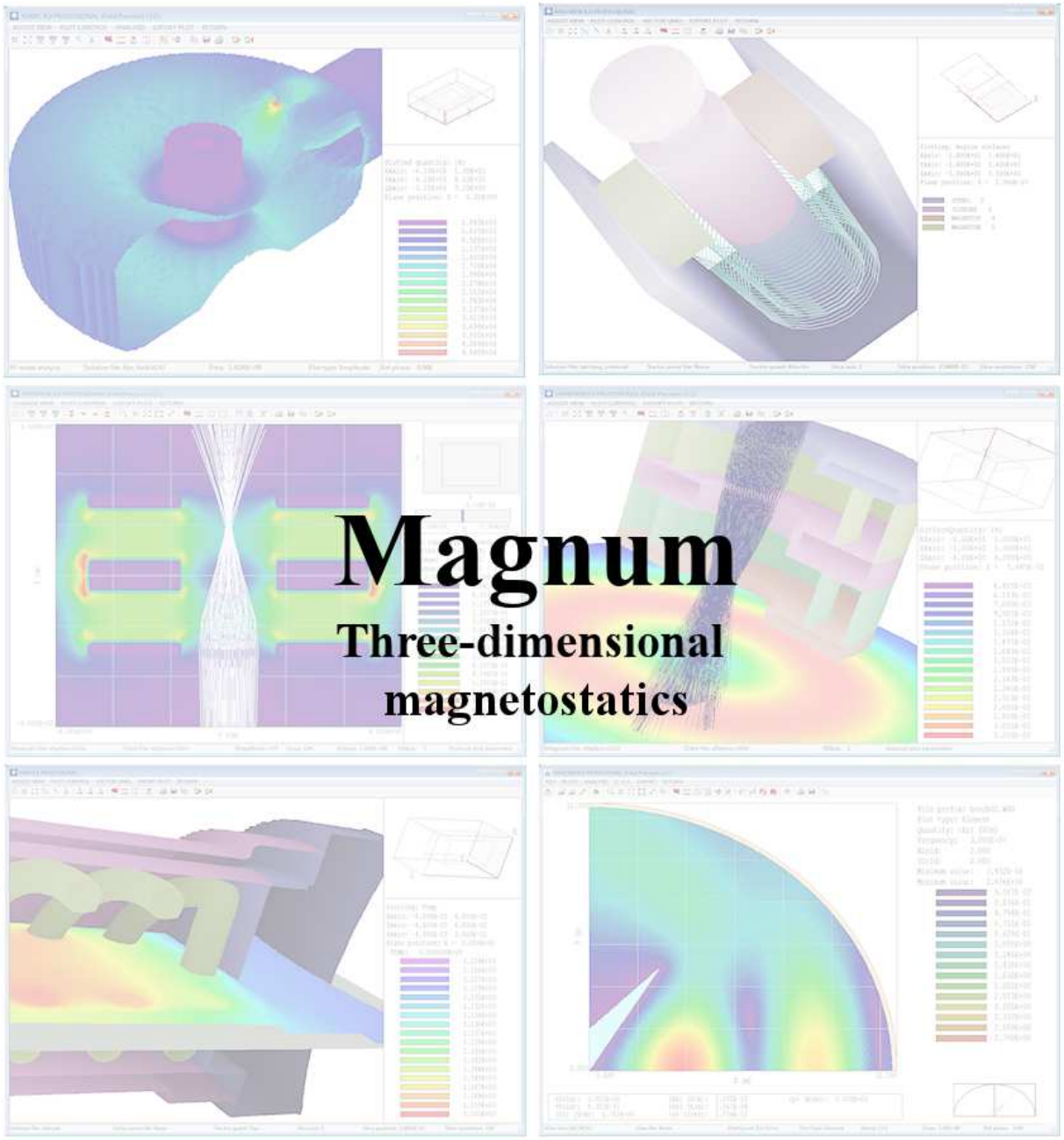
Field Precision LLC



Magnum

Three-dimensional magnetostatics

EMail: techinfo@fieldp.com

URL: http://www.fieldp.com

Contents

1 Introduction

1.1 Program function

Magnum applies finite-element methods to calculate magnetostatic fields in three-dimensional systems. The program handles effects of linear ferromagnetic materials, permanent magnets, applied currents and ideal conducting boundaries for pulsed fields. The **MagWinder** utility is used to define drive coils of any geometry. The program can operate in two modes:

- Bounded finite-element solutions, including the effects of ferromagnetic or conducting materials.
- Unbounded free-space fields resulting from a specified distribution of currents.

Three programs are included in the package:

- `MAGWINDER.EXE`. An interactive, graphical utility to define applied currents and to create standard files of current elements.
- `MAGNUM.EXE`. The main solution program to determine one or more solutions in a window or under batch file control.
- `MAGVIEW.EXE`. A postprocessor to generate plots and to analyze solutions from **Magnum**.

MetaMesh (the **AMaze** conformal mesh generator) is included to create geometry files for solutions. **Magnum** features fast and accurate calculations in random-access memory. The programs use dynamic memory allocation; therefore, the size of the solution is limited only by the installed RAM. A computer with 2 GB of memory can handle over 10 million elements.

As a quick introduction to the programs, the following section in this chapter describes a step-by-step calculation. Chapter 2 covers the programs used for a complete magnetostatic solution and the organization of input and output files. Defining applied currents is the first step in most **Magnum** simulations. Chapter 3 describes the **MagWinder** preprocessor. In this interactive program, you specify macroscopic coils model to generate large sets of three-dimensional current elements. Chapter 4 covers the structure of the **Magnum** control script. The script sets parameters for program operation and defines the material properties of regions. **Magnum** features an interactive dialog to generate basic scripts. Advanced control commands may be added with an editor. Chapter 6 describes how to run `magnum.exe` to generate a solution, either interactively in a window or autonomously under batch file control. Chapters 7 through 10 introduce the **MagView** postprocessor. This program creates a variety of two and three-dimensional plots and performs quantitative analyses. Chapter 11 is a description of the physical and mathematic foundation of MagNum. Although a detailed knowledge of theory is not necessary for basic work, the information can be helpful in the creation of effective solutions as you gain more experience. Chapter 15 summarizes the formats of the current element input file and the **Magnum** output file. This information is useful if you want to write your own analysis programs. Techniques for three-dimensional force calculations are covered in Chap. 13.

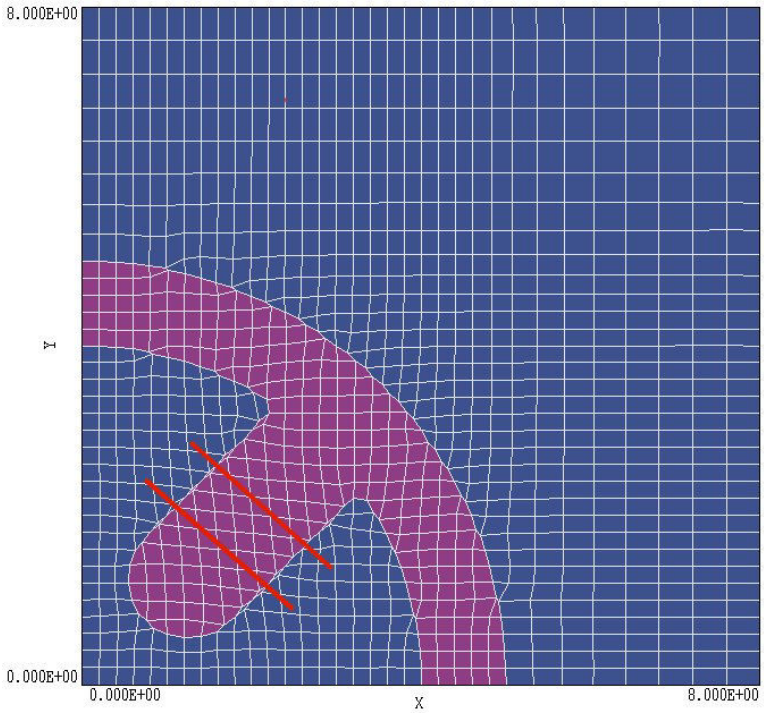


Figure 1: Mesh plot of the cross-section of the quadrupole magnet in a plane normal to z at $z = 4.0$ cm. A projection of the drive coils is shown in red.

1.2 Walkthrough example

This example gives an overview of the solution procedure. The calculation describes a finite-length quadrupole magnet. A coarse mesh and symmetry boundaries are used for a fast solution – the process takes less than 1 minute. Figure 1 shows a cross-section of the magnet in the x-y plane. The orientation was chosen so that the planes $x = 0.0$ and $y = 0.0$ are symmetry planes that satisfy the condition $B_{\parallel} = 0.0$. Therefore, it is sufficient to model only one quadrant of the magnet. Similarly, the symmetry condition $B_{\perp} = 0.0$ holds at the axial midplane ($z = 0.0$), so it is sufficient to model only the portion of the magnet in the region $z \geq 0.0$. The iron pole has an axial length of 12.00 cm. It extends from $z = 0.0$ cm to $z = 6.0$ cm in the half-plane simulation. The outer radius is 5.0 cm. A peripheral region of coarse mesh extending to $x = 8.0$ cm, $y = 8.0$ cm and $z = 16.0$ cm has been included for a good representation of fringing fields. Figure 2 shows a three-dimensional view of the pole piece and one set of drive coils.

To prepare for the run, move the example input files `SHORTQUAD.MIN`, `SHORTQUAD.CDF`, and `SHORTQUAD.SCR` to a working directory. Run `fpcontroller.exe` (the **AMaze** program launcher) and click on *Set data directory*. Move to the working directory and click *OK*. Then launch **MetaMesh**. The first step is to create a volume mesh. Table 1 shows the **MetaMesh** script `SHORTQUAD.MIN`. The mesh serves two purposes. First, the region assignments and shapes of the conformal elements define the material division of the solution volume (*i.e.*, the boundaries of the pole piece in Fig. 2). Second, the mesh serves as an armature for recording the applied fields. The calculated field values at nodes are used to determine source terms for the magnetic field solutions. Load and process the file `SHORTQUAD.MIN`, and save the mesh to create the file `SHORTQUAD.MDF`.

Table 1: MetaMesh script file SHORTQUAD.MIN

```
GLOBAL
  XMesh
     0.00  5.20  0.20
     5.20  8.00  0.40
  End
  YMesh
     0.00  5.20  0.20
     5.20  8.00  0.40
  End
  ZMesh
     0.00  6.20  0.20
     6.20 16.00  0.40
  End
  RegName 1 Air volume
  RegName 2 Quad pole
  RegName 3 Symmetry boundary
END
PART
  Type Box
  Region 1
  Fab 16.00  16.00  32.00
END
PART
  Type Extrusion
     L    4.00E+00  0.00E+00  5.00E+00  0.00E+00
     A    5.00E+00  0.00E+00  6.08398E-08  5.00E+00  0.00E+00  0.00E+00 S
     L    0.00E+00  5.00E+00  0.00000E+00  4.00E+00
     A    4.8672E-08  4.0E+00  2.2839E+00  3.2839E+00  0.0E+00  0.0E+00 S
     L    2.2839E+00  3.2839E+00  7.50000E-01  1.75000E+00 S
     A    7.50E-01  1.75E+00  7.50E-01  7.50E-01  1.25E+00  1.25E+00 S
     A    7.50E-01  7.50E-01  1.75E+00  7.50E-01  1.25E+00  1.25E+00 S
     L    1.75E+00  7.50E-01  3.28388E+00  2.28388E+00 S
     A    3.28388E+00  2.28388E+00  4.0E+00  0.0E+00  0.0E+00  0.0E+00 S
  End
  Region 2
  Fab  12.00
  Surface Region 1
END
PART
  Type BoundXDn
  Region 3
END
PART
  Type BoundYDn
  Region 3
END
ENDFILE
```

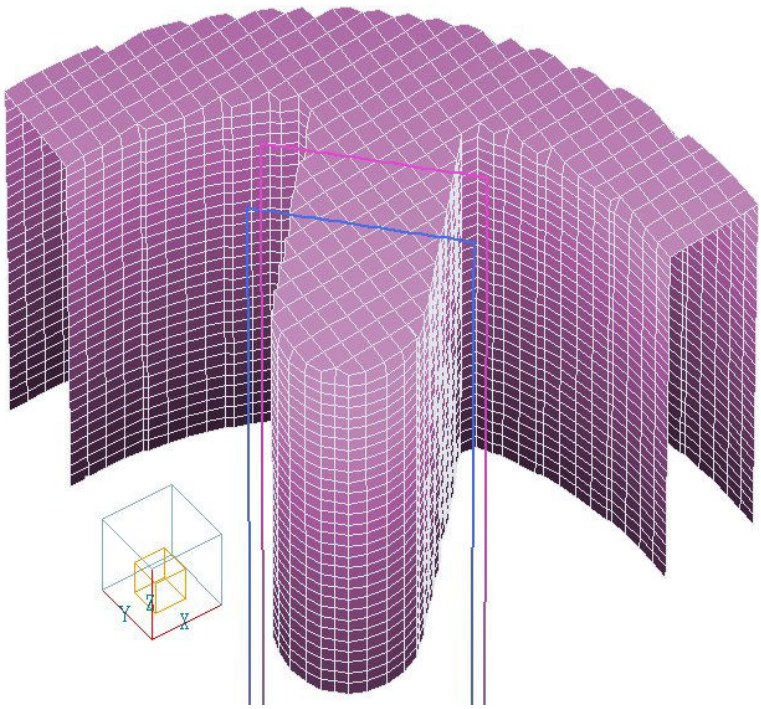


Figure 2: Three-dimensional view of the pole piece in the solution volume and a portion of the drive coils.

The second required input component is the spatial distribution of applied currents to drive the magnetic fields. The values are contained in a file with a name of the form `DATANAME.WND`. The file contains a set of current element specifications. Each data line gives the start point, end point and current of an element. The currents determine the applied magnetic intensity $\mathbf{H}_s$ at nodes of the solution volume. In most cases, you need not create this file directly. The **Magnum** package contains the utility program **MagWinder** which reads coil specifications and automatically divides the coils into small current elements.

Run `MAGWINDER.EXE` and choose *Load coil file* in the *File* menu. Pick the file `SHORTQUAD.CDF`. **MagWinder** immediately processes the file and shows a three-dimensional plot similar to Fig. 3. Table 2 shows an extract from the script. The complete file defines eight box coils that surround the four pole extensions of the quadrupole magnet. You could define a larger number of coils to give a better approximation of the winding density – we used a small number in the example to ensure a quick solution. Note that it is necessary to define the full complement of four coil sets over the complete axial range even though the solution in **Magnum** is carried out only in the first quadrant. This is a consequence of the solution technique applied in the program – **Magnum** first calculates applied fields from a Biot-Savart integral and then uses finite-element techniques to compute the field contributions from materials. Although symmetry boundaries influence material contributions to the fields, we must ensure the symmetry of the applied fields directly by including all coils. Chapter 11 gives a more detailed description of the solution process. Click on the *Save element file* command to create the file `SHORTQUAD.WND`. You can exit or minimize **MagWinder**.

We are now ready to use **Magnum**. Run the program from **AMaze**, choose the *Setup* command and pick the file `SHORTQUAD.MDF`. The program displays the dialog of Fig. 4. Note

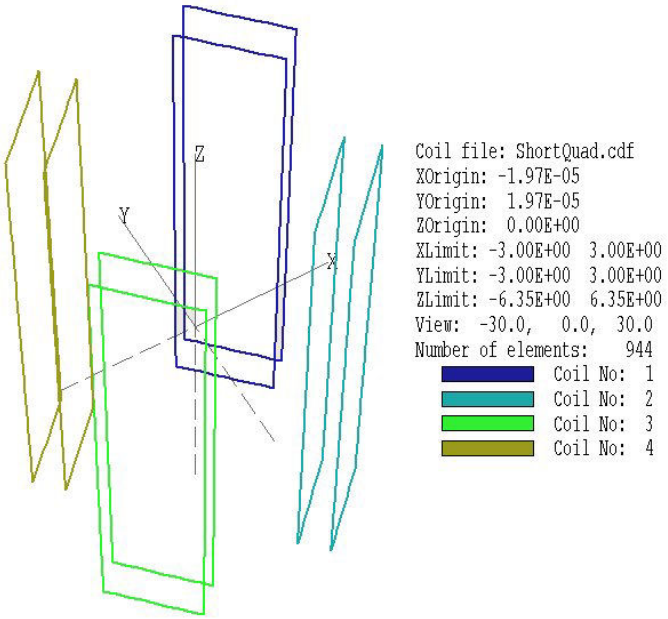


Figure 3: Current-element set created by **Magwinder** for the `SHORTQUAD` example.

Table 2: Section of the **MagWinder** script `SHORTQUAD.CDF`

```
GLOBAL
  DUnit = 100.0
  Ds = 0.250
END
COIL
  Name = Quadrant01
  Current = 1000.0
  Part
    Rotate = 90.0 0.0 -45.0  XYZ
    Shift = 1.750 1.750 0.000
    Type = Rectangle
    Fab = (-1.061, -6.354) (1.061, 6.354)
  End
  Part
    Rotate = 90.0 0.0 -45.0  XYZ
    Shift  = 2.250 2.250 0.000
    Type = Rectangle
    Fab = (-1.061, -6.354) (1.061, 6.354)
  End
END
  ...
ENDFILE
```

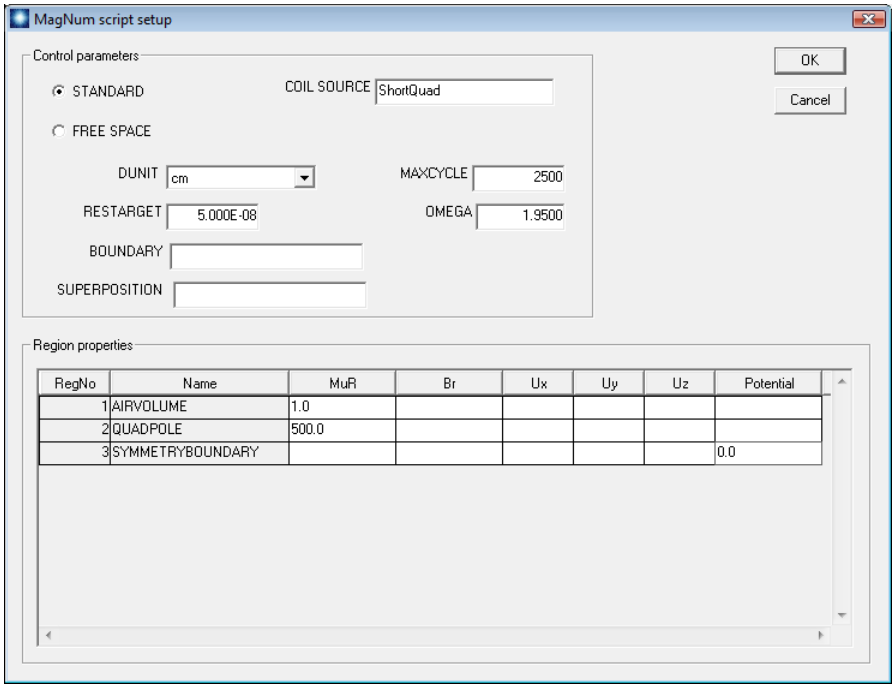


Figure 4: Setup dialog with values for the `SHORTQUAD` example.

that the grid contains a row for each mesh region. There are six columns in the grid where you can enter values: *MuR* (relative dielectric constant), *Br* (the remanence magnetic flux density in tesla), *Ux*, *Uy* and *Uz* (magnetization direction vector for a permanent magnet) and *Potential* (fixed value of dual and reduced potentials).

The entries in the *Control parameter* section of the dialog have the following functions:

- STANDARD, FREE SPACE. Radio button to set the solution type. The *standard* type is a bounded finite-element solution with magnetic materials and drive coils and/or permanent magnets. The *free-space* type is a Biot-Savart integral with specified coils.
- COIL SOURCE. The prefix of a file of current elements (`FPREFIX.WND`) for solutions with drive coils created by **MagWinder**.
- DUNIT. Set a factor to convert coordinate units used in **MetaMesh** to meters. The value is the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
- BOUNDARY and SUPERPOSITION. Advanced program capabilities described in Sect. 4.4.
- RESTARGET. Accuracy tolerance for the iterative matrix solution of the finite-element equations. The field and the following ones apply only to *Standard* solution.
- MAXCYCLE. Maximum number of cycles in the iterative solution.
- OMEGA. A parameter in the range 0.0 to 2.0 to control the iterative matrix solutions.

Table 3: **Magnum** script `SHORTQUAD.GIN`

```
SolType = STANDARD
Mesh = ShortQuad
Source = ShortQuad
DUnit =   1.0000E+02
ResTarget =   1.0000E-07
Omega =   1.9500E+00
MaxCycle =    2000
Parallel

* Region 1: AIRVOLUME
Mu(1) =   1.0000E+00

* Region 2: QUADPOLE
Mu(2) =   5.0000E+02

* Region 3: SYMMETRYBOUNDARY
Potential(3) =   0.0000E+00

EndFile
```

The column options in the region grid box determine the material properties of the regions. An entry in the *MuR* column implies a linear, ferromagnetic material. Entries in the *Br*, *Ux*, *Uy* and *Uz* define a permanent magnet with a straight-line demagnetization curve. Finally, an entry in the *Potential* column specifies that the region has a fixed value of the reduced and dual potentials. The most common entry is 0.0 to define a symmetry boundary with $B_{\|} = 0.0$. The values shown in Fig. 4 define the following region characteristics:

- Region 1. Air, $\mu_r = 1.0$.
- Region 2. Iron pole, $\mu_r = 500.0$.
- Region 3. Fixed-potential symmetry boundaries at $x = 0.0$ and $y = 0.0$.

Set up the dialog as shown and click *OK*. **Magnum** uses the information in the dialog to create the script `SHORTQUAD.GIN` shown in Table 3. Chapter 4 reviews the script format and advanced program capabilities.

To carry out the solution, click on the *Run/Start run* menu command or tool. In the dialog, pick the file `SHORTQUAD.GIN`. **Magnum** reads the mesh and current element information, calculates applied fields, determines element matrices and coupling coefficients for the finite-element solution and then proceeds with two solution stages. The entire process takes less than a minute. The program creates the binary output file `SHORTQUAD.GOU` and the diagnostic listing `SHORTQUAD.GLS`.

When the solution is complete, start the program **MagView**. Pick *Load solution file* from the *File* menu or click on the tool. Choose the file `SHORTQUAD.GOU` in the dialog. **MagView** can generate a wide variety of 2D and 3D plots. Figure 5 shows an example, a 3D representation

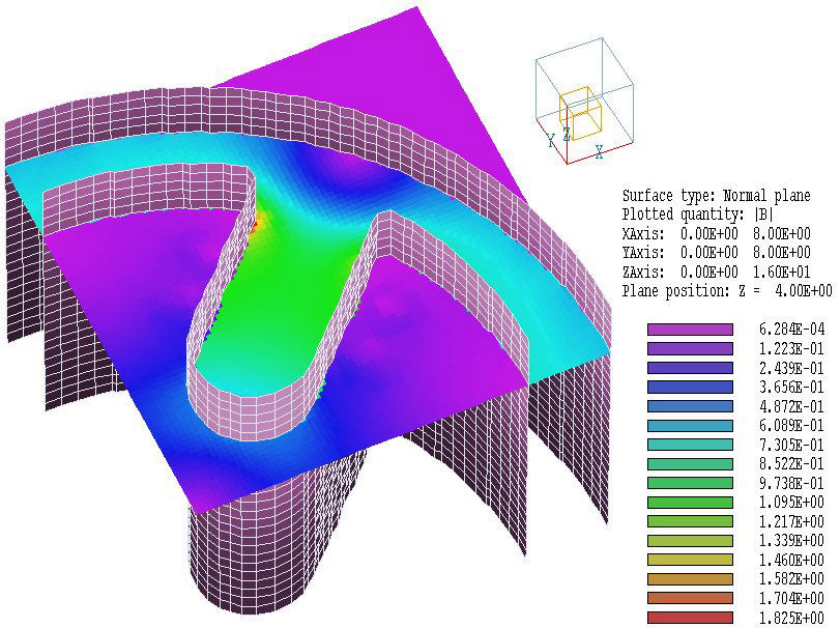


Figure 5: Three-dimensional view of physical surfaces with a plot of $|\mathbf{B}|$ in a plane normal to z.

of the pole piece with $|\mathbf{B}|$ plotted in a plane normal to the z axis. Chapters 7, 9 and 10 give detailed information on plotting capabilities. Here, we shall create a simple example to test the program. Click on *Plane plots* to bring up the plane plot menu.

The default plot shows the variation of $|\mathbf{B}|$ in a plane normal to z at the midpoint. Click on the *Set plane* command. In the dialog move the slider to $z = 4.0$ cm, within the region of the pole. Change plot limits to $x_{max} = 4.0$ cm and $y_{max} = 4.0$ cm. Finally, click on *Plot style* in the *Plot control* menu. You should see a plot similar to Fig. 6. If you try some of the other plot styles, note that plane plots may give ragged edges on sloped or curved surfaces. This is because the plot is based on a simple division of the plane into a rectangular array with no special provisions for material boundaries. Slice plots give more accurate representations, although the available styles are more limited.

Although plots are interesting, the primary function of **Magnum** is to generate numbers. Return to the main menu and click on *Run script* in the *File* menu. In the dialog, pick the file `SHORTQUAD.SCR`. There is a delay while **Magnum** performs an analysis. To begin, we shall inspect the instructions in the data script. In the *File* menu click on *Edit script* and choose `SHORTQUAD.SCR`. The internal program editor loads and shows the following content:

```
INPUT ShortQuad.GOU
OUTPUT ShortQuad.DAT
NSCAN 50
RECORD FIELD
SCAN 0.0 0.0 1.0  1.0  1.0  1.0
SCAN 0.0 0.0 5.0  1.0  1.0  5.0
SCAN 0.0 0.0 6.5  1.0  1.0  6.5

ENDFILE
```

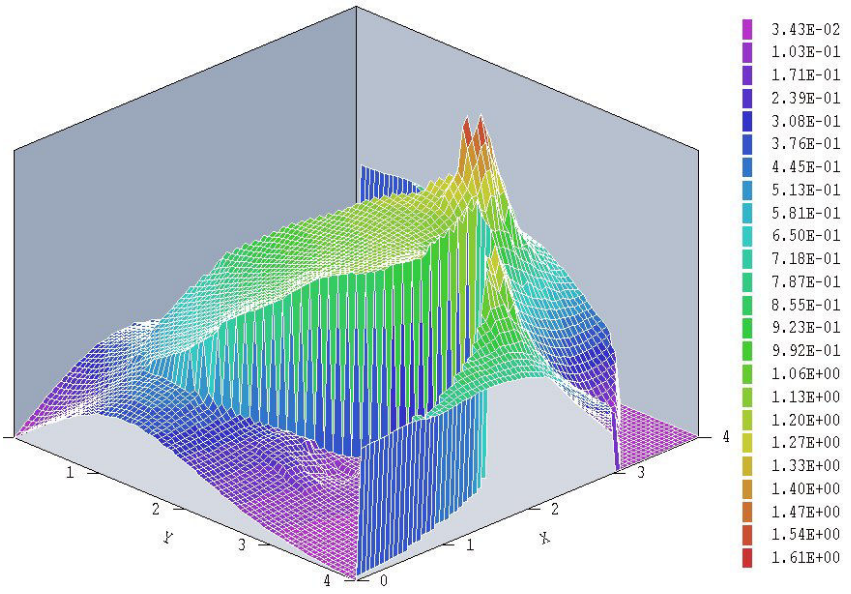


Figure 6: Plot of $|\mathbf{B}|$ in a plane normal to z at $z = 4.0$ cm.

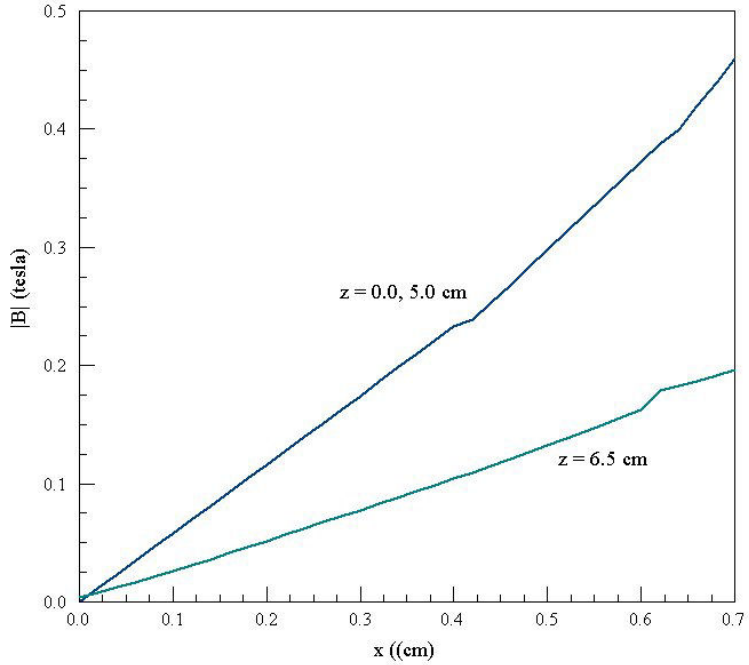


Figure 7: Variation of $|\mathbf{B}|$ as a function of x along a 45^o line at three axial positions

The first statement ensures that the proper solution file is loaded, and the second statement opens a data file to write the results. The command *NScan 50* instructs the program to compute 51 values (50 intervals) along a line scan. The next three lines compute scans of values from the axis to the point($x = 0.7$, $y = 0.7$). Results are plotted in Fig. 7. The lines at $z = 1.0$ and 5.0 cm are almost identical. The field variation near the axis is linear and grows faster than linear near the pole face. The non-linearity results from the fact that the pole does not have an ideal hyperbolic shape. Note that there are small discontinuities in the plot. We have shown these as a reminder that numerical results are never exact. The interpolation routines have done the best job possible with the coarse mesh of Fig. 1. Improved results could be achieved with a finer mesh near the axis.

2 Organizing Magnum calculations

2.1 Procedures and files

The **Magnum** package contains three components:

- **MagWinder** creates lists of input specifications for applied-field coils and generates files of current elements in a standard format.
- The **Magnum** program computes the physical solution
- **MagView** is a dedicated post-processor for analyses of the results.

All three programs can run in two modes: interactively in a window or autonomously in the background under batch file control. The autonomous mode allows automatic processing of large or repetitive data sets.

The **MagWinder** pre-processor creates and analyzes coil definition files with names of form `FPREFIX.CDF`. Chapter 3 describes operation of the program. Three input files may be required for a **Magnum** solution:

- A script that sets control parameters and describes the physical properties associated with regions.
- A list of drive current elements to compute the applied field.
- A **MetaMesh** output file that describes the conformal hexahedron mesh. The file contains node coordinates and the region numbers of elements and nodes.

The **MagWinder** output file has a name of the form `FPREFIX.WND`. The **MetaMesh** output file always has a name of the form `RUNNAME.MDF`. The **Magnum** script must have a name of the form `RUNNAME.GIN`. The file may be prepared with the *Setup* command in **Magnum** or directly with an editor. **Magnum** issues an error message if any input files are not available in the current working directory. To organize data, the resulting output file is assigned the name `RUNNAME.GOU`.

To review, a simulation usually consists of the following steps:

- Prepare a **MetaMesh** script (`MPREFIX.MIN`) using **Geometer** or a text editor to define the solution space. In simulations with conductive or ferromagnetic materials, the conformal mesh describes the division of the solution space into air and material regions. In free-space calculations, the mesh is a convenient set of node positions for calculations of applied fields.
- Run **MetaMesh** to create a file `MPREFIX.MDF` of standard mesh information. This file could be used as input for multiple **Magnum** simulations or for other **AMaze** solution programs.

Table 4: **Magnum** files

Name form	Function	Status
MPREFIX.MIN	Description of simulation geometry, input to **MetaMesh**	Required
CPREFIX.CDF	Description of applied field coils, input to **MagWinder**	Required
CPREFIX.WND	Output from **MagWinder**, input to **Magnum**	Required
MPREFIX.MDF	Output from **MetaMesh**, input to **Magnum**	Required
RUNNAME.GIN	Description of material properties, input to **Magnum**	Required
RUNNAME.GOU	Output from **Magnum**, input to **MagView**	Required
SPREFIX.SCR	Analysis control, input to **MagView**	Optional
SPREFIX.DAT	Analysis data output from **MagView**	Optional

- Prepare a **MagWinder** coil definition file (WPREFIX.CDF) to specify the geometry and currents of drive coils. This file is not necessary if fields are created solely by permanent magnets.
- Run **MagWinder** to cut the coils into small current elements and generate a standard element file (FPREFIX.WND).
- Use the *Setup* command or an editor to prepare a **Magnum** script (RUNNAME.GIN) that sets control parameters and defines the material properties of regions.
- Run **Magnum** to create a solution file RUNNAME.GOU. This file contains the following node quantities: spatial coordinates, applied field components $\mathbf{H}_s$, reduced potential ϕ and dual potential ψ (in calculations with ferromagnetic materials).
- Optionally, prepare an **AMaze** analysis script (SPREFIX.SCR) to control a **MagView** analysis session.
- Run **MagView** to create plots or to generate numerical data using the information in RUNNAME.GOU.

Although a calculation involves several steps, the approach saves time in the long run. Splitting complex solutions into small steps is always a good practice. An input scripts is a permanent record of the setup with complete information necessary to regenerate the solution. Furthermore, the scripts may often be used in other solutions with small modifications. Table 4 summarizes the input and output files used in **Magnum**.

2.2 Script conventions

The **Magnum** input script FPREFIX.GIN is a text file with data lines containing commands and parameters. The script must end with the *EndFile* command. The programs make no

distinction between upper and lower case. Entries on a line may be separated by the following delimiters:

```
Space, blank
Comma [,]
Tab
Colon [:]
Equal sign [=]
Left parenthesis [(]
Right parenthesis [)]
```

You may use any number of delimiters in a line. This means that you can add indentations and customize the appearance of the script. For example, the two lines

```
Epsi 2 5.56
Epsi(2) = 5.56
```

have the same effect.

Magnum ignores blank lines and comment lines. Comment lines begin with the symbol [*] (asterisk). Most parameters are real numbers. The following formats are valid.

```
1.000
5.67E6
6.8845E+09
5
```

The final number is interpreted as 5.0.

Magnum accepts commands in any order. The program reads and analyzes all commands before starting the solution. Generally, it is good practice to put control commands at the beginning and to group commands that set material properties by region. The following example illustrates a complete script:

```
* MagNum Script (Field Precision)
* File: relay.GIN
SolType = STANDARD
Mesh = relay
Source = relay
DUnit =   1.0000E+02
ResTarget =   1.0000E-06
Omega =   1.9850E+00
MaxCycle =    1000
* Region 1: AIR
Mu(1) =   1.0000E+00
* Region 2: SOLECORE
Mu(2) =   5.0000E+02
* Region 3: PLATE
Mu(3) =   5.0000E+02
EndFile
```

You can may place text in any format after the *EndFile* command. Annotations may be helpful when you return to a simulation after a period of time.

3 MagWinder – defining applied current

3.1 Program function

Input files of *current elements* are used in **Magnum** to define drive currents. Magnet coils are represented numerically by dividing them into a large set of short segments. Sometimes drive coils may be simple, but often they follow complex paths in three-dimensional space. **MagWinder** provides an interactive environment where you can build magnet windings step-by-step. Several features of **MagWinder** help in the task of current-element generation:

- A comprehensive set of parametric models for common coil configurations (solenoids, helices,...).
- Versatile graphical displays to show the state of the assembly.
- Interactive dialogs to modify the geometry, position and orientation of components.

MagWinder may be used as a command-line utility with an existing `CDF` file with command of the form:

```
[PATH\]MAGWINDER FPREFIX <Enter>
```

An interactive **MagWinder** session involves three steps: 1) specify a set of parametric models to define one or more drive coils, 2) divide the coils into elements and 3) record the results in a file for input to **Magnum**. The program creates two types of text data files:

- A list of model parameters containing all necessary information to regenerate elements. The file has a name of the form `FPREFIX.CDF` (**C**oil **D**efinition **F**ile). You can reload the script into **MagWinder** to make changes in geometry or element lengths.
- The **Magnum** element file `FPREFIX.WND`, which represents a specific embodiment of the parametric models. Section 3.13 reviews the file organization.

It is important to save the `CDF` file if you want to repeat a calculation or to change an assembly. The script can regenerate an element file, but the `CDF` file cannot be recovered from the contents of the element file.

The next sections in this chapter describe how to create `CDF` scripts. The easiest approach is to employ the interactive features of **MagWinder** (Sections 3.2 through 3.9). You can also build a script and make changes directly with a text editor. Sections 3.10 through 3.12 describe the script syntax. Either way, the process is easier if you understand the definitions of a few terms:

- **Element**. A differential element of current familiar from introductory electromagnetism courses. An element extends from vector position $\mathbf{x}_s$ to $\mathbf{x}_e$. It has length $dl = |\mathbf{x}_e - \mathbf{x}_s|$ and carries a current dI. For accuracy, the length of elements should be shorter than the distance from the coil to the magnetic field calculation point. It is important to note

that an element is not treated as a filament in **Magnum**, but rather as a cylinder with uniform current density distributed over a diameter equal to its length. This convention avoids numerically-infinite values of applied magnetic field which can compromise code accuracy and plot quality.

- **Part**. A basic building block consisting of wires with a specified shape, position and orientation. The shape is defined by the choice of a model. **MagWinder** has fifteen models ranging from simple (line segment between points, circular coil,...) to complex (solenoid with radial thickness, helical coil,...).

- **Coil**. A set of associated parts that carry the same current. For physically-meaningful solutions, the parts should form self-connected circuits.

- **Assembly**. The set of coils that defines all drive currents for the **Magnum** calculation.

For reference, an assembly may contain up to 150 coils. The maximum number of parts to build all coils is 2500. The assembly may contain up to 2,000,000 current elements.

