this second set of user expressions: force densities MSX, MSY, MSZ (in units of force/length<sup>2</sup>) and torque densities MSTX, MSTY, MSTZ (in units of force/length). These system variables include unit factors to give the correct answers for any choice of units. (See [Maxwell Stress integrands \[page 665\]](#).)

### Constant for homogeneity

It is sometimes necessary to store the current value of a system variable or expression. The \$CONSTANT command copies the current value of a variable or expression into a user defined

of the flux density along a line the following set of commands could be used. First, the field at the reference point (10,0,0) is calculated and stored in a constant.

- Example - to calculate the homogeneity of flux density
- ```
point 10 0 0 by  
$constant #byrf by
```

The value of BY assigned to the constant #BYRF corresponds to value at the point (10, 0, 0). Next the homogeneity can be calculated.

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

## Post-Processor Data Files

The Opera-3d Post-Processor reads data from the direct access database files created by the Modeler or Pre-Processor and updated by the analysis programs. Other files can be created and read by the Post-Processor for use by the results display commands. See

- [The CONDUCTOR Command \[page 727\]](#) for details of conductor data files,
- [The GRAPH Command \[page 751\]](#) for graph files,
- [The IDEAS Command \[page 758\]](#) for Universal files,

and below for databases, grid files, table files and track files.

### Opera-3d Database Files

Opera-3d database files contain all the information used and calculated by the analysis programs. This includes the finite element mesh including boundary conditions, the material properties and BH curves, the conductors and solution options. These files can only be read by Opera-3d Post-Processor. [The SHOW Command \[page 804\]](#) can be used to list the contents of database.

### Writing to database files

Databases are opened by [The ACTIVATE Command \[page 679\]](#) and [The LOAD Command \[page 770\]](#) in read-only mode so that they are not unnecessarily changed by the Post-Processor. A database will be write-enabled by [The COPYCASE Command \[page 733\]](#), [The SYMMETRY Command \[page 811\]](#) or [The TABLE Command \[page 817\]](#) according to the following rules which depend on the version numbers of the database and Post-Processor:

| Version Numbers | Rule                                                                                                                                                                                                                                      |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| same version    | The database will be write-enabled when necessary with no questions asked.                                                                                                                                                                |
| database older  | The user will be asked before the database is write enabled, because this will update the version number of the database to that of the software. No question will be asked when running from a command file; permission will be assumed. |
| database newer  | The database will never be write enabled.                                                                                                                                                                                                 |

### GRID files

The **GRID** command writes a limited number of field values to a file to enable users to interface to their own post-processing software. The files can be text or binary. [See "The GRID Command" on](#)

page 754.

The following items are output: X, Y, Z and up to 9 other items which can be specified by the user.

In *text* files, table file format 1 or 2 is used (see [TABLE Files \[page 674\]](#)). Format 1 files contain values in internal units; format 2 data is in user units.

*Binary* files contain one record per field point with up to 12 DOUBLE PRECISION (8 byte) values in user units.

## TABLE Files

Table file format is used by the following commands:

|                    |                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ARITHMETIC</b>  | reads values from two table files and combines them, writing the answers to a third table file. See "The ARITHMETIC Command" on page 687. |
| <b>FIT</b>         | reads coordinates and field values from a file and fits a Fourier series. See "The FIT Command" on page 747.                              |
| <b>GRAPH</b>       | ignores the header records (types 1, 2 and 3) and reads the rest of the file to plot a graph. See "The GRAPH Command" on page 751.        |
| <b>GRID</b>        | writes coordinates and field values at a grid of points to a table file. See "The GRID Command" on page 754.                              |
| <b>MAP</b>         | reads coordinates and field values from a table file and draws a contour or vector map. See "The MAP Command" on page 775.                |
| <b>PLOT</b>        | reads coordinates and field values from a table file and plots a graph. See "The PLOT Command" on page 783.                               |
| <b>TABLE</b>       | read and writes table files. See "The TABLE Command" on page 817.                                                                         |
| <b>\$ FUNCTION</b> | read a function definition from a file (see User Function Command [page 60]).                                                             |

Table files are text files with formatted and free-format data. There are 3 different formats, versions 0, 1 and 2. The format number is specified on the first line of the table (record type 1). The default format for files created by the **GRID** and **TABLE** commands is 2.

### Record type 1

The first line of a table file contains the number of data points in the file and a format number. The number of points is specified as 3 numbers, so that 1, 2 or 3-dimensional arrays of points can be stored. For example, a total of 1000 points could be stored as a line with record 1 containing

1000 1 1 2

or as a 2-dimensional patch, with 40 points on side 1 and 25 points on side 2, indicated by

40 25 1 2

or as a 3 dimensional grid of points:

10 10 10 2

See [Record type 4 \[page 676\]](#) for information on the order of the points in 2-dimensional patches and 3-dimensional grids.

The 4<sup>th</sup> number on record type 1 holds the format number of the table file. Zero is assumed when the format number is missing.

The format number specifies how the data is laid out in the file and the units used for the data.

- **Format 0:** this format can no longer be written but files in this format can still be read.
- **Format 1:** the data in the file is in internal units (cgs). The unit expressions in the file are used to convert the data to internal units when the file is written and back to user units when data from the file is displayed.
- **Format 2:** the data in the file is in user units. The unit expressions in the file are used to convert the data to internal units when a file is read and back to user units when the data is displayed.

## Record type 2

The second record is repeated up to 12 times, once for each column in the file.

- **Format 0:** The record holds an integer (the column number), followed by a space, and then the column description of up to 80 characters. In most contexts, the string is compared to X, Y or Z to identify the coordinates, but otherwise is not decoded.  
The equivalent FORTRAN format is `(I2, 1X, A)`.
- **Format 1:** The record holds an integer and 2 character strings separated by one or more spaces. The integer is the column number (1 to 12). The first character string is used as the column name; the second is the unit expression in square brackets. In most contexts, the column name is compared to X, Y or Z to identify the coordinates, but otherwise is not decoded. The unit expression should be in terms of the unit conversion factors, e.g. `LENGU`, `FLUXU`, etc. (see [The UNITS Command \[page 835\]](#)).
- **Format 2:** The record holds an integer and 2 character strings separated by one or more spaces. The integer is the column number (1 to 12). The first character string is used as the column name; the second is the unit expression in square brackets. In most contexts, the column name is compared to X, Y or Z to identify the coordinates, but otherwise is not decoded. The unit expression should be in terms of unit names, e.g. `METRE`, `TESLA`, etc (see [The UNITS Command \[page 835\]](#)).

## Record type 3

The third record terminates the list of columns.

- **Format 0:** The third record has the same format as record 2 but the integer value must be zero. It indicates the end of the list of column headings and holds in the character string the unit name associated with the file. The units names which are recognized are: [CGS], [METRE], [MM], [MICRON] and [INCH].  
The equivalent FORTRAN format is (I2, 1X, A).
- **Formats 1 and 2:** The third record contains a single zero to indicate that there are no more column headings.

## Record type 4

The final record contains the data and is repeated for as many points as indicated by the product of the first 3 numbers in record 1. The order of the records is important. If the data represents a patch with 30x50 points (record type 1 of the table starts with 30 50 1) the patch is formed of 50 lines with 30 points on each line; the first 30 records of type 4 represent the first line, the second 30 records represent the second line, etc. There must be 50 sets of 30 records forming the complete patch.

Similarly, if the data represents a 3-dimensional grid of 20x30x40 points (record type 1 of the table starts with 20 30 40), the first 20 records of type 4 represent the first line of the grid; the first 30 sets of 20 records represent the first layer of the grid; there must be 40 sets of 600 records representing all the layers of the grid.

There should be up to 12 values on each record, as many as the column headings given by record 2.

- **Format 0:** no unit conversion is done except for the coordinates, X, Y and Z which are in the units specified on record 3.
- **Format 1:** all the values in the file are in internal units.
- **Format 2:** all the values in the file are in the units specified on the corresponding records of type 2.

The following is an example of a typical format 2 data file:

```

4 3 1 2
1 X [METRE]
2 Y [METRE]
3 Z [METRE]
4 TY*BZ-TZ*BY [TESLA]
5 TZ*BX-TX*BZ [TESLA]
6 TX*BY-TY*BX [TESLA]
0
0.10000 0.20000 0.12625 0.49232E-03 -0.12105E-03 0.13192E-03
0.20000 0.20000 0.12625 0.49232E-03 0.12105E-03 0.13192E-03
0.30000 0.20000 0.12625 0.16246E-02 -0.10484E-03 0.16246E-02
0.40000 0.20000 0.12625 0.16246E-02 0.10484E-03 0.16246E-02
0.10000 0.23000 0.12625 0.13180E-03 -0.11982E-03 0.49189E-03
0.20000 0.23000 0.12625 0.13180E-03 0.11982E-03 0.49189E-03
0.30000 0.23000 0.12625 -0.13180E-03 -0.11982E-03 0.49189E-03
0.40000 0.23000 0.12625 -0.13180E-03 0.11982E-03 0.49189E-03
0.10000 0.29000 0.12625 -0.16246E-02 -0.10484E-03 0.16246E-02
0.20000 0.29000 0.12625 -0.16246E-02 0.10484E-03 0.16246E-02
0.30000 0.29000 0.12625 -0.49232E-03 -0.12105E-03 0.13192E-03
0.40000 0.29000 0.12625 -0.49232E-03 0.12105E-03 0.13192E-03

```

## TRACK files

The Post-Processor **TRACK** command and the Charged Particle solver store the trajectory coordinates in a binary unformatted file. Coordinates are in cm and velocities in cm/s. There is no indication of the number of tracks in the file; the data must be read with end and error control to detect when all data has been read. The contents of the file for each track is as follows:

**Record 1:** 20 **INTEGER** values, **ITRAK**. Most **ITRAK** values can be ignored; the following entries are important.

**ITRAK (1)** is the number of steps, **NSTEP**.

**ITRAK (2)** is a format flag, affecting records 4, 5, 7, 8, and 9.

**ITRAK (3)** is the emitter number.

**ITRAK (7)** is 0 (local coordinates) or 1 (global coordinates)

**Record 2:** 20 **DOUBLE PRECISION** values, **RTRAK**.

**RTRAK (1)** is the current associated with the track.

**RTRAK (2)** is the mass.

**RTRAK (3)** is the charge.

**RTRAK (4)** is the step-length

**RTRAK (5-20)** can be ignored.

**Record 3:** **NSTEP DOUBLE PRECISION** values, x coordinates of the track points.

**Record 4:** **NSTEP DOUBLE PRECISION** values, y coordinates of the track points or, if **ITRAK (2)** equals 2, z coordinates of the track points.

**Record 5:** **NSTEP DOUBLE PRECISION** values, z coordinates of the track points or, if **ITRAK (2)** equals 2, y coordinates of the track points.

**Record 6:** **NSTEP DOUBLE PRECISION** values, x-component of the velocity.

**Record 7:** **NSTEP DOUBLE PRECISION** values, y-component of the velocity or, if **ITRAK (2)** equals 2, z-component of the velocity.

**Record 8:** **NSTEP DOUBLE PRECISION** values, z-component of the velocity or, if **ITRAK (2)** equals 2, y-component of the velocity.

**Record 9:** **NSTEP DOUBLE PRECISION** values, the current in the track. This record only exists if **ITRAK (2)** equals 3. For other values or **ITRAK (2)** use **RTRAK (1)**.

As explained above, track files are in a binary format. They can be converted to ASCII files with a small utility program called **READTRAC**. It reads a binary file and writes it in a portable text format, or reads a text file and produces the binary track file. The executable can be found in the installation directory /bin folder. **READTRAC** can be run with no arguments and the program will prompt for the required information. Filenames should exclude extensions and should not be "quoted" if there are spaces. Alternatively it can be run with the three command line arguments:

Windows:

**READTRAC .EXE** **inputfile [A|B]** **outputfile**

Linux:

**readtrack** **inputfile [A|B]** **outputfile**

where either **A** or **B** should be specified depending upon whether the **inputfile** is ASCII or Binary. Both **inputfile** and **outputfile** should be filenames with extensions and when used on the command line they must be "quoted" if they contain spaces.

Example:

```
READTRAC.EXE lens.track B lenstrack.txt
```

## The **ACTIVATE** Command

---

### Summary

Make an Opera-3d database available to the program.

### Toolbutton



### Command line parameters

| Command         | <b>ACTIVATE</b> |                                                     |
|-----------------|-----------------|-----------------------------------------------------|
| Parameter       | Default         | Function                                            |
| <b>FILE</b>     | <i>none</i>     | Name of database file.                              |
| <b>SYMMETRY</b> | 1               | Rotational symmetry around local Z axis.            |
| <b>RXY</b>      | <b>NO</b>       | Reflection in local XY plane.                       |
|                 |                 | <b>NO</b> No reflection.                            |
|                 |                 | <b>YES</b> Reflection with zero Z field.            |
|                 |                 | <b>INVERSE</b> Reflection with zero X and Y fields. |
| <b>RYZ</b>      | <b>NO</b>       | Reflection in local YZ plane.                       |
|                 |                 | <b>NO</b> No reflection.                            |
|                 |                 | <b>YES</b> Reflection with zero X field.            |
|                 |                 | <b>INVERSE</b> Reflection with zero Y and Z fields. |
| <b>RZX</b>      | <b>NO</b>       | Reflection in local ZX plane.                       |
|                 |                 | <b>NO</b> No reflection.                            |
|                 |                 | <b>YES</b> Reflection with zero Y field.            |
|                 |                 | <b>INVERSE</b> Reflection with zero Z and X fields. |
| <b>CASE</b>     | 1               | Simulation number or * for the last simulation.     |

| Command              | ACTIVATE        |                                                                                         |
|----------------------|-----------------|-----------------------------------------------------------------------------------------|
| <b>MODELSYMMETRY</b> | <b>DATABASE</b> | Symmetry of the model:                                                                  |
|                      |                 | <b>DATABASE</b> Use the symmetry from the background region in the database             |
|                      |                 | <b>NONE</b> No symmetry                                                                 |
|                      |                 | <b>USER</b> Use the parameters <b>SYMMETRY</b> , <b>RXY</b> , <b>RYZ</b> and <b>RZX</b> |
| <b>XORIGIN</b>       | 0               | X-coordinate of the coordinate system origin.                                           |
| <b>YORIGIN</b>       | 0               | Y-coordinate of the coordinate system origin.                                           |
| <b>ZORIGIN</b>       | 0               | Z-coordinate of the coordinate system origin.                                           |
| <b>THETA</b>         | 0               | Euler angle defining the coordinate system.                                             |
| <b>PHI</b>           | 0               | Euler angle defining the coordinate system.                                             |
| <b>PSI</b>           | 0               | Euler angle defining the coordinate system.                                             |

## Notes

The **ACTIVATE** command opens and checks a database file created by one of the analysis programs. The parameter **FILE** sets the file name. If no file name extension is given an extension of *op3* is assumed.

## Menu mode

In menu mode, the **ACTIVATE** command is automatically followed by [The LOAD Command \[page 770\]](#) and no parameters other than **FILE** can be given. The other parameters of the **ACTIVATE** command can be changed using other commands:

- If the database contains more than one simulation, the program automatically shows the [Load Another Simulation](#) dialog ([The SIMULATION Command \[page 806\]](#)), so that the required simulation can be loaded.
- Databases created by the Modeller contain the model symmetry information. The **ACTIVATE** command uses this information. For other databases, or to override the model symmetry, [The SYMMETRY Command \[page 811\]](#) can be used. It also allows local coordinate system for the database to be set.

## Command line mode

In command line mode the **ACTIVATE** command can be used to choose the simulation number (**CASE**), as well as to set the model symmetry and local coordinate system. Symmetry codes can be used to create the complete model from the section which was analysed. If **MODELSYMMETRY=DATABASE**, the program will use the information given in the Modeller by [The BACKGROUND Command \[page 135\]](#).

## Local coordinate system

**ACTIVATE** sets a local coordinate system by its origin (**XORIGIN**, **YORIGIN**, **ZORIGIN**) and Euler angles (**THETA**, **PHI**, **PSI**). This repositions the whole model in space as if it had been defined in that position. The local coordinate system of the loaded database can be adjusted using **The SYMMETRY Command [page 811]**.

## Symmetry

Symmetry codes can be used to create the complete model from the section which was analysed.

- **MODELSYMMETRY=DATABASE:** The information given in the Modeller by **The BACKGROUND Command [page 135]** is used to specify the symmetry. If this information does not exist in the database, **MODELSYMMETRY=USER** will be assumed instead. The symmetry information for the loaded database can be adjusted and stored in the database for subsequent activation using **The SYMMETRY Command [page 811]**.
- **MODELSYMMETRY=USER:** In this case the complete model can be formed by rotation in the local Z axis and reflection in the local coordinate planes.

Three types of user symmetry are available.

- **SYMMETRY=n** creates **n** copies by rotating through  $360/n$  degrees around the local Z axis. The sign of the parameter **SYMMETRY** determines the direction of the field in the copies of the model. If **SYMMETRY** is negative the field direction is reversed in alternate copies.
- **RUV=YES** should be used when the field normal to the local *uv* plane is zero. The program reflects the geometry in the *uv* plane of the local system and inverts the sign of the normal field in the reflected copy.
- **RUV=INVERSE** should be used when the tangential field in the local *uv* plane is zero. The program reflects the geometry in the *uv* planes of the local system and inverts the sign of the tangential field in the reflected copy.

The field reflections apply to the principal field of the simulation, i.e. **H** for magnetic field models (Electromagnetic, Magnetization and Magnetostatic solvers), **E** for electric field models (Charged Particle, Current Flow, Electrostatic and High Frequency solvers) and  $\nabla T$  for thermal and Quench models.

The maximum number of reflected and rotated copies of the model is 2048.

## Simulation number

For databases with more than one simulation, the **CASE** parameter should be set to choose which simulation should be activated. **CASE=\*** indicates the last simulation in the database. After activating a database, the system variable **CASES** is updated with the number of simulations in the database.

Up to 100 files can be active at a time, each with its own coordinate system. In this way devices which form a complicated system, but which can be analysed separately can be linked together for post-processing. The **SEARCH** parameter of **The SET Command [page 798]** controls the way in which multiple active database files are searched for field points.

## Status

The status of the file last successfully **ACTIVATE**d is recorded in the **ACTIVATE\_STATUS** system variable with the following meanings:

| <b>ACTIVATE_STATUS</b> | <b>Meaning</b>                                                 |
|------------------------|----------------------------------------------------------------|
| -1                     | Simulation pending                                             |
| 0                      | Simulation solved successfully                                 |
| 1                      | Simulation failed                                              |
| 3                      | Solution present but the nonlinear iterations did not converge |
| -999                   | Simulation is not valid (simulation disabled)                  |

## The **ANIMATION** Command

---

### Summary

Record or play back an animation file.

### Toolbutton



### Command line parameters

| Command           | <b>ANIMATION</b> |                                                                                            |                                   |
|-------------------|------------------|--------------------------------------------------------------------------------------------|-----------------------------------|
| Parameter         | Default          | Function                                                                                   |                                   |
| <b>ACTION</b>     | <b>NEW</b>       | Action on animation file:                                                                  |                                   |
|                   |                  | <b>NEW</b>                                                                                 | Create or overwrite an animation. |
|                   |                  | <b>ADD</b>                                                                                 | Add to or create an animation.    |
|                   |                  | <b>PLAY</b>                                                                                | Play back a stored animation.     |
| <b>FILE</b>       | <i>none</i>      | File name to be recorded or played back.                                                   |                                   |
| <b>DELAY</b>      | 0                | The delay in seconds between frames when playing back an animation.                        |                                   |
| <b>SCALE</b>      | 100              | The picture size of the animation (in percent of the original size) when playing back.     |                                   |
| <b>LOOP</b>       | <b>NO</b>        | <b>NO</b>                                                                                  | Stop at the end of the animation. |
|                   |                  | <b>YES</b>                                                                                 | Restart at the beginning.         |
| <b>TIMETOPLAY</b> | 0                | The time in seconds after which the animation viewer will close (0 keeps the viewer open). |                                   |

### Notes

The **ANIMATION** command can be used to record or play back an animation file. An animation file is a series of compressed images stored in the "Portable Network Graphics" format using a **png** extension.

The following **ACTIONs** are used to add the current screen display to an animation file or play back an existing file:

- **NEW**: creates a new file or overwrites an existing file with one image.

- **ADD**: creates a new file with one image or adds a new image to an existing file.
- **PLAY**: launches an animation viewer.

The play-back parameters can be used to adjust the initial behaviour of the animation viewer:

- the **DELAY** between frames,
- the size of the play-back window relative to the original window size (**SCALE** as a percentage),
- whether the animation will play once (**LOOP=NO**) or restart at the beginning when it gets to the end (**LOOP=YES**), and
- how long the animation will play for (**TIMETOPLAY** in seconds).

The animation viewer has toolbuttons to change the way the animation is displayed (it can be stopped, single stepped or played in a continuous loop). The currently displayed frame in the animation viewer can be copied to the system clipboard.

## The ARC Command

---

### Summary

Calculate fields along a circular arc defined by centre and end points.

### Toolbutton



### Command line parameters

| Command   | ARC     |                                                                     |
|-----------|---------|---------------------------------------------------------------------|
| Parameter | Default | Function                                                            |
| X1        | none    | X-coordinate of the first point on the arc.                         |
| Y1        | none    | Y-coordinate of the first point on the arc.                         |
| Z1        | none    | Z-coordinate of the first point on the arc.                         |
| X2        | none    | X-coordinate of the last point on the arc.                          |
| Y2        | none    | Y-coordinate of the last point on the arc.                          |
| Z2        | none    | Z-coordinate of the last point on the arc.                          |
| XC        | 0       | X-coordinate of the centre of curvature.                            |
| YC        | 0       | Y-coordinate of the centre of curvature.                            |
| ZC        | 0       | Z-coordinate of the centre of curvature.                            |
| NP        | 100     | Number of steps between the first and last points, i.e NP+1 points. |
| BUFFER    | Arc     | Name of buffer to store field values.                               |

### Notes

The **ARC** command evaluates field quantities along a circular arc for use by

- [The FIT Command \[page 747\]](#)
- [The PLOT Command \[page 783\]](#).

For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)).

The arc is specified by its end points ( $X_1, Y_1, Z_1$  and  $X_2, Y_2, Z_2$ ) and centre of curvature ( $X_C, Y_C, Z_C$ ) (see [Figure 7.2, on page 769](#)). The end points and centre of curvature must not be collinear, otherwise the plane of the arc cannot be determined. The radius of the arc is taken from the distance between the start point and the centre. The last point on the arc is calculated from the intersection of the circle and the radial line through the specified end point ( $X_2, Y_2, Z_2$ ). The minor arc is always chosen. Major arcs can be specified with [The CIRCLE Command \[page 718\]](#).

The positions of the end and centre points are affected by any local coordinate system defined with [The SET Command \[page 798\]](#). The vector field quantities are evaluated with respect to the Global Coordinate System for the active file.

The field quantities are evaluated at  $NP+1$  points along the arc and are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## Graphs of field values

The **Fields on an Arc** dialog runs 3 commands:

- [The AXESVIEW Command \[page 689\]](#) (optional) to create a new graph.
- The **ARC** command.
- [The DATALINE Command \[page 735\]](#) to add the line of calculated values to the graph.

## The ARITHMETIC Command

---

### Summary

Combine data in two table files and create a third.

### Toolbutton



### Command line parameters

|                      |                  |                                            |
|----------------------|------------------|--------------------------------------------|
| Command              | ARITHMETIC       |                                            |
| Parameter            | Default          | Function                                   |
| FILE1                | none             | Name of first input data file.             |
| FILE2                | none             | Name of second input data file.            |
| FILE3                | none             | Name of output data file.                  |
| VALUE                | VAL1+VAL2        | Expression for values in output data file. |
| NAME1, ..., NAME12   | names from FILE1 | Names for the columns in output data file. |
| UNIT1A, ..., UNIT12A | units from FILE1 | Units for the columns in output data file. |

### Notes

The ARITHMETIC command combines the values in two *table* files and creates a new file of the same type. The format of the files is described in section TABLE Files [page 674].

The two input files, FILE1 and FILE2, must contain the same number of records in the same configuration. For example, if they both represent points on a 2-dimensional patch, they must have the same numbers of points in the two directions given on the first record. They must also contain the same number of columns and be written with the same format.

By default the output file, FILE3, is created using the number of field points, the column names and unit expressions of FILE1. The column names and unit expressions can be replaced using parameters NAME1 to NAME12 and UNIT1A to UNIT12A. For each column in each line of numerical data, the value from FILE1 is assigned to variable VAL1, then the value from FILE2 to VAL2 and the result of evaluating the expression given by VALUE is written to FILE3.

The values in columns with names X, Y, Z, POTCODE, MATCODE, ELEMENT or NODE are not combined. The values from FILE1 are used.

## Format 1

The values in the input and output files are in internal units and no unit conversion takes place.

## Format 2

The values in the input files are converted into internal units using the unit expressions in the files. After calculating the output values, they are converted for output into the current set of units in the program using the unit expressions from **FILE1** or the **UNITnA** parameters. N.B. this might not correspond to the units in either of the input files.

## The **AXESVIEW** Command

---

### Summary

Create new graph axes, add lines, vector sets and annotations and print to file.

### Usage

The **AXESVIEW** command creates a new graph. Once a graph exists, its properties can be modified and an annotation can be added using the **Graphs** tab property sheets. Any number of graphs can be created.

When the program starts, a graph called **Default** already exists. Other graphs can be created using the dialogs which plot values of fields along lines or create graphs from data files. New graphs can also be created using the context menu in the **Graphs** tab.

Three types of graph are available:

- **CARTESIAN**: horizontal (X) and vertical (Y) axes can be used to plot lines created by [The DATALINE Command \[page 735\]](#).
- **POLAR**: radial ( $r$ ) and azimuthal ( $\theta$ ) axes can be used to plot lines created by [The DATALINE Command \[page 735\]](#).
- **VECTOR** diagrams are special cartesian axes for displaying sets of vectors created by [The DATAVECTORSET Command \[page 740\]](#). The vectors are created by [The DATAVECTOR Command \[page 738\]](#).

The **AXESVIEW** command can also add an existing line or vector set to a graph or can modify the display attributes of a line, vector set or vector on a graph, e.g. the colour, line style, symbols, etc. The same line or vector set can be displayed with different attributes on multiple graphs.

One of the right-click options on a graph is **Export Graph Commands**. This gives access to **AXESVIEW** commands that can be used to recreate the current graph including its lines and annotation. The commands can be copied into the Command File Editor so that recreation of the graph can be scripted.

## Common command line parameters

| Command   | AXESVIEW |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter | Default  | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| OPTION    |          | Action of the command:<br>ANNOTATION Add annotation to a graph.<br>CLEARALL Delete all graph data.<br>CREATE Create a new graph.<br>DELETE Delete the named graph.<br>EXPORT Export graph attributes.<br>LINE Add a line to a graph or change attributes of a line already on a graph.<br>MODIFY Load graph attributes and/or modify attributes of a graph.<br>PRINT Print the named graph to a picture file.<br>SHOW Show the named graph.<br>VECTOR Change attributes of a vector.<br>VECTORSET Add a vector set to a vector diagram or change attributes of a vector set. |
| GRAPH     |          | The name of the graph (not needed for OPTION=CLEARALL).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

Every time the **AXESVIEW** command is used, the **OPTION** must be supplied; the **GRAPH** name is also required for all options except **CLEARALL**. The **OPTIONS** are as follows:

- **ANNOTATION**: add an annotation to a graph. Annotations are text strings which can be positioned anywhere on the graph window. See "Additional parameters for adding an annotation" on page 700.
- **CLEARALL**: delete all graph data (buffers, lines, vectors, vector sets and graphs). The **Default** graph is automatically recreated.
- **CREATE**: create a new set of cartesian or polar axes or a vector diagram. Styles can be set for the graph, including axes, grid and a legend. See "Additional parameters for creating or modifying axes" on the facing page.
- **DELETE**: delete a graph.

- **EXPORT**: save the attributes of the graph (settings for title, grid, labelling, etc.) in a style sheet file (\*.ini). Attribute files can be applied to a graph using **OPTION=MODIFY**. See "Additional parameters for saving graph attributes" on page 700.
- **LINE**: add an existing line to a graph or change the attributes (style, colour, etc.) of a line which is already displayed on a graph. See "Additional parameters for a line" on page 697.
- **MODIFY**: change the attributes of a graph by optionally loading a style sheet file (\*.ini) and then by applying any other parameters on the command line.
- **PRINT**: create a picture file (\*.jpg or \*.png) containing a graph. See "Additional parameters for printing to file" on page 701. The currently displayed graph can also be printed or copied to a file or the clipboard using the **PRINT** The PRINT Command and **PICTURE** commandsThe PICTURE Command.
- **SHOW**: if more than one graph exists, move the named graph to the front so that it can be seen.
- **VECTOR**: change the attributes (style, colour, etc.) of a vector in a vector set displayed on a graph. See "Additional parameters for a vector" on page 698.
- **VECTORSET**: add a vector set to a vector diagram or change the attributes (scaling factor, rotation) of a vector set already in a vector diagram. See "Additional parameters for a vector set" on page 699.

## Additional parameters for creating or modifying axes

| Command                                         | AXESVIEW             |                                                                                                                                                                                                                                                                                        |                  |                      |              |                  |               |                 |
|-------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|----------------------|--------------|------------------|---------------|-----------------|
| Parameter                                       | Default              | Function                                                                                                                                                                                                                                                                               |                  |                      |              |                  |               |                 |
| <b>MAKEUNIQUE</b>                               | <b>NO</b>            | Make the <b>GRAPH</b> name unique: <b>YES</b> or <b>NO</b> .<br>(Only for <b>CREATE</b> .)                                                                                                                                                                                             |                  |                      |              |                  |               |                 |
| <b>TYPE</b>                                     |                      | Graph type (only for <b>CREATE</b> ):<br><table border="1" data-bbox="822 1284 1346 1410"> <tr> <td><b>CARTESIAN</b></td><td>Cartesian (XY) axes.</td></tr> <tr> <td><b>POLAR</b></td><td>Polar (rθ) axes.</td></tr> <tr> <td><b>VECTOR</b></td><td>Vector diagram.</td></tr> </table> | <b>CARTESIAN</b> | Cartesian (XY) axes. | <b>POLAR</b> | Polar (rθ) axes. | <b>VECTOR</b> | Vector diagram. |
| <b>CARTESIAN</b>                                | Cartesian (XY) axes. |                                                                                                                                                                                                                                                                                        |                  |                      |              |                  |               |                 |
| <b>POLAR</b>                                    | Polar (rθ) axes.     |                                                                                                                                                                                                                                                                                        |                  |                      |              |                  |               |                 |
| <b>VECTOR</b>                                   | Vector diagram.      |                                                                                                                                                                                                                                                                                        |                  |                      |              |                  |               |                 |
| <b>STYLESSHEET</b>                              |                      | Name of stylesheet file to be read and applied before other parameters. (Only for <b>MODIFY</b> .)                                                                                                                                                                                     |                  |                      |              |                  |               |                 |
| <b>TITLE</b>                                    | Title                | Title of the graph.                                                                                                                                                                                                                                                                    |                  |                      |              |                  |               |                 |
| <b>BGRED</b><br><b>BGGREEN</b><br><b>BGBLUE</b> | 255, 255, 255        | The background colour (see <b>Colours</b> [page 701]).                                                                                                                                                                                                                                 |                  |                      |              |                  |               |                 |
| <b>BGIMAGE</b>                                  |                      | The name of a file containing an image to be used as the background.                                                                                                                                                                                                                   |                  |                      |              |                  |               |                 |

|                                                                          |                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------------------------------------|--------------|--------------------------------------------|---------------|------------------------------------|------------|--------------------------|-----------------|--------------------------------------------------------------------------|
| Command                                                                  | <b>AXESVIEW</b>                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| Parameter                                                                | Default                                                                  | Function                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>BORDERRED</b><br><b>BORDERGREEN</b><br><b>BORDERBLUE</b>              | 255, 255, 255                                                            | The border colour (see <a href="#">Colours [page 701]</a> ).                                                                                                                                                                                                                                                                                                                                                                                         |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>TITLEDISPLAYED</b>                                                    | <b>YES</b>                                                               | Display title in the graph: <b>YES</b> or <b>NO</b> .                                                                                                                                                                                                                                                                                                                                                                                                |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>TITLEFONTFAMILY</b><br><b>TITLEFONTSIZE</b><br><b>TITLEFONTWEIGHT</b> | Tahoma, 12, 75                                                           | Title font family, point size and weight (see <a href="#">Fonts [page 701]</a> ).                                                                                                                                                                                                                                                                                                                                                                    |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>TITLERED</b><br><b>TITLEGREEN</b><br><b>TITLEBLUE</b>                 | 0, 0, 0                                                                  | The title colour (see <a href="#">Colours [page 701]</a> ).                                                                                                                                                                                                                                                                                                                                                                                          |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>TITLEALIGNMENT</b>                                                    | <b>CENTRE</b>                                                            | Title alignment mode:<br><br><table> <tr> <td><b>LEFT</b></td> <td>The title starts at the left of the window.</td> </tr> <tr> <td><b>RIGHT</b></td> <td>The title ends at the right of the window.</td> </tr> <tr> <td><b>CENTRE</b></td> <td>The title is centred horizontally.</td> </tr> </table>                                                                                                                                                | <b>LEFT</b> | The title starts at the left of the window. | <b>RIGHT</b> | The title ends at the right of the window. | <b>CENTRE</b> | The title is centred horizontally. |            |                          |                 |                                                                          |
| <b>LEFT</b>                                                              | The title starts at the left of the window.                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>RIGHT</b>                                                             | The title ends at the right of the window.                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>CENTRE</b>                                                            | The title is centred horizontally.                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>LEGENDDISPLAYED</b>                                                   | <b>YES</b>                                                               | Display the legend: <b>YES</b> or <b>NO</b> .                                                                                                                                                                                                                                                                                                                                                                                                        |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>LEGENDPOSITION</b>                                                    | <b>RIGHT</b>                                                             | Legend position mode:<br><br><table> <tr> <td><b>LEFT</b></td> <td>At the left of the window.</td> </tr> <tr> <td><b>RIGHT</b></td> <td>At the right of the window.</td> </tr> <tr> <td><b>BOTTOM</b></td> <td>At the bottom of the window</td> </tr> <tr> <td><b>TOP</b></td> <td>At the top of the window</td> </tr> <tr> <td><b>FLOATING</b></td> <td>At position given by <b>LEGENDHPOSITION</b> and <b>LEGENDVPOSITION</b>.</td> </tr> </table> | <b>LEFT</b> | At the left of the window.                  | <b>RIGHT</b> | At the right of the window.                | <b>BOTTOM</b> | At the bottom of the window        | <b>TOP</b> | At the top of the window | <b>FLOATING</b> | At position given by <b>LEGENDHPOSITION</b> and <b>LEGENDVPOSITION</b> . |
| <b>LEFT</b>                                                              | At the left of the window.                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>RIGHT</b>                                                             | At the right of the window.                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>BOTTOM</b>                                                            | At the bottom of the window                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>TOP</b>                                                               | At the top of the window                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>FLOATING</b>                                                          | At position given by <b>LEGENDHPOSITION</b> and <b>LEGENDVPOSITION</b> . |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>LEGENDHPOSITION</b>                                                   | 0.88                                                                     | Relative horizontal position for legend (0 is left, 1 is right).                                                                                                                                                                                                                                                                                                                                                                                     |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |
| <b>LEGENDVPOSITION</b>                                                   | 0.08                                                                     | Relative vertical position for legend (0 is top, 1 is bottom).                                                                                                                                                                                                                                                                                                                                                                                       |             |                                             |              |                                            |               |                                    |            |                          |                 |                                                                          |

|                                                                                                                                                                                                                    |                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command                                                                                                                                                                                                            | AXESVIEW                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Parameter                                                                                                                                                                                                          | Default                             | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| GRIDDISPLAY                                                                                                                                                                                                        | YES                                 | Should grid be displayed: YES or NO:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| GRIDMODE                                                                                                                                                                                                           | VALLHALL                            | Which grid lines should be displayed:<br>VALLHALL      Vertical: all<br>Horizontal: all<br>VALLHMAJOR    Vertical: all<br>Horizontal: major<br>VALLHNONE     Vertical: all<br>Horizontal: none<br>VMAJORHALL    Vertical: major<br>Horizontal: all<br>VMAJORHMAJOR   Vertical: major<br>Horizontal: major<br>VMAJORHNONE   Vertical: major<br>Horizontal: none<br>VNONEHALL     Vertical: none<br>Horizontal: all<br>VNONEHMAJOR   Vertical: none<br>Horizontal: major<br>VNONEHNONE    Vertical: none<br>Horizontal: none |
| <b>The colours, widths and styles of the grid lines can be set using the following parameters, where <i>grid</i> can be one of: GRIDMAJOR, GRIDMINOR, AZIMUTHMAJOR, AZIMUTHMINOR, RADIALMAJOR and RADIALMINOR.</b> |                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| gridRED                                                                                                                                                                                                            | Black (0, 0, 0) for major grids     | The grid colour (see Colours [page 701]).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| gridGREEN                                                                                                                                                                                                          | Gray (160,160,164) for minor grids. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| gridBLUE                                                                                                                                                                                                           |                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| gridWIDTH                                                                                                                                                                                                          | 1.0                                 | Grid line width.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Command                                                                                                         | AXESVIEW  |                                                          |
|-----------------------------------------------------------------------------------------------------------------|-----------|----------------------------------------------------------|
| Parameter                                                                                                       | Default   | Function                                                 |
| <i>grid</i> STYLE                                                                                               | DOT       | Grid line style:                                         |
|                                                                                                                 |           | SOLID Solid lines.                                       |
|                                                                                                                 |           | DASH Dashed lines.                                       |
|                                                                                                                 |           | DOT Dotted lines.                                        |
|                                                                                                                 |           | DASHDOT Alternating dashes and dots.                     |
|                                                                                                                 |           | DASHDOTDOT Alternating dashes and pairs of dots.         |
| <b>Cartesian axes can be controlled using the following parameters, where <i>axis</i> can be one of X or Y.</b> |           |                                                          |
| <i>axis</i> DISPLAYED                                                                                           | YES       | Display axis: YES or NO.                                 |
| <i>axis</i> SCALETYPE                                                                                           | LINEAR    | Axis scale type:                                         |
|                                                                                                                 |           | LINEAR Linear scale.                                     |
|                                                                                                                 |           | LOG10 Logarithmic scale.                                 |
| <i>axis</i> TICKDISPLAYED                                                                                       | YES       | Display tick marks: YES or NO.                           |
| <i>axis</i> MAJORTICKMARKS                                                                                      | 5         | Number of major tick marks along the axis.               |
| <i>axis</i> MINORTICKMARKS                                                                                      | 5         | Number of minor tick marks between adjacent major ticks. |
| <i>axis</i> MAJORTICKLENGTH                                                                                     | 0         | The length of a major tick mark.                         |
| <i>axis</i> MINORTICKLENGTH                                                                                     | 0         | The length of a minor tick mark.                         |
| <i>axis</i> LABELADJUST                                                                                         | AUTOMATIC | Axis label mode:                                         |
|                                                                                                                 |           | AUTOMATIC Label automatically selected.                  |
|                                                                                                                 |           | MANUAL User defined label.                               |
|                                                                                                                 |           | NONE No label displayed.                                 |
| <i>axis</i> LABELTEXT                                                                                           |           | Axis label text if <i>axis</i> LABELADJUST is MANUAL.    |
| <i>axis</i> LABELRED<br><i>axis</i> LABELGREEN<br><i>axis</i> LABELBLUE                                         | 0, 0, 0   | Axis label colour (see Colours [page 701]).              |

|                                                                                      |                  |                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command                                                                              | <b>AXESVIEW</b>  |                                                                                                                                                                                                                                                                                                                                                                                 |
| Parameter                                                                            | Default          | Function                                                                                                                                                                                                                                                                                                                                                                        |
| <i>axisLABELFONTFAMILY</i><br><i>axisLABELFONTSIZE</i><br><i>axisLABELFONTWEIGHT</i> | Tahoma, 10, 50   | Axis label font family, size and weight (see <a href="#">Fonts [page 701]</a> ).                                                                                                                                                                                                                                                                                                |
| <i>axisLABELROTATION</i>                                                             | 0.0              | Axis values rotation angle in degrees.                                                                                                                                                                                                                                                                                                                                          |
| <i>axisLIMIT</i>                                                                     | <b>AUTOMATIC</b> | Axis scale limit type:<br><br>AUTOMATIC      Lower and upper values calculated from data.<br>LOWER            Upper value: calculated<br>Lower value: given by <i>axisLOWER</i> .<br>UPPER            Upper value: given by <i>axisUPPER</i><br>Lower value: calculated.<br>FIXED            Upper value: given by <i>axisUPPER</i><br>Lower value: given by <i>axisLOWER</i> . |
| <i>axisLOWER</i>                                                                     | 0.0              | Axis scale lower limit.                                                                                                                                                                                                                                                                                                                                                         |
| <i>axisUPPER</i>                                                                     | 10.0             | Axis scale upper limit.                                                                                                                                                                                                                                                                                                                                                         |
| <b>The following parameters are for polar axes.</b>                                  |                  |                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>AZIMUTHALDISPLAYED</i>                                                            | <b>YES</b>       | Display azimuthal axis: YES or NO.                                                                                                                                                                                                                                                                                                                                              |
| <i>AZIMUTHALRED</i><br><i>AZIMUTHALGREEN</i><br><i>AZIMUTHALBLUE</i>                 | 0, 0, 0          | Azimuthal axis colour (see <a href="#">Colours [page 701]</a> ).                                                                                                                                                                                                                                                                                                                |
| <i>AZIMUTHALSTEP</i>                                                                 | 20               | Angle between azimuthal axis values.                                                                                                                                                                                                                                                                                                                                            |
| <i>RADIALDISPLAYED</i>                                                               | <b>YES</b>       | Display radial axis: YES or NO.                                                                                                                                                                                                                                                                                                                                                 |
| <i>RADIALSCALETYPE</i>                                                               | <b>LINEAR</b>    | Radial axis scale type:<br><br>LINEAR           Linear scale.<br>LOG10           Logarithmic scale.                                                                                                                                                                                                                                                                             |

| Command                                                      | AXESVIEW  |                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter                                                    | Default   | Function                                                                                                                                                                                                                                                                                                                                                                     |
| RADIALRED<br>RADIALGREEN<br>RADIALBLUE                       | 0, 0, 0   | Radial axis colour (see Colours [page 701]).                                                                                                                                                                                                                                                                                                                                 |
| RADIALLIMIT                                                  | AUTOMATIC | Radial axis scale limit type:<br><br>AUTOMATIC      Lower and upper values calculated from data.<br><br>LOWER            Upper value: calculated<br>Lower value: given by RADIALLOWER.<br><br>UPPER            Upper value: given by RADIALUPPER<br>Lower value: calculated.<br><br>FIXED            Upper value: given by RADIALUPPER<br>Lower value: given by RADIALLOWER. |
| RADIALLOWER                                                  | 0.0       | Radial axis scale lower limit.                                                                                                                                                                                                                                                                                                                                               |
| RADIALUPPER                                                  | 10.0      | Radial axis scale upper limit.                                                                                                                                                                                                                                                                                                                                               |
| RADIALLEFT                                                   | YES       | Show radial axis at angle 0: YES or NO.                                                                                                                                                                                                                                                                                                                                      |
| RADIALRIGHT                                                  | YES       | Show radial axis at angle 180: YES or NO.                                                                                                                                                                                                                                                                                                                                    |
| RADIALTOP                                                    | YES       | Show radial axis at angle 90: YES or NO.                                                                                                                                                                                                                                                                                                                                     |
| RADIALBOTTOM                                                 | YES       | Show radial axis at angle -90: YES or NO.                                                                                                                                                                                                                                                                                                                                    |
| <b>The following parameters are for vector diagram axes.</b> |           |                                                                                                                                                                                                                                                                                                                                                                              |
| VECTORAXISDISPLAYED                                          | YES       | Display vector diagram axes: YES or NO.                                                                                                                                                                                                                                                                                                                                      |
| VECTORTICKDISPLAYED                                          | YES       | Display tick marks: YES or NO.                                                                                                                                                                                                                                                                                                                                               |
| VECTORMAJORTICKMARKS                                         | 5         | Number of major tick marks along the axis.                                                                                                                                                                                                                                                                                                                                   |
| VECTORMINORTICKMARKS                                         | 5         | Number of minor tick marks between adjacent major ticks.                                                                                                                                                                                                                                                                                                                     |
| VECTORMAJORTICKLENGTH                                        | 0         | The length of a major tick mark.                                                                                                                                                                                                                                                                                                                                             |
| VECTORMINORTICKLENGTH                                        | 0         | The length of a minor tick mark.                                                                                                                                                                                                                                                                                                                                             |

|                     |           |                                       |
|---------------------|-----------|---------------------------------------|
| Command             | AXESVIEW  |                                       |
| Parameter           | Default   | Function                              |
| VECTORTICKDISPLAYED | YES       | Display tick marks: YES or NO.        |
| VECTORLIMIT         | AUTOMATIC | Vector diagram axes scale limit type: |
|                     |           | AUTOMATIC Range calculated from data. |
|                     |           | FIXED Range given by VECTORRANGE.     |
| VECTORRANGE         | 10.0      | Range of vector diagram axes values.  |

### Additional parameters for a line

|                                  |          |                                                       |
|----------------------------------|----------|-------------------------------------------------------|
| Command                          | AXESVIEW |                                                       |
| Parameter                        | Default  | Function                                              |
| LINE                             |          | The name of the line to be added.                     |
| LINEDISPLAYED                    | YES      | Display line on graph: YES or NO.                     |
| LINETYPE                         | SOLID    | Line style:                                           |
|                                  |          | DASH Dashed line.                                     |
|                                  |          | DASHDOT Alternating dashes and dots.                  |
|                                  |          | DASHDOTDOT Repeated sequences of a dash and two dots. |
|                                  |          | DOT Dotted line.                                      |
|                                  |          | NONE No line.                                         |
|                                  |          | SOLID Solid line.                                     |
| LINEWIDTH                        | 2        | Line width.                                           |
| LINERED<br>LINEGREEN<br>LINEBLUE | 0, 0, 0  | Colour of the line (see Colours [page 701]).          |

| Command                                                                     | AXESVIEW          |                                                                                    |
|-----------------------------------------------------------------------------|-------------------|------------------------------------------------------------------------------------|
| Parameter                                                                   | Default           | Function                                                                           |
| <b>SYMBOL</b>                                                               | <b>NOSYMBOL</b>   | Symbol displayed at data points:                                                   |
|                                                                             |                   | <b>CROSS</b> Plus sign ("+").                                                      |
|                                                                             |                   | <b>DIAMOND</b> Diamond.                                                            |
|                                                                             |                   | <b>DTRIANGLE</b> Triangle pointing down.                                           |
|                                                                             |                   | <b>ELLIPSE</b> Ellipse or circle.                                                  |
|                                                                             |                   | <b>HEXAGON</b> Hexagon.                                                            |
|                                                                             |                   | <b>HLINE</b> Horizontal line.                                                      |
|                                                                             |                   | <b>LTRIANGLE</b> Triangle pointing left.                                           |
|                                                                             |                   | <b>NOSYMBOL</b> No symbol.                                                         |
|                                                                             |                   | <b>RECTANGLE</b> Rectangle or square.                                              |
|                                                                             |                   | <b>RTRIANGLE</b> Triangle pointing right.                                          |
|                                                                             |                   | <b>STAR1</b> 8-pointed star ( <b>CROSS</b> and <b>XROSS</b> ).                     |
|                                                                             |                   | <b>STAR2</b> 6-pointed star, solid colour.                                         |
|                                                                             |                   | <b>TRIANGLE</b> <b>UTRIANGLE</b> Triangle pointing up.                             |
|                                                                             |                   | <b>VLINE</b> Vertical line.                                                        |
|                                                                             |                   | <b>XROSS</b> Diagonal cross.                                                       |
| <b>SYMBOLSIZE</b>                                                           | 6                 | Size of symbol.                                                                    |
| <b>LEGENDTEXT</b>                                                           | <i>line name</i>  | The legend text for the line.                                                      |
| <b>LEGENDFONTFAMILY</b><br><b>LEGENDFONTSIZE</b><br><b>LEGENDFONTWEIGHT</b> | Tahoma,<br>10, 50 | Legend font family, point size and weight (see <a href="#">Fonts [page 701]</a> ). |

### Additional parameters for a vector

| Command          | AXESVIEW |                                                              |
|------------------|----------|--------------------------------------------------------------|
| Parameter        | Default  | Function                                                     |
| <b>VECTOR</b>    |          | The name of the vector to be added to a vector set.          |
| <b>VECTORSET</b> |          | The name of the vector set that the vector will be added to. |

| Command                          | AXESVIEW |                                                       |
|----------------------------------|----------|-------------------------------------------------------|
| Parameter                        | Default  | Function                                              |
| VECTORDISPLAYED                  | YES      | Display vector on graph: YES or NO.                   |
| LINETYPE                         | SOLID    | Line style:                                           |
|                                  |          | DASH Dashed line.                                     |
|                                  |          | DASHDOT Alternating dashes and dots.                  |
|                                  |          | DASHDOTDOT Repeated sequences of a dash and two dots. |
|                                  |          | DOT Dotted line.                                      |
|                                  |          | NONE No line.                                         |
|                                  |          | SOLID Solid line.                                     |
| LINEWIDTH                        | 2        | Line width.                                           |
| LINERED<br>LINEGREEN<br>LINEBLUE | 0, 0, 0  | Colour of the vector (see Colours [page 701]).        |
| SYMBOLDISPLAYED                  | YES      | Display arrowhead on vector: YES or NO.               |
| SYMBOLSIZE                       | 6        | Size of arrowhead.                                    |

### Additional parameters for a vector set

| Command            | AXESVIEW |                                              |
|--------------------|----------|----------------------------------------------|
| Parameter          | Default  | Function                                     |
| VECTORSET          |          | The name of the vector set to be added.      |
| VECTORSETDISPLAYED | YES      | Display vector set on graph: YES or NO.      |
| VECTORSETROTATE    | 0        | Rotation to be applied to the vector set.    |
| VECTORSETSCALE     | 1        | Scaling factor to applied to the vector set. |

### Additional parameters for adding an annotation

| Command                                                        | AXESVIEW          |                                                                      |
|----------------------------------------------------------------|-------------------|----------------------------------------------------------------------|
| Parameter                                                      | Default           | Function                                                             |
| ANNODISPLAYED                                                  | YES               | Display annotation: YES or NO.                                       |
| ANNOTEXT                                                       |                   | The annotation text.                                                 |
| ANNOFONTFAMILY<br>ANNOFONTSIZE<br>ANNOFONTWEIGHT               | Tahoma,<br>10, 50 | Annotation font, size and weight (see Fonts [page 701]).             |
| ANNORED<br>ANNOGREEN<br>ANNOBLUE                               | 0, 0, 0           | Annotation text colour (see Colours [page 701]).                     |
| ANNOBACKGROUND                                                 | NO                | Show background colour behind annotation: YES or NO.                 |
| ANNOBACKGROUNDRED<br>ANNOBACKGROUNDGREEN<br>ANNOBACKGROUNDBLUE | 255, 255,<br>255  | Annotation background colour (see Colours [page 701]).               |
| ANNOALIGNMENT                                                  | LEFT              | Annotation text alignment mode.                                      |
|                                                                |                   | LEFT      The text starts at the left of the window.                 |
|                                                                |                   | RIGHT     The text ends at the right of the window.                  |
|                                                                |                   | CENTRE    The text is centred horizontally.                          |
| ANNOPOSITIONX                                                  | 0.03              | Relative horizontal position for annotation (0 is left, 1 is right). |
| ANNOPOSITIONY                                                  | 0.02              | Relative vertical position for annotation (0 is top, 1 is bottom).   |

### Additional parameters for saving graph attributes

| Command    | AXESVIEW |                                              |
|------------|----------|----------------------------------------------|
| Parameter  | Default  | Function                                     |
| STYLESHEET |          | Name of style sheet for attributes of graph. |

## Additional parameters for printing to file

|           |          |                                                                          |
|-----------|----------|--------------------------------------------------------------------------|
| Command   | AXESVIEW |                                                                          |
| Parameter | Default  | Function                                                                 |
| FILENAME  |          | Name of a <b>jpg</b> or <b>png</b> file when <b>PRINT</b> ing the graph. |

## Colours

Colours of objects in a graph are defined by values of *objectRED*, *objectGREEN* and *objectBLUE* in the range 0 to 255.

