

# **NEW FEATURES IN OPERA V8.5**

Copyright © 1999-2002 by Vector Fields Limited, England

This document was prepared using Adobe® FrameMaker®.

UNIX is a trademark of X/Open Company Ltd.

HPGL is a trademark of Hewlett-Packard Incorporated

PostScript is a trademark of Adobe Systems Incorporated

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

X Window System is a trademark of Massachusetts Institute of Technology

I-DEAS is a trademark of Structural Dynamics Research Corporation

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

# CONTENTS

## *Chapter 1* *Introduction*

About this document .....	1-1
Frequently Asked Questions .....	1-2

## *Chapter 2* *User Interface*

Introduction .....	2-1
--------------------	-----

## *Chapter 3* *OPERA-2d*

Overview .....	3-1
Linear Motion (LM) .....	3-3
The <b>LMMOTION</b> Command .....	3-18
The <b>SOLVE</b> Command .....	3-21

## *Chapter 4* *OPERA-3d*

Overview .....	4-1
Modeller .....	4-2
Post Processor .....	4-4

*Chapter 5**New Modeller Features*

Cutaway Overlap With Regularisation .....	5-1
Building a Library of Sub-Models .....	5-4
Blend and Chamfer Edges .....	5-6
Swift Selection of Entities .....	5-10

*Chapter 6**Getting Started - Post Processing*

Post Processing the Getting Started Model .....	6-1
Displaying the Model .....	6-4
Checking the Solution .....	6-6
Field Homogeneity .....	6-8
Force Between Poles .....	6-11

*Chapter 7**Implementation Notes*

Windows .....	7-1
---------------	-----

# *Chapter 1*

## **Introduction**

### **About this document**

---

---

This New Features document highlights the new facilities that are available within the OPERA software. Both OPERA-2d and OPERA-3d are covered by this document.

The complete documentation is available on the distribution CD, in PDF format. In addition the Reference Manuals are included as electronic Help files for Microsoft Windows implementations.

## Frequently Asked Questions

---

The Frequently Asked Questions document is available on the distribution CD, and has been extended since the last release. The new topics included in the FAQ document are as follows:

### *Platforms and Compatibility*

- FlexLM Licensing on UNIX
- InstallShield of OPERA 8.5 on WinNT4
- Multiple Processors in a PC
- Network-Dongle on a Stand Alone PC
- Page File is too small (WinNT)
- Running OPERA off-line (Advanced Users)
- Sony Vaio Notebook
- Spawned Sub-Processes on UNIX
- Supported Hardware OPERA v8.5

### *Command Language*

- Batch Processing
- **CLEAR** Command in a *.comi* file

### *Basics and Field Theory*

- Constants used in OPERA
- Differences between OPERA-2d and OPERA-3d
- Discrepancy between Measured and Calculated Field
- Gradient of a Field along a Line
- Particle Energy and Velocity
- Relativistic Correction

### *OPERA-2d*

- Changing External Circuits for an RM Restart
- Restart
- Re-scaling a model

***OPERA-3d***

- Combining 2 or more Meshes in Pre Processor 1
- Combining 2 or more Meshes in Pre Processor 2
- Conductors
- Conductor Only Energy Calculation
- Constant Perimeter Ends (CPEND)
- Construction Line Limit in the Pre Processor
- Currents on the Surface of Superconducting Sheets
- Detached Surface Restart
- Estimated RMS Error in the *.res* File 1
- Estimated RMS Error in the *.res* File 2
- Flux Plot
- Hysteresis in ELEKTRA
- Impedance of a Coil
- Iterations in ELEKTRA
- Multiply Connected Regions 2
- Multiply Connected Regions 5
- Sub-Problems (Field Zoom)
- Restarts
- Superconducting Coils in TOSCA

***Geometric Modeller***

- ACIS surface mesh
- Merging Cells with different Data Level
- Mesh Generation
- Moving of Points
- Non Unique Mapping
- Parameterization
- Picking by Name
- Questions about the Modeller
- SAT File Import
- Skin-Depth in the Modeller

- Volume Mesh Error
- Volume Meshing

### *Fluxes, Forces and Torques*

- Adequate Meshing
- Dielectrics
- Modelling a SRM

### *Space Charge Modelling*

- Absolute Tolerance in the Emitter file
- Electrostatic Problem in SCALA
- Estimating Errors in the Current Density of a Particle Beam
- Maximum Distance between Sample Rays
- Maximum Step Length - Stopping of Particles
- Maximum Step Length and Accuracy
- Sampling Distance
- Some Emitters Do Not Emit
- Tabulating Emitter Relationship (Langmuir/Fry Emitters)
- Tolerance on Trajectory Calculations
- Total Current intersecting a Line
- Under-relaxation Factor in Space Charge Problems

### *Imprting and Exporting*

- I-DEAS Export
- Material Orientation and I-DEAS

### *Examples*

- Thermal Emission in OPERA-2d
- Tracking a Particle in a Constant Magnetic Field

# Chapter 2

## User Interface

### Introduction

---

The major change to the user interface of Version 8.5 is in the OPERA-3d post processor, which now uses the same style of interface as the Geometric Modeller. Many internal changes have been made to this interface to improve the layout, low-lighting and general ease of use.

#### Cancel Button

In the OPERA-3d Modeller and post processor the cancel button  can be used to interrupt execution of a command input file.

#### Log Files Directory

The *lp*, *log* and *backup* files created by the interactive programs are now stored in a sub-directory of the current directory called *opera\_logs*.

#### String Variables

Two predefined string variables are available. On any input line **&NOW&** will be replaced by the current time and **&TODAY&** by the date. The values of **NOW** and **TODAY** cannot be changed by the user.



# Chapter 3

## OPERA-2d

### Overview

---

---

#### BH Curves and Linear Materials

It is no longer necessary to supply BH curves for linear materials even for non-linear analyses. If no BH curve is in the data, the analysis programs will use the region permeability.

BH curves must still be given for permanent magnet materials to provide a value for the coercive force ( $H_c$ ).

#### OPERA-2d/LM

The OPERA-2d/LM analysis program is a new module created for the analysis of linear motion. For XY plane symmetry models translational motion in both X and/or Y and rotational motion about a point can be specified. For axi-symmetric models motion in the Z direction can be specified. Full details are provided, see “[Linear Motion \(LM\)](#)” on page 3-3. There is an additional pre processor command, **LMMOTION**, to correspond to this new solver, see “[OPERA-2d/LM](#)” on page 3-1.

## Pre and Post Processor

### **CONTOUR** *Command*

The **CONTOUR** command now uses value of the parameter **FILL** last set by the **RECONSTRUCT** command to decide whether to display the material outlines on top of zone contours.

### **LMMOTION** *Command*

To control the selection of moving parts within the LM solver, the **LMMOTION** command has been added. This allows the mesh generator to recognise a group of regions as being the moving part (see “[The LMMOTION Command](#)” on page 3-18).

### **SOLVE** *Command*

A number of changes have been made to the **SOLVE** command. The **LOGFILE** sub-command is now available for use in the transient (TR) analysis as well as the LM and RM solvers. Within the transient analysis, currents in each external circuit can be traced through the full analysis. Within the LM and RM solvers there is a range of other values associated with the motion and forces acting on the moving part. The changes are highlighted in “[The SOLVE Command](#)” on page 3-21.

### *Time Tables*

The transient analysis programs (LM, RM and TR) now look for time table files in the directory containing the data file rather than the current directory.

There is an alteration to the definition of the **TTOFF** function, as shown below:

<b>TTOFF</b>	$t < 0: F = F(0)$ $t \geq 0: F = \text{cubic splines}$ The program prompts for the name of a time-table file. $F(0)$ is the value of the function in the table file at time $t=0$ .
--------------	--

## Linear Motion (LM)

---

The Linear Motion program (OPERA-2d/LM) is a Transient Eddy Current Solver, extended to include the effects of motion. The solution can have XY symmetry, where motion can be in both X and Y directions, as well as allowing for rotational motion about a point. The solution can also have axi-symmetry, where the motion is restricted to motion along the axial Z. The solver also provides for the use of external circuits and coupling to mechanical equations.

OPERA-2d/LM models make use of a special remeshing technique that requires that the user separates the model into 3 groups of regions: the moving regions, the regions that allow motion and the static sections. This process is controlled using the **LMMOTION** command (see the OPERA-2d/PP reference manual). During the time-stepping solution process the moving section of the model is repositioned and a reconnection mesh between moving and static sections created.

The meshing facilities made available by the **LMMOTION** command may also be used advantageously with any other OPERA-2d solver for any model, since the part of the model can easily be displaced without requiring as much mesh continuity.

OPERA-2d/LM models may be constructed with either linear or quadratic elements. It is advisable to maintain a balanced mesh (i.e. mesh of similar size elements) in the regions that allow motion and at the boundary of the moving part (although they do not have to match exactly).

### The Equations Solved

OPERA-2d/LM solves the vector diffusion equation with the magnetic vector potential (**SET SOLUTION=AT**) as the unknown variable:

$$\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} - \mathbf{H}_c \right) = \mathbf{J}_s - \sigma \frac{\partial \mathbf{A}}{\partial t} \quad (3.1)$$

in which the current density has been split into the prescribed sources,  $\mathbf{J}_s$  and the induced currents,  $\sigma \frac{\partial \mathbf{A}}{\partial t}$ . In two dimensions, only the z components of  $\mathbf{A}$  and  $\mathbf{J}_s$  exist. Equation (3.1) can be simplified to:

$$-\nabla \cdot \frac{1}{\mu} \nabla A_z - (\nabla \times \mathbf{H}_c)_z = J_s - \sigma \frac{\partial A_z}{\partial t} \quad (3.2)$$

## Driving Functions

The 'driving' field is provided by source currents,  $J_s$ , non-zero potential boundary conditions and permanent magnet coercive forces,  $\mathbf{H}_c$ . The permanent magnets obviously create a dc field. The other driving terms can have a prescribed shape in time. Each value of conductor number (region parameter **N**) can be related to a different driving function, with, if necessary, every region having a different function.

The driving function options are:

<b>Transient Driving Functions</b>	
<b>Name</b>	<b>Function</b>
<b>COSINE</b>	$t < 0: F = 1$ $t \geq 0: F = \cos(2\pi ft - \phi)$ . The program prompts for $f$ and $\phi$ .
<b>DC</b>	Uniform in time from $t = -\infty$ to $t = +\infty$
<b>PEAK</b>	$t < 0: F = 0$ $t \geq 0: F = a \exp\left(\frac{-t^2}{b}\right)$ . The program prompts for $t_c$ . $a$ and $b$ are chosen such that $F=1$ at $t=t_c$ .
<b>RAMP</b>	$t < 0: F = 0$ $t \geq 0, t \leq t_c: F = \frac{t}{t_c}$ $t > t_c: F = 1$ . The program prompts for $t_c$ .
<b>EXPONENTIAL</b>	$t < 0: F = 0$ $t \geq 0: F = 1 - \exp\left(\frac{-t}{t_c}\right)$ . The program prompts for $t_c$ .
<b>SINE</b>	$t < 0: F = 0$ $t \geq 0: F = \sin(2\pi ft - \phi)$ . The program prompts for $f$ and $\phi$ .
<b>STEP</b>	$t < 0: F = 0$ $t \geq 0: F = 1$

<b>Transient Driving Functions (continued)</b>	
<b>TTOFF</b>	$t < 0: F = F(0)$ $t \geq 0: F = \text{cubic splines}$ The program prompts for the name of a time-table file. F(0) is the value of the function in the table file at time $t=0$ .
<b>TTON</b>	$t < 0: F = 0$ $t \geq 0: F = \text{cubic spline}$ The program prompts for the name of a time-table file.

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 up to 1000 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. 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 switch-on cases, the function value at zero time need not be zero, but it is assumed that the function has value zero for all time before zero. Beyond the last value of time in the table the function continues with the same cubic function calculated for the last section of the table.

The driving functions may be viewed using the **Graphs** Option, available from the **FILE** Top Level Menu.

## Eddy Current Conductors

In some problems it is necessary to allow eddy currents to flow in driving conductors, redistributing the current density, or to limit the total eddy current in a conductor to zero. This is achieved by solving an extra equation for each group of regions which make up such a conductor. This equation limits the total current,  $I$ ,

flowing in terms of  $\frac{\partial A}{\partial t}$  and a potential gradient,  $\nabla V$ :

$$-\int_{\Omega_i} \sigma \left( \frac{\partial A}{\partial t} + \nabla V \right) \partial \Omega = I \tag{3.3}$$

The effect of the potential gradient is a spatially uniform current density over the conductor,  $J^*$  given by

$$J^* = -\sigma \nabla V \tag{3.4}$$

and becomes an extra unknown in the modified equation (3.2). The following two equations are solved together

$$-\nabla \cdot \frac{1}{\mu} \nabla A_z - \nabla \times \mathbf{H}_c = J^* - \sigma \frac{\partial A_z}{\partial t} \quad (3.5)$$

$$\int_{\Omega_j} \left( -\sigma \frac{\partial A}{\partial t} + J^* \right) \partial \Omega = \int_{\Omega_j} J_s \partial \Omega \quad (3.6)$$

Equation (3.6) is repeated for every group of regions with a different value of **N** and **SYMMETRY**  $\neq 0$ .  $J_s$  is given by **DENSITY**.

## External circuits

Transient solutions can be excited either by current sources or by voltage sources connected via external circuits to the model. The current density that can be defined for each region is supplied by a current source. Independent of the properties of the coils the current is given as a function of time. The voltage driven option allows a set of coils to be defined as a circuit which is connected to an external voltage source in series with an external resistance, capacitance and inductance. A coil is a set of regions that have the same conductor number **N**.

The Reference Manual provides a full description of External Circuits options. Hints on the usage of external circuits are given in a separate Application Note.

## Time Stepping

The finite element method used in OPERA-2d/LM is similar to that described previously. However the terms in  $A$  and  $\frac{\partial A}{\partial t}$  in equations (3.2) both yield matrices, referred to as **R** and **S**. The Galerkin procedure leads to a matrix equation

$$\mathbf{R}\mathbf{A} + \mathbf{S} \frac{\partial \mathbf{A}}{\partial t} + \mathbf{B} = 0 \quad (3.7)$$

where  $\mathbf{A}$  is now a vector of unknown potentials and  $\mathbf{B}$  is a vector of driving terms. The solution of equation (3.7) is also based on the Galerkin procedure.  $\mathbf{A}$  and  $\mathbf{B}$  are discretised in time using a first order function of time:

$$\mathbf{A}(t) = (1 - \tau) \mathbf{a}_n + \tau \mathbf{a}_{n+1} \quad (3.8)$$

$$\mathbf{B}(t) = (1 - \tau) \mathbf{b}_n + \tau \mathbf{b}_{n+1} \quad (3.9)$$

where

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \quad (3.10)$$

and  $a_n$  and  $b_n$  are values of  $\mathbf{A}$  and  $\mathbf{B}$  at time  $t_n$ . Using  $\tau$  as the weight in a Galerkin weighted residual solution of equation (3.7) leads to a recurrence relationship between  $a_{n+1}$  and  $a_n$ :

$$\left( \mathbf{R}(1 - \theta) - \frac{\mathbf{S}}{\Delta t} \right) a_n + \left( \mathbf{R}\theta + \frac{\mathbf{S}}{\Delta t} \right) a_{n+1} + b_n(1 - \theta) + b_{n+1}\theta = 0 \quad (3.11)$$

where  $\theta = 1$ .