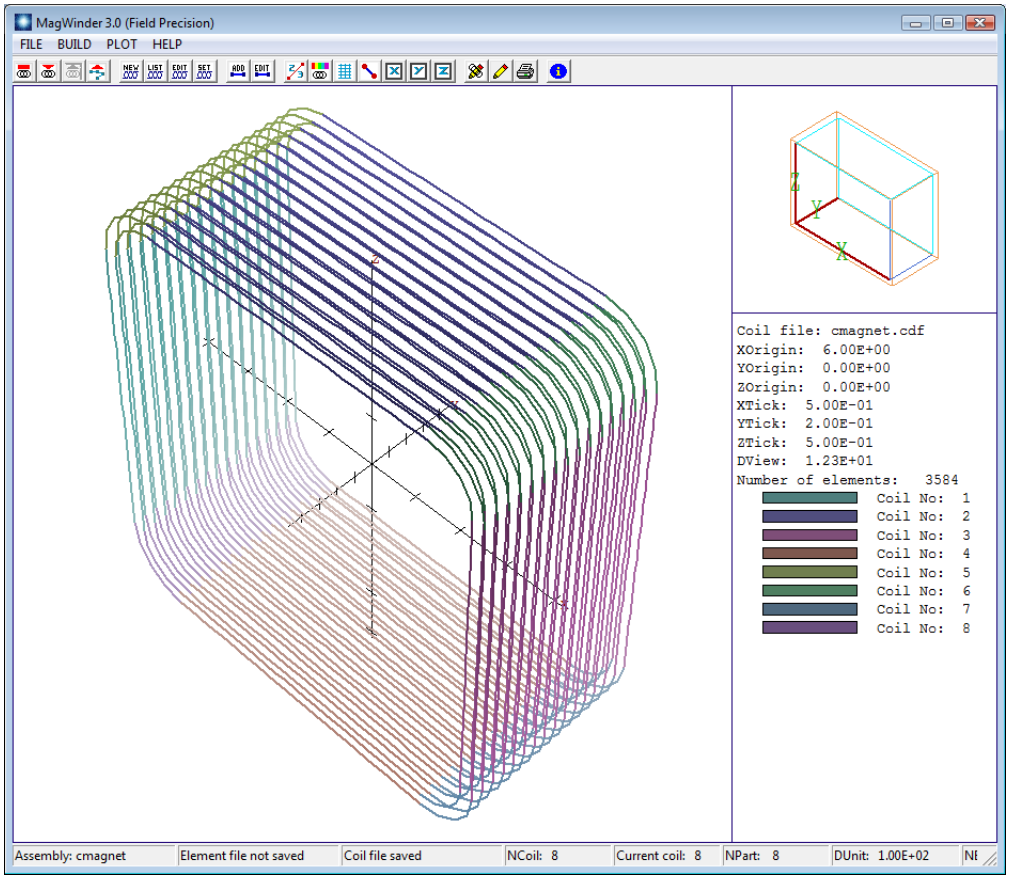


Figure 8: Working environment of **MagWinder**.

3.2 Starting an assembly

To create a `CDF` file, run `MagWinder` and choose the *File/New coil assembly* command. In the dialog, supply a descriptive name that will be used as the default prefix for the `CDF` and element files. The prefix should have length from 1 to 46 characters and may not include standard **AMaze** delimiters (*i.e.*, use underscores rather than spaces). You must choose dimensional units that will be used for positions and length. For custom units, pick *Other*. In this case, a dialog will prompt for a conversion factor $DUnit$. The factor equals the number of units per meter. For example, to work in units of feet, set $DUnit = 3.281$. Finally, you have the option to specify a default global element length $DsGlobal$. The length applies to parts in coils that do not have individual values of Ds.

When you exit the dialog, **MagWinder** updates the status bar to show the assembly name and the unit conversion factor. Note that the plot window is initially blank. Plots are generated from elements, and there are no elements until you define at least one coil with at least one part.

We shall use the example shown in Fig. 8 to illustrate the procedure. Six rectangular loops are arranged to generate a sextupole field over a 50 cm length along z. To follow the example in the following sections, start a new assembly with the name `SEXTUPOLE` using dimensions of cm and setting $DsGlobal = 1.0$.

NEW COIL ASSEMBLY
Initialize all variables and begin a new assembly. **MagWinder** issues a prompt if current work has not been saved.

3.3 Adding a coil

Use the *New coil* command to add a coil to an assembly. The command activates the dialog of Fig. 9. Coils are numbered from 1 to 250 as they are added and assigned default names such as `COIL001`. You can change the name to a more descriptive title up to 80 characters in length. You must supply a value for the current – this value applies to all parts that constitute the coil. Optionally, you can enter a local value Ds for element length that applies only to components of the coil. This feature is useful to represent small components in a large assembly.

Entries in the boxes in the lower half of the dialog are used to set global values of shifts and rotations that apply to all parts of the coil. These operations are performed after individual shifts and rotations of parts within a coil. With this feature, you can construct a complex set of parts and then move it as a rigid body. Section 3.5 gives a detailed discussion of positioning operations for coils and individual parts. You need not enter these values immediately. They can be changed at any point in the session.

For the `SEXTUPOLE` example, we shall use a strategy that minimizes the number of positioning operations and reduces the chances of error. Rather than flip the loops to reverse the field direction, we shall use the same relative orientation for all loops and divide them into two sets (*i.e.*, coils) with positive and negative current values. Loops 1, 3 and 5 have $I = +10,000$ A while loops 2, 4 and 6 have $I = -10,000$ A. Accordingly, we set up the first coil with parameters shown in Fig.9. Click the *New coil* command again, and give the second coil the name `Negative` and assign current $I = -10,000$ A. After you exit the dialog, the status bar shows that the assembly contains two coils with no parts or elements.

NEW COIL
Add a coil to the assembly. The new coil becomes the current coil for editing. You can assign a name, current, local element length and global rotations and shifts that affect all components of the coil. These parameters may be changed using the *Edit coil* command.

3.4 Adding a part

Coil editing operations (such as adding parts) are applied to the *current coil*. You can change the current coil with the *Set current coil* command. The resulting dialog shows a list of defined coils. To choose a coil, press one of the buttons on the right-hand side and then click *OK*.

The *Add part* command brings up the dialog of Fig. 10. The entries in the top group define the model type and parameters. The entries in the bottom group determine the orientation and position of individual parts within a coil. Regarding the top group, all boxes for real-number and integer parameters are initially inactive. The first step is to pick a model type for the part in the drop-down list box at the top. When a model is specified, **Magwinder** activates appropriate parameter fields and adds descriptive labels. Section 3.6 describes the geometries and parameters of available models.

To continue the `SEXTUPOLE` example, click on *Set current coil* and choose the first coil (`Positive`). To define the first rectangular loop, click on *Add part* and choose the `RECTANGLE`

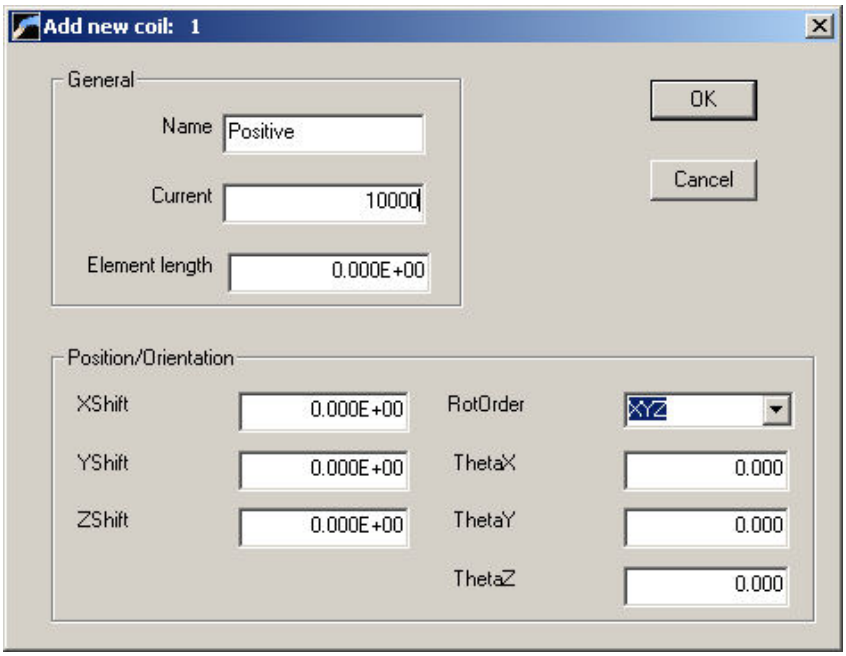


Figure 9: New coil dialog.

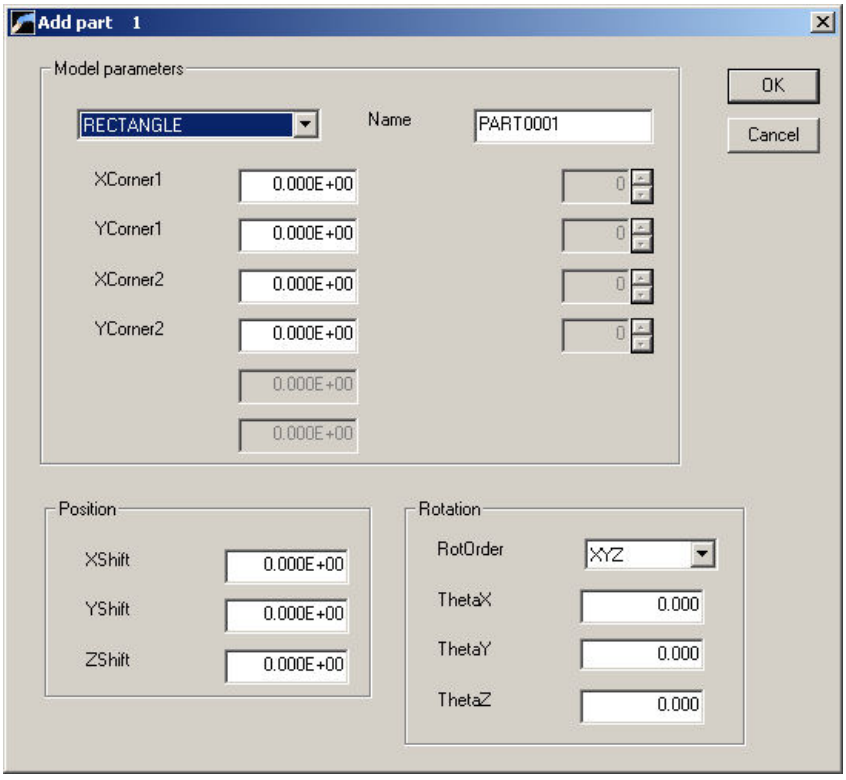


Figure 10: Add part dialog.

model. Enter `LOOP01` in the *Name* field. The loop has narrow dimension 5.0 cm and long dimension 50.0 cm. Enter the following values to define two corners of the rectangle: $x_{c1} = -25.0$, $y_{c1} = -2.5$, $x_{c2} = 25.0$ and $y_{c2} = 2.5$. Click *OK* to exit the dialog. The plot shows a rectangle with the desired dimensions centered in the x-y plane. We must make changes from the default position and orientation to place it in the assembly.

ADD PART

Open a dialog to create a new part in the current coil. Specify the model type and position and orientation parameters.

EDIT PART

Pick a part in the current coil and change any of the parameters.

SET CURRENT COIL

Pick the current coil for editing operations and the addition of new parts.

3.5 Position and orientation of parts and coils

You can rotate and shift individual parts within a coil and also apply global positioning operations that affect all parts of a coil. The positioning procedure is similar to the operations necessary to construct a physical coil:

- Fabricate the part on the workbench by specifying the model and the dimensions. The *workbench frame* is equivalent to the default orientation and position of the model.
- Rotate the part in three dimensions so that it has the the correct orientation relative to the coil.
- Move the part from the workbench to its position in the coil.
- When the coil is complete, you have the option to rotate or shift it.

We can choose items with different features (models) from a standard parts bin and fabricate them by setting dimensional parameters. Simple shapes like line segments may connect any two points in the workbench frame. Complex parts like helices have specific orientations and positions with respect to the workbench. The order of operations is important. In **MagWinder**, rotations always precede shifts.

ROTORDER

You can achieve any orientation in space by making rotations about the x, y and/or z Cartesian axes of the workbench coordinate system. Rotations are not commutative; therefore, the final orientation depends on the order. Use the pull-down list box to specify the order in which rotations are applied. The default is `XYZ`.

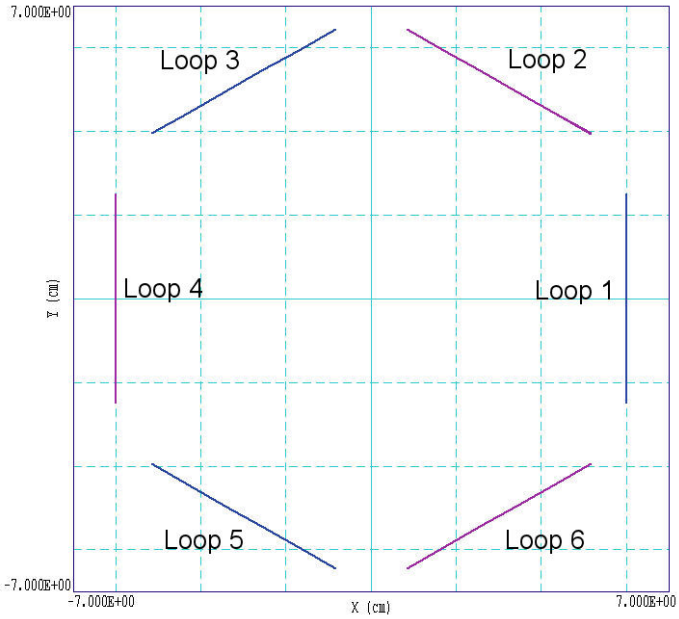


Figure 11: Positions of the six loops in the `SEXTUPOLE` example. View normal to the z axis.

THETAX
THETAY
THETAZ
Rotation angles about the x, y and z axes. Enter values in degrees. Negative angles are allowed.

XSHIFT
YSHIFT
ZSHIFT
Apply shifts along x, y and/or z to move the part from the workbench to its position in the coil.

To illustrate, we shall position the first loop in the `SEXTUPOLE` example. Click the *Edit part* command and press the button next to `LOOP01`. **MagWinder** displays the dialog of Fig. 10. Note that fields contain all currently-defined values. To make the long axis of the rectangle point along z, apply a rotation $\theta_y = 90.0^o$ by entering the number 90.0 in the `THETAY` field. Click *OK* and check the plot to confirm that the part has the correct orientation. Figure 11 shows a scaled view of the configuration in a plane normal to z. The goal is to position the six loops at 60^o intervals at a radius $R = 6.0$ cm. To move `LOOP01` to its final position, use the *Edit part* command and enter the value 6.0 in the `XSHIFT` field.

3.6 Part models

The standard parts bin contains fifteen models with which you can construct most practical magnet coils. Enter lengths in units specified by *DUnit* and angles in degrees. All models are defined with respect to the workbench coordinate frame. Their positions and orientations in the simulation space may be adjusted by specifying local or global shifts and rotations.

LINE

The line segment is the simplest and most versatile part. It is a vector that connects any two points in the workbench coordinate system. The model involves six parameters: the coordinates of the start point (x_s, y_s, z_s) and the end point (x_e, y_e, z_e). A positive value of current flows from the start point to the end. **MagWinder** attempts to cut lines into equal segments with length less than or equal to Ds. For short lines, the minimum number of segments is 2.

RECTANGLE

This model consists of four line segments that define a rectangular loop. A rectangle lies in the x-y plane of the workbench coordinate system at $z = 0.0$. The model has four parameters to define two corners of the rectangle in the x-y plane: (x_1, y_1) and (x_2, y_2). Current flows in the sense of positive rotation. (*i.e.*, if you point the thumb of your right hand along z, positive current flows in the direction of your fingers.) As with lines, the minimum number of segments per side is 2.

CIRCLE

This model defines a circular coil in the x-y plane of the workbench at $z = 0.0$. The single parameter is the radius R. The circle is centered at position $(x = 0.0, y = 0.0)$. Current flows in the direction of positive rotation. The minimum number of elements in a circle is 12.

ELLIPSE

This command creates an elliptical coil in the x-y plane of the workbench at $z = 0.0$ that follows the curve:

$$\left(\frac{x}{R_x}\right)^2 + \left(\frac{y}{R_y}\right)^2 = 1. \tag{1}$$

The two real-number parameters are R_x and R_y. Positive current flows in the direction of positive rotation. The minimum number of elements is 12.

ARC

The model requires three real-number parameters: R, θ_s and θ_e. The arc has radius R and lies in the x-y plane of the workbench with center point at $x = 0.0, y = 0.0$. The start angle (relative to the x axis) is θ_s and the end angle is θ_e. Positive current flows in the direction of positive rotation.

HELIX

This model is useful to create a circular coil or twisted wire pairs. The model involves four real-number parameters: R, Z_s, Z_e and $Pitch$. The helix winds along z and has a circular projection of radius R in the x-y plane centered at the origin. The helix extends along z from Z_s to Z_e. The quantity $Pitch$ is the axial distance for a full revolution. Therefore, the number of turns is

$$N = \frac{|Z_e - Z_s|}{Pitch}. \tag{2}$$

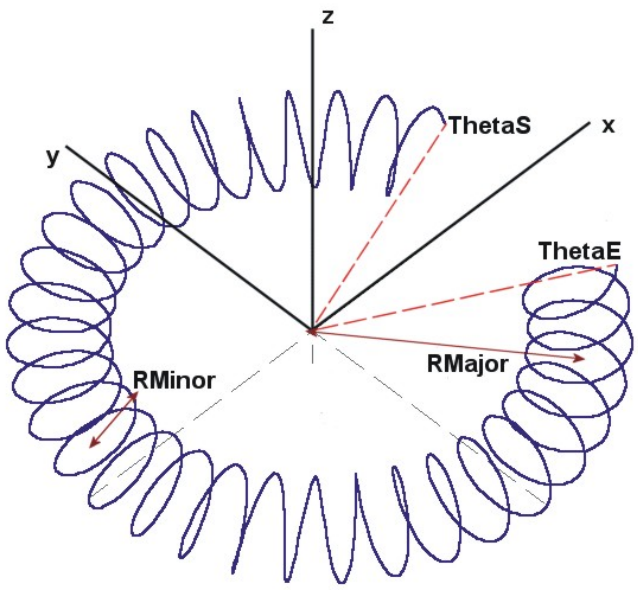


Figure 12: Parameters used in the TORUS model.

By convention, a helix starts at position $(R, 0.0)$ in the x-y plane at $z = Z_s$. The end position in the x-y plane is determined by N. Enter a value in the ThetaZ field to change the start position. The minimum number of elements per turn is 12. By default, the helix has a positive sense of rotation as it moves from Z_s to Z_e.

TORUS

The torus (Fig. 12) involves five real-number parameters R_{maj}, R_{min}, θ_{pitch}, θ_s and θ_e and the optional parameter θ_{init}. The toroidal winding creates an azimuthal field. The torus is centered at position (0.0, 0.0, 0.0) with major radius R_{maj} in the x-y plane. Figure 12 defines the parameters R_{min} and θ_{pitch}. The pitch angle (in degrees) is the rotation about the z axis per turn. A TORUS starts at angle θ_s (in the x-y plane relative to the x axis) and ends at θ_e. Current flows in the direction of positive rotation. The minimum number of elements per turn is 12. By default, the winding starts at an angle of 0.0^o relative to the plane of the minor radius. You can change this value with the parameter θ_{init}.

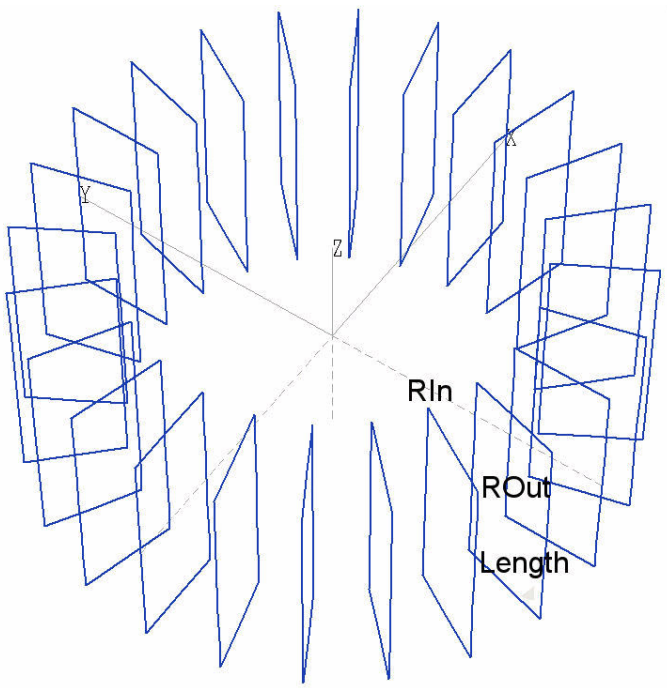


Figure 13: Parameters of the TORUSR model. The example contains 24 loops from 0.0° to 345.0°.

TORUSR

This model (Fig. 13) creates an object consisting of a set of loops with rectangular cross section arranged to form a torus. The model involves five real-number parameters (R_{in}, R_{out}, L, θ_s and θ_e) and one integer parameter N. The torus is centered at position (0.0,0.0,0.0) with inner radius R_{in} and outer radius R_{out} in the x-y plane. The parameter L is the coil height along z. The part consists of N loops that cover the range from θ_s to θ_e in the x-y plane.

The next group of models is useful for constructing volumetric coils with non-zero cross-section. The models automatically create parallel sets of elements with transverse spacing on the order of *DsGlobal* or *Ds*. All volumetric assemblies have a fixed orientation and position in the assembly coordinate system. Note that volumetric models may generate large numbers of elements. You can ensure that solutions are efficient by using volumetric assemblies only where they are necessary. For instance, suppose you want to find fields near the surface of a circular coil with a rectangular cross section. It is not necessary to model the entire circular coil using the ELBOWR model if you only need to know fields at a representative position. Instead, define a short azimuthal section of the coil with the ELBOWR model and then fill in the remainder with the ARC model. Because of the $1/R^2$ variation in the Biot-Savart integral of **Magnum**, the contribution from a distant coil segment has little dependence on the coil cross section.

SOLENOID

The solenoid model (Fig. 14) generates elements to represent a solenoid with radial thickness. It involves four required real-number parameters R_{min} R_{max} and L and three optional integer parameters N_R, N_Z and N_C. The quantity R_{min} is the inner radius, R_{max} is the outer radius and L is the axial length. The solenoid consists of a set of nested circular loops centered in

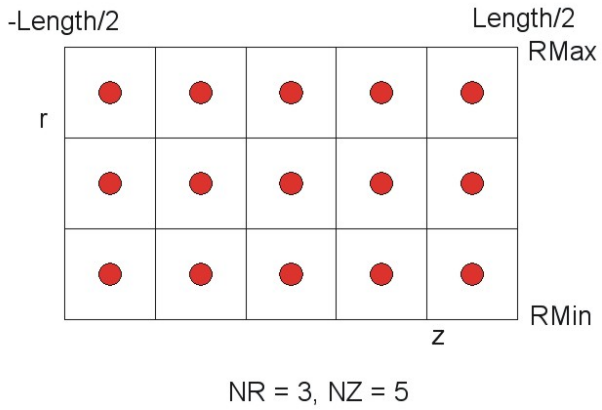


Figure 14: Arrangement of wires in the SOLENOID model.

the x-y plane and arrayed along z. The wires extend from $-L/2$ to $L/2$ in the z direction. There are three optional integer numbers: N_R (number of radial layers), N_Z (number of axial layers) and N_C (number of azimuthal segments in each circular coil). If the numbers are not specified, **MagWinder** makes a choice based on the value of *DsGlobal* or *Ds*. The coil current I_c equals the total number of A-turns in the SOLENOID. Therefore, the current per circular loop is $I_c/(N_r \times N_z)$.

BAR

A BAR is a set of straight wires parallel to z that fill a rectangular cross section in the x-y plane. The model involves three required real-number parameters (L_x, L_y and L_z) and two optional integer parameters (N_x and N_y). The required parameters are the cross section dimensions (L_x and L_y) and the axial length L_z. The BAR extends from $z = -L_z/2$ to $z = L_z/2$ in the workbench frame. The number of elements along z is determined by *DsGlobal* or *Ds*. You have the option to specify the number of parallel wires by entering values for N_x and N_y. The BAR carries a total current equal to the coil current I_c. Therefore, the current per wire is $I_c/(N_x \times N_y)$.

ELBOWR

The ELBOWR model is an elbow with a rectangular cross section. The elbow is equivalent to a SOLENOID that extends over a limited angular range. There are four required real-number parameters: R_{min}, R_{max}, L and θ. The quantities R_{min}, R_{max} and L are identical to those used for the SOLENOID. The quantity θ is the angular extent (in degrees) in the x-y plane. The ELBOWR extends from the x axis of the workbench frame to θ. There are three optional integer parameters: N_r (number of wires along r), N_z (number of wires along z) and N_θ (number of elements along the arc). Figure 15 shows a combination of the BAR and ELBOWR models to define the drive coil for a C-magnet. The ELBOWR carries a total current equal to the coil current I_c. Therefore, the current per wire is $I_c/(N_r \times N_z)$.

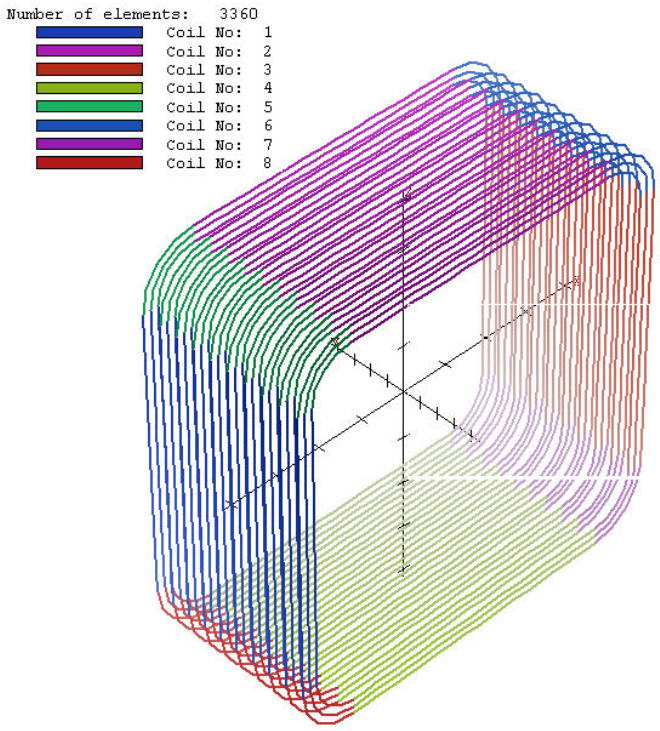


Figure 15: Magnet winding composed or four `BAR` models and four `ELBOWR` models.

ROD

A `ROD` is similar to a `BAR`. The difference is that it has an approximately circular cross section. There are two required real-number parameters: R (radius) and L (length). In this model **MagWinder** sets up a set of parallel line currents in a hexagonal pattern in the x-y plane. The hexagons have scale size on the order of $DsGlobal$ or Ds. The total current of all wires in the `ROD` equals I_c.

ELBOWC

The `ELBOWC` model represents an elbow with a circular cross section. It involves three real-number parameters: R_{out} (outer radius), R_{in} (inner radius) and θ (angular extent in the x-y plane). The `ELBOWC` model starts at the x axis and extends to θ. You can combine `ELBOWC` and `ROD` models to create a continuous set of parallel wires. The total current of all wires in the `ELBOWC` equals I_c.

POLYNOID

A `POLYNOID` (Fig 16) is similar to a `SOLENOID`. The difference is that the individual coils are polygons rather than circles. There are four required parameters. The real-number parameters R_{min}, R_{max} and L are identical to those of the `SOLENOID`. The integer N_s is the number of polygon sides. There are three optional parameters: N_r (number of radial layers), N_z (number of axial layers) and N_s (number of elements per side). The coil current I_c equals the total number of A-turns in the `POLYNOID`. Therefore, the current per circular loop is $I_c/(N_r \times N_z)$.

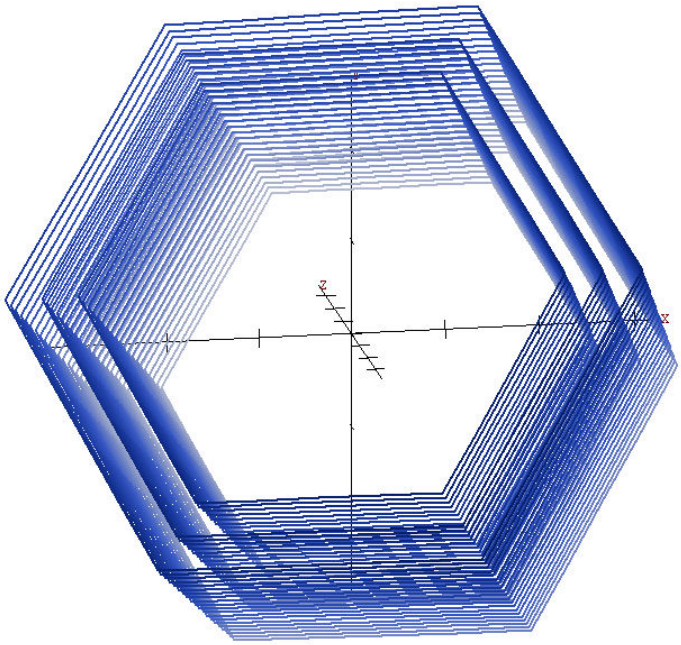


Figure 16: POLYNOID with $R_{max} = 6.0$, $R_{min} = 4.0$, $L = 8.0$, $N_s = 6$, $N_r = 3$ and $N_z = 20$.

COILBOX

A COILBOX (Fig 17) is used to produce a multi-turn magnet winding around a rectangular mandrel. In the assembly space, the currents flow along the x and y directions. There are five required real-number parameters: L_x (the full length in x), L_y (the length in y), L_z (the height in z), W (the thickness of the winding) and R (the outer radius of the four corners. The thickness W is constant around the winding as would be the case for a real assembly. The parameter R must be greater than or equal to W. When $R = W$, the inner winding has a sharp bend at the corners. Other constraints are that $R < L_x/2$ and $R < L_y/2$. The coil current I_c is divided equally between the parallel windings to give a total of I_c A-turns. Positive current flows in the direction of positive rotation viewed from the $+z$ direction. The element length and the number of model wires are determined by *Ds* or *DsGlobal*.

3.7 Editing coils and parts

Use the *Edit coil* command to change coil parameters. Changes apply to all existing and future parts associated with the coil. The command brings up the dialog of Fig. 9. The fields contain the present parameter values for the coil. The *Coil parameters* command provides a quick alternative to change selected coil properties.

COIL MANAGEMENT

This command brings up the dialog of Fig. 18. The grid has a row for each coil in the assembly. You can modify the entries in the white boxes (name, current and local element width). The *Display* check box determines whether the coil is displayed in 2D and 3D views. (Note that this setting does not affect whether the coil is recorded in the CDF and element files.) Make a choice in the *Select* column and the *Current coil* button to pick the current coil. You can delete

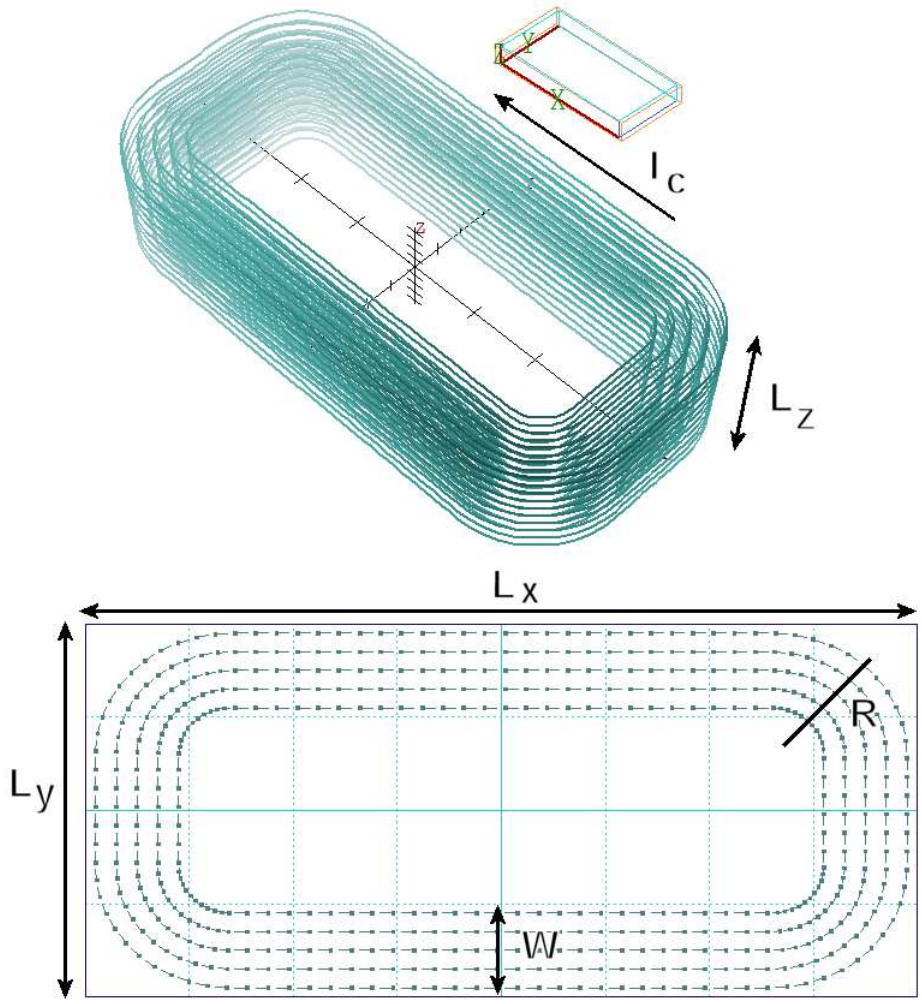


Figure 17: `COILBOX` with $L_x = 4.0$, $L_y = 2.0$, $L_z = 1.0$, $W = 0.5$, $R = 0.7$ and $Ds = 0.1$.

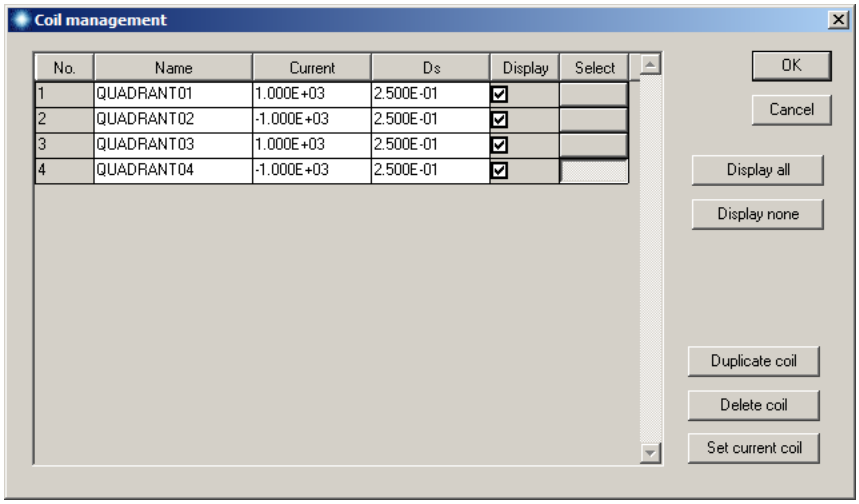


Figure 18: Coil management dialog

or duplicate a coil and its associated parts by using the *Select* column and the *Delete coil* of *Duplicate coil* button.

You can change any part in the current coil with the *Edit part* command. To modify parts in a different coil, first use the *Set current coil* command. The following commands change the properties of parts in an assembly.

DELETE PART

Remove a part from the current coil. **MagWinder** displays a list of available parts. Press the button on the right-hand side for the part you want to delete.

COPY PART

Select a part from the current coil and copy it to a temporary buffer.

CUT PART

Remove a part from the current coil and copy it to a temporary buffer.

PASTE PART

Use the contents of the temporary buffer to create a part in the current coil.

UNDO

Reverse the previous coil or part editing operation.

As an example, to move a part to a different coil, cut it from the current coil, change the current coil and then use the paste operation.

The editing operations are particularly useful for the `SEXTUPOLE` example because the assembly is composed of six similar parts with different orientations. Before making further changes,

it is a good idea to save the present state of the assembly. Use the *Save coil file* command to record your work. Then, make sure that the current coil is set to `POSITIVE`. Use the *Copy part* command to put `LOOP01` in the buffer. Then use the *Past part* command twice to add two parts to the coil. Note that the plot does not change because the three parts have identical shapes and positions.

Three parts are available when you click the *Edit part* command. Choose the second part. In the dialog, change its name to `LOOP03`. Inspection of Fig. 11 shows that we need to rotate the part by 120° and to move it to the appropriate position. Change the following fields: $ThetaZ = 120.0$, $XShift = 6.0\cos(120^\circ) = -3.000$, and $YShift = 6.0\sin(120^\circ) = 5.196$. The loop appears in the correct position when you exit the dialog. Similarly, edit the third part. Assign the name `LOOP05` and apply rotations and displacements for 240°.

To complete the assembly, we need to add parts to the `NEGATIVE` coil. Change the current coil and then use the paste operation three times to create three parts in the second coil. Use the *Edit part* command as before with the following names and rotations: `LOOP02` (60°), `LOOP04` (180°) and `LOOP06` (300°). The plot should be similar to Fig. 8. Save the completed assembly with the *Save coil file* command. We shall add the other five loops after we discuss editing commands.

3.8 Displaying an assembly

MagWinder creates plots in two modes:

- Three-dimensional plots (with perspective and shading) provide an overview of the assembly.
- Two-dimensional plots are useful for precise work.

In the 3D mode (Fig. 8), the display is divided into three sections: 1) the main plot area, 2) the orientation area (top-right) and 3) the information area (bottom-right). The orientation area shows the current view and plot boundaries. You can control the view with the mouse. The operation is similar to that in **MetaMesh** and **MagView**:

- Move the mouse cursor to the center of the plot area. Press the left button to zoom in and the right button to expand the view. Note that the change is reflected in the orientation area, and the plot is regenerated when you release the mouse buttom.
- Move the mouse cursor to the right side of the plot area. The shape changes to an arrow. Press the left button to walk around the object. Press the right button to shift the view to the right.
- Similar actions apply when you move the mouse cursor to the top, left and bottom of the plot area.

Two-dimensional plots (Fig. 11) are created by projecting elements to a chosen Cartesian plane. With the correct window size, the plots preserve true scaling.

The following commands control the plot display.

TOGGLE 2D/3D
Switch between the 2D and 3D plot modes.

DISPLAYED COILS
You can suppress the display of coils for clarity. Use the buttons in the grid box of the dialog to change the display status of individual coils. Note that all coils are recorded in coil and element files, independent of their display status.

GRID CONTROL
Use this command to suppress display of a grid or to change grid parameters from the defaults. The dialog options depend on whether the current plot mode is 2D or 3D. Three-dimensional plots show the Cartesian axes relative to an origin. You can control the following functions in the *Grid control* dialog:

- Enable or suppress display of the Cartesian axes.
- Set the origin of the Cartesian axis. If the *Fix origin* box is unchecked, then the program places the origin at the centroid of all current elements. If the box is checked, then the origin is at the position set in the text fields *X origin*, *Y origin* and *Z origin*. Enter dimensions in units set by *DUnit*.
- Enable or suppress the display of tick marks along the Cartesian axes. When Include tick marks is checked, the values in the text fields *X tick*, *Y tick* and *Z tick* are used.

In the 2D mode, you can suppress the grid and/or specify grid intervals in the horizontal and vertical directions.

TOGGLE ORTHOGRAPHIC
The default 3D plot shows a view from a distance DView from the assembly with perspective. This command toggles between orthographic and perspective views. In an orthographic plot, the viewpoint is at infinity.

ENDPOINT DISPLAY
Elements are normally plotted as thick lines. In the endpoint mode, **MagWinder** marks the start and end points of all elements. The mode is useful to check whether element sizes are appropriate.

ARROW DISPLAY
Use this command to show the orientation of current elements in 2D plots. The display is useful to check the polarity of elements of a part when rotations have been applied. Elements are displayed as sperm cells, swimming in the direction of the head. The endpoint displayed is deactivated in the arrow plot mode.

