



PerMag 8.0
Finite-element Magnetostatics
with Permanent Magnets

PO Box 13595, Albuquerque, NM 87192 U.S.A.
Telephone: +1-505-220-3975
Fax: +1-617-752-9077
E mail: techinfo@fieldp.com
Internet: <http://www.fieldp.com>

Contents

1	Introduction	4
1.1	Program functions	4
1.2	Learning PerMag	5
1.3	Finite-element solution procedure	6
1.4	Scripts and data files	8
2	Elements of magnetostatic theory	9
2.1	Field equations and boundary conditions	9
2.2	Permanent magnet properties	12
3	Creating and analyzing solutions with PerMag	16
3.1	Mesh generation	16
3.2	Creating the PerMag script	17
3.3	Finite-element solution	20
3.4	Solution analysis	20
3.5	Permanent magnet calculation	21
4	PerMag solution reference	26
4.1	PerMag script format	26
4.2	Program control commands	28
4.3	Commands for simple material properties	30
4.4	Nonlinear magnetic materials	31
4.5	Permanent magnet materials	33
4.6	Boundary and superposition commands	34
4.7	Running PerMag	37
4.8	Field energy listings	39
4.9	Computational utilities	40
4.10	Format of the PerMag output file	43
5	TriComp analysis functions	46
5.1	File menu commands	46
5.2	Plot menu commands	48
5.3	Saving and loading views	51
5.4	Analysis menu commands	52
5.5	Scan plot menu	55
5.6	Vector tools	55
5.7	Analysis script commands	57
6	Standard PerMag calculated quantities	61
6.1	PerMag variables	62
6.2	Standard configuration	63
6.3	Magnetic force calculations	65

7	Building custom analysis configurations	70
7.1	Configuration file structure	70
7.2	Introduction to RPN	72
7.3	Operators, parameters and variables	73
7.4	Building expressions	75

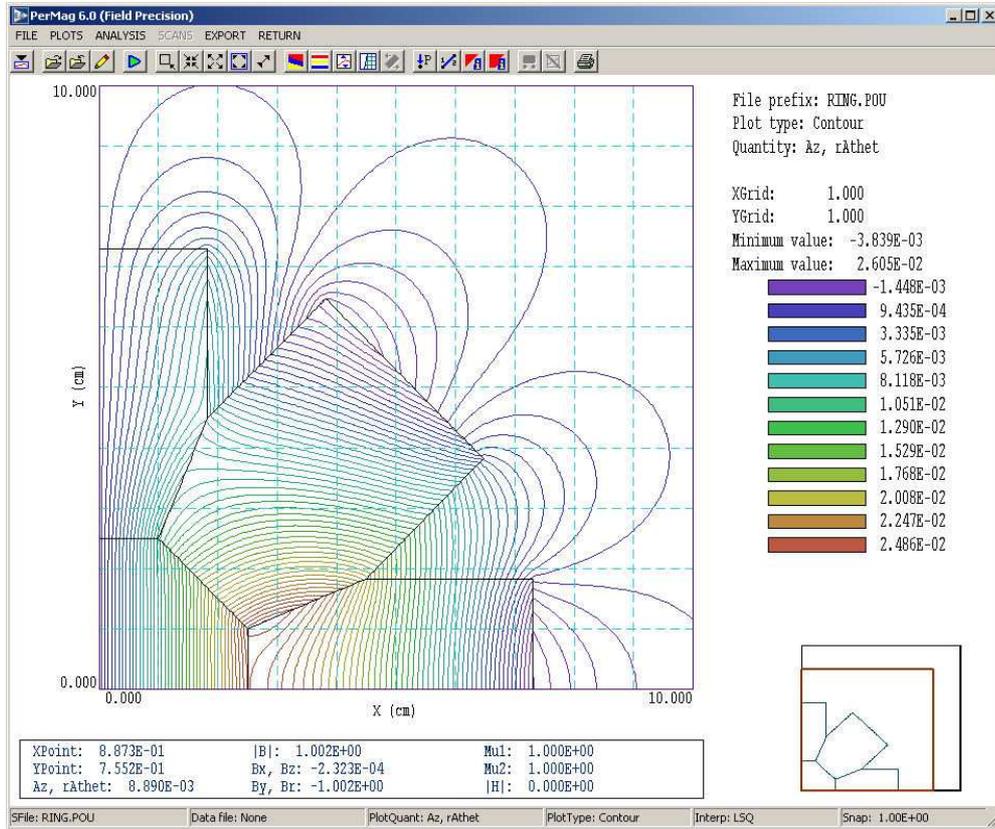


Figure 1: Screenshot – **PerMag** in the analysis mode.

1 Introduction

1.1 Program functions

PerMag is a versatile numerical tool for the design of electromagnets and permanent magnet devices. The program calculates magnetostatic fields in complex geometries with coils, linear or non-linear ferromagnetic materials, anisotropic materials and permanent magnets. The unitized, self-contained package addresses all aspects of the problem: mesh generation, finite-element solution, analysis and plotting. **PerMag** employs finite-element methods on variable-resolution conformal triangular meshes for high accuracy and speed. The mesh size is limited only by the installed memory on your computer. The program handles three-dimensional cylindrical problems (symmetry in θ) and two-dimensional rectangular problems (arbitrary variations in x and y with infinite extent in z).

The intuitive graphical-user-interface makes it easy to learn **PerMag** and to perform quick application setups. In contrast to other field solution software **PerMag** features complete data transparency and advanced technical functions. Input operations are automatically recorded in text scripts that provide documentation of your work. Scripts make it easy to reconstruct solutions and to share setups with colleagues. Advanced capabilities include the following:

- Support for the full spectrum of permanent magnet materials including those with non-linear demagnetization curves (such as Alnico and ceramics).
- Representation of anisotropic materials with the option for non-linear permeability variations along each axis.
- Calculation of field energy and energy-product arranged by region.
- Automatic determination of stored energy, peak field values, magnetic forces and torques.

Note that **PerMag** solutions are limited to static fields. The **TriComp** solution program **Nelson** finds solutions with AC magnetic fields and eddy currents. The **Pulse** program solves the general problem of magnetic field diffusion into resistive materials with arbitrary temporal variations of coil current.

1.2 Learning PerMag

The size of this manual reflects the extensive capabilities of the **PerMag** package. It is not necessary to read the entire document to address most applications. We have organized the chapters to help you get started quickly. You can investigate advanced topics as you gain more experience.

- Browse the next chapter if you would like to review some basics of permanent magnet theory. You will probably return to the material to resolve issues (such as boundary conditions) as you get more involved in the programs.
- Scan Chap. 1 of the **Mesh** manual to understand **TC**, the **TriComp** program launcher.
- Be sure to read Chapter 2 of the **Mesh** manual, which reviews essential concepts for conformal meshes in finite-element solutions.
- Chapter 3 of the **Mesh** manual follows a walkthrough example to introduce construction of a solution geometry with the interactive **Mesh** drawing editor. The exercise introduces fundamental tools you will need for your own simulations. Chapter 5 of the **Mesh** manual describes how to convert a script into a conformal mesh.
- Read Chapter 3. The walkthrough example covers the steps in creating and analyzing a finite-element solution with **PerMag**.
- Run some of the prepared examples supplied with the package. The examples give insights into solution options and may serve as templates for your own work.
- As you gain experience you will want to take advantage of the full range of **Mesh/PerMag** capabilities. Chapter 4 of the **Mesh** manual is a comprehensive reference on the drawing editor. Chapter 5 of the **Mesh** manual covers processing, plotting and repairs of meshes. Chapters 4, 5 and 6 summarize the solution and analysis functions of **PerMag**. Chapter 7 describes how to create custom analysis configurations.

- Additional chapters cover advanced techniques in **Mesh**. Chapters 6 and 7 of the **Mesh** manual show how to make direct entries in scripts to invoke advanced control features. Chapter 8 describes how to create meshes directly from photographic and data images. This feature is useful to model complex or irregular systems that are difficult to describe with simple geometric specifications.

1.3 Finite-element solution procedure

An in-depth understanding of finite-element numerical methods is not necessary to use **PerMag**. Nonetheless, it is important to have a clear idea of fundamental concepts to create effective solutions. This section describes background material to understand the steps in a **PerMag** solution.

The term *field* indicates a quantity (scalar or vector) defined over a region of space. Examples of fields include the vector magnetic induction \mathbf{B} in a magnetostatic solution, electric and magnetic fields in an electromagnetic solution and the scalar temperature T in a thermal solution. Variations of field quantities are usually described by continuous partial differential equations, such as the Poisson equation (Sect. 2). These equations can be solved directly by analytic methods if the system geometry and material properties are simple (for example, ferrite layers with uniform μ_r between concentric cylinders). Analytic solutions are extremely difficult in systems with asymmetric structures or non-linear materials. Furthermore, closed-form results are often expressed in terms of series expansions that must be evaluated numerically. For all but the simplest problems, it is usually quicker and more accurate to employ a direct numerical approach.

The fundamental issue in numerical field solutions is that digital computers cannot directly solve continuous equations. On the other hand, computers are well suited to solving large sets of coupled linear equations. The goal of all numerical field methods (finite-difference, finite-element or boundary-element) is to convert the governing differential equations into a set of coupled linear equations. The solution of the linear equations approaches the results of the differential equation when the set becomes large.

The basis of the finite-element approach is to divide the full solution volume into a number of small volumes, or *elements*. Here, the term *small* indicates that element dimensions are much less than the scale length for variations of field quantities. The division of the volume is called the *computational mesh*. Figure 2 shows the type of mesh used for the two dimensional solutions of **PerMag**. The figure defines three terms that will be used throughout this manual:

- **Element**. Volume divisions of the system.
- **Nodes**. Points where elements intersect.
- **Facets**. Surfaces between two elements.

The elements in Fig. 2 have triangular cross-sections. In a planar solution, the cross-section lies in the x - y plane and the elements extend an infinite distance in the z direction. In a cylindrical solution, an element is a figure of revolution about the axis with a triangular cross-section in the z - r plane.