### **Adaptive Time Stepping**

All drive functions start by using a minimum time step. After the first step, the time-step is automatically adjusted to achieve time-stepping relative errors of less than a user supplied tolerance. The time step is increased when the error is much less than the tolerance, or reduced if the error comes close to the tolerance. The time-step can never fall below the minimum time-step. For non-linear problems, a scheme is linked with the solution of equation (3.11).

### **Fixed Time Step**

Alternatively, the user can opt for the fixed time-stepping option, in which case the time-step will remain constant (as defined by the user).

## **Boundary Conditions**

OPERA-2d/LM supports the fixed potential ( $\mathbf{Fn}=\mathbf{V}$ ,  $\mathbf{Vn}=\mathbf{value}$ ) boundary conditions but does not cater for periodic boundaries. All other boundaries should be set to  $\mathbf{Fn}=\mathbf{DV}$ ,  $\mathbf{DVn}=\mathbf{0}$  (or to  $\mathbf{Fn}=\mathbf{NO}$ , which is equivalent).

## **Permanent Magnets**

Permanent magnets can be specified by a non-zero value of  $H$  for zero  $B$  as the first point of a BH curve. The easy direction is given by the region parameter **PHASE** (see the **BHDATA** command).

## Restarts

OPERA-2d/LM can be restarted from a results file. This allows linear motion solutions to be continued for larger values of time, or alternatively the Statics (ST), Steady State AC (AC) or Transient (TR) analysis programs can provide a solution for restart, enabling ‘switch-off’ cases to be studied. Statics and AC solutions look like transient solutions at  $t=0$ .

## Motion

### Variable speed

The user may specify the speed of the moving part as a function of time during the duration of the transient analysis. For example, the rotational speed (in RPM) of a 4-pole 60 Hz machine rising towards synchronous speed could be given by

$$\text{Speed} = 1800 \left( 1 - e^{-\frac{t}{0.01}} \right) \quad (3.12)$$

with an exponential rise time constant of 10msec. The user specifies the rotational speed in radians/second using an expression for a **CONSTANT** called **#ROTSPEED** in a *.comi* file. To implement equation (3.12), the file contains

```
$CONS #ROTSPEED 1800/60*2*pi*(1-EXP(-TTIME/0.01))
$CONS #SPEEDX 0.0
$CONS #SPEEDY 0.0
```

where **TTIME** is the system variable defining the current time in the transient. Clearly, the commands in the file could be considerably more complex with different expressions defining the behaviour at different times by using conditional **\$IF** clauses and output values from the solver (a list of these is given in the table in “Output log file” on page 3-9).

The following is a table defining the list of variables that should be defined within the file for controlling the speed.

<b>XY symmetry, variable velocity</b>	
<b>#SPEEDX</b>	Speed in the X direction (length units / s)
<b>#SPEEDY</b>	Speed in the Y direction (length units / s)
<b>#ROTSPEED</b>	Rotational speed (rad/s)

<b>Axi-symmetry, variable velocity</b>	
<b>#SPEEDZ</b>	Speed in the Z direction (length units / s)

### *Mechanical coupling*

Rather than using a user-specified fixed or varying speed, OPERA-2d/LM can compute the speed based on values of acceleration. Both the linear and angular acceleration may be defined by the user. This allows conditional behaviour, for instance, where additional load force / torque may be added to the machine subject to its operating speed or where the load has a time dependent behaviour. Similarly, to the variable speed option, the user defines **CONSTANTS** that control the acceleration at each time.

<b>XY symmetry, mechanical coupling</b>	
<b>#ACCELX</b>	Acceleration in the X direction (length units / s <sup>2</sup> )
<b>#ACCELY</b>	Acceleration in the Y direction (length units / s <sup>2</sup> )
<b>#ROTACCEL</b>	Rotational acceleration (rad/s <sup>2</sup> )

<b>Axi-symmetry, mechanical coupling</b>	
<b>#ACCELZ</b>	Acceleration in the Z direction (length units / s <sup>2</sup> )

For example, a body being accelerated in the X direction could be controlled using

```
$CONSTANT #ROTACCEL 0.0
$CONSTANT #ACCELY 0.0
$CONSTANT #MASS 10.0
$CONSTANT #ACCELX LMXFORCE/#MASS
```

In this example, the only force acting on the moving body is the electromagnetic force **LMXFORCE**.

### *Output log file*

The transient analysis programs store their results at a set of times defined by the user. When the LM solver is run and a long initial transient is expected, it is convenient to be able to observe the progress of the solution. The output from all time steps can be logged to a file in the format required by the **GRAPH** command. A typical log file would contain the time, speed and forces so that these may be graphed as the solution progresses.

A set of these outputs is defined when setting the analysis options. The outputs can be any of the following values (or expressions based upon them)

<b>General variables</b>	
<b>TTIME</b>	Current solution time
<b>I1, I2, ... , IN</b>	Current flowing in each external circuit defined in the model.

<b>Variables for XY symmetry</b>	
<b>LMXSHIFT</b>	Displacement in the X direction (length units)
<b>LMYSHIFT</b>	Displacement in the Y direction (length units)
<b>LMROTANGLE</b>	Rotational displacement (degrees)
<b>LMXSPEED</b>	Speed in the X direction (length units / s)
<b>LMYSPEED</b>	Speed in the Y direction (length units / s)
<b>LMROTSPEED</b>	Rotational speed (rad/s)
<b>LMXFORCE</b>	Force in the X direction (Force unit / length unit)
<b>LMYFORCE</b>	Force in the Y direction (Force unit / length unit)
<b>LMTORQUE</b>	Torque on the moving part (Force units)

<b>Variables for axi-symmetry</b>	
<b>LMZSHIFT</b>	Displacement in the Z direction (length units)
<b>LMZSPEED</b>	Speed in the Z direction (length units / s)
<b>LMZFORCE</b>	Force in the Z direction (Force units)

## Preparing an OPERA-2d/LM Model

### *Arranging Groups*

The preparation of the finite element model for the LM Solver requires that the model is separated into 3 groups of regions, namely the MOVINGGROUP, the MEDIUMGROUP and the STATIC regions. The names for the former two groups are chosen by the user and subsequently assigned their special properties using the **LMMOTION** command.

Figure 3.1 shows a primitive linear motor. Three phase windings, arranged in their respective slots, are excited sequentially to attract the moving part into place. Symmetry is exploited in this model, hence the modelling of half the machine.

The moving part region together with thin slices of air surrounding it are grouped under the name MOVE. The air regions along which the moving part will slide are grouped under the name ENV. Figure 3.2 and Figure 3.3 illustrate the members of each Group.

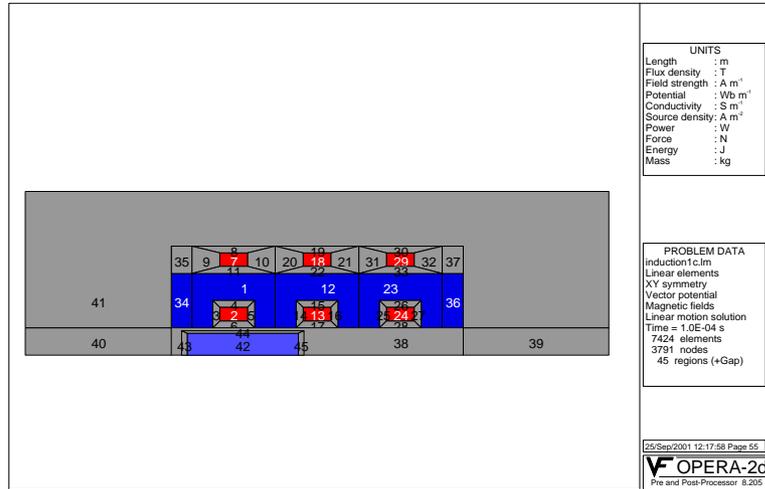


Figure 3.1 A simple linear motor model

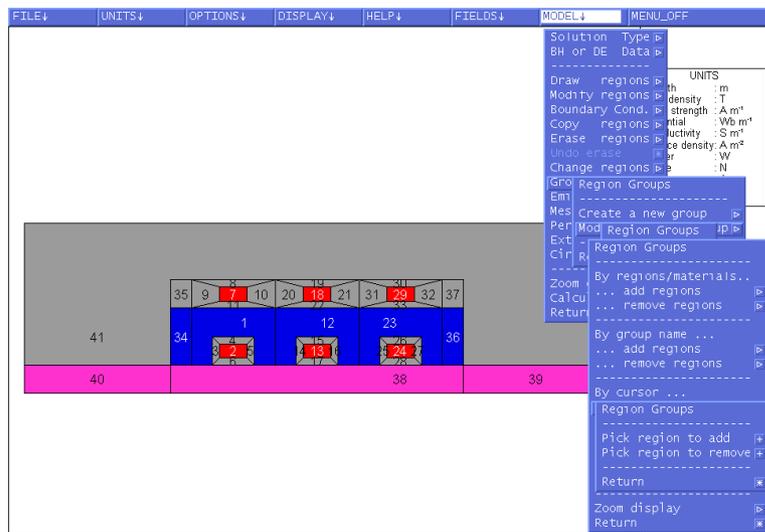


Figure 3.2 Members of the MEDIUMGROUP

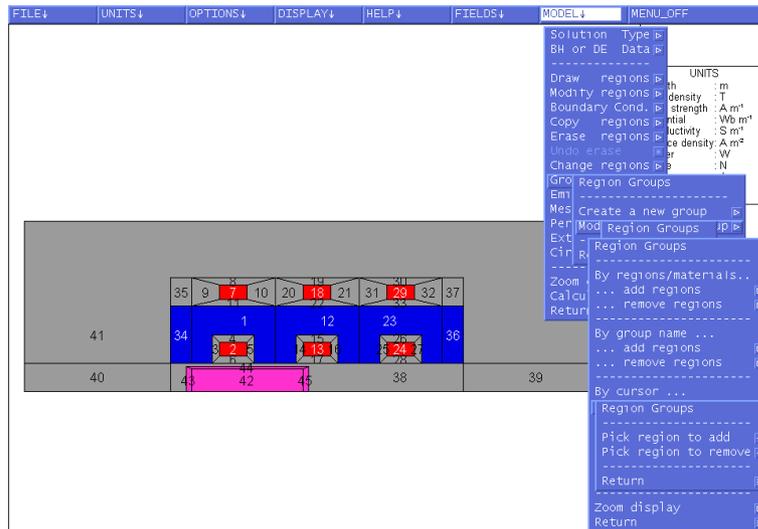
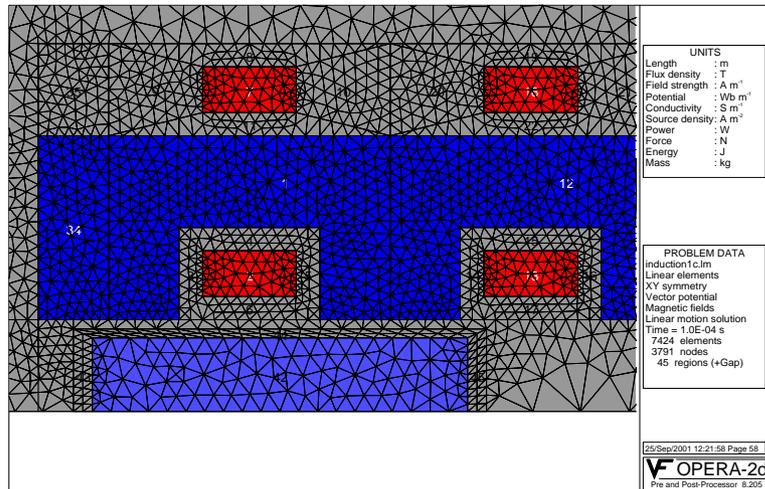


Figure 3.3 Members of the MOVINGGROUP

**Modelling  
aspects of the  
MOVINGGROUP**

The regions defining the **MOVINGGROUP** are drawn on top of the regions defining the **MEDIUMGROUP**. This unusual operation can only be valid with the scope of subsequently using the **LMMOTION** command. If symmetry is exploited in the model, such as the case of the present example, drawing regions of the **MOVINGGROUP** will result in the user being prompted to add points to the underlying regions. The user must reply **NO** to this.

In its simplest form, the **MOVE** Group would only contain one region, namely the moving part. However, in order to obtain accurate answers on force calculations, there should be at least three layers of elements in the air gap between the stator and moving part. The latter is therefore surrounded by air regions with Q-shaped elements which are added to the **MOVE** Group, ensuring a finer discretisation in the region of interest. Figure 3.4 shows the mesh around the plunger area.



**Figure 3.4 Mesh around the moving region**

*The  
**LMMOTION**  
 command*

The MOVE and ENV regions are assigned their special properties by selecting Model -> Draw Regions -> Linear motion air gap and completing the dialog box shown in Figure 3.5.

In keyboard mode the command is

```
LMMOTION MESHING=YES MOVINGGROUP=MOVE MEDIUMGROUP=ENV
CENTREX=0 CENTREY=0
```

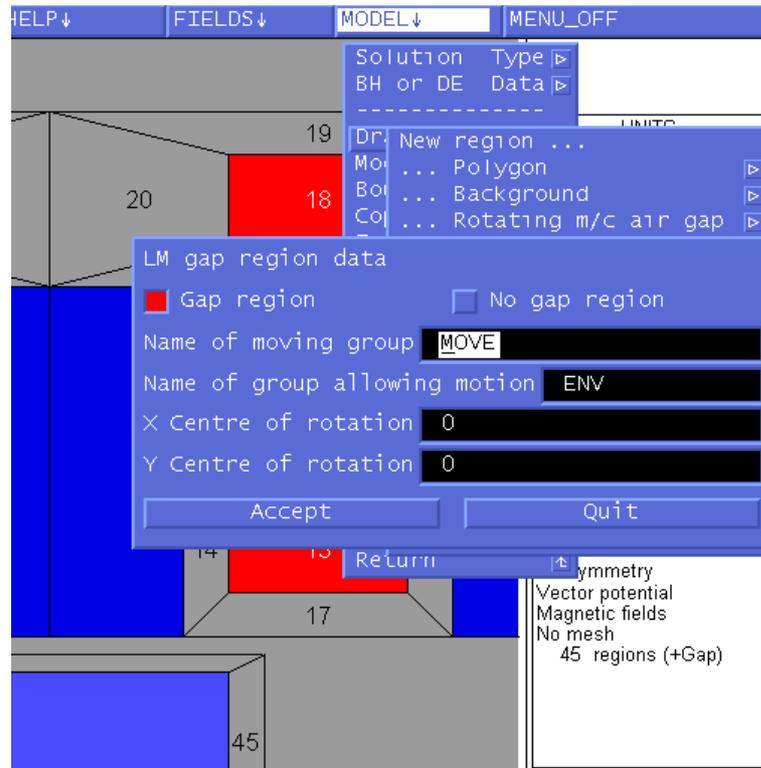


Figure 3.5 Assigning special properties

### *Variable speed or Variable acceleration*

The LM solver offers a variety of operation modes through the use of command (\*.comi) files. Command files can be used to assign a variable velocity; this is applicable in cases where a function of the velocity is known. Here is an example:

```
$CONS #SPEEDX 100*COS(2*PI*1000*TTIME)
$CONS #SPEEDY 0
$CONS #ROTSPEED 0
```

This will actually result in the moving part performing an oscillatory motion along the +/- X direction.