XNORMAL
YNORMAL
ZNORMAL

These commands change the projection plane in the 2D plot mode. In the 3D mode, the commands change the viewpoint to $+x$, $+y$ or $+z$.

COORDINATES

This command functions only in 2D plots. It sets the program in coordinate mode. Coordinates in the normal plane are displayed in the status bar when you move the mouse into the plot area. Click the right button or press ESC to exit coordinate mode.

INITIALIZE DISPLAY

This command performs the following functions in the 3D mode: 1) set the viewpoint to a reference with 45° azimuth and elevation, 2) set the zoom factor to include the whole assembly and 3) center the display. The command has no effect in the 2D mode.

DEFAULT PRINTER

Send the current plot (2D or 3D) to the default Windows printer. Be sure to set the desired printer before running **MagWinder**.

SAVE PLOT FILE

Use this command to create a graphics file of the current plot in either Windows Bitmap (BMP) or Portable Network Graphics (PNG) formats. In the dialog, specify the format, the size in pixels and the file prefix. The graphics file is created in the current directory.

COPY TO CLIPBOARD

Copy the current plot (2D or 3D) to the clipboard in Windows MetaFile format.

The following commands are active in 3D plots when a mesh file has been loaded.

MESH REGION DISPLAY

This command raises a dialog with a grid showing the regions of the mesh. Check the regions that should be displayed. The *Wireframe* checkbox determines whether mesh regions are displayed as solids or as wireframes. The Outline checkbox determines whether facets borders are included when regions are plotted as solids.

MESH REGION CLIPPING

You can add clip planes for the display of region boundaries to make coils more visible. In the dialog, set the minimum and maximum dimensions along the Cartesian axis for the inclusion of region facets. The default values are the maximum dimensions of the mesh (i.e., all facets included).

3.9 Additional MagWinder features

LOAD COIL FILE
Load a `CDF` file for viewing or editing. You can modify existing coils and parts or add new ones.

SAVE ELEMENT FILE
Create a `WND` file for input to **Magnum**.

LOAD MESH FILE
Load a mesh file (`MDF`) created by **MetaMesh** to display the surfaces of physical objects (iron, permanent magnets,...). **MagWinder** shows both mesh regions and coils in 3D plots. This command is a active only when a coil assembly is displayed.

CLOSE MESH FILE
Remove the display of region surfaces from 3D plots.

EDIT COIL FILE
Use the internal text editor to view or to modify the currently-loaded `CDF` file. As shown Fig. 19, the editor a has features to help you remember available options. Click on a menu entry to add a symbolic command or model definition to the text. The block operations *Comment/Uncomment* and *Indent/Unindent* work on selected text. If you make changes, be sure to reload the file with the *Load coil file* command.

VIEW ELEMENT FILE
Use the internal editor in read-only mode to inspect any `WND` file.

EDIT FILE
Use the internal editor to view or to modify any file.

MAGNUM MANUAL
Display this manual in your default PDF viewer. The file `magnum.pdf` must be in the same directory as `magwinder.exe`.

3.10 Structure of the coil definition file

The **MagWinder** script is a text file that provides a succinct description and permanent record of a coil assembly. You can build scripts using the interactive environment of **MagWinder** or compose them directly with a text editor. Sometimes, the editing approach is more efficient. For example, suppose the assembly consisted of 50 identical coils (each composed of several parts) in a linear array. It is relatively easy to copy the text for the coil section, paste it 50 times and modify $ZShift$ values to create the array.

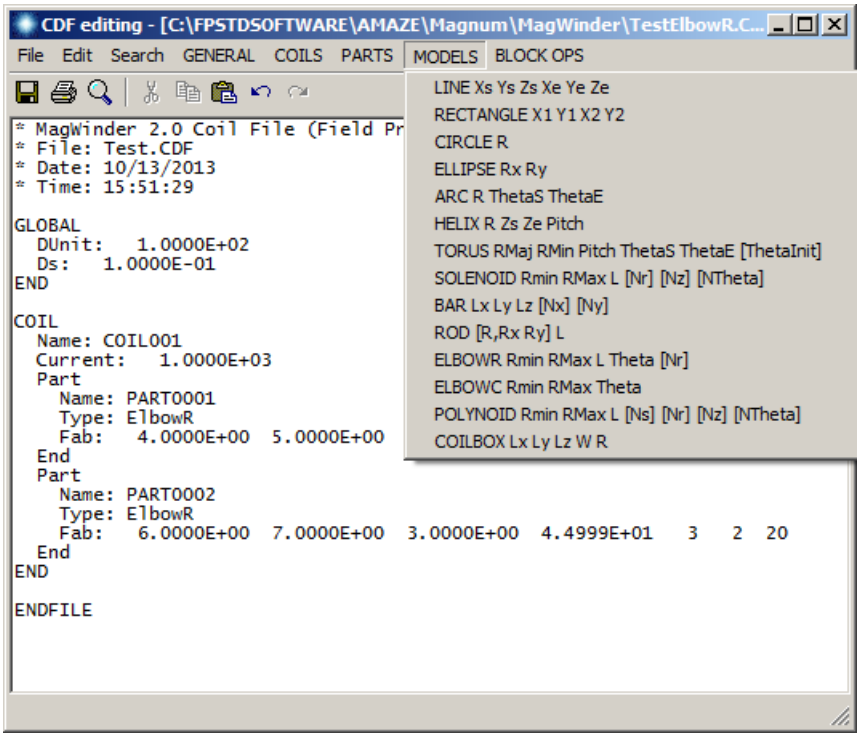


Figure 19: Menu options for the CDF file editor.

This section covers general features of the CDF file. The script consists of a set of commands processed in sequence. The file follows the standard rules for Field Precision scripts. Blank lines are ignored. You can include any number of comment lines that begin with an asterisk (*). **MagWinder** ignores all information after the *EndFile* command, so you can add annotations in any format. Commands are analyzed with a free-form parser. Sets of characters are grouped into words separated by the following delimiters: Space [ ], Comma [,], Tab, Colon [:], Equal sign [=], Left parenthesis [(] and Right Parenthesis [)]. Delimiters may be used in any combination, so you have considerable latitude to choose the appearance of your script. For example, the following commands have the same meaning:

```
Fab 2.0 0.5 10.0 0.0 350.0
Fab = (2.0, 0.5) (10.0, 0.0, 350)
Fab: 2.0, 0.5, 10.0, 0.0, 350
```

The CDF file consists of a *Global* section and up to 250 *Coil* sections. Each *Coil* section may contain any number of *Part* sections (as long as the total number of parts does not exceeded 2500). Table 5 shows the general file layout.

The *Global* section must appear at the beginning of the CDF file. The following two commands may appear in any order between the commands *Global* and *End*. They are shown in symbolic form and in the form that the might appear in the script.

Table 5: Structure of the CDF file

```
GLOBAL
   (Global commands)
END
COIL
  (Coil 1 commands)
  Part
    (Part 1 commands)
  End
  ...
END
COIL
  (Coil 2 commands)
  Part
    (Part n commands)
  End
  ...
END
...
ENDFILE
```

DUNIT DUnit
DUnit = 1.0E6
MagWinder works internally in SI units (lengths in meters). It is often convenient to enter dimensions in alternate units. The quantity *DUnit* is a factor to convert input dimensions to meters. It equals the number of units per meter. For example, to enter dimensions in inches, set $DUnit = 39.37$. If the command does not appear, the default is $DUnit = 1.0$.

DS DsGlobal
DS = 0.1
The quantity *DsGlobal* is the approximate length and diameter of current elements in units set by *DUnit*. Smaller values of element length give more accuracy but result in longer run times. It is good practice to set element lengths explicitly. In the absence of a specification, **MagWinder** will try to pick reasonable values for different part models. When a *Ds* command appears in a *Coil* section, the new value replaces *DsGlobal* for all parts of the coil.

3.11 Coil commands

A coil is a set of wires carrying the same current. **Magnum** will calculate the magnetic field corresponding to any set of wires, connected or unconnected. It is your responsibility to ensure that the coil definitions are physically correct. *Coil* sections follow the *Global* section. Commands for a coil section appear between the commands *Coil* and *End*. A *Coil* section may contain any number of *Part* sections and the following commands.

CURRENT Current
CURRENT = -125.0
A *Current* command must appear in each *Coil* section. The parameter *Current* is the amplitude of the coil current in amperes.

DS DsCoil
DS = 0.100
The Ds command sets the approximate length of current elements for the present coil. It is useful, for example, if you have a large assembly that contains small coils. If the Ds command does not appear, the default $DsGlobal$ will be used.

NAME CoilName
NAME = Orbit_Correction_Coil_015
MagWinder numbers coils in the order they appear in the file. You can also assign a descriptive name (from 1 to 80 characters). The name will be displayed if you load the `CDF` file into the **MagWinder** interactive environment.

ROTATE ThetaX ThetaY ThetaZ [RotOrder]
ROTATE 0.0 90.0 45.0 ZXY
This command specifies global rotation angles for all parts of the coil about the Cartesian axes of the assembly coordinate system. Rotations can be performed about the x, y or z axes. The parameters $ThetaX$, $ThetaY$ and $ThetaZ$ are the respective angles in degrees. The optional string parameter $RotOrder$ controls the order in which rotations are performed. The default is `XYZ`.

SHIFT XShift YShift ZShift
SHIFT -10.0 0.0 5.65
This command controls global translations of all parts of the coil. The parameters $XShift$, $YShift$ and $ZShift$ are the components of the displacement vector. Enter the displacements in the units set by $DUnit$.

To clarify, rotations and shifts are performed in the following order:

- An individual part is rotated relative to its default position in the workbench frame to the proper orientation in the coil.
- The part is shifted to its position in the coil.
- Global rotations are applied to coil parts relative to their position in the coil.
- Global shifts are applied to the coil parts.

3.12 Part commands

A *Coil* section may contain any number of *Part* sections. A *Part* section has the form:

```
PART
  (Part commands)
END
```

The following commands may appear within a *Part* section.

ROTATE ThetaX ThetaY ThetaZ [RotOrder]
ROTATE 0.0 90.0 45.0 ZXY

This command specifies local rotation angles to orient the part in the coil. Rotations are performed about the x, y or z axes. The parameters $ThetaX$, $ThetaY$ and $ThetaZ$ are the respective angles in degrees. The optional string parameter $RotOrder$ controls the order in which rotations are performed. The default is `XYZ`.

SHIFT XShift YShift ZShift
SHIFT -10.0 0.0 5.65

Perform a local translation of a part to its position in the coil. The parameters $XShift$, $YShift$ and $ZShift$ are the components of the displacement vector. Enter the displacements in the units set by $DUnit$.

TYPE

Set the model for the part. The options (discussed in Section 3.6) are `LINE`, `RECTANGLE`, `CIRCLE`, `ELLIPSE`, `ARC`, `HELIX`, `TORUS`, `TORUSR`, `SOLENOID`, `BAR`, `ELBOWR`, `ROD`, `ELBOWC` and `POLYNOID`. In addition, the `LIST` type (described below) must be defined by direct entry in the script.

FAB

Set real-number and/or integer parameters for the model. The number of entries and the interpretation of values depends on the model type. Table 6 summarizes the choices.

The `LIST` part type is useful if you cannot form a shape from the available models or if you have your own mathematical specification for a path in three-dimensional space. In this case, the *Type* command takes the form:

```
TYPE List
  x0 y0 z0
  x1 y1 z1
   ...
  xN yN zN
END
```

The effect is to create a set of N contiguous elements from (x_0, y_0, z_0) to (x_N, y_N, z_N). Each element carries the coil current in the direction from the start point to the end. Enter the coordinates in units set by $DUnit$. Each line contains three real numbers in any valid format separated by any of the valid delimiters (i.e., space, comma, tab, ...). The list may contain any number of data lines as long as the maximum number of elements (2,000,000) is not exceeded. A script may contain a maximum of 12 lists.

Table 6: *Fab* command parameters

Type	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**
LINE	x_s	y_s	z_s	x_e	y_e	z_e	
RECTANGLE	x_{c1}	y_{c1}	x_{c2}	y_{c2}			
CIRCLE	R						
ELLIPSE	R_x	R_y					
ARC	R	θ_s	θ_e				
HELIX	R	Z_s	Z_e	Pitch			
TORUS	R_{maj}	R_{min}	θ_{pitch}	θ_s	θ_e	$[\theta_{init}]$	
TORUSR	R_{in}	R_{out}	L	θ_s	θ_e	N_c	
SOLENOID	R_{min}	R_{max}	L	$[N_r]$	$[N_z]$	$[N_\theta]$	
BAR	L_x	L_y	L_z	$[N_x]$	$[N_y]$		
ELBOWR	R_{min}	R_{max}	L	θ	$[N_r]$	$[N_z]$	$[N_\theta]$
ROD	R	L					
ROD	R_x	R_y	L				
ELBOWC	R_{min}	R_{max}	θ				
POLYNOID	R_{min}	R_{max}	L	$[N_s]$	$[N_r]$	$[N_z]$	$[N_\theta]$
COILBOX	L_x	L_y	L_z	W	R		

3.13 Structure of the current element file

Table 7 shows a portion of the current-element `FPREFIX.WND` created by **MagWinder** for input to **Magnum**. The header lists the number of coils ($NCoil$) and the number of elements ($NElem$). The coil section lists the current in amperes for each coil. The element section contains $NElem$ data lines. Each line contains the associated coil number, the coordinates of the start and end points (in meters) and the element current. Note that the current of an element may be less than the coil current if the element is constructed from a volumetric model such as a `SOLENOID`. Current flows from the start point to the end point.

Table 7: Initial section of a **MagWinder** current-element file

```
Magnum Current Element File (Field Precision, Albuquerque NM)
NCoil:   2
NElem:    660

Coil     I
 No     (A)
===============
  1  1.0000E+04
  2 -1.0000E+04

Coil   XStart      YStart      ZStart
 No     (m)         (m)         (m)
=======================================
  1  5.9980E-02 -2.5000E-02  2.5000E-01
  1  5.9981E-02 -2.5000E-02  2.4000E-01
  1  5.9982E-02 -2.5000E-02  2.3000E-01
  1  5.9983E-02 -2.5000E-02  2.2000E-01
  1  5.9983E-02 -2.5000E-02  2.1000E-01
  ...
          XEnd        YEnd        ZEnd       I
           (m)         (m)         (m)      (A)
      ================================================
        5.9981E-02 -2.5000E-02  2.4000E-01  1.0000E+04
        5.9982E-02 -2.5000E-02  2.3000E-01  1.0000E+04
        5.9983E-02 -2.5000E-02  2.2000E-01  1.0000E+04
        5.9983E-02 -2.5000E-02  2.1000E-01  1.0000E+04
        5.9984E-02 -2.5000E-02  2.0000E-01  1.0000E+04
        ...
```

4 Magnum script commands

4.1 Control commands for all calculation modes

This chapter reviews the set of commands that may be included in **Magnum** scripts. The first group, discussed in this section, controls program operation in any type of calculation. Each command is shown symbolically and in a form that it might assume in a program.

SOLTYPE [Standard, Free]
SOLTYPE = Standard

This required command sets the **Magnum** solution type. Options for the single parameter are the strings *Standard* and *Free*. A *Standard* solution includes the effects of materials, either ideal conductors, permanent magnets or ferromagnetic materials. Usually, the input mesh (`MDF`) has conformal elements that follow the material boundaries. For solutions with coils, the currents defined by the element file (`WND`) interact with the materials. In the *Standard* mode, **Magnum** generates finite-element solutions for the reduced potential ϕ (and the dual potential ψ when some materials have $\mu_r \gg 1.0$). In the *Free* mode, there are no materials and the magnetic field is created only by applied currents. In this case the function of the mesh is to define node points at which to calculate $\mathbf{H}_s$ from the given current elements. The condition $\mu_r = 1.0$ applies over the full solution volume. The boundaries of the solution volume have no effect on the field values – the solution constitutes a segment of a solution in infinite space. In the *Free* mode, **Magnum** makes no finite-element calculation and sets the values $\phi = 0.0$ and $\psi = 0.0$ in the output file.

For clarification, the following paragraphs describe typical calculations in the three modes.

- **Standard Mode**. The **MetaMesh** script `MPREFIX.MIN` describes a conformal mesh where regions represent magnetic materials such as poles and permanent magnets. The mesh generator creates the file `MPREFIX.MDF`. If the simulation includes coils, the file `SPREFIX.CDF` defines the coil geometries. From this input **MagWinder** generates the file `SPREFIX.WND`. The three files required by Magnum for a solution with coils are `MPREFIX.MIN`, `SPREFIX.WND` and `RUNNAME.GIN`. **Magnum** computes the node values of H_{xs}, H_{ys}, H_{zs}, ϕ and ψ. The quantities are used in **MagView** to determine total fields resulting from coils and materials.

- **Free Mode**. In most *Free* mode calculations the input mesh is non-conformal (box elements) and contains a single region to represent vacuum. Nonetheless, you could employ a conformal mesh with multiple regions if you wanted to use the capabilities of **MagView** to find field energy in or flux through shaped objects. A current-element file `SPREFIX.WND` is required in this mode. **Magnum** computes nodal values of H_{xs}, H_{ys}, H_{zs} and sets $\phi = 0.0$ and $\psi = 0.0$ at all points (*i.e.*, no material contributions).

The following program control commands may appear in all modes.

MESH MPrefix
MESH = TriggerConfig
The prefix of the **MetaMesh** output file (`MPrefix.MDF`) to use for the definition of the solution geometry. If the command does not appear in a script `RUNNAME.GIN`, **Magnum** searches for the default file `RUNNAME.MDF`.

SOURCE SPrefix
SOURCE = CMagnet
This command specifies an element file that lists drive currents for the applied fields. The parameter *SPrefix* is the prefix of a fill with a name of the form `SPREFIX.WND`. The program issues an error message if the file is not available in the working directory. The currents are used to generate applied fields in both the *Standard* and *Free* modes.

DUNIT Unit
DUNIT = MM
DUNIT = 100.0
Use this command to sets units for coordinates in MDF input file. The *Unit* argument may be one of the following strings for common units: `angstrom`, `nanometer`, `micrometer`, `mil`, `mm`, `cm`, `inch`, `foot`, `yard`, `meter`, `kilometer` or `mile`. For custom units, enter a real number equal to the number of mesh units per meter. For example, to signal that you used dimensions of centimeters in the **MetaMesh** script, set $Unit = 100.0$. Alternatively, if the mesh dimensions are in inches, use $Unit = 39.37$. The quantity *Unit* is recorded in the output file and is used in **MagView** for the input and output of positions. Default: $Unit = 1.0$.

FORMAT [Text, Binary]
FORMAT = Text
By default, **Magnum** creates an output file `FPREFIX.GOU` in binary format. You can create output files in text format to make it easier to port results to your own analysis programs. The string parameter can assume the values *Binary* or *Text*. **MagView** can read files in either format. Chapter 15 describes the file structure for both formats.

PARALLEL
This command invokes multiprocessor support for the 64-bit program running on a multi-core machine. In this case, **Magnum** will use the full resources of the computer during the matrix inversion, giving a substantial reduction of the run time. Otherwise, the operating system will assign the program to one processor, leaving the other free for other tasks. Do not employ parallel processing if you are running multiple instances of **Magnum**. The command has no effect on the 32-bit program version.

4.2 Field-solution control commands

The following commands are valid only in *Standard* mode calculations.

RESTARGET Restarget
RESTARGET: 5.0E-6

The numerical calculation of fields with magnetic materials requires the solution of a large set of coupled linear equations, one for each active node in the solution volume. **Magnum** uses an iterative technique based on corrections that reduce the error in the reduced or dual potential at a point compared to predictions from values at neighboring nodes. The residual is an average of the relative errors over all nodes in the solution space during an iteration. This command sets a target value for the residual. The program stops if the error drops below the value. For good accuracy, the relative residual should be less than 10^{-6}. If the value of *ResTarget* is too low, the program may not converge because of roundoff errors. In this case, you can terminate a solution manually if you are running **Magnum** in the interactive mode by using the Stop run menu command. Default value: $ResTarget = 1.0 \times 10^{-7}$.

MAXCYCLE MaxCycle
MAXCYCLE = 2500

The integer parameter *MaxCycle* is the maximum number of iteration cycles. **Magnum** saves the solution and stops when it reaches *MaxCycle* even if *ResTarget* has not been attained. Default value: $MaxCycle = 2000$.

OMEGA Omega [OmegaDual]
OMEGA: 1.92

The real-number parameter *Omega* is the over-relaxation factor used to correct potential errors during the solution for the reduced potential. This quantity may have a significant effect on the run time. If the command does not appear, **Magnum** automatically picks values for the reduced-potential calculation that vary with the iteration cycle following the Chebyshev acceleration prescription. With this command, you can set a value manually. The value must be between 0.0 and 2.0 for numerical stability. Higher values generally give faster convergence. Optionally, you may include an independent value for the relaxation factor to use for the dual-potential calculation (inside iron regions).

DUALPOT

Generate and record a dual-potential solution in additional to the reduced potential solution. The values may be used to calculate **B** in the *DualPot* interpolation mode of **MagView** for improved acccuracy in regions where $\mu_r \gg 1.0$. (see Sect. 11.4).

INITVAL FPREFIX
INITVAL = DIPOLE01

In some circumstances, you can reduce the solution time significantly for large meshes by using precomputed initial values. Magnum normally sets the quantities H_{xs}, H_{ys}, H_{zs}, ϕ and ψ

equal to zero at all nodes that do not have the fixed-potential condition. When this command appears, the program sets initial values at variable nodes equal to those in the file `FPREFIX.GOU`. One application for the command is to run additional relaxation cycles to improve the accuracy of an existing solution. You can also use the command if you make small changes to applied current or the geometry of materials. The solution file `FPEFIX.GOU` should be available in the working directory and must have the same foundation mesh as the input file `MPREFIX.MDF`. In other words, the values of I_{max}, J_{max}, K_{max}, x_{min}, y_{min}, z_{min}, x_{max}, y_{max} and z_{max} must be the same. Within this limitation, there may be small geometric differences between objects in the two files. If there are large differences, the procedure may not significantly reduce the solution time.

NCHECK NCheck
NCHECK = 20
The operations to calculate *ResTarget* take time; therefore, **Magnum** usually does not make a check on every iteration cycle. The integer parameter is the number of relaxation cycles between convergence checks. Default value: $NCheck = 25$.

CHECKIRON [Off, On]
CHECKIRON = Off
By default, **Magnum** checks that no drive current elements pass through iron regions. The process is time-consuming if there are many elements. Use this command to deactivate the check if you are sure that the mesh geometry and coil set are consistent.

BUNI [X,Y,Z] B0
BUNI(Z) = 0.05
This command is useful to model small magnetic objects (steel or permanent magnets) immersed in a uniform field. The flux density may point in the x, y or z direction and has magnitude *Bo* (tesla). In response, **Magnum** sets values of the applied field at all points in the solution volume to fixed, uniform values. To use this command, you must assign a node plane region in **MetaMesh** to the solution-volume boundaries along the field direction. For example, there would be parts of type *BoundZDn* and *BoundZUp* for a flux density $B_o\mathbf{z}$. If the parts have region number 5, then the command:

```
Potential(5) = 0.0
```

should appear in the **Magnum** script. The *BUni* and *Source* commands may not both appear in a script. The field components created by magnetic objects are constrained to be normal to the two boundaries along the applied field direction and parallel to the other boundaries. Therefore, you should leave some space around the objects to approximate a free-space condition.

4.3 Commands for material properties

Material properties in magnetostatic solutions with linear materials are relatively simple so only a few commands are required. Both symbolic and sample forms given. These commands apply only in the *Standard* mode.

MU RegNo MuR
MU(4) = 523.0
This command sets the relative magnetic permeability for all elements associated with a filled region. The integer parameter is the number of a region defined in **MetaMesh**. The second parameter is the value of relative magnetic permeability, $\mu_r = \mu/\mu_0$ (dimensionless). As discussed in Sect. 11.3, assign a small value of magnetic permeability (*e.g.*, 10^{-4}) to model a highly-conductive material that excludes magnetic fields. Note that excessively small or large values of μ_r may inhibit solution convergence. The default for all unspecified regions is $\mu_r = 1.0$.

POTENTIAL RegNo Phi
POTENTIAL(5) = 0.0
The main use of this command is to implement symmetry boundaries where the magnetic flux density is normal to the surface. The value applies to both ϕ (in the reduced potential calculation) and to ψ (in the dual potential calculation). The material contribution to the magnetic flux density is normal to a boundary with $\phi = constant$, B_m(parallel) = 0.0. The total magnetic flux density is normal to a boundary with $\psi = 0.0$, B(parallel) = 0.0. You may pick any value for ϕ or ψ when there is a single symmetry boundary in a solution. In this case, only relative differences in the potential are significant. You must exercise caution when there are two boundaries. Setting fixed values may introduce artifices. In the example of Sect. 1.2, the symmetry of the fields dictates that the line integral of $\mathbf{B}$ and $\mathbf{B}_m$ between the planes $x = 0.0$ and $y = 0.0$ is always zero. Therefore, it is valid to set $\phi = 0.0$ and $\psi = 0.0$ on both boundaries.

PERMAG RegNo Br Ux Uy Uz
PERMAG(6) = 1.12 (0.00, 0.00, 1.00)
This command defines an ideal permanent magnet material. Four real-number parameters follow the region number. The quantity Br is the remanence magnetic field (in tesla). The quantities $[Ux, Uy, Uz]$ define a vector that points along the axis of magnetization (*easy axis*). **Magnum** normalizes these numbers to define a unit vector.

An ideal permanent magnet is one in which the state of domain orientation does not depend on external influences such as the dimensions of air gaps or the presence of coils and other permanent magnets. In this case the normal demagnetization curve (B versus $\mu_0 H$ along the magnetization axis) is a straight line with a slope of -45°. Modern materials like neodymium-iron and samarium-cobalt satisfy this condition. The model applies to conventional magnet materials (like Alnico) only for moderate values of coercive force ($H \ll H_c$). For a discussion of the properties of permanent magnets, see S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997), Sects. 9.6 and 9.7. Table 8 lists properties of some common permanent-magnet materials. Figure 20 shows a two-dimensional benchmark calculation performed with **Magnum**. The example of a ring dipole is taken from J.L. Warren, *et.al.*, **Reference Manual for the Poisson/Superfish Group of Codes** (Los Alamos National Laboratory, LA-UR-87-126, 1987), pg. 10-17. The theoretical value for the dipole field on axis is 1.000 tesla.The **Magnum** calculation gives 1.003 tesla.

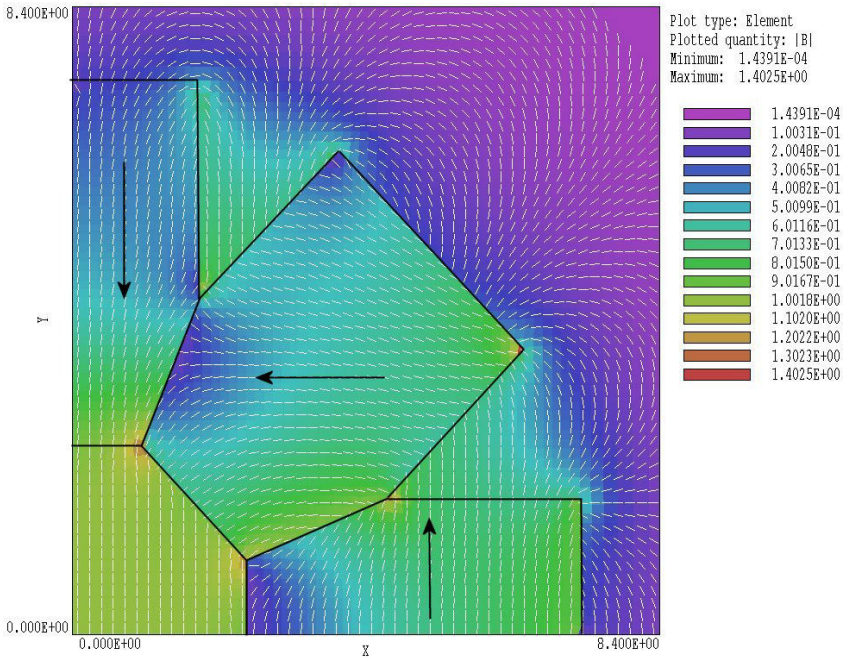


Figure 20: **Magnum** calculation of a permanent-magnet ring dipole (single quadrant with symmetry boundaries). Vector lines with color-coding by $|\mathbf{B}|$. Black arrows show magnetization vectors.

The following Internet sites provide useful information on permanent magnet design:

`http://www.magnetsales.com/Design/DesignG.htm`
`http://www.magnetweb.com/design2.htm`
`http://www.magnet.au.com/fluxgraphs-entry.html`
`http://www.intemag.com/designtools/designguide.asp`

4.4 Modeling small details in a large-scale solution

Magnum features a useful technique for calculations of fields near small objects in a large solution space. Suppose we wanted a precise calculation of the effect of small changes in the profile of a magnet pole face. One option is to perform a series of calculations for the full magnet with small elements to resolve details of the pole face. This approach is inefficient for two reasons:

- It involves repeated recalculation of fields in regions remote from the gap that are relatively unaffected by the changes.
- The run time is long because of the large mesh size.

Figure 21 illustrates an alternate approach. We carry out a global solution at relatively coarse mesh resolution and then create a second solution that covers a subvolume of the original (green outline). The microscopic solution has smaller elements and resolves details of magnetic materials. The question is how to incorporate the macroscopic fields correctly into

Table 8: Properties of some permanent magnet materials (all values in tesla)

Material	B_r	$\mu_0 H_c$	Material	B_r	$\mu_0 H_c$
17% Cobalt steel	0.95	0.015	NdFeB 27	1.05	1.00
36% Cobalt steel	0.95	0.024	NdFeB 27H	1.085	1.00
Alnico 1	0.72	0.044	NdFeB 28	1.08	1.01
Alnico 12	0.58	0.095	NdFeB 30	1.10	1.06
Alnico 2	0.72	0.055	NdFeB 30H	1.12	1.07
Alnico 2 (sintered)	0.69	0.052	NdFeB 32	1.16	1.08
Alnico 4	0.55	0.070	NdFeB 32H	1.15	1.09
Alnico 5	1.25	0.055	NdFeB 35	1.23	1.13
Alnico 5 (sintered)	1.09	0.062	NdFeB 35H	1.21	1.16
Alnico 6	1.00	0.075	NdFeB 37	1.24	1.18
Alnico 8	0.82	0.165	Neo Flex6	0.545	0.49
Alnico 8 (sintered)	0.74	0.150	Platinum-cobalt	5.90	0.36
Carbon steel	1.00	0.005	Remalloy or Comol	1.05	0.025
Ceramic 1	0.23	0.186	Silmanal	0.055	0.60
Ceramic 5	0.38	0.24	SmCo 16	0.80	0.76
Ceramic 8	0.385	0.295	SmCo 18	0.85	0.80
Chromium steel	0.97	0.0065	SmCo 20	0.905	0.83
Cunico	0.34	0.066	SmCo 22	0.95	0.86
Cunife (wire)	0.54	0.055	SmCo 22B	0.96	0.78
Flexible regular	0.16	0.137	SmCo 24	0.995	0.72
HF1	0.22	0.20	SmCo 26	1.05	0.92
HF2	0.245	0.22	SmCo 28	1.075	0.84
HF3	0.265	0.22	Tungsten steel	1.03	0.007
Hyflux powder	6.60	0.039	Vectolite	1.60	0.10
NdFeB 24	0.98	0.95	Vicalloy 1	0.88	0.030
NdFeB 26	1.04	1.00	Vicalloy 2 (wire)	1.00	0.051

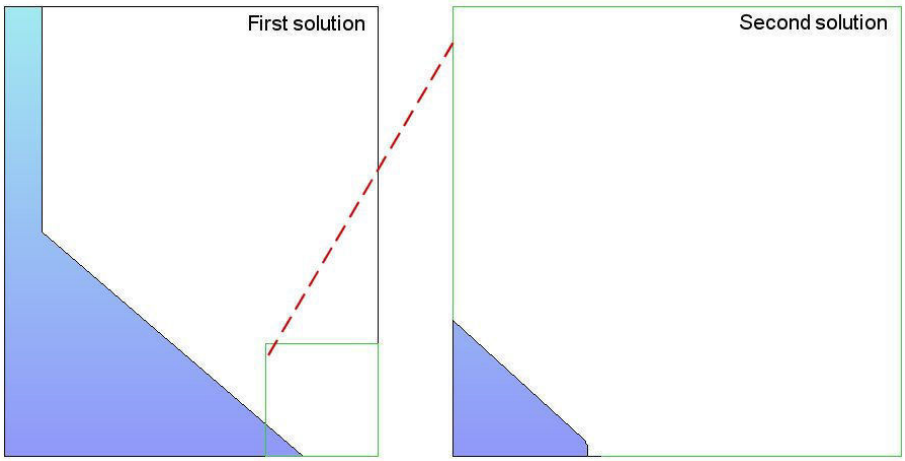


Figure 21: Creating a microscopic solution using the *Boundary* command.

the microscopic solution. One resolution is to enclose the second solution within a Dirichlet boundary. Values of the potentials ϕ and ψ on the microscopic solution boundary are calculated by interpolation at the corresponding point in the macroscopic solution (dashed red line). The total solution will be approximately correct as long as new features (such as the rounded tip in Fig. 21) are well-removed from the boundaries.

Definition of the variable Dirichlet boundary is controlled by the following command that appears in the control script of the microscopic solution:

BOUNDARY FPrefix [BndScale] [BoundStat]
BOUNDARY FEmitMacro 1.0 111110

The quantity *FPrefix* (a string) is the prefix of the output file for the macroscopic solution, `FPREFIX.GOU`. The quantity *BndScale* (a real number) is an optional scaling factor applied to potential values transferred from the macroscopic to the microscopic solution. The default is *BndScale* = 1.0. The optional parameter *BoundStat* is a string with six characters, either '0' or '1'. It can be used to control whether a specific boundary is assigned the computed Dirichlet condition. The order of boundaries in the string is XDn, XUp, YDn, YUp, ZDn and ZUp. A entry of '1' indicates that the Dirichlet condition should be applied. The default is *BoundStat* = '111111'.

In response to the *Boundary* command, **Magnum** loads the macroscopic solution into memory and sets up the apparatus for making interpolations of reduced potential ϕ and dual potential ψ. **Magnum** loops through all nodes of the microscopic solution and performs a field interpolation to set an initial value from the macroscopic solution. If a node is on an active boundary of the solution volume, the point is marked as *fixed potential*. This means that potential values will not change during the relaxation solution. **Magnum** issues an error message under the following conditions: 1) the output file for the macroscopic solution is not available in the working directory, 2) the microscopic solution volume does not fit completely inside the macroscopic solution or 3) an interpolation failed. The program makes no further validity checks.

The procedure is best illustrated with an example. Input files for the runs `MAGNUMBOUND` and `MAGNUMBOUNDS` are included with the package. The files `MAGNUMBOUND.MIN`, `MAGNUMBOUND.GIN` and `MAGNUMBOUND.CDF` describe a large-scale solution for a simple, cylindrical laboratory magnet with a flat upper and lower pole faces. A microscopic solution covering the volume near the magnet gap is defined by the files `MAGNUMBOUNDS.MIN` and `MAGNUMBOUNDS.GIN`. The **MetaMesh** input file from the microscopic solution contains the same part definitions as those in the file for the macroscopic solution. **MetaMesh** automatically clips the parts and includes only portions that fit in the smaller volume. The file `MAGNUMBOUNDS.MIN` has smaller mesh elements and includes an additional part that represents a thin extension on the outside of the pole. The extension (with thickness 0.150 cm in r and 0.0625 cm in z) extends over the range $0.0 \leq \theta \leq \pi/2$.

After creating, meshes for both solutions, we run **Magnum** to generate the output file for the macroscopic, `MAGNUMBOUND.GOU`. The solution time is about 10 minutes. Figure 23 (top) shows the variation of $|\mathbf{B}|$ in the gap. To conclude, we to run the microscopic solution using the script listed in Table 9. Note the presence of the *Boundary* command. Although the boundary is fixed, it is necessary to include the *Source* command to set applied field values inside the

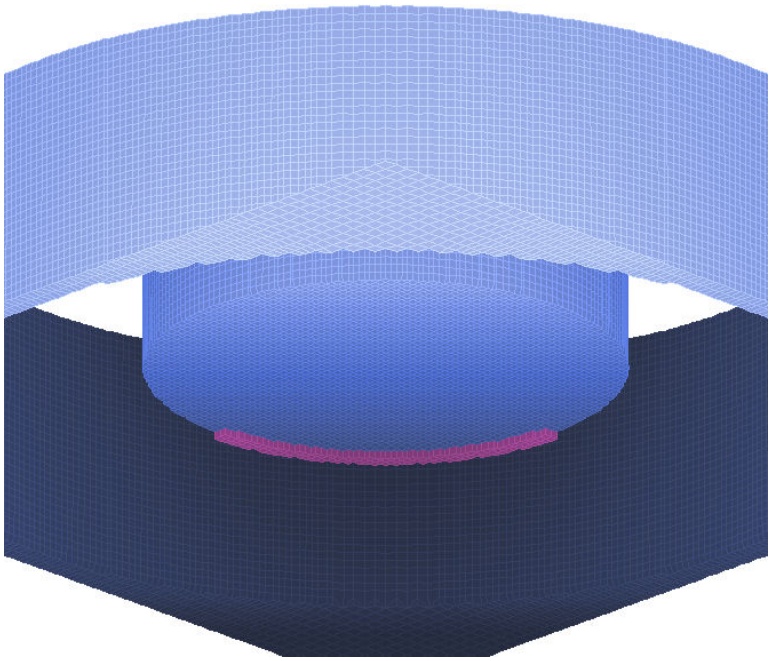


Figure 22: Example `MAGNUMBOUND`: three-dimensional view of the upper pole face in the microscopic solution showing the pole extension (violet).

micrsopic volume. Figure 23 (bottom) shows the resulting field. The effect of the asymmetric extension is apparent. Note the improved accuracy for fields at all positions resulting from the finer mesh. The solution time is less than 5 minutes. The procedure is valid as long as the new details in the microscopic solution do not significant change the properties of the global magnetic circuit. In this case, the presence of the extension should make a small change in the gap reluctance.

In response to the *Superposition* command, **Magnum** superimposes values from a large-scale solution (macroscopic) on a small-scale solution (microscopic). The following statement may appear in the command script for the microscopic solution:

SUPERPOSITION FPrefix [SScale]
SUPERPOSITION UniField

The quantity *FPrefix* (a string) is the prefix of the output file for the macroscopic solution. The quantity *SScale* (a real number) is a scaling factor applied to potential values transferred from the macroscopic to microscopic solutions. The default is $BndScale = 1.0$.