The mesh in Fig. 2 has the important property of *conformality*. The term means that the triangles have been specially shaped to conform to the boundaries between materials (*regions*).

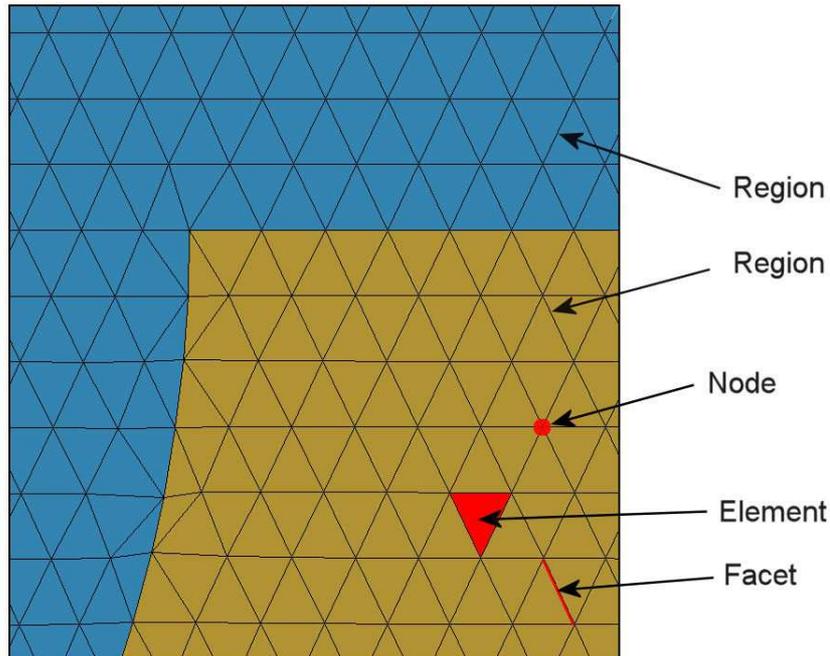


Figure 2: Conformal triangular mesh – definition of terms.

As a result, each element has an unambiguous material identity. The finite-element method is based on two approximations: 1) material properties in an element are uniform and 2) elements are small enough so that the field quantities can be approximated by simple interpolation functions. With these assumptions, the governing differential equation can be integrated over elements surrounding a node to yield a linear equation. This equation relates the field quantity at the node to those at the surrounding nodes. The coupled set has one linear equation for each node in the mesh. In **PerMag**, solution of the set representing the Poisson equation in a planar geometry gives the axial component of vector potential \mathbf{A}_z at each node. In a cylindrical simulation the program calculates the stream function, $\psi = r\mathbf{A}_\theta$. We can then perform two-dimensional interpolations to find values at intervening points, or take numerical spatial derivatives to find the components of the magnetic flux density.

With this background, we can understand the steps in an **PerMag** solution:

1. You define the boundaries of material objects in a solution volume for your application. The task is usually accomplished with the drawing editor of **Mesh**. You can also specify control information such as the target element sizes that may affect the accuracy and run time of the solution. The result are recorded in a text file (*script*) with sets of line and arc vectors that outline coils, magnetic materials or permanent magnets.
2. **Mesh** analyzes the boundary specifications and automatically generates a set of conformal triangles such as those of Fig. 2. The program creates an output file that lists the locations of nodes and the identities of elements.
3. You define the material properties of regions in the solution volume. Usually, this task is performed through an interactive dialog in **PerMag**.

Table 1: **PerMag** files

Name	Function
MName.MIN	Mesh input script (definition of foundation mesh and region outlines)
MName.MLS	Mesh diagnostic listing
MName.MOU	Mesh output (node locations and element identifies)
PName.PIN	PerMag input script (run control and material properties)
AName.DAT	Optional PerMag input tables to define properties of non-linear materials
PName.PLS	PerMag diagnostic listing
EName.POU	PerMag output file (node locations, vector potential and relative magnetic permeability)
AName.SCR	PerMag script for automatic data analysis

4. **PerMag** reads the mesh geometry and applies the material parameters to generate the linear equation set.
5. **PerMag** solves the coupled equations using iterative methods and records the node potentials and coordinates in an output file. This file serves as a permanent record of the solution that can be re-loaded for latter analysis.
6. You can use the interactive graphical environment of **PerMag** to explore the solution. The program creates a wide variety of plots and performs quantitative calculations of field magnetic quantities. You can also write scripts for automatic control of complex analyses.

1.4 Scripts and data files

Mesh and **PerMag** read and generate several types of files. For a typical solution it is not necessary for your to deal directly with them – the user interface takes care of data organization. On the other hand, if you generate a large amount of data it’s a good practice to archive application results in individual folders. Furthermore, there are advanced features of **Mesh** and **PerMag** that require direct entries to the input scripts. For convenience, all input scripts, data tables and output files are in text format. Both **Mesh** and **PerMag** feature integrated text editors. Table 1 lists the file types and functions in the **PerMag** package. Note that the suffixes indicate the file function.

2 Elements of magnetostatic theory

2.1 Field equations and boundary conditions

The text S. Humphries, **Field Solutions on Computers** (CRC Press, Boca Raton, 1997) gives detailed information on the theory of magnetostatics and the application of finite-element methods in **PerMag**. This section summarizes some basic concepts that may be helpful in preparing solutions.

Magnetic fields are created by current flow. We can divide currents into three classes: 1) drive, 2) material, and 3) permanent magnet. Drive currents flow in coils and have specified values independent of other solution objects. Material currents result from alignment of domains in soft magnetic materials in response to an applied field. Material currents are included in finite-element calculations through spatial variations of the relative magnetic permeability μ_r . Permanent magnet currents result from the locked alignment of domains in hard magnetic materials. These currents flow in the absence of an applied field.

Consider a region with planar two-dimensional symmetry. The only allowed components of drive current density and vector potential are j_z and A_z . The vector potential is related to spatial variations of the drive current density and magnetic permeability by the Poisson equation:

$$\nabla \cdot \left(\frac{1}{\mu_r} \right) \nabla A_z = -\mu_0 j_z. \quad (1)$$

The magnetic flux density can be determined from the vector potential by:

$$\mathbf{B} = \nabla \times A_z. \quad (2)$$

A finite-element treatment yields values of A_z at mesh nodes. The material properties μ_r and j_z are associated with elements.

We can write Eq. 1 in component form as:

$$\frac{\partial}{\partial x} \left(\frac{1}{\mu_r} \right) \frac{\partial A_z}{\partial x} + \frac{\partial}{\partial y} \left(\frac{1}{\mu_r} \right) \frac{\partial A_z}{\partial y} = -\mu_0 j_z. \quad (3)$$

The components of magnetic flux density are:

$$B_x = \frac{\partial A_z}{\partial y}, \quad B_y = -\frac{\partial A_z}{\partial x}. \quad (4)$$

There are two implications of Eq. 4 that we site without proof:

- Lines of \mathbf{B} lie along contours of A_z .
- The magnetic flux per unit length in z between two lines of \mathbf{B} is equal to difference in vector potential associated with the lines.

The factors form the basis for field line plots in **PerMag** and have important implications for boundary conditions in magnetic-field solutions.

In a cylindrical system the non-zero components of current density and vector potential point in the θ direction. The governing equation for vector potential is:

$$\frac{1}{r} \frac{\partial}{\partial r} r \left(\frac{1}{\mu_r} \right) \frac{\partial A_\theta}{\partial r} + \frac{\partial}{\partial z} \left(\frac{1}{\mu_r} \right) \frac{\partial A_\theta}{\partial z} = -\mu_0 j_\theta. \quad (5)$$

The magnetic flux-density components are related to the vector potential by:

$$B_r = -\frac{1}{r} \frac{\partial(rA_\theta)}{\partial z} = -\frac{1}{r} \frac{\partial\psi}{\partial z}, \quad (6)$$

$$B_z = \frac{1}{r} \frac{\partial(rA_\theta)}{\partial r} = \frac{1}{r} \frac{\partial\psi}{\partial r}. \quad (7)$$

The quantity $\psi = rA_\theta$ in Eq. 7 is called the *stream function*. The finite-element routines in **PerMag** calculate values of ψ at nodes. Cylindrical solutions have the following properties:

- Lines of \mathbf{B} lie along contours of ψ .
- The magnetic flux between two lines of \mathbf{B} is equal to $(1/2\pi)$ times the difference in ψ values associated with the lines.

With this background, we can understand of the two types of boundary conditions used in **PerMag**:

- *Dirichlet* boundary points have a fixed value of vector potential or stream function. Equations 4 and 7 imply that lines of \mathbf{B} are parallel to such the boundary. A line region of uniform vector potential or stream function represents a perfectly-conducting metal surface. An internal filled region at fixed vector potential can be used to represent a metal body that excludes the magnetic field.
- A *Neumann* boundary is one where the normal derivative of the vector potential or stream function is specified. The boundaries in **PerMag** are limited to the special cases $\partial A_z / \partial n = 0.0$ and $\partial \psi / \partial n = 0.0$. The condition implies that \mathbf{B} is normal to the boundary. One of the advantages of the finite-element method is that all non-fixed boundaries automatically satisfy the special Neumann condition, even if they are slanted or curved. Neumann boundaries are often used to reduce computation time for symmetric systems.

Because finite-element solutions apply to finite volumes, magnetic fields may not extend to infinity. We must choose the system geometry and boundary conditions to approximate free-space fields. We can take advantage of two facts:

- Magnetic field lines are circulatory – they must follow closed paths.
- Fields are usually confined to localized regions in magnetic circuits.