Alternatively, a comi file can also be used to assign a variable acceleration, based on the force exerted on the moving part at any given time. An example of this follows:

```
$CONS #ACCELY 0
$CONS #ROTACCEL 0
$CONS #MASS 0.1
```

```
$CONS #SPRINGFORCE LMXSHIFT*5.2E-03
$CONS #ACCELX (LMXFORCE-#SPRINGFORCE)/#MASS
```

In this example, the total accelerating force acting on the moving part is a function of the electromagnetic force exerted on it, as well as the mechanical force acted upon it by the spring to which it is attached.

## Preparing an OPERA-2d/LM run

The data required to analyse the model is input using the **SOLVE** command (see Reference Manual for details). The user can set the following information:

- Linear or Non-linear solution

The linear solution options simply solves with the value of permeability (permeability) given by the region parameter **PERM**. A BH curve is still required to give a value for the coercive force. The Non-linear option solves for the non-linear materials as defined by the BH (DE) characteristics defined for material numbers greater than 2. If Non-Linear is chosen the following parameters are set:

- Number of iterations can be set if a non-linear solution is requested. The value entered sets a limit on the maximum number of non-linear iterations.
- Tolerance is the convergence tolerance to be applied to the relative change in the solution.

- Time Step - Adaptive or Fixed

The user has a choice to fix the time step at which each solution is calculated, or set an adaptive time step. In Adaptive Time-stepping integration, a time-step variation is to achieve a user supplied relative tolerance between successive time-steps. In general, values should be between 0.03 and 0.0001, but values outside this range can be given.

- Output Times

A list of times at which the solution is stored may be supplied.

- Motion - variable speed or mechanical coupling

This defines how the motion of the moving part of the model is determined. (See “Motion” on page 3-8.)

- ‘New solution’ or Restart

This is only available if the *.op2* file already contains a solution. The TR solution program allows a solution to be continued from the result contained in an existing results file created by one of the analysis programs ST, AC or TR.

The driving function options are:

<b>Transient Driving Functions</b>	
<b>Name</b>	<b>Function</b>
<b>COSINE</b>	$t < 0: F = 1$ $t \geq 0: F = \cos(2\pi ft - \phi)$ . The program prompts for $f$ and $\phi$ .
<b>DC</b>	Uniform in time from $t = -\infty$ to $t = +\infty$
<b>PEAK</b>	$t < 0: F = 0$ $t \geq 0: F = a \exp\left(\frac{-t^2}{b}\right)$ . The program prompts for $t_c$ . $a$ and $b$ are chosen such that $F=1$ at $t=t_c$ .
<b>RAMP</b>	$t < 0: F = 0$ $t \geq 0, t \leq t_c: F = \frac{t}{t_c}$ $t > t_c: F = 1$ . The program prompts for $t_c$ .
<b>EXPONENTIAL</b>	$t < 0: F = 0$ $t \geq 0: F = 1 - \exp\left(\frac{-t}{t_c}\right)$ . The program prompts for $t_c$ .
<b>SINE</b>	$t < 0: F = 0$ $t \geq 0: F = \sin(2\pi ft - \phi)$ . The program prompts for $f$ and $\phi$ .
<b>STEP</b>	$t < 0: F = 0$ $t \geq 0: F = 1$
<b>TTOFF</b>	$t < 0: F = F(0)$ $t \geq 0: F = \text{cubic splines}$ The program prompts for the name of a time-table file. $F(0)$ is the value of the function in the table file at time $t=0$ .
<b>TTON</b>	$t < 0: F = 0$ $t \geq 0: F = \text{cubic spline}$ The program prompts for the name of a time-table file.

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 up to 1000 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. 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 switch-on cases, the function value at zero time need not be zero, but it is assumed that the function has value zero for all time before zero. Beyond the last value of time in the table the function continues with the same cubic function calculated for the last section of the table.

## The **LMMOTION** Command

---

### Menu Route:

MODEL↓

Draw regions → Linear motion air gap

### Command Line Parameters:

Sub-Command	<b>LMMOTION</b>		
Parameter	Default	Function	
<b>MESHING</b>	<b>NO</b>	<b>NO</b>	Mesh assuming that there will be no motion
		<b>YES</b>	Mesh using the groups specified
<b>MOVINGGROUP</b>	<i>none</i>	Group name for regions that are moving	
<b>MEDIUMGROUP</b>	<i>none</i>	Group name for the regions that allow movement	
<b>CENTREX</b>	0.0	Coordinate about which rotation occurs	
<b>CENTREY</b>	0.0		

### Notes

The **LMMOTION** command allows the mesh generation to allow for moving meshes required by the **LM** solver.

### Parameters

The **LMMOTION** command allows a more flexible mesh generation necessary for linear motion problems (LM). If **MESHING=YES** then this meshing technique will be used to generate the finite element mesh.

This mesh generation technique allows the positioning of a set of moving regions over another set of regions that allow movement. The moving regions are grouped together within the group defined in parameter **MOVINGGROUP**. The regions that allow movement are grouped together within the group defined in the **MEDIUMGROUP** parameter.

If rotational motion is required, the point of rotation is given by **CENTREX** and **CENTREY** parameters. If this is combined with linear motion, the centre of rotation is also moved.

## Use of the LM Gap region

The linear motion meshing must be used with the **LM** solver. It can also be used to generate meshes for other solvers. It is particularly useful where positional variations are being studied, as it allows modifications to a model to be simplified.

Some important features regarding **LMMOTION** meshing:

- Any number of regions may be grouped within the **MOVINGGROUP**, and where the regions within this group are joined along a region side, continuity of subdivision information must be enforced.
- Any number of regions may be grouped within the **MEDIUMGROUP**, and where the regions within this group are joined along a region side, continuity of subdivision information must be enforced. Mesh continuity with other regions must also be enforced as normal.
- All regions in the **MOVINGGROUP** must lie within the regions in the **MEDIUMGROUP**.
- There is no requirement for mesh continuity between regions in the **MOVINGGROUP** and those in the **MEDIUMGROUP**.
- Regions in the **MOVINGGROUP** may touch or share an external boundary of the model, but may not go beyond the boundary.
- Regions in the **MOVINGGROUP** must not touch any boundary of a region that is not part of the **MEDIUMGROUP**.
- Regions in the **MEDIUMGROUP** should all be defined with the properties of air.
- Periodicity and non-zero potential boundary conditions may not be used.
- It is recommended that the element size at the boundary of the **MOVINGGROUP** is comparable with the element size within the **MEDIUMGROUP**, so that the element edges forming the region boundary during the remeshing process are similar in size.

The following is a brief summary of the meshing process:

- All regions except those in **MOVINGGROUP** are meshed normally. Those in **MOVINGGROUP** are meshed independently.
- All elements in the **MEDIUMGROUP** that lie underneath the **MOVINGGROUP** regions are removed and the element edges forming the boundary of the hole found. The boundary edges of the **MOVINGGROUP** elements are added to this set.
- The region formed from this set of element edges is meshed, and the new elements added into the final mesh to form a gap between moving and static parts.

- The elements from the moving regions are also included in the final mesh.

During the LM solution process, the moving regions are repositioned and the final 2 stages of this process are repeated for each new position. It is therefore very important that the regions grouped within the **MEDIUMGROUP** extend to allow sufficient motion.

## The **SOLVE** Command

---

### Menu Route:

FILE↓  
Write file → Analysis data

### Command line Parameters:

Command	<b>SOLVE</b>	
Parameter	Default	Function
<b>TYPE</b>	<i>none</i>	Module to be used for the analysis of the model
		<b>AC</b> Steady-state AC analysis
		<b>LM</b> Linear Motion analysis
		<b>RM</b> Rotating Machine analysis
		<b>SA</b> Stress analysis
		<b>SP</b> Space charge analysis
		<b>ST</b> Static analysis
		<b>TH</b> Thermal analysis
		<b>THTR</b> Thermal transient analysis
		<b>TR</b> Transient time varying analysis
		<b>VL</b> Velocity analysis

The **SOLVE** command has a single parameter for the type of analysis module that is to be used for calculation of the field solution. This command must be executed before writing the analysis file, and the correct solver **TYPE** must be specified.

Other options are available as sub-commands within the solve command, allowing various parameters to be set for the analysis program.

<b>SOLVE Keyboard Sub-commands</b>	
Sub-command	Function
<b>CASE</b>	Give case data, either scale factors, frequencies or output times, for the analysis program
<b>DATA</b>	Give data specific to the options available in the analysis programs
<b>DRIVE</b>	Select the drive functions to be used by the drives in transient, linear motion and rotating machine analysis
<b>LOGFILE</b>	Create an additional output file from TR/LM/RM.
<b>QUIT</b>	Quit the <b>SOLVE</b> command

## CASE Sub-command

The **CASE** sub-command allows multiple cases to be set up and solved in a single analysis file. The different modules have different case types available.

Modules	Case type
ST/VL/SP	Scale factor used to multiply all drive sources (current density, non-zero potential boundaries)
AC	Frequencies for which the steady state harmonic analysis will be run
TR/LM/RM/THTR	Output times at which a solution will be stored
SA/TH	Multiple cases are not available

The solution file can contain multiple solutions (one for each case value set) that can be read in by setting the **CASE** parameter of the **READ** command.

### Command line Parameters:

Sub-command	<b>CASE</b>	
Parameter	Default	Description
<b>COMMAND</b>	<b>ADD</b>	Command option for generating the list of cases
		<b>ADD</b> Adds a new case value to the end of the list of cases
		<b>INSERT</b> Inserts a new case at the position given
		<b>REPLACE</b> Replaces the case value at the position given
		<b>DELETE</b> Deletes the case value from the position specified
	<b>LIST</b> Lists the current case list	
<b>POSITION</b>	<i>none</i>	Sets the position in the case list for inserting, replacing or deleting values
<b>VALUE</b>	<i>none</i>	Sets the value for adding, inserting or replacing

For output times (TR, LM and RM), all values in the list are ordered in numerical order, so inserting and adding may have the same effect. Any duplicate values are removed from the list. If adaption is being used, the adaption will happen on the first case in the list only. All subsequent cases will be solved using this mesh.

## DATA Sub-command

### Command line Parameters:

Sub-command	DATA			
Parameter	Default	Modules	Function	
<b>LINEAR</b>	<b>YES</b>	ST/ AC/ TR/ VL/ LM/ RM	Use linear analysis	
			<b>NO</b>	Use non-linear analysis
			<b>YES</b>	Use linear analysis
<b>NITERATIONS</b>	21	ST/ AC/ TR/ VL/ LM/ RM	Maximum number of non-linear iterations that will be used	
<b>TOLERANCE</b>	0.001	ST/ AC/ TR/ VL/ LM/ RM	Tolerance for convergence of the non-linear iterations	
<b>ITTYPE</b>	<b>NEWTON</b>	ST/ VL	Iteration type to be used for non-linear updates	
			<b>NEWTON</b>	Use Newton-Raphson non-linear updates
			<b>SIMPLE</b>	Use simple iteration updates
<b>CMU</b>	<b>NO</b>	AC	Use complex permeability	
<b>MUTYPE</b>	<b>DC</b>	AC	Use a DC or Incremental permeability	
			<b>DC</b>	Use a DC permeability
			<b>INCREMENT</b>	Use an incremental permeability

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
VELTYPE	LINEAR	VL	Use a linear or rotational velocity (see comment on units below)	
			LINEAR	Region velocity is linearly directed along the Y or Z axis
			ROTATION	Region velocity is interpreted as a rotational velocity around the origin
UPWINDING	NONE	VL	Selects the type of upwinding to be used	
			NONE	Do not use any upwinding
			HUGHES	Use Hughes upwinding (not available in axi-symmetric problems)
			STREAM	Use streamline upwinding
ADELEMENTS	*	ST/ AC/ VL	Maximum number of elements that will be used in the model when using adaptive refinement	
ADITERATIONS	0	ST/ AC/ VL	Maximum number of refinement iterations that will be used	
			0	Do not use any adaptive mesh refinement
ADACCURACY	5	ST/ AC/ VL	Final accuracy at which adaptive refinement will stop	
TSTEP	0	TR/ LM/ RM	Fixed time step	

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
<b>TSTOLERANCE</b>	0.001	TR/LM/ RM	Tolerance for adaptive time stepping	
<b>TSTYPE</b>	<b>ADAPT</b>	TR/LM/ RM	Selects the type of time stepping algorithm to be used	
			<b>FIXED</b>	Uses fixed time steps ( <b>TSTEP</b> )
			<b>ADAPT</b>	Adapts the time steps until the tolerance is reached
<b>RMVELOCITY</b>	3000	RM	Rotational velocity (RPM) for the rotating machines module	
<b>SPRELAXATION</b>	0.001	SP	Initial under-relaxation factor	
<b>SPITERATIONS</b>	30	SP	Maximum number of iterations that will be used	
<b>SPTOLERANCE</b>	0.001	SP	Tolerance for convergence of the space charge iterations	
<b>RESTART</b>	<b>NO</b>	ST/ AC/ TR/ VL/ LM/ RM/ SP	Restart from the current solution	
			<b>NO</b>	Use non-linear analysis
			<b>YES</b>	Use linear analysis
<b>SATYPE</b>	<b>STRESS</b>	SA	Selects the type of stress analysis to be used	
			<b>STRESS</b>	Use plain stress
			<b>STRAIN</b>	Use plain strain
			<b>AXISYMM</b>	Use axisymmetric stratified
<b>TSINITIAL</b>	0	LM/ RM/ TR	Initial time step for adaptive timestepping.	

Sub-command	DATA (continued)			
Parameter	Default	Modules	Function	
MOTIONTYPE	FIXED	LM/ RM	LM/ RM velocity:	
			COUPLED	Velocity calculated using mechanical coupling.
			FIXED	Constant velocity (RM only)
			VARIABLE	Velocity calculated from expression in a command input file
MOTIONCOMI		LM/RM	Name of command input file for COUPLED or VARIABLE velocity	
INERTIA	1	RM	Moment of inertia of rotor in COUPLED velocity.	
FTORQUE	0	RM	Frictional torque applied to rotor in COUPLED velocity.	
ATORQUE	0	RM	Torque applied to the rotor in COUPLED velocity	
VTORQUE	0	RM	Speed varying torque coefficient in COUPLED velocity	
LENGTH	0	RM	Magnetic length of the rotor.	

The data sub-command can be used to specify any options required by the chosen analysis program. The options that must be set will vary depending upon the analysis program.