## Fonts

Fonts used to display text (title, annotation, and labels) are defined by the *objectFONTFAMILY*, *objectFONTSIZE* and *objectFONTWEIGHT*.

- The font families depend on the fonts available on the local system.
- Font sizes are defined in points.
- Font weights should be in the range 0 to 99.
  - 50 indicates normal and
  - 75 indicates bold.

## Interacting with graphs

It is possible to interact with a graph using the keyboard or mouse buttons.

- Selecting a graph object: left mouse button; this makes the parameters for the object appear in the **Graph Objects** property sheet so that they can be adjusted as required.
- Symmetric zoom in or out: middle button wheel.
- Rubber-box zoom in: click and drag with **<shift>** and left button.
- Zoom out to full size: **<Esc>** or **<shift>** and right button.
- Data values on a line: **<ctrl>** and left button activates cross-hairs cursor which tracks the graph coordinates and the values at the closest data point on each of the lines on a cartesian graph.

## The **BHDATA** Command

---

### Summary

Plot graphs of BH characteristics used in analysis.

### Toolbutton



### Command line parameters

| Command          | <b>BHDATA</b> |                                                                |
|------------------|---------------|----------------------------------------------------------------|
| Parameter        | Default       | Function                                                       |
| <b>MATERIAL</b>  | 1             | Material name or number.                                       |
| <b>DIRECTION</b> | X             | Direction for anisotropic materials (X, Y or Z).               |
| <b>OPTION</b>    | <b>VIEW</b>   | Options:                                                       |
|                  |               | <b>VIEW</b> View graphs of characteristic and its derivatives. |
|                  |               | <b>LIST</b> List values of B and H.                            |

### Notes

The **BHDATA** command displays the material characteristics used by the analysis programs. The parameter, **MATERIAL**, specifies the material name or number. A list of material names can be obtained with **MATERIAL=!**.

The two **OPTIONS** allow:

- **VIEW**: display graphs:
  - $BvH$  (data values)
  - $BvH$  (interpolated values)
  - $MvH$  (interpolated values)
  - $\mu vH$  (interpolated values)
  - $\frac{\partial \mu}{\partial H} vH$  (interpolated values)

The data can be also exported to a BH file using the **Export** option from the BH Viewer's **File** menu or the  icon on the toolbar.

The BH Viewer also displays  $\delta\mathbf{B}/\delta\mathbf{H}$  from the last 2 points on a BH curve and  $\mathbf{M}$  from the last point on the curve.

- **LIST:** list values of **B** and **H**.

For isotropic or laminated materials, one curve is displayed; for anisotropic materials, the curve for the specified local coordinate **DIRECTION** is displayed.

## The **BODY** Command

---

### Summary

Calculate Lorentz forces or voltages in conductors.

### Toolbutton



### Command line parameters

|            |             |                                                                                                                                                                                                                                                                                                     |
|------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command    | <b>BODY</b> |                                                                                                                                                                                                                                                                                                     |
| Parameter  | Default     | Function                                                                                                                                                                                                                                                                                            |
| N1         | 2           | Number of gauss points in direction 1.                                                                                                                                                                                                                                                              |
| N2         | 2           | Number of gauss points in direction 2.                                                                                                                                                                                                                                                              |
| N3         | 2           | Number of gauss points in direction 3.                                                                                                                                                                                                                                                              |
| X0         | 0           | X coordinate of point of action for torque.                                                                                                                                                                                                                                                         |
| Y0         | 0           | Y coordinate of point of action for torque.                                                                                                                                                                                                                                                         |
| Z0         | 0           | Z coordinate of point of action for torque.                                                                                                                                                                                                                                                         |
| TIMEOPTION | AVERAGE     | Steady-state ac options:<br><br>AMPLITUDE      Calculate amplitude of oscillations.<br>ANGLE or NO      Calculate integrals at time of SET command.<br>AVERAGE or YES      Calculate time-average values.<br>PEAK      Calculate peak values.<br>PHASE      Calculate phase angles of oscillations. |
| OMITSELF   | NUMBER      | Omit self-forces from conductor being integrated:<br><br>LABEL      Omit all conductors with the same drive label.<br>NONE      Include all conductors.<br>NUMBER      Omit the conductor by number.                                                                                                |
| LAYER      | 1           | Layer number of a multiple turn conductor.                                                                                                                                                                                                                                                          |
| NLAYERS    | 1           | Number of layers in a multiple turn conductor.                                                                                                                                                                                                                                                      |

| Command         | <b>BODY</b>    |                                               |
|-----------------|----------------|-----------------------------------------------|
| <b>LOOP</b>     | 1              | Loop number of a multiple turn conductor.     |
| <b>NLOOPS</b>   | 1              | Number of loops in a multiple turn conductor. |
| <b>INTEGRAL</b> | <b>LORENTZ</b> | Quantity to be integrated:                    |
|                 |                | <b>LORENTZ</b> Lorentz force.                 |
|                 |                | <b>VOLTAGE</b> Voltage drop.                  |

## Notes

The **BODY** command integrates fields over volumes of the source conductors. The command operates as part of a 2-stage process:

1. form a list of conductors using [The CONDUCTOR Command \[page 727\]](#). An empty list implies all conductors.
2. integrate forces or voltages using the **BODY** command.

Two different integrals can be performed:

- **LORENTZ** forces: the integral of  $J \times B$  to give force and torque.
- **VOLTAGE** drop: integrals of vector potential and current density to give the inductive and resistive voltage drop.

Conductors which are formed from multiple turns can be subdivided to pick out individual turns or groups of turns which form layers or loops within the winding. In a solenoid, loops are groups of windings at the same radius; layers are groups of windings at the same axial coordinate. In other conductor shapes, the loops and layers are defined in similar ways. (Loops and layers will not work in circuits built from arcs, straights or bricks.) Loops, layers or individual turns can be specified by giving the numbers of loops and layers (**NLOOP**, **NLAYER**) and the **LOOP** and **LAYER** numbers.

Each section of each conductor is represented by an 8 or 20 node finite element. Gaussian quadrature integration with a selectable number of Gauss points (in the range 1 to 10, or values 16 or 32) is used to calculate the integrals.

The parameters **N1**, **N2** and **N3** specify the numbers of Gauss points in each direction:

- **N1**:
  - between corners 1 and 2 of solenoids and bricks,
  - local Y direction in racetracks, bedsteads, arcs and straights,
  - radial direction in helical and constant perimeter ends;
- **N2**:
  - between corners 2 and 3 of solenoids and bricks,
  - local X direction in racetracks and bedsteads,
  - azimuthal direction in helical and constant perimeter ends;
- **N3**: the direction of current flow.

## Lorentz force integrals

Unless **BODY** is used for conductor only models it is essential that the conductors are completely enclosed in the finite element mesh. If the mesh does not completely contain the conductor the **SYMMETRY** and reflection parameters of [The SYMMETRY Command \[page 811\]](#) must be used to recreate the complete model.

The torque calculation uses the parameters **X0**, **Y0** and **Z0** to define the fixed point.

The system variables **FX**, **FY** and **FZ** and **TORQX**, **TORQY** and **TORQZ** are updated with the values of force and torque.

The program calculates the force acting at the centroid of each section of each conductor. The coordinates and force on each section are listed in the file *Post\_n.lp*. The total force on each conductor and the total force on all conductors are displayed and listed in the file. If a conductor has reflections or symmetries, all copies are treated as one conductor.

Users should critically examine the forces to ensure that the net force on the conductor set has not arisen from field cancellation errors. The accuracy of force calculations can be enhanced by use of the integral conductor field recovery option ([The SET Command \[page 798\]](#) with **COIL=INTEGRATE**). This option uses the values of **CURD** and **TOLERANCE** in the conductor definition which can be reset using [The CONDUCTOR Command \[page 727\]](#). Changing **CURD** allows conductors to be switched on or off during the force calculations in order to find the self and mutual components of the force. Changing **TOLERANCE** further affects the accuracy of the conductor field calculations.

The force of a closed conductor loop on itself should be zero. Because of cancellation effects it is difficult to achieve a precise zero for some conductor geometries (non-rectangular cross-sections, etc.). The self-force error can be omitted by using the **OMITSELF** parameter:

- **OMITSELF=NUMBER** should be used for closed loop conductors (solenoids, racetracks, bedsteads, helical and constant perimeter ends). It has no effect on arcs, straights or bricks.
- **OMITSELF=LABEL** should be used for arcs, straights and bricks, assuming that all the conductors forming a closed loop have the same drive label.
- The self-force is omitted by not calculating the field from the conductor or conductors being integrated over. This option only has an effect with the integral conductor field recovery option ([The SET Command \[page 798\]](#) with **COIL=INTEGRATE**).

## Steady-state ac

In steady-state alternating current models, the force is a function of time with the form

$$F = A + B\cos(2\omega t) - C\sin(2\omega t) \quad (7.1)$$

The values *B* and *C* have little meaning on their own. The time-average value, *A* is the value commonly required. This and other values can be calculated directly using the **TIMEOPTION** parameter:

- **ANGLE** around ac cycle: *F*, with  $2\omega t$  given by **ACTIME** in [The SET Command \[page 798\]](#);
- **AMPLITUDE** of oscillations:  $(B^2+C^2)^{1/2}$ ;

- time-**AVERAGE**:  $A$ ;
- **PEAK** value: if  $A > 0$ ,  $A + (B^2 + C^2)^{1/2}$ ; if  $A < 0$ ,  $A - (B^2 + C^2)^{1/2}$ ;
- **PHASE** angle of oscillations:  $\text{atan}(-C/B)$ .

## Voltage drop integrals

The integrals for voltage drop integral can only be done for volume meshed conductors in circuits.

The values are stored in system variables. For conductor number  $n$ :

- inductive voltage drop:  $\text{VL}_n$ ;
- resistive voltage drop:  $\text{VR}_n$ .

In steady-state alternating current models, only instantaneous values can be calculated, using the value of time given by [The SET Command \[page 798\]](#). The **TIMEOPTION** parameter is ignored.

## The **BUFFER** Command

---

### Summary

Select the active field buffer, delete and rename buffers.

### Toolbutton



### Command line parameters

| Command        | <b>BUFFER</b> |                                                                      |
|----------------|---------------|----------------------------------------------------------------------|
| Parameter      | Default       | Function                                                             |
| <b>NAME</b>    | <i>none</i>   | Name of field buffer. * for all buffers, with <b>OPTION=DELETE</b> . |
| <b>OPTION</b>  | <b>SWITCH</b> | Command option:                                                      |
|                |               | <b>CREATENEW</b> Change the mode for new buffers.                    |
|                |               | <b>DELETE</b> Delete <b>NAMEd</b> buffer.                            |
|                |               | <b>OVERWRITE</b> Change the mode for new buffers.                    |
|                |               | <b>RENAME</b> Rename the <b>NAMEd</b> buffer.                        |
|                |               | <b>SWITCH</b> Make the <b>NAMEd</b> buffer the active buffer.        |
| <b>NEWNAME</b> | <i>none</i>   | New name for the buffer.                                             |

### Notes

When fields are calculated the values are stored in a named buffer. The **BUFFER** command is used to **SWITCH** to the buffer which will be used when field values are displayed or processed; it can also **DELETE** and **RENAME** buffers and select whether buffers should be overwritten.

### Commands which use field buffers

The following commands can read field values from field buffers. If necessary, they should be preceded by a **BUFFER** command to **SWITCH** to the required buffer.

- [The FIT Command \[page 747\]](#)
- [The MAP Command \[page 775\]](#)
- [The PLOT Command \[page 783\]](#)

- [The TABLE Command \[page 817\]](#)
- [The VIEW Command \[page 840\]](#)

The GUI dialogs for the above commands issue the **BUFFER** command to **SWITCH** to the buffer and to update the dialog with the field components which are available.

The following commands write to named buffers. They each have a parameter **BUFFER** for specifying the name of the buffer.

- [The ARC Command \[page 685\]](#)
- [The CARTESIAN Command \[page 710\]](#)
- [The CIRCLE Command \[page 718\]](#)
- [The FIT Command \[page 747\]](#)
- [The LINE Command \[page 768\]](#)
- [The POINT Command \[page 787\]](#)
- [The POLAR Command \[page 788\]](#)
- [The SPHERICAL Command \[page 807\]](#)
- [The TABLE Command \[page 817\]](#)

## Buffer names

Default buffer names are the same as the commands which create them. In **CREATENEW** mode, buffers are not overwritten; if an existing name is used, the program will append an index number to make the name unique. In **OVERWRITE** mode, a new buffer will overwrite an existing buffer of the same name. When the program starts, **OVERWRITE** mode is selected.

Buffer names are not case sensitive, but the program does preserve the case of the name as first defined.

Buffers can be **RENAMEd** or **DELETEd**. Use **BUFFER OPTION=DELETE NAME=\*** to delete all the buffers associated with the currently loaded database. If a buffer which represents values along a line is deleted, the corresponding graph buffer object will be deleted.

## Buffers and databases

Each active database has its own set of buffers. Reloading an active database will make the buffers associated with that database available. See "The LOAD Command" on page 770.

Buffers are lost when the program is ended or a database is deactivated. The field values can be saved in table files using [The TABLE Command \[page 817\]](#).

## The **CARTESIAN** Command

---

### Summary

Calculate fields over a patch specified in xyz coordinates.

### Toolbutton



### Command line parameters

| Command   | <b>CARTESIAN</b> |                                                   |
|-----------|------------------|---------------------------------------------------|
| Parameter | Default          | Function                                          |
| X1        | 0                | X-coordinate of the first corner of the surface.  |
| Y1        | 0                | Y-coordinate of the first corner of the surface.  |
| Z1        | 0                | Z-coordinate of the first corner of the surface.  |
| X2        | <i>none</i>      | X-coordinate of the second corner of the surface. |
| Y2        | <i>none</i>      | Y-coordinate of the second corner of the surface. |
| Z2        | <i>none</i>      | Z-coordinate of the second corner of the surface. |
| X3        | <i>none</i>      | X-coordinate of the third corner of the surface.  |
| Y3        | <i>none</i>      | Y-coordinate of the third corner of the surface.  |
| Z3        | <i>none</i>      | Z-coordinate of the third corner of the surface.  |
| X4        | <i>none</i>      | X-coordinate of the fourth corner of the surface. |
| Y4        | <i>none</i>      | Y-coordinate of the fourth corner of the surface. |
| Z4        | <i>none</i>      | Z-coordinate of the fourth corner of the surface. |
| X5        | <i>none</i>      | X-coordinate of the mid-point on side 1.          |
| Y5        | <i>none</i>      | Y-coordinate of the mid-point on side 1.          |
| Z5        | <i>none</i>      | Z-coordinate of the mid-point on side 1.          |
| X6        | <i>none</i>      | X-coordinate of the mid-point on side 2.          |
| Y6        | <i>none</i>      | Y-coordinate of the mid-point on side 2.          |
| Z6        | <i>none</i>      | Z-coordinate of the mid-point on side 2.          |

| Command   | <b>CARTESIAN</b> |                                            |
|-----------|------------------|--------------------------------------------|
| Parameter | Default          | Function                                   |
| X7        | none             | X-coordinate of the mid-point on side 3.   |
| Y7        | none             | Y-coordinate of the mid-point on side 3.   |
| Z7        | none             | Z-coordinate of the mid-point on side 3.   |
| X8        | none             | X-coordinate of the mid-point on side 4.   |
| Y8        | none             | Y-coordinate of the mid-point on side 4.   |
| Z8        | none             | Z-coordinate of the mid-point on side 4.   |
| CORNERS   | 4                | Number of points defining surface: 4 or 8. |
| N1        | 10               | Number of points on sides 1 and 3.         |
| N2        | 10               | Number of points on sides 2 and 4.         |
| BUFFER    | Cartesian        | Name of buffer to store field values.      |

## Notes

The **CARTESIAN** command evaluates field quantities on 4 or 8-noded surface patches.

- If the number of points in both directions is greater than one, the results can be displayed by [The MAP Command \[page 775\]](#).
- Single lines of points can be displayed by [The PLOT Command \[page 783\]](#).

For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)).

The patch is specified by its corner points (**X1,Y1,Z1**, **X2,Y2,Z2**, **X3,Y3,Z3** and **X4,Y4,Z4**) and optional mid-side points (**X5,Y5,Z5**, **X6,Y6,Z6**, **X7,Y7,Z7** and **X8,Y8,Z8**). The mid-side points are ignored if **CORNERS=4**, and included if **CORNERS=8**. Point 5 is on side 1 (point 1 to point 2); point 6 is on side 2 (point 2 to point 3); point 7 is on side 3 (point 3 to point 4); point 8 is on side 4 (point 4 to point 1).

The positions of the points are defined in the local coordinate system defined with [The SET Command \[page 798\]](#). The field vectors are evaluated in the Global Coordinate System.

The field quantities are evaluated at **N1\*N2** points. The coordinates of points other than the corner and mid-side points are found by interpolation using 4 or 8-noded isoparametric two-dimensional finite element shape functions.

The fields are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## The **CEDITOR** Command

---

### Summary

View the circuit in the Circuit Editor.

### Toolbutton



### Command line parameters

|               |                      |                                                                       |
|---------------|----------------------|-----------------------------------------------------------------------|
| Command       | <b>CEDITOR</b>       |                                                                       |
| Parameter     | Function             |                                                                       |
| <b>OPTION</b> | Command option:      |                                                                       |
|               | <b>CLEAR</b>         | Clear all circuit data and close the Circuit Editor window.           |
|               | <b>CLOSE</b>         | Close the Circuit Editor window.                                      |
|               | <b>COPYCOMPONENT</b> | Copy a circuit component.                                             |
|               | <b>IMPORT</b>        | Import a circuit from a file adding to existing data.                 |
|               | <b>LOAD</b>          | Clear all circuit data and load a circuit from a file.                |
|               | <b>MODIFY</b>        | Modify the value of the property of a component.                      |
|               | <b>PLACE</b>         | Position an unused component in the circuit.                          |
|               | <b>PRINT</b>         | Create a picture of the circuit in a file.                            |
|               | <b>REVERSE</b>       | Change the direction of a component in the circuit.                   |
|               | <b>SAVE</b>          | Save circuit data in a file.                                          |
|               | <b>START</b>         | Open the Circuit Editor window with any existing circuit information. |
|               | <b>UPDATE</b>        | Copy any changes made interactively into the model.                   |

|                        |                                                                                                                       |
|------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>FILENAME</b>        | Name of circuit data file for <b>LOAD</b> , <b>IMPORT</b> or <b>SAVE</b> .<br>Name of picture file for <b>PRINT</b> . |
| <b>COMPONENTNAME</b>   | Name of component for <b>MODIFY</b> .                                                                                 |
| <b>PROPERTYNAME</b>    | Name of property for <b>MODIFY</b> .                                                                                  |
| <b>VALUE</b>           | New value of property for <b>MODIFY</b> .                                                                             |
| <b>SOURCECOMPONENT</b> | Name of the source component in <b>COPYCOMPONENT</b> or the adjacent component in <b>PLACE</b> .                      |
| <b>PREFIX</b>          | A prefix to be added to the names of components <b>IMPORT</b> ed.                                                     |

## Notes

The **CEDITOR** command controls the Circuit Editor. The Circuit Editor is designed for interactive use. Once started, wires and components can be drawn on the Circuit Editor window (see [Figure 7.1, below](#)) and properties can be assigned to components using its menus, toolbars and property sheets.

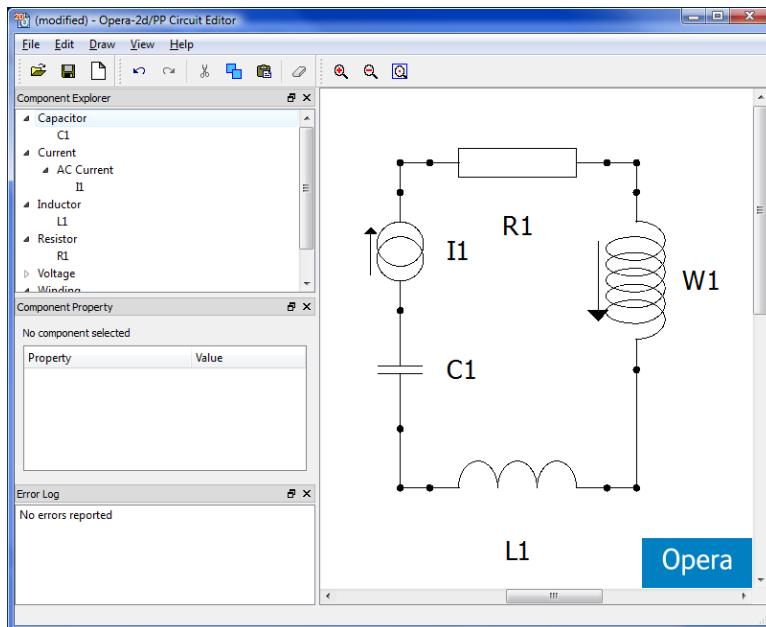


Figure 7.1 A simple RLC circuit including an Opera winding

The figure shows a circuit from Opera-2d. However, the circuit editor is the same and circuit data files (**\*.vfc**) are compatible between Opera-2d and Opera-3d. Some of the options shown depend on the current analysis type; for example **Functional Voltage** is not available if harmonic analysis has been selected in the Modeller.

When the Circuit Editor is closed, the data is transferred to the Modeller and will be saved to any data file (\*.**opc**). The Circuit Editor can save and load its own data files (\*.**vfc**) so that circuit information can be transferred between models.

## Options

In normal interactive use, the Modeller issues the command:

```
ceditor option=start
```

After the circuit data has been prepared (see [Circuit Editor Interaction \[page 715\]](#)), the action of closing the window gives the option of accepting or cancelling the changes in the model.

The other **OPTION**s allow the Circuit Editor to be controlled from the command line or a command file. The topology of a circuit can only be modified interactively but the properties of circuit components can be changed by command line.

Options to control the Circuit Editor window:

- **START**: to open the Circuit Editor window.
- **CLOSE**: to close the Circuit Editor window, accepting any changes made interactively since the last **UPDATE**.
- **PRINT**: to create a picture file of the circuit. The file format is determined by the file type of the **FILENAME**. Files are overwritten without asking.

The following **OPTION**s do not require the Circuit Editor window to be open:

- **CLEAR**: to delete all circuit data and close the Circuit Editor window if it is open. Any circuit data stored in the application will also be cleared.
- **MODIFY**: to change the value of a property of a component, e.g.  
`ceditor option=modify component=r1 resistance 10`

**MODIFY** is designed for scripting, not for interactive use. Component names and property names are not case sensitive but they cannot be abbreviated. Names which contain spaces must be enclosed in single quotation marks, e.g.

```
propertyname='resistance per unit length'
```

If **MODIFY** is used to change a property value of a component, it is necessary for the component to be deselected and reselected before the changed property can be seen in the Circuit Editor window.

Changes made with **MODIFY** cannot normally be undone. The behaviour is undefined if the circuit is also being modified graphically.

- **COPYCOMPONENT**: to copy the component given by **SOURCECOMPONENT** to create a new component with the name given by **COMPONENTNAME**. The new component can then be added to the circuit using the **PLACE** option.
- **PLACE**: to insert an unused component given by **COMPONENTNAME** so that it is placed immediately after the component given by **SOURCECOMPONENT**.
- **REVERSE**: to reverse the component given by **COMPONENTNAME** in the circuit.

- **UPDATE:** to copy any changes made interactively into the model<sup>1</sup>. It is necessary to **UPDATE** after a set of **MODIFY** or **LOAD** commands to accept changes, otherwise a question will be asked to confirm the changes.
- **LOAD, IMPORT** and **SAVE:** to access Circuit Editor data files (\*.vfc).  
**INSERT** can rename the components read from a **vfc** file so that they do not match the names of existing components. This is done by adding a **PREFIX** to the names of the new components. The **PREFIX** can be empty if none of the names in the **IMPORT**ed circuit is already used by components in the existing circuit.

The **CEDITOR** command cannot be used if existing circuit data has been defined using the **CIRCUIT** command. To access the **CEDITOR** command all previously defined circuit data must be cleared.

However the **CIRCUIT** command can be used to view the circuit data created by the **CEDITOR** command, if there are no errors in the circuit.

## Circuit Editor Interaction

### Drawing wires

Double clicking in empty space on the diagram can be used to start a new wire. Once started, double-clicking continues the wire. Double-clicking on an existing wire makes the new wire join the exiting wire at the position clicked. Wires can pass over an existing wire, and no connection node will be formed.

As the wire is being drawn, a preview of the wires will be shown if it is possible to create a connection between the start and end points with 1 or 2 wires.

Press the left button to complete the wire.

Cancel the drawing by moving out of the view pane or by clicking the right mouse button.

### Drawing components

Position the mouse over a component or component type in the left hand list view.

Press and hold the left mouse button and drag the component into the diagram. The position of the component will be displayed as the cursor moves if it is possible to drop the component at that position. Release the left mouse button to drop the component into the diagram at the cursor position.

Components can be dropped onto empty space or onto wires that are sufficiently long to include the component. If positioned on a vertical wire, the component will be rotated to form connections with the wire.

To cancel the drop move the cursor back into the left hand component list view.

---

<sup>1</sup>The **UPDATE** option is not available in the Post-Processor.

Double clicking over a component will select the component for property modification in the left hand property sheet. Values can be entered into the mutual inductance matrix for the whole circuit using the menu route: **Edit -> Set Mutual Inductance**.

## Opera-3d windings

Circuit winding elements available in the Modeller are displayed in the **Component Explorer** under **Windings**. These should be dragged into the relevant positions in the circuit diagram. The mesh type of the winding can only be set in the Modeller.

## Opera-3d bulk eddy current conductors

Bulk Eddy current conductors available in the Modeller are displayed in the **Component Explorer** under **Bulk Conductor**. These should be dragged into the relevant positions in the circuit diagram. Values for the symmetry factors for the length and cross-sectional area of the conductor must be set if the complete conductor is modelled by a symmetric sub-section.

## Selecting components to move them

Components can be selected by:

- Pressing the left mouse button while over a component.
- Pressing the **Ctrl** or **Shift** key while selecting a second component will allow multiple components to be selected. If **Ctrl** or **Shift** is not pressed, all other components will be deselected when the mouse button is released.
- Pressing the left mouse button over empty space in the diagram, holding and dragging to create a selection area.

If the selection area starts from left and moves right, components that are completely enclosed in the selection area will be selected.

If the selection area starts at the right and is dragged left, any component that intersects the selection area will be included.

## Moving or copying selected components

Select the components to be moved as above.

Press and hold the left mouse button while the cursor is over one of these items. As the cursor is moved a preview of the moved components will be displayed. The items will not be displayed if the position to which they are moved is invalid.

Release the left mouse button when the components are in the correct place.

By default, connections between components will be maintained for the moving parts.

To disconnect the moving components press the **Ctrl** key while moving the cursor or when the button is released.

To take a copy of the components and leave the existing components in-situ, press the **shift** key while moving and dropping the components.

The mouse button press will perform a select on a new component, allowing movement of a single component by pressing and moving as a single operation.

## Copy/Cut and Paste

Selected items can be copied by using the **Edit -> Copy** menu item (**Ctrl+C**).

Items can be pasted by pressing **Edit -> Paste** menu item (**Ctrl+V**). On pressing the paste menu item, the Circuit Editor enters a paste positioning mode. A preview of where the components will be pasted is shown as the cursor moves over the diagram.

To position the components, press the left mouse button. To cancel the paste operation, press the right mouse button.

## Other operations

Other operations can be performed on components once they are selected using a context menu which appears after a right mouse button click.

Components can be:

- Reversed, connections are maintained.
- Rotated, connections are broken and the operation will only happen if the rotated component can be positioned in its new orientation.
- Deleted or Removed. Removed components are removed from the diagram but remain in the component list view.

Finally, a set of operations is available to help with building and saving the circuits on file:

- Unlimited Undo and Redo until a further change is made.
- Save, Load or Import a transferable circuit file (**\*.vfc**).
- Export to Opera-3d as a command file (**\*.comi**).
- Generate a report containing details of all the components in a **html** file.

## The CIRCLE Command

---

### Summary

Calculate fields along a circular arc specified by radius and angles.

### Toolbutton



### Command line parameters

| Command   | CIRCLE  |                                                                     |
|-----------|---------|---------------------------------------------------------------------|
| Parameter | Default | Function                                                            |
| RADIUS    | none    | Radius of circular arc.                                             |
| TH1       | 0       | Azimuthal-coordinate of the first point on the arc.                 |
| TH2       | 360     | Azimuthal-coordinate of the last point on the arc.                  |
| ZC        | 0       | Axial coordinate of arc.                                            |
| NP        | 100     | Number of steps between the first and last points, i.e NP+1 points. |
| BUFFER    | Circle  | Name of buffer to store field values.                               |

### Notes

The CIRCLE command evaluates field quantities along a circular arc for use by

- The FIT Command [page 747]
- The PLOT Command [page 783].

For each field point all the currently available system variables are calculated and stored in a named field BUFFER (see System Variables [page 657]).

The circular arc is specified by its radius (RADIUS), the azimuthal coordinates of its end points (TH1 and TH2) and the axial coordinate (ZC) (see Figure 7.2, on page 769). The plane of the arc is the local XY plane through local coordinates (0, 0, ZC), using the local coordinate system defined with The SET Command [page 798].

Arcs can also be defined using The ARC Command [page 685], except that it does not allow arcs of 180 to 360 degrees to be defined.

The vector field quantities are evaluated with respect to the Global Coordinate System for the active file.

The field quantities are evaluated at  $NP+1$  points along the arc and are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## Graphs of field values

The **Fields on a Circle** dialog runs 3 commands:

- [The AXESVIEW Command \[page 689\]](#) (optional) to create a new graph.
- The **CIRCLE** command.
- [The DATALINE Command \[page 735\]](#) to add the line of calculated values to the graph.

## The **CLEAR** Command

---

### Summary

Clear all data and re-initialize all commands.

### Toolbutton



### Command line parameters

|                        |              |                                                                                         |
|------------------------|--------------|-----------------------------------------------------------------------------------------|
| Command                | <b>CLEAR</b> |                                                                                         |
| Parameter              | Default      | Function                                                                                |
| <b>USERVARIABLES</b>   | <b>YES</b>   | Clears all numeric user variables and user defined functions: <b>YES</b> or <b>NO</b> . |
| <b>STRINGVARIABLES</b> | <b>YES</b>   | Clears all string variables: <b>YES</b> or <b>NO</b> .                                  |
| <b>LOGFILES</b>        | <b>NO</b>    | Restarts <b>Ip</b> and <b>log</b> files: <b>YES</b> or <b>NO</b> .                      |
| <b>GRAPHING</b>        | <b>YES</b>   | Clear all graph data: <b>YES</b> or <b>NO</b> .                                         |
| <b>PYTHON</b>          | <b>YES</b>   | Reset the Python interpreter: <b>YES</b> or <b>NO</b> .                                 |
| <b>OBJECTSTORE</b>     | <b>YES</b>   | Clear all <b>OperaObjects</b> from the object store: <b>YES</b> or <b>NO</b> .          |

### Notes

This command is used to reset the Post-Processor to its initial state. The parameters can be used from the console or in command files to adjust the behaviour. From the GUI, the default settings are always used.

- **USERVARIABLES=NO**: numeric user variables are not deleted. This setting allows **CLEAR** to be used in a command loop. Without this setting a **CLEAR** command in a command loop will cause the command input to be aborted, i.e. any pending commands from the command loop or from any other loops or any command input file containing the loop will be ignored and command input will return to the user.

If numeric user variables are not deleted then user defined functions are kept as well.

- **STRINGVARIABLES=NO**: string variables created by the user are not deleted.
- **LOGFILES=YES**: the contents of the **Ip** and **log** files (see [Output Files \[page 25\]](#)) will be deleted and only subsequent input and output will be recorded.

- **GRAPHING=YES**: all the graph buffers, lines, vectors and graphs will be deleted. The **Default** graph will be recreated.
- **PYTHON=YES**: all variables and functions in the Python memory will be cleared.
- **OBJECTSTORE=YES**: OperaObjects (used for transfer of data between Opera and Python "Python Arrays in Post-Processing" on page 891) will be cleared.

Following a **CLEAR** command (unless the **CLEAR** command is in a command input file), the program looks for a file called ***opera.comi***, first in the current project folder and then in the user's home folder, and if it exists, it is opened and read as a **\$ COMINPUT** file (see [Command Input Files \[page 61\]](#)) before control is passed back to the user. This allows the user to supply an individual choice of default values for commands or define frequently used **\$ CONSTANTs** and **\$ PARAMETERs** (see [User Variable Commands \[page 55\]](#)).

## The COLOUR Command

### Summary

Adjust the colours used for the display.

### Toolbutton



### Command line parameters

| Command   | COLOUR  |                                                          |
|-----------|---------|----------------------------------------------------------|
| Parameter | Default | Function                                                 |
| OPTION    | SET     | Option:                                                  |
|           |         | <b>EXPORT</b>                                            |
|           |         | Save the current set of colours as a command input file. |
|           |         | <b>LOAD</b>                                              |
|           |         | Load current values as defaults.                         |
|           |         | <b>RESTORE</b>                                           |
|           |         | Restore the default set of colours.                      |
|           |         | <b>SET</b>                                               |
|           |         | Set new values.                                          |

| Command            | COLOUR      |                                  |                                                                                    |
|--------------------|-------------|----------------------------------|------------------------------------------------------------------------------------|
| Parameter          | Default     | Function                         |                                                                                    |
| <b>LABEL</b>       | <b>TEXT</b> | Label of colour to be redefined. |                                                                                    |
|                    |             | <b>BACKGROUND</b>                | The background colour.                                                             |
|                    |             | <b>CONDUCTORS</b>                | The colour of conductors.                                                          |
|                    |             | <b>HIGHLIGHTING</b>              | The high-light colour for selected conductors and for field points.                |
|                    |             | <b>MAPFIRSTCONTOUR</b>           | The colour of the minimum contour level ( <b>MAP</b> command).                     |
|                    |             | <b>MAPLASTCONTOUR</b>            | The colour of the maximum contour level ( <b>MAP</b> command).                     |
|                    |             | <b>SURFACEFIRSTCONTOUR</b>       | The colour of the minimum contour level ( <b>THREED</b> and <b>VIEW</b> commands). |
|                    |             | <b>SURFACELASTCONTOUR</b>        | The colour of the maximum contour level ( <b>THREED</b> and <b>VIEW</b> commands). |
|                    |             | <b>TEXT</b>                      | The colour of text and axes.                                                       |
|                    |             | <b>TRAJECTORIES</b>              | The colour of trajectories.                                                        |
|                    |             | <b>VECTORINSIDE</b>              | The colour of the inside of vector cones.                                          |
|                    |             | <b>VECTOROUTSIDE</b>             | The colour of the outside of vector cones.                                         |
|                    |             | <i>material_name</i>             | The colour of a material.                                                          |
| <b>RED</b>         | <i>none</i> | Amount of red for colour.        |                                                                                    |
| <b>GREEN</b>       | <i>none</i> | Amount of green for colour.      |                                                                                    |
| <b>BLUE</b>        | <i>none</i> | Amount of blue for colour.       |                                                                                    |
| <b>TRANSLUCENT</b> | <b>NO</b>   | <b>NO</b>                        | Switch off colour translucency                                                     |
|                    |             | <b>YES</b>                       | Set colour property to be translucent                                              |

## Notes

The **COLOUR** command allows the user to adjust the colours used for parts of the display. With **OPTION=SET**, the colour identified by **LABEL** is redefined using the new values for **RED**, **GREEN** and **BLUE** given by numbers in the range 0 to 255 and a **TRANSLUCENT** flag.

- **xxxxFIRSTCONTOUR** and **xxxxLASTCONTOUR**: The program interpolates colours between the colours used for the extreme values.
- **CONDUCTORS** and materials: When a database is loaded the colours used by the Modeller or Pre-Processor are also loaded.

Making a material colour translucent (**TRANSLUCENT=YES**) makes other materials inside it visible. Only one material (or all the contour colours) can be translucent at a time. All other materials will be opaque (**TRANSLUCENT=NO**). There are two sets of contour colours, one for the **MAP** command, and one for the **THREED** command (which are also used for particle tracks on the 3D display in the **VIEW** command). Only one set, or one material, can be made translucent at a time.

Colours can be saved to a command input file called ***post\_colours.comi*** with **OPTION=EXPORT**. The default set of colours can be restored using **OPTION=RESTORE**.

## The **COMBINE** Command

---

### Summary

Select both magnetic and electric fields for particle trajectory calculations.

### Toolbutton



### Command line parameters

| Command         | <b>COMBINE</b> |                                                                                           |
|-----------------|----------------|-------------------------------------------------------------------------------------------|
| Parameter       | Default        | Function                                                                                  |
| <b>BOTH</b>     | <b>YES</b>     | Combine magnetic and electric fields for particle trajectories: <b>YES</b> or <b>NO</b> . |
| <b>SYMMETRY</b> | 1              | Rotational symmetry of additional field around local Z axis.                              |
| <b>RXY</b>      | <b>NO</b>      | Reflection of additional field in local XY plane.                                         |
|                 |                | <b>NO</b> No reflection.                                                                  |
|                 |                | <b>YES</b> Field parallel to XY.                                                          |
|                 |                | <b>INVERSE</b> Field normal to XY.                                                        |
| <b>RYZ</b>      | <b>NO</b>      | Reflection of additional field in local YZ plane.                                         |
|                 |                | <b>NO</b> No reflection.                                                                  |
|                 |                | <b>YES</b> Field parallel to YZ.                                                          |
|                 |                | <b>INVERSE</b> Field normal to YZ.                                                        |
| <b>RZX</b>      | <b>NO</b>      | Reflection of additional field in local ZX plane.                                         |
|                 |                | <b>NO</b> No reflection.                                                                  |
|                 |                | <b>YES</b> Field parallel to ZX.                                                          |
|                 |                | <b>INVERSE</b> Field normal to ZX.                                                        |
| <b>MAGSCALE</b> | 1              | Scaling factor for magnetic field.                                                        |
| <b>ELESCALE</b> | 1              | Scaling factor for electric field.                                                        |

## Notes

The **COMBINE** command switches combined field particle tracking on or off. Before it is used, a database containing electric and magnetic fields must be activated and loaded. This can be achieved in two ways:

1. Additional magnetic fields might already be present in a Charged Particle simulation. They are used in the Charged Particle solver for particle trajectory calculations. They can exist as a result of conductor data in the database file or can be added to the database using the **TABLE** command for [Copying fields from one simulation to another \[page 820\]](#).
2. [The TABLE Command \[page 817\]](#) can be used to copy electric or magnetic fields from one simulation into another so that the receiving simulation has both field types.

N.B. It is not sufficient to have an electric field simulation and a magnetic field simulation in the same database. Both fields must be in the same simulation.

The geometric symmetry of the electric and magnetic models must be the same. However, the field symmetry will often be different. If rotational and reflection symmetry is necessary on the **ACTIVATE** command for the model already activated, then the appropriate additional field symmetry should be specified on the **COMBINE** command using parameters **SYMMETRY**, **RXY**, **RYZ** and **RZX**.

The electric and magnetic fields can be scaled using the factors **MAGSCALE** and **ELESCALE**.

After switching the combined tracking option on, [The TRACK Command \[page 831\]](#) can be used in the usual way.

## The CONDUCTOR Command

### Summary

Define, erase, modify, list, import and export conductors.

### Toolbuttons



### Command line parameters

| Command       | CONDUCTOR             |                                                                       |
|---------------|-----------------------|-----------------------------------------------------------------------|
| Parameter     | Default               | Function                                                              |
| <b>ACTION</b> | <b>ADD</b>            | Action:                                                               |
|               |                       | <b>ADD</b> Add conductor(s) to list.                                  |
|               |                       | <b>DEFAULT</b> Set default values to match selected conductors.       |
|               |                       | <b>DEFINE</b> Define a new conductor.                                 |
|               |                       | <b>DEFRESET</b> Reset default values to match last conductor defined. |
|               |                       | <b>ERASE</b> Erase selected conductors.                               |
|               |                       | <b>EXPORT</b> Export conductors to a data file.                       |
|               |                       | <b>HIDE</b> Hide conductors which are displayed.                      |
|               |                       | <b>IMPORT</b> Import conductors from a data file.                     |
|               |                       | <b>LIST</b> List conductor data.                                      |
|               |                       | <b>MODIFY</b> Modify selected conductors.                             |
|               |                       | <b>PICK</b> Add or remove conductor from list.                        |
|               |                       | <b>REMOVE</b> Remove conductor(s) from list.                          |
|               |                       | <b>RESET</b> Empty list of conductors.                                |
| <b>LABEL</b>  | <b>ALL_CONDUCTORS</b> | Show conductors previously hidden.                                    |
|               |                       | <b>STARTPICK</b> Switch on picking facility.                          |
|               |                       | <b>STOPPICK</b> Switch off picking facility.                          |
|               |                       | <b>TOGGLEADD</b> Toggles conductor(s) in the list.                    |
|               |                       | Conductors to be added or removed from selection list:                |
| <b>TYPE</b>   | <b>SOLENOID</b>       | <i>number</i> Conductor number.                                       |
|               |                       | <i>drive_label</i> Drive label.                                       |
|               |                       | <b>ALL_CONDUCTORS</b> All conductors.                                 |
| <b>XCEN1</b>  | 0                     | X coordinate of origin of local system 1.                             |

| Command   | CONDUCTOR |                                                                                                     |
|-----------|-----------|-----------------------------------------------------------------------------------------------------|
| Parameter | Default   | Function                                                                                            |
| YCEN1     | 0         | Y coordinate of origin of local system 1.                                                           |
| ZCEN1     | 0         | Z coordinate of origin of local system 1.                                                           |
| THETA1    | 0         | Euler angle $\theta$ (local system 1).                                                              |
| PHI1      | 0         | Euler angle $\phi$ (local system 1).                                                                |
| PSI1      | 0         | Euler angle $\psi$ (local system 1).                                                                |
| XCEN2     | 0         | X coordinate of origin of local system 2.                                                           |
| YCEN2     | 0         | Y coordinate of origin of local system 2.                                                           |
| ZCEN2     | 0         | Z coordinate of origin of local system 2.                                                           |
| THETA2    | 0         | Euler angle $\theta$ (local system 2).                                                              |
| PHI2      | 0         | Euler angle $\phi$ (local system 2).                                                                |
| PSI2      | 0         | Euler angle $\psi$ (local system 2).                                                                |
| IRXY      | 0         | Reflection code in xy plane of local system 1.                                                      |
| IRYZ      | 0         | Reflection code in yz plane of local system 1.                                                      |
| IRZX      | 0         | Reflection code in zx plane of local system 1.                                                      |
| SYMMETRY  | 1         | Symmetry code.                                                                                      |
| CURD      | 100       | Current density.                                                                                    |
| TOLERANCE | 0         | Tolerance on flux density (zero for default tolerance, negative for single filament approximation). |
| PHASE     | ONE       | Drive label.                                                                                        |
| X1        | 3         | X coordinate of corner of conductor cross section.                                                  |
| Y1        | 3         | Y coordinate of corner of conductor cross section.                                                  |
| X2        | 3         | X coordinate of corner of solenoid cross section.                                                   |
| Y2        | 4         | Y coordinate of corner of solenoid cross section.                                                   |
| X3        | 4         | X coordinate of corner of solenoid cross section.                                                   |
| Y3        | 4         | Y coordinate of corner of solenoid cross section.                                                   |
| X4        | 4         | X coordinate of corner of solenoid cross section.                                                   |
| Y4        | 3         | Y coordinate of corner of solenoid cross section.                                                   |
| CU1       | 0         | Curvature of cross section of solenoid (points 1 to 2).                                             |

| Command   | CONDUCTOR                                                                   |                                                                                                                                              |
|-----------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter | Default                                                                     | Function                                                                                                                                     |
| CU2       | 0                                                                           | Curvature of cross section of solenoid (points 2 to 3).                                                                                      |
| CU3       | 0                                                                           | Curvature of cross section of solenoid (points 3 to 4).                                                                                      |
| CU4       | 0                                                                           | Curvature of cross section of solenoid (points 4 to 1).                                                                                      |
| A         | 1                                                                           | Thickness of conductor in x or radial direction.                                                                                             |
| B         | 1                                                                           | Width of conductor in y or azimuthal direction.                                                                                              |
| H1        | 5                                                                           | Length of straight section.                                                                                                                  |
| H2        | 1                                                                           | Length of upright ( <b>BEDSTEAD</b> )<br>Local Z coordinate of midpoint of cross-over ( <b>HELIX</b> ).                                      |
| R1        | 0.5                                                                         | Radius: inner radius of arc ( <b>RACETRACK</b> , <b>BEDSTEAD</b> , <b>ARC</b> ).<br>Radius of cylinder ( <b>HELIX</b> , <b>CPEND</b> ).      |
| R2        | 0.6                                                                         | Radius: inner radius of arc ( <b>BEDSTEAD</b> ).<br>Width of cross-over ( <b>HELIX</b> ).<br>Radius of generating cylinder ( <b>CPEND</b> ). |
| PHI       | 30                                                                          | Angle of <b>ARC</b> .                                                                                                                        |
| ALPHA     | 45                                                                          | Angle of straight from mid plane of cylinder ( <b>HELIX</b> and <b>CPEND</b> ).                                                              |
| BETA      | 60                                                                          | Angle of end of helix ( <b>HELIX</b> ), or cutter ( <b>CPEND</b> ).                                                                          |
| FIT       | <b>TANGENTIAL</b>                                                           | Fit of straight section to cylinder: <b>TANGENTIAL</b> or <b>FITTING</b> ( <b>CPEND</b> ).                                                   |
| XB $n$    | <i>A bar, starting at the origin, with cross section 1x1 and length 10.</i> | X coordinate of node of 8 or 20 node brick conductor ( $1 \leq n \leq 20$ ).                                                                 |
| YB $n$    |                                                                             | Y coordinate of node of 8 or 20 node brick conductor ( $1 \leq n \leq 20$ ).                                                                 |
| ZB $n$    |                                                                             | Z coordinate of node of 8 or 20 node brick conductor ( $1 \leq n \leq 20$ ).                                                                 |
| FILE      | none                                                                        | Name of file for <b>IMPORT</b> or <b>EXPORT</b>                                                                                              |

|                |           |                                                            |
|----------------|-----------|------------------------------------------------------------|
| Command        | CONDUCTOR |                                                            |
| Parameter      | Default   | Function                                                   |
| DEFINEDCURRENT | YES       | Export defined or instantaneous values of current density: |
|                |           | NO Export values for ACTIME or current simulation.         |
|                |           | YES Export defined values.                                 |

## Notes

This command controls the definition, modification, import and export of conductors. The operation of the command is controlled by the **ACTION** parameter:

- **ACTION=DEFINE**: defines a new conductor. The definition of the parameters and how they apply to each conductor shape is given in [Conductors \[page 512\]](#).
- **ACTION=ERASE**: erases a list of conductors. Before conductors can be erased the list must be formed using actions **RESET**, **ADD**, **PICK** and **REMOVE** (see below). After conductors have been erased the remaining conductors are renumbered to form a contiguous set starting at 1 and the selection list is emptied.
- **ACTION=MODIFY**: modifies a list of conductors. Before conductors can be modified the list must be formed using actions **RESET**, **ADD PICK**, and **REMOVE** (see below). The default values of the conductor parameters should also be set to correspond to the conductors selected using **ACTION=N=DEFAULT**. After modifying, the default values can be returned to correspond to the last conductor defined using **ACTION=DEFRESET**.