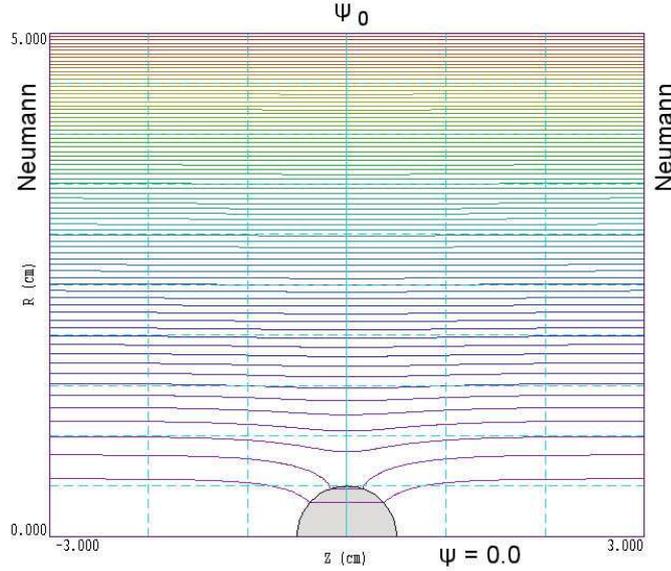


Figure 3: Solution to illustrate boundary conditions – iron sphere in a uniform axial magnetic field. Plot shows lines of \mathbf{B} separated by equal increments of magnetic flux.

The most common approach is set the Dirichlet condition $A_z = 0.0$ or $\psi = 0.0$ on the boundaries of the solution volume. In this case the boundaries act as perfectly-conducting walls, containing the flux. The fields approximate those in free space if there is a sufficient air gap around the components of the magnetic circuit. The Dirichlet condition is applied to all boundaries in the included examples except those that separate the parts of a symmetric system. Note that in cylindrical simulations, the condition $\psi = 0.0$ always holds along the axis. **PerMag** automatically sets the condition by adding a fixed-potential line region when $r_{min} = 0.0$.

The example of Fig. 3 illustrates an advanced application of boundary conditions. A iron sphere of radius R_s with $\mu_r \gg 1.0$ is immersed in a uniform field $B_z = B_0 = 0.1$ tesla. According to theory, the field everywhere inside the sphere has the value:

$$\mathbf{B} = \frac{3\mu_r}{\mu_r + 2} B_0 \mathbf{z}. \quad (8)$$

We can create a uniform field solution in the cylindrical region of Fig. 3 by setting a fixed value of stream function ψ_0 on the outer boundary at R_b . The left and right-hand boundaries have the default Neumann condition which ensures that field lines are normal to the solution volume. Noting that $\psi = 0.0$ at $r = 0.0$, the boundary value of stream function is related to the desired field by:

$$\psi_0 = \frac{B_0 R_b^2}{2}. \quad (9)$$

We choose a boundary radius $R_b \gg R_s$ so that the sphere makes a small perturbation. For $R_b = 5.0$ cm, we find that $\psi_0 = 1.25 \times 10^{-4}$ tesla-m. With $\mu_r = 250$, the predicted internal field from Eq. 8 is 0.2976 tesla. Figure 3 shows lines of \mathbf{B} calculated by **PerMag**. The code values of external and internal field are 0.0997 tesla and 0.2972 tesla.

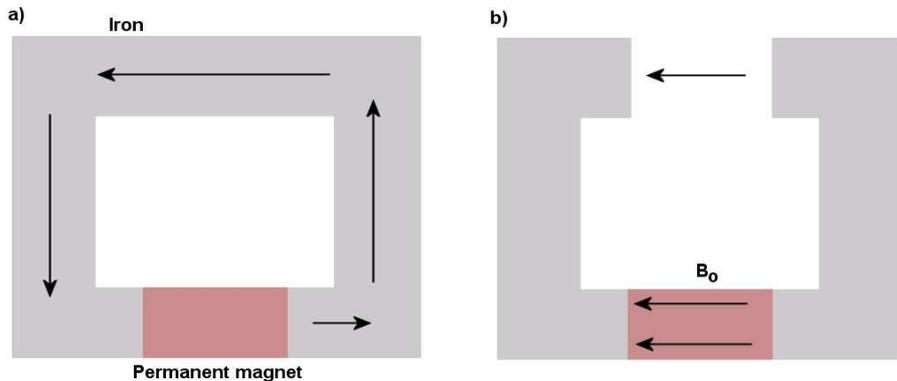


Figure 4: Permanent magnet in a magnetic circuit. *a)* Zero-reluctance circuit, all flux circulates through the iron. *b)* With an air gap, some of the flux is forced to return through the permanent magnet.

2.2 Permanent magnet properties

This section gives a brief review of permanent magnets and the quantities used to characterize their properties. Figure 4*a* shows a test setup with a permanent magnet in a zero-reluctance circuit. All magnetic flux produced by the aligned domains of the permanent magnet returns through the high- μ iron structure. The total magnetic flux density \mathbf{B} inside the magnetic material equals the flux density generated by the aligned domains. This value is called the *remanence magnetic flux density*, B_r . Suppose we add an air gap as shown in Fig. 4*b*. With the added reluctance, some of the magnetic flux is forced to return through and around the permanent magnet. We shall denote the component of return magnetic flux density parallel to the direction of magnetization as B_0 . The magnitude of B_0 rises with increasing air-gap width. The return magnetic flux depends on the geometry of the magnetic circuit and generally varies over the volume of the permanent magnet.

The *demagnetization curve* is a plot of the total magnetic flux density inside the magnet as a function of B_0 . (Note: curves for commercial materials are usually plotted in terms of the magnetic field intensity, $H_0 = B_0/\mu_0$). In an ideal material, domains are tightly locked so that their alignment is unaffected by the return flux. In this case, the total magnetic flux density inside the magnet is given by:

$$B = B_r - B_0. \quad (10)$$

The ideal curve is therefore a line with slope -45° (Fig. 5). The total magnetic flux density inside the magnet drops to zero when $B_0 = -B_c = -B_r$. The quantity $H_c = B_c/\mu_0$ is called the *coercive field*. Modern magnet materials like neodymium iron and samarium cobalt approach ideal behavior.

The return flux may reduce the degree of domain alignment in conventional materials like alnico. In this case, the magnet contribution to the total flux density is less than B_r and the demagnetization curve drops below the 45° line (Fig. 5). For these materials the demagnetization curve is non-linear and the condition $B_c < B_r$ holds. **PerMag** accurately represents non-linear materials if you enter the non-linear demagnetization curve as a table (Sect.4.5). Figure 6 shows some examples of commercial demagnetization curves for non-linear materials.

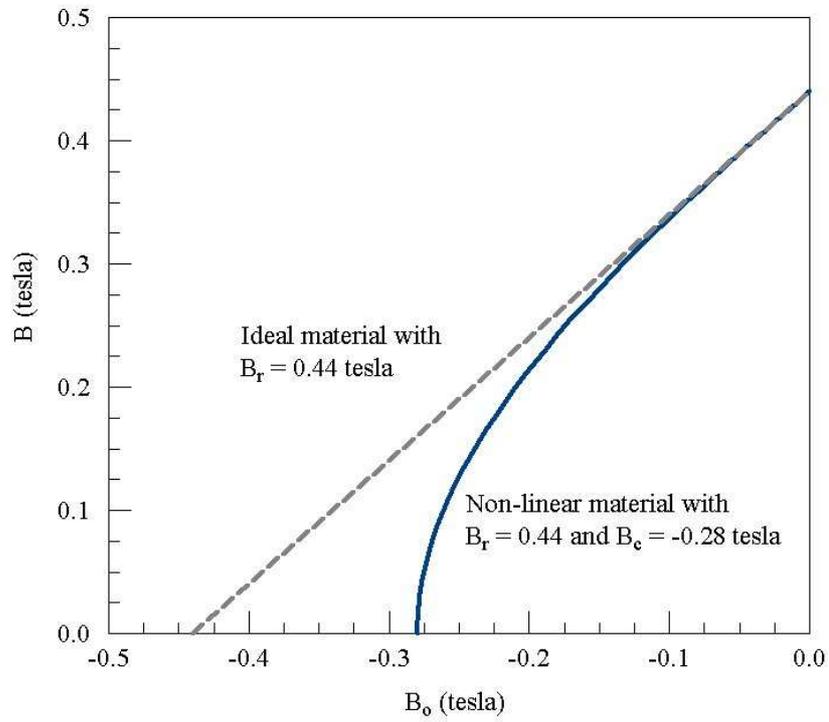


Figure 5: Demagnetization curves for ideal and non-linear permanent magnets.

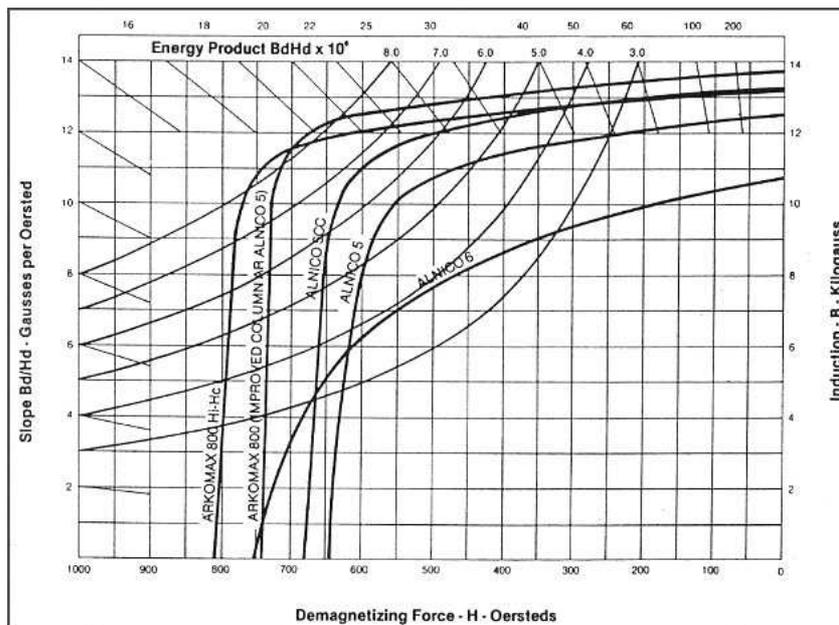


Figure 6: Demagnetization curves for several non-linear permanent magnet materials (Courtesy of Arnold Engineering).