### RM Options

The Rotating Machine Analysis program can be used with **FIXED**, **COUPLED** or **VARIABLE** velocity:

- **FIXED**: the value of **RMVELOCITY** is used for all time.
- simple **COUPLED**: the velocity is calculated by the program from the forces in the electromagnetic fields and the following mechanical properties of the rotor:
  - the **INERTIA**
  - the frictional torque (**FTORQUE**)

- an additional applied torque (**ATORQUE**)
- a speed varying torque (**VTORQUE**\*speed)
- the **LENGTH** of the rotor.
- complex **COUPLED**: More complex mechanical systems can be modelled using a command input file given by the **MOTIONCOMI** parameter. If the name of file is given, the file should calculate **#ACCEL**, the acceleration of the rotor from the system variables:
  - **RMTORQUE**: the torque per unit length from the electromagnetic fields.
  - **RMANGLE**: the position of the rotor (radians).
  - **RMSPEED**: the rotation speed (radians/second).
  - **TTIME**: the time (seconds).
  - **RMLENGTH**: the device length.
  - **RMINERTIA**: the moment of inertia.
  - **RMFTORQUE**: the frictional torque.
  - **RMVTORQUE**: the speed varying torque.
  - **RMATORQUE**: the applied torque.
  - the total torque
    - = $RMTORQUE * RMLENGTH + RMSPEED * RMVTORQUE + RMATORQUE - RMFTORQUE$
    - with the sign of **RMFTORQUE** set so that it opposes motion.

A command input file equivalent to the default mechanical system would contain:

```
$CONSTANT #TORQUE1 RMTORQUE*RMLENGTH
$CONSTANT #TORQUE2 RMSPEED*RMVTORQUE
$CONSTANT #TORQUE3 RMATORQUE-RMFTORQUE
$CONSTANT #TORQUE #TORQUE1+#TORQUE2+#TORQUE3
$CONSTANT #ACCEL #TORQUE/RMINERTIA
```

A more complex mechanical system could be modelled using a command input file containing:

```
$CONSTANT #TORQUE1 RMTORQUE*RMLENGTH
$CONSTANT #TORQUE2 RMSPEED*RMVTORQUE
$CONSTANT #TORQUE3 RMATORQUE-RMFTORQUE
$IF RMSPEED LE 3600/60*2*PI
  $CONSTANT #TORQUE #TORQUE1+#TORQUE2+#TORQUE3
$ELSE
  $CONSTANT #TORQUE #TORQUE1-RMFTORQUE
$END IF
$CONSTANT #ACCEL #TORQUE/RMINERTIA
```

- **VARIABLE**: the speed of the rotor is calculated using a command input file given by **MOTIONCOMI** which specifies the speed as a function of time, in radians/second, e.g.

```
$constant #SPEED 100*TTIME
```

Full details of the RM facilities are given in the User Guide.

### *LM Options*

The Linear Motion Analysis program can be used with coupled or variable velocity:

- **COUPLED**: Complex mechanical systems can be modelled using a command input file given by the **MOTIONCOMI** parameter. The file should calculate **#ACCELX**, **#ACCELY** and **#ROTACCEL**, the acceleration of the rotor from the system variables:
  - **LMXFORCE**, **LMYFORCE**, **LMZFORCE**, **LMTORQUE**: the force and torque from the electromagnetic fields.
  - **LMXSHIFT**, **LMYSHIFT**, **LMZSHIFT**, **LMROTANGLE**: the position of the moving part.
  - **LMXSPEED**, **LMYSPEED**, **LMROTSPEED**: the speeds.
  - **TTIME**: the time (seconds).
- **VARIABLE**: the speed of the moving part is calculated using a command input file given by **MOTIONCOMI** which specifies the speed as a function of time, e.g.
 

```
$constant #speedx 100/ttime
$constant #speedy 0
$constant #rotspeed 0
```

Full details of the LM facilities are given in the User Guide.

## **DRIVE Sub-command**

The **DRIVE** sub-command is for use only with the transient, linear motion and rotating machine analysis modules, and can be used to set the list of drive functions applied to the current sources and non-zero potential boundaries.

*Command line  
Parameters:*

Sub-Command	<b>DRIVE</b>		
Parameter	Default	Description	
<b>NUMBER</b>	<i>none</i>	Number of the drive function to be set	
		<b>LIST</b>	Option to output the list of drive numbers that can be set
<b>TYPE</b>	<i>none</i>	Type of drive that will be used for the selected drive number	
		<b>DC</b>	DC drive, $f(t) = 1$
		<b>STEP</b>	Step at $t=0$ $f(t) = 0: t < 0$ $f(t) = 1: t \geq 0$
		<b>RAMP</b>	Ramps from $f(t)=0$ at $t=0$ to $f(t)=1$ at $t=val1$
		<b>COSINE</b>	Cosine function of frequency $val1$ , with phase $val2$
		<b>SINE</b>	Sine function of frequency $val1$ , with phase $val2$
		<b>EXPONENTIAL</b>	Exponential increase with time constant given by $val1$
		<b>PEAK</b>	Function generating a peak at time given by $val1$
		<b>TTON</b>	Use a timetable of points with $f(t)=0$ at $t < 0$
<b>TTOFF</b>	Use a timetable of points with $f(t)=f(0)$ at $t < 0$ $f(0)$ is the value of the function in the table file at time $t=0$ .		
<b>VAL1</b>	<i>none</i>	First value for the different functions	
<b>VAL2</b>	<i>none</i>	Second value for the functions (only used by <b>COSINE</b> and <b>SINE</b> )	
<b>FILE</b>	<i>none</i>	File name for use with timetable drive types <b>TTON</b> and <b>TTOFF</b>	

The program generates a list of all conductor labels that have been defined, together with any external circuits that have been created, and assigns each of these a number. This list can be seen by using the command **DRIVE NUMBER=LIST**

The drive type for any of these can be set by the **DRIVE** command, together with the correct parameters. A default drive is also available. This is used to set any

drive that has conductor number 0, as well as any of the drive labels that has no drive function specified.

Time-table files should be created by an editor. They consist of 2 columns of numbers in free-format, the first giving values of time and the second of function value. Consecutive lines with the same value of time introduce a discontinuity of function value or time-derivative into the cubic-spline fitting. The data in time-table files can be displayed using the pre and post processor **GRAPH** command

## **LOGFILE Sub-command**

### *Command line Parameters:*

Sub-Command	<b>LOGFILE</b>		
Parameter	Default	Description	
<b>FILE</b>	<i>none</i>	Name of log file.	
<b>COL1</b>	<b>TTIME</b>	Expression for column 1 of log file.	
<b>COL2</b>	<b>RMANGLE</b>	Expression for column 2 of log file.	
<b>COL3</b>	<i>none</i>	Expression for column 3 of log file.	
<b>COL4</b>	<i>none</i>	Expression for column 4 of log file.	
<b>COL5</b>	<i>none</i>	Expression for column 5 of log file.	
<b>COL6</b>	<i>none</i>	Expression for column 6 of log file.	
<b>COL7</b>	<i>none</i>	Expression for column 7 of log file.	
<b>COL8</b>	<i>none</i>	Expression for column 8 of log file.	
<b>LOGGING</b>	<b>NO</b>	Log file options:	
		<b>NO</b>	No log file
		<b>SOLUTION</b>	Logging to file with name derived from solution file name.
		<b>YES</b>	Logging to <b>FILE</b>

The **LOGFILE** sub-command is for use only with the transient, linear motion and rotating machine analysis modules, and can be used to request an additional output file which tabulates the progress of the analysis. The command specifies the name of the file and up to 8 system variables or expressions which will be evaluated and added to the file at every time step.

The system variables available are those which can be used in coupled velocity command files (See “RM Options” on page 3-26. See “LM Options” on page 3-28.).

## **QUIT Sub-command**

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



# *Chapter 4*

## **OPERA-3d**

### **Overview**

---

---

The major new feature of OPERA-3d Version 8.5 is the new Graphical User Interface for the post processor which has been updated to match the Modeller.

The Modeller has also been enhanced with, among other things, blending and chamfering, and features to specify periodicity for TOSCA and SCALA.

The rest of this Chapter gives an overview of the changes in the Modeller and the post processor. Chapter 5 gives more detailed information on the new features in the Modeller. Chapter 6 includes the post processing section from the 3D Getting Started, showing the use of the new post processor.

# Modeller

---

## Periodicity

Periodicity has been implemented within the modeller. This allows specification of geometric and field symmetry to allow only a section of the model to be built. The mesh at paired faces is matched so that the periodicity can be implemented. This feature may be used by both TOSCA and SCALA.

## Automatic potential cuts

Automatic potential cuts have been implemented for TOSCA. The cuts allow automatic resolution of multiply connected potentials.

In-built pre processor and Modeller routines will correct a TOSCA model that contains multiply connected regions. The software has the capability of checking for multi-valued total magnetic scalar potential regions, and introducing potential “cuts” where appropriate in order to rectify the problem.

This information is stored in the database and can be used by TOSCA to calculate the correct potentials and field distribution. The use of cuts in TOSCA can be disabled by a new parameter, **POTENTIALCUT** in the **SOLVERS SETTINGS** sub-command.

The ELEKTRA solver is completely unaffected by these functions. It is therefore still necessary for the user to produce non-multiply connected models.

## BLEND Command

This command allows blending and chamfering of faces at edges. Two faces joining at an edge can be blended by forming a curved face of fixed radius that joins both faces tangentially. Alternatively, the edge can be chamfered by giving a distance from the edge along each face at which to plane off the edge. See “[Blend and Chamfer Edges](#)” on page 5-6.

## Cutaway option

A new cutaway option has been added to the **COMBINE** command, allowing the user to cut one body from many. **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. See “[Cutaway Overlap With Regularisation](#)” on page 5-1.

## Improved copying and transformations

Conductor copying is now applied to local coordinate system 1 by default, unless this is the working coordinate system. During copy and transformations, additional labels can be added to the items being copied, so that they can easily be grouped for later operations.

## NULL material label

Regions of the model can now be excluded from the model by giving them a material label of **NULL**. Any element with **NULL** material label will not form part of the finite element solution.

## WINDOW, MOUSE and TITLE commands

Greater control of the display and operation of mouse has been implemented in command mode, so that such operations may be scripted. A new **TITLE** command also allows a title to be added to the display.

## Mesh Generation

### *Surface Mesh*

The faces of all cells are meshed sequentially. On execution of the **MESH** command, the outline frame of the face being processed is highlighted. A progress bar is also displayed at the bottom of the Modeller.

The **MESH** command may take some time for complex models. On completion the status bar at the bottom of the window displays the message “Surface Mesh”.

### *Volume Mesh*

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 highlighted. A progress bar is displayed at the bottom of the Modeller window.

The **FILL** command may take some time for complex models. On completion the status bar at the bottom of the window displays the message “Volume Mesh”.

## Post Processor

---

### GUI

The OPERA-3d post processor has been updated to use the same graphical user interface as the Modeller. This gives a look and feel which matches other software running under Microsoft Windows or UNIX. To make use of the features of the GUI, several commands have been changed, added or removed.

### Selection Commands

Commands which operate on a selection of objects (**BODY**, **CONDUCTOR**, **ENERGY**, **SELECT** and **VOLUME**) have improved facilities for creating the selection before performing their operations. For these commands there is a two stage process: create a list of labels, then operate on that list. The **BODY** command uses the list created by the **CONDUCTOR** command.

1. Create a list of labels: labels can be
  - added to the list (**ACTION=ADD LABEL=name**)
  - removed from the list (**ACTION=REMOVE LABEL=name**)
  - all removed from the list (**ACTION=RESET**)
  - added or removed in groups with additional 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_MATERIALS</b>	all material names
<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

- The **SELECT** command has all of the above options; other commands have the subset which applies to conductors or to volumes as appropriate.

- For conductors, the label name can be either the conductor number or the drive label.
  - The **CONDUCTOR** command also allow conductors to be selected using the mouse with **ACTION=PICK**.
2. Operate on the list of labels:
- **BODY**
  - **CONDUCTOR ACTION=ERASE, EXPORT, LIST or MODIFY**
  - **ENERGY ACTION=INTEGRATE**
  - **SELECT ACTION=SELECT**
  - **VOLUME ACTION=INTEGRATE**

Figure 4.1 shows how these options are presented to the user for the **SELECT** command. Similar Dialog Boxes are used for the other commands.

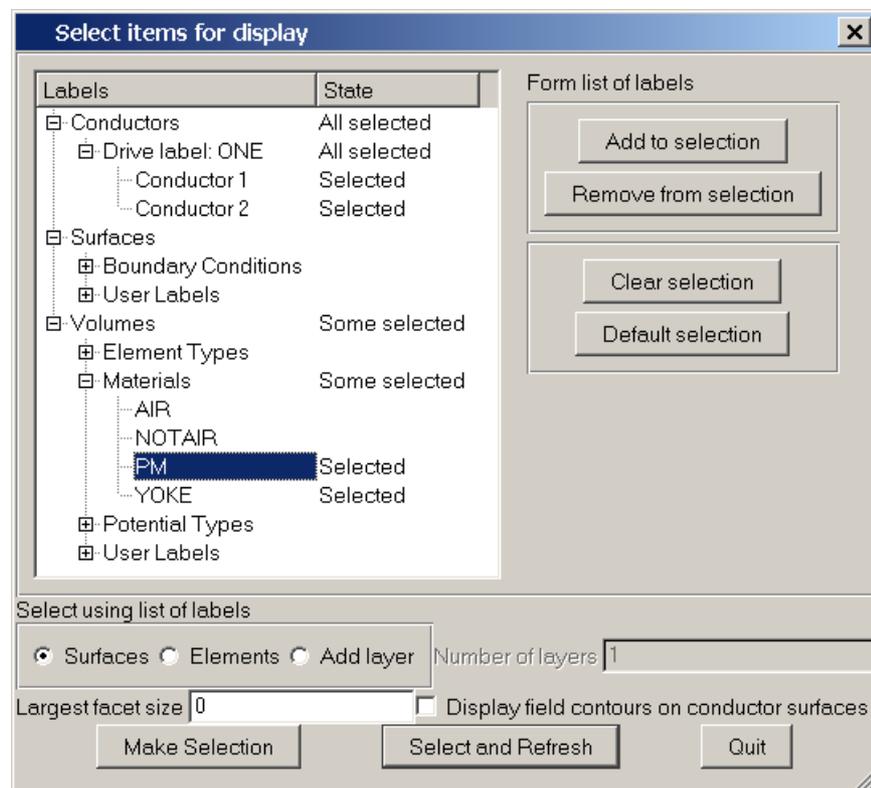


Figure 4.1 The **SELECT** Command Dialog Box

## ACTIVATE and LOAD Commands

The File menu contains a Recent Files option to activate and load databases from previous runs.

**ACTIVATE** will now accept **CASE=\*** to indicate the last simulation in a database. This means that any user variable or expression used for **CASE** must be specified using the **%INT** function, e.g.

```
ACTIVATE CASE=%int (#number).
```

A new system variable (**CASES**) holds the number of simulations in a database. It is set each time a database is activated.