It is possible to set new values for parameters to expressions involving the old values, for example to double the current densities. The command sequence to do this to all conductors would be:

```
CONDUCTOR ACTION=ADD LABEL=ALL_CONDUCTORS
CONDUCTOR ACTION=DEFAULT
CONDUCTOR ACTION=MODIFY CURD=CURD*2
CONDUCTOR ACTION=DEFRESET
```

- **ACTION=DEFAULT**: sets the default values to correspond to the selected list of conductors in preparation for modification. If the conductor type is unique, the string variable **CONDUCTOR\_TYPE** is given the appropriate value.
- **ACTION=DEFRESET**: resets the default values to correspond to the last conductor defined after using **ACTION=MODIFY**.
- **ACTION=LIST**: lists conductor data. If there is a selected list of conductors, only those conductors will be listed. Otherwise, all conductors will be listed. If the conductors have been displayed, the list includes the approximate volume. The values are approximate for curved conductor shapes because they are calculated using the displayed conductor segments which have quadratic curves rather than true circular arcs.

- **ACTION=IMPORT:** reads conductor data from a conductor data file. Conductor data files are compatible with those created by the Pre-Processor using [The CONDUCTOR Sub-command WRITE \[page 391\]](#) and the Modeller using [The EXPORT Command \[page 211\]](#). For hints on how to write compatible command scripts for all 3 programs see [Conditional Commands \[page 52\]](#).

- **ACTION=EXPORT:** writes conductor data to a file. If there is a selected list of conductors, only those conductors will be included. Otherwise, all conductors will be included.

For steady-state ac and transient simulations, there is a choice of current density values which can be used in the exported data:

- **DEFINEDCURRENT=YES:** use the current densities from the conductor definitions;
- **DEFINEDCURRENT=NO:** use the calculated values of current density corresponding to **ACTIME** or the simulation time.

If the conductor is connected to a circuit, the calculated value is always used.

- **ACTION=ADD:** adds conductors to the selected list by label. Labels can be conductor numbers, drive labels or **ALL\_CONDUCTORS**.
- **ACTION=REMOVE:** removes conductors from the selected list by label. Labels can be conductor numbers, drive labels or **ALL\_CONDUCTORS**.
- **ACTION=TOGGLEADD:** adds non-selected conductors to the selected list or removes already selected conductors from the selected list.
- **ACTION=STARTPICK** switches on conductor picking.
- **ACTION=PICK** adds conductor number given by **LABEL** to the list if it is not already in the list, or removes it if already there.

This command is automatically generated by pointing at the appropriate conductor and double-clicking the left mouse button.

- **ACTION=STOPPICK** switches off conductor picking.
- **ACTION=RESET:** empties the selected list of conductors.
- **ACTION=HIDE** or **SHOW:** temporarily hides or shows again the conductors which have been displayed.

The system variable **CONDUCTORS** holds the number of conductors.

See [Conductors \[page 512\]](#) for full details of all the conductor shapes and parameters.

## The COPYCASE Command

---

### Summary

Appends a new simulation to the active database. The new simulation is created from the loaded simulation and can be marked as a restart case.

### Toolbutton



### Command line parameters

|                   |                 |                                                                 |
|-------------------|-----------------|-----------------------------------------------------------------|
| Command           | <b>COPYCASE</b> |                                                                 |
| Parameter         | Default         | Function                                                        |
| <b>NEWCASE</b>    | <i>none</i>     | The frequency or output time for the new simulation             |
| <b>CONTINUE</b>   | <b>YES</b>      | Continue from current simulation:                               |
|                   |                 | <b>YES</b> Continue from the current solution                   |
|                   |                 | <b>NO</b> Begin a new simulation                                |
| <b>ITERATIONS</b> | 21              | Number of additional iterations for a Charged Particle analysis |
| <b>TOLERANCE</b>  | 0.01            | Convergence tolerance for a Charged Particle analysis.          |

### Notes

The **COPYCASE** command adds a new unsolved simulation to a database based on the currently loaded simulation. The details of the new simulation and the parameters used depend on the type of simulation being copied.

- Magnetostatics: the scaling factor for the new simulation is given by **NEWCASE**.
- Steady-State AC simulations: the frequency for the new simulation is given by **NEWCASE**.
- Transient simulations: the output time of the new simulation is given by **NEWCASE**. The user can choose to continue from the time of the currently loaded simulation (**CONTINUE=YES**) or to start again from zero ((**CONTINUE=NO**)).
- Charged Particle simulations: the new simulation will make some additional **ITERATIONS** to a new convergence **TOLERANCE**, starting from the existing solution.
- Other simulations: tables of temperatures, heat sources, forces, etc. which are used in functional material properties can be updated before running the new simulation.

See also [Writing to database files \[page 673\]](#).

## Examples

Restarting a transient analysis with multiple output times is illustrated in [Restarting Transient Analyses \[page 632\]](#).

The ***Opera-3d User Guide*** contains an Application note on "Coupled EM and Thermal Analysis in Opera-3d" and one on "Restarting Simulations", which illustrate the use of the **COPYCASE** command.

## The **DATALINE** Command

---

### Summary

Create line data for display as a line on a graph.

### Usage

The **DATALINE** command creates data and optionally displays it as a line on a graph. The line data is extracted from a buffer of data created by a field calculation command or read from a file by the **LOGGEDDATAFILE** command.

Once a line exists on a graph, its properties can be modified using the Graph tab property sheets or the **AXESVIEW** command.

### Command line parameters

| Command               | <b>DATALINE</b> |                                                                                                                          |
|-----------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------|
| Parameter             | Default         | Function                                                                                                                 |
| <b>OPTION</b>         |                 | Action of the command:                                                                                                   |
|                       |                 | <b>CREATE</b> Create new line data.                                                                                      |
|                       |                 | <b>DELETE</b> Delete the named line data.                                                                                |
| <b>BUFFER</b>         |                 | The name of the buffer providing the source data for the line.                                                           |
| <b>XCOMPONENT</b>     | <b>DISTANCE</b> | The name of a column or an expression in terms of one or more columns providing X coordinates of points on the graph.    |
| <b>YCOMPONENT</b>     |                 | The name of a column or an expression in terms of one or more columns providing Y coordinates of points on the graph.    |
| <b>ERRORCOMPONENT</b> |                 | The name of a column or an expression in terms of one or more columns providing error bar values at points on the graph. |
| <b>NAME</b>           |                 | The name of the line.                                                                                                    |
| <b>MAKEUNIQUE</b>     | <b>NO</b>       | Make the line <b>NAME</b> unique: <b>YES</b> or <b>NO</b> .                                                              |
| <b>DESCRIPTION</b>    |                 | Descriptive text which will be displayed with the line data.                                                             |
| <b>MINX</b>           | *               | The lower limit of the X axis. Use * to use the minimum value from the data.                                             |
| <b>MAXX</b>           | *               | The upper limit of the X axis. Use * to use the maximum value from the data.                                             |

|                      |                 |                                                                                    |
|----------------------|-----------------|------------------------------------------------------------------------------------|
| Command              | <b>DATALINE</b> |                                                                                    |
| Parameter            | Default         | Function                                                                           |
| <b>INTERPOLATION</b> | <b>LINEAR</b>   | The interpolation of the data:                                                     |
|                      |                 | <b>CUBIC</b>                                                                       |
|                      |                 | Use cubic-spline interpolation between data points.                                |
|                      |                 | <b>CUBICDERIVATIVES</b>                                                            |
|                      |                 | Use the first-derivative of the cubic-spline interpolations.                       |
| <b>GRAPH</b>         |                 | <b>DERIVATIVES</b>                                                                 |
|                      |                 | Use the first-derivative of the cubic-splines at the data points.                  |
| <b>GRAPH</b>         |                 | <b>LINEAR</b>                                                                      |
|                      |                 | Use the data points.                                                               |
| <b>GRAPH</b>         |                 | The name of the graph. If a name is supplied, the line will be added to the graph. |

## Creating a new line

New lines are created from the named **BUFFER** by evaluating the expressions for **XCOMPONENT** and **YCOMPONENT**. These components should be expressed in terms of the columns of the buffer. The column names include **DISTANCE** (the accumulated distance along the line) and for buffers created by field calculations along an arc or circle, **ANGLE**.

If the name of a **GRAPH** is given, the line will be displayed on the graph. Line **NAMEs** are automatically generated by the dialogs but can be changed before the command is issued if required. The **NAME** of the line will be shown in the graph legend. It is therefore a good idea to make the names of lines meaningful. To avoid confusion, do not use spaces or hyphens in a **NAME**. The legend text can be changed in the property sheet or by using [The AXESVIEW Command \[page 689\]](#).

Graph lines are initially displayed in a sequence of 16 colours. Colours and other line properties can be changed in the **Graph Objects** data sheet or by using [The AXESVIEW Command \[page 689\]](#).

## Interpolations

The values for **YCOMPONENT** can be interpolated in 4 ways:

- **LINEAR**: no additional values will be calculated. Straight lines will be drawn between data points.
- **CUBIC**: cubic spline interpolations will be used to calculate additional points between the buffer data points.
- **CUBICDERIVATIVES**: cubic spline interpolations will be matched to the buffer data points. The first derivative of these cubic splines will be displayed at the data points and at additional points between the data points.
- **DERIVATIVES**: cubic spline interpolations will be matched to the buffer data points. The first derivative of these cubic splines will be displayed at the data points with straight lines between them.

## System variables

Five system variables are created by the **DATALINE** command. The variable names start with the **NAME** of the line. If the **NAME** contains spaces these will be replaced by underscores in the names of the variables.

- *line\_INTEGRAL*: the integral under the curve.
- *line\_MAXIMUM*: the maximum Y or radial axis value.
- *line\_MINIMUM*: the minimum Y or radial axis value.
- *line\_XATMAXIMUM*: the X or azimuthal axis value corresponding to the maximum.
- *line\_XATMINIMUM*: the X or azimuthal axis value corresponding to the minimum.

## The DATAVECTOR Command

---

### Summary

Create or delete a vector.

### Usage

The **DATAVECTOR** command creates or deletes a vector. The data for the vector can be provided as cartesian values ( $\Delta x$ ,  $\Delta y$ ) or polar values ( $r$ ,  $\theta$ ).

A vector can be in more than one vector set but can not occur more than once in a vector set.

Vectors can be added to a vector set using [The DATAVECTORSET Command \[page 740\]](#) and displayed on a vector diagram using [The AXESVIEW Command \[page 689\]](#) which can also be used to adjust its attributes (colour, line type, etc.).

### Command line parameters

|               |                   |                                                                                  |  |
|---------------|-------------------|----------------------------------------------------------------------------------|--|
| Command       | <b>DATAVECTOR</b> |                                                                                  |  |
| Parameter     | Default           | Function                                                                         |  |
| <b>OPTION</b> |                   | Action of the command:                                                           |  |
|               |                   | <b>CREATE</b> Create a new vector.                                               |  |
|               |                   | <b>DELETE</b> Delete the named vector.                                           |  |
| <b>NAME</b>   |                   | The name of the vector.                                                          |  |
| <b>TYPE</b>   |                   | Type of data when creating a new vector:                                         |  |
|               |                   | <b>CARTESIAN</b> Values represent vector components, $\Delta x$ and $\Delta y$ . |  |
| <b>VALUE1</b> |                   | <b>POLAR</b> Values represent length and angle, $r$ and $\theta$ .               |  |
| <b>VALUE2</b> |                   | $\Delta y$ or $\theta$ value of vector.                                          |  |

### Creating a new vector

New vectors are created using **OPTION=CREATE**. The vector data, **VALUE1** and **VALUE2**, can be interpreted in two ways:

- **TYPE=CARTESIAN** uses **VALUE1** to specify the horizontal (X) component of the vector and **VALUE2** to specify the vertical (Y) component.
- **TYPE=POLAR** uses **VALUE1** to specify the length of the vector and **VALUE2** to specify its angle in degrees with respect to the horizontal axis.

The origin of the vector is specified when the vector is added to a vector set.

## System variables

Four system variables are created by the **DATAVECTOR** command. The variable names start with the **NAME** of the vector.

- ***vector\_X***: the horizontal (X) component of the vector.
- ***vector\_Y***: the vertical (Y) component of the vector.
- ***vector\_R***: the length of the vector.
- ***vector\_TH***: angle of the vector with respect to the horizontal axis.

## Deleting a vector

Vectors can be deleted by **NAME** using **OPTION=DELETE**.

## The DATAVECTORSET Command

---

### Summary

Create, delete or modify contents of a vector set.

### Usage

The **DATAVECTORSET** command controls vector sets. It can create or delete a vector set or change its contents by adding or removing vectors.

A vector can be in more than one vector set but can not occur more than once in a vector set.

A vector set can be displayed on a vector diagram using [The AXESVIEW Command \[page 689\]](#) which can also be used to adjust its attributes (scaling factor and rotation angle).

### Command line parameters

| Command       | <b>DATAVECTORSET</b>   |                                                                                           |
|---------------|------------------------|-------------------------------------------------------------------------------------------|
| Parameter     | Default                | Function                                                                                  |
| <b>OPTION</b> | Action of the command: | <b>ADD VECTOR</b> Add a vector to a vector set.                                           |
|               |                        | <b>CREATE</b> Create a new vector set.                                                    |
|               |                        | <b>DELETE</b> Delete the named vector set.                                                |
|               |                        | <b>REMOVEVECTOR</b> Remove a vector from the vector set.                                  |
| <b>NAME</b>   |                        | The name of the vector set.                                                               |
| <b>VECTOR</b> |                        | The name of a vector to be added or removed.                                              |
| <b>ORIGIN</b> |                        | The vector added will start at the end of the vector given by <b>ORIGIN</b> if specified. |

### Creating and deleting a vector set

Vector sets can be **CREATE**d and **DELETE**d by **NAME**. No other parameters are needed.

### Adding a vector to a vector set

A vector can be added to a **NAME**d vector set by specifying the **VECTOR** to be added and, optionally, the name of a second vector which provides the **ORIGIN**. The origin must be the name of a vector already in the vector set. The vector added will start at the end of the vector given as the origin. If the origin is not specified, the vector added will start at (0,0).

Vectors are created using [The DATAVECTOR Command \[page 738\]](#).

### Removing a vector from a vector set

A vector can be removed from a **NAMED** vector set by specifying the **VECTOR** to be removed. Vectors can only be removed if they have not been used as the origin of other vectors in the set.

### Interacting with vector sets

Vectors can also be added to vector sets using drag and drop in the **Graph Objects** data view. A vector dropped onto a vector already in a vector set will start at the end of that vector; vector dropped onto a vector set will start at (0,0).

Other commands are available in the **Graph Objects** context menus.

## The **END** Command

---

### Summary

End the Post-Processor

### Command line parameters

|               |            |
|---------------|------------|
| Command       | <b>END</b> |
| No Parameters |            |

### Notes

The **END** command stops the Opera-3d Post-Processor. All data files are closed.

## The ENERGY Command

---

### Summary

Calculate volume integrals to obtain stored energy, power loss and Lorentz forces.

### Toolbutton



### Command line parameters

| Command    | ENERGY      |                                                         |
|------------|-------------|---------------------------------------------------------|
| Parameter  | Default     | Function                                                |
| ACTION     | INTEGRATE   | Create list of volumes or integrate:                    |
|            |             | ADD Add volume(s) to list.                              |
|            |             | INTEGRATE Integrate.                                    |
|            |             | REMOVE Remove volume(s) from list.                      |
|            |             | RESET Empty list of volumes.                            |
|            |             | TOGGLEADD Toggles volume(s) in the list.                |
| LABEL      | ALL_VOLUMES | volumes to be added or removed from list:               |
|            |             | <i>material</i> Material name.                          |
|            |             | <i>label</i> Volume label.                              |
|            |             | ALL_VOLUMES All volumes.                                |
| TIMEOPTION | AVERAGE     | Steady-state ac options:                                |
|            |             | AMPLITUDE Calculate amplitude of oscillations.          |
|            |             | ANGLE or NO Calculate integrals at time of SET command. |
|            |             | AVERAGE or YES Calculate time-average values.           |
|            |             | PEAK Calculate peak values.                             |
|            |             | PHASE Calculate phase angles of oscillations.           |
| ADAPTIVE   | NO          | Adaptive integration switch.                            |
|            |             | NO Use 8 gauss-points in each element.                  |
|            |             | YES Use up to 216 gauss-points in each element.         |

## Notes

The **ENERGY** command integrates the stored energy, Lorentz force on induced currents, power loss and volume in the whole model space or in a labelled set of elements and updates the following system variables:

| Integrals                 |                                                                                                                                                                                                                                                                 |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Variable                  | Integrand                                                                                                                                                                                                                                                       |
| <b>ENERGY</b>             | magnetic fields, static and transient: nonlinear materials<br>$\int_{B_r}^B H \cdot dB$<br>magnetic fields: other cases<br>$\frac{1}{2}(B - B_r) \cdot H$<br>electrostatic fields:<br>$\frac{1}{2}D \cdot E$<br>stress analysis:<br>$\frac{1}{2}\sigma\epsilon$ |
| <b>COENERGY</b>           | magnetic fields, static and transient: nonlinear materials<br>$\int_{H_c}^H B \cdot dH$<br>magnetic fields: other cases<br>$\frac{1}{2}B \cdot (H - H_c)$                                                                                                       |
| <b>ELECENER</b>           | high-frequency:<br>$\frac{1}{2}D \cdot E$                                                                                                                                                                                                                       |
| <b>FX</b>                 | $J_v B_z - J_z B_v$ (eddy currents)                                                                                                                                                                                                                             |
| <b b="" fy<=""></b>       | $J_z B_x - J_x B_z$ (eddy currents)                                                                                                                                                                                                                             |
| <b b="" fz<=""></b>       | $J_x B_y - J_y B_x$ (eddy currents)                                                                                                                                                                                                                             |
| <b b="" power<=""></b>    | <b>J.E</b> (current flow and eddy currents)                                                                                                                                                                                                                     |
| <b b="" hyspower<=""></b> | Additional time-average power loss for hysteretic materials<br>(complex $\mu$ )                                                                                                                                                                                 |
| <b b="" volume<=""></b>   | 1                                                                                                                                                                                                                                                               |

The command operates as a 2-stage process:

1. form a list of volumes. Initially all volumes are in the list.
  - **ACTION=RESET** empties the list.
  - **ACTION=ADD** adds volumes to the list by **LABEL**.
  - **ACTION=REMOVE** removes volumes from the list by **LABEL**.
  - **ACTION=TOGGLEADD** toggles the membership of a volume in the list.

- Labels can be material names, volume labels including element and potential types and user labels or **ALL\_VOLUMES**.
2. integrate forces using **ACTION=INTEGRATE**.

The volume integral of other expressions of system variables can be calculated using [The VOLUME Command \[page 848\]](#).

## Integration Method

The basis for the integration is the finite element mesh. The integrals are performed in each element using first-order gaussian quadrature. However, in reduced potential volumes if the coil field is calculated by integration, the first-order quadrature might be insufficient to match the field variation in an element. Switching on adaptive integration (**+ADAPTIVE**) enables the program to use up to 9<sup>th</sup>-order gaussian quadrature in each element to increase the accuracy of the integrals. See [The SET Command \[page 798\]](#) for information about field calculation methods.

The **ENERGY** command includes rotated and reflected copies of the mesh specified by [The ACTIVATE Command \[page 679\]](#) in its integration.

Volume integrals can also be done using [The VOLUME Command \[page 848\]](#). However, if equivalent expressions are used for **COMPONENT** in the **VOLUME** command, the **ENERGY** command will give more accurate results because it derives its integrands from the solution potentials rather than averaged field values where possible.

## Steady-state AC Models

In steady-state alternating current models, the energy and power are functions of time with the form

$$E = A + B\cos(2\omega t) - C\sin(2\omega t) \quad (7.2)$$

The values *B* and *C* have little meaning on their own. The time-average value, *A* is the value commonly required. This and other values can be calculated directly using the **TIMEOPTION** parameter:

- **ANGLE** around ac cycle: *E*, with  $2\omega t$  given by **ACTIME** in [The SET Command \[page 798\]](#);
- **AMPLITUDE** of oscillations:  $(B^2+C^2)^{1/2}$ ;
- **time-AVERAGE**: *A*;
- **PEAK** value: if  $A > 0$ ,  $A+(B^2+C^2)^{1/2}$ ; if  $A < 0$ ,  $A-(B^2+C^2)^{1/2}$ ;
- **PHASE** angle of oscillations: **atan**( $-C/B$ ).

## Energy and Coenergy

Two forms of the energy integral are provided for nonlinear magnetostatics and transient simulations: energy, *E*, and coenergy, *E<sub>c</sub>*. (In linear systems, coenergy is equal to energy.) They should be used as follows in virtual work calculations to obtain force, *F*:

- if the currents are held constant during motion,

$$F = \frac{dE_c}{dx} \quad (7.3)$$

- if the flux is held constant during motion,

$$F = -\frac{dE}{dx} \quad (7.4)$$

## The **FIT** Command

---

### Summary

Fit Fourier series to field values previously calculated along a line or calculate field values on a sphere and fit Legendre polynomials.

### Toolbutton



### Command line parameters

|                  |                |                                                                                                                                                                                                                      |
|------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command          | <b>FIT</b>     |                                                                                                                                                                                                                      |
| Parameter        | Default        | Function                                                                                                                                                                                                             |
| <b>FILE</b>      | <b>TEMP</b>    | Name of the table file containing the field points and values. <b>FILE=TEMP</b> means use the current field buffer.                                                                                                  |
| <b>TYPE</b>      | <b>FOURIER</b> | Type of fitting.<br><br>FOURIER   FOURIER series fitting to <b>ARC</b> , <b>CIRCLE</b> or <b>LINE</b> values.<br>LEGENDRE   LEGENDRE polynomial fitting to values on a sphere around local coordinate system origin. |
| <b>COMPONENT</b> | <b>X</b>       | Field component for fitting.                                                                                                                                                                                         |
| <b>PRINT</b>     | <b>YES</b>     | Print options.<br><br>LOG   Output to log file <b>Post_n.lp</b> .<br>SCREEN   Output to screen.<br>YES   Output to both <b>SCREEN</b> and <b>LOG</b> file.<br>NO   No output.                                        |
| <b>RADIUS</b>    | 1              | Radius of sphere for <b>LEGENDRE</b> polynomial fitting.                                                                                                                                                             |
| <b>ORDER</b>     | 10             | Maximum degree of <b>LEGENDRE</b> polynomials.<br>Maximum order of <b>FOURIER</b> series.                                                                                                                            |
| <b>COLUMN</b>    | 4              | Column of table file.                                                                                                                                                                                                |
| <b>BUFFER</b>    | Fit            | Name of buffer to store field values calculated for <b>LEGENDRE</b> fitting.                                                                                                                                         |

## Notes

The **FIT** command is used to fit **FOURIER** series or **LEGENDRE** polynomials to field values.

### Fourier series

**FOURIER** series fitting uses field values previously calculated by

- [The ARC Command \[page 685\]](#),
- [The CIRCLE Command \[page 718\]](#) or
- [The LINE Command \[page 768\]](#).

The values used for fitting can come from 2 sources:

- a field buffer: **FILE=TEMP**, with the buffer specified by [The BUFFER Command \[page 708\]](#). Expressions for the **COMPONENT** can use as variables any of the system variables which are currently available (see [System Variables \[page 657\]](#)).
- a table file: If no file name extension is given the extension *table* is assumed. The structure of the files is given in [TABLE Files \[page 674\]](#). The component name and values are read from the specified **COLUMN**.

The values are assumed to span a complete cycle. The Fourier coefficients calculated by the **FIT** command are stored in system variables **A\_n** and **B\_n**. The number of coefficients printed and stored is set by **ORDER**, but this cannot be greater than  $n/2$ , where  $n$  is the number of field points.

### Legendre polynomials

**LEGENDRE** polynomial fitting first calculates values on the surface of a sphere which must be completely inside the model space. For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)). The radius of the sphere is given by the **RADIUS** parameter, and its origin and orientation are defined by the local coordinate system of [The SET Command \[page 798\]](#). The results are given for a spherical polar coordinate system whose axis is in the local Z direction. The maximum order of polynomial can be set with the **ORDER** parameter; it is limited to 30. However for **ORDER** between 9 and 14, coefficients up to order 14 will be given and for **ORDER** greater than 14 coefficients up to order 30 will be given. The values at the field points used by the fitting algorithm can be displayed by [The MAP Command \[page 775\]](#).

The Legendre polynomial coefficients calculated by the **FIT** command are stored in system variables, **A\_n\_m** and **B\_n\_m**.

## The FLUXLINKAGE Command

---

### Summary

Calculate total flux linked by conductors.

### Command line parameters

|             |                                        |                                                                                                                                                                                                                                                                       |           |                                  |       |                                        |       |                                     |
|-------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------------------------------|-------|----------------------------------------|-------|-------------------------------------|
| Command     | FLUXLINKAGE                            |                                                                                                                                                                                                                                                                       |           |                                  |       |                                        |       |                                     |
| Parameter   | Default                                | Function                                                                                                                                                                                                                                                              |           |                                  |       |                                        |       |                                     |
| SPACING     |                                        | Minimum distance between field points in integrals.                                                                                                                                                                                                                   |           |                                  |       |                                        |       |                                     |
| AXIALPOINTS |                                        | Number of integrals in axial direction.                                                                                                                                                                                                                               |           |                                  |       |                                        |       |                                     |
| TIMEOPTION  | AVERAGE                                | Steady-state ac options:<br><table> <tr> <td>AMPLITUDE</td> <td>Calculate amplitude of the flux.</td> </tr> <tr> <td>ANGLE</td> <td>Calculate flux at time of SET command.</td> </tr> <tr> <td>PHASE</td> <td>Calculate phase angles of the flux.</td> </tr> </table> | AMPLITUDE | Calculate amplitude of the flux. | ANGLE | Calculate flux at time of SET command. | PHASE | Calculate phase angles of the flux. |
| AMPLITUDE   | Calculate amplitude of the flux.       |                                                                                                                                                                                                                                                                       |           |                                  |       |                                        |       |                                     |
| ANGLE       | Calculate flux at time of SET command. |                                                                                                                                                                                                                                                                       |           |                                  |       |                                        |       |                                     |
| PHASE       | Calculate phase angles of the flux.    |                                                                                                                                                                                                                                                                       |           |                                  |       |                                        |       |                                     |

### Notes

The FLUXLINKAGE command calculates the total flux linked by solenoid, racetrack and bedstead conductors. The command operates as part of a 2-stage process:

1. form a list of conductors using [The CONDUCTOR Command \[page 727\]](#). An empty list implies all possible conductors.
2. integrate normal flux density using the FLUXLINKAGE command.

The value of flux linked is saved in the system variable, FLUX.

### Integration process

The calculation of the flux linking a conductor is done by summing integrals over a number of patches which together capture all the flux linking the conductor. The patches can be repeated and averaged at a number of axial positions. Parameters can be adjusted to control the accuracy of the integration:

- SPACING: specifies the minimum distance between field points in each patch.
- AXIALPOINTS: specifies the number of axial positions of the patches.

### Steady-state ac

In steady-state ac models, values can be calculated in 3 ways using the TIMEOPTION parameter:

- the flux at the **ANGLE** around ac cycle given by **ACTIME** in **The SET Command [page 798]**;
- **AMPLITUDE** of the flux;
- **PHASE** angle the flux.

## The **GRAPH** Command

---

### Summary

Plot graphs from data in text files on the **2d graphics** tab.

### Command line parameters

|                 |                  |                                                                                                                                                                                                                              |
|-----------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command         | <b>GRAPH</b>     |                                                                                                                                                                                                                              |
| Parameter       | Default          | Function                                                                                                                                                                                                                     |
| <b>FILE</b>     | <i>none</i>      | File name                                                                                                                                                                                                                    |
| <b>XVALUE</b>   | <b>COL1</b>      | Expression for x-axis values.                                                                                                                                                                                                |
| <b>XLABEL</b>   | <b>X</b>         | Label for x-axis.                                                                                                                                                                                                            |
| <b>XMINIMUM</b> | *                | X-axis minimum value. Use * for automatic scaling.                                                                                                                                                                           |
| <b>XMAXIMUM</b> | *                | X-axis maximum value. Use * for automatic scaling.                                                                                                                                                                           |
| <b>YVALUE</b>   | <b>COL2</b>      | Expression for y-axis values.                                                                                                                                                                                                |
| <b>YLABEL</b>   | <b>Y</b>         | Label for y-axis.                                                                                                                                                                                                            |
| <b>YMINIMUM</b> | *                | Y-axis minimum value. Use * for automatic scaling.                                                                                                                                                                           |
| <b>YMAXIMUM</b> | *                | Y-axis maximum value. Use * for automatic scaling.                                                                                                                                                                           |
| <b>TITLE</b>    | <i>none</i>      | Additional title for line.                                                                                                                                                                                                   |
| <b>STYLE</b>    | <b>AUTOMATIC</b> | Line style:<br>AUTOMATIC      Program chooses a different style for each graph drawn on the same axes.<br>NOLINE          No line<br>0                Solid line.<br>>0               Broken line.                           |
| <b>COLOUR</b>   | <b>AUTOMATIC</b> | Line colour:<br>AUTOMATIC      Program chooses a different colour for each graph drawn on the same axes.<br>name             Colour name: TEXT, RED, GREEN, BLUE, YELLOW, CYAN or MAGENTA<br>>0               Colour number. |

| Command       | <b>GRAPH</b>  |                                                                                                                                                                                                                     |
|---------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter     | Default       | Function                                                                                                                                                                                                            |
| <b>SYMBOL</b> | <b>NONE</b>   | Symbols at data points:                                                                                                                                                                                             |
|               |               | <b>AUTOMATIC</b> Program chooses a different symbol for each graph drawn on the same axes.                                                                                                                          |
|               |               | <b>NONE</b> or 0 No symbol.                                                                                                                                                                                         |
|               |               | <i>name</i> Symbol name:<br>Triangle pointing <b>UP</b> ,<br>Triangle pointing <b>DOWN</b> ,<br><b>PLUS</b> sign,<br>Diagonal <b>CROSS</b> ,<br><b>SQUARE</b> , <b>DIAMOND</b> ,<br><b>OCTOGON</b> or <b>STAR</b> . |
|               |               | >0 Symbol number.                                                                                                                                                                                                   |
| <b>ERASE</b>  | <b>YES</b>    | Old graph erasure switch:                                                                                                                                                                                           |
|               |               | <b>NO</b> New line drawn on existing axes.                                                                                                                                                                          |
|               |               | <b>YES</b> Graphics window cleared and new axes drawn.                                                                                                                                                              |
| <b>OPTION</b> | <b>VALUES</b> | Display option:                                                                                                                                                                                                     |
|               |               | <b>CUBICDERIVATIVES</b> Display the first derivatives of the data (interpolated).                                                                                                                                   |
|               |               | <b>DERIVATIVES</b> Display the first derivatives of the data (straight lines between data points).                                                                                                                  |
|               |               | <b>INTERPOLATIONS</b> Display cubic-spline interpolations between data values.                                                                                                                                      |
|               |               | <b>VALUES</b> Display straight lines between data values.                                                                                                                                                           |

## Notes

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

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

The data format expected by the **GRAPH** command is flexible. The file should consist of up to 99 columns of numbers, with any number of values in each column. However, text can be embedded within the numerical data. Each line of the file is parsed into a maximum of 99 fields separated by

spaces or commas. The fields are identified as character data or numerical data. If there is numerical data on a line it is kept; character data is ignored. The number of columns is given by the number of numerical data items on the first line which contains any numbers. If subsequent lines contain fewer numerical values, the number of columns is reduced.

Comment lines (starting with /) are ignored.

In the Opera-3d Post-Processor the **GRAPH** command can recognise [TABLE Files \[page 674\]](#) by the filename extension **table**. When reading such files, the header records (types 1, 2 and 3) are ignored and only the data records (type 4) are processed for the graph.

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

The data evaluated from the expressions **XVALUE** and **YVALUE** can be displayed in 3 ways:

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

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

The **MINIMUM** and **MAXIMUM** values along the graph are assigned to system variables. The positions at which the extreme values occur are saved in system variables **XATMINIMUM** and **XATMAXIMUM**.

## The **GRID** Command

---

### Summary

Calculate field values on a grid of points and write a table file.

### Command line parameters

| Command              | <b>GRID</b>  |                                                           |
|----------------------|--------------|-----------------------------------------------------------|
| Parameter            | Default      | Function                                                  |
| X0                   | 0            | X-coordinate at corner of grid.                           |
| Y0                   | 0            | Y-coordinate at corner of grid.                           |
| Z0                   | 0            | Z-coordinate at corner of grid.                           |
| DXG                  | 1            | X increment between grid points                           |
| DYG                  | 1            | Y increment between grid points                           |
| DZG                  | 1            | Z increment between grid points                           |
| NXG                  | 1            | Number of points in X direction                           |
| NYG                  | 1            | Number of points in Y direction                           |
| NZG                  | 1            | Number of points in Z direction                           |
| FILE                 | <i>none</i>  | Name of file to store values.                             |
| BINARY               | NO           | Binary file switch: YES or NO.<br>NO implies a text file. |
| F1                   | X            | Expression for values in column 1                         |
| F2                   | Y            | Expression for values in column 2                         |
| F3                   | Z            | Expression for values in column 3                         |
| F4, ..., F12         | <i>none</i>  | Expression for values in column 4 to 12                   |
| NAME1, ..., NAME12   | F1, ..., F12 | Names for columns                                         |
| UNIT1G               | LENGU        | Unit expression for column 1                              |
| UNIT2G               | LENGU        | Unit expression for column 2                              |
| UNIT3G               | LENGU        | Unit expression for column 3                              |
| UNIT4G, ..., UNIT12G | 1.0          | Unit expressions for columns 4 to 12                      |
| FORMAT               | 2            | Table file format                                         |

## Notes

The **GRID** command is provided to facilitate an interface to other post-processing programs. It calculates field values over a 1, 2 or 3 dimensional space defined by one corner (**X0**, **Y0**, **Z0**), the increments in each direction (**DXG**, **DYG**, **DZG**) and the number of points in each direction (**NXG**, **NYG**, **NZG**). The coordinates of the points are defined with respect to the local coordinate system of [The SET Command \[page 798\]](#). The field points and components are output to a file in the Global Coordinate System.

Expressions for values in up to 12 columns can use any of the system variables which are currently available (see [System Variables \[page 657\]](#)).

## Text files

In text files (**-BINARY**) table file format is used (see [TABLE Files \[page 674\]](#)) using the **FORMAT** given. The column names and unit expressions can be set using the **NAME<sub>n</sub>** and **UNIT<sub>n</sub>G** parameters.

**FORMAT 1** files are written in internal units and **FORMAT 2** file in the user's units. The unit expressions should be in terms of the unit conversion factor system variables, e.g. **FLUXU/LENGU** (see [The UNITS Command \[page 835\]](#)). For **FORMAT 2** files, the program will convert the expression to the user's current choice of units, e.g. **TESLA/METRE**.

If no file name extension is given, the extension **table** is supplied for text files.

## Binary files

Binary files (**+BINARY**) are written using grid file format (see [GRID files \[page 673\]](#)) with the data in the user's units.

If no file name extension is given, the extension **grid** is supplied for binary files.

## The **GUIOPTIONS** Command

---

### Summary

Set console visibility and other window options.

### Command line parameters

|                      |                                                                       |                    |
|----------------------|-----------------------------------------------------------------------|--------------------|
| Command              | <b>GUIOPTIONS</b>                                                     |                    |
| Parameter            | Function                                                              |                    |
| <b>OPTION</b>        | Action of command                                                     |                    |
|                      | <b>LOAD</b>                                                           | Used by the GUI.   |
|                      | <b>SET</b>                                                            | Update the values. |
| <b>CONSOLEVIEW</b>   | Console visible: <b>YES</b> or <b>NO</b>                              |                    |
| <b>CONSOLEDOCK</b>   | Console docked: <b>YES</b> or <b>NO</b>                               |                    |
| <b>CONSOLEBUFFER</b> | Number of lines saved in the console.                                 |                    |
| <b>EXTRASVIEW</b>    | View Units, Model Data and Local Coordinates: <b>YES</b> or <b>NO</b> |                    |
| <b>PRINT3DCARD</b>   | Print from graphics card: <b>YES</b> or <b>NO</b>                     |                    |
| <b>BACKINGSTORE</b>  | Graphics backing store: <b>YES</b> or <b>NO</b>                       |                    |
| <b>WINDOWWIDTH</b>   | Width of window in pixels.                                            |                    |
| <b>WINDOWHEIGHT</b>  | Height of window in pixels.                                           |                    |

### Notes

The **GUIOPTIONS** command can be used to change the options which affect the appearance and behaviour of the GUI window. Default settings are loaded from and changes are saved in the registry<sup>1</sup>. Saving the window size and position is optional.

- Console - the command input and text output console can be visible or hidden (**CONSOLEVIEW**) and when it is visible it can be docked (at the bottom of the main window) or undocked (a separate window) (**CONSOLEDOCK**). The scrolling buffer is limited to a number of lines (**CONSOLEBUFFER**).
- Information box, containing UNITS, MODEL DATA and Local Coordinates can be visible or hidden (**EXTRASVIEW**).

---

<sup>1</sup>On Linux systems, registry values are saved in files in the directory  
*~/config/VectorFields*.

- Printing - the option to print the image stored in the graphics card is preferable with most graphics card (**PRINT3DCARD**) but can be changed if necessary. The alternative copies the image from the screen display.
- Backing store must be enabled to allow the program to re-display 2D pictures if the window size is changed. However, for very complex models, e.g. graphs of large numbers of trajectories, backing store can make the graphics run more slowly.
- The **WINDOWWIDTH** and **WINDOWHEIGHT** can be set to ensure a consistent size when creating pictures (see [The PICTURE Command \[page 781\]](#) and [The PRINT Command \[page 790\]](#)). These parameters are only available from the command line.

The console window, showing the commands issued to the software, may be displayed by right clicking anywhere in the tab area. On the resulting context sensitive menu, select **Show/Hide** items and click on the tick box for **Console**.

## The IDEAS Command

---

### Summary

Create or append to an I-deas Universal File.

### Command line parameters

| Command           | IDEAS         |                                                                                                        |
|-------------------|---------------|--------------------------------------------------------------------------------------------------------|
| Parameter         | Default       | Function                                                                                               |
| <b>FILE</b>       | <b>none</b>   | Name of Universal File.                                                                                |
| <b>MODE</b>       | <b>APPEND</b> | Mode of operation:                                                                                     |
|                   |               | <b>APPEND</b> Append results to an existing file.                                                      |
|                   |               | <b>CREATE</b> Create a new file with nodes, elements, material names, boundary conditions and results. |
|                   |               | <b>DATAONLY</b> Create a new file with node, elements, material names and boundary conditions.         |
| <b>TYPE</b>       | <b>REAL</b>   | Result type:                                                                                           |
|                   |               | <b>COMPLEX</b> Results from steady-state ac analysis                                                   |
|                   |               | <b>REAL</b> Results from statics or transient analysis.                                                |
| <b>BASIS</b>      | <b>NODE</b>   | Basis of results:                                                                                      |
|                   |               | <b>ELEMENT</b> Values at every node of every element                                                   |
|                   |               | <b>NODE</b> Values at every node                                                                       |
| <b>FIELD</b>      | <b>SCALAR</b> | Field type:                                                                                            |
|                   |               | <b>SCALAR</b> Scalar field                                                                             |
|                   |               | <b>VECTOR</b> Vector field                                                                             |
| <b>COMPONENT</b>  | <b>X</b>      | Expression for real part of scalar field values.                                                       |
| <b>ICOMPONENT</b> | <b>Y</b>      | Expression for imaginary part of scalar field values ( <b>TYPE=COMPLEX</b> )                           |
| <b>VX</b>         | <b>X</b>      | Expression of real part of x-component of vector field.                                                |
| <b>IVX</b>        | <b>X</b>      | Expression for imaginary part of x-component of vector field                                           |
| <b>VY</b>         | <b>Y</b>      | Expression of real part of y-component of vector field.                                                |
| <b>IVY</b>        | <b>Y</b>      | Expression for imaginary part of y-component of vector field                                           |

|           |              |                                                              |
|-----------|--------------|--------------------------------------------------------------|
| Command   | <b>IDEAS</b> |                                                              |
| Parameter | Default      | Function                                                     |
| VZ        | Z            | Expression of real part of z-component of vector field.      |
| IVZ       | Z            | Expression for imaginary part of z-component of vector field |
| LABEL     | 1            | Analysis dataset label                                       |
| NAME      | VF           | Analysis dataset name                                        |

## Notes

The **IDEAS** command creates or appends to an I-deas Universal file. When the file is created (**MODE-E=CREATE**) the finite element data is written first, followed by a results dataset if available. In **APPEND** mode, only the results dataset is written. In **DATAONLY** mode a file is created with only data and no results datasets.

Only Hexahedral and Tetrahedral elements can be included. Other Mosaic mesh elements (Prisms and Pyramids) cannot be included and an I-deas Universal file will not be created if the model contains them.

## Finite Element Data

The finite element data written to a Universal File consists of the following datasets:

| Number | Dataset Name               |
|--------|----------------------------|
| 151    | Header                     |
| 164    | Units                      |
| 1700   | Material Database Header   |
| 1703   | Material Database Property |
| 1705   | Material Database Variable |
| 1710   | Material Database Material |
| 789    | Physical Property Table    |
| 2411   | Nodes - Double Precision   |
| 2412   | Elements                   |
| 792    | Boundary Conditions        |

Dataset 164 specifies the length unit for the data. The 'Units description' includes the units of flux density, current density and electric field strength.

Dataset 1710 is repeated for each ‘material’, where a material is formed for each combination of material name, potential type and element type which exists in the model, for example a typical Universal File Material Name would be `IRON_TOTAL_QUADRATIC`. The ‘Material Number’ and ‘Material Name’ are the only significant data in the dataset 1710.

Dataset 792 is repeated for each volume face of the original model which has a boundary condition.

## Results Data

Results data are written using dataset 2414 ‘Analysis Data’. The simplest form is a real scalar field at nodes (`TYPE=REAL, BASIS=NODE, FIELD=SCALAR`) which can be specified using one **COMPONENT** expression. At the other extreme is a complex vector field on elements, allowing discontinuity between each element and its neighbours (`TYPE=COMPLEX, BASIS=ELEMENT, FIELD=VECTOR`). This would be specified using 6 expressions, for the real and imaginary parts of the X, Y and Z components.

All other combinations of **TYPE**, **BASIS** and **FIELD** are also allowed, for example, to store the electric field strength in an electrostatic example, not allowing for any discontinuities:

```
type=real,basi=node,fiel=vect,vx=ex,vy=ey,vz=ez
```

## Limitations for Quadratic Elements

If the finite element model in the Opera-3d database was created from a Universal File with the Pre-Processor using [The IDEAS Command \[page 452\]](#), and it also contains quadratic elements, the results should not be **APPENDED** to the original Universal File, since the positions and ordering of the quadratic nodes will not match. A new Universal File should be **CREATED** to contain the finite element model and results.

## Integration Commands

In many situations, the result which is required is calculated by integrating field quantities along a line, over a surface or in a volume. Opera-3d Post-Processor has several commands which can perform integration, some for specific purposes and others which allow more general facilities.

### Line Integrals

Line integrals are calculated by [The PLOT Command \[page 783\]](#).

- The line should be defined and field quantities evaluated first by [The ARC Command \[page 685\]](#), [The CIRCLE Command \[page 718\]](#) or [The LINE Command \[page 768\]](#).
- [The PLOT Command \[page 783\]](#) then draws a graph of any field component expression and also calculates the line integral.

[The GRAPH Command \[page 751\]](#) can also be used to calculate line integrals. It uses values read from a data file.

### Surface Integrals

There are two types of surface integral:

- *Over a surface of the model:* currently selected by [The SELECT Command \[page 793\]](#). It can be the surface of a material or potential type, a boundary condition surface, etc.
  - [The INTEGRATE Command \[page 763\]](#) calculates force and torque by integration virtual work or Maxwell Stress tensor over the surface. The surface must completely enclose the bodies experiencing the force.
  - [The SURFACE Command \[page 809\]](#) integrates any field component expression over the surface.
- *Over a general surface patch:*
  - The surface should be defined and the field quantities evaluated first using [The CARTESIAN Command \[page 710\]](#), [The POLAR Command \[page 788\]](#) or [The SPHERICAL Command \[page 807\]](#).
  - [The MAP Command \[page 775\]](#) displays distribution of the field component expression and also calculates the area integral.

### Volume Integrals

Volume integrals can be performed over volumes selected by labels (including material names, potential names, element types, **NOTAIR** and **ALL\_VOLUMES**). The basis for the integration is the finite element mesh; higher order integration formulae are available for functions which vary rapidly over the elements.

- The ENERGY Command [page 743] has built-in integrands to calculate energy, power-loss and Lorentz forces in the complete model including replications specified on the ACTIVATE command.
- The VOLUME Command [page 848] can integrate any field component expression. Replications of the model are omitted from the integration.

The BODY Command [page 704] integrates Lorentz forces over the volume of the conductors.

## The INTEGRATE Command

### Summary

Integrate forces on bodies enclosed by the selected surfaces.

### Toolbutton



### Command line parameters

|            |               |                                                                                                                                                                                                                                                                        |
|------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command    | INTEGRATE     |                                                                                                                                                                                                                                                                        |
| Parameter  | Default       | Function                                                                                                                                                                                                                                                               |
| X0         | 0             | X coordinate of point of action for torque.                                                                                                                                                                                                                            |
| Y0         | 0             | Y coordinate of point of action for torque.                                                                                                                                                                                                                            |
| Z0         | 0             | Z coordinate of point of action for torque.                                                                                                                                                                                                                            |
| TIMEOPTION | AVERAGE       | Steady-state ac options:<br>AMPLITUDE Calculate amplitude of oscillations.<br>ANGLE or NO Calculate integrals at time of SET command.<br>AVERAGE or YES Calculate time-average values.<br>PEAK Calculate peak values.<br>PHASE Calculate phase angles of oscillations. |
| METHOD     | MAXWELLSTRESS | Method switch.<br>MAXWELLSTRESS Integrate Maxwell Stress tensor over the surface.<br>VIRTUALWORK Integrate energy in elements touching the surface.                                                                                                                    |

### Notes

The INTEGRATE command integrates over the surface selected by [The SELECT Command \[page 793\]](#) to calculate the force and torque on the bodies enclosed by the surface. The parameters (X0, Y0, Z0) define the global coordinates of the point of action of the torque. The results are also stored in the system variables FX, FY and FZ and TORQX, TORQY and TORQZ.

Two **METHODs** are available:

- **VIRTUALWORK**: integrates the rates of change in energy in the elements surrounding the selected surface due to small changes in the positions of the nodes on the surface. This is usually the more accurate method, but is limited by the requirement that the elements surrounding the surface must represent linear isotropic materials and use **TOTAL** potential.

The integration does not include rotated or reflected images of the model unless these have been included by the **SELECT** command. There is one exception to this rule: if the elements used for integration are all in the gap region of a Motional EM simulation, all rotational images are included.

- **MAXWELLSTRESS**: integrates the Maxwell stress tensor over the facets in the selected surface. This can be used on any surface, even on surfaces which cut through materials.

Better results can often be obtained for both **METHODs** if a layer of air is added to the selected surfaces before **INTEGRATE** to move the integration surface away from the material corners where the field may be singular (See "The **SELECT** Command" on page 793.). In any case, on a material surface, the **INTEGRATE** command uses the values from the air side of the interface.

If the force required is acting on the coils, it is usually more accurate to select the surface of the reduced scalar potential volume containing the coils and **INTEGRATE** over that or use **The BODY Command** [page 704].

## Cancellation

The program also calculates the "cancellation". Forces are often the differences between opposing forces. If the opposing forces are almost equal, the cancellation level will approach 100%. If almost all the force is in one direction, the cancellation will be close to zero. The formula used to calculate the cancellation,  $C$  is:

$$C = \left( 1 - \frac{|\sum f|}{\sum |f|} \right) \times 100 \quad (7.5)$$

where  $f$  is the contribution to the force or torque from one element (virtual work) or facet (Maxwell stress).

## Other commands

The **MAP Command** [page 775] can also be used to integrate Maxwell stress over more general surfaces. The system variables which can be used as integrands are described in **Maxwell Stress integrands** [page 665].

The **SURFACE Command** [page 809] can be used to integrate other field quantities over the selected surface.