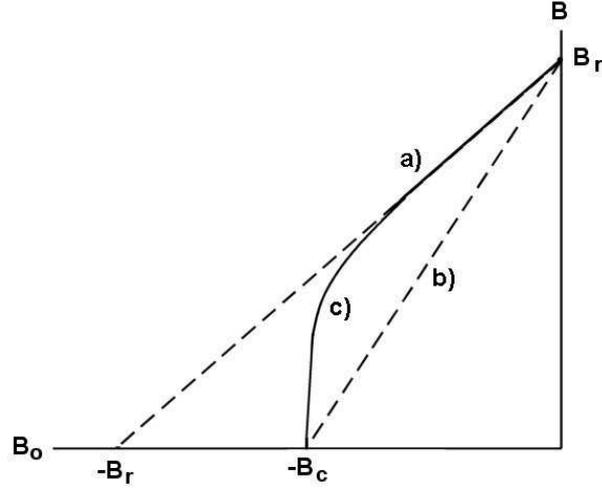


Figure 7: Non-linear material with an unknown demagnetization curve. *a)* Ideal material. *b)* Straight line between $-B_c$ and B_r . *c)* Approximation used in the **PerMag** two-point model.

A practical problem working with non-linear magnetic materials is that the full demagnetization curves are often difficult to find. The manufacturer may provide only values for B_r and H_c . If you simply model the curve as a straight line between B_r and B_c (Fig. 7*b*), the predicted flux values will be quite low. For example, the magnet of Figure 8 uses BEC material Y-38 with $B_r = 0.44$ tesla and $B_c = 0.28$ tesla and produces a measured gap flux density of 1.22 tesla. Approximating the demagnetization curve as a straight line from B_r to B_c in **PerMag** gives a predicted gap field of only 0.64 tesla. Figure 7*c* illustrates a better approach. The approximating demagnetization curve defined by the following equation passes through the points $(-B_c, 0/0)$ and $(0/0, B_r)$ with infinite slope at $(-B_c, 0.0)$ and 45° slope at $(0.0, B_r)$:

$$B_0 = -B_r \left[\left(1 - \frac{B}{B_r}\right) - \left(1 - \frac{B_c}{B_r}\right) \left(1 - \frac{B}{B_r}\right)^{\left[\frac{1}{1-B_c/B_r}\right]} \right]. \quad (11)$$

In the benchmark example, the calculation with the two-point curve gives 1.27 tesla for the gap flux density.

Table 2: Properties of some permanent magnet materials

Material	B_r (tesla)	$\mu_0 H_c$ (tesla)	Material	B_r (tesla)	$\mu_0 H_c$ (tesla)
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 Fine 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

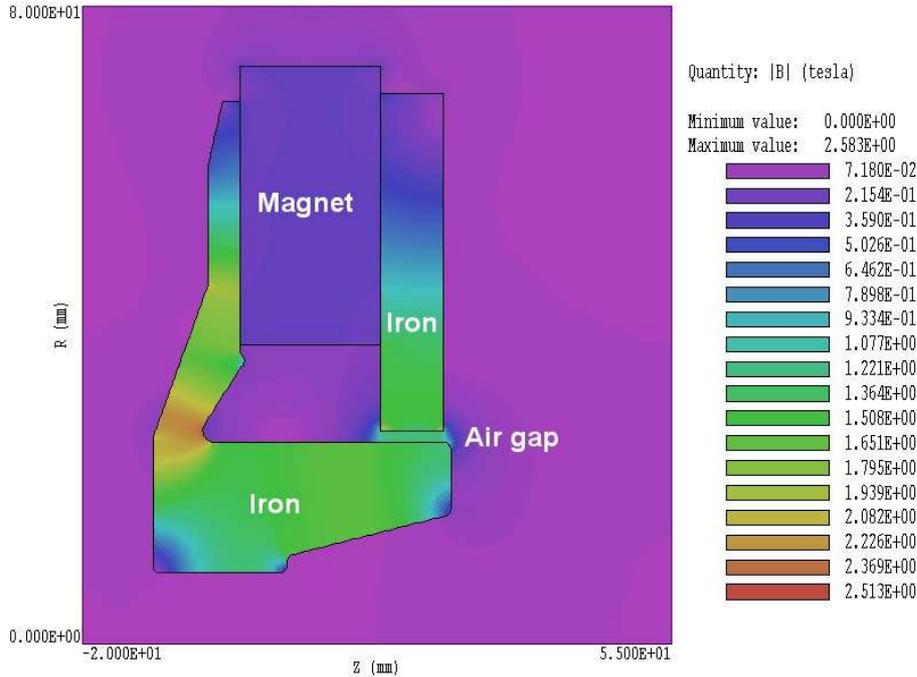


Figure 8: Benchmark calculation - non-linear material with an unknown demagnetization curve.

3 Creating and analyzing solutions with PerMag

3.1 Mesh generation

As a quick introduction to **PerMag** techniques and capabilities, we shall work through two examples. The first demonstrates a solution with drive coils, non-linear materials and symmetry boundaries. Figure 9 shows the geometry, an H-magnet. The configuration is commonly used in particle accelerators to create dipole fields. The length of the magnet is usually much larger than the transverse dimensions; therefore, the two-dimensional planar approximation is valid in regions distant from the ends. Only one-quarter of the magnet cross section is shown in Fig. 9. By symmetry, lines of magnetic flux density are vertical at the magnet midplanes in x and y . Therefore, we can eliminate the left-hand portion of the magnet and apply the Dirichlet condition $A_z = 0.0$ (Sect. 2.1) along the line $x = 0.0$. The magnetic flux density is normal to the line $y = 0.0$ so we can eliminate the lower half of the magnet and apply a Neumann condition along the bottom boundary. We also employ the Dirichlet condition along the top and right-hand boundaries, equivalent to the assumption that all magnetic flux is contained within the iron core. In the case of a highly-saturated core with significant leakage flux, it may be necessary to include an air volume around the outside of the solution volume.

To start, copy the files `HMAG.MIN`, `HMAG.SCR` and `SOFTIRON.DAT` to a convenient working directory. Run **TC** and make sure that the *Data folder* is set correctly. Run **Mesh** and load `HMAG.MIN`. Pick the *Edit script/graphics* command to enter the drawing editor. Here, you can use the display capabilities to confirm the following region assignments:

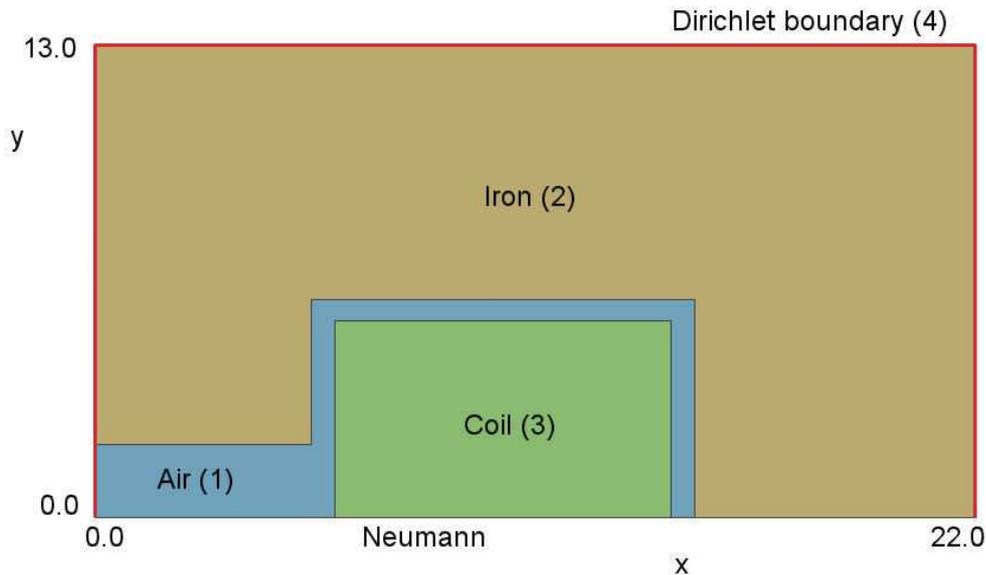


Figure 9: Geometry for the H magnet calculation. Dimensions in cm.

- **Region 1.** Air gap.
- **Region 2.** One-quarter of the iron flux return structure.
- **Region 3.** The top half of the right-hand side of the magnet winding. The current in the winding is directed into the page. The Dirichlet symmetry condition means that there is a symmetric virtual coil on the left-hand side that carries current out of the page.
- **Region 4.** An open region consisting of nodes around the left-hand, top and right-hand boundaries with the assigned condition $A_z = 0.0$.

Note that the bottom boundary automatically assumes the special Neumann condition so that no specification is required. Abandon the drawing and return to the main menu. Click the *Process* command. The completed mesh is shown in Fig. 10. Choose the *Save mesh (MOU)* command. You can now close or minimize **Mesh**.

3.2 Creating the PerMag script

Run **PerMag** from **TC**. The screen is initially blank and the status bar indicates that the program is waiting for input. Note the prominent tools marked *1*, *2* and *3*. The notation is a reminder of the three steps in a magnetostatic solution:

1. Set up program controls and material properties.
2. Generate and solve the finite-element equations.
3. Analyze the solution.

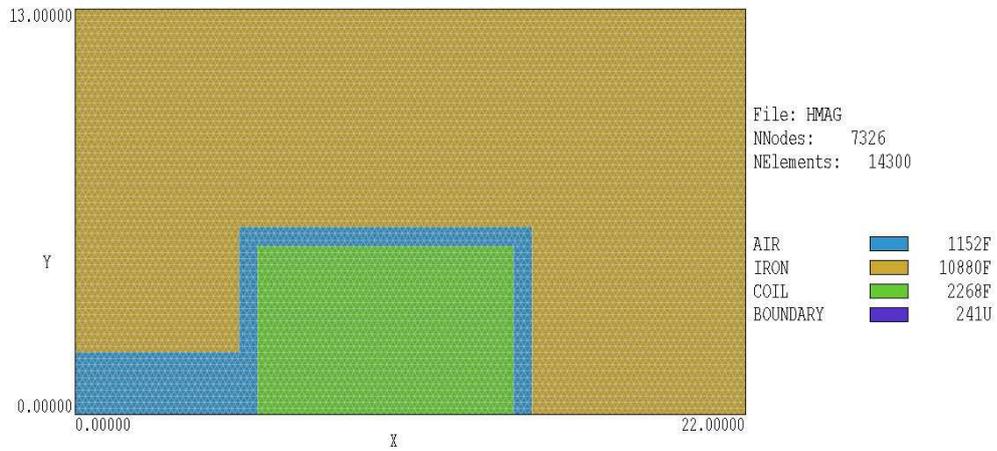


Figure 10: Mesh for the H magnet calculation. Dimensions in cm.