**ACTIVATE** now reads the units from the database and interprets the local coordinate system origin parameters (**XORIGIN**, **YORIGIN**, **ZORIGIN**) in the database length unit.

**LOAD 0** can be used to close the currently loaded database.

## BHDATA Command

The **BHDATA** command no longer cycles through all 3 directions of anisotropic materials. It has a new parameter, **DIRECTION**, which can be set to **X**, **Y** or **Z** to select the appropriate curve.

## BODY Command

The **BODY** command operates on the conductors selected by the **CONDUCTOR** command (See “Selection Commands” on page 4-4.) or on all conductors if the list is empty.

## CLEAR Command

The **CLEAR** command now removes all user variables.

## COLOUR Command

The colour command has been simplified. Colours are referenced by label names (**BACKGROUND**, **TEXT**, **CONDUCTORS**, **VECTORINSIDE**, **VECTOROUTSIDE**, **FIRSTCONTOUR**, **LASTCONTOUR**, **TRAJECTORIES** and material names). Databases created by version 8.1 or later of the

Modeller or pre processor contain the colours used for each material and the conductors. These colours are read into post processor when the database is loaded.

The **RED**, **GREEN** and **BLUE** components of the new colour should be given in the range 0 to 255.

It is no longer necessary to set the maximum number of material or contour colours.

## **CONDUCTORS Command**

The **CONDUCTORS** command has been rewritten as a single command with **ACTIONS** to **DEFINE**, **MODIFY**, **ERASE**, **LIST**, **IMPORT** and **EXPORT**. The **ERASE**, **EXPORT**, **MODIFY** and **LIST** actions operate on a previously selected list of conductors (See “Selection Commands” on page 4-4.).

The **IMPORT** and **EXPORT** actions use files which are compatible with the Modeller and the pre processor. Changes to conductors are immediately reflected in the display.

## **DEVICE, DISPLAY and DUMP Commands**

The **DEVICE**, **DISPLAY** and **DUMP** commands have been removed. The functionality of **DEVICE** and **DUMP** is now included in **PICTURE** (page 4-8) and **PRINT**(page 4-8). The **THREED** command (page 4-9) has been enhanced to include additional features of **DISPLAY**.

## **ENERGY Command**

The **ENERGY** command has a new way of selecting which volumes are included in the integrations (See “Selection Commands” on page 4-4.).

## **HELP Command**

The **HELP** command has been removed. The GUI has a Help menu which gives access to on-line manuals.

## HISTOGRAM and MAP Commands

The **HISTOGRAM** command has been amalgamated with the **MAP** command. The **MAP CONTOUR** parameter controls the style of contour map: **ZONES**, **LINES**, **HISTOGRAM** or **NONE**. **ZONE** and **HISTOGRAM** contour maps are displayed with translucent colours so that parts of the model behind the map show through (figure 4.2). Multiple maps can be created and displayed. The new

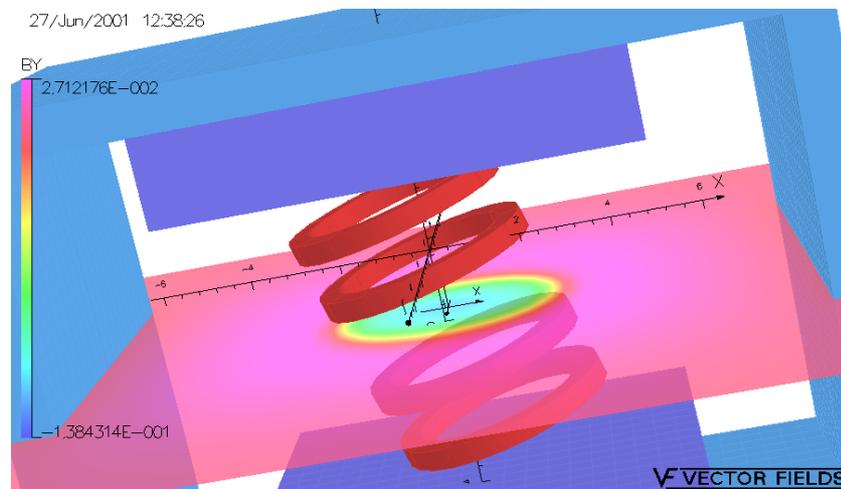


Figure 4.2 A Contour Map

**ERASE** parameter can be used to request that existing maps are deleted.

## MOUSE Command

The new **MOUSE** command swaps the functionality of the middle and right mouse buttons so that the software can be used with a 2-button mouse.

## PICTURE Command

The new **PICTURE** command copies the picture to the clipboard or a bitmap file.

## PRINT Command

The old **PRINT** command is now part of the **TABLE** command.

The new **PRINT** command sends the current picture to a printer.

## SELECT Command

The **SELECT** command has a new way of selecting which parts of the model are included in the display (See “Selection Commands” on page 4-4.). It has one additional action, **ACTION=DEFAULT**, which selects according to the simulation type. For example, it would select all conductors and all materials except for **AIR** for TOSCA magnetic field and ELEKTRA analyses.

Now that conductors can be selected or omitted individually using this new technique, the **COIL** parameter is only used to specify whether field quantities are to be displayed on the surfaces of conductors.

When the list of labels has been formed, the **ACTION=SELECT** can be used with the **OPTION** parameter to select **SURFACES** or **ELEMENTS** or to **ADD** a number of **LAYERS** of elements to the selected surface. Layers can also be removed by specifying a negative number.

The **ZTOLERANCE** parameter is now used to improve displays with **CUT=FRONT** and **CUT=BACK**.

Subdivision of facets using the **ACCURACY** parameter has been restricted to conductor facets only.

## SYSVARIABLE Command

The **SYSVARIABLE** command has been enhanced with options **UNIT=DEFAULT** and **REFLECTION=DEFAULT**. For all the system variable names created by the analysis programs, the post processor knows the unit expression and reflections to apply. The user should only specify a non-default unit or reflection when adding a system variable previously added to the database using the **TABLE** command.

## TABLE Command

The **TABLE INFILE=SELE** command has taken over the function of the old **PRINT** command to output a table of field values at the coordinates of the facets which make up the display.

## THREED Command

The **THREED** command has been enhanced with view parameters (**SIZE**, **ROTX**, **ROTY**, **ROTZ**, **XORIGIN**, **YORIGIN**, **ZORIGIN** and **PERSPECTIVE**)

which can be used with **OPTION=SETVIEW** to specify a particular view of the model.

After the view has been changed using the mouse, **OPTION=GETVIEW** can be used to update the values of the view parameters to match the current view. **OPTION=REFRESH** recreates the display (if necessary) without changing the view.

The 3d Viewer Menu has been replaced by the **WINDOW** command and by making use of the mouse buttons: left for rotate, middle for zoom and right for translate. Predefined views of the model are also available using toolbar buttons.

## **VOLUME** Command

The **VOLUME** command has a new way of selecting which volumes are included in the integrations (See “[Selection Commands](#)” on page 4-4.).

## **WINDOW** Command

The new **WINDOW** command can be used to switch on or off parts of the display: solid view of the model, outline view of the model, contour map, vector map etc. It provides the functionality of the 3d Viewer Menu **Options** of earlier versions.

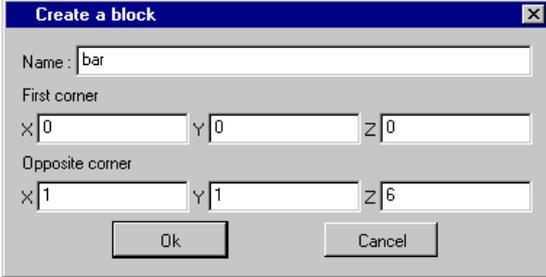
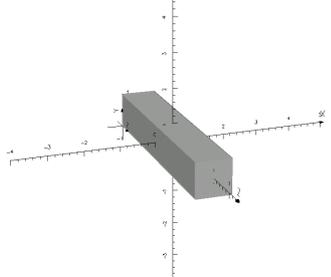
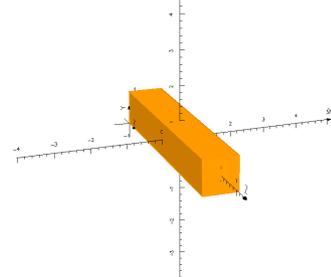
The options are also available using toolbar buttons.

# Chapter 5

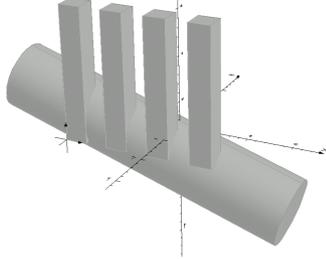
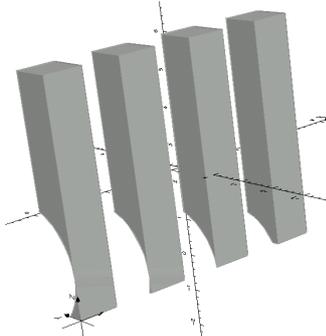
## New Modeller Features

### Cutaway Overlap With Regularisation

The new operation is **Cutaway Overlap** which is very similar to the **Trim Overlap** operation. The difference is that a cylinder is used as a cutting tool to cut the overlapping parts from the 4 rectangular bars in the following. The operation is useful when several entities shall be trimmed in one step. The cylinder has to be picked first, followed by the other bodies that overlap, then the **Cutaway Overlap** command can be issued.

<p>Create a Block</p> 		
<p>Pick bodies</p>  	<p>Select the block by double clicking with the left mouse button.</p>	

Operations → Copy		
<p>Create a Cylinder</p>		
<p>Pick bodies</p>	<p>Using the mouse select the cylinder first, then the four bars.</p>	

Operations → Combine bodies → Cutaway Overlap with regularisation		
	The resultant geometry still contains 5 bodies but the overlapping parts between the cylinder and the 4 bars have been cut away from the 4 bars.	
Hide bodies 	By now hiding the cylinder, the result of the operation can be seen more clearly. The cylinder has been left untouched, whereas the 4 bars have been trimmed.	

## Cutaway Overlap Without Regularisation

The **Cutaway Overlap** operation can also be applied without regularisation. Similar to the previous example this operation produces the same result to that achieved in the operation with regularisation, as no faces are shared between the cylinder and the 4 bars. The operation can be tried out quickly by pressing the **Undo** button twice, which reverts to the status before the last operation, and then applying the **Cutaway Overlap without regularisation**.

Before moving to the next section, clear all the data using:

**FILE** ↓  
Close

## Building a Library of Sub-Models

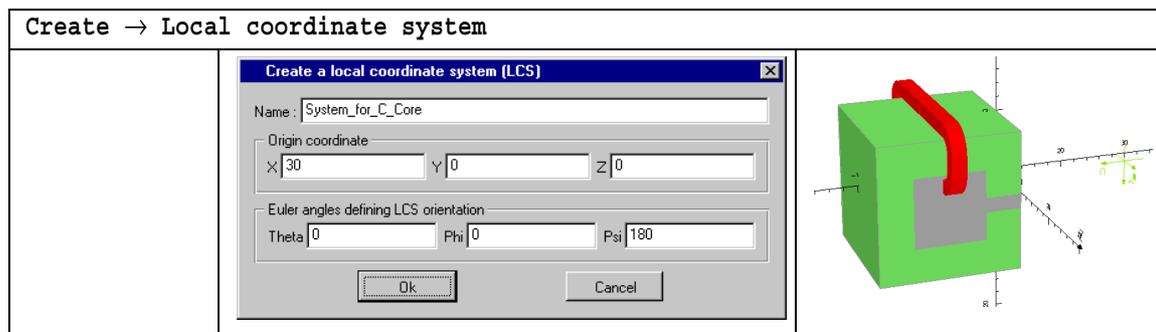
The Modeller allows users to build a model from several parts, which can be created and saved separately. This example uses the C-core magnet, which is described in the OPERA-3d User Guide. The data file (.opc) is provided in the OPERA installation (see the sub-folder *Examples/3D*) with the background region omitted.

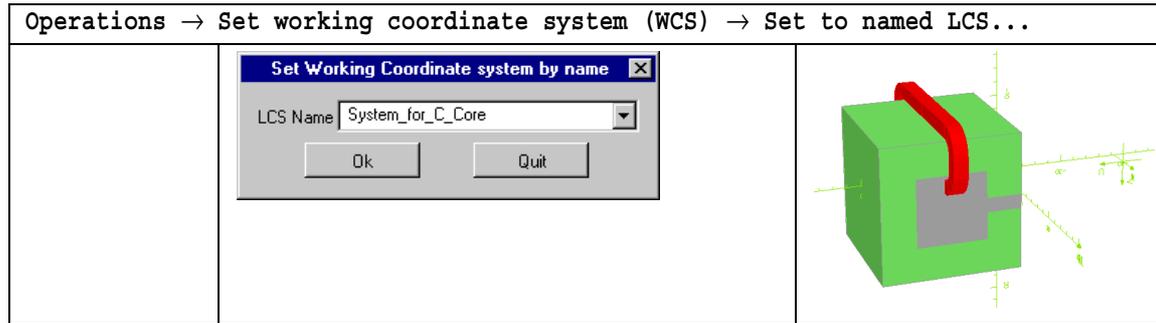
The geometry of a model or parts of a model can be saved in two different types of files (.opc file or .sat file). The main differences are that a .sat file cannot hold any conductor, magnetic material, boundary condition or volume data. The .sat files are the format of geometric model that allows data exchange between ACIS based CAD systems. The .opc files are Vector Fields extension to .sat to support the OPERA analysis programs. Either type of file can be used to build up a model from parts.

Before starting this example, the Modeller should be reset to its initial state using **File** → **Close**.

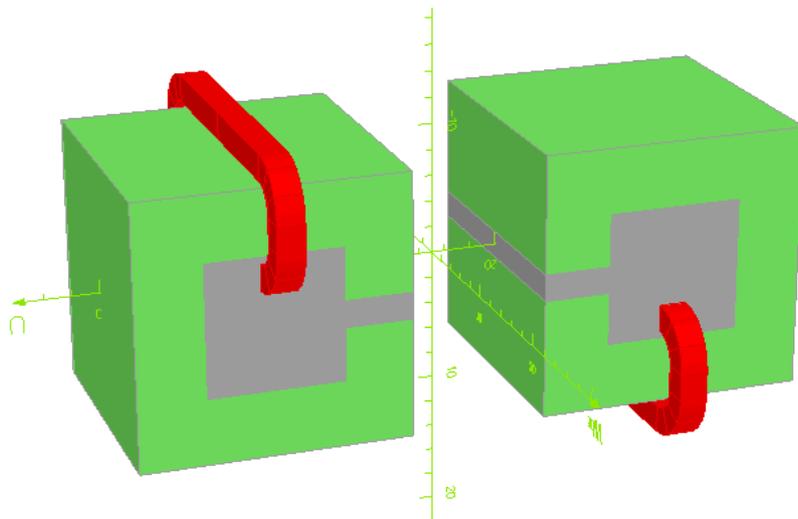
Load the C-core example using the **File** **Open** icon:  and selecting the file from the appropriate directory. Use **Model** → **Delete Model Body**, and then delete the Background region.