When the *Superposition* command is issued, **Magnum** opens the file `FPREFIX.GOU`. The program performs an interpolation in this solution space to determine the reduced potential ϕ_{mac}, applied magnetic field $\mathbf{H}_{mac}$ and dual potential ψ_{mac} at each node location in the microscopic solution and adjusts values according to:

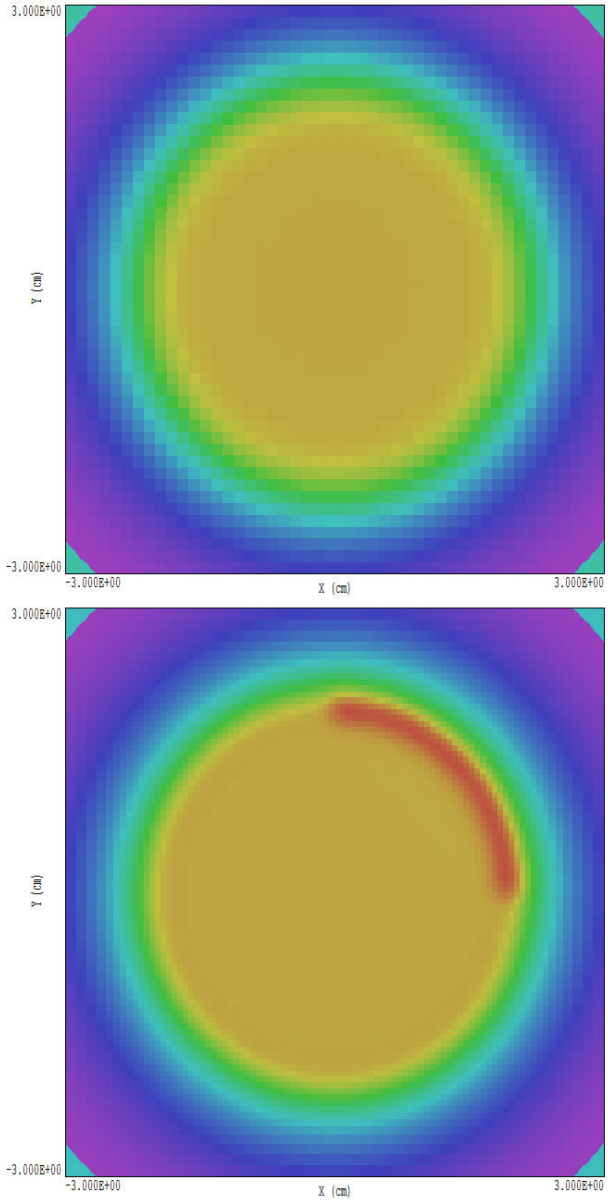


Figure 23: Example `MAGNUMBOUND`: details of the variation of $|\mathbf{B}|$ in the x-y plane at $z = 0.281$ cm. Values range: 0.0 tesla (violet), 1.35 tesla (red). Top: macroscopic solution. Bottom: microscopic solution with pole extension.

Table 9: Contents of the script `MAGNUMBOUNDS.GIN`

```
SolType = STANDARD
Mesh = MAGNUMBOUNDS
Source = MAGNUMBOUND
MaxCycle = 300
ResTarget = 1.0E-7
Boundary = MAGNUMBOUND
DUnit =   1.0000E+02
Parallel
Mu(1) = 1.0
Mu(2) = 2000.0
EndFile
```

$$\phi'_{mic} = \phi_{mic} + SScale \times \phi_{mac}, \tag{3}$$

$$\mathbf{H}'_{mic} = \mathbf{H}_{mic} + SScale \times \mathbf{H}_{mac}, \tag{4}$$

$$\psi'_{mic} = \psi_{mic} + SScale \times \psi_{mac}, \tag{5}$$

before writing the output file. You must ensure that the superposition is physically correct. A superpositions is not valid for solutions where perfect conductors or ferromagnetic materials in the microscopic solution make significant local changes to the field of the macroscopic solution.

5 Nonlinear materials and program restart

5.1 Ferromagnetic materials in strong magnetic fields

Atomic currents in ferromagnetic materials align in the presence of an applied magnetic field. These currents may make a large contribution to the total magnetic field. Because of this property, ferromagnetic materials play critical roles in magnets, inductors and shields. For a brief review of the properties of material response in magnetic circuits, see S. Humphries, **Principles of Charged Particle Acceleration** (Wiley, New York, 1986), Sects. 5.3 – 5.8 (available at http://www.fieldp.com/cpa/cpa.html).

Following the discussions in Chap. 11, the magnetic flux density vector $\mathbf{B}$ in an isotropic magnetically-active material is related to the magnetic field intensity vector $\mathbf{H}$ by:

$$\mathbf{B} = \mu_r \ (\mu_0 \mathbf{H}) = \mu_r \mathbf{B}_0, \tag{6}$$

where μ_r is the *relative magnetic permeability* of the material. Equation 6 defines the quantity

$$\mathbf{B}_0 = \mu_0 \mathbf{H}, \tag{7}$$

The applied magnetic flux density is proportional to the magnetic field and has the same units (tesla) as the magnetic flux density $\mathbf{B}$. We can write the relative magnetic permeability as:

$$\mu_r = \frac{\mathbf{B}}{\mathbf{B_0}}. \tag{8}$$

In *linear* materials, μ_r has a fixed value independent of the field amplitude. In this case, the equations discussed in Chap. 11 are linear – all terms in the governing equation involve at most the first power of $\mathbf{H}$. Consequently, the solution procedure is straightforward. In contrast, the equations are nonlinear when μ_r depends on $|\mathbf{H}|$. Nonlinear solutions are much more difficult, and there is no guarantee that all boundary-value problems have a unique answer.

Nonlinear behavior in ferromagnetic materials results from *magnetic saturation*. At low field levels, the fractional alignment of atomic currents is roughly proportional to $\mathbf{B}_0$ so that μ_r is approximately independent of field amplitude. At high values of $\mathbf{B}_0$, most of the atomic currents are aligned so the the material contribution cannot increase. Therefore, μ_r decreases with rising field amplitude. The *saturation magnetic flux density* B_s is the maximum available contribution from the material. Typical values are $B_s \sim 0.5$ tesla for ferrites and $B_s \sim 2 - 3$ tesla for iron.

The two-dimensional **PerMag** code calculates vector potential, equivalent to $\mathbf{B}$. For this code, it is most convenient to express the relative magnetic permeability as a function of the amplitude of magnetic flux density, $\mu_r(|\mathbf{B}|)$. In contrast, **Magnum** employs the reduced-potential method and solves for the total magnetic field intensity. Therefore, it is necessary to express μ_r as a function of $|\mathbf{B}_0|$.

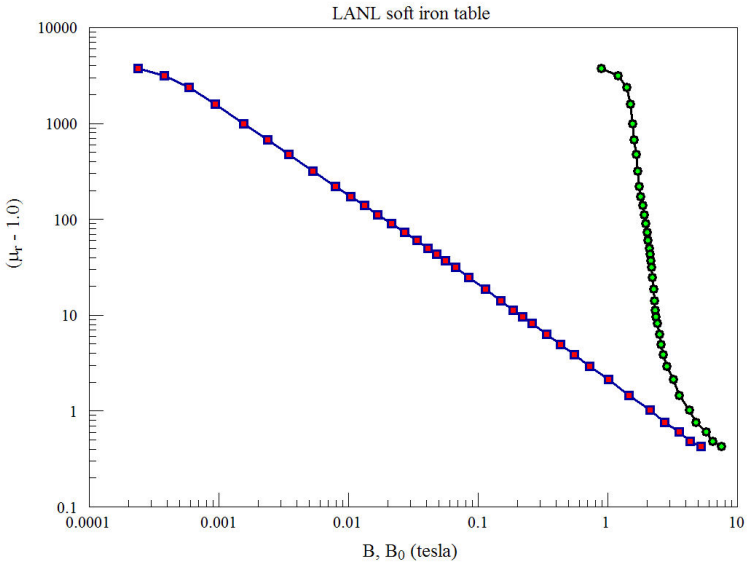


Figure 24: Relative magnetic permeability of soft iron as a function of $|\mathbf{B}|$ (green circles) and $|\mathbf{B}_0|$ (red squares). Adapted from the Superfish/Poisson Manual, Los Alamos National Laboratory.

5.2 Preparing material tables

In this section, we shall discuss how to represent data for nonlinear magnetic materials for inclusion in **Magnum**. Figure 24 plots values from a table for generic soft iron prepared by Los Alamos National Laboratory and widely used in numeric codes. Each data line in the table contains a value of $|\mathbf{B}|$ and the corresponding value of μ_r. Because μ_r must approach unity at field levels above saturation, we have plotted the quantity $(\mu_r - 1)$. The green circles show values in the table for $\mu_r(|\mathbf{B}|)$. The function is highly nonlinear with a sharp drop near the saturation field $B_s \sim 3.0$ tesla. The table can be converted to the form $\mu_r(|\mathbf{B}_0|)$ simply by dividing the $|\mathbf{B}|$ entries by μ_r. The results are plotted as red squares in Fig. 24. Note that this variation is smoother and simpler. For this reason, **Magnum** converges more effectively than codes based on a direct solution for the vector potential.

It is often more convenient to display the properties of magnetic materials in terms of the quantity:

$$\gamma = \frac{1}{\mu_r} = \frac{\mathbf{B}_0}{\mathbf{B}}. \tag{9}$$

This representation has two advantages:

- The quantity γ has a well-defined range, 0.0 to 1.0.
- Material contributions to the solution are roughly proportional to γ (*i.e.*, a solution for a material with $\mu_r = 10,000$ is almost indistinguishable from that for a material with $\mu_r = 2,000$.)

Magnum makes changes in values of γ to seek a self-consistent solution. Figure 25 shows the data of Fig. 24 plotted in terms of γ. The relative simplicity of the curve $\gamma(|\mathbf{B}_0|)$ is apparent.

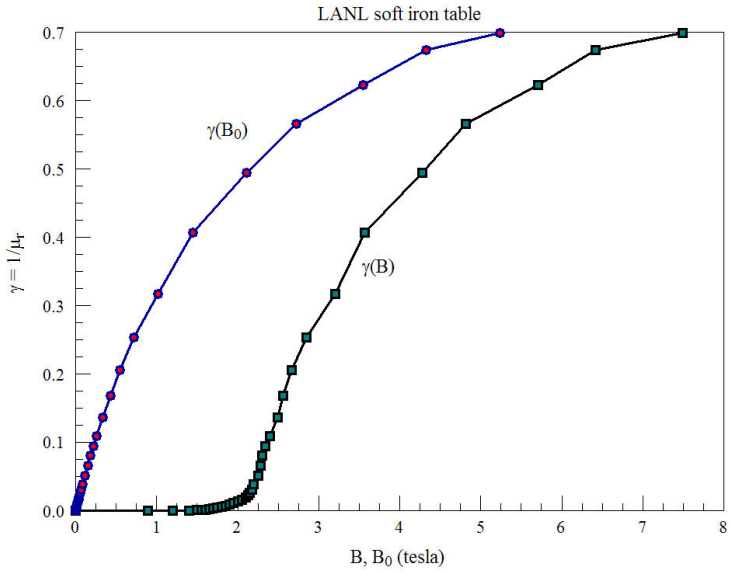


Figure 25: The parameter $\gamma = 1/\mu_r$ for soft iron as a function of $|\mathbf{B}|$ (green squares) and $|\mathbf{B}_0|$ (red circles). Adapted from the Superfish/Poisson Manual, Los Alamos National Laboratory.

Magnum requires tables with data lines that contain a value of $|\mathbf{B}_0|$ and a corresponding value $\mu_r(|\mathbf{B}_0|)$. Lines should appear in order of increasing $|\mathbf{B}_0|$ and should define a smooth curve. We have supplied tables for several common materials in the example library. If you have a tabulated hysteresis curve (B versus H), you can use the *Mu table generator* command in the **Magnum** *Help* menu to convert the data to the standard form (Sect. 5.7).

Measured data on magnetic materials often cover a limited range in B_0. You may need values of μ_r at high field values to investigate saturation or to ensure that a solution converges. In many cases it is possible to extrapolate experimental curves analytically using a spreadsheet. Suppose we have data that extends to the saturation range for a material. The term *saturation* implies that the applied field is so strong that all magnetic domains in the material are aligned. In this case, the maximum contribution of the material to the flux density is B_s. Therefore, the total magnetic flux density approaches:

$$B \cong B_s + B_0. \tag{10}$$

Using Eq. 10, we can write the relative magnetic permeability (B/B_0) as:

$$\mu_r \cong 1 + \frac{B_s}{B_0}. \tag{11}$$

We can invert Eq. 11 to derive an expression for the saturation flux density:

$$B_s \cong (\mu_r - 1)\ B_0. \tag{12}$$

Assuming that you have ported the experimental values of B_0 and μ_r to a spreadsheet, use the following procedure.

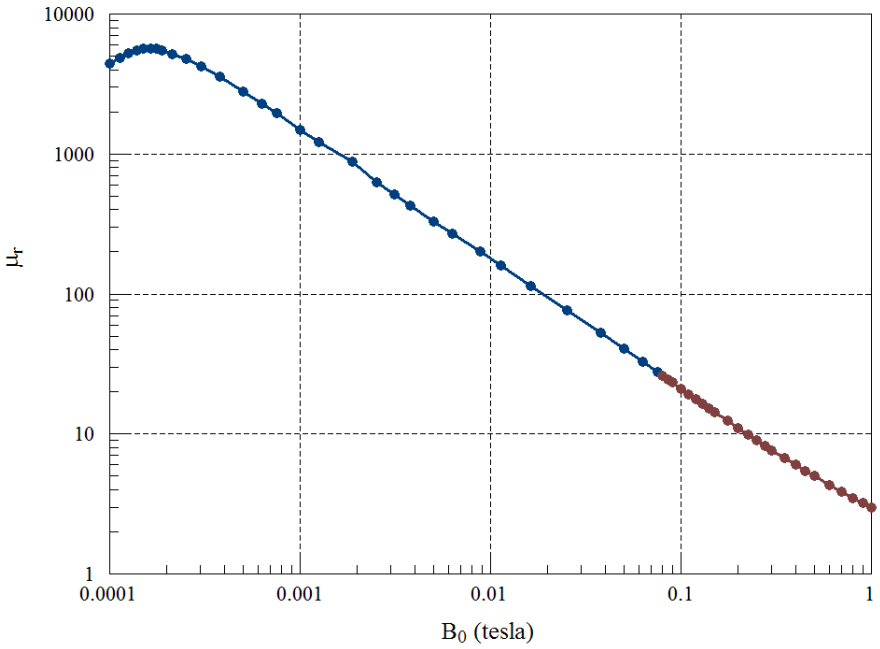


Figure 26: Extension of the saturation curve for 50H470 steel. Blue points are experimental values while red points represent the analytic extrapolation.

- Set up cell formulas to apply Eq. 12 to the last few values. If the curve extends to the saturation region, the results should approach a consistent value of B_s.
- Add addition values of B_0 to the sheet extending to the desired range. Fill out the μ_r entries using the value of B_s and Eq. 11 as a cell formula.

Figure 26 illustrates an application. The blue symbols are experimental values for 50H470 steel. As we have discussed, the variation of relative permeability in terms of B_0 is relatively simple. Estimates of the saturation flux for the final four entries converge to $B_s = 2.00$ tesla. The red curve in Fig. 26 shows points on the analytic continuation.

5.3 Control commands for nonlinear materials

Use the following alternate form of the *Mu* command to define an isotropic material with nonlinear magnetic permeability.

MU RegNo TABLE TabName
MU(4,TABLE) = INDFERRITE.DAT
The magnetic permeability for the elements of region *RegNo* are computed from a table of values stored in the file `TABNAME`. The data file must be available in the working directory.

The key to flexibility in modeling nonlinear materials is program input through *tabular functions*. A permeability table is a text file consisting of up to 256 data lines that contains values of $|\mathbf{B}_0|$ and the corresponding relative magnetic permeability μ_r. You can prepare tabular function files with a text editor or spreadsheet. You can also use published data or digitized

Table 10: Excerpts from the file SOFTIRON.MAT

```
* Generic soft iron table
* Adapted from Superfish/Poisson Code Group
* Los Alamos National Laboratory
*    B0        MuR
* ===================
  0.00000   4075.4511
  0.00024   3768.2831
  0.00038   3166.8099
  0.00059   2380.6469
  0.00094   1595.4230
  0.00155   997.8273
  ...
  1.45189   2.4550
  2.11397   2.0238
  2.72564   1.7660
  3.55229   1.6061
  4.32817   1.4830
  5.23745   1.4298
ENDFILE
```

experimental traces. Table 10 shows a portion of the tabular function SOFTIRON.DAT supplied with the **Magnum** package. Note that the file syntax conforms to the standard script rules. The free-form parser accepts real numbers in any format with a choice of delimiters. You can add documenting comment lines starting with an asterisk. The end of the data is marked with the *EndFile* command. You can add text annotations in any format after *EndFile*.

Note that the intervals between B_0 values need not be uniform. The maximum number of tables for all regions is 32. A table requires a minimum of 5 entries. You must ensure that the tabular functions extend over the full range of B_0 that will be encountered in the solution.

INTERP [Linear, Spline]
INTERP = Linear

In the default mode, **Magnum** uses cubic splines for interpolation of the permeability tables. This method minimizes recalculation time and gives smooth interpolations that aid convergence. It is important to note that the quality of the interpolation depends on the nature of the numerical data. The cubic spline routines will faithfully match values at the entry points and preserve continuity of the first and second derivatives. If the data are discontinuous, the cubic spline values may be quite inaccurate between points. You can check the fidelity of the interpolations by inspecting the GLS listing file. After recording the sorted table, **Magnum** writes a sample set of interpolated values. Note that solutions may fail to converge if the table has discontinuities of value or slope.

A numerical solution with nonlinear materials presents a challenge: 1) the spatial variation of μ_r depends on the field distribution and 2) the field distribution depends on values of μ_r. A direct solution is not feasible. Instead, **Magnum** uses the following iterative approach:

- Read the mesh file and compute a file of element matrices. Keep the file open for recomputations of the coupling coefficients in the node equations.
- Compute the coupling coefficients and perform an initial solution, evaluating the magnetic permeability in nonlinear materials using values of B_0 computed from either 1) the free-space field resulting from drive currents or 2) the value of H from the reduced potential and applied field values of a solution loaded with the *InitVal* command (see Sect. 5.4).
- Solve for the reduced potential using the standard iterative matrix procedure.
- Modify μ_r in the elements of non-linear materials using the new values of $\mu_0 H$. Regenerate the node equations and return to step 3. Repeat $NCycle$ times.
- Use the final values of reduced potential and $\mu_r(x, y, z)$ for a single computation of the dual potential inside ferromagnetic materials.
- Record the nodal values of ϕ, H_x, H_y, H_z and ψ and the element values of μ_r in a file for analysis with **MagView**.

The process generally converges to a unique solution if the variation $\mu_r(|\mathbf{B}_0|)$ is smooth and simple. Here are some pointers for constructing successful solutions:

- The nonlinear calculation may have multiple solutions if $\mu_r(|\mathbf{B}_0|)$ has a complex variation. Fortunately, realistic materials generally have simple curves as in Fig. 25. Try to avoid unnecessary details – remember that the field solution in a material with $\mu_r = 10000$ is almost indistinguishable from a material with $\mu_r = 500$.
- The run time for solutions with nonlinear materials is much longer. There is no reason to perform such a solution if materials are below saturation and μ_r has a high value everywhere.

AVG Alpha NCycle
AVG = (0.60, 10)

This command controls the nonlinear solution process. The integer number $NCycle$ is the number of correction cycles for the magnetic permeability. The run time will be approximately $NCycle$ times that for an equivalent linear solution. The default is $NCycle = 7$. The parameter α determines how material values are averaged between cycles. Averaging is performed on the quantity γ defined in Eq. 9:

$$\gamma' = \alpha\gamma(|\mathbf{B}_0^{new}|) + (1 - \alpha)\gamma(|\mathbf{B}_0^{old}|). \tag{13}$$

The quantity α should have a value equal to or less than unity. Pick a lower value if the solution does not converge. The default is $\alpha = 1.0$.

5.4 Control commands for run restart

Because of the computational demands of nonlinear solutions, we have added restart capabilities to **Magnum**. You can use a previously-computed solution to set initial values or to improve the accuracy. The feature is controlled by the following command;

INITVAL FPrefix
INITVAL = DemoCalc
The string parameter is the prefix of a **Magnum** solution file with a name of the form `FPREFIX.GOU`, available in the working directory. Both the previous and present solution must use the same mesh. In response to the command, **Magnum** opens the previous solution, checks the consistency of the mesh and reads node values. The program sets initial values of the reduced potential ϕ and dual potential ψ from values in the previous solution. If there is no *Source* command for the present solution, **Magnum** also sets values for the applied magnetic field intensity $\mathbf{H}_s$.

Here are some suggested uses for the *InitVal* command.

- Checking that current elements do not penetrate iron regions and calculating the applied field are time consuming operations if the current element file has a large number of entries. You can use to precomputed applied field solution (either in *FreeSpace* or *Standard* mode) to avoid repeating these operations.
- You can investigate the effect of small changes of material properties or applied fields.
- You can use a linear solution as an optimized starting point for a nonlinear solution.
- You can improve the accuracy of any solution in *Standard* mode.

5.5 Benchmark example

The example library contains a set of files (prefix `MAGNUMNL`) to illustrate a nonlinear calculation with restart. Figure 27 shows the geometry of the laboratory magnet defined by the file `MAGNUMNL.MIN`. The system has cylindrical symmetry to allow a comparison with results from the **PerMag** code. A field is created in an air gap by a set of coils and an iron core. The gap has radius 2.0 cm and height $L_g = 0.5$ cm. The outer radius of the core is 6.0 cm. The file `MAGNUMNL.CDF` defines nine drive coils with a total current $NI = 20,000$ A-turn, high enough to drive the inner portion of the core well into saturation. The file `TESTMUBO.DAT` defines $\mu_r(|\mathbf{B}_0|)$ for the core material.

To begin, use **MetaMesh** and **MagWinder** to create the required input files. The first **Magnum** input file (`MAGNUMNLO1.GIN`) controls a calculation with linear material. The core has the fixed value of relative magnetic permeability $\mu_r = 2000$. From Ampere's law, the predicted gap field for an ideal circuit is

$$B_g \cong \frac{\mu_0 NI}{L_g}. \tag{14}$$

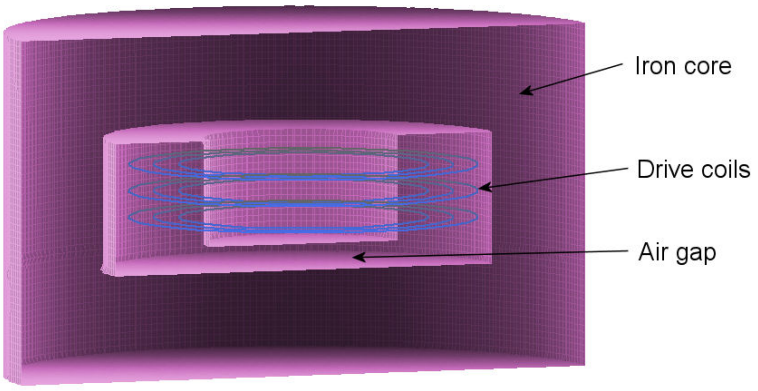


Figure 27: Geometry for example MAGNUMNL, cutaway view.

The calculated value is $B_g = 5.040$ tesla. **Magnum** gives a value at the center of the gap ($z = 0.25$ cm, $r = 0.0$ cm) of 4.983 tesla, while **PerMag** yields 4.981 tesla.

For the nonlinear calculation, **PerMag** works from the curve $\mu_r(|\mathbf{B}|)$. To achieve good convergence, the program requires a large number of adjustment cycles ($NCycle = 10000$) and a small averaging factor ($G_{av} = 0.01$). **PerMag** performs the matrix relaxation solution and material averaging simultaneously, while **Magnum** alternates between the two. Table 11 shows the script MAGNUMNL02.GIN. The program loads the same mesh used for the linear calculation and retrieves values from MAGNUMNL01.GOU. A *Source* command does not appear, so **Magnum** uses applied field values from the initial solution. The averaging factor used in **Magnum**, $G_{av} = 0.6$, is much higher than the value used in **PerMag**. The *Avg* command calls for 5 cycles of material adjustment. After each cycle, the program makes entries like the following in the listing file:

```
Nonlinear calculations: adjustment of permeability values
   Permeability adjustment cycle:      3
   Cycles in iterative matrix solution:    150
   Final relative residual:   4.9540E-07
   Standard deviation of permeability adjustment factors:   2.8971E-03
```

The final listed quantity equals

$$\sqrt{\frac{\sum_i \Delta\gamma_i{}^2}{N_i}}. \tag{15}$$

where the sum is taken over elements of nonlinear materials and $\Delta\gamma_i$ is the change in element i. The value indicates the relative error in material properties and should decrease with the number of adjustment cycles. In solutions with strong saturation, the error may initially increases as large changes are applied to the material.

The remaining **Magnum** scripts (MAGNUMNL03.GIN, MAGNUMNL04.GIN) start from the previous solution and add additional cycles of material adjustment. Figure 28 shows a scan of B_z as a function of r within the air gap at $z = 0.25$ cm. The red line shows the final **PerMag** result. The strong saturation effects reduce the peak gap field from 5.040 tesla to 2.570 tesla. The blue lines represent **Magnum** solutions with different values of $NCycle$. The values shift away from the initial linear solution with increasing material adjustments and closely approach the

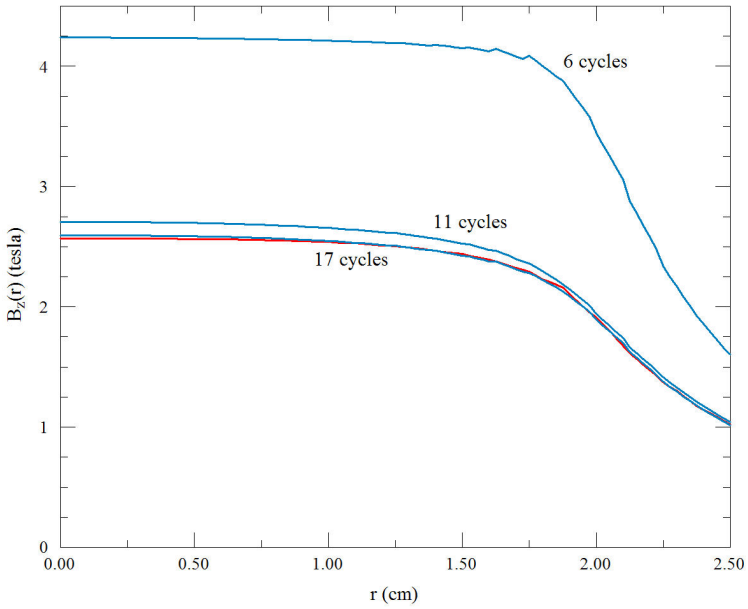


Figure 28: Radial scans of B_z at $z = 0.25$ cm. Red line: **PerMag** solution. Blue lines: **Magnum** solutions with different values of $NCycle$.

PerMag results at $NCycle = 17$. The peak value computed by **Magnum** is $B_z = 2.595$ tesla. The **PerMag** and **Magnum** predictions differ by less than 1%. The agreement is striking considering that the two solutions use different mesh types and different functional forms for the material permeability. Finally, Fig. 29 shows the spatial distribution of μ_r within the core. The figure shows the upper half of the core in the slice plane $y = 0.0$ cm.

5.6 Boundary command in nonlinear calculations

The *Boundary* command, discussed in Sect. 4.4, may be used in calculations with nonlinear materials. To review, the purpose of the command is to make a detailed model of a region within a macroscopic solution. Generally, structures are represented with coarse resolution in the macroscopic solution and fine resolution in the microscopic solution. **Magnum** sets fixed

Table 11: Contents of the file `MAGNUMNL.GIN`

```
SolType = STANDARD
Mesh = MAGNUMNL
MaxCycle = 150
ResTarget = 2.0E-7
InitVal = MAGNUMNL01
DUnit =   1.0000E+02
Mu(1) = 1.0
Mu(2,TABLE) = TESTMUB0.DAT
Avg = (0.6, 5)
EndFile
```

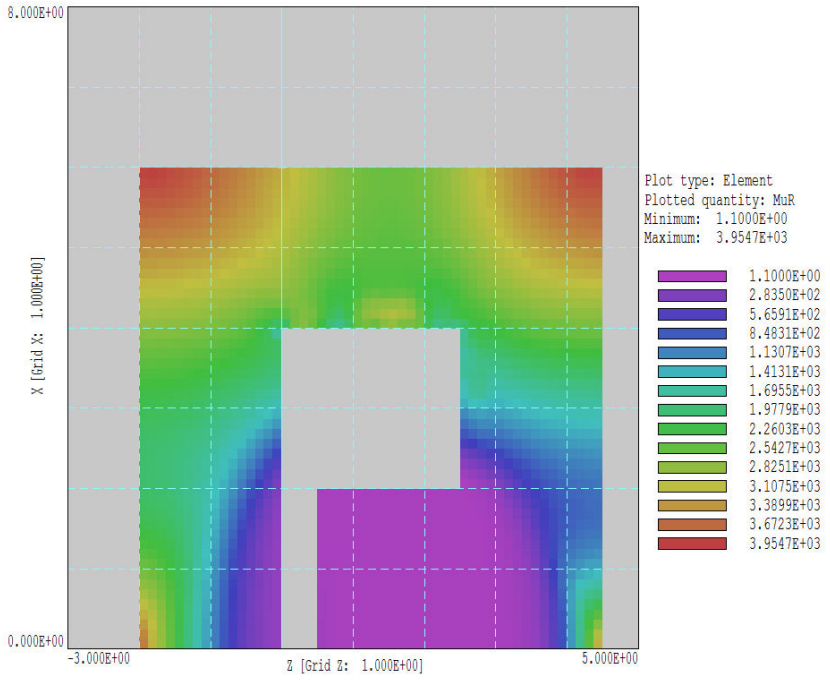


Figure 29: Spatial distribution of μ_r for the final **Magnum** solution in the plane $y = 0.0$.

values of magnetic potentials on the boundary of the microscopic solution determined from interpolations within the macroscopic solution.

We can best illustrate techniques with an example. The **MagWinder** input file `BDemoNL.CDF` defines two circular coils of radius 3.5 cm in plane normal to the z axis at $z = \pm 3.0$ cm. Each coil carries 75 kA. Two iron disks of radius 3.00 cm and height 3.00 cm are located at $z = \pm 3.0$ cm. To make the problem three-dimensional, we add a pole extension of thickness 0.2 cm as shown in Fig. 30. The extension in the top and bottom poles cover the same azimuth so that the plane $z = 0.0$ cm is a symmetry boundary (*i.e.*, the magnetic flux density is normal to the plane). In other words, the magnetic field vector is normal to the boundary. The large coil currents ensure that the iron poles are highly saturated.

The macroscopic solution (`BDemoNLMacro.MIN`) extends over the range -10.0 cm $\leq x, y \leq$ +10.0 cm, 0.0 cm $\leq z \leq$ 10.0 cm. The air space on the outside is included to approximate free-space fields. The approximate element size near the pole extension is 0.25 cm in x and y and 0.20 cm in z. Therefore, the thickness of the extension equals one element. The large element size combined with coarse resolution in the surrounding air volume, ensures a relatively quick calculation of the nonlinear problem.

The macroscopic solution is performed in two parts. The first part, controlled by the **Magnum** input file `BDemoNL01.GIN`, is a preliminary solution to find good starting values for the magnetic potentials. Here, the iron is treated as a linear material with a low value of magnetic permeability, $\mu_r = 5.0$. The file `BDemoNL02.GIN` controls the nonlinear, macroscopic solution. The script includes the commands:

```
Avg = 0.5 18
Initval = BDemoNL01
```

It is not necessary to include a *Source* command. Values for the calculated applied field are

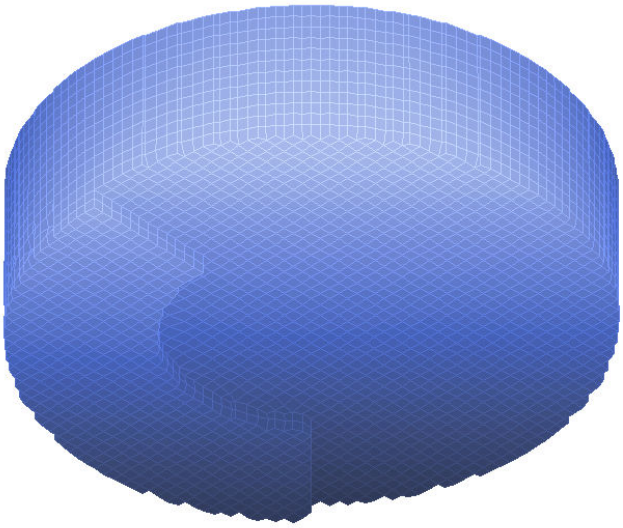


Figure 30: Extension of the iron poles in the example `BDemoNL` for the mesh of the microscopic solution (element size approximately 0.125 cm).

imported from the first solution in response to the *InitVal* command. The material `1006Steel` has a saturation magnetic flux density of $B_s = 2.19$ tesla.

The microscopic solution volume defined by `BDemoNLMicro.MIN` covers a reduced range (-5.0 cm $\leq x, y \leq$ +5.0 cm, 0.0 cm $\leq z \leq$ 3.0 cm) with an element size of 0.125 cm. Note that it is permissible for the boundary to pass through the nonlinear material. The solution is also divided into parts controlled by two scripts:

- `BDemoNL03` is a linear solution with $\mu_r = 5.0$ used to calculate initial values that speed convergence of the nonlinear solution. The script includes the following command:

  ```
  Boundary = BDemoNL02 1.0 111101
  ```

  Values on the boundary are set by interpolation in the nonlinear macroscopic solution. The parameter `111101` specifies that fixed values should be used on all boundaries except the *ZDn* symmetry boundary, allowing a high-accuracy calculation of fields on the midplane.

- `BDemoNL04` is the nonlinear calculation using boundary values from `BDemoNL02` and initial values from `BDemoNL03`.

Combining nonlinear solutions with the boundary technique requires a fairly involved solution sequence. The effort is worthwhile in calculations that require high-accuracy field values near small structures. Figure 31 shows a comparison of the field magnitude in the plane $z = 1.1$ cm for the macroscopic and microscopic solutions. The values are consistent. Clearly, the microscopic solution contains enhanced local information. Finally, Fig. 32 shows iron saturation profiles in the plane $y = 0.0$ cm for both solutions.

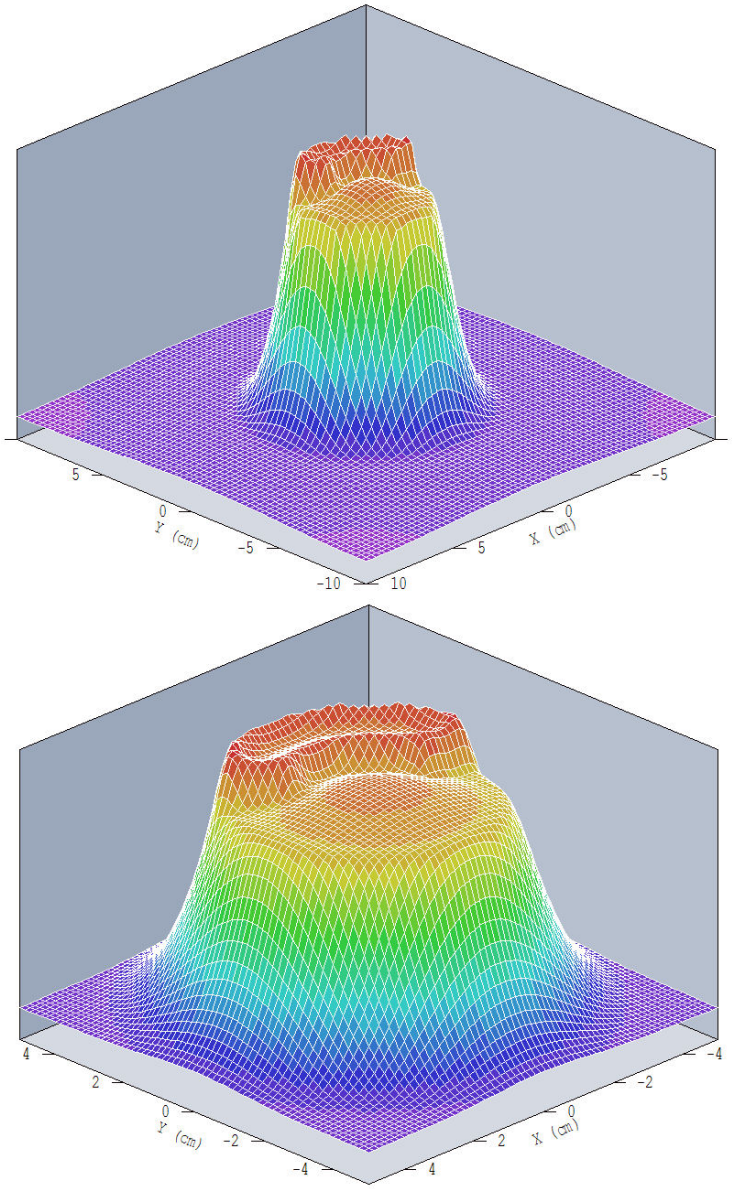


Figure 31: Plot of $|\mathbf{B}|$ in the plane $z = 1.1cm$. Top: macroscopic solution. Bottom: microscopic solution. The peak field value is 1.9 tesla.

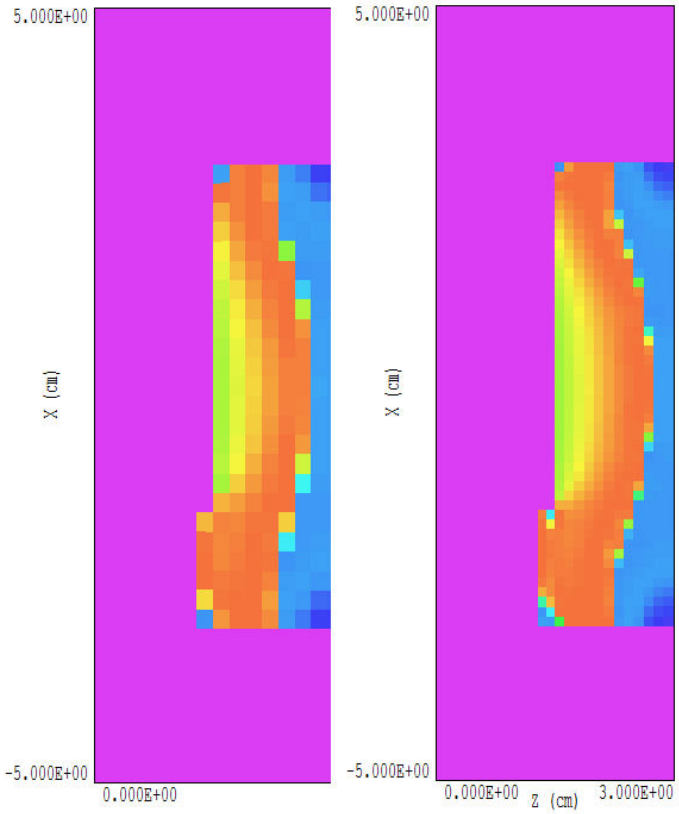


Figure 32: Plot of μ_r in the plane $y = 0.0$. Left: portion of the macroscopic solution. Right: full extent of the microscopic solution. Violet: $\mu_r = 1.0$, red: $\mu_r = 6.0$

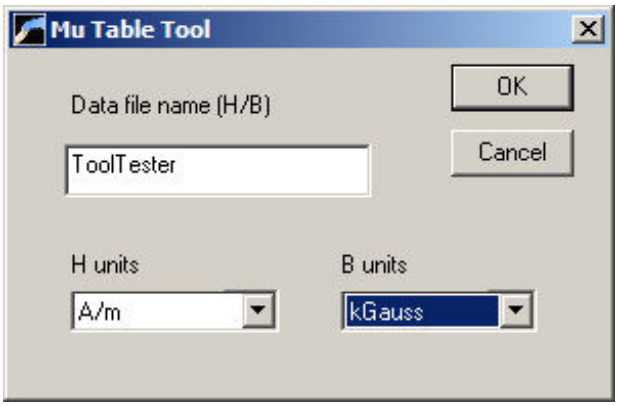


Figure 33: *Mu table tool* dialog

5.7 Mu table tool

The *Help* menu of **Magnum** contains a useful utility for converting hysteresis curves for magnetic materials (B versus H) to the standard form used in **Magnum**, B_0 versus μ_r. In the string field, enter the full name of the data file that contains the hysteresis curve. The file may contain up to 256 data lines. Each line contains a value of H followed by the corresponding value of B. Comment lines beginning with an asterisk are allowed. The file must terminate with the *EndFile* command. In the menu fields, pick units for H and B, then click *OK*. **Magnum** creates a file with the name `FPREFIX.OUT`, where *FPREFIX* is the prefix of the input file. The first section of the file is a listing of B_0 and B in units of tesla in the form of comment lines. The data section of the file consists of lines containing B_0 and $\mu_r = B/B_0$.