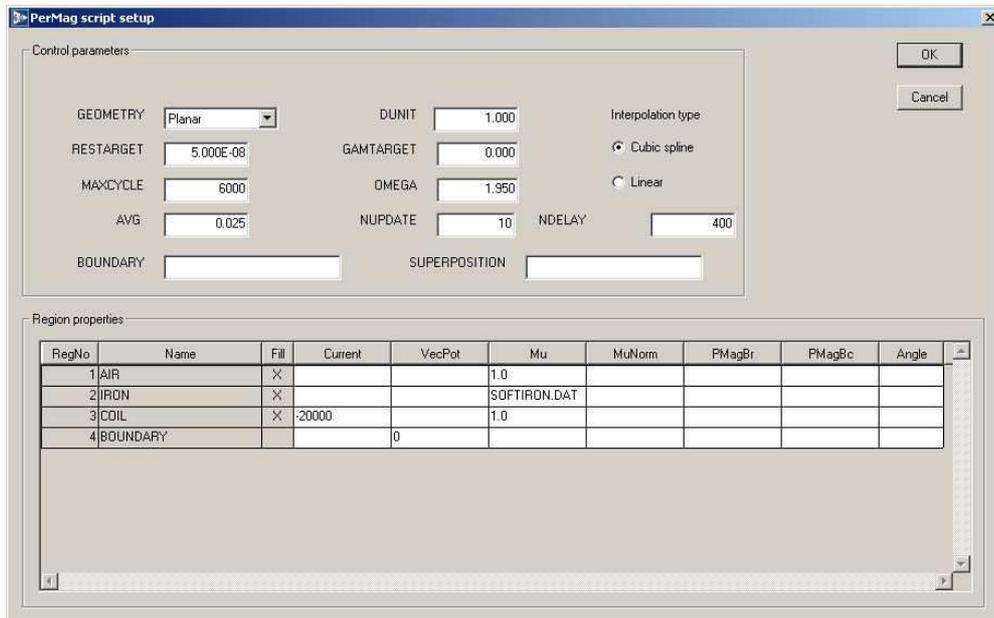


Figure 11: Dialog to set control and material properties with parameters for the H magnet calculation.

To start the first operation, we must identify the **Mesh** output file that defines the system geometry. Click the *1* tool or the *Setup* menu command. In the dialog, choose the file **HMAG.MOU**. **PerMag** loads the information and displays the dialog of Fig. 11. Note that the grid contains a row for each mesh region with a notation of the *Fill* status. There are seven columns in the grid where you can enter values: *Current* (total drive current in amperes), *VecPot* (fixed value of vector potential in tesla-m or stream function in tesla-m²), *Mu* (relative magnetic permeability for isotropic materials and value along Axis 1 for anisotropic materials), *MuNorm* (relative magnetic permeability along Axis 2 for anisotropic materials), *PMagBr* (residual magnetic induction for permanent magnets in tesla), *PMagBc* (coercive magnetic induction for permanent magnets in tesla) and *Angle* (orientation of Axis 1 for anisotropic materials and easy-axis for permanent magnets in degrees). Section 4.1 describes how entries in a row determine the material type.

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

- **GEOMETRY**. The symmetry of the solution: *planar* or *cylindrical*. A planar solution varies in *x-y* and has infinite length in *z*, while a cylindrical solution has symmetry in θ .
- **RESTART**. Accuracy tolerance for the iterative matrix solution of the finite-element equations.
- **MAXCYCLE**. Maximum number of cycles in the iterative solution.
- **AVG**. Averaging parameter for adjustment of non-linear materials, a quantity much smaller than unity.
- **DUNIT**. A factor to convert the units used for coordinates in the **Mesh** file to meters. The value is the number of mesh units per meter: 39.37 for inches, 100.0 for cm.
- **GAMTARGET** Accuracy tolerance for adjustment of non-linear materials.
- **OMEGA**. A parameter in the range 0.0 to 2.0 to control the iterative matrix solution.
- **NUPDATE**. Number of solution iteration cycles between adjustments of non-linear materials.
- **NDELAY**. Number of solution iteration cycles before beginning adjustment of non-linear materials.
- **BOUNDARY** and **SUPERPOSITION**. Advanced program capabilities described in Sect.4.6.

Section 4.2 describes the control parameters in detail. The default values are usually sufficient. The values in the region grid define the following characteristics:

- Region 1. Air: $\mu_r = 1.0$.
- Region 2. Iron, a non-linear material with relative magnetic permeability defined by values in the file **SOFTIRON.DAT** (Fig. 12).
- Region 3. The drive coil section with $\mu_r = 1.0$ and current -20,000 A-turn. The total drive current amplitude for the H-magnet is 40,000 A-turn.

Table 3: PerMag script HMAG.PIN

```

Mesh = HMAG
Geometry = Rect
DUnit = 1.0000E+02
ResTarget = 1.0000E-08
Omega = 1.9500E+00
MaxCycle = 20000
Avg = 1.0000E-02
Update = 10 300
* Region 1: AIR
Mu(1) = 1.0000E+00
* Region 2: IRON
Mu(2, Table) = SOFTIRON.DAT
* Region 3: COIL
Current(3) = 2.0000E+04
Mu(3) = 1.0000E+00
* Region 4: BOUNDARY
VecPot(4) = 0.0000E+00
EndFile

```

- Region 4. The boundary with fixed vector potential $A_z = 0.0..$

Fill in the boxes as shown in Fig. 11 and click *OK*. **PerMag** uses the information in the dialog to create the script HMAG.PIN shown in Table 3. Chapter 4 reviews the script format and advanced program capabilities.

3.3 Finite-element solution

The next step is to generate and to solve the finite-element equations. Click the tool marked 2 or the menu command *Solve*. Accept the default entry of HMAG.PIN in the dialog by clicking OK. The blue screen color indicates that the program is in solution mode, and the status bar reports the progress of the operation. When complete, **PerMag** creates the output file HMAG.POU which contains full information on the mesh as well as values of the vector potential at each node and relative permeability in elements. Section 4.10 describes the file format.

3.4 Solution analysis

We can now use the information in HMAG.POU to create plots and to perform quantitative analyses. Choose either the tool marked 3 or the *Analyze* menu command. In the dialog, accept the default choice of data file. **PerMag** loads the file, sets up the analysis menu and toolbar, and creates a default field-line plot. Take a moment to inspect the organization of the working environment. The status bar at the bottom lists the following information: solution file name, data file name, plot quantity, plot type, interpolation method and mouse snap status. The data file gives you the option to record the results of analysis operations in text format. The plot window is divided into four areas. The main plot is at the upper left and the plot legend is

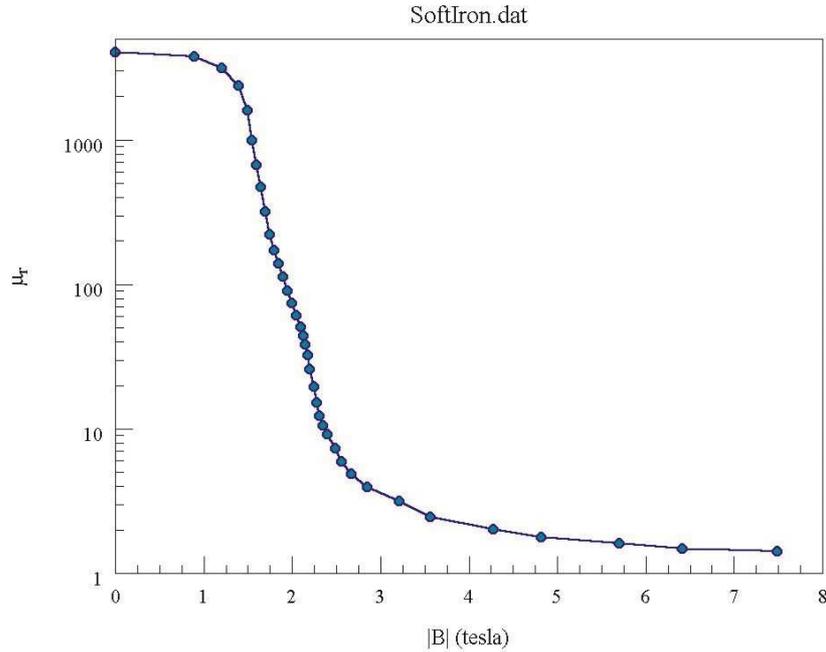


Figure 12: Variation of μ_r with $|\mathbf{B}|$ defined by the table SOFTIRON.DAT.

at the upper right. The orientation area at the lower right is a miniature representation of the solution volume with an outline showing the current view. The information area at the bottom left appears in response to analysis commands such as *Point calculation* or *Region properties*.

The following chapter gives detailed reference material on analysis function of **PerMag**. You can experiment with the program capabilities to create a wide variety of plots. Figure 13 shows a three-dimensional surface plot of pole saturation. To complete the session, we shall run a script to perform an automatic calculation. To inspect the file content, click on *Edit script* in the *File* menu and pick the file `HMAG.SCR`. The file has the content

```
INPUT HMAG.POU
OUTPUT HMAG
NSCAN 10
SCAN 0.0 0.0 5.0 0.0
ENDFILE
```

The first command ensures that the file `HMAG.POU` is loaded while the second command opens the data file `HMAG.DAT`. The *NScan* and *Scan* commands instruct the program to calculate and to list field values at eleven evenly-spaced points along the magnet midplane. Table 4 shows an excerpt from the listing.