Following this, the same C-core will be loaded a second time into the Modeller. A new local coordinate system will be used to displace the second copy and to rotate it by 180 degrees at the same time.





The C-core is again loaded into the Modeller, using: **Create** → **Insert from file...** and selecting the appropriate file.



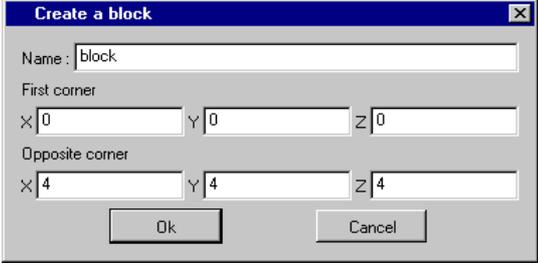
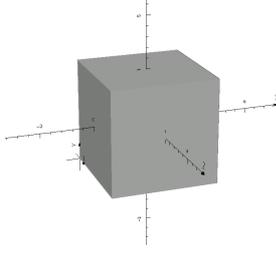
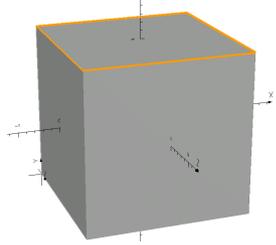
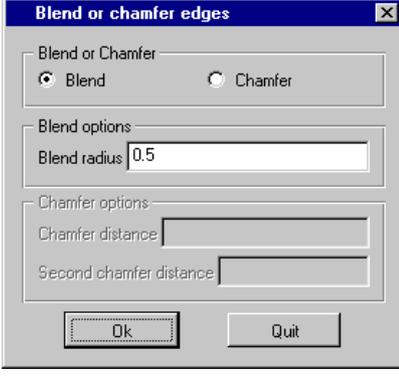
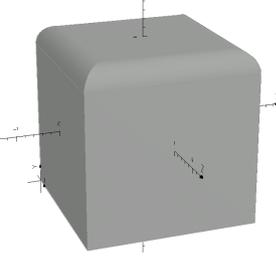
Based on this procedure, very complex models can be built up using parts from a user defined library.

## Blend and Chamfer Edges

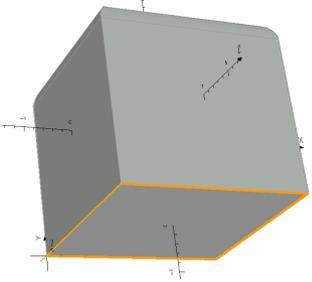
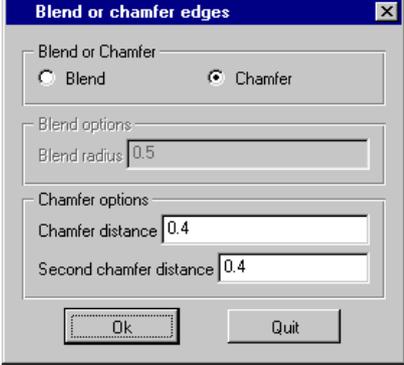
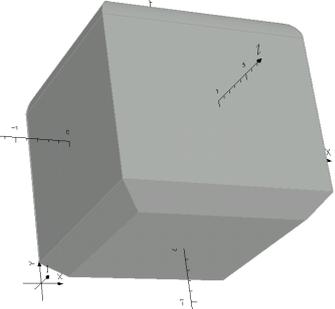
This example is self contained and needs no previous models. Therefore the Modeller should be reset to its initial state using **File** → **Close**.

**Blend** and **Chamfer** are both operations which can be applied to edges of a volume. The **Blend** operation smooths an edge using a radius, whereas the **Chamfer** operation smooths an edge by using a plane.

If the operations are applied to an “outer” edge, material will be cut away; if there is an “inner” edge, material will filled in. This gives a total number of 4 combinations, which all will be explained in this example.

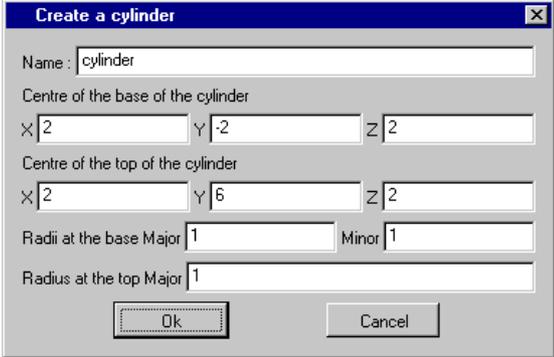
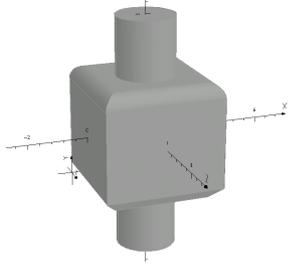
<p>Create a block</p> 		
<p>Pick edges and Pick entity</p>  	<p>Zoom in using the mouse and select the 4 edges on top of the block. The selected edges will turn orange.</p>	
<p><b>Operations</b> → <b>Blend or chamfer edges...</b></p>		
		

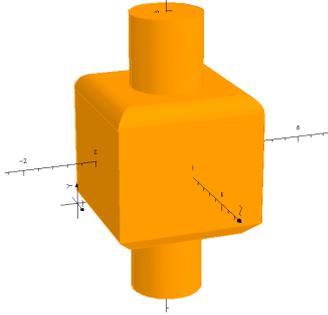
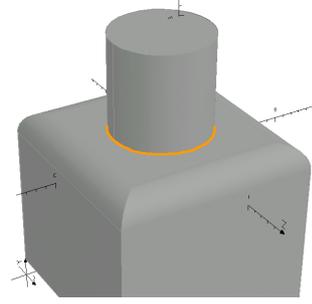
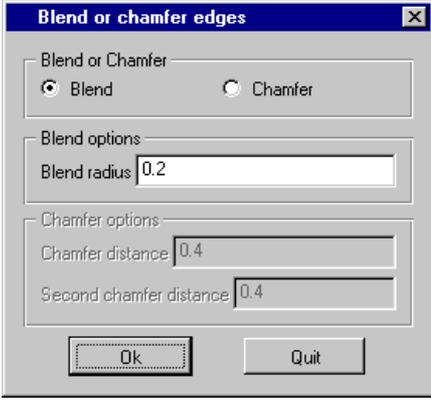
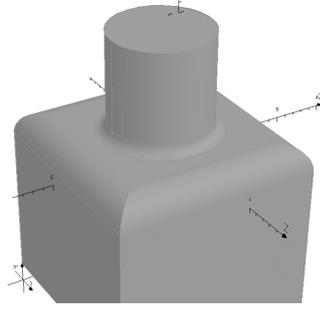
In the next operation the bottom of the block will be chamfered.

<p>Pick edges and Pick entity</p> 	<p>Rotate the block using the mouse, and select the 4 edges at the bottom of the block. The selected edges will turn orange.</p>	
<p><b>Operations</b> → <b>Blend or chamfer edges...</b></p>		
		

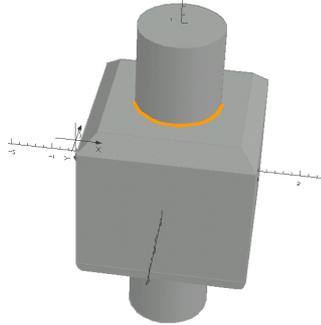
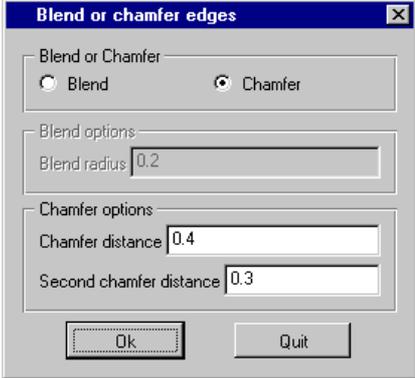
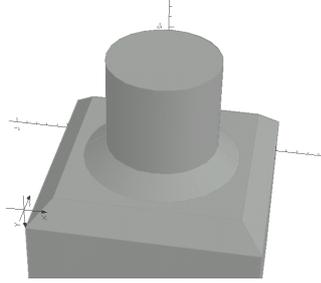
The second **chamfer distance** in the example above has been chosen to be the same as the first one, which will cut a 45 degrees section off the block. It is also possible to specify different values.

In the second half of this section a new cylinder will be defined, which cuts through the block. This allows **blend** and **chamfer** at an “inner” edge.

<p>Create a cylinder</p> 		
--	--	---

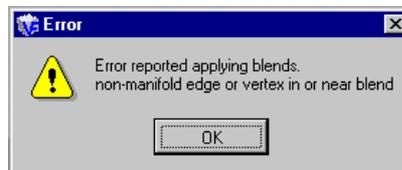
<p>Pick body and Pick entity</p> 	<p>Pick the cylinder and the block using the mouse. The sequence is not important as the next operation will be a <b>Union with regularisation</b>.</p> <p>Note that it is necessary to choose <b>Union with regularisation</b> so that only a single cell is created, and to ensure all edges have no more than two faces touching them (the edges are not manifold).</p>	
<p><b>Operations</b> → <b>Combine bodies</b> → <b>Union, with regularisation</b></p>		
<p>Pick edge</p> 	<p>Select the edge on top of the cube, where the cylinder meets the cube, as indicated on the right picture.</p> <p>It does not matter whether a 180 degrees section or both 180 degrees sections are selected; the program will always perform the operation on the whole circle (a feature of the underlying ACIS kernel).</p>	
<p><b>Operations</b> → <b>Blend or chamfer edges...</b></p>		
		

The last operation will be a **Chamfer** of the bottom junction. Use the mouse to rotate the body and select the edge where the cylinder meets the cube.

<p>Pick edge</p> 	<p>Select the edge at the bottom of the cube, where the cylinder meets the cube, as indicated on the right picture.</p>	
<p><b>Operations</b> → <b>Blend or chamfer edges...</b></p>		
		

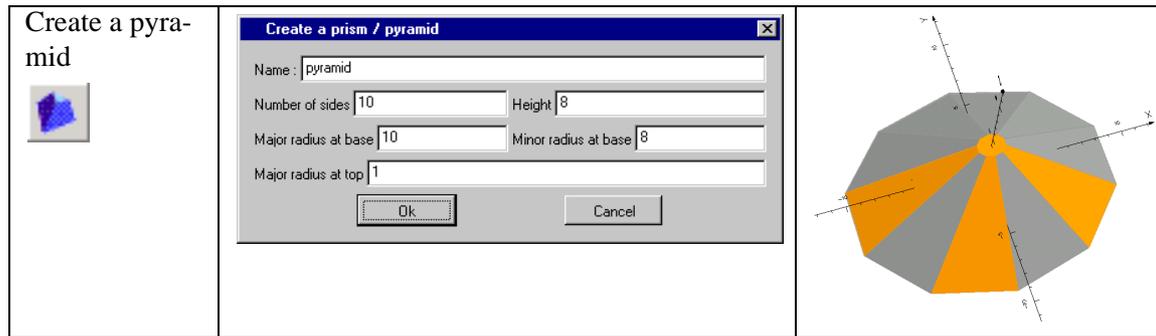
The **Blend** and **Chamfer** operations can only be applied if an edge is not manifold i.e. where more than two faces meet at an edge.

This can be demonstrated by using **Undo**  to return to the **Boolean Union** and making the **Union, without regularisation**. This will create 4 cells in the body and the edge at the junction of the cylinder and block will now belong to four faces. When the chamfer or blend is applied the user will receive a warning message.

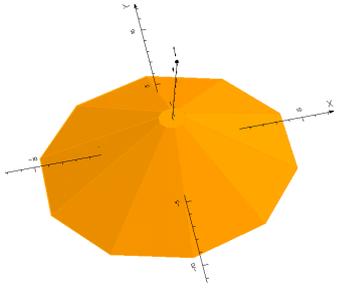


## Swift Selection of Entities

Reset the Modeller to its initial state using **File** → **Close**.



This pyramid has many faces, and selecting them one by one needs 12 mouse double-clicks. This can be reduced to 1 mouse double-click and a simple operation by the following procedure.

<p>Pick body and Pick entity</p> 	<p>Select the pyramid as a body.</p>	
<p>Pick face</p> 	<p>Press the <b>Pick face</b> button. The pyramid is still selected as a body.</p> 	
<p>Change type of picked entities</p> 	<p>Press the <b>Change type of picked entities</b> button. As a result of this operation all faces on the selected body are selected instead. This can be seen on the display of the entities, which are listed on the lower right hand side of the Modeller window.</p> 	

The procedure can be very useful for example in electrostatic models, where the faces of a body have to be selected to apply a label for voltage boundary conditions. Another application might be to select all edges of a body in one go and apply **Blend** or **Chamfer** operations.

## Chapter 6

# Getting Started - Post Processing

## Post Processing the Getting Started Model

---

The post processing shown here is typical of the major requirements for an MRI system. Many other features are available in the post processor and are covered in other worked examples in the OPERA-3d User Guide.

This example uses the MRI model, which is described in the OPERA-3d User Guide Getting Started. The data file (*.opc*) is provided in the OPERA installation (see the sub-folder *Examples/3D*) with the background region omitted. This file can be used to prepare a database, and run the TOSCA analysis to obtain a solution for post processing. The following script (taken from the OPERA-3d Getting Started) perform the post processing on this model.

### Microsoft Windows Platforms

From the **OPERA** Console window, select **Post-Processor** from the **OPERA-3d** menu.

### Unix Platforms

The **OPERA** window offers a choice of **OPERA-3d** modules again. The user enters **post**.

### Loading the Solution

When the post processor starts, the display shown in Figure 6.1 is obtained.