## Steady-state ac

In steady-state alternating current models, the force is a function of time with the form

$$F = A + B\cos(2\omega t) - C\sin(2\omega t) \quad (7.6)$$

The values  $B$  and  $C$  have little meaning on their own. The time-average value,  $A$  is the value commonly required. This and other values can be calculated directly using the **TIMEOPTION** parameter:

- **ANGLE** around ac cycle:  $F$ , with  $2\omega t$  given by **ACTIME** in **The SET Command [page 798]**;
- **AMPLITUDE** of oscillations:  $(B^2+C^2)^{1/2}$ ;
- time-**AVERAGE**:  $A$ ;
- **PEAK** value: if  $A>0$ ,  $A+(B^2+C^2)^{1/2}$ ; if  $A<0$ ,  $A-(B^2+C^2)^{1/2}$ ;
- **PHASE** angle of oscillations: **atan**( $-C/B$ ).

## The ISOSURFACE Command

---

### Summary

Display the surface where the field is constant.

### Toolbutton



### Command line parameters

| Command   | ISOSURFACE |                                                                                    |
|-----------|------------|------------------------------------------------------------------------------------|
| Parameter | Default    | Function                                                                           |
| COMPONENT | X          | Field component for the surface.                                                   |
| VALUE     | 0          | Value of field component on the surface.                                           |
| COLOUR    | AUTOMATIC  | Line colour for display of the surface:                                            |
|           |            | AUTOMATIC Program chooses a different colour for each surface drawn with ERASE=NO. |
|           |            | name Colour name: TEXT, RED, GREEN, BLUE, YELLOW, CYAN or MAGENTA                  |
|           |            | >0 Colour number.                                                                  |
| ERASE     | YES        | Multiple surfaces option:                                                          |
|           |            | CLEAR Clear existing surfaces and do not display a new one.                        |
|           |            | NO Add to existing surfaces.                                                       |
|           |            | YES Replace any existing surface.                                                  |

### Notes

The ISOSURFACE command displays a surface where a field component has a constant value. The surfaces are overlaid on the three dimensional pictures of the geometry created by [The THREED Command \[page 823\]](#). Any number of surfaces can be added to the picture (ERASE=NO) or a surface can replace any previously displayed (ERASE=YES). The surfaces and the list of values can be removed from the display using ERASE=CLEAR or temporarily hidden using [The WINDOW Command \[page 851\]](#).

## Field values

The calculation of the surface is a two-stage process.

1. The field **COMPONENT** is calculated at all the nodes in the model. Expressions for the **COMPONENT** can use as variables any of the system variables which are currently available (see [System Variables \[page 657\]](#)). The field calculation method is restricted to nodal interpolation; coil files may be calculated by integration (see [The SET Command \[page 798\]](#)).
2. The nodal values are interpolated within each element to find and display the surface with the specified **VALUE**.

## The **LINE** Command

### Summary

Calculate fields along a straight line.

### Toolbutton



### Command line parameters

| Command   | <b>LINE</b> |                                                                            |
|-----------|-------------|----------------------------------------------------------------------------|
| Parameter | Default     | Function                                                                   |
| X1        | 0           | X-coordinate of the first point on the line.                               |
| Y1        | 0           | Y-coordinate of the first point on the line.                               |
| Z1        | 0           | Z-coordinate of the first point on the line.                               |
| X2        | <i>none</i> | X-coordinate of the last point on the line.                                |
| Y2        | <i>none</i> | Y-coordinate of the last point on the line.                                |
| Z2        | <i>none</i> | Z-coordinate of the last point on the line.                                |
| NP        | 100         | Number of steps between the first and last points, i.e <b>NP+1</b> points. |
| BUFFER    | Line        | Name of buffer to store field values.                                      |

### Notes

The **LINE** command evaluates field quantities along a straight line

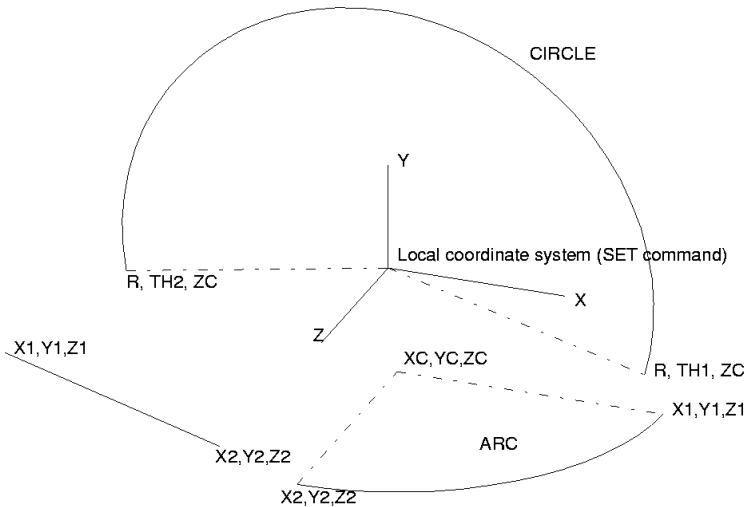


Figure 7.2 The **ARC**, **CIRCLE** and **LINE**

for use by

- [The FIT Command \[page 747\]](#)
- [The PLOT Command \[page 783\].](#)

For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)).

The line is specified by its end points ( $X1, Y1, Z1$  and  $X2, Y2, Z2$ ) (see Figure 7.2, above), the positions of which are affected by any local coordinate system defined with [The SET Command \[page 798\]](#). The vector field quantities are evaluated with respect to the Global Coordinate System for the active file. The field quantities are evaluated at  $NP+1$  points along the line and are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## Graphs of field values

The **Fields on a Straight Line** dialog runs 3 commands:

- [The AXESVIEW Command \[page 689\]](#) (optional) to create a new graph.
- The **LINE** command.
- [The DATALINE Command \[page 735\]](#) to add the line of calculated values to the graph.

## The LOAD Command

---

### Summary

Nominate an active database for display of geometry and field calculations.

### Toolbutton



### Command line parameters

|             |             |                                            |
|-------------|-------------|--------------------------------------------|
| Command     | <b>LOAD</b> |                                            |
| Parameter   | Default     | Function                                   |
| <b>FILE</b> | <i>none</i> | Name or number of an active database file. |

### Notes

The **LOAD** command makes an active database file resident. Files are activated by [The ACTIVATE Command \[page 679\]](#). The **LOAD** command makes the data in an active file available for the display and field evaluation commands. The details of the loaded or resident file are shown in the Information Box at the right-hand side of the display.

The parameter **FILE** sets the file name. If no file name extension is given an extension of *op3* is assumed. **FILE** can also specify the number of the active file. This can be useful if the same file has been activated more than once with different parameters on the **ACTIVATE** command. The active files can be listed with [The SHOW Command \[page 804\]](#). If no file name or number is given the most recently activated file is assumed.

**LOAD** sets the string variable **TITLE** to the first line of the title from the database and the following system variables, depending on the simulation type:

- **FREQ**: the frequency of a harmonic simulation.
- **TTIME**: the time of a transient simulation.

There are two special forms of the **LOAD** command:

- **Close loaded database** (**LOAD FILE=0**) can be used to close the currently loaded file so that it can be accessed by other programs. If the simulation was unsolved, the file is also removed from the list of active files.
- **Deactivate loaded database** (**LOAD FILE=-1**) can be used to close the currently loaded file and remove it from the list of active files.

The menu system automatically executes the **LOAD** command when a file is opened with the Open (Activate+Load) option.

**LOAD** saves the name of the file in string variable **LOADFILENAME**.

**LOAD** saves the symmetry of the model in system variables **ROTATIONS**, **REFXY**, **REFYZ** and **REFZX** (see [The SYMMETRY Command \[page 811\]](#)). This uses the values in the database or supplied using the parameters of [The ACTIVATE Command \[page 679\]](#).

## The LOGGEDDATAFILE Command

---

### Summary

Import buffer data from or export buffer data to a file.  
Create a buffer one row at a time.

### Toolbuttons



**Plot graph**



**Create buffer from text file**

### Command line parameters

|                   |                                                            |                                                               |
|-------------------|------------------------------------------------------------|---------------------------------------------------------------|
| Command           | <b>LOGGEDDATAFILE</b>                                      |                                                               |
| Parameter         | Default                                                    | Function                                                      |
| <b>FILENAME</b>   | Name of the file to be read or written.                    |                                                               |
| <b>BUFFER</b>     | The name of the buffer to be created or saved.             |                                                               |
| <b>MAKEUNIQUE</b> | Make the <b>BUFFER</b> name unique: YES or NO.             |                                                               |
| <b>OPTION</b>     | <b>IMPORT</b>                                              |                                                               |
|                   | <b>APPEND</b>                                              | Append data in <b>DATALIST</b> to <b>BUFFER</b> .             |
|                   | <b>CREATE</b>                                              | Create <b>BUFFER</b> with column names in <b>HEADERLIST</b> . |
|                   | <b>EXPORT</b>                                              | Export <b>BUFFER</b> to a <b>csv</b> file.                    |
|                   | <b>IMPORT</b>                                              | Import data from a file to <b>BUFFER</b> .                    |
| <b>HEADERLIST</b> | A list of column names used to create a buffer.            |                                                               |
| <b>DATALIST</b>   | A list of values forming a row to be appended to a buffer. |                                                               |

### Notes

The **LOGGEDDATAFILE** command is used to create or save graphing buffer data. Buffers are 2-dimensional arrays of data: each column represents a particular quantity (e.g. a field component or a circuit current); each row represents values of the quantities at a particular point in space or time. Columns can optionally have names.

There are 2 ways of creating a buffer:

- **OPTION=IMPORT** to read values from a text file;
- **OPTION=CREATE** to specify column names followed by 2 or more commands with **OPTION=N=APPEND** to add rows of data.

## Importing from a file

Text files read by **LOGGEDDATAFILE OPTION=IMPORT** should contain 2 types of line:

1. Column header lines are optional but if they exist they should be at the top of the file. They must start with "/" which should be followed by one name for each column. All such lines must contain the same number of names. The names in second or subsequent lines can be used as alternative column names.
2. Data lines each contain the same number of numerical values, one value for each column.

The data items in a line can be separated by spaces or commas.

Table files, i.e. files with names **\*.table**, can also be read by the **LOGGEDDATAFILE** command. The format of table files is described in [TABLE Files \[page 674\]](#).

The program also defines column names **COL1**, **COL2**, **COL3**, etc., whether or not column names are defined by the file.

## Exporting to a file

**LOGGEDDATAFILE OPTION=EXPORT** creates files using the commonly used **csv** (comma-separated values) format. The first line contains the column names, if they exist or names **COL1**, **COL2**, **COL3**, etc. if they do not.

## Creating a buffer row-by-row

**LOGGEDDATAFILE OPTION=CREATE** creates a new buffer and defines the column names. The parameter, **HEADERLIST**, is a list parameter so should appear several times on the command line in order to define each column.

For example, to create a buffer called **v\_i** with columns **TIME**, **v** and **i**:

```
LOGGEDDATAFILE OPTION=CREATE BUFFER=v_i,
HEADERLIST=TIME HEADERLIST+=v HEADERLIST+=i
```

Any number of rows can be appended to a buffer using the list parameter, **DATALIST**, for example:

```
LOGGEDDATAFILE OPTION=APPEND BUFFER=v_i,
DATALIST=0 DATALIST+=1 DATALIST+=0
LOGGEDDATAFILE OPTION=APPEND BUFFER=v_i,
DATALIST=0.1 DATALIST+=0.5 DATALIST+=1
LOGGEDDATAFILE OPTION=APPEND BUFFER=v_i,
DATALIST=0.2 DATALIST+=0.3 DATALIST+=1.5
```

For more information on list parameters, see [List Parameter Values \[page 36\]](#).

The program also defines column names **COL1**, **COL2**, **COL3**, etc., whether or not column names are defined by the file.

## Graph plotting

After importing or creating a buffer from a file or row-by-row, the data can be graphed using the following commands.

- [The AXESVIEW Command \[page 689\]](#) (optional) to create a new graph.
- [The DATALINE Command \[page 735\]](#) to add a line to the graph.

The **Plot Graph** dialog with **Text file** selected as input runs 3 commands: **AXESVIEW**, **LOGGEDDATAFILE** and **DATALINE**.

## The **MAP** Command

---

### Summary

Display contours of field vectors on 2-dimensional surface patches.

### Toolbutton



### Command line parameters

|                  |              |                                                                                                          |
|------------------|--------------|----------------------------------------------------------------------------------------------------------|
| Command          | <b>MAP</b>   |                                                                                                          |
| Parameter        | Default      | Function                                                                                                 |
| <b>FILE</b>      | <b>TEMP</b>  | Name of table file containing field points and values. <b>FILE=TEMP</b> means do not use a file.         |
| <b>CONTOUR</b>   | <b>ZONES</b> | Display <b>COMPONENT</b> as a contour map.                                                               |
|                  |              | <b>HISTOGRAM</b> Contours displayed as a histogram.                                                      |
|                  |              | <b>LINES</b> Contour lines.                                                                              |
|                  |              | <b>NO</b> No contour map.                                                                                |
|                  |              | <b>YES</b> or <b>ZONES</b> Coloured zone contours.                                                       |
| <b>COMPONENT</b> | <b>X</b>     | Field component for contours.                                                                            |
| <b>MIN</b>       | *            | Minimum contour value. * for automatic setting.                                                          |
| <b>MAX</b>       | *            | Maximum contour value. * for automatic setting.                                                          |
| <b>VECTORS</b>   | <b>NO</b>    | Vector display switch.                                                                                   |
|                  |              | <b>NO</b> Vectors not displayed.                                                                         |
|                  |              | <b>YES</b> Vectors displayed as cones with components specified by <b>VX</b> , <b>VY</b> and <b>VZ</b> . |
| <b>VX</b>        | <b>X</b>     | Expression for x-component of vectors.                                                                   |
| <b>VY</b>        | <b>Y</b>     | Expression for y-component of vectors.                                                                   |
| <b>VZ</b>        | <b>Z</b>     | Expression for z-component of vectors.                                                                   |

| Command             | MAP              |                                                                                                                      |
|---------------------|------------------|----------------------------------------------------------------------------------------------------------------------|
| Parameter           | Default          | Function                                                                                                             |
| <b>PRINT</b>        | <b>NO</b>        | Print options.                                                                                                       |
|                     |                  | <b>NO</b> No output.                                                                                                 |
|                     |                  | <b>YES</b> Output to log file.                                                                                       |
| <b>HEIGHT</b>       | <b>SIZE/3</b>    | Height of histogram.                                                                                                 |
| <b>ERASE</b>        | <b>YES</b>       | Multiple map options:                                                                                                |
|                     |                  | <b>NO</b> Add to existing maps.                                                                                      |
|                     |                  | <b>YES</b> Replace any existing map.                                                                                 |
| <b>LINES</b>        | 10               | Number of line contours.                                                                                             |
| <b>COLSCALAR</b>    | 4                | Table file column for scalar value.                                                                                  |
| <b>COLVECTOR</b>    | 5                | First table file column for vector values.                                                                           |
| <b>GRID</b>         | <b>NO</b>        | Superimpose a grid though data points onto the contour map: <b>YES</b> or <b>NO</b> .                                |
| <b>TITLE</b>        |                  | A title for the contour map.                                                                                         |
| <b>CORNER</b>       | 0                | The corner of the contour map to be used as the starting position for the title: 1, 2, 3 or 4. 0 indicates no title. |
| <b>VECSBYSIZE</b>   | <b>YES</b>       | Vector size options:                                                                                                 |
|                     |                  | <b>NO</b> All vectors the same size. Magnitude of field indicated by colour.                                         |
|                     |                  | <b>YES</b> Vector size indicates magnitude of field. All vector the same colour.                                     |
| <b>VMIN</b>         | *                | Minimum vector length. * for automatic setting.                                                                      |
| <b>VMAX</b>         | *                | Maximum vector length. * for automatic setting.                                                                      |
| <b>VSCALEOPTION</b> | <b>AUTOMATIC</b> | Vector scaling options:                                                                                              |
|                     |                  | <b>ABSOLUTE</b> Vectors scaled by <b>VABSSCALE</b> .                                                                 |
|                     |                  | <b>AUTOMATIC</b> Vectors automatically scaled relative to the map size and then by <b>VAUTOSCALE</b> .               |
| <b>VAUTOSCALE</b>   | 1                | Automatic scaling factor for vectors.                                                                                |
| <b>VABSSCALE</b>    | 1                | Absolute scaling factor for vectors.                                                                                 |

## Notes

The **MAP** command is used to display field values calculated with

- The **CARTESIAN** Command [page 710]

- The FIT Command [page 747] (with **TYPE=LEGENDRE**)
- The POLAR Command [page 788]
- The SPHERICAL Command [page 807]

Maps are overlaid on the three dimensional pictures of the geometry created by [The THREEED Command \[page 823\]](#). Any number of contour maps can be added to the picture (**ERASE=NO**) or a map can replace any previously displayed (**ERASE=YES**). The maps and the scale of values can be temporarily removed from the display using [The WINDOW Command \[page 851\]](#).

Two types of map are available: **CONTOURs** of a scalar **COMPONENT** and **VECTORS** of a field vector with components specified by **VX**, **HY** and **VZ**. **CONTOURs** and **VECTORS** can be displayed with one command.

## Contour maps

Contour maps are displayed if **CONTOUR** is not set to **NO**.

- The contour map can appear at the location of the field points (**contour=zones** or **contour=lines**) or displaced in proportion to the component value as a 3D histogram (**contour=histogram**).
- **LINES** sets the number of contour lines. If **LINES=1**, the line will pass through the positions with the minimum value.
- **HEIGHT** sets the height of the histogram. The default value gives a height of one third of the **THREED** command **SIZE** parameter.
- **GRID=YES** adds a grid though the data points to the contour map.
- A **TITLE** can be added, starting at one of the **CORNERs** of the contour map. **CORNER=0** can be used to disable the title.
- The values of the maximum and minimum contour values can be set with **MAX** and **MIN**. If either **MAX** or **MIN** is set to **\*** the value is calculated from the range of values to be displayed.
- The **MAP** command also calculates the integral of the **COMPONENT**, and stores its value in the system variable **INTEGRAL**. The maximum and minimum values of **COMPONENT** are stored in system variables **MAXIMUM** and **MINIMUM**. The locations of the extreme values are stored in **XATMINIMUM**, **YATMINIMUM**, **ZATMINIMUM**, **XATMAXIMUM**, **YATMAXIMUM**, **ZATMAXIMUM**.

## Vector maps

Vector maps are displayed if **vectors=yes**. The lengths of the vectors can be scaled in several ways:

- **VECSYSIZE=YES** displays vectors of different sizes with the vector length representing the magnitude of the vector.
  - **VSCALEOPTION=AUTOMATIC** calculates a scaling factor so that the longest vector is **VAUTOSCALE**\*5% of the map size.
  - **VSCALEOPTION=ABSOLUTE** scales the vectors by **VABSSCALE**.

- **VECSBYSIZE=NO** displays all the vectors the same size. The colours of the vectors indicate the size of the field they represent. The size of the vectors is **VAUTOSCALE**\*5% of the model size. The setting of **VSCALEOPTION** is ignored.
- The lengths of the longest and shortest vectors can be set with **VMAX** and **VMIN**. If either **VMAX** or **VMIN** is set to \* the limiting length is calculated from the vectors to be displayed. The limiting length is the field value (**SQRT (VX^2+VY^2+VZ^2)**) before any scaling.

## Field values

The values used for the map can come from 2 sources:

- a field buffer: **FILE=TEMP**, with the buffer specified by [The BUFFER Command \[page 708\]](#). Expressions for the scalar **COMPONENT** and the components of the vectors, **VX**, **VY**, **VZ** can use as variables any of the system variables which are currently available (see [System Variables \[page 657\]](#)). The component values are recalculated if the component expression is changed.<sup>1</sup>
- a table file: If no file name extension is given the extension *table* is assumed. The structure of the files is given in [TABLE Files \[page 674\]](#). The component name and values are read from the column given by **COLSCALAR**; the vector components are read from the columns given by **COLVECTOR** and the two subsequent columns.

The **MAP** command can optionally **PRINT** the values it uses to the file **Post\_n.ip**:

- the field point coordinates
- the values of the scalar **COMPONENT**
- the values of the vector components, **VX**, **VY** and **VZ**

If **LINE** contours have been selected the start and end coordinates of all the line segments which make up the contours are also printed.

---

<sup>1</sup>If a component expression includes a user variable and the definition of that variable is changed, but the component expression remains the same, the component values will not be updated.

## The **MOUSE** Command

---

### Summary

Sets the functionality of the mouse buttons.

### Toolbutton



### Command line parameters

|                      |                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------|
| Command              | <b>MOUSE</b>                                                                                         |
| Parameter            | Function                                                                                             |
| <b>BUTTON</b>        | Which button?                                                                                        |
| <code>LEFT</code>    | Left button                                                                                          |
| <code>MIDDLE</code>  | Middle button or wheel                                                                               |
| <code>RIGHT</code>   | Right button                                                                                         |
| <b>MODIFIER</b>      | Additional controls:                                                                                 |
| <code>CONTROL</code> | Simultaneously press the <code>control</code> key                                                    |
| <code>DOUBLE</code>  | Press the mouse button twice in quick succession (only available with the <code>LEFT</code> button). |
| <code>NONE</code>    | No additional controls                                                                               |
| <code>SHIFT</code>   | Simultaneously press the <code>shift</code> key                                                      |
| <b>ACTION</b>        | Action performed:                                                                                    |
| <code>CHECK</code>   | Check that all necessary actions have been assigned                                                  |
| <code>CLEAR</code>   | No action                                                                                            |
| <code>DEFAULT</code> | Restore default settings                                                                             |
| <code>PAN</code>     | Pan the picture                                                                                      |
| <code>PICK</code>    | Pick the high-lighted entity                                                                         |
| <code>ROTATE</code>  | Rotate the picture                                                                                   |
| <code>ZOOM</code>    | Zoom the picture                                                                                     |

## Notes

The **MOUSE** command allows the 4 actions performed by the mouse in the 3d graphics window to be assigned to the 3 mouse buttons. The current settings are saved when the program ends for subsequent runs.

The default mouse functions are:

| Action | Button and Modifier                                                                                            |
|--------|----------------------------------------------------------------------------------------------------------------|
| PAN    | RIGHT or RIGHT with SHIFT                                                                                      |
| PICK   | DOUBLE click LEFT                                                                                              |
| ROTATE | LEFT                                                                                                           |
| ZOOM   | MIDDLE. If the middle button is a wheel, turning the wheel can also be used for ZOOM (Microsoft Windows only). |

While the view of the picture is being changed with rotate, pan or zoom, the program hides some features to improve responsiveness. The whole picture can be seen during these operations by using the shift or control key as well as the appropriate mouse button, unless that combination has been assigned to another function.

There are 3 additional **ACTIONS**:

- **CHECK** can be used to ask the program whether all necessary actions are provided by the current settings. **MOUSE ACTION=CHECK** is run automatically when the **Mouse buttons** dialog is closed.
- **CLEAR** can be used to remove any action from a button/modifier combination.
- **DEFAULT** can be used to restore all the settings to the above defaults.

## The **PICTURE** Command

---

### Summary

Saves the current display in a file or on the clipboard.

### Toolbuttons



### Command line parameters

| Command         | <b>PICTURE</b> |                                                     |                                  |
|-----------------|----------------|-----------------------------------------------------|----------------------------------|
| Parameter       | Default        | Function                                            |                                  |
| <b>SAVE</b>     | <b>NO</b>      | <b>NO</b>                                           | Copy file to clipboard.          |
|                 |                | <b>YES</b>                                          | Save picture in a file.          |
| <b>FILENAME</b> |                | The name of the file to be saved..                  |                                  |
| <b>TYPE</b>     | <b>PNG</b>     | Type of image file format used to save the picture. |                                  |
|                 |                | <b>BMP</b>                                          | Window Bitmap                    |
|                 |                | <b>JPG</b> or <b>JPEG</b>                           | Joint Photographic Experts Group |
|                 |                | <b>PNG</b>                                          | Portable Network Graphics        |
|                 |                | <b>PPM</b>                                          | Portable Pixmap                  |
|                 |                | <b>TIFF</b>                                         | Tagged Image File Format         |
|                 |                | <b>XBM</b>                                          | X11 Bitmap                       |
|                 |                | <b>XPM</b>                                          | X11Pixmap                        |

### Notes

This command allows the current display to be stored on the clipboard, or in a file.

If **SAVE=NO** the image is placed on the clipboard and can then be pasted into another application. Under Linux, the image is placed on the X-Selection.

With **SAVE=YES**, the image is stored in a file. The filename can be specified in the **FILENAME** parameter, and the format can be selected with the **TYPE** parameter.

The size of the window can be set using [The GUIOPTIONS Command \[page 756\]](#) to ensure a consistent size for pictures.

An image can be printed directly using [The PRINT Command \[page 790\]](#).

## The **PLOT** Command

---

### Summary

Plot a graph and integrate field values from **LINE**, **ARC**, **CIRCLE** or **SPHERICAL** commands.

### Command line parameters

|                  |             |                                                                                                                                                                                                                                                           |
|------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command          | <b>PLOT</b> |                                                                                                                                                                                                                                                           |
| Parameter        | Default     | Function                                                                                                                                                                                                                                                  |
| <b>FILE</b>      | <b>TEMP</b> | Name of table file containing field points and values. <b>FILE=TEMP</b> means do not use a file; use the current field buffer.                                                                                                                            |
| <b>COMPONENT</b> | <b>X</b>    | Field component for graph.                                                                                                                                                                                                                                |
| <b>YMINIMUM</b>  | *           | Minimum value for Y-axis of graph. * for automatic setting.                                                                                                                                                                                               |
| <b>YMAXIMUM</b>  | *           | Maximum value for Y-axis of graph. * for automatic setting.                                                                                                                                                                                               |
| <b>ERASE</b>     | <b>YES</b>  | Erase screen switch:<br><br><b>CARTESIAN</b> Erase old picture first and draw new cartesian axes.<br><b>NO</b> Draw new graph on existing axes.<br><b>POLAR</b> Erase old picture first and draw new polar axes.<br><b>YES</b> Same as <b>CARTESIAN</b> . |
| <b>PRINT</b>     | <b>LOG</b>  | Print options:<br><br><b>LOG</b> Output values to log file, <b>Post_n.ip</b> .<br><b>NO</b> No output of values.<br><b>SCREEN</b> Output values to screen.<br><b>YES</b> Output values to screen and log file.                                            |
| <b>GRAPH</b>     | <b>YES</b>  | Graph drawing switch:<br><br><b>NO</b> Do not draw graph.<br><b>YES</b> Draw graph.                                                                                                                                                                       |
| <b>TITLE</b>     | <i>none</i> | Additional title for graph.                                                                                                                                                                                                                               |

| Command   | PLOT      |                                                                                     |
|-----------|-----------|-------------------------------------------------------------------------------------|
| Parameter | Default   | Function                                                                            |
| LOCAL     | GLOBAL    | Coordinate switch for plots on cartesian axes:                                      |
|           |           | CARTESIAN Report cartesian coordinates in the local coordinate system.              |
|           |           | GLOBAL Report coordinates with respect to Global Coordinate System.                 |
|           |           | NO Same as GLOBAL.                                                                  |
|           |           | POLAR Report cylindrical polar coordinates in the local coordinate system.          |
|           |           | SPHERICAL Report spherical polar coordinates in the local coordinate system.        |
|           |           | YES Same as CARTESIAN.                                                              |
| ABSCISSA  | NUMBER    | Abscissa of graph for cartesian axes:                                               |
|           |           | DISTANCE DISTANCE along line.                                                       |
|           |           | NUMBER Point NUMBER.                                                                |
| OFFSET    | 0         | Offset in distance for overlaying second and subsequent graphs on cartesian axes.   |
| STYLE     | AUTOMATIC | Line style:                                                                         |
|           |           | AUTOMATIC Program chooses a different style for each graph drawn on the same axes.  |
|           |           | 0 Solid line.                                                                       |
|           |           | >0 Broken line.                                                                     |
| COLOUR    | AUTOMATIC | Line colour:                                                                        |
|           |           | AUTOMATIC Program chooses a different colour for each graph drawn on the same axes. |
|           |           | name Colour name: TEXT, RED, GREEN, BLUE, YELLOW, CYAN or MAGENTA                   |
|           |           | >0 Colour number.                                                                   |
| COLUMN    | 4         | Table file column for values.                                                       |

## Notes

The **PLOT** command is used to plot graphs and list values of field components previously calculated by

- The ARC Command [page 685],

- The CIRCLE Command [page 718],
- The LINE Command [page 768].

Field values on 2-dimensional patches which only have one point in one direction can also be displayed by the **PLOT** command. These can be calculated using

- The CARTESIAN Command [page 710],
- The POLAR Command [page 788],
- The SPHERICAL Command [page 807].

The values used for the map can come from 2 sources:

- a field buffer: **FILE=TEMP**, with the buffer specified by The BUFFER Command [page 708]. Expressions for the **COMPONENT** can use as variables any of the system variables which are currently available (see System Variables [page 657]).
- a table file: If no file name extension is given the extension *table* is assumed. The structure of the files is given in TABLE Files [page 674]. The component name and values are read from the column given by **COLUMN**.

Two styles of graph are available: cartesian and polar.

- **ERASE=CARTESIAN** is used to select a cartesian graph.  
The coordinates of the field points are displayed on the horizontal axis of the graph and in the list of values. The parameter **LOCAL** specifies how the coordinates should be shown:
  - **GLOBAL**: positions in the global coordinate system;
  - **CARTESIAN**: positions in the field point local coordinate system;
  - **POLAR**: positions in the field point local coordinate system, transformed to cylindrical polar coordinates;
  - **SPHERICAL**: positions in the field point local coordinate system, transformed to spherical polar coordinates.
- **ERASE=POLAR** is used to select a polar graph.
- **ERASE=NO** is used to a second or subsequent line to the same axes.

If more than one graph is drawn on the same set of cartesian axes (**ERASE=NO**), the horizontal coordinates or **ABSCISSA**e of the points used can correspond to the point **NUMBER**s or to the **DISTANCE** along the line. If **ABSCISSA=DISTANCE** an **OFFSET** can be added to the distance along the line.

The parameters allow choices of where the numeric values should be displayed (**PRINT** options), whether or not a graph should be plotted or not (+GRAPH or -GRAPH) and what the Y axis scale limits should be (**YMIN** and **YMAX**).

A **TITLE**, in addition to the value of the integral and the **COMPONENT**, can be added to the key at the bottom of the graph.

When multiple graphs are drawn on the same axes, each line is drawn in a different style or colour. The choice of **STYLE** and **COLOUR** can be **AUTOMATIC** or selected by the user.

The **PLOT** command also calculates the integral of the **COMPONENT**, and stores its value in the system variable **INTEGRAL**. The maximum and minimum values of **COMPONENT** are stored in system variables **MAXIMUM** and **MINIMUM**. The locations of the extreme values are stored in **XATMINIMUM**, **YATMINIMUM**, **ZATMINIMUM**, **XATMAXIMUM**, **YATMAXIMUM**, **ZATMAXIMUM**.

There is no access to the **PLOT** command through the graphical user interface. The **PLOT** command is superseeded by [in The DATALINE Command](#), which also provides additional functionality.

## The **POINT** Command

### Summary

Calculate fields values at a point.

### Toolbutton



### Command line parameters

| Command   | POINT   |                                |
|-----------|---------|--------------------------------|
| Parameter | Default | Function                       |
| XP        | 0       | X-coordinate of the point.     |
| YP        | 0       | Y-coordinate of the point.     |
| ZP        | 0       | Z-coordinate of the point.     |
| COMPONENT | X       | Field component to be printed. |

### Notes

The **POINT** command evaluates field values at a point. The point is specified by parameters **XP**, **YP** and **ZP** which are with respect to the local coordinate system of [The SET Command \[page 798\]](#).

All the currently available system variables are calculated at the point (see [System Variables \[page 657\]](#)). Their values can be used in subsequent **\$ CONSTANT** or **\$ PARAMETER** commands (see [User Variable Commands \[page 55\]](#)). The **POINT** command prints out the coordinates and the value of the **COMPONENT**. Expressions for the **COMPONENT** can use any of the system variables and any user variable.

## The **POLAR** Command

---

### Summary

Calculate fields over a patch specified in rθz coordinates.

### Toolbutton



### Command line parameters

| Command   | <b>POLAR</b> |                                                   |
|-----------|--------------|---------------------------------------------------|
| Parameter | Default      | Function                                          |
| R1        | none         | R-coordinate of the first corner of the surface.  |
| T1        | none         | θ-coordinate of the first corner of the surface.  |
| Z1        | none         | Z-coordinate of the first corner of the surface.  |
| R2        | none         | R-coordinate of the second corner of the surface. |
| T2        | none         | θ-coordinate of the second corner of the surface. |
| Z2        | none         | Z-coordinate of the second corner of the surface. |
| R3        | none         | R-coordinate of the third corner of the surface.  |
| T3        | none         | θ-coordinate of the third corner of the surface.  |
| Z3        | none         | Z-coordinate of the third corner of the surface.  |
| R4        | none         | R-coordinate of the fourth corner of the surface. |
| T4        | none         | θ-coordinate of the fourth corner of the surface. |
| Z4        | none         | Z-coordinate of the fourth corner of the surface. |
| N1        | 10           | Number of points on sides 1 and 3.                |
| N2        | 10           | Number of points on sides 2 and 4.                |
| BUFFER    | Polar        | Name of buffer to store field values.             |

### Notes

The **POLAR** command evaluates field quantities on 4-noded surface patches in cylindrical polar coordinates.

- If the number of points in both directions is greater than one, the results can be displayed by [The MAP Command \[page 775\]](#).
- Single lines of points can be displayed by [The PLOT Command \[page 783\]](#).

For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)).

The patch is specified by its corner points (**R1,T1,Z1**, **R2,T2,Z2**, **R3,T3,Z3** and **R4,T4,Z4**) in cylindrical polar coordinates. The  $\theta=0$  plane is the ZX plane.

The field vectors are evaluated in the Global Coordinate System.

The field quantities are evaluated at **N1\*N2** points. The coordinates of points other than the corners are found by linear interpolation in R,  $\theta$  and Z using a 4-noded isoparametric two-dimensional finite element shape function. The surface therefore can be a circular cylinder, a plane, a cone or even a spiral. To achieve a circular patch, or a sector of a circle it is necessary to put two of the 4 defining points at zero radius, each with the same  $\theta$  coordinate as one of the other two points.

The fields are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## The **PRINT** Command

### Summary

Prints the current display.

### Toolbutton



### Command line parameters

|               |              |
|---------------|--------------|
| Command       | <b>PRINT</b> |
| No parameters |              |

### Notes

This command allows the current display to be printed as a bitmap. A dialog of the available printers and their options for printing is opened. The current display can then be printed on the selected printer.

The printers available are determined by the operating system.

The size of the window can be set using [The GUIOPTIONS Command \[page 756\]](#) to ensure a consistent size for pictures.

An image can be saved to file or copied to the clipboard using [The PICTURE Command \[page 781\]](#).

## The **PROCESSLINE** Command

---

### Summary

Process line data to create a new line.

### Usage

The **PROCESSLINE** command processes line data to create a new line which is a Fourier series or polynomial approximation to the data. It optionally displays the new line on a graph.

### Command line parameters

|               |                                       |                                                                                          |
|---------------|---------------------------------------|------------------------------------------------------------------------------------------|
| Command       | <b>PROCESSLINE</b>                    |                                                                                          |
| Parameter     | Default                               | Function                                                                                 |
| <b>LINE</b>   | The name of the line to be processed. |                                                                                          |
| <b>OPTION</b> | <b>POLYNOMIAL</b>                     | Type of process:                                                                         |
|               |                                       | <b>FOURIER</b> Fourier fitting.                                                          |
|               |                                       | <b>HALFCOS</b> Fourier fitting to data which represents a half or a quarter of a period. |
|               |                                       | <b>HALFSIN</b>                                                                           |
|               |                                       | <b>QUARTERCOS</b>                                                                        |
|               |                                       | <b>QUARTERCOS</b>                                                                        |
| <b>ORDER</b>  | 3                                     | Polynomial fitting.                                                                      |
|               |                                       | The order of the Fourier series or polynomial to be fitted.                              |
| <b>GRAPH</b>  |                                       | The name of a graph to display the processed line.                                       |

### Notes

The **PROCESSLINE** command processes the given **LINE**, fitting either a Fourier series or a polynomial to the data.

A new line is created from the polynomial over the same range and displayed on the **GRAPH**, if the graph name has been specified. The coefficients defining the Fourier series or polynomial are saved as system variables.

### Fourier fitting

- The Fourier series **OPTIONS** are:
  - **FOURIER**: the data represents a full period.

- **HALFCOS**: the data represents the first half period of an even function ( $f(-x)=f(x)$ ).
- **QUARTERCOS**: the data represents the first quarter period of an even function ( $f(-x)=f(x)$ ).
- **HALFSIN**: the data represents the first half period of an odd function ( $f(-x)=-f(x)$ ).
- **QUARTERSIN**: the data represents the first quarter period of an odd function ( $f(-x)=-f(x)$ ).
- Fourier fitting finds Fourier coefficients up to the given **ORDER**.
- The new line is called ***name\_Fn*** where ***name*** is the original line name and ***n*** is the order. It represents a full period for all Fourier options.
- The Fourier coefficients are saved in system variables called ***name\_Fn\_X\_i*** where
  - ***name*** is the original line name,
  - ***n*** is the order,
  - **X** is **A** for coefficients of sine or **B** for coefficients of cosine, and
  - ***i*** the index of the coefficient.

For example, a line called **FIELD** is fitted using a 3rd order Fourier series, the fitted values can be obtained for any value of the horizontal axis value using an expression such as:

```
field_f3_B_0+
field_f3_A_1*sin(theta)+field_f3_B_1*cos(theta) +
field_f3_A_2*sin(2*theta)+field_f3_B_2*cos(2*theta) +
field_f3_A_3*sin(3*theta)+field_f3_B_3*cos(3*theta)
```

where theta is zero at the start of the line and  $2\pi$  at the end. (N.B. The expression has been shown on several lines to aid legibility; when used in the software it must be entered on one line.)

## Polynomial fitting

- Polynomial fitting finds the best fit polynomial of the given **ORDER**.
- The new line is called ***name\_Pn*** where ***name*** is the original line name and ***n*** is the order of the polynomial.
- The polynomial coefficients are saved in system variables called ***name\_Pn\_i*** where ***name*** is the original line name, ***n*** is the order and ***i*** the index of the coefficient. For example, a line called **FIELD** is fitted using a 3rd order polynomial, the fitted values can be obtained for any value of the horizontal axis value using an expression such as:

```
field_p3_0+field_p3_1*x+field_p3_2*x^2+field_p3_3*x^3
```

## The **SELECT** Command

---

### Summary

Select surface facets or elements to be displayed.

### Toolbuttons



**Select components for display and calculation**

**Default Select and Refresh**

**Repeat Select and Refresh**

**Cut Plane**

### Command line parameters

| Command       | <b>SELECT</b> |                                                                                         |
|---------------|---------------|-----------------------------------------------------------------------------------------|
| Parameter     | Default       | Function                                                                                |
| <b>ACTION</b> | <b>ADD</b>    | Create list of labels or select:                                                        |
|               |               | <b>ADD</b> Add label to list of labels to be selected.                                  |
|               |               | <b>DEFAULT</b> Create a list of labels suitable for type of analysis loaded.            |
|               |               | <b>HIDE</b> Add label to list of labels to be hidden.                                   |
|               |               | <b>REMOVE</b> Remove label from list.                                                   |
|               |               | <b>RESET</b> Empty the list of labels.                                                  |
|               |               | <b>RESELECT</b> Make selection. If no list of labels exists, create default list first. |
|               |               | <b>SELECT</b> Make selection.                                                           |
|               |               | <b>TOGGLEADD</b> Toggles label in the list.                                             |

| Command           | SELECT          |                                                                                                      |
|-------------------|-----------------|------------------------------------------------------------------------------------------------------|
| Parameter         | Default         | Function                                                                                             |
| <b>LABEL</b>      | <b>NOTAIR</b>   | Label to be added or removed from list.                                                              |
|                   |                 | <i>material_name</i> Material names including <b>AIR</b> or <b>NOTAIR</b>                            |
|                   |                 | <i>potential_type</i> Potential type: <b>REDUCED</b> , <b>TOTAL</b> or <b>VECTOR</b> .               |
|                   |                 | <i>element_type</i> Element type: <b>LINEAR</b> or <b>QUADRATIC</b> .                                |
|                   |                 | <i>bc_name</i> Boundary condition name.                                                              |
|                   |                 | <i>user_label</i> Labels added by the Modeller or Pre-Processor including <b>ALL</b> .               |
|                   |                 | <i>drive_label</i> Drive label: all conductors with this drive.                                      |
|                   |                 | <i>conductor_number</i> Conductor number.                                                            |
| <b>OPTION</b>     | <b>SURFACES</b> | Type of selection with actions <b>SELECT</b> or <b>RESELECT</b> .                                    |
|                   |                 | <b>ADD</b> Select surfaces then add a number of <b>LAYERS</b> of elements.                           |
|                   |                 | <b>SURFACES</b> Select surfaces.                                                                     |
|                   |                 | <b>ELEMENTS</b> Select elements.                                                                     |
| <b>COIL</b>       | <b>NO</b>       | Conductor selection switch.                                                                          |
|                   |                 | <b>NO</b> Conductors selected for geometry only.                                                     |
|                   |                 | <b>YES</b> Conductors selected for field display.                                                    |
| <b>CUT</b>        | <b>NO</b>       | Cut plane switch.                                                                                    |
|                   |                 | <b>NO</b> No <b>CUT</b> plane.                                                                       |
|                   |                 | <b>YES</b> Select surfaces in the <b>CUT</b> plane.                                                  |
|                   |                 | <b>FRONT</b> Select surfaces in front of <b>CUT</b> plane.                                           |
|                   |                 | <b>BACK</b> Select surfaces behind <b>CUT</b> plane.                                                 |
| <b>THETA</b>      | 0               | $\theta$ Euler angle of <b>CUT</b> plane.                                                            |
| <b>PHI</b>        | 0               | $\phi$ Euler angle of <b>CUT</b> plane.                                                              |
| <b>ZCUT</b>       | 0               | Z-coordinate of <b>CUT</b> plane in local coordinate system defined by <b>THETA</b> and <b>PHI</b> . |
| <b>ZTOLERANCE</b> | 1.0E-5          | Tolerance on <b>ZCUT</b> for <b>CUT=YES</b> .                                                        |
| <b>ACCURACY</b>   | 0               | Maximum facet size on conductors.                                                                    |
|                   |                 | 0 Use element sizes.                                                                                 |
|                   |                 | >0 Subdivide facets so that largest sub-division of a facet is not greater than <b>ACCURACY</b>      |

|                 |               |                                                                                           |
|-----------------|---------------|-------------------------------------------------------------------------------------------|
| Command         | <b>SELECT</b> |                                                                                           |
| Parameter       | Default       | Function                                                                                  |
| <b>LAYERS</b>   | 1             | Number of layers added (or removed if <b>LAYERS</b> is negative) with <b>OPTION=ADD</b> . |
| <b>SYMMETRY</b> | <b>NONE</b>   | How much of the full model (including symmetry copies) will be displayed:                 |
|                 |               | <b>FULL</b> All symmetry copies                                                           |
|                 |               | <b>HALF</b> Half of the copies.                                                           |
|                 |               | <b>NONE</b> Only the section analysed.                                                    |
|                 |               | <b>THIRD</b> One third of the copies.                                                     |
|                 |               | <b>THREE_QUARTERS</b> Three quarters of the copies.                                       |
|                 |               | <b>TWO_THIRDS</b> Two thirds of the copies.                                               |

## Notes

The **SELECT** command creates a "display buffer" containing a selection of the surfaces of the finite element mesh and conductors which are to be displayed with [The THREED Command \[page 823\]](#). (The selection can also be used in [The SURFACE Command \[page 809\]](#) and [The TABLE Command \[page 817\]](#)).

Selection is a two stage process:

1. create a list of labels: labels can be

- added to the list (**ACTION=ADD LABEL=name**)  
For conductors, the label name can be either the conductor number or the drive label.
- removed from the list (**ACTION=REMOVE LABEL=name**)
- toggled in the list (**ACTION=TOGGLEADD LABEL=name**)
- all removed from the list (**ACTION=RESET**)
- added or removed in groups with additional label names:

| Additional labels           | Meaning                              |
|-----------------------------|--------------------------------------|
| <b>ALL_BOUNDARIES</b>       | all surfaces with boundary condition |
| <b>ALL_CONDUCTORS</b>       | all conductors                       |
| <b>ALL_ELEMENTS</b>         | all element types                    |
| <b>ALL_MATERIALSURFACES</b> | all material names on surfaces       |
| <b>ALL_MATERIALVOLUMES</b>  | all material names on volumes        |

|                         |                               |
|-------------------------|-------------------------------|
| <b>ALL_POTENTIALS</b>   | all potential types           |
| <b>ALL_SURFACES</b>     | all surface labels            |
| <b>ALL_USERSURFACES</b> | all user surface labels       |
| <b>ALL_USERVOLUMES</b>  | all user volume labels        |
| <b>ALL_VOLUMES</b>      | all volume labels             |
| <b>NOTAIR</b>           | all material names except air |

- marked as hidden (**ACTION=HIDE LABEL=name**). *Hidden* take precedence over *selected* and can be used to hide parts of the model already selected. For example, to see the **QUADRATIC AIR** elements, add **AIR** and hide **LINEAR**.
- chosen to match the current simulation type (**ACTION=DEFAULT**). The default selections are:

| <b>Simulation</b>                                      | <b>Default Selection</b>               |
|--------------------------------------------------------|----------------------------------------|
| Electromagnetic,<br>Magnetization and<br>Magnetostatic | <b>NOTAIR</b><br><b>ALL_CONDUCTORS</b> |
| conductors only                                        | <b>ALL_CONDUCTORS</b>                  |
| Current Flow,<br>Quench and<br>Thermal                 | <b>NOTAIR</b>                          |
| Charged Particle and<br>Electrostatic                  | <b>NOTAIR</b><br><b>VOLTAGE</b>        |
| High Frequency                                         | <b>ALL_MATERIALVOLUMES</b>             |

**ACTION=RESELECT** also creates the default list of labels, but only if the list is currently empty.

## 2. operate on the list of labels (**ACTION=SELECT** or **ACTION=RESELECT**):

- **OPTION=SURFACES** selects three types of element facets:  
facets which lie in Modeller surfaces or Pre-Processor facets which have labels given by the **LABEL** parameter.  
facets which lie on the surface of Modeller cells or Pre-Processor volumes which have the labels given by the **LABEL** parameter.  
**LABEL=ALL** indicates the element facets on the exterior surface of the mesh.
- **OPTION=ELEMENTS** selects surfaces of all the elements which have volume labels given by the **LABEL** parameter. **LABEL=ALL** indicates all elements.
- **OPTION=ADD** selects surfaces and then moves the selected surface outwards (if number of layers is greater than zero) or inwards (if number of layers is negative) by a number of **LAYERS**.

When creating the set of facets for display, the **SELECT** command can optionally include some or all of the copies of the model implied by symmetry. The **SYMMETRY** parameter can be set to:

- **NONE** to only include the part of the model analysed.
- **FULL** to include all the symmetry copies.
- A symbolic fraction (**QUARTER**, **THIRD**, **HALF**, **TWO\_THIRDS**, **THREE\_QUARTERS**) to include some of the symmetry copies.
- A numerical fraction, e.g. **7/32**, to select some other fraction of the model.

The part of the finite element mesh selected can also be restricted by a cut plane. The **CUT** parameter can be set to **FRONT** or **BACK** to select surfaces of elements in front or behind the plane or to **YES** to select surfaces which lie in the plane. The tolerance on the position of the plane (**ZTOLERANCE**) should be set greater than zero to ensure that element surfaces are found correctly for **CUT=YES** and facets which touch the cut plane from the "wrong side" are eliminated for **CUT=FRONT** and **CUT=BACK**. The cut plane is defined as local XY plane of a coordinate system given by Euler angles **THETA** and **PHI** and the normal distance to the global origin **ZCUT**.

The **ACCURACY** parameter can be used to subdivide conductor surface facets. If **ACCURACY>0**, the facets are subdivided into smaller facets so that no facet linear dimension is larger than **ACCURACY**. If **ACCURACY=0**, no subdivision takes place.

## The **SET** Command

---

### Summary

Set options which affect the field calculations: method, point locations, time, etc.