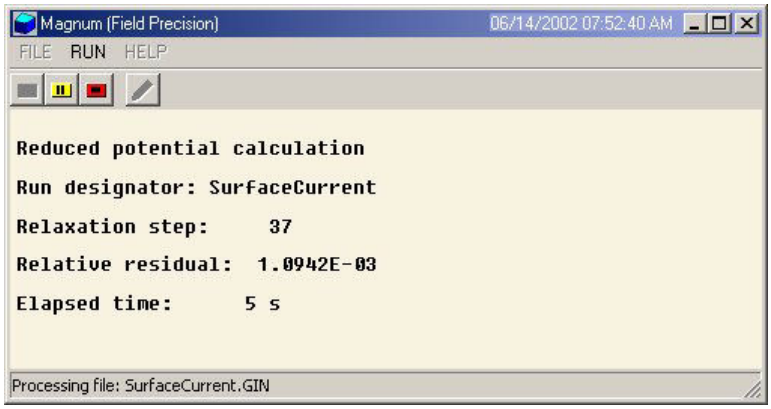


Figure 34: **Magnum** display during a solution.

6 Running the Magnum program

6.1 Interactive operation

The program `magnum.exe` can run interactively in a window. In this mode you can perform several solutions in a session and temporarily leave the program to work on other tasks. You can launch the program from **AMaze** or create you own shortcuts. Figure 34 shows the program window.

The program menu has four main commands: *File*, *Setup*, *Run* and *Help*. The following commands appear in the *File* popup menu.

EDIT SCRIPT

This command opens the internal editor to inspect or to modify text Magnum input scripts (`FPREFIX.GIN`), shown in Fig. 35. The menu references available commands. Click on an entry to add a symbolic form of a command at the cursor position. Choosing a file from an alternate directory does not change the working directory of the program. Note that the main program will be inactive until you exit the editor.

EDIT LISTING FILE
EDIT FILE

The commands open the internal editor. With the *Edit listing file* command you can pick files with names of the form `FPREFIX.GLS`. The *Edit file* command shows all available files. Choosing a file from an alternate directory does not change the working directory of the program. Note that the main program will be inactive until you exit the editor.

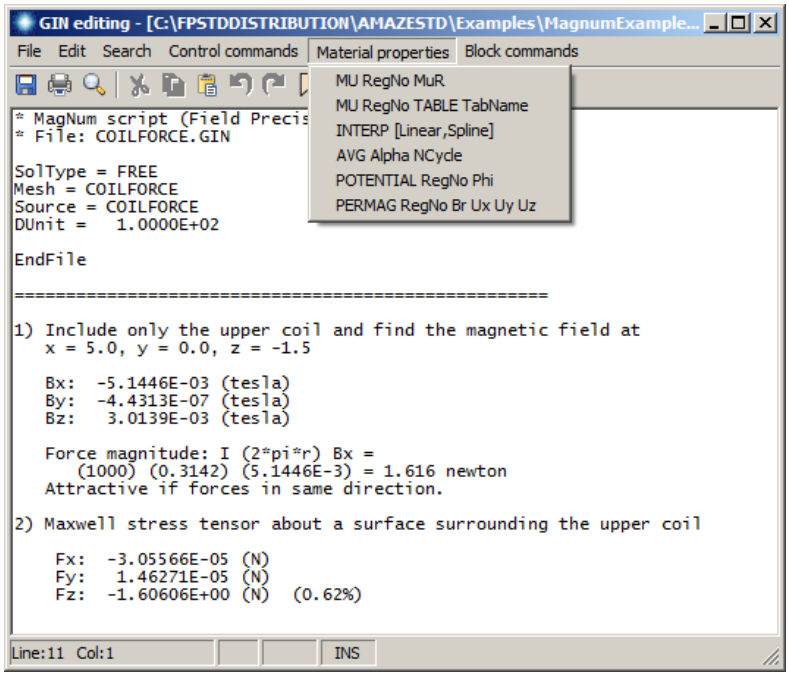


Figure 35: Editing **Magnum** input files.

SETUP

In response to this command, **Magnum** prompts for a **MetaMesh** file and brings up the dialog of Fig. 4. Fill in values for control parameters and material properties to create a basic script. You can use an editor to modify the script or to add the advanced functions described in Chap. 4.

The *Run* menu has three commands.

START RUN

Pick an input file with a name of the form `FPREFIX.GIN` to start a solution. The working directory changes if you pick a file from an alternate directory. The run begins if the requested **MetaMesh** file `MPREFIX.MDF` is available. **Magnum** displays information in the window to show the progress of program operations (Fig. 34).

PAUSE RUN

The intensive calculations of **Magnum** make demands on the resources of your computer, possibly causing other tasks to run slowly. If you need to perform critical work, you can pause the solution program during the relaxation process and restart it later without loss of data. Note that if you have a dual-processor machine, the Windows operating system will shunt tasks so that you have approximately the power of one processor when **Magnum** is running. To use the full power of the machine, you can launch two instances of **Magnum** to perform two independent calculations.

STOP RUN

This command terminates the program during the relaxation process and saves output data. For example, you may want to stop a run at a moderate value of convergence to check whether the problem has been correctly defined.

The *Help* menu has a single command.

MAGNUM MANUAL

The command displays this manual using your default PDF viewer. The file `Magnum.pdf` must be in the same directory as `Magnum.exe`.

6.2 Automatic runs under batch file control

Batch file control is a useful option for running large technical programs like **Magnum**. You can prepare scripts to organize complex operations. The sequenced programs run automatically in the background. This feature is particularly attractive on dual-processor machines.

To run a single **Magnum** calculation in the background, go to the command prompt from Windows and log to the data directory that contains the required `MDF` and `GIN` files. For example, suppose the data files `SWITCH.MDF` and `SWITCH.GIN` are in `\AMAZE\BUFFER` and that the program `magnum.exe` is in the directory `\AMAZE`. From `\AMAZE\BUFFER` type

```
..\Magnum SWITCH <Enter>
```

The program runs silently, writing detailed information in the listing file `SPECT.GLS`. If the solution is successful, the program creates the output file `SPECT.GOU` in the data directory.

The main function of the command mode is autonomous operation under batch file control. As an example, suppose you have prepared the input files `SPECT01.MIN`,...,`SPECT08.MIN` and `SPECT01.HIN`,...,`SPECT08.HIN` in the directory `\AMAZE\BUFFER`. Next you create the following batch file `SWRUN.BAT` in the data directory using a text editor:

```
@ECHO OFF
ECHO Main SPECT data run
START /WAIT ..\METAMESH.EXE SPECT01
START /WAIT ..\MAGNUM.EXE SPECT01
START /WAIT ..\METAMESH.EXE SPECT02
START /WAIT ..\MAGNUM.EXE SPECT02
 ...
START ..\METAMESH.EXE SPECT08
START ..\MAGNUM.EXE SPECT08
```

Type

```
SWRUN <Enter>
```

to generate all solutions without the need for further keyboard input.

7 MagView – file operations and plane plots

The function of the **MagView** post-processor is create plots and to calculate quantities from **Magnum** solution files. The program has the following menus: *File operations*, *Analysis*, *Plane plots*, *Slice plots*, *Surface plots* and *Help*. Initially, only the *File operations* and *Help* menus are active. You must load a data file in order to create plots or to perform analyses.

7.1 File operations

This section reviews options in the *File operations* menu.

LOAD SOLUTION FILE (GOU)

MagView displays a dialog with a list of solution files with names of the form `FPREFIX.GOU`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*. The program loads the solution and updates the status bar. If data retrieval is successful, the analysis and plot menus become active.

SOLUTION FILE INFORMATION

The command shows a message box with information on the currently-loaded data file.

LOAD CONFIGURATION FILE

Configuration files determine the plot and analysis quantities of **MagView**. The files have names of the form `FPREFIX.CFG`. The file `MAGVIEW_STANDARD` is supplied with the program. **MagView** loads this configuration the first time you run the program. Use this command to change to a different configuration. The program remembers the last configuration used and reloads it. You can customize **MagView** by creating your own configuration file (see Chap. 12).

LOAD COIL FILE (WND)

Load a coil file to calculate forces on applied field coils and to include them surface-type plots. The *Load coil file* command displays a dialog with a list of current element files with names of the form `FPREFIX.WND`. Changing the directory in the dialog changes the program working directory. Pick an available file and click *OK*.

CLOSE COIL FILE

This command removes applied field coils from slice and surface plots

COIL FILE INFORMATION

The command displays information on the currently-loaded current element file.

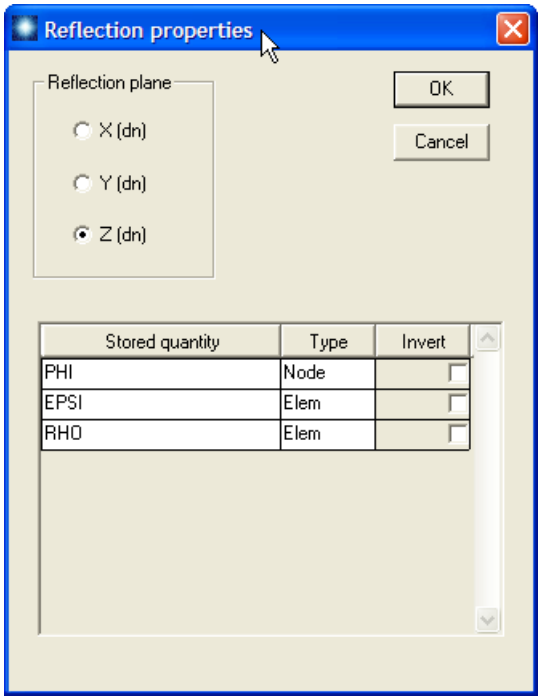


Figure 36: Dialog to create a reflected solution

SAVE REFLECTED SOLUTION

The use of a symmetry boundary saves considerable time in a large 3D calculation but produces half a solution. There are instances where it is useful to have numerical values for the full solution volume (*e.g.*, creating plots or tracking orbits in **OmniTrak** or **GamBet**). Use this command to create a data file for the currently-loaded solution that includes values of electrostatic potential reflected across a symmetry boundary. To begin, you must set up the original solution so that the symmetry plane corresponds to the lower solution-volume boundary in either x, y or z. Load the solution, click on the command and use the *Save* dialog to specify an output file name. The next dialog (Fig. 36) shows a set of options for the reflection. Designate a reflection plane with the radio buttons. The grid field shows the stored quantities in the solution file. The *Invert* check box determines the reflection symmetry. Table 12 shows inversion settings for the two types of magnetic-field symmetry. You must ensure that the solution is appropriate for the reflection parameters that you choose. **MagView** does not check the physical consistency of the output values. You can load the modified file and check validity with the plot functions of **MagView**.

RUN SCRIPT

Sometimes you may want to perform complex or repetitive analyses on a set of similar solutions. Analysis script operation is a powerful feature of **MagView**. This command displays a dialog with a list of analysis scripts (suffix `SCR`) that you have created. Pick a file and click *OK*. Changing directories in the dialog changes the working directory of the program. The analysis script can load data files, open and close history files, and perform any of the numerical functions described in this manual. Section 8.2 reviews the analysis script language.

Table 12: Inversion settings for reflection about $z = 0.0$

Stored quantity	**Invert**
Mirror symmetry ($B_\perp = 0.0$)	
H_x	
H_y	
H_z	
ϕ	X
ψ	X
Cusp symmetry ($B_\parallel = 0.0$)	
H_x	
H_y	
H_z	X
ϕ	
ψ	

CREATE SCRIPT

Use this command to create scripts using the internal editor. A box requests a file prefix. The resulting script file will be saved as `FPREFIX.SCR`. Next, the program opens the file in the editor and writes the reference list of allowed commands shown in Table 13. Enter commands in the space above *EndFile*. After saving the file, you can run it using the *Run script* command.

EDIT SCRIPT

Use this command to change an existing script file. The dialog lists files in the current directory with the subscript `SCR`. Changing directories does not change the working directory of the program.

OPEN DATA FILE

Several of the analysis commands like *Point calculation* and *Line scan* generate quantitative information. You can automatically record the data generated during an analysis session by opening a data file. Supply a file prefix in the dialog or accept the default. The text data file has a name of the form `FPREFIX.DAT` and will be stored in the working directory. You can use an editor to view the file or to extract information for mathematical analysis programs or spreadsheets.

CLOSE DATA FILE

Use this command if you want to start a new file to record data. The data file is automatically closed when you exit the postprocessor. Otherwise, you must close the file before using the *Edit data file* command or loading the file into another program. Failure to close the file may result in a Windows Resource Sharing Error.

EDIT DATA FILE

View or modify files with names of the form `FPREFIX.DAT`.

Table 13: Create script - default file content

```
* Analysis script
* Insert commands here...
ENDFILE


      --- Command summary ----

INPUT InFileName
 [Close current solution file and load FileName]
OUTPUT OutFileName
 [Close current data file and open OutFileName]
CONFIGURATION ConfigFileName
 [Load a new configuration, ConfigFileName]
POINT xp yp zp
 [Point field calculation at the given coordinates]
SCAN xp1 yp1 zp1 xp2 yp2 zp2
 [Scan along a line between the given coordinates]
GENSCAN
   xp1 yp1 zp1
   ...
   xpn ypn zpn
END
 [Write field values at listed coordinates]
VOLUMEINT [RegNo]
 [Volume integrals, optionally for region RegNo]
SURFACEINT Reg1 Reg2 ... RegN
 [Surface integrals around the region set]
MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ
 [Write a matrix of field values to the file FileName]
NSCAN  100
 [Set the number of points in a line scan]
INTERPOLATION [LSQ,Linear]
 [Set the interpolation method]
ENDFILE
 [Terminate the analysis]
```

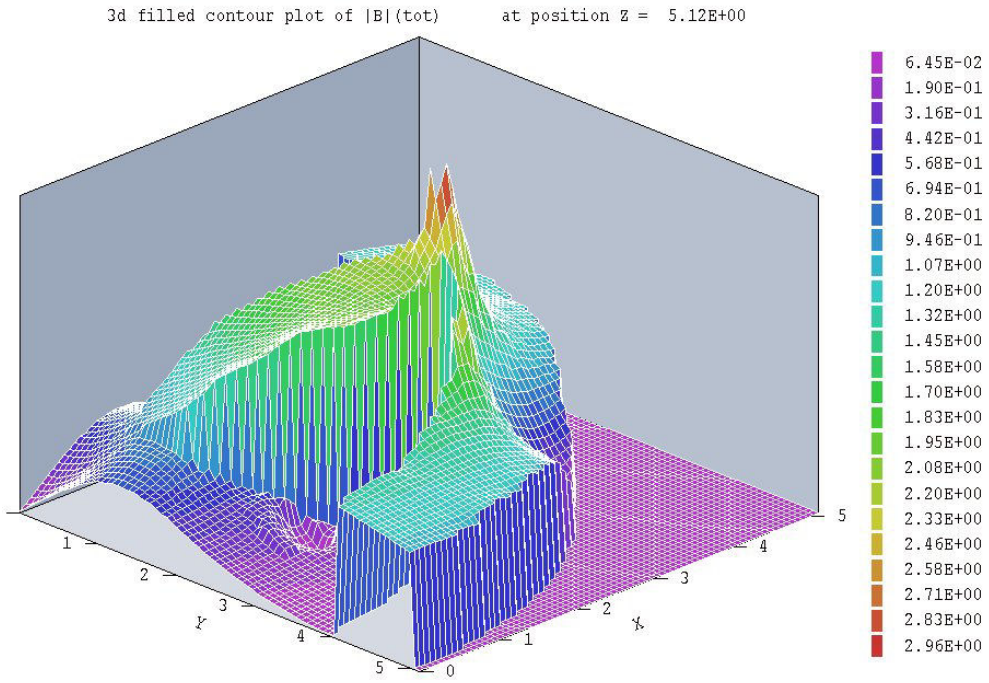


Figure 37: Plane plot of $|\mathbf{B}|$ – type *3D filled contour*.

EDIT FILE

Use the program editor to view or to modify any text file.

The *Help* menu shows program information and contains the following command:

INSTRUCTION MANUAL

Displays this document in your default PDF viewer. The file `magnum.pdf` must be in the same directory as `magView.exe`.

7.2 Plane plots

Plane plots (Fig. 37) are two-dimensional plots that show the variation of quantities over planes thAT cut the solution volume. There are three differences from the slice plots discussed in the next section:

- Plane plots are created from a rectangular mesh of values while slice plots are built using the computational mesh. Slice plots show precise region boundaries and equipotential lines in true scale.
- Because they are simpler to construct, plane plots support a wider variety of plot styles.
- Plane plots are not limited to slice normal to a Cartesian axis.

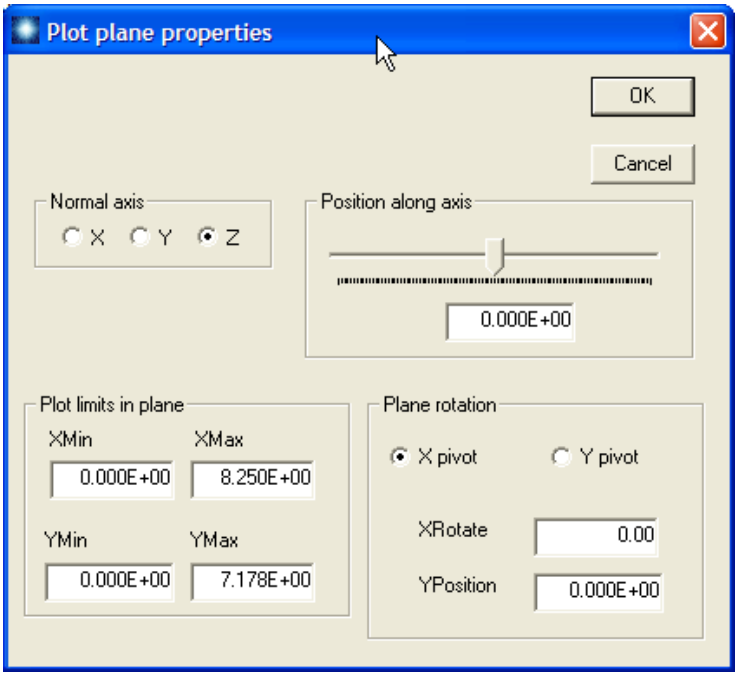


Figure 38: Set plane dialog.

SET PLANE

This command brings up the dialog of Fig. 38 to set the plane for the plot. Set the normal axis with the radio buttons at the top. For example, for a choice of z the plot will be created in the x-y plane. You can use the slider bar to set the position along the normal axis or type a value in the box. The range of the slider bar is automatically set to the limits of the solution volume along the normal axis. The field group at the bottom-left determines the plot range within the normal plane. The default settings are the limits of the solution volume. Note that plane plots are constructed to fill the maximum area. They do not preserve scaling in the normal plane. The final command group at the bottom-right controls an optional rotation of the plot plane. The radio buttons set the pivot axis, the quantity *Rotate* is the rotation angle in degrees and *Position* is the pivot position along the direction normal to the pivot axis.

The following commands are in the *Plot control* popup menu:

PLOT STYLE

This command brings up the dialog of Fig. 39 to set the plot style. As an example, Fig. 37 shows the *Filled contours 3D* plot style. The numbers at the bottom give the resolution of the mesh used to create the plot. Higher values give more detail but require longer regenerate times. The default is a 150×150 mesh.

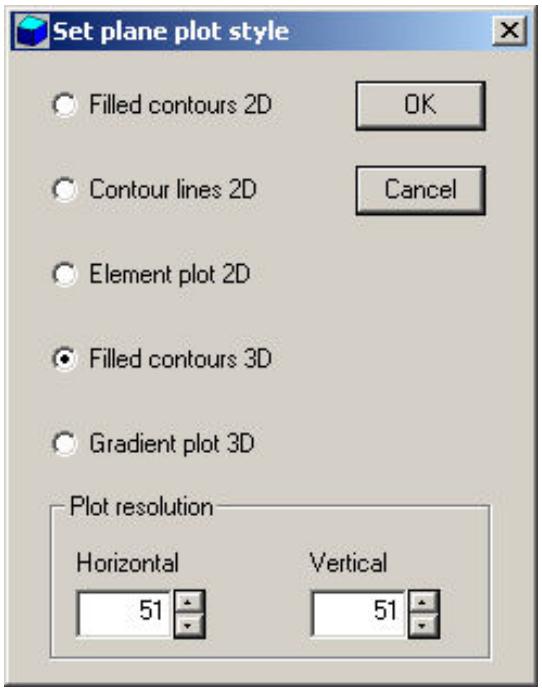


Figure 39: Plot style dialog.

PLOT QUANTITY

Set the quantity to be plotted. The choices depend on definitions in the `INTERPOLATION` section of the configuration file. The following quantities are included in `MAGVIEW_STANDARD.CFG`:

- The magnitude and components of the total magnetic flux density: $|\mathbf{B}|$, B_x, B_y and B_z in units of tesla.
- The magnitude and components of the magnetic field: $|\mathbf{H}|$, H_x, H_y and H_z in units of A/m.
- The relative magnetic permeability and its inverse: μ_r and $\gamma = 1/\mu_r$. These quantities are meaningful for solutions that include nonlinear materials (Chap.5). Plots of γ are useful to display regions of saturation.

The display of the magnetic flux density ($\mathbf{B}$) and its components depends on the setting of the interpolation mode (see Sect. 11.4) In the *Standard* mode, the quantities correspond to the total magnetic flux density calculated using reduced potential values (ϕ) in air and conductive regions and dual potential values (ψ) in iron regions. In the *Reduced* mode, the program calculates $\mathbf{B}$ from the reduced potential in all regions. Values of $\mathbf{H}$ are always calculated from the reduced potential and do not depend on the interpolation mode.

PLOT LIMITS

Set limits for the plotted quantity. When *Autoscale* is active, **MagView** automatically sets limits based on the range of values in the solution file. To plot an absolute range, uncheck the box and enter numbers for the minimum and maximum values. In either case, the program

shows lines in plots of type *Contour lines 2D* at a uniform interval. You can set values for contour lines manually. See Sect. 9.2 for information.

ROTATE PLOT

This command is active only for the *Filled contours 3D* and *Gradient plot 3D* styles. You can rotate the plot in 90° increments for the best view.

The commands of the *Export plot* menu are used to generate hardcopy or to create plot files.

DEFAULT PRINTER

With this command, a **MagView** plot may be directed to any installed Windows printer (*e.g.*, network printers, postscript drivers, PDF drivers...). Note that the current screen plot is sent to the default Windows printer. If necessary, change the default in Windows before running **MagView**.

SAVE PLOT FILE

Use this command to create a graphics file of the current plot in either Windows Bitmap (BMP) or Portable Network Graphics (PNG) formats. In the dialog, specify the format, the size in pixels and the file prefix. The graphics file is created in the current directory.

COPY TO CLIPBOARD

The current plot is copied to the clipboard in Windows Metafile format. You can then paste the image into graphics software.

7.3 Saving and loading views

The creation of plots for presentations may involve some effort. With the following two commands, you can save all the current view parameters and immediately restore the plot.

SAVE NAMED VIEW

Save the view parameters for the current plot. Quantities such as the slice axis, slice position and zoom limits are saved for two dimensional plots. Parameters for three-dimensional plots include the viewpoint position, displayed regions and cut planes. The information is stored in a text file in the current directory with a name of the form `FPREFIX.FPV`.

LOAD NAMED VIEW

Load a view file and refresh the plot. Note that you must be in the appropriate plot menu to retrieve a view. Views of plane plots must be loaded to the Plane Plot Menu.

The file contains the complete set of plot parameters. This excerpt illustrates the format:

```
Program: AMaze
2D/3D: 2D
DisplayBy: Regions
Outline: On
NSlice:     40
SliceAxis: YAxis
PlotType: LogElemUp
XPMin: -1.500000E+00
XPMax:  4.250000E+00
...
```

If a specific solution file is loaded, the plot will be restored exactly. The saved view feature in **MagView** has two useful features if a different meshes are loaded:

- Dynamic adaptation to different solutions.
- Option for user control of the view parameter set.

Regarding the first feature, there are situations where you want to create consistent views of a set of solutions with different geometries, maintaining a similar appearance. Some plot properties (like the viewpoint rotation matrix) are applicable to any solution, but others (like region cut planes or slice plot limits) depend on the geometry. **MagView** checks each plot parameter for validity. If a parameter is outside the allowed range for the currently-loaded solution, the program computes an alternative. The goal is to preserve as many features of the view as possible.

You can modify view files with an editor. The order of entries is not rigid. On input, **MagView** uses a free-form parser. If a parameter is missing, the program simply makes no change from the value current in the program. The implication is that you can modify a saved view to include only elements essential to your application. For example, you could compare a series of assemblies with different sizes, maintaining an orthographic 3D view from the same point in Cartesian space. In this case, the view file would contain only the entries:

```
DView:  1.000000E+37
R11:  8.660253E-01
R12: -5.000002E-01
R13:  0.000000E+00
R21:  1.669031E-01
R22:  2.890846E-01
R23:  9.426408E-01
R31: -4.713208E-01
R32: -8.163510E-01
R33:  3.338061E-01
```

8 MagView – numerical analysis

The commands of the *Analysis* popup menu are used to determine numerical values for the solution. In the interactive mode, the menu is active only when a data file is open.

8.1 Automatic analyses operations

POINT CALCULATION

The *Point calculation* command brings up a dialog where you enter the x, y and z coordinates of any point in the solution space. In contrast to the *Point calculation* command of the *Slice plots* menu, the point is not limited to a slice plane. When you click *OK*, **MagView** displays an information box and also writes the results to the data file in the following format:

```
   --- Point Field Analysis ---
Position: [  3.0000E-01,   4.0000E-01,  -2.5000E-02]
Region number:   1
Bx:   3.260803E-04
By:   4.385794E-04
Bz:  -3.195617E-01
BMag:   3.195622E-01
Hx:   2.594865E+02
Hy:   3.490104E+02
Hz:  -2.542992E+05
HMag:   2.542995E+05
MuR:   1.000000E+00
Gamma:   1.000000E+00
```

LINE SCAN

In the *Analysis* menu, scans may be performed along arbitrary straight lines between any two points in the solution volume. When you click the *Line scan* command a dialog appears in which you can specify the start and end points. Enter values in units set by *DUnit*. In other words, if you used dimensions of cm in **MetaMesh** and the **HiPhi** script contains the command $DUnit = 100.0$, then enter the dimensions in cm. In the interactive mode, **MagView** creates a screen plot of a selected quantity determined by the *Set scan quantity* command. Features of the scan plot environment are discussed in Sect. 9.4. The following is an example of a listing.

```
------- Field scan between points -------
   XStart:   5.0000E-02   YStart:   3.5000E-01   ZStart:  -2.5000E-02
     XEnd:   1.6500E+00     YEnd:   3.5000E-01     ZEnd:  -2.5000E-02

X            Y             Z             NReg
 5.000000E-02  3.500000E-01 -2.500000E-02     1
 6.066667E-02  3.500000E-01 -2.500000E-02     1
 7.133334E-02  3.500000E-01 -2.500000E-02     1
 8.200000E-02  3.500000E-01 -2.500000E-02     1
 9.266667E-02  3.500000E-01 -2.500000E-02     1
 1.033333E-01  3.500000E-01 -2.500000E-02     1
     ...
                  Bx            By            Bz
             3.986103E-04  2.957171E-03 -3.107407E-01
             4.918569E-04  2.886671E-03 -3.110280E-01
             5.851036E-04  2.816171E-03 -3.113153E-01
             6.284067E-04  2.754149E-03 -3.114772E-01
             7.052898E-04  2.659371E-03 -3.118566E-01
             7.404999E-04  2.589851E-03 -3.120223E-01
                 ...
```

VOLUME INTEGRALS

In response to the command, **MagView** performs volume integrals of quantities defined in the `VOLUME` section of the configuration file over all elements of the mesh. The calculations are intensive, so there may be a delay for large meshes. **MagView** records results to the data file in the following form:

```
---------- Volume Integrals ----------
   Global volume:   2.359751E-04
   RegNo   Region volume
   ========================
       1  2.176711E-04
       2  6.323549E-06
       3  6.323442E-06
       4  5.657013E-06

Quantity: MagEnergy
   Global value:   2.750644E+00
   RegNo   Region volume
   ========================
       1  6.886417E-01
       2  1.844907E-03
       3  1.840312E-03
       4  2.058317E+00
```

The first table always shows the global and region volumes. Additional tables list results for quantities defined in the `VOLUME` section of the configuration file. The file `MAGVIEW_STANDARD.CFG` displays the integral of the magnetic field energy density:

$$dU_m = \frac{B^2}{2\mu_0\mu_r}. \tag{16}$$

Note that the quantity is valid only for air regions or regions of linear magnetic material (*i.e.*, a fixed value of μ_r). You can use the energy values to determine inductance in a solution with a single coil that carries current I_0 from the equation:

$$U_m = \frac{L\ I_0^2}{2}. \tag{17}$$

In Eq. 17 the quantity U_m is the global field energy.

SURFACE INTEGRALS

In response to this command, **MagView** takes integrals of defined quantities over the selected region surfaces. The program displays the dialog of Fig. 40 which includes a row for each region along with the name assigned in **MetaMesh** and forwarded by **Magnum**. Activate the check boxes in the *Internal* column to specify the regions to include in the set. The default is to include the full exterior surface of the region set (*i.e.*, all remaining regions are *External*). Use the check boxes to specify *External* regions. For example, to take a surface integral over the boundary between Regions 2 and 5, set Region 2 as *Internal* and Region 5 as *External*. Note that **MagView** does not include facets on the boundary of the solution volume in surface integrals.

The calculated quantities are defined in the `SURFACE` section of the configuration file. The following quantities are defined in `MAGVIEW_STANDARD.CFG`:

- The magnetic flux from the *Internal* regions to the *External* regions, $\int\int\ \mathbf{B}\cdot\mathbf{n}dA$.
- Components of the body force on the *Internal* region set determined from the Maxwell stress tensor. In this case, the *Internal* regions must be surrounded by air regions and all *External* regions must be included.
- Components of the torque on the Internal region set determined from the Maxwell stress tensor. For this calculation, be sure to set values for the torque origin in the `PROGPARAM` section of the configuration file ($\$xt, \$yt, \$zt$).

MATRIX FILE

This command controls a feature that is useful if you want to write your own analysis routines or port results to mathematical software. In response to the command, **MagView** performs interpolations over a specified box region on a regular grid of values. It is much easier to use results in this form than to deal with the conformal mesh directly. The command calls up the dialog of Fig. 41. Specify the dimensions of the box along each axis (in units set by *DUnit*) and the number of calculation intervals. To illustrate, calculations are performed at positions with x coordinates given by

$$x = x_{min} + \frac{n\ (x_{max} - x_{min})}{n_x}, \tag{18}$$

where $n = 0, 1, 2, ..., n_x$. For example, if you set $x_{min} = 0.5$, $x_{max} = 1.5$ and $n_x = 10$, the calculations are performed at points with $x = 0.5$, 0.6, ..., 1.4, 1.5. The recorded quantities depend on definitions in the configuration file and the setting specified in the Recoded quantities

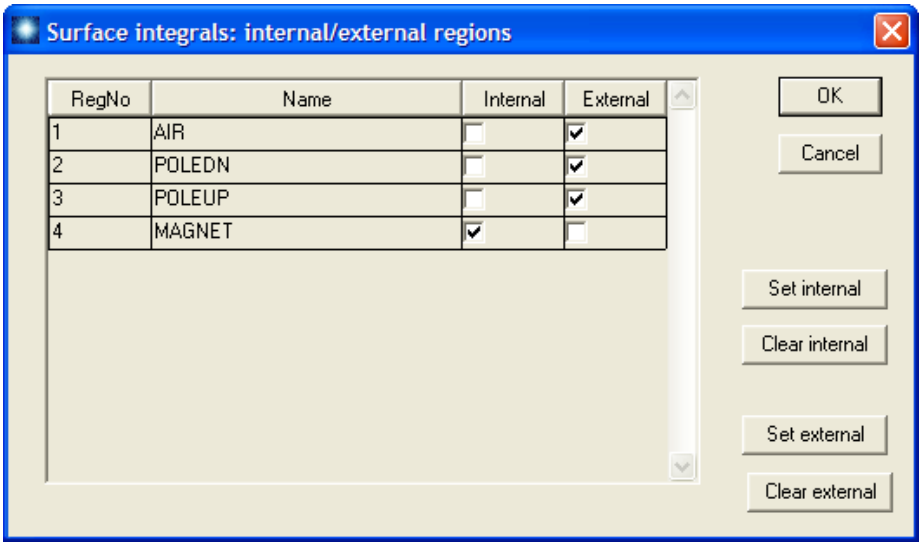


Figure 40: Dialog to pick the region set for surface integrals.

command. The interpolation routine returns zero if a portion of the box is outside the solution volume or inside an electrode. You can also specify an output file name. The format of the text file is described in Chap. 15. If you want a two-dimensional matrix of values in a single plane, set either N_x, N_y or N_z equal to zero. For example, if $N_x = 0$, then a set of calculations is performed over the specified range of y and z at the single position $x = (x_{min} + x_{max})/2.0$.

SCAN QUANTITY

With this command you can pick the quantity that will be displayed in screen and exported plots of line scans. Pick the quantity from the list box and click *OK*. The listed quantities depend on the entries in the `INTERPOLATION` section of the configuration file. This setting has no effect on the data file listing which includes all values specified by the *Recorded quantities* command.

SET NUMBER OF SCAN POINTS

This command sets the number of line scan points in the screen plot and data file listing. The default value is 150 and the maximum number is 500.

LSQ/LINEAR INTERPOLATION

Set the interpolation method for analysis and plots. The default is a second-order least-squares-fit procedure that uses data collected from nearby elements. Under the *Linear* option, only node values from the target element are included. This option is useful in tight spaces near material boundaries.

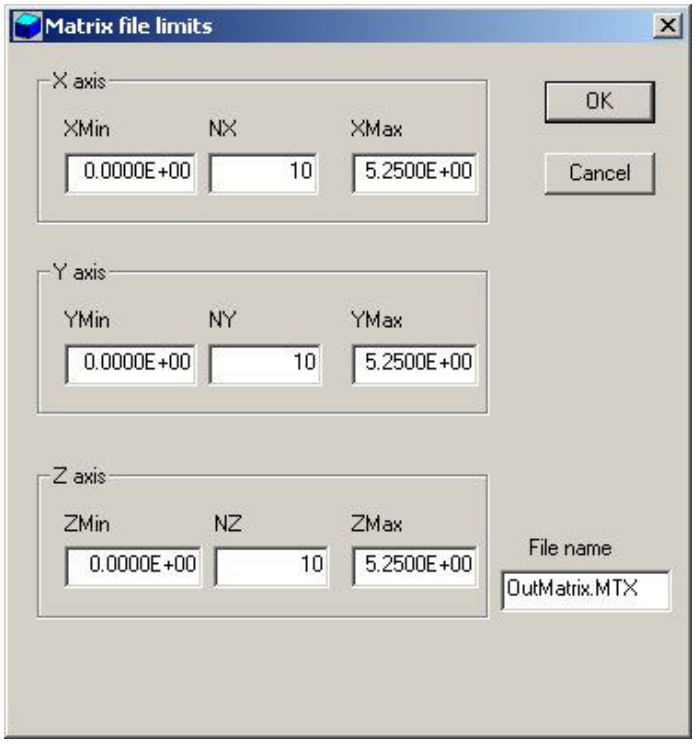


Figure 41: Create matrix file dialog.

B INTERPOLATION MODE

Choose the method for calculating the magnetic flux density. In the default *Standard* mode, the equation $\mathbf{B} = \mu(\mathbf{H}_s - \nabla\phi)$ is applied over the full solution volume. In the *DualPot* interpolation mode, **MagView** applies the equation $\mathbf{B} = \mu(\mathbf{H}_s - \nabla\phi)$ at points inside elements with $\mu_r \cong 1.0$ and uses $\mathbf{B} = -\mu\nabla\psi$ in elements with $\mu_r \gg 1.0$. This mode functions only when the *DualPot* command appears in the **Magnum** script. Otherwise, the program records zero values of ψ.

RECORDED QUANTITIES

This command determines the calculated quantities included in scan listings and matrix files. Under the option *Field quantities*, **MagView** records the first three quantities defined in the `INTERPOLATION` section of the configuration file. The quantities are B_x, B_y and B_z in the standard configuration. Use this option to export field tables to **OmniTrak**. Under the *Full set* option, **MagView** records all quantities defined in the configuration file.

8.2 MagView analysis scripts

MagView scripts are useful for automating complex or repetitive analysis procedures. For example, you may want to compare a line scan of values between two points in a set of similar solutions. In the interactive mode, it would be necessary to type in the same coordinates for each scan. An analysis script can load each solution file, perform the scan and record all information in a single data file. In the **MagView** main menu, the following commands in the *File* popup menu (Sect. 7.1) apply to analysis scripts.

RUN SCRIPT

Pick and run a script file with a name of the form `FPREFIX.SCR`. The script can operate on the currently-loaded solution file or load a different file. The last solution file loaded by the script becomes the current file for plots and interactive analysis.

CREATE SCRIPT

Use this command to create an analysis script using the internal program editor.

EDIT SCRIPT

Use this command to change an existing script with the internal program editor.

MagView may be called as a command line task. The calling syntax is

```
[progpath\]MAGVIEW [scriptpath\]SPREFIX
```

where the script has the file name `SPREFIX.SCR`. You should supply the full path `[scriptpath]` if the script is not in the current directory.

Command line operation opens several possibilities.