3.5 Permanent magnet calculation

To illustrate a permanent magnet calculation, we shall review the simulation of Fig. 14. The input files are `PPM.MIN`, `ALNICO.DAT` and `PPM.SCR`. The calculation addresses a periodic permanent-magnet array (PPM). The assembly is used for electron beam transport in high-power traveling

Table 4: Field values in the magnetic midplane calculated with the script HMAG.SCR.

X	By, Br	H
0.000E+00	-1.245E+00	9.909E+05
5.000E-01	-1.245E+00	9.908E+05
1.000E+00	-1.245E+00	9.906E+05
1.500E+00	-1.244E+00	9.902E+05
2.000E+00	-1.243E+00	9.893E+05
2.500E+00	-1.241E+00	9.873E+05
3.000E+00	-1.236E+00	9.832E+05
3.500E+00	-1.225E+00	9.749E+05
4.000E+00	-1.205E+00	9.586E+05
4.500E+00	-1.167E+00	9.284E+05
5.000E+00	-1.105E+00	8.790E+05

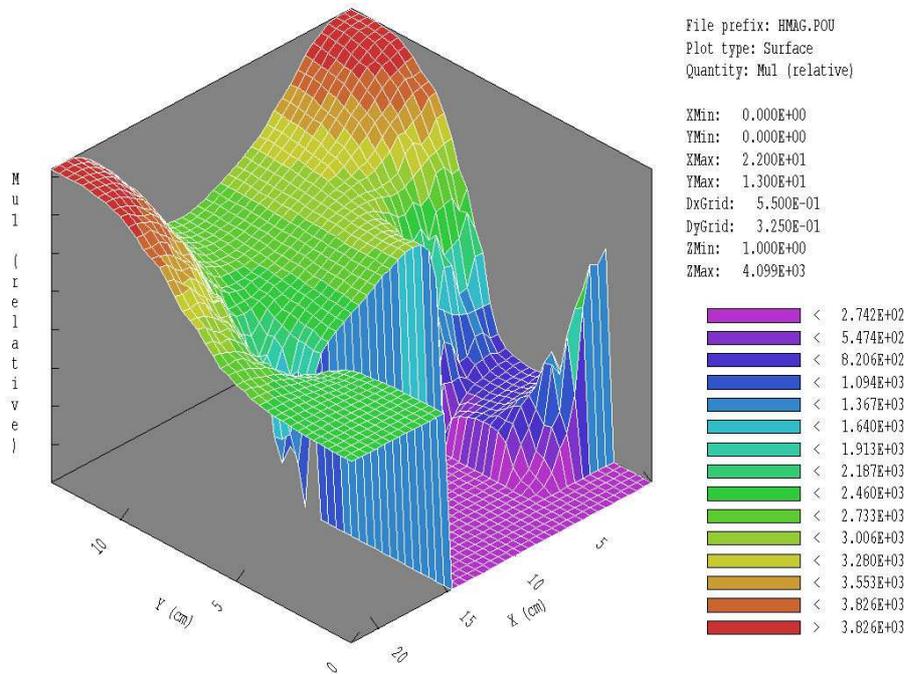


Figure 13: Surface plot of the variation of μ_r in the iron region. The air gap is at the lower-right portion of the plot.

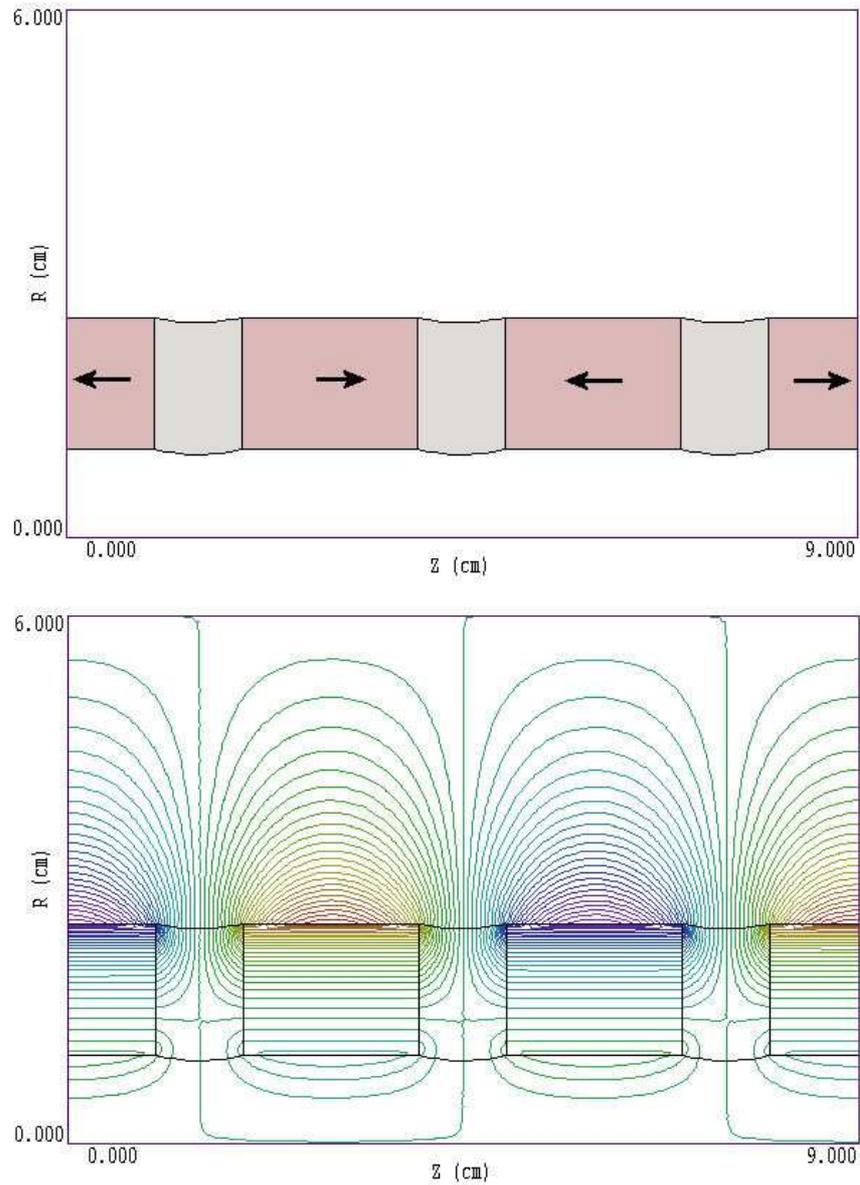


Figure 14: Periodic permanent magnet array for electron-beam focusing. *Top*) Geometry covering two periods of the array. Shaped iron poles are shown in gray, and annular magnets are shown in red. Arrows indicate the magnetization direction. *Bottom*) Calculated lines of magnetic flux density.

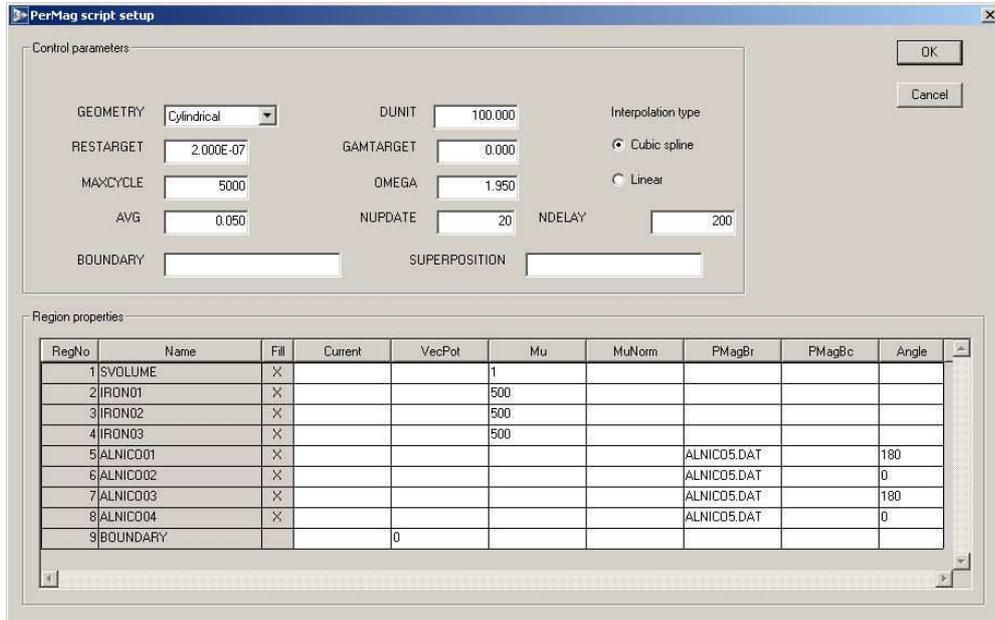


Figure 15: Dialog to set material properties – setup for the PPM example.

wave tubes. Annular permanent magnets have their magnetization axis along z . The magnetization direction reverses between adjacent magnets, producing a periodic variation of solenoid field on the axis. Regions 5, 6, 7 and 8 are magnets with magnetization in $-z$, $+z$, $-z$ and $+z$ respectively. To model an infinitely long system, we represent Regions 5 and 8 as half magnets and apply the Neumann condition on the left and right sides. The magnetic material is Alnico5. In comparison to modern materials, Alnico has a relatively low energy product and undergoes some demagnetization in a circuit with an air gap. Regions 2, 3 and 4 are shaped soft-iron inserts to conduct field lines radially. Region 9 (not shown) is composed of lines along the upper and lower boundaries where $\psi = 0.0$.

Process the MIN file with **Mesh** then run **PerMag**. Choose the *Setup* command to bring up the dialog for program parameters. Enter the values shown in Figure 15. Exit the dialog and save the information in the default file HMAG.PIN. Run the example with the *Solve/Run* command. The relaxation process and adjustment of material properties take place simultaneously to minimize the run time. **PerMag** creates the listing file PPM.PLS and the output file PPM.POU. The first file includes a listing of volume-averaged energy product for each permanent magnet region.