### Toolbuttons



### Command line parameters

| Command   | SET      |                                                                                                                                              |
|-----------|----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter | Default  | Function                                                                                                                                     |
| FIELDS    | NODAL    | Field calculation method:                                                                                                                    |
|           |          | BOTH<br>Interpolation of <b>NODAL</b> values with magnetic fields and flux densities calculated by <b>INTEGRATION</b>                        |
|           |          | NODAL<br>Interpolation of <b>NODAL</b> values.                                                                                               |
|           |          | INTEGRATION<br><b>INTEGRATION</b> of equivalent magnetization and current sources to calculate magnetic fields and flux densities.           |
| COIL      | NODAL    | Field calculation method for conductors in <b>REDUCED</b> potential elements (ignored if <b>FIELD=INTEGRATION</b> ).<br>NODAL<br>INTEGRATION |
|           |          | Interpolation of <b>NODAL</b> values.<br><b>INTEGRATION</b> of current density.                                                              |
|           |          |                                                                                                                                              |
| ACTIME    | 0        | Time for steady-state ac results: angle in degrees around ac cycle.                                                                          |
| SEARCH    | RESIDENT | SEARCH for field points:<br>RESIDENT<br>ALL                                                                                                  |
|           |          | In <b>RESIDENT</b> (loaded) file only.                                                                                                       |
|           |          | In <b>ALL ACTIVE</b> files.                                                                                                                  |

| Command      | SET                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
|--------------|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|----------|-----------------------------------------------------|--------|-----------------------------|-----------|------------------------------|--------------|------------------------------------------------------------------------------|
| Parameter    | Default                                                                      | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| XLOCAL       | 0                                                                            | X-coordinate of the origin of the input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                               |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| YLOCAL       | 0                                                                            | Y-coordinate of the origin of the input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                               |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| ZLOCAL       | 0                                                                            | Z-coordinate of the origin of the input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                               |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| TLOCAL       | 0                                                                            | $\theta$ Euler angle of input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                                         |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| PLOCAL       | 0                                                                            | $\phi$ Euler angle of input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                                           |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| SLOCAL       | 0                                                                            | $\psi$ Euler angle of input local coordinate system.                                                                                                                                                                                                                                                                                                                                                                                                                           |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| ABORT        | YES                                                                          | Abort field calculations if a field point cannot be found: YES or NO                                                                                                                                                                                                                                                                                                                                                                                                           |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| JCOIL        | NO                                                                           | Calculate source current density at field points in coils: YES or NO                                                                                                                                                                                                                                                                                                                                                                                                           |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| LOOK         | ANYWHERE                                                                     | <p>Choose which material to use for field points on a material interface:</p> <table> <tr> <td>AIR</td><td>Air</td></tr> <tr> <td>ANYWHERE</td><td>First material found which contains the field point</td></tr> <tr> <td>NOTAIR</td><td>Any material other than air</td></tr> <tr> <td>SELECTION</td><td>Volumes SELECTed for display</td></tr> <tr> <td>TRAJECTORIES</td><td>Trajectory volumes, i.e. all the AIR except volumes with the BEAMSTOP label.</td></tr> </table> | AIR | Air | ANYWHERE | First material found which contains the field point | NOTAIR | Any material other than air | SELECTION | Volumes SELECTed for display | TRAJECTORIES | Trajectory volumes, i.e. all the AIR except volumes with the BEAMSTOP label. |
| AIR          | Air                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| ANYWHERE     | First material found which contains the field point                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| NOTAIR       | Any material other than air                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| SELECTION    | Volumes SELECTed for display                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| TRAJECTORIES | Trajectory volumes, i.e. all the AIR except volumes with the BEAMSTOP label. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| FULLSEARCH   | YES                                                                          | Search every element before concluding that a field point is outside the mesh: YES or NO                                                                                                                                                                                                                                                                                                                                                                                       |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |
| DISPLAY      | NO                                                                           | Display field contours on the surfaces of conductors: YES or NO                                                                                                                                                                                                                                                                                                                                                                                                                |     |     |          |                                                     |        |                             |           |                              |              |                                                                              |

| Command                    | SET     |                                                                                         |
|----------------------------|---------|-----------------------------------------------------------------------------------------|
| Parameter                  | Default | Function                                                                                |
| <b>ELEMENTFORCEDENSITY</b> | NO      | Calculate element force densities:                                                      |
|                            |         | <b>ALLLORENTZ</b> Calculate EFD using Lorentz forces in all elements                    |
|                            |         | <b>ALLMAXWELLSTRESS</b> Calculate EFD using Maxwell stress integration in all elements  |
|                            |         | <b>LORENTZ</b> Calculate EFD using Lorentz forces (non-magnetic elements only)          |
|                            |         | <b>MAXWELLSTRESS</b> Calculate EFD using Maxwell stress integration in non-air elements |
|                            |         | <b>NO</b> No calculation of EFD                                                         |
| <b>AVERAGEFORCEDENSITY</b> | YES     | Calculate time-average element force density in steady-state ac simulations:            |
|                            |         | <b>NO</b> Calculate the values corresponding to <b>ACTIME</b> .                         |
|                            |         | <b>YES</b> Calculate time-average values.                                               |
| <b>LOCALACTIVE</b>         | NO      | Activate the local coordinate system for display: <b>YES</b> or <b>NO</b>               |

## Notes

The **SET** command defines parameters which affect the way the field calculation commands operate.

## Field Calculation Methods

The **FIELDS** and **COIL** parameters set the method of field calculation used when processing results. The method also depends on the potential type at the field point.

### Nodal fields

**FIELDS=NODAL**

- **total scalar potential** or **vector potential** regions: The total field is calculated by interpolation of the **NODALLY** averaged values.

- **reduced scalar potential** regions: the field is a combination of the field from the conductors and the field from the finite element mesh. The **COIL** parameter can be set to:
  - **NODAL** to request interpolation of nodal values supplied by the analysis programs.
  - **INTEGRATION** to request evaluation of the conductor fields by direct integration of the defined currents<sup>1</sup>.

## Integral fields

**FIELDS=INTEGRATION**

- The magnetic field strength, **H**, and flux density, **B**, are calculated from magnetization and current density sources (Electromagnetic, Harmonic High Frequency, Magnetization and Magnetostatic simulations only). The field points can be anywhere inside or outside the model. Other field values will be set to zero.
- Magnetic fields can also be calculated by integration of current density sources in Current Flow and Quench simulations.

## Integral fields with nodal

**FIELDS=BOTH**

- All possible fields are first calculated using nodal interpolation. The magnetic field strength and flux density values are then replaced with values calculated by integration.

The **FIELDS** and **COIL** parameters have no effect when recovering fields from conductor-only models. In this case the magnetic field strength, **H** and flux density, **B**, are calculated by integration.

## Field Point Searching

It is possible to have several active database files with the Opera-3d Post-Processor. The **SEARCH** parameter is used to choose the **RESIDENT** file only or **ALL** active files. If **ALL** is selected then for each field point each file will be loaded in turn until the field point is found.

It is sometimes useful to restrict field calculations to points inside particular materials or volumes. This can be used to avoid the multivalued fields on interfaces between different materials and potentials or to prevent particle trajectories from entering solid materials. In order to force the program to use the values in particular materials or volumes, the parameter **LOOK** can be set to

- **AIR**: look only in air.
- **NOTAIR**: look everywhere except in air.
- **SELECTION**: look only in the volumes **SELECTed** for display (see [The SELECT Command \[page 793\]](#)).

---

<sup>1</sup>**COIL=INTEGRATION** is only available in reduced potential air volumes and not at all for Motional EM and Quench simulations.

- **TRAJECTORIES:** look only in **AIR** volumes excluding any with the **BEAMSTOP** label. This is the same restriction as in the trajectory calculations in SCALA (see [The Charged Particle Algorithm \[page 548\]](#)).
- **ANYWHERE:** (the default) use the first occurrence of the field point found irrespective of material or potential.

This options does not affect the **THREED** and **SURFACE** commands (the elements used are set by the **SELECT** command) or the **INTEGRATE** command (the air-side of an interface is used if possible).

## Aborting Field Calculations

When fields are calculated at many points in one command (**ARC**, **CIRCLE**, **LINE**, **CARTESIAN**, **POLAR**, **SPHERICAL** and **TABLE** commands, etc.), the calculations will end (abort) if a point cannot be found. In this case the values for the point not found and all subsequent points will be set to zero (**SET ABORT=YES**). With **SET ABORT=NO**, the field calculations continue for all the requested field points.

The algorithm for testing whether a point is inside the mesh first uses the bounding box of the mesh. If this test succeeds, but the element containing the field point cannot be found, the program can optionally perform a **FULLSEARCH** of every element before concluding that the point is outside. This can be necessary if the surfaces of the model are concave or if there are spaces with no mesh inside the outer boundaries.

If the **LOOK** parameter has been set to restrict the field points to specific volumes or materials, a field point will be "not found" if it exists in the mesh but is in the wrong volume or material.

## Local Coordinate System

The field points given with the **ARC**, **CARTESIAN**, **CIRCLE**, **FIT**, **GRID**, **LINE**, **POINT**, **POLAR** and **SPHERICAL** commands are all defined with respect to a local coordinate system for field calculations. This coordinate system is set by its origin (**XLOCAL**, **YLOCAL**, **ZLOCAL**) and [Euler Angles \[page 83\]](#) (**TLOCAL**, **PLOCAL**, **SLOCAL**). The dialog to set the local coordinate system can also be accessed via the **Set field point local coordinate system** button on the field calculation dialogs. The local coordinate system, if it is different from the Global Coordinate System, is shown on the display in orange and can be used as the major coordinate system for the display by setting **LOCALACTIVE=E=YES**.

## Steady-state AC Results

Steady-state ac results are stored in the Post-Processor as complex numbers. The real and imaginary parts are combined by the program to provide the field values at an angle around the ac cycle (**ACTIME=angle**) (see [System Variables \[page 657\]](#)).

## Current Density in Coils

For field points inside the conductors, the current density vector can optionally be calculated and stored in system variables **JCX**, **JCY** and **JCZ**. This can be selected using **SET +JCOIL**.

## Field Contours on Conductor Surfaces

When [The THREED Command \[page 823\]](#) displays field contours on the surfaces of the model, conductors are normally shown in the conductor colour rather than in field contour colours. Conductor surfaces can also be contoured using **SET +DISPLAY**.

## Element Force Densities

In electromagnetic simulations, element force densities can optionally be calculated using any of the field calculation commands. The values for the current field point are stored in system variables for modulus **EFD**, **EFDMOD** and direction **EFDX**, **EFDY**, **EFDZ** or **EFDR** and **EFDZ**.

There are 2 calculation methods.

- The Lorentz force density method finds  $\mathbf{J} \times \mathbf{B}$  in an element. It is suitable for elements where there are currents and the relative permeability is 1.
- The Maxwell stress integration method integrates the Maxwell stress tensor over the surface of an element. It can be used in any material and includes forces from currents and magnetization but is less accurate than the Lorentz force density method for non-permeable conducting materials.

The calculation is controlled by parameters of the **SET** command:

- **ELEMENTFORCEDENSITY** can be set to:
  - **NO**: do not calculate element force density;
  - **YES**: automatically select Lorentz forces or Maxwell stress integration depending on material properties;
  - **MAXWELLSTRESS**: calculate using Maxwell stress integration in non-air elements; elements in meshed conductors are included;
  - **ALLMAXWELLSTRESS**: calculate using Maxwell stress integration in all elements;
  - **LORENTZ**: calculate using Lorentz forces only in elements where relative permeability is 1;
  - **ALLLORENTZ**: calculate using Lorentz forces for all elements (this leaves out any forces from magnetization).
- In AC simulations, **AVERAGEFORCEDENSITY** can be set to **NO** to use the calculate instantaneous values using **ACTIME** or **YES** to calculate the time-average forces.

## The **SHOW** Command

---

### Summary

List details of the active simulations including the local coordinate systems and replications.

### Command line parameters

| Command       | <b>SHOW</b>   |                                                  |
|---------------|---------------|--------------------------------------------------|
| Parameter     | Default       | Function                                         |
| <b>TITLE</b>  | <b>NO</b>     | Display user titles: <b>YES</b> or <b>NO</b> .   |
| <b>OPTION</b> | <b>ACTIVE</b> | Listing option:                                  |
|               |               | <b>ACTIVE</b> All <b>ACTIVE</b> files.           |
|               |               | <b>FULL</b> Full details of resident simulation. |
|               |               | <b>LOADED</b> All simulations in resident file.  |

### Notes

The **ACTIVATE** Command [page 679] and **The LOAD Command** [page 770] allow many simulations to be active, but only one to be loaded. The same simulation can be activated several times with different parameters. Each database can contain several simulations. To keep track of these files and simulations, the **SHOW** command displays information about their contents.

There are 3 options:

- **OPTION=ACTIVE**: this gives a summary about all the active simulations. More information is given about the currently loaded simulation.
- **OPTION=FULL**: this gives complete information about the currently loaded simulation including all the material properties and analysis specific data.
- **OPTION=LOADED**: this lists all the simulations in the file containing the currently loaded simulation with more information about the currently loaded simulation.

The information includes:

| <b>Database Information</b>                                  |   | <b>Value of OPTION</b>            |                                             |                                                 |
|--------------------------------------------------------------|---|-----------------------------------|---------------------------------------------|-------------------------------------------------|
| <b>item</b>                                                  |   | <b>ACTIVE</b><br>All active files | <b>FULL</b><br>Details of loaded simulation | <b>LOADED</b><br>All simulations in loaded file |
| index number                                                 | • | •                                 | •                                           | •                                               |
| file name                                                    | • | •                                 | •                                           | •                                               |
| analysis program                                             | • | •                                 | •                                           | •                                               |
| analysis type                                                | • | •                                 | •                                           | •                                               |
| simulation number                                            | • | •                                 | •                                           | •                                               |
| number of simulations in the file                            | • | •                                 | •                                           | •                                               |
| simulation status                                            | • | •                                 | •                                           | •                                               |
| unit set                                                     |   | •                                 |                                             |                                                 |
| user title                                                   | • | •                                 |                                             |                                                 |
| file creation date, directory and computer                   | • | •                                 |                                             |                                                 |
| time point (transient) or frequency (steady-state ac)        | • | •                                 | •                                           | •                                               |
| all time points (transient) or frequencies (steady-state ac) |   | •                                 |                                             |                                                 |
| linear or nonlinear analysis                                 | • | •                                 | •                                           | •                                               |
| analysis program specific data                               |   | •                                 |                                             |                                                 |
| number and types of conductors and current density values    |   | •                                 |                                             |                                                 |
| circuit loops                                                |   | •                                 |                                             |                                                 |
| boundary condition labels                                    |   | •                                 |                                             |                                                 |
| material names                                               | • | •                                 |                                             |                                                 |
| material data                                                |   | •                                 |                                             |                                                 |
| numbers of nodes and elements                                |   | •                                 |                                             |                                                 |
| local coordinate system and replications                     | • | •                                 |                                             |                                                 |
| coordinate limits                                            | • | •                                 |                                             |                                                 |

## The **SIMULATION** Command

### Summary

Load another simulation from the database.

### Toolbuttons



**Change the current loaded case**

**Load the previous case in the simulation**

**Load the next case in the simulation**

### Command line parameters

|             |                   |                                                 |
|-------------|-------------------|-------------------------------------------------|
| Command     | <b>SIMULATION</b> |                                                 |
| Parameter   | Default           | Function                                        |
| <b>CASE</b> | 1                 | Simulation number or * for the last simulation. |

### Notes

For databases with more than one simulation, the **SIMULATION** command can be used to load a different case from the database. The **CASE** parameter should be set to choose which simulation should be loaded. **CASE=\*** indicates the last simulation in the database.

In menu mode, the **Load Another Simulation** dialog is populated with a list of simulations with information such as transient time or frequency so that the required simulation can be selected easily. The dialog appears automatically after a new database is opened, so that the simulation can be chosen.

The **SHOW Command [page 804]** can be used to list all the simulations in the database.

## The SPHERICAL Command

---

### Summary

Calculate fields along an arc or over a patch on a spherical surface.

### Toolbutton



### Command line parameters

| Command       | <b>SPHERICAL</b> |                                                               |
|---------------|------------------|---------------------------------------------------------------|
| Parameter     | Default          | Function                                                      |
| <b>RADIUS</b> | <i>none</i>      | Radius of the sphere.                                         |
| <b>THETA1</b> | <i>none</i>      | Starting polar ( $\theta$ ) coordinate of the arc or patch.   |
| <b>THETA2</b> | <i>none</i>      | Final polar ( $\theta$ ) coordinate of the arc or patch.      |
| <b>PHI1</b>   | <i>none</i>      | Starting azimuthal ( $\phi$ ) coordinate of the arc or patch. |
| <b>PHI2</b>   | <i>none</i>      | Final azimuthal ( $\phi$ ) coordinate of the arc or patch.    |
| <b>NTHETA</b> | 10               | Number of points on polar lines.                              |
| <b>NPHI</b>   | 10               | Number of points on azimuthal lines.                          |
| <b>BUFFER</b> | Spherical        | Name of buffer to store field values.                         |

### Notes

The **SPHERICAL** command evaluates field quantities on 4-cornered surface patches in spherical polar coordinates.

- If the number of points in both directions is greater than one, the results can be displayed by [The MAP Command \[page 775\]](#).
- Single lines of points can be displayed by [The PLOT Command \[page 783\]](#).

For each field point all the currently available system variables are calculated and stored in a named field **BUFFER** (see [System Variables \[page 657\]](#)).

## Arcs

An arc is specified by its **RADIUS** with one of **NTHETA** and **NPHI** equal to 1 and the other specifying the number of points along the arc.

- If **NTHETA=1**, the azimuthal arc (parallel to the equator) starts at (**THETA1, PHI1**) and ends at (**THETA1, PHI2**).
- If **NPHI=1**, the polar arc (parallel to the meridian) starts at (**THETA1, PHI1**) and ends at (**THETA2, PHI1**).

## Patches

A patch is specified by its **RADIUS** and its extent in the polar and azimuthal directions (**THETA1** to **THETA2** and **PHI1** to **PHI2**). The field quantities are evaluated at **NTHETA\*NPHI** points. The coordinates of points are found by linear interpolation in  $\theta$  and  $\phi$ .

The polar angles ( $\theta$ ) should be in the range  $0^\circ$  (pole at local  $z=RADIUS$ ) to  $180^\circ$  (pole at local  $z=-RADIUS$ ). The azimuthal angles ( $\phi$ ) should be in the range  $0^\circ$  to  $360^\circ$ . The positive X half of the local ZX plane is at azimuthal angle  $0^\circ$ , and the positive Y half of the local YZ plane is at azimuthal angle  $90^\circ$ .

The field vectors are evaluated in the Global Coordinate System.

The fields are stored in the program in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers. It is also possible to store the evaluated field quantities in a file using [The TABLE Command \[page 817\]](#).

## The **SURFACE** Command

---

### Summary

Integrate field quantity over selected surfaces.

### Tolbutton



### Command line parameters

|                  |                |                                                              |
|------------------|----------------|--------------------------------------------------------------|
| Command          | <b>SURFACE</b> |                                                              |
| Parameter        | Default        | Function                                                     |
| <b>TAVERAGE</b>  | <b>YES</b>     | Time-average switch:                                         |
|                  |                | <b>NO</b> Calculate integrals at time of <b>SET</b> command. |
|                  |                | <b>YES</b> Calculate time-average integrals.                 |
| <b>COMPONENT</b> | <b>X</b>       | Field component to be integrated.                            |

### Notes

The **SURFACE** command integrates a field component expression over the surface selected by [The SELECT Command \[page 793\]](#). Expressions for the **COMPONENT** can use as variables any of the system variables (see [System Variables \[page 657\]](#)) and any user variables.

The system variables **INTEGRAL** and **AREA** are updated with the value of the integral and the area of the selected surface.

In steady-state alternating current models, for integrands which are the product of two field quantities, the integral is a function of time with the form

$$F_x = A + B\cos(2\omega t) + C\sin(2\omega t) \quad (7.7)$$

The values *B* and *C* have little meaning on their own. The time-average value, *A* is the value commonly required. This can be calculated directly using **+TAVERAGE**.

The values of *B* and *C* can be found by setting the times to 0, 45 and 90 (see [The SET Command \[page 798\]](#)), to give values of  $F_x$  at each time:  $F_0$ ,  $F_{45}$  and  $F_{90}$  with the **-TAVERAGE** option.

$$\begin{aligned} A &= \frac{F_0 + F_{90}}{2} \\ B &= \frac{F_0 - F_{90}}{2} \\ C &= F_{45} - A \end{aligned} \quad (7.8)$$

The following commands can be used to achieve this:

- Example - time-average  $F_x$ :

```
set time=0
surface -taverage
$constant #fx0 fx
set time=45
surface
$constant #fx45 fx
set time=90
surface
$constant #fx90 fx
$parameter #fxa 0.5* (#fx0+#fx90)
$parameter #fxb 0.5* (#fx0-#fx90)
$parameter #ffc #fx45-#fxa
```

The values  $F_0$ ,  $F_{45}$  and  $F_{90}$  have little meaning on their own.

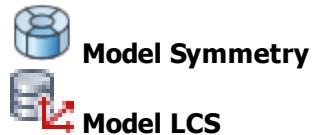
## The **SYMMETRY** Command

---

### Summary

Change the symmetry and local coordinate system of the loaded database.

### Toolbuttons



### Command line parameters

| Command   | SYMMETRY |                                                 |
|-----------|----------|-------------------------------------------------|
| Parameter | Default  | Function                                        |
| SYMMETRY  |          | Rotational symmetry around local Z axis.        |
| RXY       |          | Reflection in local XY plane.                   |
|           |          | NO      No reflection.                          |
|           |          | YES     Reflection with zero Z field.           |
|           |          | INVERSE    Reflection with zero X and Y fields. |
| RYZ       |          | Reflection in local YZ plane.                   |
|           |          | NO      No reflection.                          |
|           |          | YES     Reflection with zero X field.           |
|           |          | INVERSE    Reflection with zero Y and Z fields. |
| RZX       |          | Reflection in local ZX plane.                   |
|           |          | NO      No reflection.                          |
|           |          | YES     Reflection with zero Y field.           |
|           |          | INVERSE    Reflection with zero Z and X fields. |

| Command              | SYMMETRY        |                                                                             |
|----------------------|-----------------|-----------------------------------------------------------------------------|
| <b>MODELSYMMETRY</b> | <b>DATABASE</b> | Symmetry of the model:                                                      |
|                      | <b>DATABASE</b> | Use the symmetry from the background region in the database                 |
|                      | <b>NONE</b>     | No symmetry                                                                 |
|                      | <b>USER</b>     | Use the parameters <b>SYMMETRY</b> , <b>RXY</b> , <b>RYZ</b> and <b>RZX</b> |
| <b>XORIGIN</b>       | 0               | X-coordinate of the coordinate system origin.                               |
| <b>YORIGIN</b>       | 0               | Y-coordinate of the coordinate system origin.                               |
| <b>ZORIGIN</b>       | 0               | Z-coordinate of the coordinate system origin.                               |
| <b>THETA</b>         | 0               | Euler angle defining the coordinate system.                                 |
| <b>PHI</b>           | 0               | Euler angle defining the coordinate system.                                 |
| <b>PSI</b>           | 0               | Euler angle defining the coordinate system.                                 |
| <b>SAVE</b>          | <b>NO</b>       | Save user defined symmetry in the database: <b>YES</b> or <b>NO</b> .       |

## Notes

The **SYMMETRY** command changes the symmetry and local coordinate system of the currently loaded database.

## Symmetry

Symmetry codes can be used to create the complete model from the section which was analysed. When symmetry is in use, all copies of the analysed section can be used in field calculations. Some or all of the copies can be selected for display using [The SELECT Command \[page 793\]](#).

- **MODELSYMMETRY=DATABASE**: the program will use the information stored in the database to create the complete model. This information is either the symmetry specified in the Modeller using [The BACKGROUND Command \[page 135\]](#) or user symmetry specified and **SAVED** with an earlier **SYMMETRY** command.
- **MODELSYMMETRY=USER**: the program will create the complete model using the parameters of this command. This can be used when there is no symmetry information in the database or to override the symmetry information in the database.
- **MODELSYMMETRY=NONE**: only use the part of model which was analysed.

## User defined symmetry

The complete model can be formed by rotation in the local Z axis and reflections in the local coordinate planes. Three types of user symmetry are available.

- **SYMMETRY=n** creates **n** copies by rotating through  $360/n$  degrees around the local Z axis. The sign of the parameter **SYMMETRY** determines the direction of the field in the copies of the model. If **SYMMETRY** is negative the field direction is reversed in alternate copies.
- **RUV=YES** should be used when the field normal to the local *uv* plane is zero. The program reflects the geometry in the *uv* plane of the local system and inverts the sign of the normal field in the reflected copy.
- **RUV=INVERSE** should be used when the tangential field in the local *uv* plane is zero. The program reflects the geometry in the *uv* planes of the local system and inverts the sign of the tangential field in the reflected copy.

The field reflections apply to the principal field of the simulation, i.e. **H** for magnetic field models (Electromagnetic, Magnetization and Magentostatics), **E** for electric field models (Charged Particle, Current Flow, Elektrostatic and High Frequency) and  $\nabla T$  for thermal models (Quench and Thermal).

The maximum number of reflected and rotated copies of the model is 2048.

User defined symmetry can be written to the database (**SAVE=YES**) and will replace any symmetry information already there. (See also [Writing to database files \[page 673\]](#).)

## Local coordinate system

The **SYMMETRY** command sets a local coordinate system by its origin (**XORIGIN**, **YORIGIN**, **ZORIGIN**) and Euler angles (**THETA**, **PHI**, **PSI**). This repositions the whole model in space as if it had been defined in this position.

## System variables

The symmetry parameters are saved in the following system variables:

|                  |                                             |                    |
|------------------|---------------------------------------------|--------------------|
| <b>ROTATIONS</b> | The value of the <b>SYMMETRY</b> parameter. |                    |
| <b>REFXY</b>     | 0                                           | <b>RXY=NO</b>      |
|                  | 1                                           | <b>RXY=YES</b>     |
|                  | -1                                          | <b>RXY=INVERSE</b> |
| <b>REFYZ</b>     | 0                                           | <b>RYZ=NO</b>      |
|                  | 1                                           | <b>RYZ=YES</b>     |
|                  | -1                                          | <b>RYZ=INVERSE</b> |
| <b>REFZX</b>     | 0                                           | <b>RZX=NO</b>      |
|                  | 1                                           | <b>RZX=YES</b>     |
|                  | -1                                          | <b>RZX=INVERSE</b> |

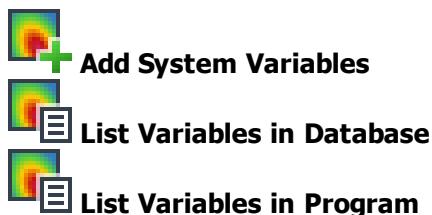
## The **SYSVARIABLE** Command

---

### Summary

Control which system variables are available for field calculations.

### Toolbuttons



### Command line parameters

| Command     | <b>SYSVARIABLE</b> |                                                                    |
|-------------|--------------------|--------------------------------------------------------------------|
| Parameter   | Default            | Function                                                           |
| <b>MODE</b> | <b>PROGRAM</b>     | Action required:                                                   |
|             |                    | <b>ADD</b> Add a system variable to the program from the database. |
|             |                    | <b>DBASE</b> List the system variables in the database.            |
|             |                    | <b>DELETE</b> Delete a system variable from the program.           |
|             |                    | <b>PROGRAM</b> List the system variables in the program.           |
| <b>NAME</b> | <i>none</i>        | Stem name of the variable to be added or deleted. ! for a list.    |
| <b>TYPE</b> | <b>SCALAR</b>      | Type of variable to be added or deleted:                           |
|             |                    | <b>SCALAR</b> scalar.                                              |
|             |                    | <b>VECTOR</b> vector (X, Y and Z components).                      |
| <b>UNIT</b> | <b>DEFAULT</b>     | Unit conversion expression.                                        |
|             |                    | <b>DEFAULT</b> default unit for known variable names.              |
|             |                    | <i>expression</i> unit expression.                                 |

| Command           | <b>SYSVARIABLE</b> |                                                                                  |
|-------------------|--------------------|----------------------------------------------------------------------------------|
| Parameter         | Default            | Function                                                                         |
| <b>REFLECTION</b> | <b>DEFAULT</b>     | Reflection in model boundaries:                                                  |
|                   |                    | <b>DEFAULT</b> default reflection for known variable names and simulation types. |
|                   |                    | <b>INVERSE</b> change of sign.                                                   |
|                   |                    | <b>NO</b> no reflection.                                                         |
|                   |                    | <b>YES</b> same sign.                                                            |

## Notes

The **SYSVARIABLE** command lists, adds and deletes system variables.

Most loading of system variables happens automatically within the program, variables being loaded when they are required. For example, for a Magnetostatic solution, the potential, field strength and flux density are loaded when a database is activated and loaded. The source field strength is loaded automatically if integral coil fields are selected in the **SET** command. System variables are reloaded when a new simulation is loaded.

However, the user might sometimes need a variable which is available in the database but has not yet been loaded, or, for reasons of efficiency, might want to delete some of the system variables which have been loaded. The **SYSVARIABLE** command provides this functionality as well as listing the variables in the database and program.

See [The SET Command \[page 798\]](#) for how to add system variables **JCX**, **JCY** and **JCZ**.

## Adding System Variables

When system variables are loaded the program follows the procedure detailed in [System Variables \[page 657\]](#).

The unit conversion expression for the quantity represented by the system variable should also be given so that the values are scaled appropriately to the user's choice of units. Unit expressions should be given in terms of the following names, which correspond to the parameter names of [The UNITS Command \[page 835\]](#). The program recognizes the variable names used by the solvers and applies the correct unit expression if **UNIT=DEFAULT** is specified.

Unit conversion factors: **LENGU**, **FLUXU**, **FIELU**, **SCALU**, **VECTU**, **CONDU**, **CURDU**, **POWEU**, **FORCU**, **ENERU**, **ELECU**, **DISPU**, **MASSU**.

The program also needs to know how to adjust the signs of the system variables when returning values in reflected images of the mesh. **REFLECTION=YES** gives the same symmetry as the principal field of the simulation, i.e. **H** for magnetic field models, **E** for electric field models and  $\nabla T$  for thermal

models. The program recognizes the variable names used by the solvers and applies the correct reflection if **REFLECTION=DEFAULT** is specified.

Remember that when referencing system variables in the **SYSVARIABLE** command, only the stem of the name should be given after removing the leading **R** or **I** and, for a vector, the trailing **x**, **y** or **z**.

## Deleting System Variables

When system variables are deleted from the program it follows this procedure:

- **TYPE=SCALAR**, for example the magnetic scalar potential, **NAME=POT**: The program deletes the real part (**RPOT**). If the imaginary part (**IPOT**) is available it will be deleted as well. The following variables will be marked as no longer being available: **RPOT, IPOT**.
- **TYPE=VECTOR**, for example the magnetic field strength, **NAME=H**: The program deletes each component (real and imaginary parts if available). The following variables will be marked as no longer being available: **RHX, IHX, RHY, IHY, RHZ, IHZ**.

Remember that when referencing system variables, only the stem of the name should be given after removing the leading **R** or **I** and, for a vector, the trailing **x**, **y** or **z**.

## Listing System Variables

4 types of list are available:

- **MODE=ADD, NAME=!** and **MODE=DBASE** lists the system variables available in the database; **MODE=DBASE** also shows the type of variable:
  - *Node No. indexed* indicates a variable has a value at each node. There are multiple nodes at the surfaces of each cell. This means that such a variable can be discontinuous at material surfaces or boundary conditions.
  - *Node indexed multivalued by material* indicates a variable which is discontinuous at the surfaces of materials.
  - *Node indexed multivalued by potential code* indicates a variable which is discontinuous at the surfaces of different potential types.
  - *Element No. indexed* indicates a variable which has a constant value over each element.
  - *Edge No. indexed* indicates solution of edge variable elements. This should be loaded as a vector.
  - *Face No. indexed* indicates an additional solution from edge variable elements. It should also be loaded as a vector.

In multiphysics analyses, the results from earlier simulations in the database are also available. The list indicates the simulation number for each variable.

- **MODE=DELETE, NAME=!** and **MODE=PROGRAM** lists the system variables in the program. **MODE=PROGRAM** also shows those defined within the software which are marked (**program**) and are always available and so cannot be deleted. Those marked (**database**) have been loaded from the database and therefore can be deleted.

## The **TABLE** Command

---

### Summary

Read and write table files of field values at points.

### Toolbuttons



### Command line parameters

| Command   | <b>TABLE</b> |                                                                                   |
|-----------|--------------|-----------------------------------------------------------------------------------|
| Parameter | Default      | Function                                                                          |
| INFILE    | <b>none</b>  | Name of input data file or option to provide field point coordinates:             |
|           |              | <b>filename</b> Read field points from the file.                                  |
|           |              | <b>DBASE</b> All the nodes in the database.                                       |
|           |              | <b>ELEMENTS</b> The centroids of all the elements in the database.                |
|           |              | <b>NODES</b> All the nodes in the database.                                       |
|           |              | <b>SELECTION</b> All the nodes on the <b>SELECT</b> ed surface.                   |
|           |              | <b>SUBELEMENTS</b> The centroids of the elements in the <b>SELECT</b> ed volumes. |
|           |              | <b>SUBNODES</b> The nodes in the <b>SELECT</b> ed volumes.                        |
|           |              | <b>TEMP</b> The field points from the current field buffer.                       |

| Command                     | TABLE        |                                                                                                                      |
|-----------------------------|--------------|----------------------------------------------------------------------------------------------------------------------|
| Parameter                   | Default      | Function                                                                                                             |
| <b>OUTFILE</b>              | <b>none</b>  | Name of output data file or option for destination of field values:                                                  |
|                             |              | <b>filename</b> Output field points to the file.                                                                     |
|                             |              | <b>DBASE</b> Add fields to the database ( <b>INFILE</b> must contain values for all nodes or all element centroids). |
|                             |              | <b>NULL</b> No output.                                                                                               |
|                             |              | <b>SUBELEMENTS</b> Add fields to the database ( <b>INFILE</b> must contain a subset of element centroid values).     |
|                             |              | <b>SUBNODES</b> Add fields to the database ( <b>INFILE</b> must contain a subset of nodal values).                   |
|                             |              | <b>TEMP</b> Save the values in the programs internal buffer.                                                         |
| <b>F1</b>                   | X            | Expression for values in column 1.                                                                                   |
| <b>F2</b>                   | Y            | Expression for values in column 2.                                                                                   |
| <b>F3</b>                   | Z            | Expression for values in column 3.                                                                                   |
| <b>F4, ..., F12</b>         |              | Expression for values in column 4 to 12.                                                                             |
| <b>NAME1, ..., NAME12</b>   |              | Names for columns.                                                                                                   |
| <b>UNIT1T, ..., UNIT3T</b>  | <b>LENGU</b> | Unit expressions for column 1 to 3.                                                                                  |
| <b>UNIT4T, ..., UNIT12T</b> | 1.0          | Unit expressions for columns 4 to 12.                                                                                |
| <b>COLUMNS</b>              | 4            | Number of columns in output file.                                                                                    |
| <b>BUFFER</b>               | Table        | Name of output buffer to store field values.                                                                         |
| <b>FORMAT</b>               | 2            | Table file format.                                                                                                   |

## Notes

The **TABLE** command reads and writes coordinates and field values in Table file. It is used to transfer data between analysis programs and to interface to other software. It performs three tasks:

1. Read an input source, given by **INFILE**, to obtain field point coordinates and other columns of field values.
2. Calculate field values at the points, if necessary.
3. Output or store the fields.

The value of **INFILE** specifies the source of the field point coordinates:

- **INFILE=TEMP**: the field points contained in an internal buffer created by

- The ARC Command [page 685],
- The CARTESIAN Command [page 710],
- The CIRCLE Command [page 718],
- The FIT Command [page 747] (with **TYPE=LEGENDRE**),
- The LINE Command [page 768],
- The POLAR Command [page 788],
- The SPHERICAL Command [page 807].

The buffer can be specified with [The BUFFER Command \[page 708\]](#).

- **INFILE=NODES** or **INFILE=DBASE**: the coordinates of the nodes in the current database are used.
- **INFILE=ELEMENTS**: the coordinates of the element centroids in the current database are used.
- **INFILE=SELECTION**: the coordinates of the nodes on the selected surfaces are used. Note that tables created using this option cannot subsequently be used with **OUTFILE=SUBNODES**; the option **INFILE=SUBNODES** should be used to create the table instead.
- **INFILE=SUBLEMENTS**: the centroid coordinates of the elements in the volumes **SELECTed** for display will be output.
- **INFILE=SUBNODES**: the nodes in the volumes **SELECTed** for display will be output. Before using this option, [The SET Command \[page 798\]](#) command should be used to look for field points only in selected volumes. This will ensure that the correct values of fields are obtained at nodes where the fields are multivalued.

If there are no selected volumes, the nodes on selected surfaces will be used.

The option **INFILE=SUBNODES** should be used in preference to **INFILE=SELECTION** if it is necessary to load the table back into the database using **OUTFILE=SUBNODES**.

- **INFILE=filename**: field point coordinates are read from a table file.

The value of **OUTFILE** specifies the destination of the field values:

- **OUTFILE=TEMP**: the current set of field values is calculated and stored in a named field **BUFFER**. See [The BUFFER Command \[page 708\]](#) for more information about field buffers.
- **OUTFILE=DBASE**, **OUTFILE=SUBLEMENTS** or **OUTFILE=SUBNODES**: these are intended for use with **INFILE=filename**, where the input file contains coordinates and additional fields.
  - **DBASE**: If the number of entries matches the number of nodes or the number of elements in the current simulation, the field values will be added to the simulation as nodal or element data as appropriate.
  - **SUBLEMENTS**: The input file should contain values for some or all of the elements in the model. As each record of the Table file is processed, the field value(s) will be applied to the first element which contains the field point coordinates. There must not be more than one record for an element and the records must occur in the same order as the elements in the model.

When a new database vector is created using this option, the values will be set to zero for elements not included in the Table file. If the database vector exists, only the data for elements in the imported Table file will be changed.

- **SUBNODES:** The input file should only contain values for the nodes in the volumes **SELECT**ed for display.

When a new database vector is created using this option, the values will be set to zero for nodes not included in the Table file. If the database vector exists, only the data for nodes in the imported Table file will be changed.

Before these new fields can be used in other post-processing commands, they must be added to the list of currently available system variables using [The SYSVARIABLE Command \[page 814\]](#).

See also [Writing to database files \[page 673\]](#).

- **OUTFILE=NULL:** this option does no field calculation or output, but allows the format of an input file to be tested.
- **OUTFILE=filename:** the available set of field values is calculated and the values of up to 12 component expressions are output to a table file. The number of values is specified by the **COLUMNS** parameter.

The expressions for the field values, **F1** to **F12**, can use as variables any of the system variables (see [System Variables \[page 657\]](#)) and any user variables. They are evaluated at each field point in the input file. All the values for one point appear in one record of the output file. Note that surface normal directions are only available with **INFILE=SELECTION**.

The name and unit expression for each column can also be set using parameters **NAME<sub>n</sub>** and **UNIT<sub>nT</sub>**. If a **NAME<sub>n</sub>** is not specified, the corresponding **F<sub>n</sub>** will be used instead.

**FORMAT 1** files are written in internal units and **FORMAT 2** file in the user's units. The unit expressions should be in terms of the unit conversion factor system variables, e.g. **FLUXU/LENGU** (see [The UNITS Command \[page 835\]](#)). For **FORMAT 2** files, the program will convert the expressions to the user's current choice of units, e.g. **FLUXU/LENGU** might become **TESLA/METRE**.

When using tables to copy fields from one simulation to another, the internal **NAMEs** should be used, for example, **RHX**, **RHY**, **RHZ** (and **IHX**, **IHY**, **IHZ** if complex) (see example below).

If no file name extension is given, the extension *table* is supplied for the input and output data files.

The format of the data files is given in section [TABLE Files \[page 674\]](#) which includes an example output data file.

## Copying fields from one simulation to another

The **TABLE** command can be used to copy fields from one simulation to another even if the finite element meshes are different in the two simulations. There are four steps which are illustrated in the [Example \[page 821\]](#):

1. Start with the simulation which will receive the additional fields and create a table file containing its nodal coordinates.
2. Change to the simulation which already contains the fields. Evaluate the fields at the coordinates in the table created in step 1 and write the coordinates and calculated fields to a second table file.

3. Return to the receiving simulation and load the table created in step 2 to store the field values in the database.
4. Add the field values to the list of system variables which are available to post-processing commands using [The SYSVARIABLE Command \[page 814\]](#). (This is done automatically by [The COMBINE Command \[page 725\]](#) when combined field particle tracking is switched on.)

These steps are illustrated below to add magnetic fields from an electromagnetic analysis to a SCALA simulation, so that the SCALA particle tracking can make use of the magnetic fields. The first step is to create a simulation for analysis with SCALA, using either the Modeller or the Pre-Processor. Before solving the simulation with SCALA, the "unsolved" simulation must be loaded into the Post-Processor and commands similar to those below should be used before using SCALA.

## Example

```
/ Step 1: Activate and load SCALA database
activate file=space.op3
load
/ Form table of node coordinates in file
/ spacenodes.table
table infile=nodes outfile=spacenodes.table,
  f1=x f2=y f3=z,
  unit1t=lengu unit2t=lengu unit3t=lengu,
  columns=3
/ Step 2: Activate and load electromagnetic model
activate file=magnet.op3
load
/ Form table of fields in file magnetfields.table
table infile=spacenodes.table,
  outfile=magnetfields.table,
  f1=x f2=y f3=z f4=rhx f5=rhy f6=rhz,
  unit1t=lengu unit2t=lengu unit3t=lengu,
  unit4t=fielu unit5t=fielu unit6t=fielu,
  columns=6
/ Step 3: Add the fields from file magnetfields.table
/ to the SCALA database
load file=1
table infile=magnetfields.table outfile=dbase
/ Step 4: make the new system variables visible
/ (only needed before field calculation commands)
sysvariable mode=add name=h type=vector
```

Exactly the same technique can also be used to add magnetic fields to a simulation which contains an electrostatic simulation so that combined electric and magnetic field particle tracking can be done (see [The COMBINE Command \[page 725\]](#)). Adding electric fields to a magnetic field simulation can be done in the same way, except that electric fields should be specified using **f4=rex f5=rey f6=rez** in the **TABLE** command of step 2.

The method is also used by the thermal solvers to include eddy current heating and by any electromagnetic analysis to include the magnetization calculated by the Magnetization solver(see Opera-3d User Guide).

## Application Notes

The Opera-3d User Guide contains Application notes which illustrate the use of the **TABLE** command:

- "Coupled EM and Thermal Analysis in Opera-3d" illustrates the use of the **TABLE** command to transfer material properties between one simulation and another.
- "Simulation of Magnetization" shows how to transfer magnetization between analyses.

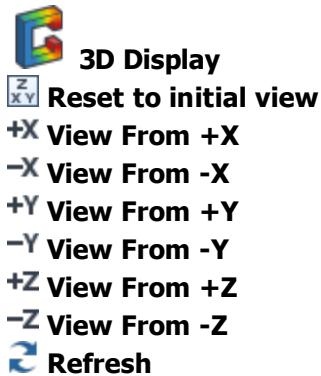
## The **THREED** Command

---

### Summary

Control the display of the geometry.

### Toolbuttons



### Command line parameters

| Command   | THREED  |                                                                                                             |
|-----------|---------|-------------------------------------------------------------------------------------------------------------|
| Parameter | Default | Function                                                                                                    |
| OPTION    | REFRESH | Command option:                                                                                             |
|           |         | GETVIEW      Retrieve view parameters after mouse interaction.                                              |
|           |         | REFRESH      Refresh picture without changing the view.                                                     |
|           |         | SETVIEW      Refresh picture using the view parameters.                                                     |
| SIZE      | 10      | Display extends from the origin by SIZE in each direction.<br>SIZE=0 requests the initial view of the model |
| ROTX      | 20      | Rotation of model around X axis.                                                                            |
| ROTY      | 20      | Rotation of model around Y axis.                                                                            |
| ROTZ      | 0       | Rotation of model around Z axis.                                                                            |
| XORIGIN   | 0       | X coordinate at centre of picture                                                                           |
| YORIGIN   | 0       | Y coordinate at centre of picture                                                                           |

| Command      | THREED    |                                                 |                                                                                                                               |
|--------------|-----------|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Parameter    | Default   | Function                                        |                                                                                                                               |
| ZORIGIN      | 0         | Z coordinate at centre of picture               |                                                                                                                               |
| PERSPECTIVE  | YES       | Perspective switch:                             |                                                                                                                               |
|              |           | YES                                             | Perspective view.                                                                                                             |
|              |           | NO                                              | Orthographic view.                                                                                                            |
| FACETANGLE   | 10        | Largest angle subtended by a curved facet.      |                                                                                                                               |
| LINECOLOUR   | YES       | Colour used for outlines:                       |                                                                                                                               |
|              |           | YES                                             | Material colour.                                                                                                              |
|              |           | NO                                              | Text colour.                                                                                                                  |
| XASPECT      | 1         | Scaling for X coordinates                       |                                                                                                                               |
| YASPECT      | 1         | Scaling for Y coordinates                       |                                                                                                                               |
| ZASPECT      | 1         | Scaling for Z coordinates                       |                                                                                                                               |
| TYPE         | SURFACE   | Type of display:                                |                                                                                                                               |
|              |           | SURFACE                                         | Surfaces of the model in material colours.                                                                                    |
|              |           | COMPONENT                                       | Contours of field COMPONENT on model.                                                                                         |
| COMPONENT    | X         | Field component for contours.                   |                                                                                                                               |
| MIN          | *         | Minimum contour value. * for automatic setting. |                                                                                                                               |
| MAX          | *         | Maximum contour value. * for automatic setting. |                                                                                                                               |
| VECTORS      | NO        | Vector display switch:                          |                                                                                                                               |
|              |           | NO                                              | Vectors not displayed.                                                                                                        |
|              |           | YES                                             | VECTORS displayed as cones at centres of element facets using components VX, VY and VZ, and current directions in conductors. |
| VX           | X         | Expression for x-component of vectors.          |                                                                                                                               |
| VY           | Y         | Expression for y-component of vectors.          |                                                                                                                               |
| VZ           | Z         | Expression for z-component of vectors.          |                                                                                                                               |
| VSCALEOPTION | AUTOMATIC | Vector scaling options:                         |                                                                                                                               |
|              |           | ABSOLUTE                                        | Vectors scaled by VABSSCALE.                                                                                                  |
|              |           | AUTOMATIC                                       | Vectors automatically scaled relative to the model size and then by VAUTOSCALE.                                               |
| VAUTOSCALE   | 1         | Automatic scaling factor for vectors.           |                                                                                                                               |
| VABSSCALE    | 1         | Absolute scaling factor for vectors.            |                                                                                                                               |

| Command       | THREED    |                                                 |                                                                                      |
|---------------|-----------|-------------------------------------------------|--------------------------------------------------------------------------------------|
| Parameter     | Default   | Function                                        |                                                                                      |
| VECSBYSIZE    | YES       | Vector size options:                            |                                                                                      |
|               |           | NO                                              | All vectors are the same size and the magnitude of field is indicated by the colour. |
|               |           | YES                                             | Vector size indicates the magnitude of field; all vectors are the same colour.       |
| VMIN          | *         | Minimum vector length. * for automatic setting. |                                                                                      |
| VMAX          | *         | Maximum vector length. * for automatic setting. |                                                                                      |
| DEFORMED      | NONE      | Deformed shape options:                         |                                                                                      |
|               |           | DEFORMED                                        | Outline and solid views of the model deformed.                                       |
|               |           | NONE                                            | Shapes not deformed.                                                                 |
|               |           | OUTLINE                                         | Outline view of the model deformed.                                                  |
|               |           | SOLID                                           | Solid view of the model deformed.                                                    |
| DSCALEOPTION  | AUTOMATIC | Deformation scaling options:                    |                                                                                      |
|               |           | ABSOLUTE                                        | Deformation scaled by DABSSCALE.                                                     |
|               |           | AUTOMATIC                                       | Deformation automatically scaled relative to the model size and then by DAUTOSCALE.  |
| DAUTOSCALE    | 1         | Automatic scaling factor for deformation.       |                                                                                      |
| DABSSCALE     | 1         | Absolute scaling factor for deformation.        |                                                                                      |
| USEROUTERELEM | NO        | Calculate field components:                     |                                                                                      |
|               |           | NO                                              | From elements inside the selected volume.                                            |
|               |           | YES                                             | From elements outside the selected volume.                                           |

## Notes

The **THREED** command updates the 3D picture of the model. The pictures consists of the three dimensional geometry of the finite element mesh and conductors. The **THREED** command uses as data the 'display buffer', which is created by the **SELECT** command. If 'flux-tubes' have been added to the display buffer by the **VIEW** command, they will be displayed as well. Pictures can be line-drawings or can show coloured surfaces. It is also possible to overlay the geometry with displays of the field quantities.

The colours in the pictures can represent the materials (**TYPE=SURFACE**) or alternatively, scalar field quantities, or single components of vector field quantities can be displayed as coloured zone contours (**TYPE=COMPONENT**). Vector field quantities can be displayed as **VECTORS** at the centroid of the element facets. If the simulation is stress analysis, the geometry can be displayed **DEFORMED**.

The picture consists of the surface facets and wires which make up the model. Two types of picture can be displayed:

- **TYPE=SURFACE** uses colours to represent different materials. The volumes, surfaces and edges to be displayed are chosen using [The SELECT Command \[page 793\]](#).
- **TYPE=COMPONENT** uses colours to represent different levels of **COMPONENT**. The contours are normally not shown on conductor surfaces but conductors can be included in the field contouring using [The SET Command \[page 798\]](#).

## Field values

Field values for contouring or vectors are normally calculated in the elements inside the selected surface. In some cases, e.g. electrostatic fields on the surface of an electrode, the fields in the neighbouring elements are more useful. To calculate values in the elements outside a surface, **USEOUTERELEM** should be set to **YES**.