- **MagView** runs quickly and silently in the background when laucnhed from the *Command Prompt* in a terminal window. To minimize typing, open the terminal in the working directory. Use the DOS `CHDIR` command to change directories.
- The program may be called by DOS batch files with commands of the form

  ```
  START /WAIT [progpath]\MAGVIEW [scriptpath\]SPREFIX
  START [progpath]\MAGVIEW [scriptpath\]SPREFIX
  ```

  Use batch files to organize calculations that run autonomously. The option `/WAIT` in the first example signals that the batch file waits for completion of the program to continue. This form is useful if data from **MagView** is required for the next operation. Omit the option if you want to launch several instances of **MagView** to take advantage of a multiple-core machine. Use the `CHDIR` command to set the working directory.
- Run **MagView** as a subtask from your own compiled or interpreted programs. One application is to integrate finite element solutions and analysis into an optimization loop.

8.3 Script commands

Entries on a command line may be separated by the following set of delimiters: space, tab, equal sign [=], colon [:], left parenthesis [(] or right parenthesis [)]. Each command listed below is shown in symbolic form along with an example of how they might appear in a script.

INPUT [datapath\] FileName
INPUT = E:\FEData\AprilRuns\HMagnet.GOU
Close the currently-opened solution file and load a solution file from the current directory. If `[datapath]` is not specified, the solution file must be in the working directory.

OUTPUT [datapath\] FileName
OUTPUT: WGUIDE1.DAT
Close the current data record file and open a new one. If a `[datapath]` is not specified, the file will be written in the working directory. A data file must be opened to use the following analysis commands.

CONFIGURATION [datapath\] FileName
CONFIGURATION = \FieldP\AMaze30\PHIVIEW_FORCE.CFG
Load a new MagView configuration file to change the analysis characteristics. If a `[datapath]` is not specified, the configuration file must be available in the working directory. If this command does not appear, the program uses the current configuration or the one in force in the last run.

POINT xp yp zp
POINT = (0.00, 0.05, 4.67)
Perform a point calculation and write values of the quantities defined in the *Interpolation* section of the configuration file to the data record file. Enter coordinates in the units set by $DUnit$.

SCAN xp1 yp1 zp1 xp2 yp2 zp2
SCAN = (0.00, 0.00, 15.00) (12.00, 0.00, 15.00)
Perform $(NScan + 1)$ calculations along a line in space. At each point, write the first $NRecord$ quantities defined in the `INTERPOLATION` section of the configuration file to the data record file. Enter coordinates in units set by $DUnit$.

GENSCAN
Perform any number of calculations along an arbitrary path in space. At each point, write the first $NRecord$ quantities defined in the `INTERPOLATION` section of the configuration file to the data record file. The command must be part of a structure with the following form:

```
GENSCAN
  xp1 yp1 zp1
  xp2 yp2 zp2
  ...
  xpn ypn zpn
END
```

Each data line contains three real numbers separated by spaces to define a point in the solution volume. Enter coordinates in units set by *DUnit*.

VOLUMEINT [NReg]
VOLUMEINT
VOLUMEINT 8
Perform a volume integral of quantities defined in the `VOLUME` section of the configuration file and write the results to the data record file. If a region number does not appear, the integral is taken over the full solution volume. Otherwise, the integral extends over elements with region number $NReg$. Volume integrals may take a long time, so use the region-number option if there are several regions that are not of interest.

SURFACEINT Reg1 Reg2 ... -RegI -RegJ
SURFACEINT 5 7 -10 -11 -12
SURFACE INT 2 3
Perform a surface integral of quantities defined in the `SURFACE` section of the configuration file and write the results to the data record file. Positive integer values correspond to *Internal* regions and negative to *External* regions. With no specification, all regions that are not *Internal* are taken as *External*. The integral is taken over the surface facets between any *Internal* and *External* elements. The command may include any number of regions. The regions in the *Internal* set may or may not be contiguous. Note that **MagView** does not include facets on the boundary of the solution volume in surface integrals.

MATRIX FileName XMin XMax NX YMin YMax NY ZMin ZMax NZ
MATRIX WGUIDE.MAT 1.00 1.00 10 10.00 2.00 10 2.00 12.00 20
Write values calculated at an array of location to a data file in text format. The file is named $FileName$ and is created in the current directory. The real-number values x_{min} and x_{max} are the limits of the listing volume along the x direction. The parameter N_x is an integer. Listings are made at $(N_x + 1)$ evenly spaced positions along x.

NSCAN NScan
NSCAN = 100
Set the number of intervals for line scans. The default is $NScan = 150$ and the maximum value is $NScan = 500$.

RECORD [Field, Full]
RECORD = Full
Set the number of quantities to include in data files in response to the *Scan* and *Matrix* commands. The *Field* set (default) consists of E_x, E_y and E_z. Use this option to create data tables for **OmniTrak**. The *Full* set includes all quantities defined in the `INTERPOLATION` section of the configuration file.

INTERPOLATION [LSQ,Linear]
INTERPOLATION = Linear
Set the interpolation type to a multi-element least-squares fit or a single-element linear fit.

PLOT FSaveView FOutput Nx Ny
PLOT (XAxis3D VIEW001 800 600)
Write a plot file based on data in the currently loaded solution. The string *FSaveView* is the prefix of the file `FSaveView.FPV`, a set of view parameters created with the *Save view* command (Sect. 7.3). The view file must be available in the working directory. The plot is saved in the working directory. It is in Windows Bitmap format and has the name `FOutput.BMP`. The integers N_x and N_y give the width and height of the image in pixels. To optimize compatibility with graphics format converters, pick values that are multiples of 16 (*e.g.*, 1024×768).

ENDFILE
Terminate the analysis

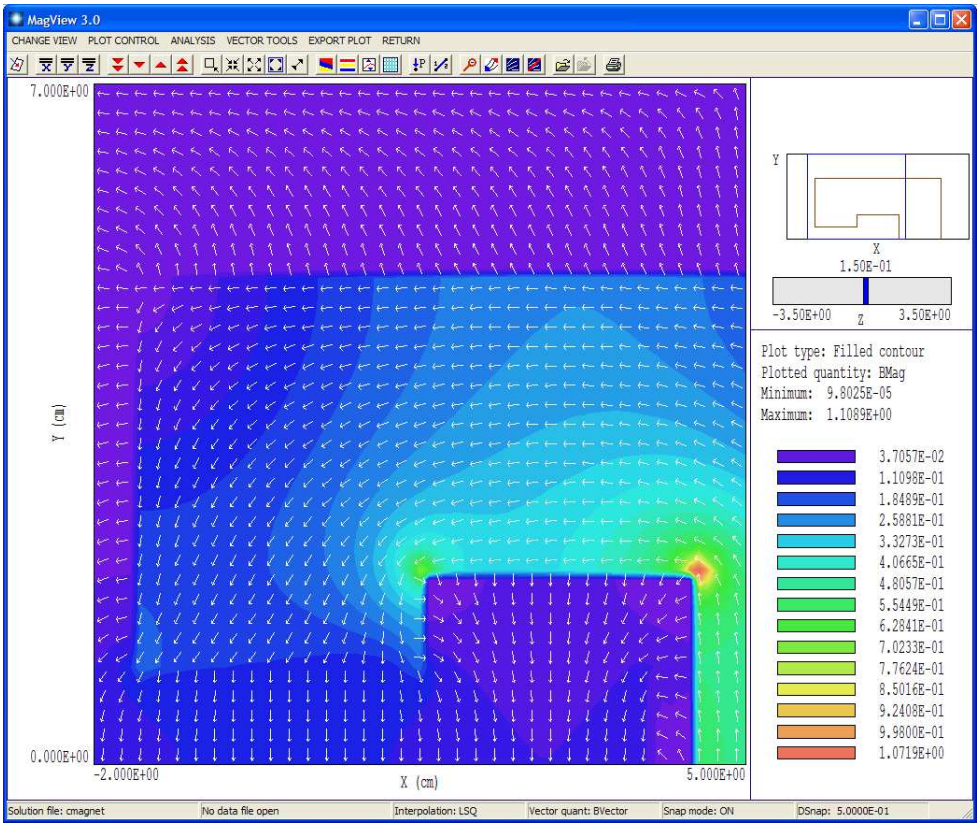


Figure 42: **MagView** working environment in the *Slice plot* menu – illustration of a filled contour plot.

9 MagView – slice plots

9.1 Setting the slice view

Slice plots are two-dimensional plots that show the variation of quantities over a plane normal to one of the Cartesian axes. In contrast to plane plots, slice plots are based on the structure of the mesh projected to a slice plane. This structure may be quite complex for a conformal mesh; therefore, slice plots require more computational effort. To facilitate the process, slices are constructed at discrete locations along the normal axis corresponding roughly to the planes of the foundation mesh. The precise rendering of mesh information enables point-and-click analysis operations (point calculation, line scan,...) in the slice. Note that when a coil file (`WND`) has been loaded, intersections of current elements with the slice plane are plotted as points. The point color corresponds to the coil number as in **MagWinder**.

The *Change view* popup menu contains commands to set the slice plane and to adjust the dimensions of the plot.

SET SLICE PLANE PROPERTIES
This command calls up the same dialog as the *Set plane* command in the plane plot menu (Fig. 38). You can change the normal axis, change the position along the normal axis, and set plot limits in the normal plane.

SLICE NORMAL TO X
SLICE NORMAL TO Y
SLICE NORMAL TO Z
Quick commands to change the normal axis.

JUMP FORWARD
STEP FORWARD
STEP BACKWARD
JUMP BACKWARD
Move along the slice axis by small or large steps. The small step is approximately one layer of the foundation mesh and the large step is 5 layers. The term *forward* implies motion toward higher indices of the normal axis. The slider bar in the orientation area to the right of the plot (Fig. 42) shows the current location.

ZOOM WINDOW
As an alternative to the entries in the *Set slice plane* dialog, you can interactively change plot limits in the normal plane using the mouse. Choose the command and move the mouse pointer into the plot area. The status bar enters coordinate mode. It shows the current mouse position in the plot. Use the left button to pick one corner and then move the mouse to create a view box. Click the left button again, and the plot regenerates. On any coordinate operation, press the *F1* key if want to enter values from the keyboard. Note that the normal plane box in the orientation area to the right of the plot (Fig. 42) shows the dimensions of the slice plane and the outline of the current zoomed view.

ZOOM IN
Enlarge the plot about the current view center.

EXPAND VIEW
Expand the plot about the current view center.

GLOBAL VIEW
Enlarge the plot boundaries to show the entire normal plane.

PAN
When the plot is zoomed, you can use this operation to shift the current view center. Use the mouse to define relative start and end points for the shift.

Figure 43: Slice plot style dialog.

9.2 Setting slice plot properties

The commands in the *Plot control* popup menu are used to set the plot style and mouse options.

SET SLICE PLOT STYLE

This command brings up the dialog of Fig. 43. The *Region* plot style is a cross-section view of the mesh element divisions color-coded by region. In contrast to the logical plane plot of **MetaMesh**, **MagView** attempts to resolve the exact mesh structure in the plane. In the *Filled contour* style (Fig. 42), the program determines discrete bands of color coding according to values of the current plot quantity. The *Contour* style shows lines of constant values of the plot quantity. Finally, an *Element* plot has color coding by the average value of the plot quantity in the element volume. When the *Element outline* box is checked, **MagView** includes facets in the *Region* and *Element* modes. In comparing relative advantages, the *Filled contour* plot provides the most attractive and accurate display in extended field regions. Use the *Element* plot for the best view of discrete field changes at the boundaries of ferromagnetic materials. The *Contour* plot is useful for checking field uniformity over a volume.

PLOT QUANTITY

Use this command to set the quantity for color-coding in *Filled contour* and *Element* plots and line values in *Contour* plots. The choices depend on entries in the currently-loaded configuration file.

PLOT LIMITS

This command opens the dialog of Fig. 44. In the *Autoscale* mode, **MagView** chooses defaults for the minimum and maximum values for colors and contours based on the range of values of the current plot quantity in the current slice. Deactivate *Autoscale* to set the values manually. The manual mode is useful for comparing absolute field distributions in different slices. In either case, **MagView** uses a uniform interval between the limits based on the number of contour lines. For plots in the *Contour* style, you can set specific values for the contour lines.

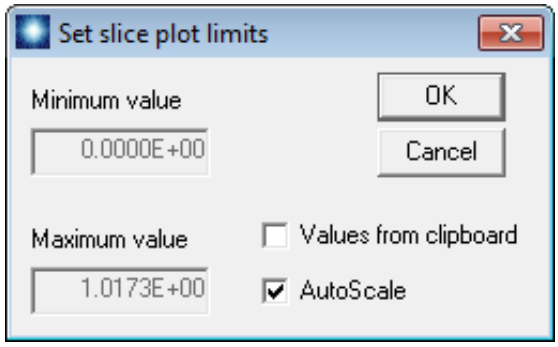


Figure 44: Dialog to set limits of the plotted quantity.

Applications include logarithmic intervals or the boundary of a region above a critical magnetic field. To use this feature, open a text document using the internal editor or an external program. Make a list of the values you want, using any valid numerical format. There should be one entry per line and the entries should appear in ascending order. Save the document if you want to compare a reference set. Highlight the values and copy them to the Windows clipboard. Then return to the plot limits dialog and check the *Values from clipboard box*. **MagView** displays an error message if the number set is invalid. Otherwise, the program reads the data, exits the dialog and updates the plot. Note that you will see a change only if the *Contour* style is displayed. Other plot styles are not affected.

RESET PLOTS TO DEFAULT

Return to the default view that applied when the solution was loaded

TOGGLE GRID

A set of dashed grid lines can be superimposed on slice plots. **MagView** automatically chooses intervals and positions so that the lines occur at convenient values along the horizontal and vertical directions (for example, 0.01 rather than 0.01153). Listings of the grid intervals are included in the axis labels. Grids corresponding to the normal plane axes are plotted as solid lines.

GRID INTERVALS

This command brings up a dialog to set intervals for grid lines in the horizontal and vertical directions. Deactivate the *AutoGrid* box and enter real numbers in the boxes. The values should be specified in the length units set by the *DUnit* parameter.

NUMBER OF CONTOUR LINES

Change the number of lines for *Filled contour* and *Contour* plots.

TOGGLE SNAP MODE

Mouse coordinates for commands such as *Zoom window*, *Pan*, and *Scan in slice* may be entered in two modes. In the normal mode, the returned position corresponds to the mouse position on

the screen. In the snap mode, the program picks a point at an even interval close to the mouse position. The returned point depends on the value of the parameter *DSnap*. For example if $DSnap = 0.1$ and the mouse is at position (6.2345,-5.6113), the returned position is (6.2000,-5.6000). The status bar displays the actual or snapped position of the mouse.

SNAP DISTANCE

Change the value of *DSnap* from the default value determined by the program.

9.3 Analyses in a slice

You can determine field values at points and along scan lines with the commands of the *Analysis* popup menu.

POINT CALCULATION

This command is useful to make quick checks of fields in the solution volume. After you click the *Point calculation* command, move the mouse into the plot area. The mouse pointer changes to a cross-hair pattern and the status bar enters coordinate mode. Click the left button to specify a point or press the *F1* key to enter the coordinates from the keyboard. Note that mouse coordinates will shift between discrete values if snap mode is active. **MagView** calculates field and material properties at the point in the normal plane given by the coordinates. The included quantities depend on definitions in the the currently-loaded configuration file. Results are recorded if a data file is open.

LINE SCAN

Line scans are one of the most useful **MagView** capabilities. After clicking on the command, supply two points with the mouse to define a scan line (or press the *F1* key to enter coordinates manually). The snap mode is useful in this application (for example, you may want the scan to extend from 0.000 to 5.000 rather than 0.067 to 4.985.) The program computes a series of values of field quantities in the normal plane at equal intervals along the line. Complete information is recorded if a data file is open. The program also makes a screen plot of the currently-selected quantity versus distance along the scan and activates the *Scan plot* display (Fig. 45). **MagView** adds fiducial lines to the plot using intelligent grid selection. This means that the plot is adjusted to fill the screen and grids are drawn at useful intervals (*i.e.*, 0.05 or 2.00).

SET SCAN QUANTITY

With this command you can pick the quantity that will be displayed in screen and exported plots of line scans. Pick the quantity from the list box and click *OK*. The listed quantities depend on the entries in the `INTERPOLATION` section of the configuration file. This setting has no effect on the data file listing which includes all requested field quantities.

SET NUMBER OF SCAN POINTS

This command sets the number of line scan points in the screen plot and data file listing. The default value is 150 and the maximum number is 500.

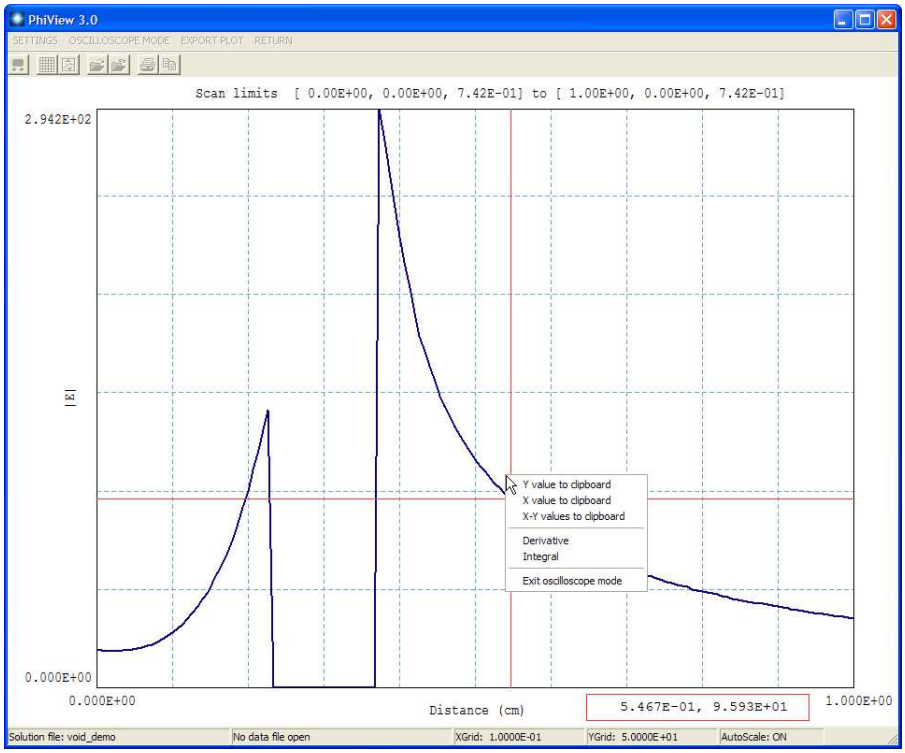


Figure 45: Scan plot display.

9.4 Scan menu

As shown in Fig. 45, the current scan quantity is plotted as a function of distance from the start point. The coordinates of the start and end points are listed above the graph. Coordinates are plotted in units set by the *DUnit* parameter. In the default mode, **MagView** sets limits so that the plot fills the screen. The program includes grid lines at even intervals. The status bar at the bottom lists the solution file, the data file (if open) and the grid intervals. The *Export plot* menu includes that standard set of commands described in the previous chapter. The *Settings* menu contains the following commands.

SYMBOL DISPLAY

Toggle the plot display to either a thick line or a set of symbols at calculated points connected by a thin line.

VERTICAL LIMITS

Override automatic selection and set vertical limits for the current quantity. To set manual limits, uncheck the *AutoScale* box and enter real-numbers values for the lower and upper limits of the plotted quantity. Check the *Autoscale* box to return to the default mode.

TOGGLE GRID

Suppress or activate the display of grid lines.

GRID INTERVALS

Override automatic selection of grid intervals. To set values for the horizontal and vertical grid intervals, uncheck the *AutoGrid* box and enter real-number values. For the horizontal direction, enter the interval in length units set by the *DUnit* parameter. Check the *Autogrid* box to return to the default mode.

OPEN DATA FILE

In the oscilloscope mode, you can record a variety of numerical values for the scan. The information will be written to a data file if already opened. Use this command to open a new data file.

CLOSE DATA FILE

Close the current data file. This command should be invoked before attempting to use the file in another program.

OSCILLOSCOPE MODE

In response to this command, the scan plot becomes interactive with many of the features of a digital oscilloscope. The commands and tools of the *Scan plot* menu are deactivated and **MagView** displays the red cross-hair lines shown in Fig. 45. Values of the distance and plot quantity at the cross-hair intersection are displayed in the red box below the plot. The intersection point follows the mouse when the cursor is inside the plot area. Note that the intersection moves discontinuously between the data points. To check points at the end of the interval, move the mouse slowly toward the left or right-hand boundaries.

If a data file is open, **MagView** records values at the current intersection point when you click the left mouse button:

```
Scan point calculation
    Distance:  1.386667E+00 (in)
    BMag:  1.811660E-01
```

Clicking the right button activates the pop-up menu shown in Fig. 45. The menu contains the following commands:

Y VALUE TO CLIPBOARD

Copy the value of the plotted quantity at the cross-hair intersection to the Windows clipboard. You can then past the number into a text document, spreadsheet or calculator program.

X VALUE TO CLIPBOARD

Copy the value of the distance (in units set by *DUnit*) at the cross-hair intersection to Windows clipboard.

XY VALUES TO CLIPBOARD

Copy the values of the distance and plotted quantity to the clipboard. The x value is on the first line and the y value on the second.

DERIVATIVE

Calculate and record the numerical derivative at the intersection point. The dimensions are determined by the *DUnit* parameter. For example, if the scan quantity is magnetic flux density in tesla and the distance is shown in centimeters, the derivative has units tela/cm. **MagView** writes the value to the clipboard and also displays the results in a dialog. If a data file is open, the program makes a record like the following one:

```
Derivative
     Location:  1.612000E+00 (in)
     BMag/(in): -2.901895E-01
```

INTEGRAL

Calculate and record the definite integral of the plotted quantity from the start point of the scan to the cross-hair intersection point. The dimensions are determined by the *DUnit* parameter. For example, if the scan quantity is the magnetic field in A/cm and the distance is shown in centimeters, the integral has units (A/cm)-cm = A. **MagView** writes the value to the clipboard and also displays the results in a dialog. If a data file is open, the program makes a record like the following one:

```
Definite integral
     Location:  1.820000E+00 (in)
     BMag-(in):  4.664221E-01
```

EXIT OSCILLOSCOPE MODE

Activate commands and tools and return to the standard *Scan plot* mode. You can also press the *Esc* key to exit the oscilloscope mode.

9.5 Slice vector tools

MagView has useful tools to display the direction of the vector quantities **B** and **H** in slice plots. Figure 46 shows the corresponding entries on the toolbar.

VECTOR PROBE

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 46. When you click on the tool and move the cursor into the slice-plot area, it changes to a semi-transparent probe that rotates about a pivot point to show the direction of the chosen field quantity. The probe functions in all plot styles and may be combined with field-line traces and scatter plots. The status bar shows the coordinates and $|\mathbf{B}|$ at the pivot point.

VECTOR LINES AT POINTS

Use this command to add projected lines of $|\mathbf{B}|$, as in Fig.47. The program enters coordinate entry mode when you click the command. Move the mouse to a point in the solution volume and click the left button. **MagView** calculates the three-dimensional path of a magnetic field

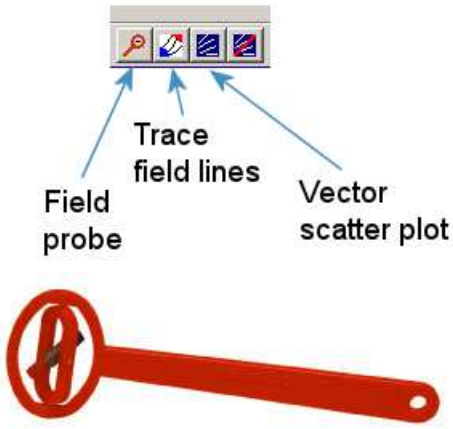


Figure 46: Tools to display vector information in the slice plot menu.

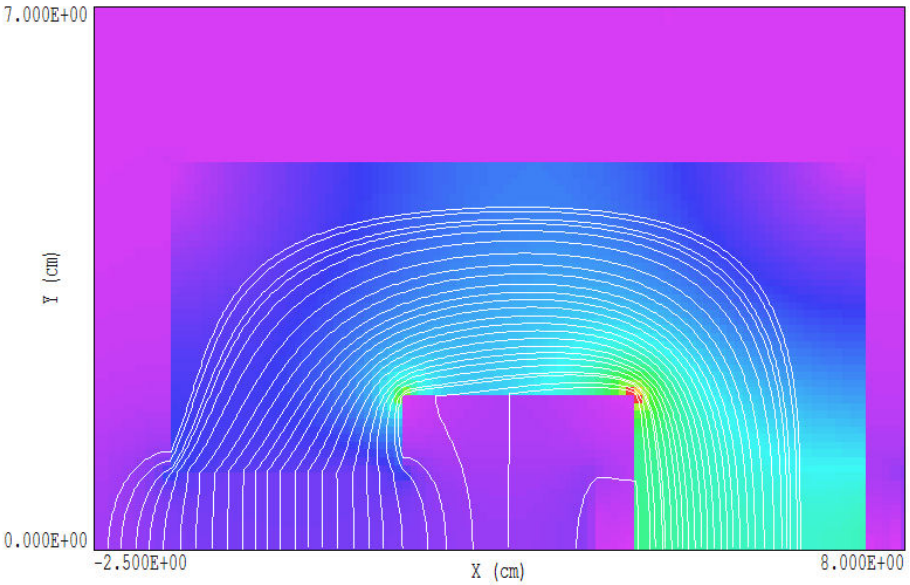


Figure 47: Element plot with added field line projections.

line that passes through the point and plots the projection in the slice plane. You can continue to add any number of lines. Click the right mouse button or press *ECS* to exit coordinate mode. The lines are not included in hardcopy or plot file exports and disappear if you change the slice view. Use a screen capture utility to record them. It is important to recognize the nature of the plots. The lines are three-dimensional curves projected to the slice plane. They may be difficult to interpret if the line does not lie close to the slice plane. For full three-dimensional field line plots, see the *Field line plot file* command in Sect. 10.

VECTOR ARROW PLOT

Superimpose a uniform distribution of vector arrows pointing in the direction of **B** and **H** in the slice plane. Arrows may be added to any of the plot types, including *Region*. They are included in hardcopy and plot file export. The arrows are preserved and adjusted if you change the slice view.

REMOVE VECTORS

Use this command to turn off the vector scatter plot mode and to remove vector tool displays from the current plot.

VECTOR QUANTITY

Choose whether to plot **B** or **H** vectors.

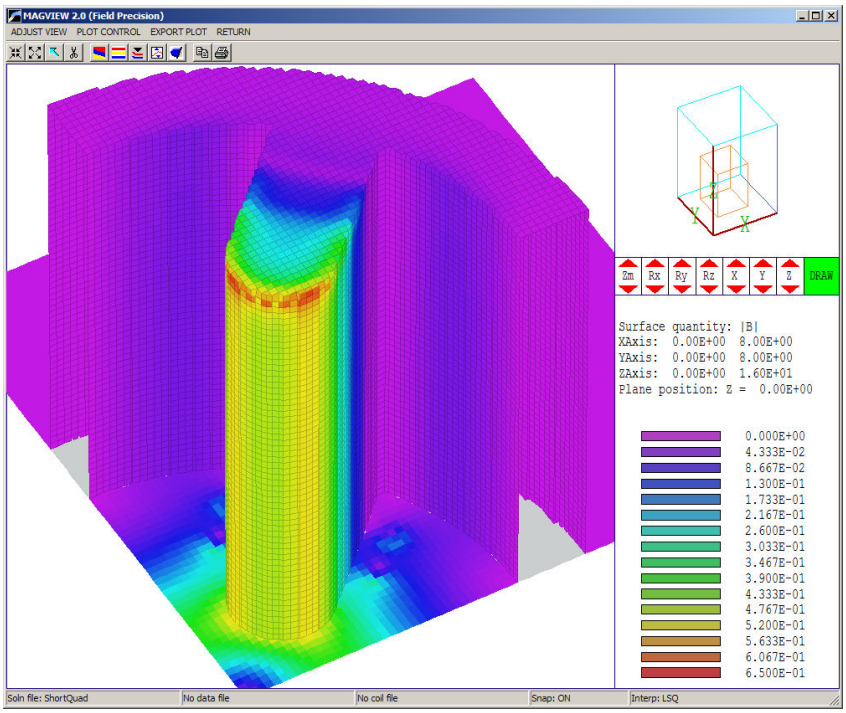


Figure 48: **MagView** working environment for surface plots.

10 MagView – surface plots

Surface plots are three-dimensional views of the solution (Fig. 48). Four types of information may be superimposed: 1) region boundaries color-coded by region number, 2) region boundaries with color-coding by a computed quantity, 3) computed quantities in a slice plane normal to one of the Cartesian axes and 4) applied field coils. Applied field coils are included whenever a current element file is open (Chap. 7). Surface plots are created from the conformal mesh and preserve true spatial scaling.

The method to control the three-dimensional display with the mouse is identical to that used in **MetaMesh**. Figure 49 shows the active areas of the screen. The central zone (A) is used for zooming in (left button) and out (right button). Hold down the left mouse button in zones B, C, D and E to walk around the object. Hold down the right mouse button in zones B, C, D and E to move the viewpoint to the right, upward, to the left and downward. Note that changes are reflected in the orientation box in the upper-right portion of the screen. The plot is updated when you release the mouse button. You can control additional aspects of the three-dimensional view with the commands of the *Adjust view* popup menu.

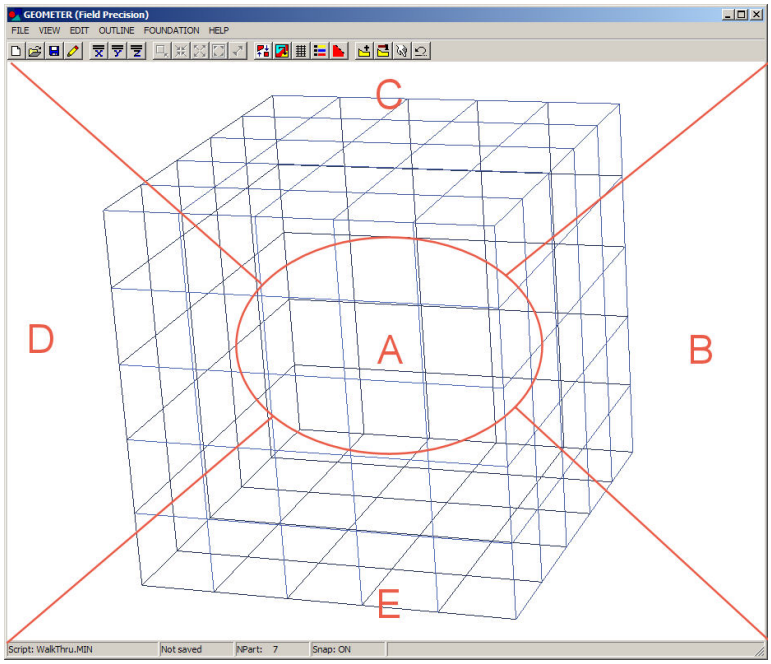


Figure 49: Active areas for mouse control of the 3D view.

SET VIEW DIALOG

This command brings up a dialog (Fig. 50) where you can set specific view angles, displacements and the relative distance to the viewpoint, $DView$. The parameter $DView$ controls perspective. Set it to a large value ($DView \gg 1.0$) for an orthographic view. The minimum value is 1.5.

RESTORE DEFAULT VIEW

This command is useful if you loose your orientation after several rotations and translations. The view is returned to the default: $\theta_x = -30^o$, $\theta_y = 0^o$ and $\theta_z = 45^o$ with the origin at the center.

+X VIEW
+Y VIEW
+Z VIEW

Rotate to views from the $+x$, $+y$ or $+z$ directions. Origin shifts are not affected

CENTER VIEW

Remove shifts by setting the origin to the center of the solution volume.

ORTHOGONAL/PERSPECTIVE VIEW

By default, **MagView** includes perspective correction in surface plots. Click on this command to move the view point to infinity. Click the command a second time to restore the view point.

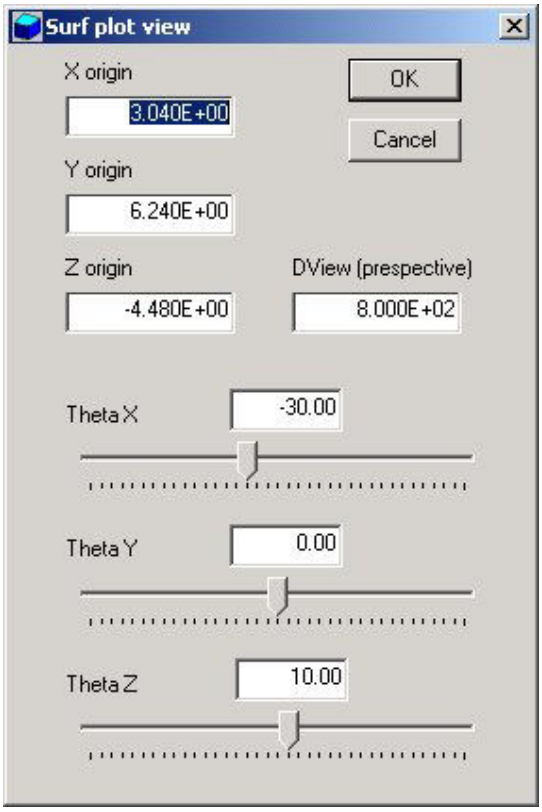


Figure 50: Dialog to set the three-dimensional view.

The commands of the *Plot control* popup menu control the appearance of the plot.

PLOT STYLE

This command brings up the dialog of Fig. 51. The group of commands on the left-hand side controls the plot style. A three-dimensional plot is constructed from color-coded facets. There are two types of facets: 1) element facets on the boundary of a region with color coding by region number or the selected computed quantity and 2) rectangular facets comprising a slice plane with color-coding by the values of the computed field quantity. Slice plane and region boundary information may be superimposed. The *Facet style* radio buttons control whether the facets of region boundaries are plotted as solid plates (hidden surface) or wireframe outlines. The *Include facet boundaries* check box determines whether the boundaries of region facets are plotted in the hidden-surface mode. (Note that the facets of the slice plane are always plotted as solids with no outline). **MagView** plots a reference grid along the boundaries of the solution volume when the *Include reference grid* box is checked. The group of commands on the right-hand side controls the slice plane. The *Include slice plane* box determines whether a slice plot is included. The radio buttons determine the axis normal to the slice plane. You may move the plane along the chosen axis by entering values in the box or moving the slider.

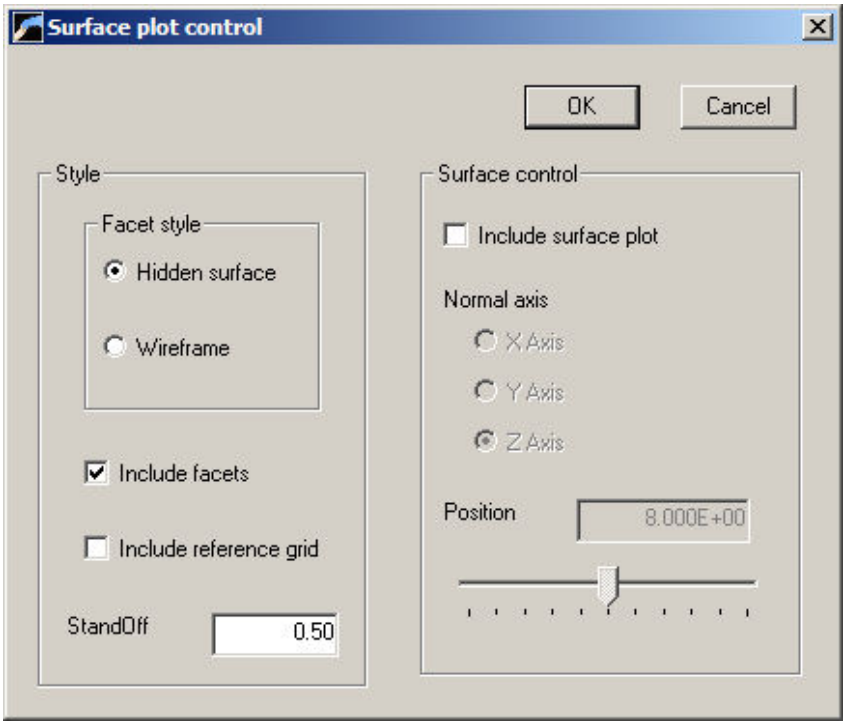


Figure 51: Surface plot control dialog.

DISPLAYED REGIONS

The command brings up the dialog of Fig. 52 where you can pick region boundaries to include in the plot. Depress a button in the *Display* column to activate a region. The buttons in the column marked $|E|$ determine the presentation style for boundary facets. If the box is uncheck, **MagView** colors facets by region number. If one or more buttons are depressed in the $|B|$ column, color code is by the computed field quantity. The information window shows the correspondence between color and the quantity. Note that the field quantity is always calculated at a point near the facet **outside** the chosen region. If a boundary separates two materials with different values of μ_r, the field values depend on which region is chosen for display.

CUT PLANES, REGIONS

In a hidden surface plot, internal geometric details or the normal plane plot may be obscured by surrounding region boundaries. This command brings up a dialog that allows you to adjust the displayed portions of region surfaces along the x, y and z axes. **MagView** does not display facets that lie outside the limits. With this feature you can create cutaway views. The default is that cut limits are set equal to the dimensions of the solution volume so that all facets are included.

COIL CUT PLANES

Use this command to limit plotted current elements when coils are displayed.

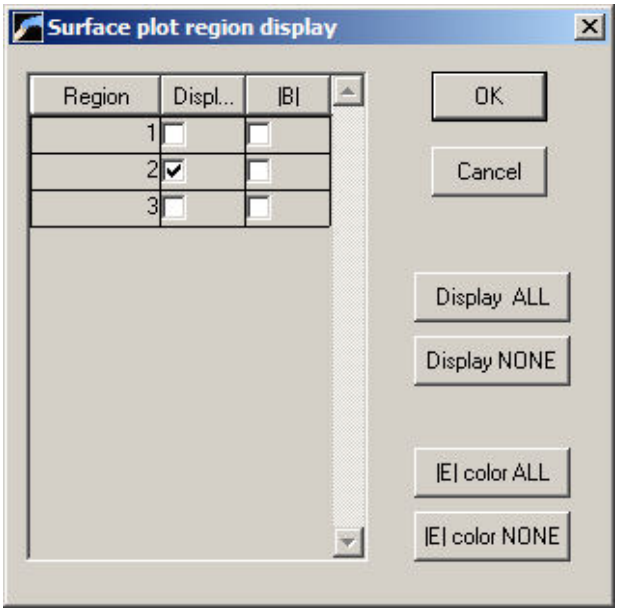


Figure 52: Region display dialog.

PLOT QUANTITY

With this command, you can change the quantity used for color-coding of region boundaries and the slice plane. The choices depend on the those defined in the currently-loaded configuration file.

SET PLOT LIMITS

In the *Autoscale* mode, **MagView** chooses colors so that the spectrum spans the range between the minimum and maximum values of the plotted quantity in the current plane. Scaling values change automatically if you change plot quantities or move to a different plane. To set the range manually, deactivate the autoscale check box and enter minimum and maximum values.

VECTOR LINES FROM POINT FILE

You can add true magnetic field lines to three-dimensional plots with this command. To use it, you must prepare a text file that contains any number of point coordinates. **MagView** reads the file and adds field lines that pass through each point. The file consists of data lines, where each line contains three real numbers (x, y, z) separated by any of the standard delimiters. Enter the coordinates in units set by *DUnit*. The file may contain comment lines that start with an asterisk and should end with the *ENDFILE* command. Lines are not plotted for target points outside the solution volume.