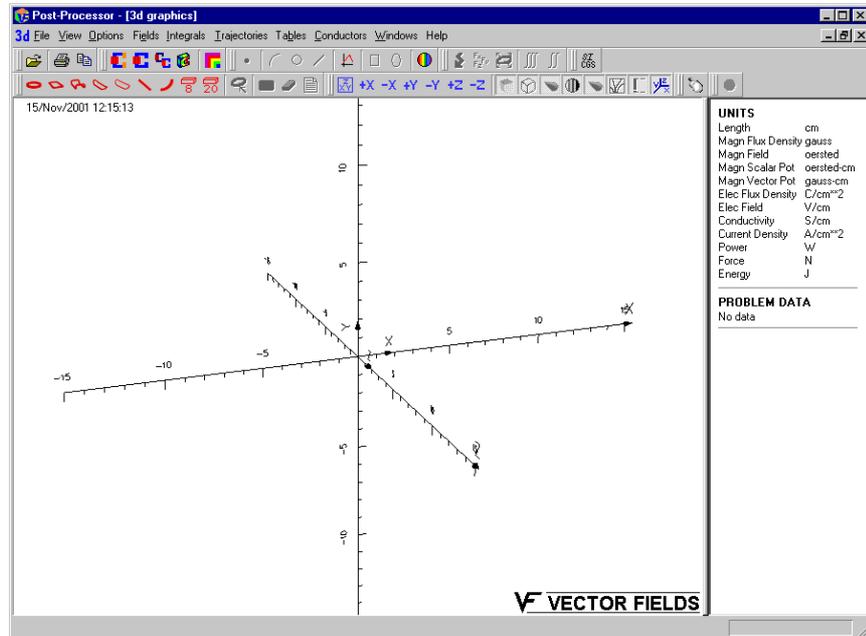
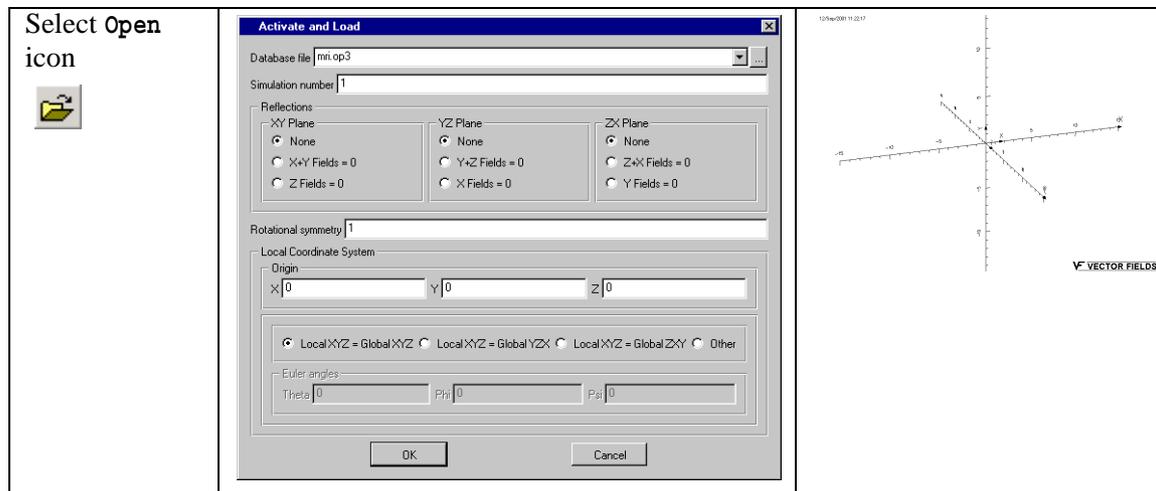


Figure 6.1 Initial OPERA-3d post processor window

The post processing features are accessed from the menubar or from icons in a similar way to the Modeller. Many of the icons are similar to the Modeller and perform the same function. Keyboard access is also obtained in the same way.

Load the solution file as follows:

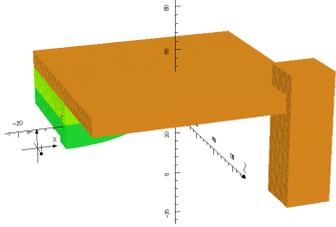


Select the database, *mri.op3* using the browser. Double clicking on the filename or clicking on **Open** in the browser enters the file name in the **Database File** box.

Click on **OK** to load the file. Note that a message is given in the lower left corner of the window to say that a TOSCA magnetostatic analysis has been loaded. Details of the analysis data are given on the right side of the graphics area. However, the display area remains unchanged.

## Displaying the Model

---

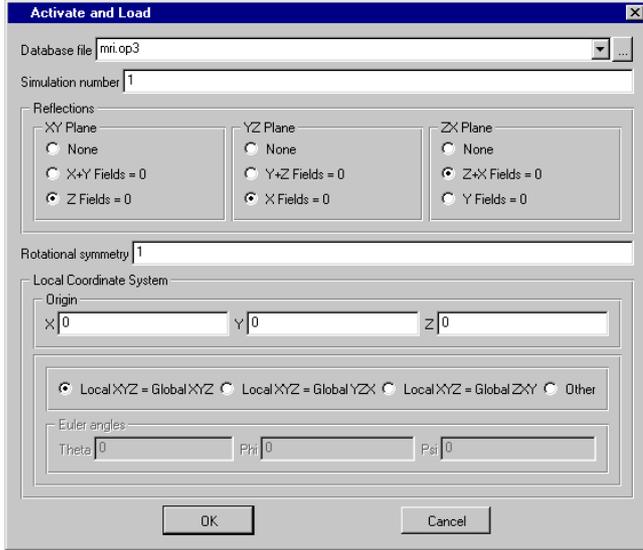
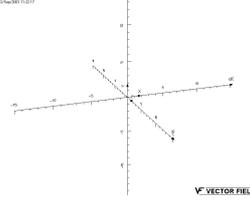
<p>SelectDefault select and refresh icon</p> 	<p>This will select all materials not called <b>air</b> for display by default. Note that the names of the materials used in the Modeller - <b>goods-teel</b>, <b>cheapsteel</b> and <b>neo</b> - are selected. The operation of the 3d-Viewer in the post processor has the same functionality as in the Modeller. A timer bar shows how the selection and display process is proceeding.</p>	
--	--	---

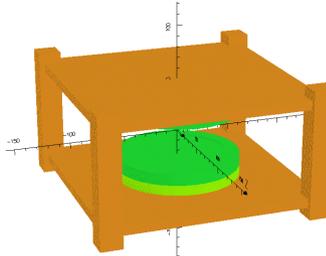
### Replicating the Model

For computational efficiency, only 1/8 of the MRI and the surrounding free space was solved in TOSCA. The post processor allows the user to replicate the model to obtain the complete magnet - which will be necessary to perform some of the post processing calculations.

The reflections not only indicate the copies of the solved geometry but how the magnetic field should be replicated in the symmetric copies. Clearly, they should match the boundary conditions defined in the Modeller for correct display of the results.

Use the radio buttons to create the full magnet. Then click on **OK**. Note that this removes the previous display of the geometry.

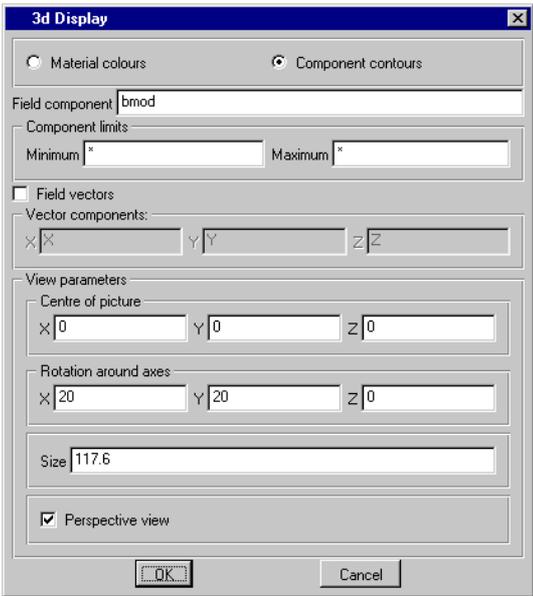
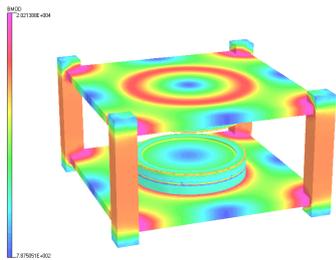
<p>Select Open icon</p> 		
---	--	---

<p>Select Default select and refresh icon</p>  <p>Select Initial view icon</p> 	<p>The initial view icon allows the complete geometry to be seen.</p>	
---	---	--

## Checking the Solution

### Results on the Geometry

Results can be superimposed on the geometry. Component BMOD gives the magnitude of the magnetic flux density

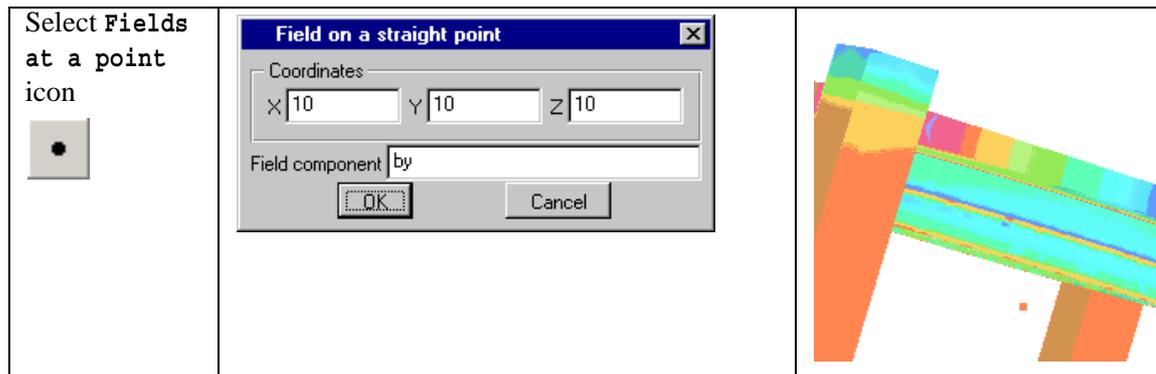
<p>Select 3D display icon</p> 		
<p>Toggle Outline view of model icon and Axes icon</p>  	<p>The surface finite element mesh is initially displayed over the geometry as well as the contours. The mesh can be toggled on and off, as can the coordinate system axes and contour labels.</p>	

The scale to the side shows the magnitude of the flux density in Gauss. The user should check that the flux density values are about as expected. The maximum value of the flux density should be about 20 kGauss and the minimum close to zero.

The user should also check that the direction of the field is correct - both in the original model and the images. This can be achieved using field vectors in the 3D display but for this model a simpler check can be made.

## Fields at a Point

To interrogate the field at a specific point:



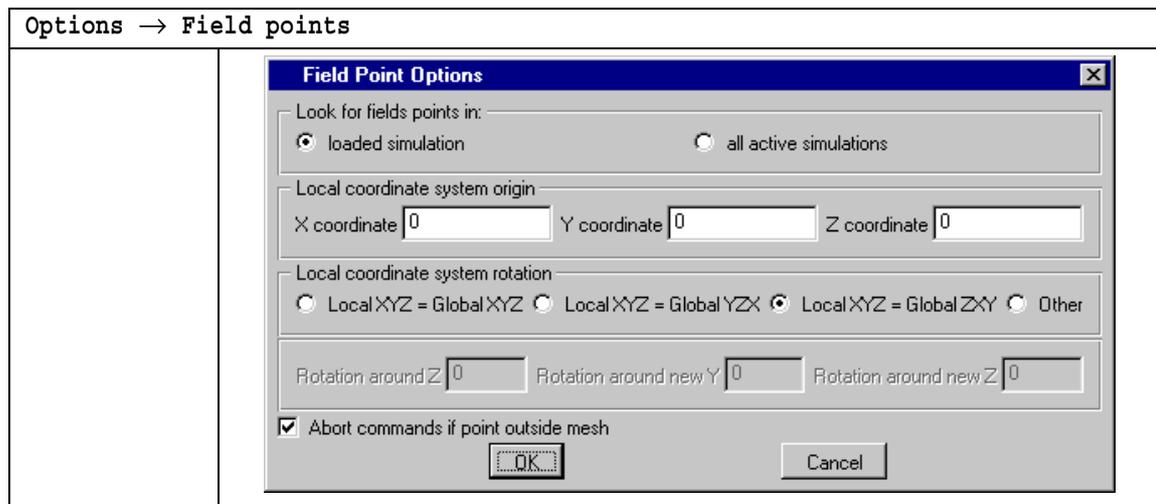
The value returned should be about +1870 Gauss. Note that the position of the point is highlighted. Because of the symmetry of the magnet, the Y-component of flux density values at  $(\pm 10, \pm 10, \pm 10)$  should also be positive and the same magnitude. Check this by changing the coordinate values. If any of the values are negative, the reflections were not entered correctly when the complete magnet model was loaded.

## Field Homogeneity

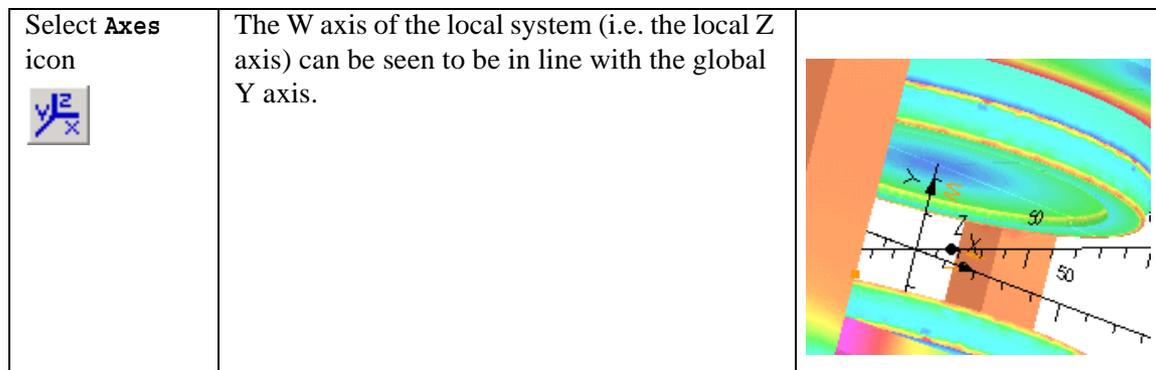
In MRI systems, it is important to have a volume of very homogeneous field. An effective way of representing the homogeneity is as the coefficients of associated Legendre polynomials.

### Local Coordinate System

The post processor assumes that it will compute the coefficients using a spherical coordinate system, such that the field is acting in the direction of its Z-axis. In this magnet, the field is acting in the Y-direction. Consequently, the first task is to define a local coordinate system such that its Z-axis is along the global Y-axis.



The orientation of the local coordinate system can be seen by making the coordinate axes visible again.

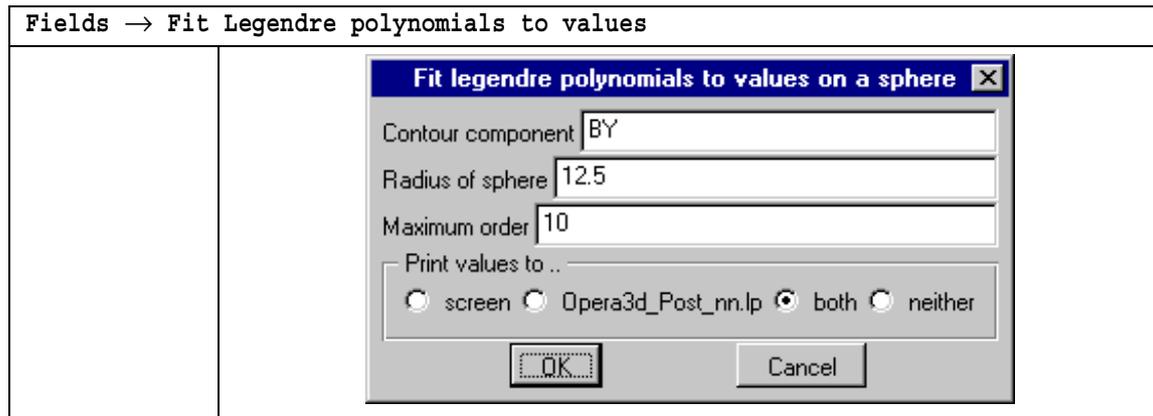


## Legendre Polynomial Coefficients

The field component is still set at **BY** from the previous **Fields at a point** evaluation, which is the correct component. The fitting to obtain the Legendre polynomial coefficients is performed on a sphere with a defined radius. The program samples the field on the surface of the sphere and fits the results to the polynomial expansion. These are reported in a separate window. The form of the polynomial of order  $N$  is

$$B_y = \sum_{n=0}^N \sum_{m=0}^n P_n^m(\cos\Theta) [\alpha_n^m \cos(m\phi) + \beta_n^m \sin(m\phi)] \quad (6.1)$$

The table of coefficients shown are the  $\alpha$  and  $\beta$  values in this equation. The order columns are  $n$  and  $m$  respectively. Consequently, the value of  $\alpha$  for  $n = m = 0$  is the mean value of the dipole field at this radius. If the sphere contains no sources (magnetic materials or currents), the associated Legendre polynomial is the solution to Laplace's equation and this value will be the field at the centre of the sphere.