## Field contours

Expressions for the scalar **COMPONENT** and vector components (**VX**, **VY**, **VZ**) can use as variables any of the program defined system variables listed in section [System Variables \[page 657\]](#). The component values are recalculated if the component expression is changed.<sup>1</sup>

The values of the maximum and minimum contour zones can be set with **MAX** and **MIN**. If either **MAX** or **MIN** is set to \* the value is calculated from the range of values to be displayed. The system variables **MAXIMUM** and **MINIMUM** are updated by the command. The locations of the extreme values are stored in **XATMINIMUM**, **YATMINIMUM**, **ZATMINIMUM**, **XATMAXIMUM**, **YATMAXIMUM**, **ZATMAXIMUM**. The [VOLUME Command \[page 848\]](#) also calculates extreme values and their locations but it only uses values in the interior of the volumes.

## Field vectors

**VECTORS** can be added to **SURFACE** or **COMPONENT** displays. The directions of the vectors are set using the parameters **VX**, **VY** and **VZ**. The lengths of the vectors can be scaled in several ways:

- **VECSBYSIZE=YES** displays vectors of different sizes with the vector length representing the magnitude of the vector.
  - **VSCALEOPTION=AUTOMATIC** calculates a scaling factor so that the longest vector is **VAUTOSCALE**\*5% of the model size.
  - **VSCALEOPTION=ABSOLUTE** scales the vectors by **VABSSCALE**.
- **VECSBYSIZE=NO** displays all the vectors the same size. The colours of the vectors indicate the size of the field they represent. The size of the vectors is **VAUTOSCALE**\*5% of the model size. The setting of **VSCALEOPTION** is ignored.

---

<sup>1</sup>If a component expression includes a user variable and the definition of that variable is changed, but the component expression remains the same, the component values will not be updated.

- The lengths of the longest and shortest vectors can be set with **VMAX** and **VMIN**. If either **VMAX** or **VMIN** is set to \* the limiting length is calculated from the vectors to be displayed. The limiting length is the field value ( $\text{SQRT}(\text{vx}^2 + \text{vy}^2 + \text{vz}^2)$ ) before any scaling.

## Deformed geometry

For stress analysis simulations, the calculated displacements can be displayed as a deformation of the geometry. Either or both the **OUTLINE** and the **SOLID** view of the geometry can be **DEFORMED** so that the original and deformed shapes can be compared. The magnitude of the deformation can be scaled in two ways:

- DSCALEOPTION=AUTOMATIC** calculates a scaling factor so that the largest displacement is **DAUTOSCALE**\*5% of the model size.
- DSCALEOPTION=ABSOLUTE** scales the displacements by **DABSSCALE**.

## Rotate, pan and zoom

After a new database is loaded, the first use of the command **THREED OPTION=REFRESH** will refresh the display to the default view which matches the size and position of the model. The view can be adjusted using the mouse buttons, which are initially set to:

- left button: rotate
- middle button: zoom
- right button: pan

but can be adjusted using [The MOUSE Command \[page 779\]](#).

## Options

The **OPTION** parameter also controls the view of the model:

- OPTION=GETVIEW**: updates the values of the parameters **SIZE**, **ROTX**, **ROTY**, **ROTZ**, **XORIGIN**, **YORIGIN**, **ZORIGIN**, **XASPECT**, **YASPECT** and **ZASPECT** following interaction with the mouse.
- OPTION=SETVIEW**: uses the current values of **SIZE**, **ROTX**, **ROTY**, **ROTZ**, **XORIGIN**, **YORIGIN**, **ZORIGIN**, **XASPECT**, **YASPECT** and **ZASPECT**.

When a local coordinate system is active (see [Local Coordinate System \[page 802\]](#)), these parameters are interpreted with respect to the field point local coordinate system.

**SIZE=0**: gives an initial view of the visible parts of the model, contours, vectors and trajectories depending on the settings of [The WINDOW Command \[page 851\]](#).

Perspective view can be switched off using **PERSPECTIVE=NO**.

The **FACETANGLE** parameter allows the user to request more precise display of curved surfaces. The program subdivides each quadratic element facet into sub-facets so that no sub-facet subtends an angle greater than **FACETANGLE**. Reducing **FACETANGLE** will increase the number of sub-facets so that they better match the curved surface.

The parameters **XASPECT**, **YASPECT** and **ZASPECT** can be adjusted to obtain views of models with high aspect ratios. The values should be between 0 and 1. For example, a pipe of radius 1 unit and length 100 units could be displayed with **XASPECT** and **YASPECT** left at the default value of 1 and **ZASPECT=0.01**.

If vectors (with size representing value) are displayed while non-uniform axis scaling is being used, the lengths of the vectors should be compared against an unscaled axis.

- **OPTION=REFRESH:** updates the picture if necessary without changing the view unless a new database has been loaded in which case the default view is used.

## Related commands

Other commands can be used to change the picture:

- The [MAP Command \[page 775\]](#) can add contour or vector maps to the picture.
- The [VIEW Command \[page 840\]](#) can add particle trajectories to the picture.
- Different parts of the display can be temporarily hidden using [The WINDOW Command \[page 851\]](#): axes, solid colours, outlines, contour maps, vectors and trajectories.
- Conductors can be temporarily hidden using [The CONDUCTOR Command \[page 727\]](#).

## The **TITLE** Command

---

### Summary

Adds title, date and time to the display.

### Command line parameters

| Command         | <b>TITLE</b>        |                          |
|-----------------|---------------------|--------------------------|
| Parameter       | Default             | Function                 |
| <b>STRING</b>   | <i>none</i>         | A graphics window title. |
| <b>POSITION</b> | <b>NONE</b>         | Title position:          |
|                 | <b>BOTTOMCENTRE</b> | Bottom centre            |
|                 | <b>BOTTOMLEFT</b>   | Bottom left              |
|                 | <b>BOTTOMRIGHT</b>  | Bottom right             |
|                 | <b>NONE</b>         | No title                 |
|                 | <b>TOPCENTRE</b>    | Top centre               |
|                 | <b>TOPLEFT</b>      | Top left                 |
|                 | <b>TOPRIGHT</b>     | Top right                |
| <b>DATE</b>     | <b>TOPLEFT</b>      | Time/date position:      |
|                 | <b>BOTTOMCENTRE</b> | Bottom centre            |
|                 | <b>BOTTOMLEFT</b>   | Bottom left              |
|                 | <b>BOTTOMRIGHT</b>  | Bottom right             |
|                 | <b>NONE</b>         | No date and time         |
|                 | <b>TOPCENTRE</b>    | Top centre               |
|                 | <b>TOPLEFT</b>      | Top left                 |
|                 | <b>TOPRIGHT</b>     | Top right                |

### Notes

The **TITLE** command controls the display of a title and the date and time. There is a choice of 6 positions for each, or **NONE** to omit that item. If the same position is chosen for both the title and the date, the title appears above the date and time.

The version number of the software can be included in the title using the string and system variables, **VERSION** using:

```
title string='Version: &version&'
```

or

```
title string='Version: %real(version,5)'
```

The first line of the user title of the currently loaded database is also available in string variable **TITLE**:

```
title string=&title&
```

## The **TRACK** Command

---

### Summary

Calculate trajectories of charged particles in electric and magnetic fields.

### Toolbutton



### Command line parameters

| Command     | <b>TRACK</b> |                                                                                                               |
|-------------|--------------|---------------------------------------------------------------------------------------------------------------|
| Parameter   | Default      | Function                                                                                                      |
| X0          | 0            | Initial X-coordinate of the particle.                                                                         |
| Y0          | 0            | Initial Y-coordinate of the particle.                                                                         |
| Z0          | 0            | Initial Z-coordinate of the particle.                                                                         |
| THETA       | 0            | Theta Euler angle defining particle direction.                                                                |
| PHI         | 0            | Phi Euler angle defining particle direction.                                                                  |
| PSI         | 0            | Psi Euler angle defining particle direction.                                                                  |
| VOLTS       | 1            | Accelerating voltage (initial energy= <b>VOLTS*CHARGES</b> [eV]).                                             |
| CHARGES     | -1           | <b>CHARGES</b> on the particle in elementary charge units. -1 is the charge on an electron.                   |
| MASS        | 1            | Particle rest <b>MASS</b> in electron rest mass units.                                                        |
| BEAMCURRENT | 1            | Current associated with the track. It can be an expression in terms of the electric field at the start point. |
| STEP        | 1            | <b>STEP</b> length between output points.                                                                     |
| NSTEP       | 100          | Number of steps to be calculated.                                                                             |
| TOLERANCE   | 0.01         | Accuracy required.                                                                                            |

| Command        | TRACK           |                                                                                                        |
|----------------|-----------------|--------------------------------------------------------------------------------------------------------|
| Parameter      | Default         | Function                                                                                               |
| <b>OPTION</b>  | <b>PARTICLE</b> | Options:                                                                                               |
|                |                 | <b>PARTICLE</b>                                                                                        |
|                |                 | Single <b>PARTICLE</b>                                                                                 |
|                |                 | <b>FUNCTION</b>                                                                                        |
|                |                 | Use electromagnetic forces on charged particle calculated from fields given by user defined variables. |
| <b>PATTERN</b> | <b>FLUXTUBE</b> | Flux line                                                                                              |
|                | <b>BEAM</b>     | A test pattern defined by <b>PATTERN</b> and <b>LINES</b> .                                            |
|                |                 |                                                                                                        |
| <b>PATTERN</b> | 1.E-4           | Size of the rectangular grid <b>TEST</b> pattern.                                                      |
| <b>LINES</b>   | 5               | Number of X and Y lines in a <b>TEST</b> pattern.                                                      |
| <b>FILE</b>    | <i>none</i>     | Name of file to save trajectories.                                                                     |
| <b>STATUS</b>  | <b>NEW</b>      | <b>STATUS</b> of the <b>TRACK</b> file:                                                                |
|                |                 | <b>CLEAR</b>                                                                                           |
|                |                 | Overwrite data in an existing file.                                                                    |
|                |                 | <b>NEW</b>                                                                                             |
|                |                 | Create a <b>NEW</b> file. The user is asked whether to overwrite the file if it already exists.        |
| <b>PRINT</b>   | <b>OLD</b>      | Append data to an existing file.                                                                       |
|                | <b>UNKNOWN</b>  | Create a new file or overwrite an existing file without asking the user.                               |
| <b>DISPLAY</b> | <b>YES</b>      | <b>DISPLAY</b> the trajectory on the 3d window: <b>YES</b> or <b>NO</b> .                              |

## Notes

The **TRACK** command calculates the trajectories of charged particles through the electric and/or magnetic fields (including full relativistic correction) or follows flux lines. Combined field particle tracking is available for Harmonic High Frequency simulations and also when additional fields have been added to an electrostatic or magnetostatics simulation (see section [The COMBINE Command \[page 725\]](#)).

The parameter, **OPTION** selects single particles (**OPTION=PARTICLE**), test patterns (**OPTION=N=BEAM**), single particles in functional fields (**OPTION=FUNCTION**) or flux lines (**OPTION=N=FLUXTUBE**).

## Single particle

The **PARTICLE** option traces the trajectory of a single charged particle moving in the magnetic, electric or combined fields (see [The COMBINE Command \[page 725\]](#)).

## Beam of particles

The **BEAM** option forms a pattern of many particles which start in the plane normal to the initial direction. The particles start from the intersection points of a square orthogonal grid. The grid can be given a size (**PATTERN**) and the number of grid **LINES** in each direction can be set.

The forces on the beam come from magnetic, electric or combined fields (see [The COMBINE Command \[page 725\]](#)).

## User defined function

The **FUNCTION** option traces the trajectory of a single particle subject to electric and magnetic fields defined as functions using user defined variables. The user can define fields which vary as a function of time, position or any other solution quantities. The user variables should be called

- **#BX, #BY, #BZ**
- **#EX, #EY, #EZ**

These user variables must be evaluated in the current electric field and flux density units<sup>1</sup>.

**Example:** To calculate the trajectory of a particle in an electric field which can be expressed as the superposition of two independently computed fields  $E = E_0 + E_1 e^{\left(\frac{1}{t}\right)}$ , and assuming that the components of E0 and E1 have been loaded into the database using the **TABLE** command, the parameter functions for **#EX, #EY** and **#EZ** could be defined as

```
$parameter #EX E0X+EXP(1/TOF)*E1X
$parameter #EY E0Y+EXP(1/TOF)*E1Y
$parameter #EZ E0Z+EXP(1/TOF)*E1Z
```

where **TOF**, the time of flight from the start of the trajectory, is used to determine the time variation of the functions.

## Flux line

A trajectory is created that follows the magnetic or electric **FLUX** line from the starting point. The sign of **CHARGE** determines the direction of travel along the flux line, its magnitude has no effect. The meaning of **FLUX** depends on the analysis type of the simulation:

|                                                                                               |                       |
|-----------------------------------------------------------------------------------------------|-----------------------|
| Electromagnetic, High Frequency, Magnetization, Magnetostatic simulations and conductors only | magnetic flux density |
| Charged Particle and Electrostatic simulations                                                | electric flux density |
| Current Flow simulations                                                                      | current density       |

<sup>1</sup>Note that this is different from Opera-2d, where the user variables must be evaluated in internal units.

## Trajectory Parameters

Tracking starts at the point **X0, Y0, Z0**. The initial coordinates (and direction) are defined in the local coordinate system of the **SET** command. In steady-state ac simulations the time at the start of the track is also given by [The SET Command \[page 798\]](#).

For charged particles, the initial direction (defined by the [Euler Angles \[page 83\]](#) **THETA, PHI** and **PSI**), the initial energy (**VOLTS\*CHARGES**) and **MASS** of the particles can be set. The initial direction is along the z-axis of the local coordinate system defined by the Euler angles.

A current is associated with each track. This can be set explicitly to a value in amps or can be calculated by the program from an expression in terms of the electric field (**EX, EY, EZ**), coordinates (**X, Y, Z**) and velocity (**VELX, VELY, VELZ**) at the start of the trajectory.

The calculation is controlled by the **STEP** length along the trajectory, the number of steps (**NSTEP**) and the relative **TOLERANCE** which applies to coordinates and velocities or flux densities. **STEP** only determines the distance between the displayed points on the trajectory (although the value should be related to the minimum size of any feature of the field distribution) - the **TOLERANCE** is achieved by adaptive integration. **STEP** is measured along the trajectory and hence the maximum total trajectory length calculated is **STEP\*NSTEP**.

The calculation continues until the number of steps (**NSTEP**) has been reached or the trajectory crosses the surface of the field calculation volume. The field calculation volume is normally the whole mesh but can be restricted to particular volumes with the **SET** command (see [Field Point Searching \[page 801\]](#)). Perhaps the most useful option is **SET LOOK=TRAJECTORIES** which limits field calculations and trajectories to **AIR** volumes, excluding any which have the label **BEAMSTOP**. This matches the algorithm used by SCALA (see [The Charged Particle Algorithm \[page 548\]](#)).

**Note:** there is no limit to the number of steps allowed per trajectory.

## Trajectory files

The trajectory coordinates are stored in a binary **FILE**. The file **STATUS** can be set to:

- **NEW**: a new file will be created to contain the data. If the file given already exists, the user will be asked if it should be overwritten.
- **UNKNOWN**: an existing file will be overwritten, or a new file will be created if the file given does not already exist. (This option is not available from the GUI dialog.)
- **OLD**: the trajectories to be appended to an existing file.
- **CLEAR**: an existing file will be overwritten.

The format of the track file is described in section [TRACK files \[page 677\]](#). If **DISPLAY=YES**, the trajectories are also displayed; if **STATUS=OLD**, the new trajectories are added to the display, otherwise any trajectories already displayed are deleted. The trajectories stored in a **TRACK** file can also be displayed and processed in a number of ways using [The VIEW Command \[page 840\]](#).

## The **UNITS** Command

---

### Summary

Set units to be used for physical quantities.

### Toolbutton



### Command line parameters

| Command       | <b>UNITS</b>   |                                     |
|---------------|----------------|-------------------------------------|
| Parameter     | Default        | Function                            |
| <b>LENGTH</b> | <b>CM</b>      | Unit for length:                    |
|               |                | <b>CM</b> centimetre                |
|               |                | <b>INCH</b> inch                    |
|               |                | <b>METRE</b> metre                  |
|               |                | <b>MICRON</b> micron                |
|               |                | <b>MM</b> millimetre                |
| <b>FLUX</b>   | <b>GAUSS</b>   | Unit for magnetic flux density:     |
|               |                | <b>GAUSS</b> gauss                  |
|               |                | <b>KGAUSS</b> kilogauss             |
|               |                | <b>TESLA</b> tesla                  |
| <b>FIELD</b>  | <b>OERSTED</b> | Unit for magnetic field strength:   |
|               |                | <b>AM</b> ampere/metre              |
|               |                | <b>OERSTED</b> oersted              |
| <b>SCALAR</b> | <b>OCM</b>     | Unit for magnetic scalar potential: |
|               |                | <b>AMPERE</b> ampere                |
|               |                | <b>OCM</b> oersted centimetre       |

| Command      | UNITS   |                                     |
|--------------|---------|-------------------------------------|
| Parameter    | Default | Function                            |
| VECTOR       | GCM     | Unit for magnetic vector potential: |
|              |         | GCM gauss centimetre                |
|              |         | WBM weber/metre                     |
| CONDUCTIVITY | SCM     | Unit for conductivity:              |
|              |         | SCM siemens/centimetre              |
|              |         | SIN siemens/inch                    |
|              |         | SM siemens/metre                    |
|              |         | SMM siemens/millimetre              |
|              |         | SMU siemens/micron                  |
| CURD         | ACM2    | Unit for current density:           |
|              |         | ACM2 ampere/centimetre <sup>2</sup> |
|              |         | AIN2 ampere/inch <sup>2</sup>       |
|              |         | AM2 ampere/metre <sup>2</sup>       |
|              |         | AMM2 ampere/millimetre <sup>2</sup> |
|              |         | AMU2 ampere/micron <sup>2</sup>     |
| POWER        | WATT    | Unit for power:                     |
|              |         | ERGS erg/second                     |
|              |         | HP horse power                      |
|              |         | WATT watt                           |
| FORCE        | NEWTON  | Unit for force:                     |
|              |         | DYNE dyne                           |
|              |         | GRAMMEF gram force                  |
|              |         | KGF kilogram force                  |
|              |         | LBF pound force                     |
|              |         | NEWTON newton                       |
| ENERGY       | JOULE   | Unit for energy:                    |
|              |         | BTU British Thermal Unit            |
|              |         | ERG erg                             |
|              |         | JOULE joule                         |

| Command              | UNITS         |                                       |
|----------------------|---------------|---------------------------------------|
| Parameter            | Default       | Function                              |
| <b>ELECTRIC</b>      | <b>VCM</b>    | Unit for electric field strength:     |
|                      |               | VCM volt/cm                           |
|                      |               | VIN volt/inch                         |
|                      |               | VM volt/metre                         |
|                      |               | VMM volt/millimetre                   |
|                      |               | VMU volt/micron                       |
| <b>DISPLACEMENT</b>  | <b>CCM2</b>   | Unit for electric flux density:       |
|                      |               | CCM2 coulomb/cm <sup>2</sup>          |
|                      |               | CM2 coulomb/metre <sup>2</sup>        |
| <b>MASS</b>          | <b>KG</b>     | Unit for mass:                        |
|                      |               | GRAM or GRAMME gram                   |
|                      |               | KG kilogram                           |
|                      |               | LB pound                              |
| <b>PRESSURE</b>      | <b>PASCAL</b> | Unit for pressure:                    |
|                      |               | DYNECM2 dyne/centimetre <sup>2</sup>  |
|                      |               | LBIN2 pound force/inch <sup>2</sup>   |
|                      |               | MPASCAL megapascal                    |
|                      |               | PASCAL pascal                         |
| <b>CHARGEDENSITY</b> | <b>CCM3</b>   | Unit for charge density:              |
|                      |               | CCM3 coulomb/centimetre <sup>3</sup>  |
|                      |               | CM3 coulomb/metre <sup>3</sup>        |
|                      |               | CMM3 coulomb/millimetre <sup>3</sup>  |
|                      |               | CMU3 coulomb/micron <sup>3</sup>      |
|                      |               | MCM3 micro coulomb/metre <sup>3</sup> |
| <b>EPOTENTIAL</b>    | <b>VOLT</b>   | Unit for electric potential:          |
|                      |               | KV kilovolt                           |
|                      |               | MICROV microvolt                      |
|                      |               | MILLIV millivolt                      |
|                      |               | MV megavolt                           |
|                      |               | VOLT volt                             |

## Notes

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 series of predefined character strings corresponding to commonly used units.

**N.B.** The unit of electric scalar potential is always volt and of charge density is coulomb/*length\_unit*<sup>3</sup>. The unit of temperature is always degrees Kelvin. Temperature gradient is given in K/*length\_unit*, whilst heat flux is in *power\_unit*/*length\_unit*<sup>2</sup>.

Unit conversion is performed on the basic system variables. Other system variables are derived from these, and will have the correct units only if consistent units are used for the basic quantities.

When a database is loaded, the current set of units is overwritten with the unit set chosen in the Modeller or Pre-Processor when the database was created (see section [The SOLVERS Command \[page 485\]](#)).

The current set of units is listed at the right-hand side of the display if the information box is visible (see [The GUIOPTIONS Command \[page 756\]](#)).

## System and String Variables

The unit conversion factors are stored in 2 sets of variables.

- Conversion between current units and internal units can be done using variables with names formed from the first 4 characters of the parameter names followed by **U**. For example, **LENGU** is the multiplying factor to convert a length into internal units.
- Conversion between current units and other units can be done using variables **CONVERTFROMunit** and **CONVERTTOunit**, for example:
  - on input, multiplying by **CONVERTFROMINCH** will convert a length in inches to the current length unit;
  - on output, multiplying by **CONVERTTOTESLA** will convert a flux density in the current unit set to tesla.

Similar variables are available for all quantities and units, i.e. all items in column 3 of the parameter table (see [Command line parameters \[page 835\]](#)).

The current unit parameter values are stored in string variables with the same name as the first set of conversion factors. For example, when the program starts, string variable **LENGU** will have the value **CM**.

## Internal Units

Internally, the software uses a mixed unit set as follows:

| Quantity                  | Unit                    |
|---------------------------|-------------------------|
| length                    | cm                      |
| magnetic flux density     | gauss                   |
| magnetic field strength   | oersted                 |
| magnetic scalar potential | oersted cm              |
| magnetic vector potential | gauss cm                |
| conductivity              | siemens/cm              |
| current density           | amp/cm <sup>2</sup>     |
| power                     | watt                    |
| force                     | newton                  |
| energy                    | joule                   |
| electric field strength   | V/cm                    |
| displacement current      | coulomb/cm <sup>2</sup> |
| mass                      | gram                    |
| pressure                  | pascal                  |
| charge density            | coulomb/cm <sup>3</sup> |
| electric potential        | volt                    |

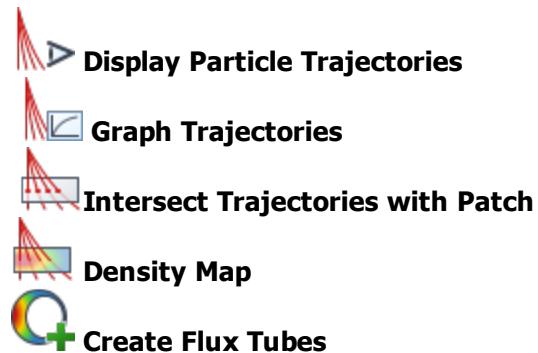
## The **VIEW** Command

---

### Summary

Display and process particle trajectories.

### Toolbuttons



### Command line parameters

|             |                |                                                                                                                                              |
|-------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Command     | <b>VIEW</b>    |                                                                                                                                              |
| Parameter   | Default        | Function                                                                                                                                     |
| <b>FILE</b> | <i>none</i>    | Name of file containing trajectories.                                                                                                        |
| <b>PLOT</b> | <b>DISPLAY</b> | Type of output required:                                                                                                                     |
|             |                | <b>DISPLAY</b> <i>DISPLAY</i> trajectories on the current view of the model.                                                                 |
|             |                | <b>INTERSECTIONS</b> Draw graphs of <b>INTERSECTIONS</b> of the trajectories with <b>CARTESIAN</b> , <b>POLAR</b> or <b>SPHERICAL</b> patch. |
|             |                | <b>TEMP</b> Calculate data for density maps from intersections with <b>CARTESIAN</b> , <b>POLAR</b> or <b>SPHERICAL</b> patch.               |
|             |                | <b>TRACKS</b> Draw graphs of the trajectories.                                                                                               |
|             |                | <b>TUBE</b> Create tubes of displayable facets around trajectories.                                                                          |

|                 |                |                                                                                                                                                                                                                                       |
|-----------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command         | <b>VIEW</b>    |                                                                                                                                                                                                                                       |
| Parameter       | Default        | Function                                                                                                                                                                                                                              |
| <b>XAXIS</b>    | <b>Y</b>       | The variable plotted on the horizontal axis of the graph.                                                                                                                                                                             |
| <b>YAXIS</b>    | <b>X</b>       | The variable plotted on the vertical axis of the graph.                                                                                                                                                                               |
| <b>XMINIMUM</b> | *              | Lower limit for graph horizontal axis. (*) for automatic setting of limit.)                                                                                                                                                           |
| <b>XMAXIMUM</b> | *              | Upper limit for graph horizontal axis. (*) for automatic setting of limit.)                                                                                                                                                           |
| <b>YMINIMUM</b> | *              | Lower limit for graph vertical axis. (*) for automatic setting of limit.)                                                                                                                                                             |
| <b>YMAXIMUM</b> | *              | Upper limit for graph vertical axis. (*) for automatic setting of limit.)                                                                                                                                                             |
| <b>LINE</b>     | <b>YES</b>     | Graph plotting style:<br><br>NO Plot using symbols.<br><br>YES Plot using lines.                                                                                                                                                      |
| <b>COLOUR</b>   | <b>YES</b>     | Use of colour in the displays:<br><br><b>FUNCTION</b> Colours represent the values of <b>FUNCTION</b> .<br><br><b>NO</b> Use text colour.<br><br><b>YES</b> Colours represent trajectory numbers.                                     |
| <b>FUNCTION</b> | <b>CURRENT</b> | Expression used to assign colours to the trajectories when <b>COLOUR=FUNCTION</b> , the density function when <b>PLOT=TEMP</b> or the radius of the tubes when <b>PLOT=TUBE</b> . The value is also printed when printing is enabled. |
| <b>PRINT</b>    | <b>NO</b>      | Printing switch:<br><br>NO No printing.<br><br>YES All points or intersections printed to dialogue file.                                                                                                                              |
| <b>SAMPLE</b>   | <b>8</b>       | Sample size used for smoothing in <b>PLOT=INTERSECTIONS</b> and <b>PLOT=TEMP</b> .<br><br><1 Fraction of total current, power or particle flux for moving window method.<br><br>>1 Number of beamlets to include in FFT smoothing.    |

| Command        | VIEW    |                                                                                        |
|----------------|---------|----------------------------------------------------------------------------------------|
| Parameter      | Default | Function                                                                               |
| ERASE          | YES     | Erase previous display for PLOT=DISPLAY, PLOT=INTERSECTIONS and PLOT=TRACKS:           |
|                |         | NO Add to existing display.                                                            |
|                |         | YES Erase display first.                                                               |
| SYMMETRY       | NO      | Include symmetry copies of tracks:                                                     |
|                |         | NO Use each trajectory once.                                                           |
|                |         | YES Include all symmetry copies of the trajectories.                                   |
| EMITTER        | ALL     | Name or number of emitter, or ALL for all emitters.                                    |
| STOPAFTERFIRST | NO      | Stop looking for intersections after finding the first in each trajectory (YES or NO). |

## Notes

The **VIEW** command re-displays and processes trajectories calculated and stored in **TRACK** files [page 677] by either

- The **TRACK** Command [page 831], or
- Charged Particle Solver [page 548].

## Types of **VIEW**

### 3d trajectories

**PLOT=DISPLAY**: this option allows the trajectories to be displayed on the current view of the model using the text colour or multiple colours (see [Viewing in Colour \[page 845\]](#)). The trajectories are overlaid on the three dimensional pictures of the geometry created by [The THREED Command \[page 823\]](#). The tracks can be temporarily removed from the display using [The WINDOW Command \[page 851\]](#). With **ERASE=NO**, trajectories are added to any trajectories already displayed; with **ERASE=YES**, existing trajectories are removed from the display first.

### Graphs of trajectories

**PLOT=TRACKS**: this option displays graphs of the trajectories. The variables plotted on the axes of the graphs can be selected from the set of [Active System Variables \[page 845\]](#). For example to display an axisymmetric projection of the results, with Z on the horizontal axis of the graph, use **VIEW XAXIS=Z, YAXIS=R**

By default, the axis limits are automatically set to contain the functions plotted but limits can be specified.

## Graphs of intersections

**PLOT=INTERSECTIONS**: this option displays graphs of the intersections of trajectories with a patch. The patch should be preselected using [The BUFFER Command \[page 708\]](#). The intersection positions on the **CARTESIAN**, **POLAR** or **SPHERICAL** patch are calculated for all trajectories and symmetry images. The search for intersections can optionally be terminated after finding the first intersection with each trajectory (**STOPAFTERFIRST=YES**).

The intersections are ordered with respect to the independent variable defined as the horizontal axis of the output graph. The current in the beams is projected onto the independent variable and the linear current density function, **DENSITY**, is calculated and added to the set of active system variables. The intersections are plotted as graphs using the set of [Active System Variables \[page 845\]](#). When the intersections are calculated, the linear current density (**DENSITY**) is also computed for each intersection point. This is evaluated statistically using varying sample sizes up to a maximum given by the **SAMPLE** parameter. The expected error is also calculated and will be displayed using error bars if **YAXIS=DENSITY** is selected. The sample size should be less than 1/10 of the number of intersections. If larger values are used the results will be smooth but the errors will be larger.

For example: to find the radial distribution of current density for a circular beam, use

```
view plot=inte,yaxis=density,
xaxis=sqrt((x-x0beam)^2+(y-y0beam)^2+(z-z0beam)^2)
```

This will collect together all the intersections at the similar radii in order to calculate the density as a function of radial coordinate.

The effect of smoothing can also be seen by comparing system variables:

- the total current (**TOTALCURRENT**) and
- the **INTEGRAL** of the smoothed current, calculated with **YAXIS=DENSITY**.

## Maps of intersections

**PLOT=TEMP**: this option calculates data for density maps of the intersections of trajectories with a patch. The patch should be preselected using [The BUFFER Command \[page 708\]](#). The intersection positions on the **CARTESIAN**, **POLAR** or **SPHERICAL** patch are calculated for all trajectories and symmetry images. The search for intersections can optionally be terminated after finding the first intersection with each trajectory (**STOPAFTERFIRST=YES**).

The density is then calculated using the **FUNCTION** expression for each intersection point on the patch:

- **FUNCTION=CURRENT** calculates the current density in the beam;
- **FUNCTION=BEAMLETPOWER** calculates the power density in the beam;
- **FUNCTION=NUMBER** calculates the particle number density in the beam;

- **FUNCTION** can also be set to any other valid expression of the [Active System Variables \[page 845\]](#).

The density is calculated by weighted integration over the patch to find a smooth distribution from the relatively small number of sample beamlets. The values are smoothed using a choice of algorithms specified by the **SAMPLE** parameter. Two smoothing methods available:

- **SAMPLE>1**: Smoothing uses a fast fourier transform (FFT).

The **SAMPLE** size determines the amount of smoothing. The expected error is also calculated and will be displayed using error bars if **YAXIS=DENSITY** is selected. The sample size should be less than 1/10 of the number of intersections. If larger values are used the results will be smooth but the errors will be larger.

- **SAMPLE<1**: Smoothing uses a moving window average.

The value of **SAMPLE** specifies the fraction of total current, power or particle flux that should be used to calculate the density. The method calculates the beamlet intersections with a surface patch and then determines the current density distribution on the surface by adapting the size of a window around each point on the surface until it contains the required fraction of the total current, power or particle flux.

This method gives more reliable answers on polar and spherical patches.

The calculated density can then be displayed using [The MAP Command \[page 775\]](#) with **COMPONENT-T=BEAMDENSITY**. The total current crossing the patch is saved in the system variable **TOTALCURRENT**.

## Flux tubes

**PLOT=TUBE**: This option creates additional facets in the display buffer. The facets for tubes around the segments of the trajectories, with the radius determined by the **FUNCTION** expression. The tubes can then be displayed using [The THREED Command \[page 823\]](#).

The flux tubes can be:

- hidden temporarily by [The WINDOW Command \[page 851\]](#). Use the **Trajectories** icon .
- cleared from the display buffer with [The SELECT Command \[page 793\]](#). Use the **Repeat selection and refresh** icon  which performs a **SELECT** and **THREED** command.

## Emitter Names and Numbers

When processing or viewing trajectories from a SCALA analysis which has multiple emitters, the trajectories can be restricted to a single **EMITTER** given by its name or number. The **EMITTER** number can be used to determine the colour of the trajectories on the display (see the next section).

To enable scripting, the number and names of the emitters are stored in variables:

- system variable **EMITTERS** holds the number of emitters;

- string variable **EMITTER\_n** holds the name of the *n*th emitter.

## Viewing in Colour

In each of the **PLOT**ting options, the lines can be displayed using the text colour (**COLOUR=NO**), with colours representing the trajectory number (**COLOUR=YES**) or with colours representing some characteristic of the position along the trajectory (**COLOUR=FUNCTION**).

Functional colours are selected using the value of the **FUNCTION** expression. **FUNCTION** can be assigned to expressions in terms of the active system variables listed below. For example,

**COLOUR=FUNCTION , FUNCTION=TOF**

will display the graphs coloured according to the time of flight from the start points.

To colour flux tubes, the **COMPONENT** has to be specified in the **THREED** command. Coloured contours are displayed on the surfaces which were added to the display buffer.

## Active System Variables

The **VIEW** command has its own set of system variables which are available in expressions for the graph plotting variables (**XAXIS** and **YAXIS**) and the **FUNCTION** used to determine the colours of the lines.

| <b>System Variables for PLOT=DISPLAY and PLOT=TRACKS</b> |                                                        |
|----------------------------------------------------------|--------------------------------------------------------|
| <b>X</b>                                                 | X coordinates of points on the trajectory.             |
| <b>Y</b>                                                 | Y coordinates of points on the trajectory.             |
| <b>Z</b>                                                 | Z coordinates of points on the trajectory.             |
| <b>R</b>                                                 | Radial coordinate of points on trajectory.             |
| <b>XSTART</b>                                            | X coordinate of the first point on the trajectory.     |
| <b>YSTART</b>                                            | Y coordinate of the first point on the trajectory.     |
| <b>ZSTART</b>                                            | Z coordinate of the first point on the trajectory.     |
| <b>RSTART</b>                                            | Radial coordinate of the first point on trajectory.    |
| <b>VELX</b>                                              | X component of particle velocity.                      |
| <b>VELY</b>                                              | Y component of particle velocity.                      |
| <b>VELZ</b>                                              | Z component of particle velocity.                      |
| <b>BEAMLETPOWER</b>                                      | The power in the track (beamlet).                      |
| <b>CURRENT</b>                                           | The current in the track (beamlet).                    |
| <b>NUMBER</b>                                            | Number of particles per second in the track (beamlet). |
| <b>TOF</b>                                               | Time of flight from the start of the trajectory.       |

| <b>System Variables for PLOT=DISPLAY and PLOT=TRACKS</b> |                                         |
|----------------------------------------------------------|-----------------------------------------|
| <b>Q</b>                                                 | Electronic charge on the particle.      |
| <b>M</b>                                                 | Mass of the particle in electron units. |
| <b>EMITTER</b>                                           | Emitter number.                         |

| <b>System Variables for PLOT=INTERSECTIONS</b> |                                                                                                                                                                            |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>X</b>                                       | X coordinates of the intersection points.                                                                                                                                  |
| <b>Y</b>                                       | Y coordinates of the intersection points.                                                                                                                                  |
| <b>Z</b>                                       | Z coordinates of the intersection points.                                                                                                                                  |
| <b>X0BEAM</b>                                  | X coordinate of the centre of the beam on the intersection surface.                                                                                                        |
| <b>Y0BEAM</b>                                  | Y coordinate of the centre of the beam on the intersection surface.                                                                                                        |
| <b>Z0BEAM</b>                                  | Z coordinate of the centre of the beam on the intersection surface.                                                                                                        |
| <b>TXBEAM</b>                                  | X component of the mean direction vector for the beam at the intersection surface.                                                                                         |
| <b>TYBEAM</b>                                  | Y component of the mean direction vector for the beam at the intersection surface.                                                                                         |
| <b>TZBEAM</b>                                  | Z component of the mean direction vector for the beam at the intersection surface.                                                                                         |
| <b>DENSITY</b>                                 | The linear current density in the beam calculated by integrating the trajectory currents along lines normal to the axis of the graph (units are amp/ <i>length_unit</i> ). |
| <b>VELX</b>                                    | X component of particle velocity.                                                                                                                                          |
| <b>VELY</b>                                    | Y component of particle velocity.                                                                                                                                          |
| <b>VELZ</b>                                    | Z component of particle velocity.                                                                                                                                          |
| <b>BEAMLETPOWER</b>                            | The power in the track (beamlet).                                                                                                                                          |
| <b>CURRENT</b>                                 | The current in the track (beamlet).                                                                                                                                        |
| <b>NUMBER</b>                                  | Number of particles per second in the track (beamlet).                                                                                                                     |
| <b>TOF</b>                                     | Time of flight from the start of the trajectory.                                                                                                                           |
| <b>Q</b>                                       | Electronic charge on the particle.                                                                                                                                         |
| <b>M</b>                                       | Mass of the particle in electron units.                                                                                                                                    |
| <b>EMITTER</b>                                 | Emitter number.                                                                                                                                                            |

| <b>System Variables for PLOT=TUBE</b> |                                                        |
|---------------------------------------|--------------------------------------------------------|
| X                                     | X coordinates of the intersection points.              |
| Y                                     | Y coordinates of the intersection points.              |
| Z                                     | Z coordinates of the intersection points.              |
| VELX                                  | X component of particle velocity.                      |
| VELY                                  | Y component of particle velocity.                      |
| VELZ                                  | Z component of particle velocity.                      |
| BEAMLETPOWER                          | The power in the track (beamlet).                      |
| CURRENT                               | The current in the track (beamlet).                    |
| NUMBER                                | Number of particles per second in the track (beamlet). |
| TOF                                   | Time of flight from the start of the trajectory.       |
| Q                                     | Electronic charge on the particle.                     |
| M                                     | Mass of the particle in electron units.                |
| EMITTER                               | Emitter number.                                        |
| All available field quantities.       |                                                        |

## Printing Trajectory Data

When printing is selected (**PRINT=YES**) and **COLOUR=FUNCTION** has been set, the following data is output to the dialogue file, *Post\_n.lp*:

- **PLOT=DISPLAY** and **PLOT=TRACKS**, for all points in all trajectories:

```
X Y Z VELX VELY VELZ FUNCTION
```

- **PLOT=INTERSECTIONS**, for all intersections:

```
CURRENT X Y Z VELX VELY VELZ FUNCTION
```

## The **VOLUME** Command

---

### Summary

Integrate field quantities over volumes.

### Toolbutton



### Command line parameters

| Command          | <b>VOLUME</b>      |                                                              |
|------------------|--------------------|--------------------------------------------------------------|
| Parameter        | Default            | Function                                                     |
| <b>ACTION</b>    | <b>INTEGRATE</b>   | Create list of volumes or integrate:                         |
|                  |                    | <b>ADD</b> Add volume(s) to list.                            |
|                  |                    | <b>INTEGRATE</b> Integrate.                                  |
|                  |                    | <b>REMOVE</b> Remove volume(s) from list.                    |
|                  |                    | <b>RESET</b> Empty list of volumes.                          |
|                  |                    | <b>TOGGLEADD</b> Toggles volume(s) in the list.              |
| <b>LABEL</b>     | <b>ALL_VOLUMES</b> | Volumes to be added or removed from list:                    |
|                  |                    | <i>material</i> Material name.                               |
|                  |                    | <i>label</i> Volume label.                                   |
|                  |                    | <b>ALL_VOLUMES</b> All volumes.                              |
| <b>TAVERAGE</b>  | <b>YES</b>         | Time-average switch.                                         |
|                  |                    | <b>NO</b> Calculate integrals at time of <b>SET</b> command. |
|                  |                    | <b>YES</b> Calculate time-average integrals.                 |
| <b>ADAPT</b>     | <b>NO</b>          | Adaptive integration switch.                                 |
|                  |                    | <b>NO</b> Use 8 gauss-points in each element.                |
|                  |                    | <b>YES</b> Use up to 216 gauss-points in each element.       |
| <b>COMPONENT</b> | <b>X</b>           | Field component to be integrated.                            |

## Notes

The **VOLUME** command integrates a field component expression in the whole model space or in labelled sets of elements. Expressions for the **COMPONENT** can use as variables any of the system variables listed in section [System Variables \[page 657\]](#) and any user variables.

The command operates as a 2-stage process:

1. form a list of volumes. Initially all volumes are in the list.
  - **ACTION=RESET** empties the list.
  - **ACTION=ADD** adds volumes to the list by **LABEL**.
  - **ACTION=REMOVE** removes volumes from the list by **LABEL**.
  - **ACTION=TOGGLEADD** toggles the membership of a volume in the list.
  - Labels can be material names, volume labels including element and potential types and user labels or **ALL\_VOLUMES**.
2. integrate forces using **ACTION=INTEGRATE**.

The system variables **INTEGRAL** and **VOLUME** are updated with the value of the integral and the total volume integrated. The minimum and maximum values of the integrand and the locations at which they occur are also stored in system variables:

- **MINIMUM, MAXIMUM**;
- **XATMINIMUM, YATMINIMUM, ZATMINIMUM**;
- **XATMAXIMUM, YATMAXIMUM, ZATMAXIMUM**.

Note that the integration points do not include points on the surface of the volume. If the minimum or maximum values occur on the surface, [The THREED Command \[page 823\]](#) will do better at calculating the extreme values and their locations.

The basis for the integration is the finite element mesh. The integrals are performed in each element using first-order gaussian quadrature. However, in reduced potential volumes if the coil field is calculated by integration or anywhere if the total field is calculated by integration, the first-order quadrature is insufficient to match the field variation in an element. Switching on adaptive integration (**+ADAPT**) enables the program to use up to 9<sup>th</sup>-order Gaussian quadrature in each element to increase the accuracy of the integrals. See section [The SET Command \[page 798\]](#) for information about field calculation methods.

Reflected and rotated images of the model, specified by the **ACTIVATE** command, are NOT included in the integrations done by the **VOLUME** command.

In steady-state alternating current models, for integrands which are the product of two field quantities, the integral is a function of time with the form

$$E = A + B\cos(2\omega t) + C\sin(2\omega t) \quad (7.9)$$

The values *B* and *C* have little meaning on their own. The time-average value, *A* is the value commonly required. This can be calculated directly using **+TAVERAGE**.

The values of  $B$  and  $C$  can be found by setting the times to 0, 45 and 90 (see section [The SET Command \[page 798\]](#)), to give values of  $E$  at each time:  $E_0$ ,  $E_{45}$  and  $E_{90}$  with the **-TAVERAGE** option.

$$\begin{aligned} A &= \frac{E_0 + E_{90}}{2} \\ B &= \frac{E_0 - E_{90}}{2} \\ C &= E_{45} - A \end{aligned} \quad (7.10)$$

The following commands can be used to achieve this:

- Example - time-average energy:

```
set time=0
volume comp=0.5*(bx*hx+by*hy+bz*hz) -taverage
$constant #en0 integral
set time=45
volume
$constant #en45 integral
set time=90
volume
$constant #en90 integral
$parameter #ena 0.5* (#en0+#en90)
$parameter #enb 0.5* (#en0-#en90)
$parameter #enc #en45-#ena
```

The values  $E_0$ ,  $E_{45}$  and  $E_{90}$  have little meaning on their own.

## The **WINDOW** Command

---

### Summary

Show or hide parts of the display.

### Toolbuttons

-  **Solid View of Model**
-  **Outline View of Model**
-  **View Vectors on Surface of Model**
-  **View Contour Map**
-  **View Vector Map**
-  **View Isosurface**
-  **View Trajectories**
-  **View Contour or Trajectory Labels**
-  **Cycle axis display mode**

### Menu route

**View -> Parts of the Display**

### Command line parameters

| Command   | <b>WINDOW</b> |                                                                      |
|-----------|---------------|----------------------------------------------------------------------|
| Parameter | Default       | Function                                                             |
| AXES      | ALL           | Show coordinate axes:                                                |
|           |               | ALL Show all axes.                                                   |
|           |               | CYCLE Cycle between ALL, MAJOR, TRIAD and NONE.                      |
|           |               | MAJOR Show major axes of active coordinate system.                   |
|           |               | NONE Hide all axes.                                                  |
|           |               | TRIAD Show triads at origins of global and local coordinate systems. |
| SOLID     | YES           | Show solid view of model: YES or NO                                  |
| OUTLINE   | YES           | Show outline view of model: YES or NO                                |

| Command    | WINDOW  |                                                                                                                                                                                                                                                                                                                                            |
|------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter  | Default | Function                                                                                                                                                                                                                                                                                                                                   |
| VECTORS    | YES     | Show vectors on the surface of the model: YES or NO                                                                                                                                                                                                                                                                                        |
| CONTOURMAP | YES     | Show contour map: YES or NO                                                                                                                                                                                                                                                                                                                |
| VECTORMAP  | YES     | Show vector map: YES or NO                                                                                                                                                                                                                                                                                                                 |
| TRACKS     | YES     | Show trajectories: YES or NO                                                                                                                                                                                                                                                                                                               |
| ISOSURFACE | YES     | Show iso-valued surfaces: YES or NO                                                                                                                                                                                                                                                                                                        |
| LABELS     | YES     | Show contour or trajectory labels:<br>CONTOUR Show labels for contour maps.<br>CYCLE Cycle through available labels and none.<br>ISOSURFACE Show labels for iso-valued surfaces.<br>NO Show no labels.<br>SURFACE Show labels for contours on the surface of the model.<br>TRACK Show labels for trajectories.<br>YES Show default labels. |

## Notes

The **WINDOW** command can be used to hide or show again parts of the display which exist. The toolbuttons toggle the visibility of each item. They can be applied to:

- the **SOLID** view of the model.
- the **OUTLINE** view of the model.
- **VECTORS** on the surface of the model.
- **CONTOURMAPs** and **VECTORMAPs**.
- **ISOSURFACEs**.
- **TRACKS** and flux tubes.
- contour, trajectory and iso-valued surface **LABELS**.

One set of labels for contour, iso-valued surface or trajectory colours can be displayed. If more than one set of labels is available, commands such as **WINDOW LABEL=SURFACE** can be used to display a particular labels set. In this case, for contours on the surface of the model will be shown. Alternatively, **WINDOW LABEL=CYCLE** can be used to cycle through the available label sets. This form of the command can be issued by clicking on the icon.

- the coordinate **AXES**. The icon issues **WINDOW AXES=CYCLE** to cycle around
  - **ALL** axes, major and triads;

- **MAJOR** axes of active coordinate system;
  - **TRIADS** at the origins of global and local coordinate systems;
  - **NONE**, i.e. all axes hidden.
-  A similar function is provided for conductors by [The CONDUCTOR Command \[page 727\]](#) if they have not already been hidden with `window solid=no outline=no`.

## System Variables in the Post-Processor

The following tables are lists of [System Variables \[page 854\]](#) and [String Variables \[page 862\]](#) which are defined and updated by the program. As described in an earlier section (see [System Variables \[page 657\]](#)), more system variables are defined to represent fields and potentials when databases are loaded; those names are not listed here.