RECORD VECTOR LINE COORDINATES

Use this command to record calculated paths of field lines. **MagView** prompts for the name of an output text file with a name of the form `FNAME.FLP`. A file extract is shown below. The command functions only if a file of starting points has been opened with the *Field line plot file* command.

```
Magnetic field line plots for file: cmagnet.HOU
   DUnit(unit conversion factor):   1.00000E+02

Start point
    X:   8.00000E+00
    Y:   2.50000E-01
    Z:   3.00000E+00
Forward
  8.00000E+00  2.50000E-01  3.00000E+00
  8.00599E+00  2.50385E-01  3.00003E+00
  8.01197E+00  2.50768E-01  3.00006E+00
  8.01796E+00  2.51151E-01  3.00009E+00
  8.02395E+00  2.51532E-01  3.00011E+00
  8.02994E+00  2.51912E-01  3.00014E+00
```

CLOSE VECTORS

Close the field line plot file and remove field lines from the current plot.

VECTOR QUANTITY

Choose whether to calculate **B** or **H** vectors.

11 Theoretical background

11.1 Reduced potential model

This chapter gives detailed information on the physical and mathematical foundation of **Magnum**. It is not necessary to understand all details. The goal is to give an idea of the steps involved in a solution to help you create effective setups. The most familiar method for numerical solution of magnetic fields is to solve the following equation for the vector potential:

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{j}_s. \tag{19}$$

In Eq. 19, the quantity $\mathbf{j}_s$ is the applied current. Currents resulting from the presence of materials are represented by the factor $1/\mu$. The variation of the quantity in space is determined by the geometric distribution of ferromagnetic and conductive materials. Equation 19 follows from the definition of vector potential,

$$\mathbf{B} = \nabla \times \mathbf{A}, \tag{20}$$

and Ampere's law,

$$\nabla \times \left(\frac{\mathbf{B}}{\mu} \right) = \nabla \times \mathbf{H} = \mathbf{j}_s. \tag{21}$$

In Eq. 21, the quantity $\mathbf{B}$ is the *magnetic flux density* and $\mathbf{H}$ is the *magnetic field*.

In two-dimensional solutions, there is only a single component of the vector potential $\mathbf{A}$ and Eq. 19 reduces to the same form as the Poisson equation used in electrostatics. Therefore two-dimensional finite-element magnetostatic solutions are no more difficult than electrostatic solutions. The computation is more difficult in three dimensions. The three components of vector potential are coupled through spatial variations in μ. The finite-element equations to represent Eq. 19 on a hexahedron mesh involve 234 coupling coefficients per node as opposed to 26 coefficients in a solution of the Poisson equation. Because of the increased computation times and storage requirements, a direct solution of Eq. 19 on a large mesh is impractical on a personal computer. There are two other problems associated with the equation:

- It is more difficult to identify boundary conditions on the vector quantity $\mathbf{A}$.
- It is challenging to convert current carried by a discrete set of applied current elements to three-dimensional current density $\mathbf{j}_s$ on a conformal mesh of hexahedrons.

Magnum utilizes the indirect *reduced potential* model for magnetostatic solutions in three dimensions. With this method, storage requirements and computation times are only slightly higher than those for an electrostatic solution. The key is to divide the magnetic field into two components:

$$\mathbf{H} = \mathbf{H}_s + \mathbf{H}_m. \tag{22}$$

The component $\mathbf{H}_s$ arises from the applied currents while $\mathbf{H}_m$ is the component created by currents in materials. By superposition, the components individually satisfy the equations:

$$\nabla \times \mathbf{H}_s = j_s, \tag{23}$$
$$\nabla \times \mathbf{H}_m = 0. \tag{24}$$

Given the spatial distribution of applied current $\mathbf{j}_s$, the component $\mathbf{H}_s$ can be determined directly from a Biot-Savart integral (Sect. 11.2) at all points in space without recourse to finite-element methods. Equation 24 implies that the material component $\mathbf{H}_m$ may be expressed as the gradient of a scalar potential:

$$\mathbf{H}_m = -\nabla\phi. \tag{25}$$

The quantity ϕ is called the *reduced potential*. The equation

$$\nabla \cdot \mathbf{B} = 0, \tag{26}$$

implies that

$$\nabla \cdot [\mu(\mathbf{H}_s - \nabla\phi)] = 0, \tag{27}$$

or

$$\nabla \cdot (\mu\nabla\phi) = \nabla \cdot (\mu\mathbf{H}_s). \tag{28}$$

Equation 28 has the form of the Poisson equation with a source term on the right-hand side. The source term can be determined from the known spatial variations of μ and $\mathbf{H}_s$.

The **Magnum** solution procedure consists of the following steps:

- Read a file (`MDF`) created by **MetaMesh** that gives node coordinates and the region identities of elements.
- In the **Magnum** script (`GIN`), read values of μ to associate with element region numbers.
- Using geometric and material information, calculate finite-element coefficients to represent Eq. 28 as a set coupled linear equations, one for each node.
- Read a file (`WND`) of applied current elements created by **MagWinder** and apply a Biot-Savart integral to determine values of $\mathbf{H}_s$ at nodes.
- Analyze the mesh to calculate source terms $\nabla \cdot (\mu\mathbf{H}_s)$ at nodes.
- Solve the coupled set of node equations by iterative methods to find the reduced potential ϕ.
- Record the applied field $\mathbf{H}_s$ and reduced potential ϕ at each node in the **Magnum** output file (`GOU`).

- In the **MagView** post-processor, the magnetic flux density is determined by interpolations using the equation:

$$\mathbf{B} = \mu(\mathbf{H}_s - \nabla\phi). \tag{29}$$

Magnum supports two computational modes. We call the full calculation with applied fields and reduced potential the *Standard* solution mode. If there are no materials that have $\mu_r \neq 1.0$, we can omit the calculation of ϕ and simply record values of $\mathbf{H}_s$ in the output file. We refer to this as the *Free* solution mode (for free-space fields). In this case, interpolations in **MagView** return values of the applied fields over the solution space. The corresponding field values are those that occur in an infinite space, unaffected by the boundaries of the solution volume.

11.2 Applied field calculation

Magnum uses a simple yet versatile method to define three-dimensional coils of any degree of complexity. The program expects to receive a list of short current elements in the `WND` input file. Section 3.13 describes the file format. Figure 53 shows the geometry of a filamentary element of length

$$L = \sqrt{(x_e - x_s)^2 + (y_e - y_s)^2 + (z_e - z_s)^2}, \tag{30}$$

at average position

$$\mathbf{r}' = \left[\frac{x_s + x_e}{2}, \frac{y_s + y_e}{2}, \frac{z_s + z_e}{2}\right] \tag{31}$$

carrying current I. The unit vector

$$\mathbf{u} = \left[\frac{x_e - x_s}{L}, \frac{y_e - y_s}{L}, \frac{z_e - z_s}{L}\right] \tag{32}$$

lies along the filament direction. The contribution to the magnetic field at position $\mathbf{r}$ from the element at position $\mathbf{r}'$ is given by

$$d\mathbf{H}_s = \frac{IL}{4\pi}\left[\frac{\mathbf{u} \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3}\right]. \tag{33}$$

Magnum computes the total applied magnetic field $\mathbf{H}_s$ at each node position by taking a sum over current elements in the `WND` file. One advantage of the approach is that numerically-derived model particle orbits in the **OmniTrak** code can be treated as a sequence of current elements. Hence it is straightforward to include the contribution of beam currents to magnetic fields. The drawback of the method is that it takes time to calculate fields created by thousands of current elements at millions of node points.

The field contribution of a filamentary current-element may be quite large if it lies close to a node. To avoid diverging field values, **Magnum** treats an element of length L as a cylinder with current-density distributed uniformly over diameter L (Figure 54). The code uses a look-up table derived from a numerical solution within the region shown as a dashed line of Fig. 54. To reduce calculation time, the code applies Eq. 33 (using the average position

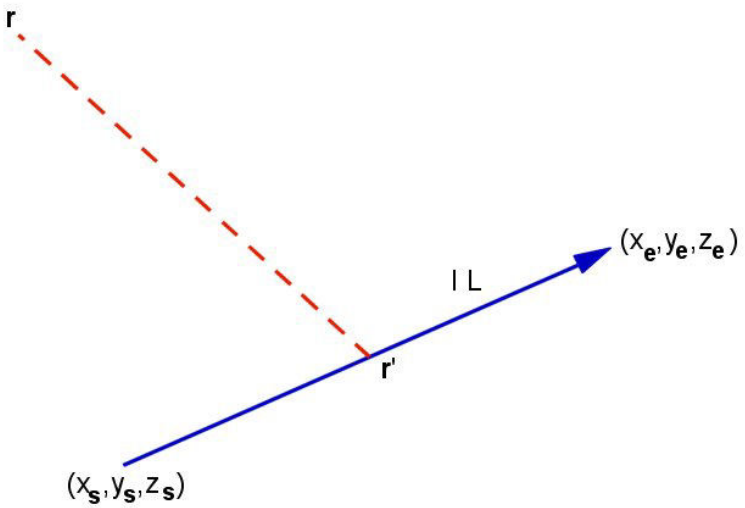


Figure 53: Current-element geometry.

of the element) outside the region. The effects of the finite-width of applied currents will be apparent in **MagView** plots. You can adjust the width through the choice of element length in **MagWinder** (Chapter3).

11.3 Representing perfect conductors

Magnum can model both linear and non-linear isotropic magnetic materials (*i.e.*, soft iron and ferrites). For linear materials the value of μ does not depend on the field amplitude. In this case, the magnetic field and flux density vectors are co-linear and have magnitudes related by

$$\mathbf{B} \;=\; \mu\,\mathbf{H} \;=\; \mu_o\,\mu_r\,\mathbf{H}. \tag{34}$$

Magnum does not handle old style permanent magnets with highly non-linear demagnetization curves.

The physical properties of high-μ materials are familiar from introductory courses on electromagnetism. In the remainder of this section, we shall discuss the properties of materials with $\mu_r \ll 1.0$ and their application in **Magnum**. In such a material, the magnetic flux density is approximately $\mathbf{B} = 0.0$. Equation 26 implies that the normal component B_n across the boundary is continuous. Therefore, the magnetic flux density just outside the material must be parallel to the surface. In other words, a material with $\mu_r \ll 1.0$ acts like a perfect conductor. Figure 55*a* shows the effect of such a material on the fields produced by an external pulsed current. You can find the distribution of surface current on the conductor by calculating the magnetic field in the adjacent air region and taking,

$$J_s = \frac{\Delta B_{\|}}{\mu_{\;0}}, \tag{35}$$

where the surface current $\mathbf{J}_s$ has units of A/m.

A more interesting situation results when a drive current is inside a low-μ region. In this case, the reduced-potential solution creates a distribution of material currents that ensure that $\mathbf{B} = 0.0$ at all points inside the material. As an example, Figure 55*b* shows a cross section of

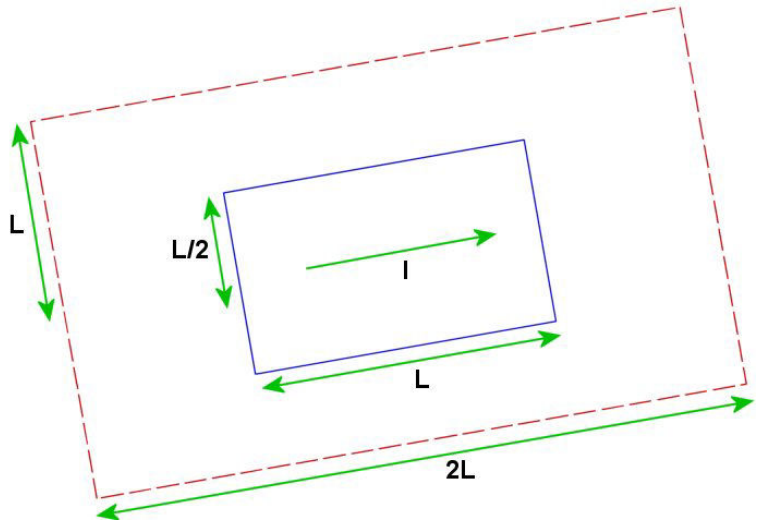


Figure 54: Representation of a current element as a cylinder of radius $L/2$. Dashed line indicates the transition boundary to the Biot-Savart calculation.

a coaxial transmission line. The inner and outer conductors are represented as materials with $\mu_r \ll 0.0$. An offset drive current flows in the z direction. Even though the drive current is displaced from the axis, the numerically-derived fields are azimuthally symmetric and closely follow the $1/r$ variation of magnitude predicted for the geometry.

An example will clarify methods for modeling perfect conductors. Figure 56 shows the geometry (the files `SURFCURRENT.MIN`, `SURFCURRENT.CDF` and `SURFCURRENT.GIN` are included in the example library). A toroidal conductor inside a toroidal chamber carries a pulsed azimuthal current of 2000 A – we want to find fields in the intervening space. The finite-element simulation treats one-quarter of the geometry. The default boundary condition of $B_\perp = 0.0$ (Sect. 11.5) is appropriate for the open ends. It is necessary to use a full circular drive current loop to ensure that the applied fields have correct symmetry at the boundaries. The condition $\mu_r = 0.0001$ is assigned to elements inside the brown facets of Fig. 56 (inner conductor) and elements outside the blue facets (outer conductor). Elements in the intervening space have $\mu_r = 1.0$. We find that the results are independent of the radius or position of the drive current loop as long as it is inside the inner conductor. Figure 57 shows calculated values of $|\mathbf{B}|$ in a normal plane, the summation of applied and material contributions. The field is zero inside the conductors and is azimuthally symmetric in the air gap.

11.4 Dual-potential model

The reduced-potential model always provides good results in regions of the simulation where the relative magnetic permeability μ_r is not much larger than unity. In iron regions where $\mu_r \gg 1.0$, the two contributions in Eq. 29 may be almost equal. The small difference is multiplied by μ_r to determine the magnetic flux density $\mathbf{B}$. Small interpolation errors in $\mathbf{H}_m$ and $\mathbf{H}_s$ may give errors in $\mathbf{B}$. For this reason, **Magnum** includes the option to generate an alternate finite-element solution for iron regions where $\mu_r \gg 1.0$. This approach is called the *dual-potential* model.

Figure 58 shows the division of a solution volume into regions representing air ($\mu_r = 1.0$),

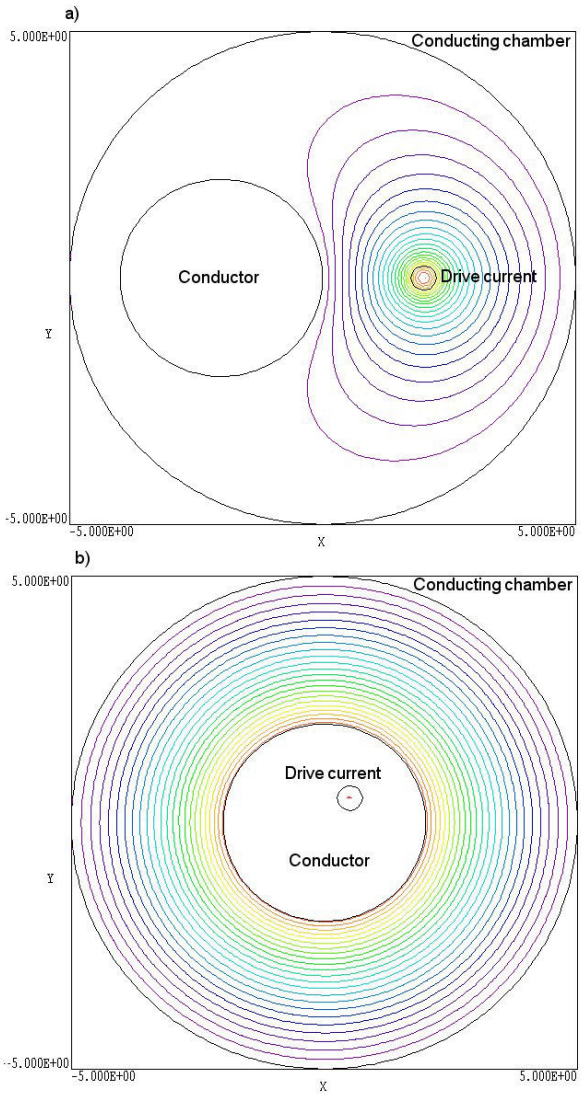


Figure 55: Effect of materials with $\mu_r \ll 1.0$ on lines of magnetic flux density $\mathbf{B}$. *a*) Drive current external to the region. *b*) Drive current internal to the region.

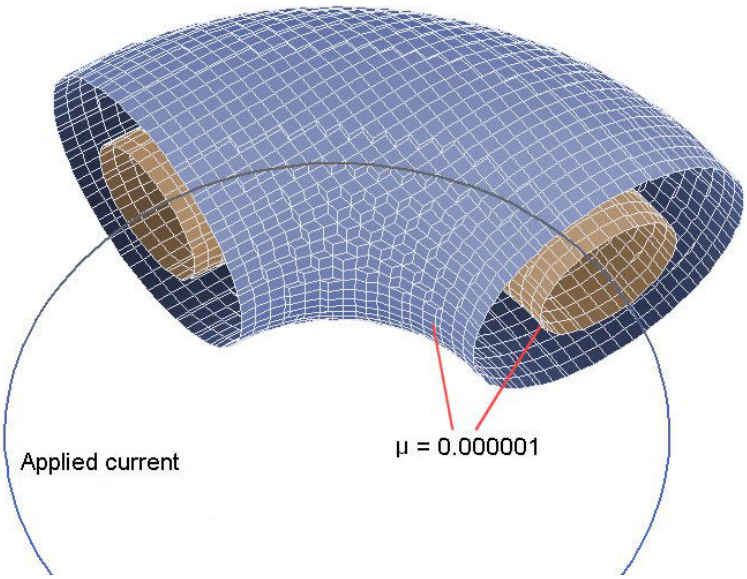


Figure 56: Toroidal conductor with a pulsed azimuthal current inside a conducting toroidal housing.

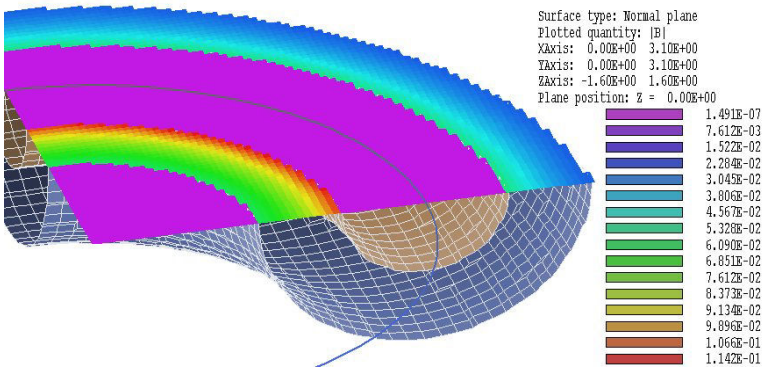


Figure 57: Example `SURFCURRENT` – $|\mathbf{B}|$ in the plane $z = 0.0$.

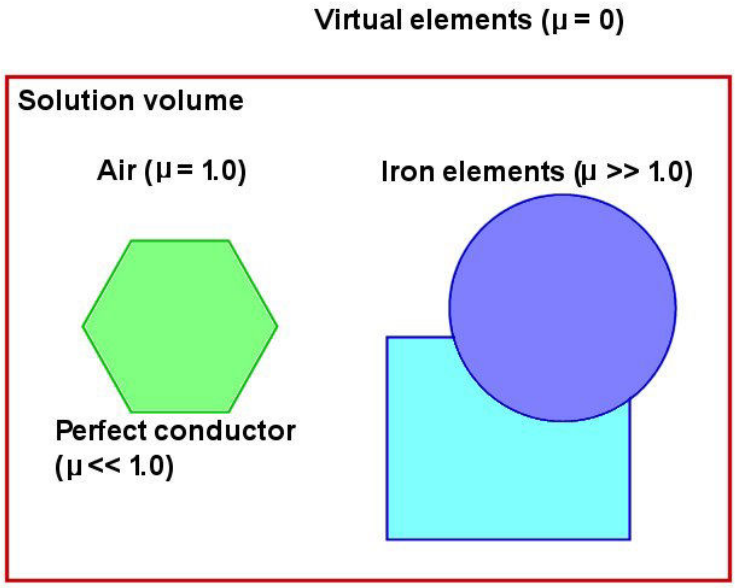


Figure 58: Schematic view – division of the solution volume for a dual-potential solution.

perfect conductors ($\mu_r \ll 1.0$) and ferromagnetic materials ($\mu_r \gg 1.0$). In a physically-correct solution, applied currents may wrap around iron objects but they do pass directly through them. With the condition $\mathbf{j}_s = 0.0$, the following condition holds on the total magnetic field inside the iron:

$$\nabla \times \mathbf{H} = 0. \tag{36}$$

Therefore, we can set $\mathbf{H}$ equal to the gradient of a scalar potential:

$$\mathbf{H} = -\nabla\Psi \tag{37}$$

In this case, Eq. 26 implies that the potential satisfies the Laplace equation:

$$\nabla \cdot (\mu\nabla\Psi) = 0 \tag{38}$$

The approach in the finite-element method is to generate discrete node equations by setting the local integrals of the left-hand side of Eq. 38 around nodes equal to zero. This procedure leads to the following equation for ψ at node i:

$$\int\int\int_{\Omega_i} \mu\nabla N_i \cdot \nabla\Psi dV = \int\int_{S_i} N_i \mu \frac{\partial\Psi}{\partial n} dS_i. \tag{39}$$

In Eq. 39 the quantity N_i is a weighting function that we can identify with the shape functions of the hexahedron elements surrounding the node. The volume integral on the left-hand side is identical in form to the integral for the reduced potential ϕ. Therefore, we can use the coupling coefficients that have already been determined for the reduced potential calculation. The new term is the surface integral on the right-hand side. The normal derivative of ψ at the surface is equivalent to

$$-\mu\frac{\partial\Psi}{\partial n} = B_{n.} \tag{40}$$

In other words, the node equations include source terms given by surface integrals of the normal component of the magnetic flux density. The form of the node weighting functions N_i and the continuity of magnetic flux guarantees that contributions from all surfaces inside contiguous iron regions equal zero, even if μ varies through the material or if two different regions intersect. Non-zero contributions occur only on the interface between iron and air or conductor regions.

With this background, we can understand the additional steps in the **Magnum** solution procedure when the dual-potential option is invoked:

- After completing the reduced-potential calculation, **Magnum** organizes the mesh and marks nodes that are connected to iron elements. Boundary nodes connected to both iron and air elements are also marked. The goal is to determine values of ψ on this node set. The set may contain multiple connected or disconnected iron regions.
- Coupling coefficients for the nodes are taken from those created for the reduced-potential calculation. Coupling coefficients to nodes outside the iron region or its boundaries are set to zero.
- Source terms for the boundary nodes are calculated from values of the normal component of **B** derived from the reduced potential model according to Eqs. 39 and 40.
- **Magnum** performs a second iterative solution to find ψ at the mesh nodes and records values in the output `GOU` file.
- Field calculations performed in **MagView** depend on the setting of the interpolation mode. In the default *Standard* mode, the equation $\mathbf{B} = \mu(\mathbf{H}_s - \nabla\phi)$ is applied over the full solution volume. In the *DualPot* interpolation mode, the program applies the equation $\mathbf{B} = \mu(\mathbf{H}_s - \nabla\phi)$ at points inside elements with $\mu_r \cong 1.0$ and uses $\mathbf{B} = -\mu\nabla\psi$ in elements with $\mu_r \gg 1.0$.

It is important to note that the dual potential calculation may fail under conditions where $\nabla\psi \equiv 0.0$ inside a material:

- Field lines in a pole travel between two symmetry boundaries with $\psi = 0.0$.
- A pole has no air gap so that lines of **H** circulate and reconnect completely with a pole (*e.g.*, a toroidal transformer core).

In these cases, do not use the *DualPot* command in **Magnum** and use the *Standard* mode for field calculations in **MagView.**

Finally, non-physical results may occur if drive currents pass through iron regions. **Magnum** incorporates two features to avoid this problem.

- **Magnum** checks the validity of the geometry after opening the mesh (`MDF`) and current element (`WND`) files. The program checks all current elements and reports an error if the average position of an element lies within an element with $\mu_r \gg 1.0$.

- With a solution (GOU) and current element (WND) file loaded in **MagView**, you can create three-dimensional plots to check the relative orientation of coils and iron regions. Note that the tests do not account for the width of current elements (Sect. 11.2). You must ensure that no portion of a finite-width element extends into an iron region.

11.5 Boundary and symmetry conditions

The application of boundary conditions in **Magnum** can be challenging because calculations may involve multiple sources of information: applied fields and material fields (from the gradient of the reduced potential ϕ) in non-iron regions and total fields from the gradient of the dual potential ψ in iron regions.

The field solution for ϕ and ψ with attendant boundary conditions has no effect on the applied field $\mathbf{H}_s$. Therefore, we must ensure the collection of current elements in the WND file generates a physically-correct applied field solution, regardless of the solution-volume limits and symmetries of the finite-element calculations for the potentials. For example, we must include the full current loop for the calculation of Fig. 56. Using only one-quarter of a the loop would give a non-physical applied field solution because the drive current would not define a circuit. Sometimes we can omit portions of a coil if the current elements make a negligible contribution to applied fields in the solution volume. For example, consider fields in a cylinder of radius R driven by a line current of infinite length. Because of the scaling of the Biot-Savart equation [Eq. 33], the contribution to the field of elements at an axial distance $z > 10R$ is less than 0.1% of the contribution from a nearby element. Therefore, it is sufficient to use a non-circuital line current of approximate length $20R$ to get a good approximation to the applied fields generated by an infinite wire. With some pre-analysis and judgement, you can substantially reduce the work involved in calculating the applied fields.

We shall next consider conditions for the calculation of the reduced potential ϕ. An underlying assumption in **Magnum** is that the solution volume is surrounded by a set of virtual elements with the property $\mu_r \ll 1.0$. Therefore, the magnetic flux density $\mathbf{B}$ is constrained to be parallel to any unspecified boundary. Equivalently, the solution volume is automatically surrounded by a virtual volume of perfect conductor. The alternative to the natural boundary condition is to specify a fixed value of reduced potential (usually $\phi = 0.0$) along the surface (*Dirichet* condition). In this case, the derivative of ϕ parallel to the surface must be zero. This condition implies that the material component of magnetic flux density $\mathbf{B}_m$ is normal to the boundary. If the applied component $\mathbf{B}_s$ is also normal, then the surface constitutes a symmetry boundary. This condition was applied in the example of Sect.1.2 to model one-quarter of the quadrupole.

Conditions on the dual potential ψ are the similar to those for ϕ. The region external to an iron region that contacts the solution boundary consists of virtual elements with $\mu_r \ll 1.0$. Therefore, the magnetic flux density $\mathbf{B}$ in an iron region is parallel to such a boundary. The condition $\psi = 0.0$ implies that the total magnetic flux density $\mathbf{B}$ is normal to the boundary. The resulting solution is physically-consistent only if the applied field $\mathbf{B}_s$ is also normal to the boundary.

12 Customizing MagView

You can create configuration files to customize the operation of **MagView** for your application. You can then switch between configurations for different types of analyses. **MagView** is supplied with a default configuration file in the same directory as the executable program: `magview_standard.cfg`. We recommend that you do not change this files. Instead, make a copy to use as a template and rename it. You can put your new configuration file anywhere, but we suggest you store all configurations in the same directory as `magview.exe`. Note that **MagView** remembers the last configuration used and attempts to reload it at the next run.

When you have created a template file, open it in an editor. Note that it is divided into the ten sections shown in Table 12. With few exceptions, do not make changes to the sections `HEADER`, `STOREDQUANT` `REGIONQUANT`, `PROGPARAM`, `RUNPARAM` and `REALTIMEPARAM`. Entries in these sections coordinate the operation of **Magnum** and **MagView**. Modifications may cause file load errors.

12.1 Defining calculated quantities

You can define custom quantities for display and analysis in the last four sections of the configuration file:

- **Interpolation**. Scalar quantities for multiple uses: 1) displays in slice, plane and surface plots, 2) results of point and scan calculations and 3) values written to matrix output files.
- **Vector**. Vector quantities for the display of field or flux lines in slice and surface plots.
- **Volume**. Scalar quantities for volume integrals over regions.
- **Surface**. Vector quantities for surface integrals over the boundaries of region sets.

Run speed is a critical issue in **MagView**. The generation of a single slice plot may require over 50,000 interpolations. To ensure fast operation, expressions for calculated quantities in the configuration file are written in reverse Polish notation (RPN). **MagView** parses the function strings once while loading the configuration file and saves them in a coded form. Thereafter, operations are performed at the speed of compiled code. The following section reviews the basics of RPN notation. Section 12.3 covers parameters and variables that may appear in expressions. Section 12.4 reviews the form of the data sections.

12.2 Introduction to RPN

Consider the expression

$$5.0 \times 4.5^3 + \left[9.2 + 0.6 \times 0.9^{(2+0.67)}\right]. \tag{41}$$

Equation 41 could be represented in algebraic notation by the string:

Table 14: General organization of the **MagView** configuration file

```
HEADER
  Control parameters (Required)
END
STOREDQUANT
  Stored quantities in the solution file (Required)
END
REGIONQUANT
  Physical properties of regions (Optional)
END
PROGPARAM
  Program parameters (Optional)
END
RUNPARAM
  Parameters set by the solution program (Optional)
END
REALTIMEPARAM
  Parameters set by the user (Optional)
END
INTERPOLATION
  Calculated quantities for plots and data output  (Required)
END
VECTOR
  Calculated quantities for vector display (Optional)
END
VOLUME
  Calculated quantities for volume integrals (Optional)
END
SURFACE
  Calculated quantities for surface integrals (Optional)
END
ENDFILE
```

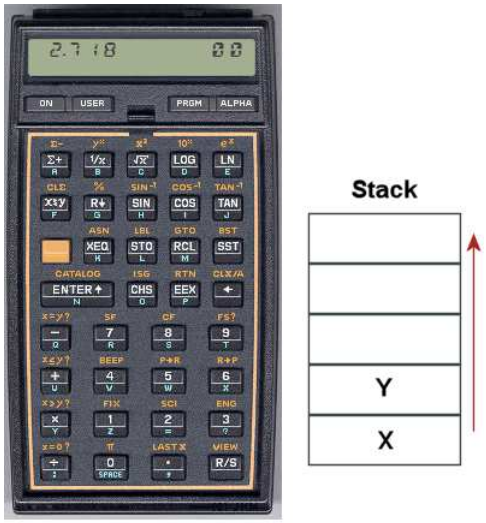


Figure 59:

```
5.0 * 4.5^3 + (9.2 + 0.6 * 0.9 ^ (2 + 0.67))
```

The rules for parsing such a string are involved, requiring an interpreter with recursive logic. The equation has the following form in RPN:

```
0.9 0.67 2 + ^ 0.6 * 9.2 + 4.5 3 ^ 5.0 * +
```

The string is parsed in strict order from left to right for expressions of any degree of complexity. A calculator with RPN logic uses the stack shown in Fig. 59. Numbers are pushed and popped at the bottom. The first two stack registers have the special names X and Y. Three simple rules govern the evaluation of RPN expressions:

- If the entry is a number, push it on the stack.
- If the entry is a unary operator (exp, ln, sin,...), apply it to the number in the X register.
- If the entry is a binary operator (+,*,^,...) combine the numbers in the X and Y registers and move the stack down.

The final result is given by the number remaining in the X register. By convention, the binary operators act in the following way:

- Addition(+): $Y + X$
- Subtraction (-): $Y - X$
- Multiplication (*): $Y \times X$
- Division (/): Y/X
- Exponentiation (^): Y^X

Reverse Polish notation eliminates all issues with regard to parsing order. Furthermore, it is simple to compile expressions. Entries are represented either by variables or operator codes.

12.3 Variables, operations and parameters

Expressions to define quantities in **MagView** may contain numbers, unary and binary operations, standard variables, standard calculated quantities and parameters. The standard variables are

```
$X, $Y, $Z, $NREG, $DUNIT
```

Note that the name of variables and parameters must start with a dollar sign. Calls to defined functions for plots, line scans and other operations are always made at a specific location in the solution space. **MagView** sets the current position [$X,$Y,$Z] before any calls are made. To illustrate, the following expression gives the distance from the origin in meters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT
```

The standard variable $DUNIT is a dimension conversion factor passed from the solution program. For example, suppose dimensions in **MetaMesh** were defined in centimeters and converted to meters for use in the solution program. In this case, an entry in the header of the solution file would set $DUNIT = 100.0. The following expression returns the distance from the origin in centimeters:

```
RDIST = $X 2 ^ $Y 2 ^ $Z ^ 2 + + @SQRT $DUNIT *
```

Unary operators have names that begin with '@'. Table 15 lists the available set. The following considerations apply with respect to the @CHS operator:

- You may include negative numbers (such as -3.1456 and -8.9E-09) in expressions. The number -22.56 is equivalent to 22.56 @CHS.
- Expressions like -$Mu0 and -&grady[3] that mix the binary operator with a variable are invalid. Instead, use forms like $Mu0 @CHS.

The following quantities represent values from the magnetostatic solution calculated at the current position:

```
&Bx
&By
&Bz
&BMag
&Hx
&Hy
&Hz
&HMag
&Q[N]
```

The names must begin with an ampersand. They are always followed by an integer in box brackets giving the number of the stored quantity. The quantity &Q[n] is the interpolated value of stored quantity n at the current position [$X,$Y,$Z]. **Magnum** includes the following stored quantities:

Table 15: Unary operators

Name	Operation	Comments
`@SIN`	$\sin(X)$	Angle in radians
`@COS`	$\cos(X)$	Angle in radians
`@TAN`	$\tan(X)$	Angle in radians
`@ASIN`	$\sin^{-1}(X)$	Returns angle in radians
`@ACOS`	$\cos^{-1}(X)$	Returns angle in radians
`@ATAN`	$\tan^{-1}(X)$	Returns angle in radians
`@EXP`	e^X	
`@LN`	$\ln X$	
`@LOG`	$\log_{10}(X)$	
`@SQRT`	$\sqrt{X}$	
`@ABS`	$\|X\|$	
`@EXCH`	$X \rightleftharpoons Y$	
`@OVERX`	$1/X$	
`@XSQ`	X^2	
`@ENTER`	$X \Rightarrow X, Y$	Copy X and push on stack
`@CHS`	$X = -X$	

- `&Q[1]` The reduced potential ϕ (node).
- `&Q[2]` The x component of applied magnetic field, H_{ax} (node).
- `&Q[3]` The y component of applied magnetic field, H_{ay} (node).
- `&Q[4]` The z component of applied magnetic field, H_{az} (node).
- `&Q[5]` The dual potential ψ (element).
- `&Q[6]` The relative magnetic permeability μ.

The program performs the interpolation and replaces the symbol with its value before evaluating the expression. The interpolation method depends on whether the stored quantity is of type *Node* or *Element*. **MagView** performs detailed interpolations of *Node* quantities. In the `LIN` mode, the program interpolates within the target element using shape functions and values of the stored quantity at the eight nodes. In the `LSQ` mode, **MagView** collects data from surrounding elements of the same region type and makes a least-squares fit to a three-dimensional, second-order function.

Magnum also transfers the following region variables:

```
$RegType
$HCVectX
$HCVectY
$HCVectZ
```

These variables assume values characteristic of the region at the current interpolation point. The variable $RegType identifies the material type and is not generally used to define interpolation quantities. The remaining variables are the vector components of the quantity $H_c = B_r/\mu_0$ for permanent magnets (in units of A/m). The values are zero in other types of materials. As an example, the following configuration file entry gives vectors plots of the direction of magnetization for permanent magnets:

```
VECTOR
  PMagDirect = $HCVectX;$HCVectY;$HCVectZ
END
```

Expressions may also include parameters defined in the PROGRARAM section. The section in the file MAGVIEW_STANDARD.CFG has this form:

```
PROGPARAM
  $Mu0 = 1.256637E-6
* 1/Mu0
  $IMu0 = 7.957748E5
* Origin for torque calculations, specify in meters
  $xt = 0.00
  $yt = 0.00
  $zt = 0.00
END
```

The following expression for the magnetic-field energy in the interpolation section uses a program parameter:

```
MagEnergy = 0.5 &BMag * &BMag * &Q[6] $Mu0 * /
```

You can define up to ten parameters in the PROGRARAM section. The name must begin with a dollar sign and may include up to 12 characters. A value in any valid real-number format follows the equal sign.

12.4 Setting up the calculated-quantity sections

The final sections of the configuration file define quantities for plots and calculations. There are four sections:

```
INTERPOLATION
  QuantityName = Definition
END
VECTOR
  QuantityName = Definition1;Definition2;Definition3
END
VOLUME
  QuantityName = Definition
END
SURFACE
  QuantityName = Definition1;Definition2;Definition3
END
```

The INTERPOLATION section may contain from zero to 50 quantity definitions. The VECTOR, VOLUME and SURFACE sections may contain a maximum of 20 quantity definitions each.

A definition line for a scalar quantity in the *Interpolation* section consists of a variable name, an equal sign and an RPN expression using the components discussed in the previous section. The following rules apply to quantity names in all sections:

- A name has a maximum length of 12 characters (the length is limited so that quantity names will fit in plots and formatted output lists).
- Names may include characters, numbers and the underscore symbol. Names may not include spaces and other punctuation marks.

The RPN expression is a string (maximum length of 300 characters) with entries separated by spaces. Here are some examples of quantity definitions:

```
EPERP = &EY @XSQ &EY @XSQ + @SQRT
LOGEXVCM = &EX 100.0 / @LOG
DX = &EX $MU0 * &Q[2] *
FIELDANGLE = &BY &BX / @ATAN $RADTODEG *
```

The last expression requires that the parameter

```
$RADTODEG = 57.2958
```

was defined in the PROGPARAM section. Note that the result of the division operator in the expression is 0.0 if &BX = 0.0.

The order in which quantities appear in the INTERPOLATION section affects some operations in **MagView**.

- The first listed quantity is the default choice for *Slice* plots You can change the choice by entering a value in the *SliceDisplay* command in the HEADER. The integer is the order of the default plot quantity as listed in the INTERPOLATION section.
- In dialogs for picking a plot quantity, the options are listed in the same order that they appear in the configuration file. Therefore, you should put important quantities near the beginning and specialized quantities near the end.
- The same order is followed in listings created by the *Matrix file* command. If you are exporting tables for use in **OmniTrak**, be sure that the first three quantities are B_x, B_y and B_z.

Each quantity in the *Vector* section requires three expressions for the x, y and z components. The format for a definition line is

```
Name = (Expression x);(Expression y);(Expression z)
```

The line may contain a maximum of 500 characters. The component expressions are separated by semicolons. The following example defines the applied magnetic field (i.e., field with no material contributions):

Table 16: Naming conventions in RPN expressions

Name form	Functions
$NAME	Program parameters
$NAME	Standard variables (*e.g.*, $X)
&NAME	Special quantities calculated from the solution (*e.g.*, &Ex)
@NAME	Unary operators in RPN expressions