--- ENERGY PRODUCT ANALYSIS OF PERMANENT MAGNET REGIONS ---

UNITS

Int(BH) in J/m (planar) or J (cylindrical)
 Volume in m3/m (planar) or m3 (cylindrical)
 Avg(BH) in J/m3

NReg	Int(BH)	Volume	Avg(BH)
5	6.33261E-02	1.64934E-05	3.83949E+03
6	1.26503E-01	3.29867E-05	3.83496E+03

7	1.26499E-01	3.29867E-05	3.83484E+03
8	6.31737E-02	1.64934E-05	3.83025E+03

The energy product, defined as the product BH along the axis of magnetization, indicates how effectively a permanent magnet material is used. The final step is to analyze the results. Enter the *Analyze* menu and confirm that the configuration produces an approximately sinusoidal variation of B_z along the axis with peak amplitude 0.054 tesla.

Table 5: Setting material models from entries in the **PerMag** dialog. (The letter N denotes a number, while S denotes a string).

Material type	Curr	Pot	Mu	Mun	Br	Bc	Ang
Fixed A_z or ψ		N					
Linear material			N				
Coil	N		N				
Anisotropic linear			N	N			N
Nonlinear isotropic			S				
Nonlinear anisotropic			S	S			N
Straight-line pmagnet					N		N
Two-point pmagnet					N	N	N
Demagnetization curve					S		N

4 PerMag solution reference

4.1 PerMag script format

You can create a control script for **PerMag** using the interactive dialog described in Sect. 3.2 or by writing the commands directly with a text editor. You can easily alter any script with an editor. This chapter gives a detailed description of the syntax and functions of script commands. Section 4.2 covers commands that control program operation. These commands are created by the entries in the *Control parameters* section of the dialog (Fig. 11). Section 4.3 through 4.5 review commands to set material properties. These commands are created by entries in the *Region properties* grid of the dialog. In this case, **PerMag** determines the material model for a region by checking the number and types of entries in the corresponding row of the region grid. The program first attempts to convert entries to a floating-point number. If the procedure fails, the program assumes that the entry is a string representing a table name. Table 5 shows conversion conventions. For example, to define a region as a drive coil, enter the numerical value of current in the *Curr* column and the number 1.0 in the *Mu* column. To specify a nonlinear permanent magnet, place the name of the demagnetization table in the *Br* column and a numerical value for the angle in the *Ang* column. Only the combinations shown in Table 5 are allowed. **PerMag** makes an extensive validity check of parameters when you attempt to close the dialog.

The **PerMag** script is a text file composed of data lines that contain commands and parameters. The script must end with the *EndFile* command. The entries on a line may be separated by the delimiters introduced in the **Mesh** manual: :

- Space [' ']
- Comma [',']

- Tab
- Colon [':']
- Left parenthesis ['(']
- Right parenthesis [')']
- Equal sign ['=']

Any number of delimiters may be used in a line. Blank lines and comment lines are ignored. Comment lines begin with an asterisk (*). **PerMag** accepts commands in any order. The following example illustrates a complete script:

```
* FILE RING.PIN
ResTarget = 5.0E-8
Geometry = Rect
DUnit = 100.0

Mu(1) = 1.0
PerMag(2) = (1.12, 90.0)
PerMag(3) = (1.12, -90.0)
PerMag(4) = (1.12, 180.0)
VecPot(5) = 0.0

EndFile
```

There are two classes of commands: program control and region properties. A control command contains a keyword and a value. Region commands set the physical properties associated with elements and nodes. They have the format:

```
Keyword RegNo Value(s)
```

Here, the integer *RegNo* is the region number defined in the **Mesh** input file. The string *Keyword* specifies the physical property. The value may be one or more numbers. As an example, the command

```
Current(2) = 3.5E4
```

sets a uniform current density in elements of region number 2 such that the total current sums to 35.0 kA. (**Note:** For back-compatibility, **PerMag** recognizes command formats from Version 1.0 through 5.0 where control commands start with the keyword *Set* and region commands start with *Region*.)

4.2 Program control commands

In the following sections, commands are written symbolically and as they might appear in the **PerMag** script:

MESH MPrefix

MESH SparkGap

Specify the **Mesh** output file that defines the geometry of the simulation. The file must have a name of the form **MPrefix.MOU** and be available in the working directory. If this command does not appear in a script with the name **EPrefix.PIN**, then **PerMag** will seek the default mesh file **EPrefix.MOU**.

DUNIT DUnit

DUNIT = 1.0E4

You can use any convenient distance units in **Mesh**. This command defines a factor to convert coordinates supplied by **Mesh** to the standard distance units of meters used in **PerMag**. The quantity *DUnit* equals the number of **Mesh** units per meter. For example, if the **Mesh** dimensions are entered in microns, set $DUnit = 1.0 \times 10^6$. The default value is 1.00. (**Note.** Spatial quantities recorded in the output file **FPrefix.POU** are always in meters. In an analysis session with **PerMag**, spatial quantities in graphs and listing files are scaled to the **Mesh** units. For example, if the **Mesh** dimensions are in cm and $DUnit = 100.0$, the spatial quantities in plots will be in cm.)

GEOMETRY [Rect, Cylin]

GEOMETRY = Cylin

PerMag handles problems in rectangular or cylindrical geometries. Rectangular systems have variations in x and y with infinite length in z . Cylindrical systems have variations in r and z with azimuthal symmetry. The parameter options are *Rect* and *Cylin*. In cylindrical solutions the program takes the z axis along the **Mesh** x direction and the r axis along y . In this case the program issues an error message if any node has a y coordinate less than 0.0.

OMEGA Omega

OMEGA = 1.96

This command controls the over-relaxation factor for the iterative matrix solution, a number in the range 0.0 to 2.0. Generally, higher values give faster convergence. Reduce the over-relaxation factor if the solution does not converge. If this command does not appear, the program uses the Chebyshev acceleration method to pick optimal values of the over-relaxation factor.

MAXCYCLE MaxCycle

MAXCYCLE = 500

The maximum number cycles for the iterative matrix solution. The default value is 2500.

RESTARTARGET ResTarg
RESTARTARGET = 5.0E-6

PerMag calculates the relative error in the magnitude of the vector potential or stream function during the iterative matrix solution. The program stops when the error falls below *ResTarg*. Solutions generally have sufficient accuracy for *ResTarg* less than about 1.0×10^{-6} . The default value is 5.0×10^{-8} .

GAMMATARGET = GamTarg
GAMMATARGET = 5.0E-5

Sets a target for the root-mean-squared error in the correction of the quantity $\gamma = 1/\mu_r$, where μ_r is the relative magnetic permeability in non-linear materials. The default value is *GamTarg* = 10^{-5} . In runs with non-linear materials, **PerMag** will continue iterations until both *ResTarg* and *GamTarg* have reached the desired values or until *MaxCycle* is exceeded.

AVG = Avg
AVG = 0.05

The quantity *Avg* controls adjustment of non-linear materials during the iterative matrix solution. For convergence, the value should be much smaller than unity. The default is 0.1. A low value corresponds to a gradual adjustment of magnetic permeability. Values in the range 0.05 or less may be necessary to achieve convergence when there are highly saturated materials. **PerMag** lists the root-mean-squared error in material values during the solution.

UPDATE NUpdate [NDelay]
UPDATE = (20, 300)

This command controls adjustment of the magnetic permeability in non-linear materials during the iterative solution. The first parameter after the keyword (*NUpdate*) is the number of iteration cycles per material update. The default value is 10. The second optional integer parameter (*NDelay*) equals the number of interaction cycles performed before material updating begins. The default is *NDelay* = 150. It is essential to delay non-linear material adjustment because the initial distribution of vector potential is an arbitrary guess that could give unrealistically large values of magnetic flux density in some elements. The delay allows the solution to relax to a state closer to a valid solution. You may need to increase *NDelay* for runs that converge slowly.

INTERP [Spline, Linear]
INTERP = Linear

Set the interpolation method for μ_r tables of non-linear materials and demagnetization curve for permanent magnets. The options are *Spline* (default) and *Linear*. Spline interpolations are forced to maintain continuous derivations and may return spurious values for noisy or discontinuous data. In this case, use the *Linear* option.

PARALLEL [NProc]**PARALLEL = 4**

Use multiple threads for the matrix inversion. The integer parameter *NProc* is the number of threads to invoke. If omitted, the number of threads equals to maximum number on the machine (8 for a Core i7 processor). This command functions only for multi-core or multi-processor machines running the 64-bit version of the software. Parallel processing may be useful for huge-mesh solutions that run several minutes. In typical runs, the computational overhead may actually increase the run time.

4.3 Commands for simple material properties

The following four command forms define the properties of linear materials in simulations with drive coils.

VECPOT RegNo VecPot**VECPOT(5) = 0.0**

The keyword *VECPOT* designates that the vector potential A_z or stream function ψ of the region nodes is fixed and will not change during the iteration solution. The command sets the amplitude of the vector potential in tesla-m or stream function in tesla-m².

MU RegNo MuR**MU(3) = 500.0**

Sets a value of relative magnetic permeability ($\mu_r = \mu/\mu_0$) in all elements of a region.

CURRENT RegNo Ic**CURRENT(3) = 1500.0**

This command assigns current I_c to a region that represents a magnet coil. A uniform value of current density j_c is assigned to all elements of the region. The current density is given by $j_c = I_c/A_c$, where A_c is the cross section area of the region. In planar solutions the current points along z (out of the page in an x - y plot). In cylindrical solutions the current is directed along θ (out of the page in a z - r plot). Note that both *Mu* and *Current* commands must appear for the same region. For most solutions, $\mu_r = 1.0$ in coil regions. For variable winding density, divide a coil into several regions.

MU RegNo Mu1 Mu2 Theta**MU(2) = (500.0, 20.0) 45.0**

Define an anisotropic magnetic material with different values of relative permeability constant along two normal axes. The first real-number value (μ_1) is the relative magnetic permeability along Axis 1, and the second number (μ_2) is the permeability along Axis 2. The third number (θ) is the angle in degrees of Axis 1 relative to the x (or z) axis (Fig. 16). The angle of Axis 2 is $\theta = 90^\circ$. For cylindrical solutions, it is important to note that all structures must have cylindrical symmetry. Therefore, if $\theta \neq 0.0^\circ$, then Axis 1 and Axis2 have the shape of cones. Because it is unlikely to encounter such a material, practical cylindrical simulations are generally limited to the choice $\theta = 0.0^\circ$ (different permeability values along z and r).