Many of the string variables (e.g., `VF_ACTIVATE_ANG`) are defined so that the GUI dialog boxes can present the correct view of the current settings.

| System Variables                                                        |                                                                                  |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Name                                                                    | Usage                                                                            |
| <code>A_n, B_n,</code><br><code>A_m_n, B_m_n</code>                     | Fourier and Legendre coefficients calculated by the <a href="#">FIT</a> command. |
| <code>ACTIVATE_STATUS</code>                                            | The status of the last <a href="#">ACTIVATED</a> simulation.                     |
| <code>ANALYSIS</code>                                                   | Id of the analysis type of the current simulation.                               |
| <code>ANALYSIS_CHARGEDPARTICLE,</code><br><code>ANALYSIS_SCALA</code>   | Unique id for Charged Particle analysis                                          |
| <code>ANALYSIS_CURRENTFLOW,</code><br><code>ANALYSIS_TOSCACURR</code>   | Unique id for Current Flow analysis                                              |
| <code>ANALYSIS_ELECTROSTATIC,</code><br><code>ANALYSIS_TOSCAELEC</code> | Unique id for Electrostatic analysis                                             |
| <code>ANALYSIS_HARMONICEM,</code><br><code>ANALYSIS_ELEKTRASS</code>    | Unique id for Harmonic Electromagnetic analysis                                  |
| <code>ANALYSIS_HARMONICHF,</code><br><code>ANALYSIS_SOPRANOSS</code>    | Unique id for Harmonic High Frequency analysis                                   |
| <code>ANALYSIS_MAGNETIZATION,</code><br><code>ANALYSIS_DEMAG</code>     | Unique id for Magnetization analysis                                             |
| <code>ANALYSIS_MAGNETOSTATIC,</code><br><code>ANALYSIS_TOSCAMAGN</code> | Unique id for Magnetostatic analysis                                             |
| <code>ANALYSIS_MOTIONALEM,</code><br><code>ANALYSIS_CARMEN</code>       | Unique id for Motional Electromagnetic analysis                                  |
| <code>ANALYSIS_MODALHF, ANALYSIS_</code><br><code>SOPRANOEV</code>      | Unique id for Modal High Frequency analysis                                      |
| <code>ANALYSIS_MODALSTRESS,</code><br><code>ANALYSIS_STRESSEV</code>    | Unique id for Modal Stress analysis                                              |
| <code>ANALYSIS_QUENCH</code>                                            | Unique id for Quench analysis                                                    |

| <b>System Variables</b>                                      |                                                                                                                        |
|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                                                  | <b>Usage</b>                                                                                                           |
| <b>ANALYSIS_STATICSTRESS,</b><br><b>ANALYSIS_STRESSST</b>    | Unique id for Static Stress analysis                                                                                   |
| <b>ANALYSIS_STATICTHERMAL,</b><br><b>ANALYSIS_TEMPOST</b>    | Unique id for Static Thermal analysis                                                                                  |
| <b>ANALYSIS_TRANSIENTEM,</b><br><b>ANALYSIS_ELEKTRATR</b>    | Unique id for Transient Electromagnetic analysis                                                                       |
| <b>ANALYSIS_TRANSIENTTHERMAL,</b><br><b>ANALYSIS_TEMPOTR</b> | Unique id for Transient Thermal analysis                                                                               |
| <b>ANALYSIS_VELOCITYEM,</b><br><b>ANALYSIS_ELEKTRAVL</b>     | Unique id for Fixed Velocity Electromagnetic analysis                                                                  |
| <b>AREA</b>                                                  | Cross sectional area of conductor during <b>CONDUCTOR MODIFY</b> .<br>Surface area returned by <b>SURFACE</b> command. |
| <b>BEAMLETPOWER</b>                                          | Power in trajectories calculated by the <b>VIEW</b> command.                                                           |
| <b>BOLTZMANN</b>                                             | Boltzmann constant in J/K.                                                                                             |
| <b>C</b>                                                     | Speed of light in m/s.                                                                                                 |
| <b>CASE</b>                                                  | Simulation number set by the <b>ACTIVATE</b> or <b>SIMULATION</b> commands.                                            |
| <b>CASES</b>                                                 | Total number of simulations in database set by the <b>ACTIVATE</b> command.                                            |
| <b>CHANGETOL</b>                                             | The tolerance used by the solver for acceptance of non-linear convergence by change.                                   |
| <b>CHARGE</b>                                                | Charge density.                                                                                                        |
| <b>CHARU</b>                                                 | Unit factor for charge density.                                                                                        |
| <b>COENERGY</b>                                              | Co-energy calculated by the <b>ENERGY</b> command.                                                                     |
| <b>COL<math>n</math></b>                                     | Column values ( $n=1,\dots,20$ ) in <b>GRAPH</b> command.                                                              |
| <b>CONDU</b>                                                 | Unit factor for conductivity.                                                                                          |
| <b>CONVERTFROMunit</b>                                       | Multiplying factor to convert from <i>unit</i> to the corresponding unit in the current unit set.                      |
| <b>CONVERTTOunit</b>                                         | Multiplying factor to convert to <i>unit</i> from the corresponding unit in the current unit set.                      |
| <b>CONDUCTORS</b>                                            | Number of conductors.                                                                                                  |
| <b>COST</b>                                                  | In AC models, $\cos(\omega t)$                                                                                         |

| <b>System Variables</b> |                                                                                  |
|-------------------------|----------------------------------------------------------------------------------|
| <b>Name</b>             | <b>Usage</b>                                                                     |
| CPSECOND                | The number of processor seconds since the start of the job.                      |
| CURDU                   | Unit factor for current density.                                                 |
| CURRENT                 | Current in trajectories calculated by the <b>VIEW</b> command.                   |
| DENSITY                 | Linear current density in a particle beam calculated by the <b>VIEW</b> command. |
| DAY                     | The day of the month.                                                            |
| DISPU                   | Unit factor for displacement current.                                            |
| ELECENER                | Energy in electric field calculated by the <b>ENERGY</b> command.                |
| ELECTRONCHARGE          | Charge of electron, elementary charge in C.                                      |
| ELECTRONENERGY          | Rest mass energy of electron in eV.                                              |
| ELECTRONMASS            | Rest mass of electron in kg.                                                     |
| ELECU                   | Unit factor for electric field strength.                                         |
| ELEMENT                 | Element number.                                                                  |
| ELEMENTVOLUME           | The volume of the element.                                                       |
| ENERGY                  | Energy in magnetic field calculated by the <b>ENERGY</b> command.                |
| ENERU                   | Unit factor for energy.                                                          |
| EPOTU                   | Unit factor for electric potential.                                              |
| EPSILON0                | Permittivity of free space in F/m.                                               |
| FIELU                   | Unit factor for magnetic field strength.                                         |
| FILEEXISTS              | Set by <b>\$ EXIST</b> command:<br>1 file exists<br>0 file does not exist        |
| FILETYPE                | Set by <b>\$ EXIST</b> command:<br>1 file<br>2 folder or directory               |
| FLUX                    | Flux linked per turn calculated by the <b>FLUXLINKAGE</b> command.               |
| FLUXU                   | Unit factor for magnetic flux density                                            |
| FMOD                    | $SQRT(FX*FX+FY*FY+FZ*FZ)$                                                        |
| FORCU                   | Unit factor for force.                                                           |

| <b>System Variables</b> |                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>             | <b>Usage</b>                                                                                                     |
| FREQ, FREQUENCY         | Frequency.                                                                                                       |
| FX, FY, FZ              | Force calculated by the BODY, ENERGY and INTEGRATE commands.                                                     |
| GRAVITY                 | The acceleration due to gravity in m/s <sup>2</sup> .                                                            |
| HCMOD                   | SQRT(HCX*HCX+HCY*HCY+HCZ*HCZ)                                                                                    |
| HCX, HCY, HCZ           | Coercive force.                                                                                                  |
| HCR                     | (HCX*X+HCY*Y)/R                                                                                                  |
| HCT                     | (HCY*X-HCX*Y)/R                                                                                                  |
| HOUR                    | The number of whole hours since midnight.                                                                        |
| HYSPOWER                | Hysteretic power calculated by the ENERGY command.                                                               |
| IJCX, IJCY, IJCZ        | Source current density.                                                                                          |
| INTEGRAL                | Result of integration.                                                                                           |
| JBEAM                   | Current density in a particle beam calculated by the VIEW command.                                               |
| JCMOD                   | SQRT(JCX*JCX+JCY*JCY+JCZ*JCZ)                                                                                    |
| JCR                     | (JCX*X+JCY*Y)/R                                                                                                  |
| JCT                     | (JCY*X-JCX*Y)/R                                                                                                  |
| JCX0                    | SQRT(RJCX^2+IJCX^2)                                                                                              |
| JCX                     | RJCX*COST+IJCX*SINT                                                                                              |
| JCXP                    | ATAN2D(IJCX;RJCX)                                                                                                |
| JCY0                    | SQRT(RJCY^2+IJCY^2)                                                                                              |
| JCY                     | RJCY*COST+IJCY*SINT                                                                                              |
| JCYP                    | ATAN2D(IJCY;RJCY)                                                                                                |
| JCZ0                    | SQRT(RJCZ^2+IJCZ^2)                                                                                              |
| JCZ                     | RJCZ*COST+IJCZ*SINT                                                                                              |
| JCZP                    | ATAN2D(IJCZ;RJCZ)                                                                                                |
| JDAY                    | The Julian day number (e.g. 1 for 1 <sup>st</sup> January and 366 for 31 <sup>st</sup> December in a leap year). |
| LENGU                   | Unit factor for length.                                                                                          |

| System Variables                                                                                                         |                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                                                                                                                     | Usage                                                                                                                                                                                         |
| <i>line_INTEGRAL</i> , <i>line_MAXIMUM</i> ,<br><i>line_MINIMUM</i> , <i>line_XATMAXIMUM</i> ,<br><i>line_XATMINIMUM</i> | The integral, extreme values and positions at the extreme values of a graph <i>line</i> created by the <b>DATALINE</b> command.                                                               |
| <i>line_Pn_i</i>                                                                                                         | Coefficients of a polynomial of order <i>n</i> fitted to a graph <i>line</i> by the <b>PROCESSLINE</b> command ( <i>i</i> = 0, 1, ..., <i>n</i> ).                                            |
| M                                                                                                                        | Particle mass in electron units calculated by the <b>VIEW</b> command.                                                                                                                        |
| MASSU                                                                                                                    | Unit factor for mass.                                                                                                                                                                         |
| MAXCHANGE                                                                                                                | The largest absolute change over the last nonlinear iteration in the solver.                                                                                                                  |
| MAXIMUM                                                                                                                  | Maximum <b>COMPONENT</b> value in <b>MAP</b> , <b>PLOT</b> , <b>THREED</b> and <b>VOLUME</b> .                                                                                                |
| MINIMUM                                                                                                                  | Minimum <b>COMPONENT</b> value in <b>MAP</b> , <b>PLOT</b> , <b>THREED</b> and <b>VOLUME</b> .                                                                                                |
| MINUTE                                                                                                                   | The number of whole minutes since the start of the hour.                                                                                                                                      |
| MODELLER                                                                                                                 | 2, see <b>PROGRAM</b> [page 859].                                                                                                                                                             |
| MONTH                                                                                                                    | The month number.                                                                                                                                                                             |
| MU0                                                                                                                      | Permeability of free space in H/m.                                                                                                                                                            |
| NODE                                                                                                                     | Node number.                                                                                                                                                                                  |
| NX, NY, NZ                                                                                                               | Normal unit vector. On surfaces of the model this is the outward normal to the selected material surface. On field point patches, the direction depends on the ordering of the corner points. |
| OPENSTREAM                                                                                                               | The number of last stream opened by <b>\$ OPEN</b> .                                                                                                                                          |
| OPERA2DPP                                                                                                                | 4, see <b>PROGRAM</b> [page 859].                                                                                                                                                             |
| PBEAM                                                                                                                    | Power density in a particle beam calculated by the <b>VIEW</b> command.                                                                                                                       |
| PI                                                                                                                       | $\pi$                                                                                                                                                                                         |
| PLANCK                                                                                                                   | Planck constant in J s.                                                                                                                                                                       |
| POSTPROCESSOR                                                                                                            | 3, see <b>PROGRAM</b> [page 859].                                                                                                                                                             |
| POWER                                                                                                                    | Power calculated by the <b>ENERGY</b> command                                                                                                                                                 |
| POWEU                                                                                                                    | Unit factor for power.                                                                                                                                                                        |
| PRESU                                                                                                                    | Unit factor for pressure.                                                                                                                                                                     |

| System Variables              |                                                                                                                                                                                                 |                                                  |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Name                          | Usage                                                                                                                                                                                           |                                                  |
| PREPROCESSOR                  | 1, see <a href="#">PROGRAM [page 859]</a> .                                                                                                                                                     |                                                  |
| PROGRAM                       | 3<br>In a command input file, <b>PROGRAM</b> can be compared against <b>MODELLER</b> , <b>OPERA2DPP</b> , <b>PREPROCESSOR</b> and <b>POSTPROCESSOR</b> to determine which program is being run. |                                                  |
| PROTONMASS                    | Proton rest mass in kg.                                                                                                                                                                         |                                                  |
| PROTONMASSRATIO               | Proton to electron mass ratio.                                                                                                                                                                  |                                                  |
| Q                             | Charge on a particle calculated by the <b>VIEW</b> command.                                                                                                                                     |                                                  |
| R                             | $\text{SQRT}(X^2+Y^2)$                                                                                                                                                                          |                                                  |
| READVALUE00                   | The number of items on a line read from a file by <b>\$ READ</b> .                                                                                                                              |                                                  |
| READVALUE01, READVALUE02, ... | The values of items read from a file by <b>\$ READ</b> .                                                                                                                                        |                                                  |
| REFXY                         | 0                                                                                                                                                                                               | Model reflection in XY plane is <b>NO</b> .      |
|                               | 1                                                                                                                                                                                               | Model reflection in XY plane is <b>YES</b> .     |
|                               | -1                                                                                                                                                                                              | Model reflection in XY plane is <b>INVERSE</b> . |
| REFYZ                         | 0                                                                                                                                                                                               | Model reflection in YZ plane is <b>NO</b> .      |
|                               | 1                                                                                                                                                                                               | Model reflection in YZ plane is <b>YES</b> .     |
|                               | -1                                                                                                                                                                                              | Model reflection in YZ plane is <b>INVERSE</b> . |
| REFZX                         | 0                                                                                                                                                                                               | Model reflection in ZX plane is <b>NO</b> .      |
|                               | 1                                                                                                                                                                                               | Model reflection in ZX plane is <b>YES</b> .     |
|                               | -1                                                                                                                                                                                              | Model reflection in ZX plane is <b>INVERSE</b> . |
| RESIDUAL                      | The remaining residual after the last nonlinear iteration in the solver.                                                                                                                        |                                                  |
| RESIDUALTOL                   | The tolerance for acceptance of nonlinear convergence by residual in the solver.                                                                                                                |                                                  |
| RJX, RJY, RJZ                 | Source current density.                                                                                                                                                                         |                                                  |

| <b>System Variables</b>             |                                                                                                                     |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                         | <b>Usage</b>                                                                                                        |
| RMSCHANGE                           | The RMS change over the last nonlinear iteration in the solver.                                                     |
| ROTL11, ... ROTL33                  | Field point local coordinate system rotation matrix.                                                                |
| ROTATIONS                           | The signed number of rotational copies of the model set by the <b>LOAD</b> and <b>SYMMETRY</b> command.             |
| ROW                                 | Row number in <b>GRAPH</b> command.                                                                                 |
| RSTART                              | Initial radial coordinate of trajectory set by the <b>VIEW</b> command.                                             |
| SCALU                               | Unit factor of magnetic scalar potential.                                                                           |
| xx_SHIFTX, xx_SHIFTY, xx_SHIFTZ     | Position of a part of a Motional EM simulation, where <b>xx</b> is the first two characters of the part group name. |
| SECOND                              | The number of seconds since the start of the minute (to nearest millisecond).                                       |
| SINT                                | In AC models, $\sin(\omega t)$                                                                                      |
| SLICETHICK                          | The thickness of a 2d-slice model.                                                                                  |
| STEFANBOLTZMANN<br>STEPHANBOLTZMANN | Stefan-Boltzmann constant in W/(m <sup>2</sup> K <sup>4</sup> ).                                                    |
| TH                                  | <b>ATAN2D(Y;X)</b>                                                                                                  |
| xx_THETAZ                           | Rotor angle of a Motional EM simulation, where <b>xx</b> is the first two characters of the rotor group name.       |
| TIME, TTIME                         | Transient time.                                                                                                     |
| TOF                                 | Time of flight of a particle calculated by the <b>VIEW</b> command.                                                 |
| TORQMOD                             | <b>SQRT(TORQX*TORQX+TORQY*TORQY+TORQZ*TORQZ)</b>                                                                    |
| TORQX, TORQY, TORQZ                 | Torque calculated by the <b>BODY</b> and <b>INTEGRATE</b> commands.                                                 |
| TOTALCURRENT                        | Total current crossing an intersection patch calculated by the <b>VIEW</b> command.                                 |
| TX, TY, TZ                          | Tangential unit vector.                                                                                             |
| TXBEAM, TYBEAM, TZBEAM              | Tangential unit vector to trajectory calculated by the <b>VIEW</b> command.                                         |
| VAL1, VAL2                          | Values from <b>FILE1</b> and <b>FILE2</b> in <b>ARITHMETIC</b> command.                                             |

| <b>System Variables</b>                                                             |                                                                                                                                                 |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                                                                         | <b>Usage</b>                                                                                                                                    |
| <code>vector_X, vector_Y</code><br><code>vector_R, vector_TH</code>                 | Components ( <code>X, Y</code> ), length ( <code>R</code> ) and angle ( <code>TH</code> ) of a vector created by the <b>DATAVECTOR</b> command. |
| <code>VECTU</code>                                                                  | Unit factor for magnetic vector potential.                                                                                                      |
| <code>VELMOD</code>                                                                 | <code>SQRT(VELX*VELX+VELY*VELY+VELZ*VELZ)</code>                                                                                                |
| <code>VELR</code>                                                                   | <code>(VELX*X+VELY*Y)/R</code>                                                                                                                  |
| <code>VELT</code>                                                                   | <code>(VELY*X-VELX*Y)/R</code>                                                                                                                  |
| <code>VELX, VELY, VELZ</code>                                                       | Velocity.                                                                                                                                       |
| <code>VERSION</code>                                                                | Version of software as a number.                                                                                                                |
| <code>VOLUME</code>                                                                 | Volume of integration calculated by the <b>ENERGY</b> and <b>VOLUME</b> commands.                                                               |
| <code>VR_n</code>                                                                   | Resistive voltage drop in conductor <i>n</i> calculated by the <b>BODY</b> command.                                                             |
| <code>X, Y, Z</code>                                                                | Field point coordinates.                                                                                                                        |
| <code>X0BEAM, Y0BEAM, Z0BEAM</code>                                                 | Coordinates at centre of beam calculated by the <b>VIEW</b> command.                                                                            |
| <code>XATMINIMUM, XATMAXIMUM</code>                                                 | Locations of extreme values in <b>GRAPH</b> command.                                                                                            |
| <code>XATMINIMUM, YATMINIMUM, ZATMINIMUM, XATMAXIMUM, YATMAXIMUM, ZATMAXIMUM</code> | Locations of extreme values in <b>MAP</b> , <b>PLOT</b> , <b>THREED</b> and <b>VOLUME</b> .                                                     |
| <code>XLOCAL, YLOCAL, ZLOCAL</code>                                                 | Field point local coordinate system origin.                                                                                                     |
| <code>XSTART, YSTART, ZSTART</code>                                                 | Coordinates of first point on a trajectory calculated by the <b>VIEW</b> command.                                                               |
| <code>YEAR</code>                                                                   | The year number.                                                                                                                                |
| <code>Z0FREE</code>                                                                 | Impedance of free space, <b>MU0*C</b> in $\Omega$ .                                                                                             |

Where the description of a variable refers to a command, the documentation of that command should be read for more information about the variable.

| <b>String Variables</b> |                                                                                                                                                                                                                                                                     |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>             | <b>Usage</b>                                                                                                                                                                                                                                                        |
| ANALYSIS                | Physics name of the current solver:<br><b>CHARGEDPARTICLE, CURRENTFLOW, ELECTROSTATIC, HARMONICEM, HARMONICHF, MAGNETIZATION, MAGNETOSTATIC, MODALHF, MODALSTRESS, MOTIONALEM, QUENCH, STATICSTRESS, STATICTHERMAL, TRANSIENTEM, TRANSIENTTHERMAL or VELOCITYEM</b> |
| ANALYSIS_PROGRAM        | Old name of the current solver (in the same order as above):<br><b>SCALA, TOSCACURR, TOSCAELEC, ELEKTRASS, SOPRANOSS, DEMAG, TOSCAMAGN, SOPRANOEV, STRESSEV, CARMEN, QUENCH, STRESSST, TEMPOST, ELEKTRATR, TEMPOTR or ELEKTRAVL</b>                                 |
| COMIBASENAME            | The name of a command input file while it is open (without the <b>.comi</b> extension).                                                                                                                                                                             |
| COMIPATH                | The name of the folder containing a command input file, while it is open.                                                                                                                                                                                           |
| CHARU                   | Symbolic name for charge density unit.                                                                                                                                                                                                                              |
| CONDU                   | Symbolic name for conductivity unit.                                                                                                                                                                                                                                |
| CURDU                   | Symbolic name for current density unit.                                                                                                                                                                                                                             |
| DISPU                   | Symbolic name for displacement current unit.                                                                                                                                                                                                                        |
| ELECU                   | Symbolic name for electric field strength unit.                                                                                                                                                                                                                     |
| ENERU                   | Symbolic name for energy unit.                                                                                                                                                                                                                                      |
| EPOTU                   | Symbolic name for electric potential unit.                                                                                                                                                                                                                          |
| FIELU                   | Symbolic name for magnetic field strength unit.                                                                                                                                                                                                                     |
| FLUXU                   | Symbolic name for magnetic flux density unit.                                                                                                                                                                                                                       |
| FORCU                   | Symbolic name for force unit.                                                                                                                                                                                                                                       |
| LENGU                   | Symbolic name for length unit.                                                                                                                                                                                                                                      |
| LOADFILENAME            | The name of the last file <b>LOADED</b> .                                                                                                                                                                                                                           |
| MASSU                   | Symbolic name for mass unit.                                                                                                                                                                                                                                        |
| NOW                     | Time in <b>hh:mm:ss</b> format.                                                                                                                                                                                                                                     |
| POWEU                   | Symbolic name for power unit.                                                                                                                                                                                                                                       |

| <b>String Variables</b>         |                                                                                                                                                                                                |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                     | <b>Usage</b>                                                                                                                                                                                   |
| PRESU                           | Symbolic name for pressure unit.                                                                                                                                                               |
| PROJECTFOLDER                   | The project folder.                                                                                                                                                                            |
| SCALU                           | Symbolic name for magnetic scalar potential unit.                                                                                                                                              |
| READSTRING00                    | The complete line read from a file by \$ READ.                                                                                                                                                 |
| READSTRING01, READSTRING02, ... | The items read from a file by \$ READ.                                                                                                                                                         |
| TITLE                           | The first line of the database title.                                                                                                                                                          |
| TODAY                           | Date in dd/mmm/yyyy format.                                                                                                                                                                    |
| VECTU                           | Symbolic name for magnetic vector potential unit.                                                                                                                                              |
| VERSION                         | Version number as a character string.                                                                                                                                                          |
| VF_ACTIVATE_ANG                 | Current settings of ACTIVATE parameters THETA, PHI and PSI:<br><br>ACTIVATEXYZ      0, 0, 0<br>ACTIVATEYZX      90, 0, 90<br>ACTIVATEZXY      90, 90, 180<br>ACTIVATEOTHER    any other angles |
| VF_CART_TAB                     | Active tabpane of the CARTESIAN command dialog: XY, YZ, ZX or O.                                                                                                                               |
| VF_CMIM                         | Value of the \$COMINPUT MODE parameter                                                                                                                                                         |
| VF_COND_PICK                    | Is conductor picking enabled?<br><br>STARTPICK      enabled<br>STOPPICK        disabled                                                                                                        |
| VF_CUTANGLES                    | Current settings of SELECT parameters THETA and PHI:<br><br>CUTXY      0, 0<br>CUTYZ      90, 0<br>CUTZX      90, 90<br>CUTOTHER    any other angles                                           |
| VF_GLOBALVECXYZ                 | Holds the vector name in MAP and THREED, if VX, VY and VZ are x, y, and z components of the same vector.                                                                                       |

| <b>String Variables</b> |                                                                                                                    |
|-------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>             | <b>Usage</b>                                                                                                       |
| VF_LCS1_ANGLES          | Current settings of <b>CONDUCTOR</b> parameters <b>THETA1</b> , <b>PHI1</b> and <b>PSI1</b> :                      |
|                         | LCS1XYZ   0, 0, 0                                                                                                  |
|                         | LCS1YZX   90, 0, 90                                                                                                |
|                         | LCS1ZXY   90, 90, 180                                                                                              |
|                         | LCS1OTHER   any other angles                                                                                       |
| VF_LCS2_ANGLES          | Current settings of <b>CONDUCTOR</b> parameters <b>THETA2</b> , <b>PHI2</b> and <b>PSI2</b> :                      |
|                         | LCS2XYZ   0, 0, 0                                                                                                  |
|                         | LCS2YZX   90, 0, 90                                                                                                |
|                         | LCS2ZXY   90, 90, 180                                                                                              |
|                         | LCS2OTHER   any other angles                                                                                       |
| VF_LINE_TAB             | Active tabpane of the <b>LINE</b> command dialog: <b>X</b> , <b>Y</b> , <b>Z</b> or <b>O</b> .                     |
| VF_MAP_VNNN             | Are the <b>MAP</b> parameters <b>VX</b> , <b>VY</b> and <b>VZ</b> the x, y and z components of the same vector?    |
|                         | ONE   same vector                                                                                                  |
|                         | THREE   different vectors                                                                                          |
| VF_POLAR_TAB            | Active tabpane of the <b>POLAR</b> command dialog: <b>CYL</b> , <b>DISK</b> or <b>O</b> .                          |
| VF_SET_ANGLES           | Current settings of <b>SET</b> parameters <b>TLOCAL</b> , <b>PLOCAL</b> and <b>SLOCAL</b> :                        |
|                         | SETEXYZ   0, 0, 0                                                                                                  |
|                         | SETYZX   90, 0, 90                                                                                                 |
|                         | SETZXY   90, 90, 180                                                                                               |
|                         | SETOOTHER   any other angles                                                                                       |
| VF_THREED_VNNN          | Are the <b>THREED</b> parameters <b>VX</b> , <b>VY</b> and <b>VZ</b> the x, y and z components of the same vector? |
|                         | ONE   same vector                                                                                                  |
|                         | THREE   different vectors                                                                                          |

| <b>String Variables</b> |                                                                                                                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>             | <b>Usage</b>                                                                                                                                                                                    |
| VF_TRACK_ANGLES         | Current settings of <b>TRACK</b> parameters <b>THETA</b> , <b>PHI</b> and <b>PSI</b> :<br>TRACKXYZ   0, 0, 0<br>TRACKYZX   90, 0, 90<br>TRACKZXY   90, 90, 180<br>TRACKOTHER   any other angles |
| VF_TRACK_PARTICLE       | Current settings of <b>TRACK</b> parameters <b>MASS</b> and <b>CHARGES</b> :<br>ELECTRON   1, -1<br>PROTON   1838.65, 1<br>OTHER   any other values                                             |
| VF_VIEW_INTE_AX         | Current settings of <b>VIEW</b> parameters <b>XAXIS</b> and <b>YAXIS</b> :<br>XINTE   X, DENSITY<br>YINTE   Y, DENSITY<br>RINTE   R, DENSITY<br>OINTE   any other values                        |
| VF_VIEW_TRAC_AX         | Current settings of <b>VIEW</b> parameters <b>XAXIS</b> and <b>YAXIS</b> :<br>XTRAC   X, Y<br>RTRAC   R, Z<br>OTRAC   any other values                                                          |
| VL_n                    | Inductive voltage drop in conductor <i>n</i> calculated by the <b>BODY</b> command.                                                                                                             |
| WORKINGFOLDER           | The working folder.                                                                                                                                                                             |
| YESORNO                 | Pre-answer "yes-no" questions with <b>YES</b> or <b>NO</b> .                                                                                                                                    |

# **Chapter 8**

# **Python and Opera**

## **Introduction**

---

This chapter expands on the introduction to Python and Opera earlier on in this document ("Opera Python" on page 65).

### **Embedded Python**

An embedded Python 3.4.3 interpreter is constructed when an Opera program is started. This may be accessed from:

- any comi interface, such as the console or a command file;
- a hook callback registered to a hook point in a Solver simulation ("Integrating Python Functions with Opera" on page 886); or
- a registered calculation function callback in a user variable expression.

The first of these is achieved using the `$ PYTHON` command ("Opera Python" on page 65), for example the code below sets the value of `myvar` (in the Python memory) to 3:

```
$PYTHON COMMAND='myvar = 3'
```

The second and third require a Python function to be registered to a specific hook point or calculation number ("Integrating Python Functions with Opera" on page 886). Once registered, when Opera triggers such points, the Python interpreter is invoked to execute registered Python functions.

In all cases, the invocation of the interpreter will trigger the requested Operation to be performed in Python, and control will be returned to the calling Opera environment once this operation is completed. If the requested operation is a Python function, or is provided as multiple lines separated by an appropriate line break (typically `\n`), the entire function (or all the provided lines) will be executed before control is returned. Consider the following commands:

```
$CONSTANT #ALPHA 3.0
$PYTHON COMMAND=def fun1(a,b):\n    return a+b\n$CONSTANT #BETA 4.0
$PYTHON COMMAND=c=fun1(1,2)
$CONSTANT #DELTA 5.0
```

1. First the Opera variable **#ALPHA** is assigned a value.
2. Next the Python interpreter is invoked and the function **fun1 (a,b)** is defined by the execution of the three lines of Python code, separated by the line break '**\n**'.
3. Once defined, control is returned to Opera, and the variable **#BETA** is assigned a value.
4. The Python interpreter is again invoked, and the Python variable **c** is assigned the value returned by the function **fun1**.
5. Once the value of **c** is assigned, control again returns to the Opera command processor.

The **operafea** module provides a means to interact between Python and Opera, for more information "[The operafea Embedded Python Module](#)" on page 872. If any Opera commands are executed from Python, they are executed immediately. For example consider the Python function,

```
import operafea

def fun1():
    operafea.addConstantUsrVar("#A",str(2),
        "value of A")
    a = operafea.getUsrVar("#A")
    print("a = ",a)
```

When run from the following comi script:

```
$CONSTANT #A 3.0
$PYTHON COMMAND=fun1()
```

The output is,

```
a = 2.0
```

## Accessing Documentation

Most Python packages, particularly those supplied by Opera contain documentation for their contents. There are many ways to access this documentation, although the simplest is often to invoke the Python "help" command, for example.

```
$PYTHON COMMAND=help (operafea)
```

An HTML version may also be produced using Python's **pydoc** module. The following is a method for generating this within Opera:

```
/ Import pydoc
$python command=import pydoc

/ Create html for operafea
$python command=pydoc.writedoc(operafea)

/ Similarly create html for any other module
/ for example numpy
```

```
$python command=import numpy
$python command=pydoc.writedoc(numpy)
```

## Memory Scope

There are several distinct memory regions which are important with regards to Opera Python. Each of these is completely separate, meaning changes to variables in one memory are not directly available in another, and should a variable be imported into one memory from another, any changes to such a variable in the new memory will not affect the variable's value in the original memory. The `operafea` module can be used to pass data between the distinct regions. The memory regions are:

- Python memory;
- Command language variable memory ('#' variables); and
- Opera memory.

Python memory is the memory accessible from the Python interpreter, and is where variables initialised in Python exist. It follows the rules for variable scoping as defined for Python. It is maintained throughout the execution of a program. It is not saved between sessions or saved to databases.

Command language variable memory is the memory directly accessible from a comi interface. This is where user variables such as those initialised by the `$CONSTANT` command exist. Such variables may be retained in a database between programs, depending on program settings.

Opera memory is a memory store for items created via the Opera interface. Variables inside this memory are accessible from both Opera and Python, although this is done through copying the variable as opposed to directly referencing it.

## NumPy and SciPy

The `NumPy` 1.9.2 and `SciPy` 0.15.1 packages are provided in Opera Python. These packages are built with the Intel MKL library which provides optimized compilations of BLAS and LAPACK routines.

## Installing Third Party Packages using pip

There are a large number of third party Python packages available that can be installed to be used within Opera. This can be achieved using `pip` from within any of the Opera programs.

The first step is to import `pip`. Having imported `pip` it can be used to install the required package.

Normally `pip` would be run from a command line with arguments, but when running from within an application like Opera it is simply a case of parsing the arguments directly to `pip.main()`.

In this example the `pyparsing` package is being installed.

```
$python import pip
$python pip.main(["install","--user","pyparsing"])
```

In the arguments list the "`--user`" argument is telling `pip` to install the package in a location with user write privileges, rather than in the default Opera Python installation folder as on most systems this is not in a user writeable location.

Once the package has been installed it is necessary to close and re-open the Opera session for the embedded Python interpreter to be initialised again so that it is aware of the new package. Having done this the package can be imported into Python in the normal way.

```
$python import pyparsing
```

## Useful resources

The following provide information on the use of Python in general:

[www.python.org](http://www.python.org)

[www.numpy.org](http://www.numpy.org)

[www.scipy.org](http://www.scipy.org)

## Limitations

The embedded Python interpreter has the following limitations in comparison to a standalone Python interpreter.

1. Standard Python debugging modules may not currently operate as expected.
2. Keyboard input requests are not implemented.

## Parallel

The embedded Python interpreter in Opera operates as a single instance per program. In effect, this means that when running Opera in parallel, only one thread may use Python at any given time, and all threads which access Python will have access to the same Python environment, and specifically the same memory. The invocation and execution of the Python interpreter instance is thread-safe, and as such there is no restriction on running Opera in parallel with Python.

However, the serial nature of Python execution may have an impact on parallel performance, particularly if the actions carried out in Python are computationally expensive in comparison to the non-Python (i.e. Opera solver) actions carried out by Opera in the same parallel region. As such, it is recommended to actively unregister Python functions when they are not needed, ("Solver Hooks" on page 887).

It is also possible for the Python instance itself to create parallel regions. This is not recommended however, as this may have unforeseen consequences on the performance of the surrounding Opera environment, which may itself be running in a parallel region. This is particularly true for Hook and Calculation callbacks.

## Usage of NumPy in Opera Python

Opera Python uses **NumPy** ([www.numpy.org](http://www.numpy.org)). This primarily allows for vector manipulation within the Opera Python environment, particularly in the context of data transfer between an Opera database and Python. Example usage for some common scenarios can be found in [The operafea Embedded Python Module \[page 872\]](#).

In these examples **NumPy** is imported with the alias **np**, from which **NumPy** functions can be accessed:

```
import numpy as np  
a = np.array([1,2,3])
```

**NumPy** also contains implementations of many mathematical functions, including basic arithmetic, matrix arithmetic, trigonometric functions and fast Fourier transforms, which operate over **NumPy** arrays. For a full description of **NumPy** functionality, see [www.numpy.org](http://www.numpy.org). Included here is a brief description of some of the key features, most notably the in-built piece-wise manipulation of array data.

Consider the **NumPy** arrays,

```
a = np.array([1,2,3])  
b = np.array([4,5,6])
```

Arithmetic Operations Operate over elements, in that the commands,

```
aplusb = a+b  
atimesb = a*b
```

create the results equivalent to ,

```
aplusb = np.array([5,7,9])  
atimesb = np.array([4,10,18])
```

To perform the dot product ( $a \cdot b$ ), the **NumPy** **dot** function may be called,

```
dotprodab = np.dot(a,b)  
#giving dotprodab = 32
```

Trigonometric functions act on entire arrays unless otherwise instructed,

```
sina = np.sin(a)  
#giving sina = [ 0.84147098 0.90929743 0.14112001]  
sinal = np.sin(a[1])  
#giving sinal = 0.909297426826
```

## Contiguity

NumPy arrays are not necessarily contiguous in memory, meaning that adjacent elements in a NumPy array are not necessarily adjacent in memory. Many NumPy functions can automatically deal with this and therefore many calculations can be performed without concern.

However, only arrays which are contiguous in memory may be added into an Opera database. This may be ensured by creating a temporary copy of the array to be added, with either Fortran or C ordering. For example, to add the double precision array "Arr" to a database with NODAL association<sup>1</sup>:

```
import numpy as np
...
tempArray = np.array(Arr, order = "F" )
simu = operafea.currentSimulation()
simu.addNumPyDblArray("RNODALAR", tempArray, "NODAL")
```

---

<sup>1</sup>For more information on the `operafea` module "The operafea Embedded Python Module" on the facing page

## The `operafea` Embedded Python Module

The `operafea` embedded Python module is supplied to allow access to database and comi variables from the Python interpreter. This includes the class definitions for objects of type `OperaSimu`, `OperaElement`, `OperaNode` ("3d Solution Interface" on page 878) and `OperaObject` ("The `OperaObject` Class" on page 876). The `operafea` module is automatically included at the console command prompt, however it is recommended to always import in any modules for which it is required. This is done using the Python command:

```
import operafea
```

Or equivalently from the comi interface as

```
$PYTHON COMMAND=import operafea
```

Detailed information on the classes and functions provided by the `operafea` module can be accessed using the help command from inside the Python interpreter:

```
help(operafea)
```

This can be accessed from the Opera console using:

```
$PYTHON COMMAND=help (operafea)
```

### Simple Usage

The `operafea` module provides an interface to software internals and allows users to easily extend Opera functionality when needed. The module's functionality is accessible via the `import operafea` Python statement. A complete listing of the functions and classes accessible from the module are summarised below.

Typical module usage in a Python script:

```
import operafea
operafea.output('hi from python')
```

Typical module usage in an Opera comi script:

```
$PYTHON COMMAND='import operafea'
$PYTHON COMMAND='operafea.output('hi from python')'
```

### Accessing and Updating Opera Variables

The `operafea` module comes with a number of functions for accessing (getting and setting) variables from Opera. The example below accesses the real and imaginary voltage and currents from a Steady-State Electromagnetic solution and creates Opera user variables for the impedance. In order to allow expressions to be used in variable definitions, the value parameter is passed in as a string.

Note: the example also makes use of Python's inbuilt handling of complex numbers.

```

import operafea
volt=operafea.getSysVar('wl_rv')+operafea.getSysVar('wl_iv')*1j
curr=operafea.getSysVar('wl_ri')+operafea.getSysVar('wl_ii')*1j
res=volt/curr

operafea.addConstantUsrVar('#zr',str(res.real),'Impedance (Real)')
operafea.addConstantUsrVar('#zi',str(res.imag),'Impedance (Imag)')

```

## operafea Module Functions

| Function name                   | Function signature                                            | Return type | Description                                                                              |
|---------------------------------|---------------------------------------------------------------|-------------|------------------------------------------------------------------------------------------|
| <b>Generic Functions</b>        |                                                               |             |                                                                                          |
| <code>command</code>            | <code>(str command)</code>                                    | None        | Execute an Opera command.                                                                |
| <code>output</code>             | <code>(str text)</code>                                       | None        | Print to the Opera output (console/lp/res file).                                         |
| <b>Calculation Functions</b>    |                                                               |             |                                                                                          |
| <code>addConstantUsrVar</code>  | <code>(str name,<br/>str value<br/>[, str desc]<br/>^)</code> | None        | Add a constant user variable into Opera's user variable calculator.                      |
| <code>addModelDimUsrVar</code>  | <code>(str name,<br/>str value [,<br/>str desc])</code>       | None        | Add a model dimension into Opera's user variable calculator.                             |
| <code>addParameterUsrVar</code> | <code>(str name,<br/>str value<br/>[, str desc])</code>       | None        | Add a parameter variable into Opera's user variable calculator.                          |
| <code>addEquationUsrVar</code>  | <code>(str name,<br/>str value<br/>[, str desc])</code>       | None        | Add an equation variable into Opera's user variable calculator.                          |
| <code>getUsrVar</code>          | <code>(str name)</code>                                       | float       | Return the value of the requested user variable from the Opera user variable calculator. |

<sup>1</sup>The square brackets indicate an optional argument

| <b>Function name</b>               | <b>Function signature</b>                               | <b>Return type</b> | <b>Description</b>                                                                         |
|------------------------------------|---------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------|
| <code>getUsrVarExpr</code>         | <code>(str name)</code>                                 | str                | Return the expression of the requested variable from the Opera user variable calculator.   |
| <code>getUsrVarDescr</code>        | <code>(str name)</code>                                 | str                | Return the description of a user variable from the Opera user variable calculator.         |
| <code>varExist</code>              | <code>(str name)</code>                                 | bool               | Check if the named variable exists in the Opera user variable calculator.                  |
| <code>removeUsrVar</code>          | <code>(str name)</code>                                 | None               | Delete the named variable from the Opera user variable calculator.                         |
| <code>getSysVar</code>             | <code>(str name)</code>                                 | float              | Returns the value of the requested system variable.                                        |
| <b>String Functions</b>            |                                                         |                    |                                                                                            |
| <code>addUsrVarStr</code>          | <code>(str name,<br/>str value<br/>[, str desc])</code> | None               | Add a string into the Opera memory with the supplied name, value and optional description. |
| <code>getUsrVarStr</code>          | <code>(str name)</code>                                 | str                | Retrieve a string user variable from Opera.                                                |
| <code>usrVarStrExist</code>        | <code>(str name)</code>                                 | bool               | Check if the named string user variable exists in Opera.                                   |
| <code>removeUsrVarStr</code>       | <code>(str name)</code>                                 | None               | Remove a string user variable from Opera.                                                  |
| <b>Object Management Functions</b> |                                                         |                    |                                                                                            |
| <code>getObject</code>             | <code>(int obj_id)</code>                               | OperaObject        | Return the <code>OperaObject</code> associated to the supplied object id.                  |
| <code>newObject</code>             | <code>(None)</code>                                     | OperaObject        | Create a new object of type <code>OperaObject</code> .                                     |

| Function name                                   | Function signature                                        | Return type              | Description                                                                   |
|-------------------------------------------------|-----------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------|
| <code>newObjectInParent</code>                  | <code>(int par_id)</code>                                 | <code>OperaObject</code> | Create a new <code>OperaObject</code> inside an existing Object (the parent). |
| <code>objectExist</code>                        | <code>(int obj_id)</code>                                 | <code>bool</code>        | Check if the <code>OperaObject</code> with the given id exists.               |
| <code>deleteObject</code>                       | <code>(int obj_id)</code>                                 | <code>None</code>        | Delete the <code>OperaObject</code> with the given id.                        |
| <code>reparent</code>                           | <code>(int obj_id int par_id)</code>                      | <code>None</code>        | Re-assign the parent of the object with the supplied id.                      |
| <code>lastCreatedObject</code>                  | <code>(None)</code>                                       | <code>OperaObject</code> | Return the last created <code>OperaObject</code>                              |
| <b>Callback Management Functions</b>            |                                                           |                          |                                                                               |
| <code>registerCalcCallback</code>               | <code>(int fnc_id, object callable)</code>                | <code>None</code>        | Register a Python function as an Opera Calculation function.                  |
| <code>unregisterCalcCallback</code>             | <code>(int fnc_id)</code>                                 | <code>None</code>        | Unregister a Python function attached to a given calculation function id.     |
| <code>registerHookCallback<sup>1</sup></code>   | <code>(str hook_id, object callable)</code>               | <code>None</code>        | Register a Python function to a specified SOLVER hook point.                  |
| <code>unregisterHookCallback<sup>2</sup></code> | <code>(str hook_id [,object callable])<sup>3</sup></code> | <code>None</code>        | Unregister Python functions attached to a specified Solver hook.              |
| <b>Buffer Management Function<sup>4</sup></b>   |                                                           |                          |                                                                               |

<sup>1</sup>Only available in the Opera-3d/Solvers.

<sup>2</sup>Only available in the Opera-3d/Solvers.

<sup>3</sup>If the optional argument 'callback' is included, only the specified callback function will be unregistered from the provided Solver hook. Otherwise, all functions attached to the specified SOLVER hook will be unregistered.

<sup>4</sup>Only available in Opera-3d/Post.

| Function name                                       | Function signature                           | Return type            | Description                                                                                                      |
|-----------------------------------------------------|----------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>addAsBuffer</code>                            | <code>(OperaObject buff [, str desc])</code> | None                   | Adds in an <code>OperaObject</code> created in Python as a graphing buffer for use in Opera's graphing facility. |
| <b>Simulation Management Function<sup>1,3</sup></b> |                                              |                        |                                                                                                                  |
| <code>currentSimulation</code>                      | <code>(None)</code>                          | <code>OperaSimu</code> | Returns the currently active simulation.                                                                         |

## The `OperaObject` Class

### Description

The purpose of the `OperaObject` class is to be a data container/mediator, carrying data between Opera and the embedded Python interpreter. Opera populates `OperaObject` instances (for example from buffer generation commands), which can be immediately exposed to Python when requested. Equally, `OperaObject` class instances can be instantiated and populated from Python and then can be used as graphing buffers or as data containers for further processing. For more information "["Python Arrays in Post-Processing"](#) on page 891.

### Example Usage

The `OperaObject` class represents a dictionary type object that is capable of storing key-value pairs, where the keys are unique strings, and the values are any of the following types: `int`, `float`, `string`, `NumPy array`, `Python object` or another `OperaObject`. The following code snippets show how to use `OperaObject`:

Import modules, create a new `OperaObject`, and check it is valid:

```
import operafea
import numpy as np
ob1 = operafea.newObject()
if ob1.isValid():
    print('i am valid')
```

Add two integers and a double into the `OperaObject`:

```
ob1.addInt('first', 1 )
ob1.addInt('second', 2 )
ob1.addDouble('myDoubleVal', 1.234 )
```

For convenience it is possible to nest `operaObjects` by assigning one object as the parent of another. In the following snippet, a second `OperaObject` containing a `NumPy` array is created inside the first.

```
ob2=operafea.newObjectInParent(ob1.getId())
array=np.array([2.1,2.2,-5.4])
ob2.addNumPyDblArray('myarr', array)
```

Note: when a `NumPy` array is appended to the object by using the function `addNumPyDblArray`, the array is copied into new storage. If you do not want data duplications please use the `OperaObject.addPyObject` function.

The snippet below uses the alternative `lastCreatedObject` function to get a handle to a newly created `operaObject`. It then adds a Python defined function into the new object and subsequently calls it and prints out the return value:

```
operafea.newObjectInParent(ob2.getId())
ob3=operafea.lastCreatedObject()
def myFunction(a, b):
    return a*b
ob2.addPyObject('pyFnc', myFunction)
myresult = ob2.getValue('pyFnc')(10, 5.5)
print( 'function_result: ', myresult )
#output: "function_result: 55"
```

To clear objects use the `CLEAR OBJECTSTORE=YES` command.

## Class Interface

The `OperaObject` and associated functions are available in all of the Opera applications.

| Function name                 | Function signature                   | Return type | Description                                                                      |
|-------------------------------|--------------------------------------|-------------|----------------------------------------------------------------------------------|
| <code>addDouble</code>        | <code>(str name, float value)</code> | bool        | Add a double precision value with the supplied name to the calling object.       |
| <code>addInt</code>           | <code>(str name, int value)</code>   | bool        | Add an integer value with the supplied name to the calling object.               |
| <code>addNumPyDblArray</code> | <code>(str name, NumPy array)</code> | bool        | Add a double precision NumPy array with the supplied name to the calling object. |

| Function name            | Function signature                  | Return type | Description                                                               |
|--------------------------|-------------------------------------|-------------|---------------------------------------------------------------------------|
| <code>addPyObject</code> | <code>(str name, object obj)</code> | bool        | Add a generic Python object with the supplied name to the calling object. |
| <code>addString</code>   | <code>(str name, str value)</code>  | bool        | Add a string with the supplied name to the calling object.                |
| <code>exist</code>       | <code>(str name)</code>             | bool        | Query the existence of an entity inside the calling object.               |
| <code>getId</code>       | <code>(None)</code>                 | int         | Return the identifier of the calling object.                              |
| <code>getParent</code>   | <code>(None)</code>                 | OperaObject | Return the parent object of the calling object.                           |
| <code>getValue</code>    | <code>(str name)</code>             | object      | Return the value of a component of the calling object.                    |
| <code>isValid</code>     | <code>(None)</code>                 | bool        | Verify the validity of the calling object.                                |
| <code>remove</code>      | <code>(str name)</code>             | bool        | Remove a requested entity from the calling object.                        |

## 3d Solution Interface

### Description

Various classes are provided for accessing and manipulating data within the Opera-3d solutions. The following classes are available in the 3d Solvers and the Post-Processor (but not the Modeller):

- `OperaSimu`: gives the user a handle into the current simulation in order to perform functions such as querying and insertion of solution vectors.
- `OperaMesh`, `OperaElement`, `OperaElementEdge`, `OperaElementFace`, `OperaNode`: for querying mesh data such as the number of elements in a simulation or the location of a particular node.

## OperaSimu Usage

During solution or Post-Processing at various stages it is possible to access the currently active simulation by using the `operafea.currentSimulation` function. The function returns an `OperaSimu` class object which allows:

- Querying of simulation vectors by name; and
- Adding user defined vectors back to the simulation.

Newly appended vectors can be used in further computations, for example in subsequent analyses in a multiphysics solution. The list of available values for a particular solver can be explored in Opera-3d/Post with the command `SYSVARIABLE MODE=DBASE`. For example the command output for the Transient Electromagnetic solver type is as follows:

```
Name Type Simulation Data val
TIMELIST Dbl Array 1 Unclassified length
RELEMENTVOLUME Dbl Array 1 Element No. indexed
RHSX Dbl Array 2 Node No. indexed
RHSY Dbl Array 2 Node No. indexed
RHSZ Dbl Array 2 Node No. indexed
...
...
```

All listed values are available during the solution or post-processing stages and can be obtained using their names. The following example extracts the array of x-directed magnetic strength (`RHSX`), doubles the values and appends the new vector with the name '`MYRHSX`'.

```
import operafea
import numpy as np
# get current simulation
simu = operafea.currentSimulation()
# get simulation vector
rhsx=simu.getNumPyDblArray('RHSX')
# get association (ELEMENT/NODAL)
assoc=simu.getDb1ArrayAssoc('RHSX')
# get units expression
u_expr=simu.getDb1ArrayUnitsExpr('RHSX')
# get units value
u_val=simu.getDb1ArrayUnitsVal('RHSX')
# create new vector
new_arr=rhsx*2.0
# append new vector to the simulation
simu.addNumPyDblArray('MYRHSX', new_arr,
                      assoc, u_expr)
```

Note that the function `getDb1ArrayAssoc` returns a string describing the solution vector type (nodal in this case). When a new array is appended the association type defines how the array is interpreted by the system.