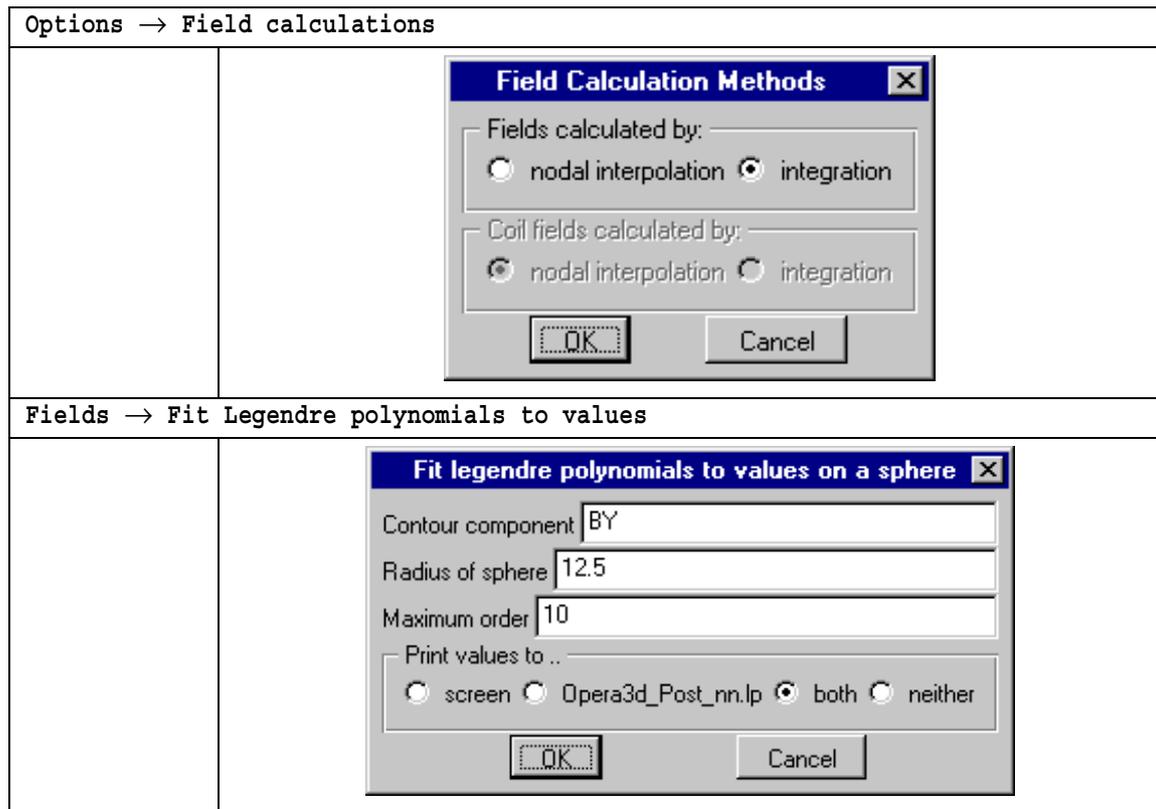
Providing the field satisfies Laplace's equation, homogeneity on any other radii spheres can also be evaluated by inclusion of an additional  $r^n$  multiplier in the equation, where  $r$  is the radius normalised from the actual evaluation radius, i.e.

$$B_y = \sum_{n=0}^N r^n \sum_{m=0}^n P_n^m(\cos\Theta) [\alpha_n^m \cos(m\phi) + \beta_n^m \sin(m\phi)] \quad (6.2)$$

As may be expected from the symmetry of the magnet, the largest harmonic contaminants are at  $n = 2, m = 0$  (-0.98 Gauss) and  $n = 4, m = 0$  (0.26 Gauss). These probably result from the return flux through the legs of the magnet.

***Integral Fields***

In the above calculation, the values of field sampled on the spherical surface were computed using the element basis functions. The program determines which element the sampling point lies in and interpolates from the nodally averaged values. An alternative method of computing the field is to integrate the magnetisation of the magnet. This has the advantage that the value computed at the sample point is not dependent on the local mesh size. This is a fairly lengthy process, but for many problems it gives improved accuracy. Note that the process can be aborted using the Cancel button .



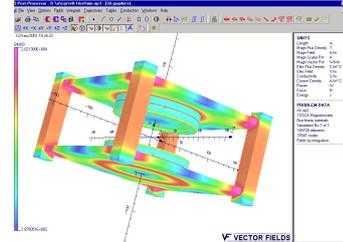
## Force Between Poles

Generally, force calculations on magnetic objects are computed by integration of the full Maxwell stress tensor over a surface surrounding the object. For a sym-

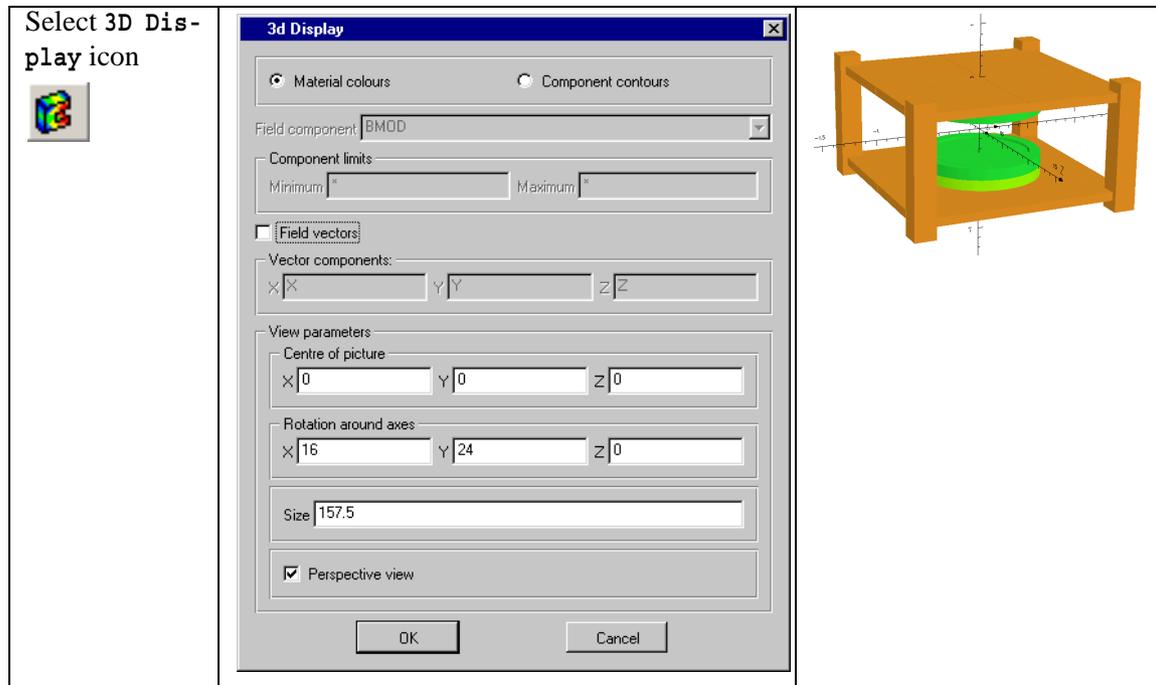
metric magnet such as this one, it is sufficient to compute the integral of  $\frac{(B_y)^2}{2\mu_0}$  on the mid-plane between the poles. This formula is valid for SI units.

### *Changing the Unit System*

Even though the model was created in CGS units, it is possible to change the unit system in the post processor.

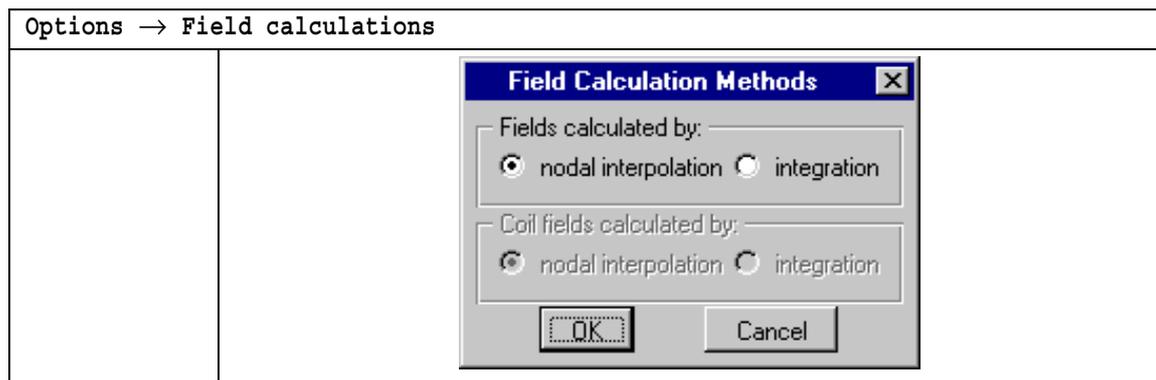
<p>Select <b>Units</b> icon</p> 	<p>Click on <b>SI Units</b> followed by <b>OK</b>. Note that the units in the graphics window are updated.</p>	
---	--	--

It is necessary to re-display the geometry in the new units. The material colour display is also restored.



## Field Values on a Plane

Before defining the surface over which the integration for the force will be performed, the field calculation method is re-set to nodally averaged fields.



It is important that the integration captures all the flux that is passing between the poles across the mid-plane. A circular plane that extends nearly to the legs of the MRI achieves this. The user defines the circular plane in local  $(r, \theta, z)$  coordinates.

The local coordinate system is still set with its local Z-axis along the global Y-axis, which is correct for the orientation of this model. Define the **POLAR** patch as follows:

Select Fields on a polar patch icon



Field on a polar patch

First corner ...  
R  Theta  Z

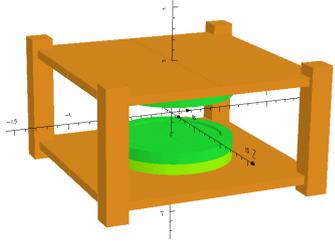
Second corner ...  
R  Theta  Z

Third corner ...  
R  Theta  Z

Fourth corner ...  
R  Theta  Z

Number of points ...  
on sides 1 and 3  on sides 2 and 4

Click on **Evaluate fields**



The force density,  $(B_y)^2/(2\mu_0)$ , can now be plotted as a contour (zone) map. This expression is written as `by**2/2/mu0`. The integral of the force density will also be returned.

Select Contour or vector map icon



Contour or vector map

Zone map
  Line map
  Histogram
  None
  Vectors
  Replace existing maps

Data:  
 Internal buffer  External table file

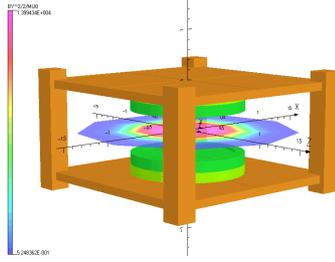
Table file:

Contour component:

Component limits:  
Minimum:  Maximum:  Number of lines:  Height of histogram:

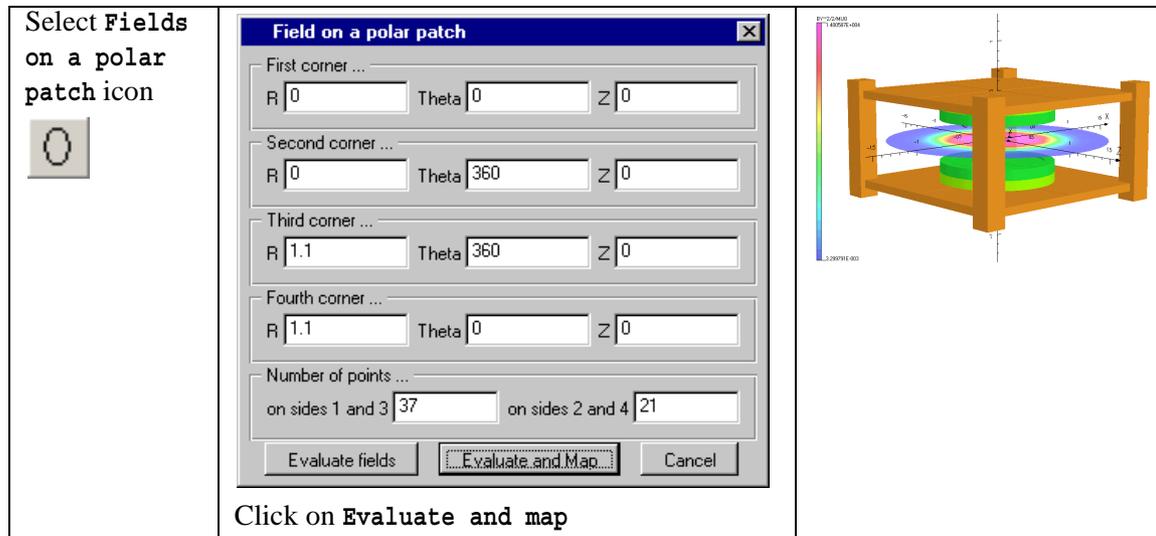
Vector components:  
x:  y:  z:

Print values to Opera3d\_Post\_n.lp



At the bottom of the graphics screen, a message shows the maximum and minimum values of the component on the map (force density) and its integral (about 14290N).

The number of evaluation points on the patch is very small (10 x 10). A more accurate result will be obtained by increasing this.



The force integral increases to about 15310 N. To be confident of the result, the user should continue to increase the number of evaluation points on the patch until the answers converge to an acceptable accuracy.

**File → Exit**

ends the post processor session.

# Chapter 7

## Implementation Notes

### Windows

---

In Windows Explorer, the file types *opc*, *opcb* and *op3* have been associated with the OPERA-3d Modeller and post processor. This means that if you double-click on a filename, the appropriate program will start and load the file:

- *opc* and *opcb* files are associated with the Modeller. The program starts and loads the data from the file.
- *op3* files are associated with the post processor. The following commands are run:

```
ACTIVATE FILE=filename  
LOAD  
SELECT ACTION=DEFAULT  
SELECT ACTION=SELECT OPTION=SURFACES  
THREED OPTION=REFRESH
```

Note that the Project Folder will not be changed and any initial command file in the Project Folder will be run before loading any data (Modeller) or running the commands (post processor).