```
HVectApp = &Q[2];&Q[3];&Q[4]
```

The volume integral of a quantity Q over a region N_R is given by

$$\int dx \int dy \int dz \; Q(x, y, z), \tag{42}$$

where only those elements with region number equal to N_R are included. You can define up to 20 quantities in the *Volume* section for integrals. The expressions should define quantities with dimensions of the form $(quant/m^3)$. Volume integrals use internal program coordinates in units of meters. Note that the quantity $DUNIT is used in **MagView** only for display labels.

The surface integral of a vector quantity **S** over a region or a set of regions is given by

$$\int \int dA \; \mathbf{S} \cdot \mathbf{n}, \tag{43}$$

where dA is an element of area on the surface of the region set and **n** is a unit vector normal to the surface pointing out of the region set. You can define up to 20 quantities in the SURFACE section for integrals. The data line for each quantity should contain the name, an equal sign, and three RPN expressions separated by semicolons for the vector components. The expressions should define quantities with dimensions of the form $(quant/m^2)$. As an example, a surface integral of the following quantity would give the magnetic flux through selected region surfaces:

```
MagFlux = &Bx;&By;&Bz
```

Note that quantity names must begin with special symbols if they are used in RPN expressions. Symbols for different quantity types are listed in Table 16. Therefore, quantity names in the PROGPARAM section must include the dollar sign. Quantity names that are defined only for display purposes need not include a special symbol. The includes quantities defined in the INTERPOLATION, VECTOR, VOLUME and SURFACE sections.

13 Force calculations with MagView

13.1 Surface integrals for force and torque

If we knew the distribution of current-density $\mathbf{j}(x, y, z)$ and magnetic flux density $\mathbf{B}(x, y, z)$ over an object, we could find the force on the object from the expression:

$$\mathbf{F} = \int\int\int dV\ \mathbf{j} \times \mathbf{B}. \tag{44}$$

Volume integrals are not practical for calculating forces on ferromagnetic materials permanent magnets and field-excluding objects. Here, the current-density distribution is concentrated in an infinitely thin surface layer (compared to the element size) and the magnetic flux density changes discontinuously at the surface. For this situation, the best approach is to perform an integral of the Maxwell stress tensor over a surface that encloses the body. This process yields good numerical accuracy and can be performed automatically using the surface-integral capabilities of **MagView**.

The reference S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1998), Sect. 10.5 discusses the physical basis of electromagnetic force calculations. The integral of force density over a volume can be converted to the following integral over the bounding surface to give the total force $\mathbf{F}$;

$$\mathbf{F} = \int\int \mathbf{S} \cdot \mathbf{n}\ dA. \tag{45}$$

where the quantity $\mathbf{S}$ is the tensor:

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}. \tag{46}$$

The integral of Eq. 45 can be implemented in **MagView** by adding the following vector definitions to the `SURFACE` section of the configuration file:

$$\begin{aligned} F_{xs} &= [S_{11}, S_{12}, S_{13}], \\ F_{ys} &= [S_{21}, S_{22}, S_{23}], \\ F_{zs} &= [S_{31}, S_{32}, S_{33}]. \end{aligned} \tag{47}$$

Expressions for torque are more complex. A differential element of torque is defined by

$$d\mathbf{t} = \mathbf{r} \times \mathbf{dF}. \tag{48}$$

Here, the vector $\mathbf{r}$ points from a torque origin $[x_t, y_t, z_t]$ to the current position:

$$\mathbf{r} = [r_x, r_y, r_z] = [(x - x_t), (y - y_t), (z - z_t)]. \tag{49}$$

The total torque corresponding to the force on a body may be written in terms of the Maxwell stress tensor as:

$$\mathbf{t} = \int\int \mathbf{r} \times (\mathbf{S} \cdot \mathbf{n})\, dA. \tag{50}$$

To add torque calculations in the **MagView** configuration file, we need to determine a torque tensor $\mathbf{T}$ such that the torque vector is given by a surface integral,

$$\mathbf{t} = \int\int \mathbf{T} \cdot \mathbf{n}\, dA. \tag{51}$$

We can compute $\mathbf{T}$ by expanding the right-hand side of Eq. 50 in component form and collecting terms with factors of n_x, n_y and n_z. The procedure gives the following results:

$$\mathbf{T} = \begin{bmatrix} r_y S_{31} - r_z S_{21} & r_y S_{32} - r_z S_{22} & r_y S_{33} - r_z S_{23} \\ r_z S_{11} - r_x S_{31} & r_z S_{12} - r_x S_{32} & r_z S_{13} - r_x S_{33} \\ r_x S_{21} - r_y S_{11} & r_x S_{22} - r_y S_{12} & r_x S_{23} - r_y S_{13} \end{bmatrix}. \tag{52}$$

13.2 Expressions for magnetostatic forces

The configuration file `MAGVIEW_STANDARD.CFG` includes definitions for force and torque calculations using surface integral methods. The Maxwell stress tensor has the following form for magnetostatic forces:

$$\mathbf{S} = \frac{1}{\mu_0} \begin{bmatrix} B_x^2 - B^2/2 & B_x B_y & B_x B_z \\ B_y B_x & B_y^2 - B^2/2 & B_y B_z \\ B_z B_x & B_z B_y & B_z^2 - B^2/2 \end{bmatrix}. \tag{53}$$

The quantity B^2 in Eq. 53 is the flux density magnitude:

$$B^2 = B_x^2 + B_y^2 + B_z^2. \tag{54}$$

Note that all field values are evaluated on the air/vacuum side of the object surface. Surface integrals of force are calculated with the vector expressions of Eq. 47 while torque components are determined using the expressions of Eq. 52.
The quantity B^2 in Eq. 53 is the field magnitude,

$$B^2 = B_x^2 + B_y^2 + B_z^2. \tag{55}$$

Figure 60 illustrates the logic of the calculation. If we have an assembly of ferromagnetic and permanent-magnet objects surrounded by an air volume, then we can take the integral of Eq. 45 over any enclosing surface. One choice is the outer boundary of the assembly components (designated surface A in Fig. 60). In this case, we evaluate magnetic field values in the air elements near the surface to ensure that the integral encloses all material currents $\mathbf{j}_m$. We could also define an arbitrary surface by enclosing the assembly inside a special diagnostic air region ($\mu = \mu_0$). The integral over the surface marked B in Fig. 60 gives the same result (to within the numerical accuracy of field interpolations). Figure 60*b* shows a case where the stress tensor integral does not give the correct result. In this case the integral extends over the surface of ferromagnetic region A that is in contract with another iron or permanent-magnet region (B).

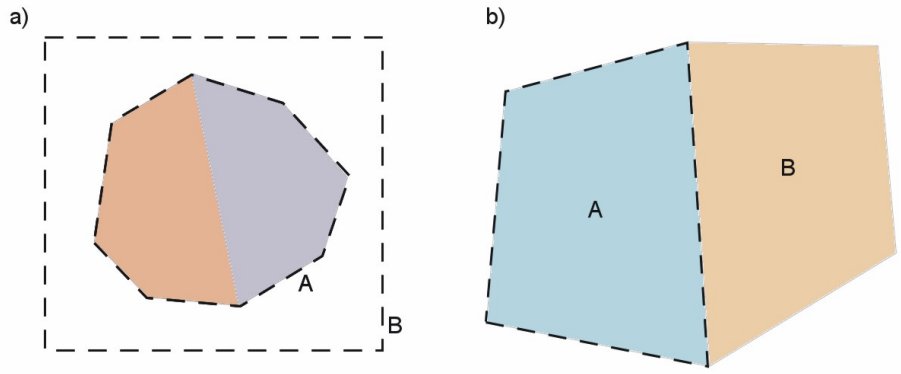


Figure 60: Surface integral of the magnetic stress tensor. *a*) Valid surfaces for an integral of magnetically-active materials. A: region surfaces bounded by air elements. B: surface of an air region that encloses the magnetic materials. *b*) Invalid surface for an integral. The enclosed material current is undefined on the common boundary.

The field values inside region B along the common boundary include the effects of the surface currents of both regions. The calculation gives the force on region A plus an indeterminate portion of the force on region B. In summary, the following rules apply to magnetic force calculations:

- Magnetic forces act on permanent magnets and objects with $\mu \neq \mu_0$.
- The surface integral of Eq. 45 over a region boundary gives the correct forces only if the region is surrounded by air elements ($\mu = \mu_0$).
- Any integration surface in air that encloses the region (or assembly of regions) yields the same net force.

13.3 Commands for force calculations

For a force calculation, use the configuration file `MAGVIEW_STANDARD.CFG`. Click on the command *Surface integral* of the *Analysis* menu. In the dialog, choose the region or set of regions of interest. Note that if one region is inside another, you must pick both regions so that the integral is performed only over the outer surface. Usually, all regions outside the *Internal* set should be marked as *External*. **MagView** employs the following procedure to find the force on a single region. The program locates all elements within the target region number and checks the six bounding facets. If the element on the opposite side of facet has a different region number, then the facet lies on the region surface. (Facets on the boundary of the solution volume or adjacent to an element with region number 0 are excluded.) For each facet, **MagView** performs a 3×3 Gaussian-quadrature integral using normal coordinates. At each point, the program calculates magnetic flux density values at a nearby location (x, y, z) in the adjacent element and determines the components of **S**. The program sets an error flag if any adjacent element has $\epsilon \neq \epsilon_0$. **MagView** constructs a unit vector normal to the facet that points out of the target region. The program then forms the dot product for Eq. 45. Each segment of the integral gives an increment of force **dF**. The sum of **dF** over the segments of each facet and over all facets of the region surface gives the total force. **MagView** can also find the net force on assemblies

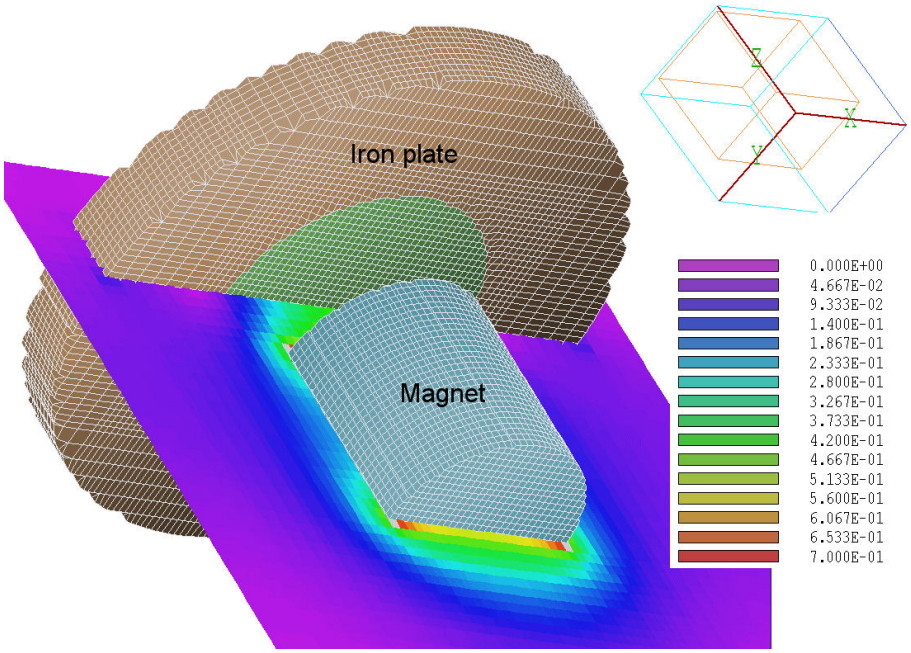


Figure 61: Geometry for the `PMAGFORCE` example.

of adjacent magnetically-active regions. In this case, the program identifies all elements of the region set. Surface integrals are performed only over facets that border on elements with region numbers that are not part of the set. If the assembly is surrounded by air elements, the procedure gives the total force. To calculate force from an analysis script, be sure that the proper configuration file is loaded and use the `SurfaceInt` command described in Sect. 8.3.

13.4 Examples

This section describes two simple examples of force calculations. The first (`PMAGFORCE`) illustrates techniques and numerical accuracy for a permanent-magnet simulation. The second (`RELAY`) demonstrates a torque calculation for a solenoid-driven relay.

Figure 61 shows the geometry for `PMAGFORCE`. A cylindrical permanent magnet (radius 3.0 cm, length 6.0 cm) is located a distance 2.0 cm from an iron plate (radius 8.0 cm, thickness 2.0 cm). The magnetization axis is along z with $B_r = 1.25$ tesla. The solution volume is large enough so that the boundaries make a small contribution to the net forces. We picked a system with approximate cylindrical symmetry for comparisons to calculations with the two-dimensional **PerMag** code. To check features of the force calculations, the iron plate is divided into two concentric regions and the permanent magnet is enclosed inside a box-shaped diagnostic region with $\mu_0 = 1.0$.

The files `PMAGFORCE.MIN` and `PMAGFORCE.GIN` define the solution. The mesh has an element size of 0.2 cm in regions of interest. Figure 62 shows the resulting variation of $|\mathbf{B}|$ in the plane $y = 0.0$. The **MagView** script listed in Table 17 performs several analysis functions. The first two calculations determine over two surfaces that enclose the permanent magnet. The result is $F_z = 143.2$ N for the surface of the permanent magnet (equivalent to the force to lift a 14 kg object) and $F_z = 148.7$ N for the diagnostic region. Theoretically, the force components F_x and F_y should equal zero. The numerical result is $|F_x, F_y| \cong (3 \times 10^{-5})\ |F_z|$. The integrals

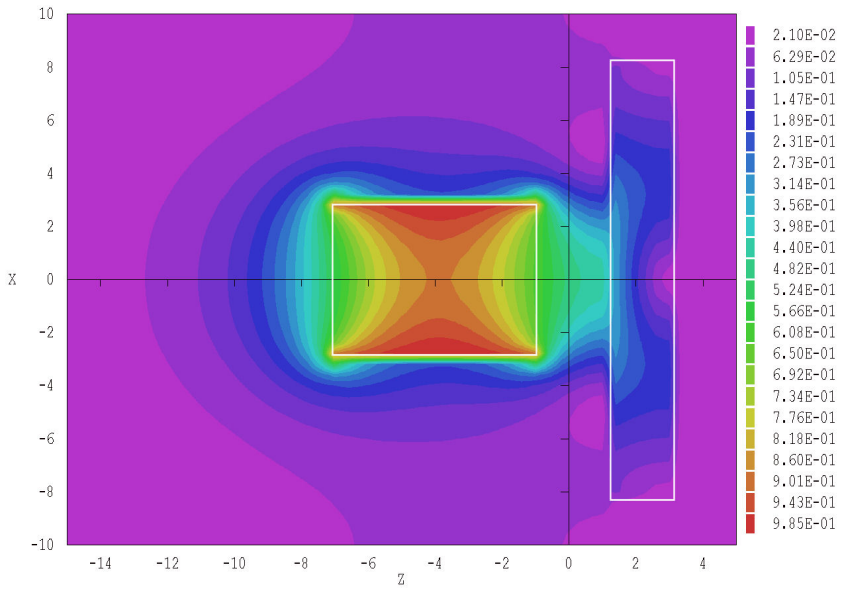


Figure 62: Example `PMAGFORCE`, variation of $|\mathbf{B}|$ in the plane $y = 0.0$.

of F_z over the two surfaces agree to within 1%. For comparison, the value determined by the two-dimensional **PerMag** code for a system with cylindrical boundaries is $F_z = 145.6$ N. The results imply that the surface-integral method of **Magnum** gives good values, even if the object has sharp edges. An integral of the magnetic stress tensor over the iron assembly gives $F_z = -143,34$ N, opposite and approximately equal to the force on the magnet.

In the second example we calculate the force and torque on the actuator of a magnetically-driven relay as a function of position (Fig. 63). The mesh (defined by the file `RELAY.MIN`) includes a cylindrical iron core (region 2) of diameter 1.0 cm and length 2.5 cm. The core is aligned along z and in wrapped in a solenoid coil defined by the file `RELAY.CDF`. The coil has inner radius 0.6 cm, outer radius 1.2 cm, length 2.0 cm and drive current 1000 A-turn. The magnetic field produces an attractive force on a square iron plate (region 3) with dimensions 1.0 cm × 1.0 cm × 0.3 cm. The plate is attached to a non-magnetic metal support that rotates about an axis parallel to x at position $y = 3.0$ cm, $z = 0.0$ cm. The goal is a find the force on the iron plate and the torque on the carrier assembly as a function of rotation angle. Figure 63 shows a rotation angle of 15°.

In creating the mesh, we could calculate the y and z displacements of the iron plate for each rotation angle. An easier approach is to employ the script-directive capability of **MetaMesh**. The part section for the iron plate has the following content:

```
#SHIFT 0.00  3.00  0.00
#ROTATE  15.00  0.00 0.00
Part
  Type Box
  Region Plate
  Fab   1.00  1.00  0.30
  Shift 0.00 -3.00 -0.15
  Surface Region Air
End
```

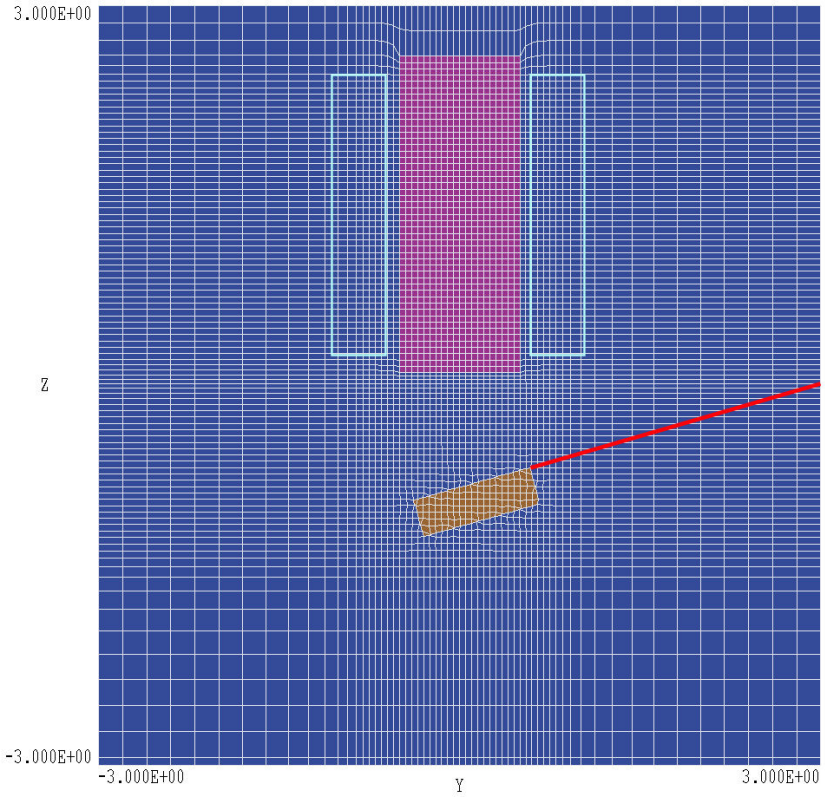


Figure 63: Geometry and mesh for example `RELAY` (view in the plane $x = 0.0$). Violet region: iron core. Light-blue outline: solenoid coil. Orange region: iron plate attached to structure (red line) hinged along an axis parallel to x at $y = 3.0, z = 0.0$.

Table 17: Contents of the **MagView** script file `PMAGFORCE.SCR`

```
*  NReg      RegName
*  ============================
*     1      VACUUM
*     2      DIAGREGION
*     3      IRONOUTSIDE
*     4      IRONINSIDE
*     5      MAGNET

INPUT PMagForce.GOU
OUTPUT PMagForce.DAT
* Integral over the outer surface of the diagnostic
* region that surrounds the magnet
SURFACEINT 2 5
* Integral over the magnet
SURFACEINT 5
* Integrals over the individual regions of the iron plate
SURFACEINT  3
SURFACEINT 4
* Integral over the surface of the iron assemble
SURFACEINT 3 4
ENDFILE



#ENDROTATE
#ENDSHIFT
```

The part is fabricated relative to an origin at $[0.0, -3.0, -0.15]$. A global rotation of 15^{o} about the x axis is applied and then a global shift of $+3.0$ cm along y. To change rotation angle, we simply change the first number in the command

```
#ROTATE  15.00  0.00 0.00
```

The program parameter of the configuration fill should include the entries:

```
$xt = 0.00
$yt = 0.03
$zt = 0.00
```

Table 18: `RELAY` example results (forces in newtons, torque in newton-m).

Angle	F_y	F_x	T_x
0.0^{o}	-1.451×10^{-5}	5.971×10^{-1}	-1.791×10^{-2}
5.0^{o}	-1.414×10^{-2}	1.209×10^{-1}	-3.574×10^{-3}
10.0^{o}	-8.649×10^{-3}	4.237×10^{-2}	-1.243×10^{-3}
15.0^{o}	-5.246×10^{-3}	1.711×10^{-2}	-5.064×10^{-4}

to set the torque origin. Note that the value of yt is given in meters. The **MagView** script `RELAY.SCR` contains the following command to perform the force/torque analysis:

```
SURFACEINT 3
```

Table 18 lists the results.

14 Inductance calculations with Magnum

14.1 Definitions – self and mutual inductance

This section describes how to use **Magnum** to calculate mutual inductances in multi-circuit systems. A common application is the determination of flux coupling in transformers. To begin, we need some definitions. A *circuit* is a contiguous wire that carries a current. In static-field calculations, the current has same value at all points in the wire. A circuit may be as simple as a wire loop or as complex as a toroidal winding with thousands of turns.

The energy method provides an easy way to find the inductance of a single-circuit system (*e.g.*, the drive coils of a magnet). You can find the total field energy by taking a volume integral in **MagView** over the quantity[1]

$$u = \frac{B^2}{2\mu_r\mu_0}. \tag{56}$$

The total field energy is related to the inductance of a single circuit (*self-inductance*) by:

$$U = \frac{L_{11}I_1^2}{2}. \tag{57}$$

The notation L_{11} denotes the inductance that results from magnetic flux coupling in Circuit 1 created by the current flowing in Circuit 1.

To address *mutual inductances*, we adopt material and notation from D.K. Cheng, **Field and Wave Electromagnetics** (Addison-Wesley, Reading, 1989), Sects. 6.11 and 6.12. Suppose we have two circuits and Circuit 1 has a drive current of I_1 A. The quantity Λ_{12} is the total magnetic flux enclosed by the turns of Circuit 2. If Circuit 2 has N_2 identical turns and Φ_{12} is the flux produced in one turn, then $\Lambda_{12} = N_2\Phi_{12}$. The *mutual inductance* between Circuits 1 and 2 is defined as:

$$L_{12} = \frac{\Lambda_{12}}{I_1}. \tag{58}$$

The quantity L_{12} (in units of henries) is the total magnetic flux in Circuit 2 per unit current in Circuit 1. According to Faraday's law, the voltage induced in Circuit 2 by a changing current in Circuit 1 is

$$V_{12} = L_{12}\frac{dI_1}{dt}. \tag{59}$$

Similarly, the *self-inductance* L_{11} is the total flux coupled to Circuit 1 per unit current in Circuit 1. To illustrate, consider the self-inductance of a tight winding of N turns on a toroidal ferrite with relative magnetic permeability μ_r. The torus has a rectangular cross section with inner radius r_o, outer radius r_i and height d. The magnetic field inside the ferrite created by the winding is approximately,

[1]Note that Eq. 56 holds only for linear materials where μ_r does not depend on the amplitudes of drive currents. This condition must also hold to define a unique circuit inductance.

$$B_\theta = \frac{\mu_r \mu_0 N_1 I_1}{2\pi r}. \tag{60}$$

Multiplying by d and taking an integral over r, the total flux linking the winding is:

$$\Lambda_{11} = \frac{\mu_r \mu_0 N_1^2 I_1 d}{2\pi} \ln\left(\frac{r_o}{r_i}\right). \tag{61}$$

Therefore, the circuit has self-inductance

$$L_{11} = \frac{\Lambda_{11}}{I_1} = \frac{\mu_r \mu_0 N_1^2 d}{2\pi} \ln\left(\frac{r_o}{r_i}\right). \tag{62}$$

14.2 Energy method for mutual inductance

In a three-dimensional magnetic field calculation, an attempt to evaluate mutual inductance by surface integrals of flux over every turn of a circuit would be laborious. The procedure would also be inaccurate if drive currents are represented by filamentary elements as in **Magnum**. A solution is to use indirect method based on volume integrals. The *field energy method* is a generalization of Eq. 57.

As a foundation, we will employ two results that are proved in the Cheng reference and many texts on introductory electromagnetism. First, mutual inductances have the following symmetry property:

$$L_{nm} = L_{mn}. \tag{63}$$

The second result is that the total field energy of a system of N circuits may be written as a sum of mutual inductances and drive currents:

$$U = \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} L_{mn} I_m I_n. \tag{64}$$

To illustrate the energy method, consider a two-circuit system like a transformer. We need three quantities to characterize the system inductance: L_{11}, L_{22} and L_{12} (which equals L_{21}). We shall excite the circuit with unit drive currents and make magnetic field energy calculations in **MagView**. The self-inductance of Circuit 1 can be determined by setting $I_1 = 1.0$ A and $I_2 = 0.0$ A. Designating the resulting field energy as U_{10}, the result is:

$$L_{11} = 2U_{10}. \tag{65}$$

A second calculation with $I_1 = 0.0$ A and $I_2 = 1.0$ yields U_{01} such that $L_{22} = 2U_{01}$. We can find the mutual inductance $M_{12} = L_{12} = L_{21}$ by a third calculation with $I_1 = 1.0$ A and $I_2 = 1.0$. Using Eq. 64, the mutual inductance is related to the known self-inductances and the field energy U_{11} by:

$$M_{12} = \frac{1}{2}\left(2U_{11} - L_{11} - L_{22}\right). \tag{66}$$

Three **Magnum** calculations are necessary to find the three quantities.

Table 19: Transformer dimensions, `InductDemo` example.

Structure	r_i (cm)	r_o (cm)	d (cm)
Core	4.00	6.50	2.50
Primary	3.75	6.75	3.00
Secondary	3.25	7.25	4.00

The method may be generalized to any number of circuits. To illustrate, consider values of self and mutual inductance with three circuits. Consider Eq. 64 in maxtrix form:

$$U = \begin{bmatrix} L_{11}I_1^2 & L_{21}I_1I_2 & L_{31}I_1I_3 \\ L_{12}I_1I_2 & L_{22}I_2^2 & L_{32}I_2I_3 \\ L_{13}I_1I_3 & L_{23}I_2I_3 & L_{33}I_3^2 \end{bmatrix}$$

where the conditions $L_{12} = L_{21}$, $L_{13} = L_{31}$ and $L_{23} = L_{32}$ apply. We calculate six solutions by setting combinations of currents equal to 0.0 or 1.0 A. For example, setting $I_1 = I_3 = 0.0$ A and $I_2 = 1.0$ A gives $U_{010} = L_{22}/2$. The self inductances are therefore given by:

$$\begin{aligned} L_{11} &= 2U_{100}, \\ L_{22} &= 2U_{010}, \\ L_{33} &= 2U_{001}. \end{aligned}$$

Similarly, setting $I_1 = I_2 = 1.0$ A and $I_3 = 0.0$ A yields

$$U_{110} = \frac{L_{21} + L_{12} + L_{11} + L_{22}}{2} = L_{12} + \frac{L_{11} + L_{22}}{2}.$$

The equation yields values for L_{12} and L_{21}. In summary, the mutual inductance values are given by

$$\begin{aligned} L_{12} = L_{21} &= U_{110} - L_{11}/2 - L_{22}/2, \\ L_{13} = L_{31} &= U_{101} - L_{11}/2 - L_{33}/2, \\ L_{23} = L_{32} &= U_{011} - L_{22}/2 - L_{33}/2. \end{aligned}$$

14.3 Example – toroidal transformer

The example `InductDemo` illustrates the method for a two-circuit system. We shall consider a high-voltage transformer consisting of primary and secondary windings on a toroidal ferrite core. The simple 3D geometry allows a comparison of numerical results to theory. The input files `InductDemo.MIN`, `InductDemo.CDF` and `InductDemo.GIN` are supplied in the example library. The core and windings have square cross sections with dimensions listed in Table 19. In operation, the core has an average magnetic permeability of $\mu_r = 500$.

We use windings with a relatively large number of turns for a good comparison with theory. We shall use the same number of turns of the primary and secondary windings ($N_1 = N_2 = 72$) and then scale the results. Figure 64 shows set 8064 current elements created by **Magwinder** from the file `InductDemo.CDF`. We have the option to set the drive currents for the two windings

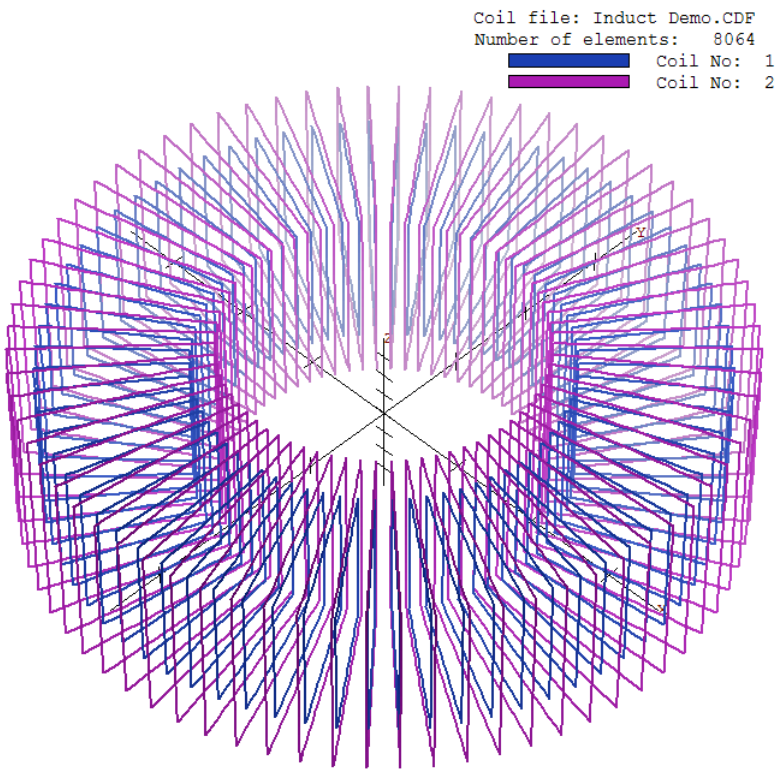


Figure 64: Primary and secondary windings created with **MagWinder** for the `InductDemo` example.

in the same or opposite directions. The choice determines the sign of M_{12}, but does not affect the magnitude.

The example employs symmetry conditions to minimize the run time. The solution includes one-eighth of the system volume over the space $x \geq 0.0, y \geq 0.0$ and $z \geq 0.0$ with an element size of 0.1 cm. The planes at $x = 0.0$ and $y = 0.0$ are set as symmetry boundaries (**B** normal to the plane), while the boundary at $z = 0.0$ has the natural boundary condition (**B** parallel to the plane). To find the self-inductance of the primary, we set $I_1 = 1.0$ A and $I_2 = 0.0$ A in the `CDF` file, generate a current-element set, carry out a field solution with **Magnum** and find the global field energy with **MagView**. The **Magnum** script `InductDemo.GIN` includes the following command:

```
CheckIron = OFF
```

The command was added after initial runs to deactivate the check for current elements that may pass through the core. The calculation is time-consuming for the large number of elements, so we can eliminate it when we are sure the calculation is running. The primary inductance was calculated by taking $I_1 = 1.0$ A and $I_2 = 0.0$ A. With no ferrite core, the field energy for one-eighth of the system was 1.1043×10^6 J. The total calculated field energy was 8.8344×10^{-6} J, giving $L_{11} = L = 1.7669 \times 10^{-5}$ H. For comparison, the inductance value predicted by Eq. 62 is 1.8283×10^{-5} H. When the high-μ core is present, it contains almost all the magnetic field energy. To estimate the primary inductance, we use $\mu_r = 500$ and the dimensions of the core in Eq. 62. The result is $L_{11} = 6.2922 \times 10^{-3}$ H, quite close to the value of $L_{11} = 6.2694 \times 10^{-3}$ H calculated with **Magnum**.

With $I_1 = 0.0$ A and $I_2 = 1.0$ A, the secondary inductance is $L_{22} = 3.2391 \times 10^{-5}$ H with no core and $L_{22} = 6.2796 \times 10^{-3}$ H with the ferrite present. As expected, $L_{11} \cong L_{22}$ with the core because the magnetic flux is concentrated in the ferrite. Applying Eq. 66, the calculated mutual inductance is $M_{12} = 6.2672 \times 10^{-3}$, equal to L_{11} and L_{22}. The result is as expected since the core gives almost perfect flux coupling between primary and secondary windings. The code values for an air-core transformer are $L_{11} = 1.7669 \times 10^{-5}$ H, $L_{22} = 3.2391 \times 10^{-5}$ H and $M_{12} = 1.8304 \times 10^{-5}$ H. If we put a voltage V_2 on the secondary, the predicted open-circuit voltage on the primary is given by $V_1/V_2 = M_{12}/L_{22} = 0.565$. This value is consistent with the ratio of the cross-section areas of the primary and secondary coils, $3^2/4^2 = 0.563$. The following scaling relationships may be used to find inductances for the actual number of primary and second windings:

$$L_{11} \rightarrow (N_1/72)^2 \; L_{11}, \tag{67}$$

$$L_{22} \rightarrow (N_2/72)^2 \; L_{22}, \tag{68}$$

$$M_{12} \rightarrow (N_1/72)(N_2/72) M_{12}. \tag{69}$$

15 Magnum file formats

Magnum creates output files in binary or text format. The mode is controlled by the *Format* command (Sect. 4.1).

- The default binary mode should be used for most applications. It gives the minimum file size and preserves the full accuracy of the double precision numbers used in **Magnum**.
- Use the text mode to port information to your own programs or if you want to inspect results with an editor.

As in **MetaMesh**, nodes are referenced with the indices [I,J,K] where I (the index along the x axis) extends from 0 to I_{max}, J (y axis) from 0 to J_{max}, and K (z axis) from 0 to K_{max}. The number of elements is approximately equal to the number of nodes. A single element (in the direction of positive x, y and z) is associated with each node for storage. Nodes and elements have integer region numbers to associate them with structures in the solution space.

The format of the **Magnum** binary output file is simple and compact. The FORTRAN code extract shown in Table 20 comprises the entire output algorithm. The header contains the following elements:

- The 6-character string *BINARY*.
- The number of regions in the solution space, $NReg$ (4-byte integer).
- A total of $NReg$ 20-character strings giving the names of regions obtained from the **MetaMesh** input file.
- A total of $NReg$ single-character strings, either I or E. The character I designates a standard material region that should be included in a MagView analysis and E indicates a fixed-potential that should be excluded.
- The mesh size parameters I_{max}, J_{max} and K_{max} (4-byte integers).
- The quantity $DUnit$, the unit conversion factor associated with the solution (double precision, 8-byte real).

The next step is to record information for each node. The quantities *RegNo* (the region number of the node) and *RegUp* (the region number of the adjacent element in the direction of increasing I, J and K) are 4-byte integers. The coordinates X,Y and Z (in meters) are 8-byte real numbers as are the following stored quantities:

Table 20: Code to create the **Magnum** binary output file

```
! File type
    Character6 = 'BINARY'
    WRITE (OutField) Character6
! NReg
    WRITE (OutField) NRegMax
! Region names
    DO N=1,NRegMax
      WRITE (OutField) Reg(N).RName
    END DO
! Region status
    DO N=1,NRegMax
      IF (Reg(N).Fixed) THEN
        Character1 = 'E'
      ELSE
        Character1 = 'I'
      ENDIF
      WRITE (OutField) Character1
    END DO
! Mesh size
    WRITE (OutField) IMax,JMax,KMax
! DUnit
    WRITE (OutField) DUnit
! Node quantities
    DO K=0,KMax
      DO J=0,JMax
        DO I=0,IMax
          MC = M(I,J,K)
          WRITE (OutField) &
          C(MC).RegNo,C(MC).RegUp,C(MC).x,C(MC).y,C(MC).z, &
          C(MC).Phi,C(MC).H.x,C(MC).H.y,C(MC).H.z,C(MC).Psi, &
          C(MC).MuUp
        END DO
      END DO
    END DO
  ENDIF
! Region quantities
    DO N=1,NRegMax
      WRITE (OutField) Reg(N).Type,Reg(N).HcVect.x, &
        Reg(N).HcVect.y,Reg(N).HcVect.z
    END DO
```

- The reduced potential ϕ at the node in units of A.
- The components of the applied magnetic field H_{ax}, H_{ax}, H_{ax} at the node in A/m.
- The dual potential ψ at the node in units of A.
- The relative magnetic permeability μ_r associated with the upper element.

Table 21 shows the header and initial node data lines from an output file in text format (note that the node data lines have been broken for readability). Five region names are shown in the example. The names are strings up to 20 characters in length with no spaces. Each name is followed by the character I (included) or E (excluded). The order of nodes is the same as that of Table tab:outputcode. Real numbers are recorded in format `(1P,E14.6)`.

Files created in response to the **MagView** *Matrix file* are in text format and have a structure similar to that of Table 21. The files are useful to transfer information to your own analysis programs and they are compatible with **OmniTrak** and **FEVision**. There are two differences from the `GOU` file in text format. First, calculations are performed over a regular mesh at locations

$$X(I) = X_{min} + I\Delta x, \tag{70}$$
$$Y(J) = Y_{min} + J\Delta y, \tag{71}$$
$$Z(K) = Z_{min} + K\Delta z. \tag{72}$$

where

$$\Delta x = (X_{max} - X_{min})/I_{max}, \tag{73}$$
$$\Delta y = (Y_{max} - Y_{min})/J_{max}, \tag{74}$$
$$\Delta z = (Z_{max} - Z_{min})/K_{max}. \tag{75}$$

Second, the set of calculated quantities depends on the choice in the the *Recorded quantities* command. The set is $[B_x, B_y, B_z]$ for the *Field components* option, where the magnetic flux density is in tesla. The quantities included under the *Full set* option depend on the **MagView** configuration file. The following quantities are included when operating under `magview_standard.cfg`: $B_x, B_y, B_z, |\mathbf{B}|, H_x, H_y, H_z.|\mathbf{H}|, \mu_r$ and $\gamma = 1/\mu_r$. The units are tesla for magnetic flux density and A/m for magnetic field.

Table 21: Header and data extract showing the form of the **Magnum** `GOU` file under the `Text` option and the **MagView** matrix file

```
TEXT
NReg:   5
AIR I
BOTTOMMAGNET I
MIDDLEMAGNET I
TOPMAGNET I
SYMBOUND E
IMax:     78
JMax:     78
KMax:     60
DUnit:  1.000000E+02

 RegNo RegUp       X             Y             Z            Phi
=====================================================================
     5     1  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
     5     1  1.250000E-03  0.000000E+00  0.000000E+00  0.000000E+00
     5     1  2.500000E-03  0.000000E+00  0.000000E+00  0.000000E+00
     5     1  3.750000E-03  0.000000E+00  0.000000E+00  0.000000E+00
     5     1  5.000000E-03  0.000000E+00  0.000000E+00  0.000000E+00
     5     1  6.250000E-03  0.000000E+00  0.000000E+00  0.000000E+00
  ...

          Hx            Hy            Hz           Psi           Mu
   ===================================================================
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  1.000000E+00
   ...
```

Index