## `OperaMesh`, `OperaElement*` and `OperaNode` Classes

The following classes provide data on elements and nodes in a simulation's mesh:

- `OperaMesh`,
- `OperaElement`,
- `OperaElementEdge`,
- `OperaElementFace` and
- `OperaNode`

They are accessible through two approaches:

### Mesh Access Through `OperaMesh`

The `OperaMesh` class: access to a simulation's mesh is available through the `OperaSimu` class:

```
import operafea
themesh=operafea.currentSimulation().getMesh()
numels=themesh.getElementCount()
for elid in xrange( 1, numels+1 ):
    element=themesh.getElement(elid)
    centroid_coord=element.getCoordAtCentre()
    x=centroid_coord[0]
    y=centroid_coord[1]
    z=centroid_coord[2]
    print( "Coordinates: %f, %f, %f" % (x,y,z) )
```

### Mesh Labels

The elements in an Opera mesh have labels attached to them to store information such as element type, potential type, material label and more. These labels are available through the `getLabelCount()` and `getLabel()` functions:

```
...
# print label information
labelCount = element.getLabelCount()
operafea.output("Labels: %d" % labelCount)
for i in range(labelCount):
    lbl=element.getLabel(i)
    operafea.output('\tLabel %d = %s' % (i,lbl))
```

The output of which is (for example):

```
Labels: 5
    Label 0 = ALL
    Label 1 = TOTAL
    Label 2 = ALUMINIUM
    Label 3 = HEXAHEDRON
    Label 4 = ALL
```

### OperaSimu Class Interface

| Function name                     | Function signature                                                                        | Return type            | Description                                                                           |
|-----------------------------------|-------------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------|
| <code>getId</code>                | <code>(None)</code>                                                                       | int                    | Returns the simulation identifier.                                                    |
| <code>getNumPyDblArray</code>     | <code>(str name)</code>                                                                   | NumPy array            | Returns a copy of the database vector with the supplied name in a NumPy array.        |
| <code>getDblArrayAssoc</code>     | <code>(str name)</code>                                                                   | str (ELEMENT or NODAL) | Returns the array association of the array with the specified name.                   |
| <code>getDblArrayUnitsExpr</code> | <code>(str name)</code>                                                                   | str                    | Returns the units expression for the array with the specified name (e.g. FLUXU).      |
| <code>getDblArrayUnitsVal</code>  | <code>(str name)</code>                                                                   | float                  | Returns the scaling factor currently applicable to the array with the specified name. |
| <code>addNumPyDblArray</code>     | <code>(str name, NumPy array data, str association, [str unit_scaling_expression])</code> | bool                   | Add a one dimensional double valued NumPy array to the simulation.                    |
| <code>exist</code>                | <code>(str name)</code>                                                                   | bool                   | Test for existence of a given field in the simulation.                                |
| <code>remove</code>               | <code>(str name)</code>                                                                   | bool                   | Remove the field with the specified name from the current simulation.                 |
| <code>addFlag</code>              | <code>(str name)</code>                                                                   | None                   | Set the named flag to evaluate as True or False <sup>1</sup>                          |
| <code>getMesh</code>              | <code>(None)</code>                                                                       | OperaMesh              | Returns an object providing access to aspects of the mesh for the current simulation. |

<sup>1</sup>Currently, the only valid flag is 'COMPUTEDATAPERTIMESTEP'

| Function name                     | Function signature                                                           | Return type | Description                                                                                                                                                                |
|-----------------------------------|------------------------------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>getCoordForIndex</code>     | <code>(str association, int index)</code>                                    | NumPy array | Returns the coordinates of the requested node or element centroid as a NumPy array.                                                                                        |
| <code>isValid</code>              | <code>(None)</code>                                                          | bool        | Test whether the simulation is valid.                                                                                                                                      |
| <code>appendOutputTime</code>     | <code>(float time)</code>                                                    | bool        | Add a new output time to the simulation. <sup>1</sup>                                                                                                                      |
| <code>getFieldsAtCoords</code>    | <code>(str name, NumPy array xyz, Numpy array elementids<sup>2</sup>)</code> | OperaObject | Returns the requested nodal database fields ( <code>names</code> ) at the provided set of coordinates ( <code>xyz</code> ). Uses the model unit set for length and fields. |
| <code>getNLConvergenceInfo</code> | <code>(None)</code>                                                          | OperaObject | Returns an Opera Object containing nonlinear convergence information.                                                                                                      |
| <code>stopNLSolver</code>         | <code>(None)</code>                                                          | bool        | Signals to the nonlinear solver to stop at the end of the current iteration.                                                                                               |

### OperaMesh Class Interface

| Function name                | Function signature                    | Return type  | Description                                                                              |
|------------------------------|---------------------------------------|--------------|------------------------------------------------------------------------------------------|
| <code>getElementCount</code> | <code>([str type]<sup>3</sup>)</code> | int          | Returns the total number of elements in the mesh.                                        |
| <code>getNodeCount</code>    | <code>([str type]<sup>1</sup>)</code> | int          | Returns the total number of nodes in the mesh.                                           |
| <code>getElement</code>      | <code>(int id)</code>                 | OperaElement | Returns an OperaElement object representing the element with the supplied element index. |

<sup>1</sup>Only effective in the 3d Solver

<sup>2</sup>The array `elementids` can be used to pass suggested element numbers for some or all of the supplied coordinates.

<sup>3</sup>Type is an optional argument and only of relevance to Motional simulations:

`type = 'STATIC'` will return the number of static elements/nodes.

`type = 'DYNAMIC'` will return the number of dynamic (gap) elements/nodes.

### OperaElement Class Interface

| Function name                  | Function signature         | Return type                   | Description                                                                                  |
|--------------------------------|----------------------------|-------------------------------|----------------------------------------------------------------------------------------------|
| <code>getCoordAtCentre</code>  | ( <code>None</code> )      | NumPy array                   | Returns the coordinates of the element centroid as a NumPy array ([X,Y,Z]).                  |
| <code>getId</code>             | ( <code>None</code> )      | int                           | Returns the index of the element in the mesh.                                                |
| <code>getLabel</code>          | ( <code>int index</code> ) | str                           | Returns the label attached to the specified element with the supplied label number (base 0). |
| <code>getLabelCount</code>     | ( <code>None</code> )      | int                           | Returns the label count associated with the parent element.                                  |
| <code>getNode</code>           | ( <code>int index</code> ) | <code>OperaNode</code>        | Returns the element node with the supplied local node number (base 0).                       |
| <code>getNodeCount</code>      | ( <code>None</code> )      | int                           | Returns the number of nodes of the element.                                                  |
| <code>getEdge</code>           | ( <code>int id</code> )    | <code>OperaElementEdge</code> | Returns the element edge with the supplied local edge <code>id</code> .                      |
| <code>getEdgeCount</code>      | ( <code>None</code> )      | int                           | Returns the number of edges of the element.                                                  |
| <code>getFace</code>           | ( <code>int id</code> )    | <code>OperaElementFace</code> | Returns the element face with the supplied local face <code>id</code> .                      |
| <code>getFaceCount</code>      | ( <code>None</code> )      | int                           | Returns the number of faces of the element.                                                  |
| <code>getNeighbourIndex</code> | ( <code>int id</code> )    | int                           | Returns the neighbour element global index on local face <code>id</code> .                   |

### OperaNode Class Interface

| Function name                | Function signature    | Return type | Description                                                     |
|------------------------------|-----------------------|-------------|-----------------------------------------------------------------|
| <code>getCoord</code>        | ( <code>None</code> ) | NumPy array | Returns the coordinates of the node as a NumPy array ([X,Y,Z]). |
| <code>getElementIndex</code> | ( <code>None</code> ) | int         | Returns the element index of the parent element.                |
| <code>getGlobalIndex</code>  | ( <code>None</code> ) | int         | Returns the global node index of the node.                      |
| <code>getLocalIndex</code>   | ( <code>None</code> ) | int         | Returns the coordinates of the node.                            |

### OperaElementEdge Class Interface

| Function name                | Function signature | Return type | Description                                                |
|------------------------------|--------------------|-------------|------------------------------------------------------------|
| <code>getElementIndex</code> | (None)             | int         | Returns the element index of the parent element.           |
| <code>getGlobalIndex</code>  | (None)             | int         | Returns the global edge index of the edge.                 |
| <code>getNode</code>         | (int id)           | OperaNode   | Returns the edge node with the supplied local node number. |
| <code>getNodeCount</code>    | (None)             | int         | Returns the number of nodes of the edge.                   |

### OperaElementFace Class Interface

| Function name                | Function signature | Return type      | Description                                                |
|------------------------------|--------------------|------------------|------------------------------------------------------------|
| <code>getElementIndex</code> | (None)             | int              | Returns the element index of the parent element.           |
| <code>getGlobalIndex</code>  | (None)             | int              | Returns the global face index of the face.                 |
| <code>getLabel</code>        | (None)             | str              | Returns the label attached to the face.                    |
| <code>getNode</code>         | (int id)           | OperaNode        | Returns the face node with the supplied local node number. |
| <code>getNodeCount</code>    | (None)             | int              | Returns the number of nodes of the face.                   |
| <code>getEdge</code>         | (int id)           | OperaElementEdge | Returns the face edge with the supplied local edge number. |

| Function name                         | Function signature | Return type | Description                                 |
|---------------------------------------|--------------------|-------------|---------------------------------------------|
| <code>getEdgeCount</code>             | (None)             | int         | Returns the number of edges of the face.    |
| <code>getNeighbourElementIndex</code> | (None)             | int         | Returns the neighbour element global index. |

# Integrating Python Functions with Opera

## Purpose

Whilst executing a stand-alone Python function is possible from the console, it may also be desirable to execute Python code during an Opera analysis or processing stage. To facilitate this, Python functions may be attached to "Hooks" in Opera's Solver programs, and to user variable commands as a "Calculation" function.

It is possible to attach Python functions to hooks in a variety of ways:

- **solver.comi**: a command file driven initialization to the solution process;
- **control.comi**: a command file driven process control for transient simulations; or
- from within other Python functions which can be added to any of the solver hooks.

It is necessary to use one of the command file options to attach a first Python function to one of the hooks as an initialization to the process of using Python within the solvers. The following example implementation, which in this case uses **solver.comi**, imports a Python file **my\_hooks.py** to make the functions contained within it available during the simulation. The following line attaches a Python initialization function to the start of the simulation process:

```
/ Import the Python file my_hooks.py
$PYTHON 'import my_hooks as mh'
/ Register function onSimuStart to the on_simu_start hook
$PYTHON 'operafea.registerHookCallback( 'on_simu_start',
mh.onSimuStart )'
```

The Python function onSimuStart then adds additional Python functions to other hooks.

A list of all currently attached callbacks (both Hook and Calculation) is returned using:

```
$PYTHON OPTION=LISTCALLBACKS
```

## Terminology

The following terminology is used here.

- **hookName**: a string containing the name of a solver hook.
- **calcNumber**: an integer identifying a specified Python callback.
- **Callback**: a Python function to be attached to or removed from a solver hook or to be executed via the "Calculation" function interface.
- **PYFNCn**: comi command executing an attached "Calculation" function with between 0 and 6 parameters inclusively. For example **PYFNC3 (1 ,x ,y ,z)** executes the "Calculation" function attached to calcNumber=1, with 3 arguments, in this case the system variables X,Y and Z.

## Calculation functions

During modelling and post processing, user variable commands (**\$PARAMETER**, **\$EQUATION**) may be defined to perform various tasks. Python functions may be used to extend this functionality through the use of the **PYFUNCn** interface. Python functions to be used in this manner need to be attached to an assigned **calcNumber** using the **operafea** command:

```
operafea.registerCalcCallback(calcNumber,callback)
```

For example, the following uses a Python function '**getThermalConductivity**' defined in a Python file "**gtc.py**" in an Opera **\$PARAMETER** command.

```
# gtc.py.
import math

def getThermalConductivity(X, Y, Z):
    if math.sqrt(Z*Z + Y*Y) < 2.0:
        return 100.0
    else:
        return 0.01

operafea.registerCalcCallback(1,getThermalConductivity)
```

Comi command interface,

```
$PYTHON FILE=gtc.py
$PARAMETER NAME=#kappa VALUE=PYFNC3(1;X;Y;Z)
```

Only one Python function may be attached to a single calcNumber. To free a calcNumber for future use, use the command:

```
operafea.unregisterCalcCallback(calcNumber)
```

## Solver Hooks

A Solver simulation runs through various stages of computation, at each stage it may be necessary to perform specific computations via the Python interface. Examples of such a requirement include:

- Monitoring field values during a solution, for example the maximum temperature in a transient thermal solution;
- Averaging losses over multiple time-steps of a transient EM solution to feed a static thermal problem; or
- Monitoring field values during a solution to determine when the system under investigation begins to behave in a time periodic (steady-state) manner.

Python functions can be registered with specific hooks during a solution. Subsequently, when a Solver hook is triggered, all attached Python functions will be automatically executed in the order in which they were attached.

The ability to attach and detach Python functions is provided via the following functions: the first attaches the Python function `callback` to the hook `hookName`, while the second detaches all callbacks hooked to `hookName`, or the individual `callback` if provided.

```
operafea.registerHookCallback(hookName, callback)
operafea.unregisterHookCallback(hookName, [callback])
```

A typical example of this functionality may be found in the User Guide.

## Available Hooks

The available hooks and their corresponding uses are listed in the table below. Functions attached to these hooks should have zero argument interfaces unless otherwise stated. For example:

```
def onTimestepOutput():
    ...
```

This prohibits the use of Python class functions called from class objects, as these pass the calling object as the first argument implicitly.

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>on_solver_start</code>   | This is called when the Solver program is first initialized. Any Python variables which should persist throughout all multiphysics cases may be initialized here. Notably this is called before a simulation is opened, so user variables and other items stored in the database cannot be accessed here.                                                                                                                                                                        |
| <code>on_simu_start</code>     | This is called at the start of each simulation to be solved (e.g. each individual multiphysics stage). Any Python variables that are needed throughout the duration of the specific case may be initialized here. User variables saved into the database (per simulation) are available here.                                                                                                                                                                                    |
| <code>on_timestep_start</code> | The first transient time solver hook. This is called during <code>TIMESTEP_STAGE=START</code> , immediately after the equivalent processing of a Comi Control File. It may be used to control the current and subsequent time-stepping behaviour using time-step control strings as is possible in a control comi file. If access to solution vectors is required for the current time-step then this must be enabled by using the ' <code>COMPUTEDATAPERTIMESTEP</code> ' flag. |

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>on_timestep_end</code>          | The second transient time solver hook. This is called once a time-step (adaptive or simple) has completed, immediately after the equivalent processing of a Comi Control File. The transient control string <code>TIMESTEP_OUTPUT</code> may be used here to dictate whether or not an output time should be prompted (i.e. the current solution written to the database). The <code>TIMESTEP_FINAL</code> control string may be set to <code>YES</code> to terminate the solution (setting to <code>NO</code> will have no effect). |
| <code>on_store_time_output</code>     | The third transient time solver hook. This is called when the database is updated with a new simulation. Primary and secondary database variables may be extracted from the current simulation here. The database may also be updated with fields constructed using Python functionality, such as time-averaged loss vectors. The transient control string <code>TIMESTEP_FINAL</code> may be updated here to force the termination of the transient solver.                                                                         |
| <code>on_nonlinear_iter_start</code>  | The first non-linear iteration hook. This is called at the start of each non-linear iteration.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>on_nonlinear_iter_end</code>    | The second non-linear iteration hook. This is called at the end of each non-linear iteration.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>on_nonlinear_iter_finish</code> | The third non-linear iteration hook. This is called when the non-linear iterations are complete.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>on_simu_end</code>              | This is called once all computations for the current multiphysics case are complete. It should be used to clean up any Python variables that are now redundant, and to detach any functions attached to transient time solver hooks which are no longer required. Opera user variables created here will be saved into the terminating simulation.                                                                                                                                                                                   |
| <code>on_solver_end</code>            | This is called once all computations are complete, before the Solver terminates. It may be used to output information still present in Python memory, and if required to clean up any remaining Python structures.                                                                                                                                                                                                                                                                                                                   |

## Execution Order

The following diagram demonstrates the activation order of the Solver hooks for a transient (time-stepping) solver. For non time-stepping solutions, the `timestep...` hooks are omitted.

- Launch solver
- **solver.comi** processed

- `on_solver_start`
  - Loop unsolved simulations
- `on_simu_start`
  - Start time-stepping
  - `control.comi(TIMESTEP_STAGE=INIT)`
  - `on_timestep_start`
  - Perform 'time-step 0'
    - Perform iterations (also in subsequent time steps)
    - `on_nonlinear_iter_start`
    - `on_nonlinear_iter_end`
    - `on_nonlinear_iter_finish`
  - `on_timestep_end`
  - `on_store_time_output`
  - Loop time-stepping
    - `control.comi(START)`
    - `on_timestep_start`
    - Time-step adaption (if adaptive time-stepping)
      - `control.comi(FULL)`
      - `control.comi(FIRSTHALF)`
      - `control.comi(SECONDHALF)`
      - `control.comi(END)`
    - `on_timestep_end`
    - If `TIMESTEP_OUTPUT==YES`
      - `on_store_time_output`
  - `on_simu_end`
- `on_solver_end`

## Registering Python functions in the solver

The first opportunity to initialize the process of attaching Python to an Opera simulation is provided via a file named **Solver.comi**. This file should be included within the working directory before the start of the simulation process and it will be run automatically. It can be used to import Python classes for later use and (as described above) attach Python functions to hooks within the solvers. Due to the point within the solution process where the file is run, it does not have access to user defined variables or other model specific properties. Instead, it is purely used for the purpose of registering additional functions to be processed within the solution stage.

## Python Arrays in Post-Processing

---

### Opera to Python

Field buffers created by commands which calculate fields along lines or on patches can be exposed to Python. The default setting can be changed using the Opera Manager **Preferences** dialog. Exposure can be switched on and off during a run of the Post-Processor using the **\$PYTHON** command (See "Opera Python" on page 65).

### Graphing Buffers

The buffers generated with graphing or 3d patch related commands are exposed to Python through **OperaObject** type objects. For example it is possible to access graphing buffer content as follows:

```
# create a buffer in Opera
operafea.command( "LOGGEDDATAFILE OPTION=IMPORT
    FILENAME='myfile.log' BUFFER='mybuffer' " )
# get buffer
buffer=operafea.lastCreatedObject()
# get graphing buffer values
X=buffer.getValue("X")
Bx=buffer.getValue("Bx")
By=buffer.getValue("By")
Bz=buffer.getValue("Bz")
# calculate mod
B=np.sqrt(Bx*Bx+By*By+Bz*Bz)
# do integration using numpy built in function
integral = np.trapz(y=B,x=X)
```

### 2d Field Buffers

Graphing buffer's values are stored as 1d **NumPy** arrays. The patch related buffers are stored as 2d **NumPy** arrays. To get values from this buffer firstly the indices need to be obtained. then using these the relevant values can be accessed, for example:

```
# get buffer
buffer = operafea.lastCreatedObject()
# get buffer data
data_arr = buffer.getValue("Spherical")
# get B component indexes
bx_index=buffer.getValue("RBX")
by_index=buffer.getValue("RBY")
bz_index=buffer.getValue("RBZ")
# get arrays of values (subset of 'data_arr')
Bx=data_arr[:,bx_index]
```

```
By=data_arr[:,by_index]
Bz=data_arr[:,bz_index]
# calculate B mod
B=np.sqrt(Bx*Bx+By*By+Bz*Bz)
# find maximum
max_b = B.max()
```

## Python to Opera

The `OperaObject` can be appended as a graphing buffer by using the `operafea.addAsBuffer` function:

```
# create numpy arrays containing random numbers
size = 100
myArray1=numpy.random.ranf(size)
myArray2=numpy.random.ranf(size)
myArray3=numpy.random.ranf(size)
myArray4=numpy.random.ranf(size)
# create new object
newob = operafea.newObject()
# append values to the object
newob.addNumPyDblArray("name1", myArray1)
newob.addNumPyDblArray("name2", myArray2)
newob.addNumPyDblArray("name3", myArray3)
newob.addNumPyDblArray("name4", myArray4)
# add this object as graphing buffer
operafea.addAsBuffer("buffer_from_py", newob)
```

# Index

---

## !

! 27

!! 46

---

## #

#

  free format variables 48

  user variables 56

#BICGSTABL 636

#ITSOLTYPE 636-637

#ITSOLVMAXIT 637

#ITSOLVNOIMPROVE 637

#ITSOLVRETRY 637

#ITSOLVTOL 637

#ITSOLVTOLLOOSE 637

#ITSOLVTOLNR 638

#MAXEDGEHDLPTS 638

#USEFULLMATRIX 639

---

## \$

\$-commands 49

\$ ABORT 46

\$ ABORTCOMI 63

\$ ASK 63

\$ ASKPARAMETER 63

\$ ASSIGN 75

\$ BACKSPACE 75

\$ BREAKERROR 63

\$ CD 80

\$ CLOSE 73

\$ COMINPUT 61

\$ CONSTANT 55, 337

\$ CYCLE 54

\$ DIALOG 67

\$ DISPLAYLINE 65

\$ DO-loops 53

\$ ELIF 52

\$ ELSE 52

\$ END 54

\$ EQUATION 55

\$ ERRORHANDLER 63

\$ EULER 83  
 \$ EXEC 77  
 \$ EXIST 80  
 \$ EXIT 55  
 \$ FILEPROMPT 64  
 \$ FOR-each loops 54  
 \$ FORMAT 74  
 \$ FUNCTION 60  
 \$ GCONSTANT  
     see \$ CONSTANT 55  
 \$ GPARAMETER  
     see \$ PARAMETER 55  
 \$ GROUPBOX 70  
 \$ IF 52  
 \$ LAYOUT 71  
 \$ MODELDIMENSION 55  
 \$ OPEN 73  
 \$ OS 77-78  
 \$ PARAMETER 55, 337  
 \$ PAUSE 63  
 \$ PROJECTFOLDER 25  
 \$ PROMPT 64  
 \$ READ 73  
 \$ SKIP 46  
 \$ STRING 55  
 \$ SYSVAR 60  
 \$ TOOLBUTTON 72  
 \$ WHILE-loops 54  
 \$ WRITE 74

\$VFDIR  
     see VFDIR 35

---

## %

%COMPARE 38  
 %ENV 39  
 %EXISTSTR 40  
 %EXISTVAR 40  
 %EXPR 40  
 %FIND 40  
 %INT 41  
 %NINT 41  
 %REAL 41  
 %SUBSTR 43  
 %VFDIR%  
     see VFDIR 35

---

## ,

,, comma 28

---

## /

/, comment 82

---

**;**

;, semicolon 34

---

**|**

|, command separator 82

---

**'**

', quotation mark 36

---

**2**

20-node brick 160

---

**3**

3d Viewer menus 506

---

**8**

8 node brick 157

---

**A**

abbreviation 26, 30

abort

DEFINE command 399, 403, 407, 413

ABORTCOMI

see \$ ABORTCOMI 63

ABS function 32

accuracy 593, 596, 603, 612, 617, 634

ACOS function 33

ACOSD function 34

ACTIVATE command 679

adaptive integration methods for RHS 421-422

adaptive time-step 126

adding labels 238

addition 31

air 252, 464

analysis 126, 310

frequencies 198

ANALYSISDATA command 122

ANIMATION command

Modeller 130

Post-Processor 683

anisotropic materials 375, 422

anisotropy 252, 464

arc 133, 190

conductors 524

- ARC command  
     Modeller 132  
     Post-Processor 685
- ARITHMETIC command 687
- arrow keys 370
- ASIN function 34
- ASIND function 34
- ASK  
     see \$ ASK 63
- ASKPARAMETER  
     see \$ ASKPARAMETER 63
- aspect ratio searching 396
- ASSIGN  
     see \$ ASSIGN 73
- asymmetric matrix 639
- ATAN function 34
- ATAN2 function 34
- ATAN2D function 34
- ATAND function 34
- automatic potential cuts 127, 501
- AXESVIEW command 689
- 
- B**
- background 293
- BACKGROUND command 135
- backscattering 562, 565
- BACKSPACE  
     see \$ BACKSPACE 73
- backup file 447
- base plane 394  
     subdivision editing 445  
     subdivision mode 410
- beam flux density 661
- beam power density 661
- BEAMLETPOWER 845-847
- bedstead 141, 190, 517
- BEDSTEAD command 140
- BEND command 143
- BH curves 252, 375, 464, 497, 702
- BHDATA command  
     Modeller 144  
     Post-Processor 702  
     Pre-processor 375
- bitmap 281, 781
- BLEND command 147
- block 148
- BLOCK command 148
- bmp file format 281, 781
- body 89-90
- BODY command 704
- boolean  
     operations 184  
     parameters 36
- BOUNDARY command 150
- boundary conditions 154, 587, 609-610, 615
- current flow 616

- default  
  scalar potential 616  
  thermal 593, 596  
  vector potential 617  
defining 427  
Definition Mode 426  
  extending the mesh 449  
derivative 428, 432  
displaying 440, 504  
electrostatics 616  
essential 593  
functional 429  
incident field 428  
labels 491  
mixed 616  
Modification Mode 480  
natural 428  
non-zero total scalar potential 421  
normal field 427, 432  
open boundaries 588, 618, 635  
perfect conductor 427  
potential 427-428, 432  
radiation 427, 432  
slip 427, 429  
symmetry 138, 275, 427-428, 432  
tangential field 427, 432  
vector potential 427-428, 432  
voltage 428  
boundary label 154, 214  
BOUNDARY sub-command  
  IDEAS command 454  
BREAKERROR  
  see \$ BREAKERROR 63  
brick  
  20 node 160, 190, 525  
  8 node 157, 190, 525  
brick conductors 157, 160, 393, 525  
BRICK20 command 159  
BRICK8 command 156  
BUFFER command 708  
built-in commands 49  
bulk conductors 177, 622
- 
- C  
c 670  
calculator 58  
capacitors 176  
CARTESIAN command 710  
cartesian graphs 689  
CASE sub-command  
  SOLVERS command 493  
CASES 666, 681  
CATIA format 244, 298  
CD  
  see \$ CD 80  
CDBox 368, 372  
CEDITOR command 162, 712

cell 89-90  
 cell properties 169  
 CELLDATA command 168  
 chamfer 143, 147  
 character values 35  
 character variables 59  
 charge density 343, 422, 660  
     unit 838  
 Charged Particle solver 126, 422, 427,  
     488, 661  
 CHECK command  
     Pre-Processor 378  
 CHECK command  
     Modeller 171  
     Pre-Processor 378  
 CHECK sub-command  
     SOLVERS command 494  
 checking 171  
 CIRCLE command 718  
 CIRCUIT command 173  
 Circuit Editor 163, 713  
 circular sheet face 196  
 CLEAR command  
     Modeller 180  
     Post-Processor 720  
     Pre-Processor 380  
 clipboard 281, 781  
 CLOSE  
     see \$ CLOSE 73  
 coenergy 745  
 coercive field 497, 660  
 coercive force 146  
 coil field integrals 126  
 COLOUR command  
     Modeller 182  
     Post-Processor 722  
     Pre-Processor 381  
 ColourBox 368, 373  
 colours 182  
 COMBINE command  
     Modeller 184  
     Post-Processor 725  
 combining bodies 184  
 COMINPUT  
     see \$ COMINPUT 61  
 command line interface 23  
 command separator 82  
 commands 26  
     built-in 49  
     Post-Processor 653  
     Pre-processor 363  
 COMMENT command 186  
 comments 82  
 COMPATIBILITY command 187  
 COMPONENT 28, 657  
 conditional commands 52  
 conductivity 252, 464, 497, 666  
     unit 489, 838

CONDUCTOR  
  DEFINE sub-command 383  
  ERASE sub-command 386  
  EXTERNAL sub-command 386  
  MODIFY sub-command 387  
  PRINT sub-command 390  
  QUIT sub-command 391  
  WRITE sub-command 391

CONDUCTOR command  
  Modeller 189  
  Post-Processor 727  
  Pre-Processor 383, 512

conductor field 602

conductor file 211, 235, 732

conductor symmetries  
  see symmetry  
    conductors 532

conductors 89, 548  
  arcs 133, 524  
  bedsteads 141, 517  
  bricks 157, 160, 525  
  constant perimeter ends 223, 322, 520  
  EXPORT command 211  
  forces 705  
  helical ends 232, 519  
  IMPORT command 235  
  in the mesh 420-421  
  racetracks 290, 515  
  single filament 531

solenoids 308, 514  
straight bars 315, 522  
voltage drop 705

cone 196

Console 226, 757

CONSTANT  
  see \$ CONSTANT 55

constant perimeter end 190, 223, 322, 520

construction lines 397

continuation lines 28

continuity 603, 610

contour 192, 340

CONTOUR command 192

contour maps 776, 825, 852

convergence 637

coordinate system  
  see local coordinate system 529

copy 331

copy picture 281, 781

COPYCASE command 733

copying base plane facets 407

correcting errors 335

COS function 34

COSD function 34

COSH function 34

COTAN function 34

cover 354

COVER command  
  Modeller 194

- creating objects 148, 196, 287, 313, 329, 353
- cube 148
- current
- charged beam 834
- CURRENT 845-847
- current density 343, 531, 661-664
- charged beam 669
  - unit 489, 838
  - vector 422, 425
- current density errors 661-662, 664
- current density in lossy dielectrics 662
- current drive 176
- current flow 422, 488, 661, 801
- Current Flow solver 661
- cuts 420
- CYCLE
- see \$ CYCLE
  - see \$ CYCLE 54
- cylinder 196
- CYLINDER command 195
- cylindrical sheet face 196
- 
- D**
- data 241
- data files
- \$ COMINPUT 362
  - BH curves 376
- command input 61
- conductors 211, 235, 392, 732
- database 486, 488-490, 673, 680, 770, 804
- grid 673, 755
- opc 243, 297
- Pre-Processor 444, 447, 481, 510
- sat 244, 297
- table 674, 748, 755, 778, 785, 818
- time-tables 493
- TRACK 677, 834
- universal 452
- data storage level 148, 169, 196, 205, 214, 287, 313, 329, 341
- database 186, 310
- DATALINE command 735
- DATAVECTOR command 738
- DATAVECTORSET command 740
- date 59, 508
- DBCASEDATA command 197
- DE curves 375
- DEBUG 378, 450
- DEFINE command 393
- DEFINE sub-command
- CONDUCTOR command 383
- defining variables 337
- deformed geometry 827
- delete 200
- DELETE command 200
- DELTA function 32

DENSITY 846  
derivative boundary condition  
    see boundary conditions  
        derivative 427  
DEVICE command 434  
device nomination 362  
DIALOG  
    see \$ DIALOG 67  
DialogBox 368, 372-373  
dialogue file 25  
diode 178  
disc 196  
discretisation 393-394, 403, 405  
displacement 664  
display 192, 233, 300, 340  
    colours 182  
DISPLAY command 436  
DISPLAYLINE  
    see \$ DISPLAYLINE 65  
division 31  
DO  
    see \$ DO 53  
double-click 372  
draft angle 319  
DRIVE command 202  
drive label 155, 429, 491, 531, 625  
DRIVE sub-command  
    SOLVERS command 495  
drives 495

---

**E**

e0 670  
easy direction 146  
eddy currents 488  
    ELEKTRA 422  
    motion induced 423  
edge 88, 353  
edge properties 205  
EDGEDATA command 205  
EDIT command 444  
eigen-values 488, 493  
electric field strength 661-664  
    unit 489, 838  
electric flux density 661-663  
    unit 489, 838  
electric scalar potential 605, 661-663  
    unit 489, 838  
Electromagnetic 662  
Electromagnetic solvers 126, 423, 427, 488  
Electrostatic solver 126, 662  
electrostatics 422, 488  
    SCALA 422  
    TOSCA 422  
element types 169, 214, 489  
    defining 422  
    displaying 440, 504

ELIF  
     see \$ ELIF 52

ELSE  
     see \$ ELSE 52

EMITTER command 206

emitter data 562, 568

emitters 548

END-do  
     see \$ END 54

END-for  
     see \$ END 54

END-if  
     see \$ END 54

END-while  
     see \$ END 54

END command  
     Modeller 210  
     Post-Processor 742  
     Pre-Processor 447

energy 745  
     unit 838

ENERGY command 743

energy loss factor 562, 566

entity 89

environment variables 35, 80, 372, 434

EQUATION  
     see \$ EQUATION 55

ERASE sub-command  
     CONDUCTOR command 386

ERRORHANDLER  
     see \$ ERRORHANDLER 63

Euler angles 83, 240, 331, 395, 422, 530, 802

EXEC  
     see \$ EXEC 77

EXIST  
     see \$ EXIST 80

EXIT  
     see \$ EXIT 55

EXP function 34

exponentiation 31

EXPORT command 211

exporting conductors 211, 392, 732

exporting data 297

expressions 30  
     boundary conditions 429  
     Post-processing 657

EXTEND command 448

extending volumes 319

EXTERNAL 378

external fields 126-127, 501, 603, 606

EXTERNAL sub-command  
     CONDUCTOR command 386

EXTRACTCELLS command 212

extrusion 319, 395, 413  
     direction 395  
     extending the mesh 448

extrusion direction 395

---

**F**

face 88  
FACEDATA command 213  
Facet Definition Mode 405  
    EDITING 445  
Facet Group Operations Mode 407  
field calculation methods 800  
field emission 557  
field point coordinates 660  
field values  
    along a line 769  
    at a point 787  
    on a general surface 711, 788  
    on a spherical surface 807  
field vectors 776, 852  
filamentary conductors 531  
filaments 177  
file  
    open 243  
    save 297  
    save as 297  
FileBox 368, 370, 373  
FILEEXISTS 80  
filenames 35  
FILEPROMPT  
    see \$ FILEPROMPT 64

files  
    command input 181, 380, 721  
FILETYPE 80  
FILL command  
    Modeller 216  
    Pre-Processor 450  
FILTER command 219  
finite element method 609  
FIT command 747  
fitted constant perimeter end 223  
FITTEDCPE command 222  
fixed temperature 154  
flux density errors 661-662, 664  
flux tubes 825, 832  
FLUXLINKAGE command 749  
FOR  
    see \$ FOR 54  
force 669, 671, 763  
    integral 705, 744  
    unit 838  
FORMAT  
    see \$ FORMAT 73  
Fourier series fitting 748, 792  
Fowler Nordheim field emission 556-557  
free format input 48  
frequency 198, 666  
FUNCTION  
    see \$ FUNCTION 60

functional boundary condition  
 see boundary conditions  
 functional 427

---

## G

Galerkin 610  
 global coordinate system 345  
 GRAPH command 751  
 graphical output 281, 286, 781, 790  
 graphical user interface 23  
 graphs 689, 735, 772, 784, 791  
 Green's theorem 610  
 GRID command 754  
 GROUPBOX  
 see \$ GROUPBOX 70  
 GUI 23  
 GUIOPTIONS command  
 Modeller 225  
 Post-Processor 756

---

## H

hard magnetic materials 146  
 heat density 423, 664  
 heat flux 154, 664-665  
 heat flux errors 664-665  
 heat tables 822

heat transfer 154, 592, 595  
 helical conductors 519  
 helical end 232  
 HELICALEND command 230  
 HELP command 451  
 help character 27  
 hide 220, 233  
 HIDE command 233  
 hide menus 374  
 hiding objects 300  
 high frequency 488  
 High Frequency 663  
 High Frequency solvers 19, 126, 427, 488  
 history 234, 284, 335  
 HISTORY command 234  
 hollow 303  
 homogeneity 671  
 horizontal menu 368  
 hysteresis 146  
 hysteresis energy loss 628, 662, 664  
 hysteretic materials 146

---

## I

I-DEAS  
 BOUNDARY sub-command 454  
 MATERIALsub-command 456  
 QUIT sub-command 457  
 READ sub-command 457

IDEAS command  
Post-Processor 758  
Pre-Processor 452  
**IF**  
    see \$ IF 52  
**IF function** 32  
**IGES format** 244, 298  
**IMPORT command** 235  
importing conductors 235, 481  
inductors 176  
inserting components 243  
**INSERTOP2FILE command** 236  
insulator 592, 595  
**INT function** 32  
**integral** 669, 761  
    field calculations 801  
    line 786  
    Lorentz force 705, 744  
    Maxwell Stress 671  
    power loss 744  
    stored energy 744  
    surface 777, 809  
    volume 744, 849  
**INTEGRATE command** 763  
**interface**  
    reduced scalar potential 421-422  
**intersecting bodies** 184  
**ISOSURFACE command** 766  
**isotropy** 252, 464

iteration data 127, 501

## J

Jacobian 379  
JE curves 375  
joining bodies 184

## K

Keyboard Shortcuts 120, 656

## L

label 238  
**LABEL command**  
    Modeller 238  
    Pre-Processor 458  
**labels** 394  
    boundary conditions 428, 491  
    DEBUG 378  
    displaying 439, 504  
    element types 423  
    EXTERNAL 378  
    material names 423  
    potential types 423  
    user defined 458

- laminations 375, 422
- LAYOUT  
see \$ LAYOUT 71
- LCS 240  
name 293
- LCS command 240
- least squares 610
- Legendre polynomial fitting 748
- length  
unit 489, 838
- limits 21
- LINE command 768
- linear algebra 636-637
- linear elements 310, 422
- linear extrusions 394, 413
- linear motion 488
- linearity 252, 464
- list 220, 241
- LIST command 241
- LOAD command  
Modeller 243  
Post-Processor 770
- local coordinate system 89, 240, 345  
anisotropic materials 422  
conductors 529  
DEFINE command (UVW) 395  
field calculations 785, 802  
name 293  
permanent magnets 422
- Post-processing 681, 813  
system variables 668
- LOFT command 246
- LOG function 34
- LOG10 function 34
- LOGGEDDATAFILE command 772
- loops 53
- 
- M**
- M 846-847
- m0 670
- magnetic field 548
- magnetic field strength 662-664  
unit 489, 838
- magnetic flux density 662-664  
unit 489, 838
- magnetic scalar potential 154, 664  
unit 489, 838
- magnetic vector potential 154, 662-663  
unit 489, 838
- Magnetization solver 663
- magnetization tables 822
- Magnetostatic solver 126, 664
- magnetostatics 420, 488
- MAP command 775
- material  
label 169, 252, 464  
name 420, 490, 497

NULL 264  
orientation 425  
properties 252, 464, 495  
**MATERIAL** sub-command of IDEAS 456  
Material Definition Mode 420  
    CARMEN 421  
    ELEKTRA 421  
    extending the mesh 449  
    SOPRANO 422  
    TOSCA 420  
material label 148, 196, 287, 313, 329  
Material Modification Mode 479  
**MATERIAL** sub-command of the SOLVERS command 495  
**MATERIALS** command  
    Modeller 248  
    Pre-Processor 462  
MAX function 33  
**MAXIMUM** 753  
maximum value 669  
Maxwell stress 666, 671, 764  
menu interface 23  
menu off 374  
menus 368  
merging faces 143, 147, 273  
**MESH** command  
    Modeller 257  
    Pre-Processor 467  
mesh control 169, 214  
mesh size 205, 341  
**MessageBox** 62-63, 82, 368, 373  
**MIN** function 33  
**MINIMUM** 753  
minimum value 669  
**MOD** function 33  
Modal HF solver 663  
model body 103  
**MODEL** command 263  
model dimension 337  
modeller.comi 87, 181, 380  
Modeller\_n.log 25  
Modeller\_n.lp 25  
**MODIFY** command 469  
**MODIFY** sub-command  
    **CONDUCTOR** command 387  
**MORPH** command 265  
motion 607  
Motional EM solver 18  
**MOUSE** command  
    Modeller 267, 779  
moving parts 423  
multi-valued potential 420  
**MULTIPHYSICS** command 269  
multiplication 31  
multiply connected 127, 501  
mutual inductance 166, 178, 716

---

**N**

name 293  
 negative periodicity 275  
 Newton-Raphson method 611  
 NINT function 33  
 nodally averaged fields 800  
 nonlinear 252, 464  
 nonlinear analysis 127, 501  
 nonlinear analysis options 126  
 normal electric 154  
 normal field boundary condition  
     see boundary conditions  
         normal field 427  
 normal magnetic 154  
 normal unit vector 660  
 NOW 59  
 NULL material 264  
 numeric values 30  
 numerical variables 56  
 NX format 244

---

**O**

OFFSET command 273  
 OPEN  
     see \$ OPEN 73  
 Opera-3d 17

opera.comi 642, 721  
 operaanl 79  
 operating system commands 77  
 orientation 343  
 OS  
     see \$ OS 77  
 output times 198

---

**P**

p 670  
 packed materials 253  
 packing factor 343, 422  
 page number 508  
 PARAMETER  
     see \$ PARAMETER 55  
 ParameterBox 368-369, 373  
 parameters 26  
 ParaSolid format 244  
 particle trajectories 842, 852  
     combined field 726, 821  
 paste 281, 781  
 PAUSE  
     see \$ PAUSE 63  
 PEC 154  
     ELEKTRA 427  
     SOPRANO 427  
 Peclet number 608

perfect conductor boundary condition  
    see boundary conditions  
    perfect conductor 427

perfect insulator 154

periodicity boundary condition  
    see boundary conditions  
    symmetry 427

PERIODICITY command 274

PERIODICITY sub-command  
    SOLVERS command 497

permanent magnets 422

permeability 252, 464, 497, 666

permittivity 252, 464, 497, 666

phase angle of conductors 531

phase lag 252, 464

pick 220

PICK command 277

pick options 220

picking 369

picking filter 220

PICTURE command  
    Modeller 281  
    Post-Processor 781

planar face 319

PLOT command 783

png file format 281, 781

POINT command 787

Point Definition Mode 399  
    EDITING 445

Point Modification Mode 470

Poisson equation 609

Poisson's equation 591

POLAR command 788

polar graphs 689

polygon 354

polynomial fit 792

positional assignments 28

positive periodicity 275

Post\_n.log 25

Post\_n.lp 25, 847

potential boundary condition  
    see boundary conditions  
    potential 427

potential cut 127, 420, 501

potential interface 603

potential types 169, 490, 603, 618  
    defining 420  
    displaying 440, 504

POTENTIAL\_CUTn 420

potentials  
    multi-valued 420  
    potential cut 420  
    reduced scalar 420-421  
    single valued 420  
    total scalar 420-422  
    vector 422

power  
    density 669

unit 838

power loss

- integral 744

Pre\_n.log 25

Pre\_n.lp 25

PRECISIONDATA command 283

preview 284

PREVIEW command 284

PRINT command

- Modeller 286
- Post-Processor 790

PRINT sub-command

- CONDUCTOR command 390

printing 286, 790

prism 196, 287

PRISM command 287

Pro/E format 245

PROCESSLINE command 791

program sizes 21

PROJECTFOLDER

- see \$ PROJECTFOLDER 25

PROMPT

- see \$ PROMPT 64

prompting

- for parameter values 46
- in command input files 63-64

properties 241

pyramid 287

---

## Q

Q 846-847

quadratic elements 310, 422

quadratic extrusions 394, 413

Quench solver 664

QUIT sub-command

- CONDUCTOR command 391
- IDEAS command 457
- SOLVERS command 499

---

## R

R, Z 845

racetrack 190, 290, 515

RACETRACK command 289

radiation 154, 592, 595

radiation boundary condition

- see boundary conditions

  radiation 427

radio-buttons 372

RAN function 33

RANGE function 33

READ

- see \$ READ 73

READ command 481

READ sub-command

- IDEAS command 457

- 
- redo 335  
REDO command 292  
reduced scalar potential 420-421, 602  
    interface 421-422  
reflect 331  
reflection symmetry 137  
regularise 200  
relaxation factor 123, 611  
RELEMENTHEAT 592  
removing objects 200  
RENAME command 293  
replay 337  
REPLAY command 294  
residual 611  
resistors 176  
restarting 632  
RHS integrals 127, 421, 501  
Richardson-Dushman 549  
RNODALHEAT 592  
rotate 331  
rotational motion 488  
rotational symmetry 138  
rotational velocity 343  
RSTART 845  
rubber-box 369  
Runge-Kutta  
    updates 126
- SAT format 245, 298  
saturation magnetization 146  
SAVE command 297  
save picture 281, 781  
saving data 297  
scalar potential boundary condition  
    see boundary conditions  
    potential 427  
scalar property 422-423  
scale 331  
Schottky field emission 556  
scoped variables 58  
scroll bars 371, 373  
secondary emission 562, 566  
secondary emission system variables 563  
SELECT command  
    Modeller 299  
    Post-Processor 793  
selecting 368  
selecting parts for display 300  
SET command 798  
SETTINGS sub-command of the SOLVERS  
    command 499  
shape function 603, 609, 617  
sheet face 89, 91, 273  
SHELL command 303  
SHOW command 804

SIBC  
see surface impedance boundary condition 255

SIGN function 33

simple updates 126

SIMULATION command 806

simulation title 186

simulations 488

SIN function 34

SIND function 34

single filament conductors 531

single valued potential 420

SINH function 34

SKETCH command 304

skin depth 620

slider bars 374

SLIP command 484, 544

slip surface 429, 484

solenoid 190, 308, 514

SOLENOID command 306

SolidWorks format 245

SOLVERS command 309, 485

source current density 660

source currents 603

source magnetic field strength 661-664

space charge beam 488

space charge density 661

sphere 313

SPHERE command 313

SPHERICAL command 807

SQRT function 34

steady-state ac 488, 491, 531, 586, 606

STEP format 245, 298

stiffness matrix 611

stored energy  
integral 744

straight 190, 315

straight bars 522

STRAIGHT command 314

strain 664

stress 665

stress analysis 827

Stress solvers 664

STRETCH command 317

STRING  
see \$ STRING 55

string variables 355, 854

Subdivision Modification Mode 474

subtracting bodies 184

subtraction 31

SUP2 function 33

SUP3 function 33

supremum 33

SURFACE command 809

surface impedance boundary condition 255, 620

SWEEP command 318

sweeping 319

SWITCH function 33  
switch in circuit 178  
symmetry 137, 274  
conductors 532  
recreating in Post-Processor 680-681, 812  
symmetry boundary condition 154  
see boundary conditions  
symmetry 427  
SYMMETRY command 811  
system variables 355, 657, 815, 854  
system variables for the VIEW command 845  
SYSVAR  
see \$ SYSVAR 60  
SYSVARIABLE command 814

---

**T**

TABLE command  
Post-Processor 817  
Pre-Processor 502  
TAN function 34  
TAND function 34  
tangential constant perimeter end 322  
tangential electric 154  
tangential field boundary condition  
see boundary conditions  
tangential field 427  
tangential magnetic 154  
tangential unit vector 660  
TANGENTIALCPE command 321  
temperature 592-593, 595, 664-665  
unit 838  
TEMPO 423  
text function 37  
text output modes 62  
thermal boundary condition 154  
thermal conductivity 252, 591  
thermal saturation limit 549  
Thermal solvers 665  
three dimensional model 612  
THREED command  
Modeller 324  
Post-Processor 823  
Pre-Processor 503  
time 59, 508, 666  
time-step 126  
time-stepping 128, 501, 624  
time-table files 493  
time harmonic 586, 606  
time of flight 845-847  
time tables 626  
timers 373  
TITLE 59, 186, 491, 770  
TITLE command  
Modeller 327  
Post-Processor 829

Pre-Processor 507  
 titles 327, 508, 829  
 TODAY 59  
 TOF 845-847  
 tolerance 123  
     conductor fields 531  
     group operations 408  
 TOOLBUTTON  
     see \$ TOOLBUTTON 72  
 topology problems 171  
 torque 669, 671, 705, 763  
 torus 329  
 TOSCA 427, 488  
 total scalar potential 420-422, 603, 634  
 TRACK command 831  
 transfer 593  
 TRANSFORM command 330, 509  
 transformations 274, 331  
 transforming base plane coordinates 407  
 transient 488, 492, 531  
 transient analysis 607  
 transient driving functions 624  
 Transient EM outut times 198  
 transition ratio 259  
 translate 331  
 tube 196  
 two dimensional model 612  
 TXBEAM, TYBEAM, TZBEAM 846

---

**U**  
 undo 284, 335  
 union 184  
 unique name 149, 194, 196, 288, 313, 329,  
     353  
 units 310  
     analysis data 489  
     BH data 376  
     database 311  
     material 256  
     Post-processing 659, 838  
 UNITS command 835  
 units display 508  
 upwinding 126, 128, 501  
 user interface 23  
 user variables 55, 337

---

**V**  
 variables 337  
     character or string 59  
     in expressions 30  
     in free format input 48  
     numerical 56  
     system 657, 815  
 variational methods 610  
 VECTOR command 339

- vector diagrams 689, 738, 740  
vector maps 776, 852  
vector potential 422  
vector potential boundary condition  
    see boundary conditions  
    vector potential 427  
vector property 422, 425  
velocity 343, 423, 425, 660  
VELX 845-847  
VELY 845-847  
VELZ 845-847  
version number 59  
vertex 88  
VERTEXDATA command 341  
vertical menu 368  
VFDIR 35, 80, 372, 376  
VFGRAPHICS 434  
VFINV 435  
VFWINDOWH 435  
VFWINDOWW 434  
VIEW command 840  
virtual work 764  
voltage 154  
    integral 705  
voltage boundary condition  
    see boundary conditions  
    voltage 427  
voltage drive 617  
voltage driven circuits 176  
volume 669, 744  
conductors 731  
label 343  
properties 169, 343  
VOLUME command  
    Modeller 342  
    Post-Processor 848  
VX and VY 28  
VX, VY and VZ 28, 657
- 
- W**
- WCS 345  
weighted residuals 610  
WHILE  
    see \$ WHILE 54  
windings 177  
WINDOW command  
    Modeller 351  
    Post-Processor 851  
wire edge 88  
wireedge 353  
WIREEDGE command 352  
working coordinate system 90, 240, 345  
WRITE  
    see \$ WRITE 73  
WRITE command 510  
WRITE sub-command  
    CONDUCTOR command 391

## X

X, Y 845-847  
X0BEAM, Y0BEAM, Z0BEAM 846  
xpm file format 281, 781  
XSTART 845

---

## Y

YESORNO 59  
yield 562, 566  
YSTART 845