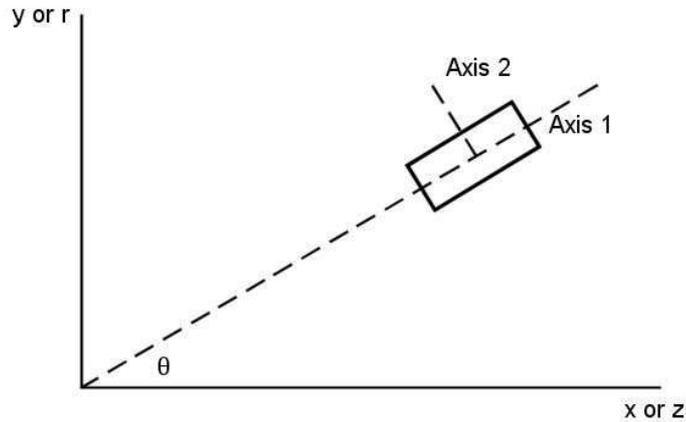


Figure 16: Orientation of axes for anisotropic materials.

4.4 Nonlinear magnetic materials

PerMag can represent soft magnetic materials like transformer steel. A soft material has a narrow hysteresis curve so that the magnetization process is almost reversible. A common reason to model non-linear materials is to determine saturation effects in magnet poles. The magnetization curve of a material is entered as a table of values of the magnetic flux density (tesla) and the corresponding relative permeability μ_r . On selected cycles of the relaxation solution, **PerMag** calculates $|\mathbf{B}|$ in elements associated with a tabulated material. The program then uses the interpolated value of the magnetic permeability to update the coupling coefficients for all connected nodes. The two relaxation processes proceed in tandem, leading to self-consistent values of the magnetic permeability and the field. For maximum speed, **PerMag** updates coupling coefficients only in non-linear regions and skips the process altogether when no tables have been entered.

The key to flexibility in modeling non-linear materials is program input through *tabular functions*. A magnetization table is a text file consisting of up to 256 data lines that contains values of magnetic flux density amplitude $|\mathbf{B}|$ 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 experimental traces. Table 6 shows a portion of the tabular function `SOFTIRON.DAT` supplied with the **PerMag** package (Fig. 12). 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*.

Although the lines in Table 6 are ordered by increasing value of $|\mathbf{B}|$, the ordering is not required. **PerMag** sorts the list before use and records the final order in the listing file. Furthermore, the intervals need not be uniform – entries in the example are clustered near the saturation transition. 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 magnetic field that will be encountered in the solution. The interpolation routines return the value $\mu_r = 1.0$ for values of $|\mathbf{B}|$ that are off the table.

You can create your own tables if you have experimental or published data for a soft magnetic material. The information is usually given in the form B versus H . If you put the values in

Table 6: Excerpts from the file SOFTIRON.MAT

```
* File SOFTIRON.DAT
* Generic soft iron table
* Adapted from Superfish/Poisson Code Group
* Los Alamos National Laboratory
  0.0000    4075.45105
  0.8944    3768.28312
  1.2000    3166.80993
  1.4000    2380.64690
  . . .
  2.8498    3.94592075
  3.2074    3.15384615
  3.5644    2.45501166
  4.2782    2.02377622
  4.8134    1.76596737
  5.7052    1.60606061
  6.4186    1.48298368
  7.4887    1.42983683
ENDFILE
```

a text file, one option is to use the *Mu Table Tool* in the **PerMag Help** menu (described in Sect. 4.9). To build a table by hand, convert the values of B and $\mu_0 H$ to units of tesla. Each data line should contain a value of B followed by the corresponding value of the relative magnetic permeability, $\mu_r = B/\mu_0 H$.

In the default mode, **PerMag** uses cubic splines for interpolation of the 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 dependent quantity and its first derivative should vary smoothly over the table range. 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 PLS listing file. After recording the sorted table, **PerMag** writes a sample set of interpolated values.

Nonlinear materials are defined with the following variations of the *Mu* command:

MU RegNo TABLE TableName

MU(5) = (Table, Steel1014.Dat)

This command defines properties of a nonlinear, isotropic magnetic material. The string parameter following the region number is the full name of a data file in the current directory.

MU RegNo TABLE Axis1Name Axis2Name Theta

MU(8) = (Table, LaminaParallel.DAT LaminaPerp.DAT) 45.0

This command form defines non-linear variations of the relative magnetic permeability along

two normal axes (Fig. 16). The first string parameter is the name of a $|\mathbf{B}| - \mu_r$ data file in the current directory for Axis 1. The second parameter is the name of the data file for Axis 2.

4.5 Permanent magnet materials

PerMag has three options to represent permanent magnets. The ideal straight-line model with a 45° demagnetization curve is a good approximation for modern materials like neodymium-iron and samarium-cobalt. Non-linear materials like Alnico and ceramics follow a more complex variation. If you have access to the non-linear demagnetization curve, you can enter it as a tabular function. Otherwise, you can simply supply values for B_r and B_c and **PerMag** will apply Eq. 11.

PERMAG RegNo Br Theta **PERMAG(9) = (1.27, 45.0)**

This command defines an ideal permanent magnet with a straight-line demagnetization curve where $B_c = -B_r$. Here, B_r is the remanence value of magnetic flux density and B_c is the coercive value of flux density ($B_c = \mu_0 H_c$). Both values are in tesla. The final parameter is the angle in degrees between the axis of magnetization (easy axis) and the x (or z) axis (Fig. 16).

PERMAG RegNo Br Bc Theta **PERMAG(5) = (1.10, -0.96) 45.0**

This command defines a two-point permanent magnet material where the demagnetization curve follows the Eq.11. The three parameters have the following meanings: B_r in tesla, B_c in tesla (the sign may be omitted) and the angle in degrees between the direction of magnetization and the x (or z) axis. Table 2 lists values of B_r and B_c for several materials. Specifications for commercial materials often give the remanence magnetic flux density in kG and the coercive magnetic field H_c in kOe. To convert from kG to tesla, multiply by 0.1. To convert H_c in kOe to B_c in tesla, multiply by 0.1.

PERMAG RegNo TABLE TableName Theta **PERMAG(9) = (TABLE, Alnico8.DAT) 30.0**

This command defines a material with a non-linear demagnetization curve. The string parameter following the keyword *Table* is the full name of the data file that contains the demagnetization curve. The file must be available in the working directory. The second parameter is the angle (in degrees) between the direction of magnetization and the x (or z) axis.

Table 7 shows an example of a demagnetization table. Data lines have the entries

B B0

where both quantities are given in tesla. The first quantity is the total flux density in the material and the second is the return flux density (equal to $\mu_0 H$). The first data line must contain the entries:

0.0 BC

Table 7: Demagnetization table for Alnico 9

```

* Alnico 9
*   B(tesla)      B0(tesla)
* =====
      0.000        -0.1360
      0.100        -0.1355
      0.200        -0.1350
      0.300        -0.1344
      0.400        -0.1335
      0.500        -0.1320
      0.600        -0.1304
      0.700        -0.1278
      0.800        -0.1232
      0.900        -0.1135
      0.950        -0.0972
      1.000        -0.0758
      1.050        -0.0495
      1.100        -0.0212
      1.125        -0.0000
ENDFILE

```

Lines should be arranged in ascending order of B and terminate at

```
BR    0.0
```

The file may contain comments, blank lines and up to 256 data lines.

To employ the method described **Field Solutions on Computers** (Sect. 9.7), **PerMag** converts the table to an internal format of B - μ , where the quantity μ is defined by:

$$B = \mu(B_c - B_0), \tag{12}$$

The program records a copy of the converted table in the PLS listing file.

4.6 Boundary and superposition commands

PerMag can perform accurate calculations of fields near small features in a large solution space using the boundary method. The example shown in Fig. 17 illustrates how the method works. We want to place iron shims on the pole faces at the gap of a C-magnet to improve the field uniformity. We anticipate experimenting with several shim geometries to find the best one. The problem is that there is a large disparity of scale. The shim thickness is only 1.0 mm, while the magnet dimension is ~ 30.0 cm. It would be a waste of time to recalculate the entire system for each shim configuration. **PerMag** offers an efficient alternative. We first create a single global solution for the C-magnet (Fig. 17, top), omitting fine details at the gap. We then perform a series of microscopic solutions near the gap (Fig. 17, bottom) with reduced element size to resolve small details. Interpolation over the macroscopic solution gives values of A_z to define a variable-potential Dirichlet boundary on the outer surface of

the microscopic solution (dashed red line). The microscopic solution is accurate as long as new features (such as the shims) are not close to the interpolation boundary. Input files to define the global solution (BOUNDARYTEST01.MIN and BOUNDARYTEST01.PIN) and a microscopic solution (BOUNDARYTEST01.MIN and BOUNDARYTEST01.PIN) are included in the example library. Figure 18 shows a plot of B_y in the midplane ($y = 0.0$) as function of distance from the gap center. The blue curve shows the field variations with no shim, while the red curve corresponds to shims of width 0.5 cm and thickness 0.1 cm.

The variable Dirichlet boundary calculation is controlled by the following command in the control script of the microscopic solution:

BOUNDARY FPrefix [BndScale]
BOUNDARY FEmitMacro

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

PerMag issues an error message under the following conditions:

- The output file for the global solution is not available in the working directory.
- The microscopic solution does not fit completely inside the global solution.
- The global and microscopic solutions have different symmetries.
- An interpolation failed.

The program makes no further validity checks. You must ensure that the geometry of the microscopic solution represents a correct section of the global space.

A node in the microscopic solution is on the boundary under two conditions:

- It has indices $K = 1$, $K = K_{max}$, $L = 1$ or $L = L_{max}$. In solutions with cylindrical symmetry, points on the axis ($L = 1$ and $r = 0.0$) are not set to the Dirichlet condition.
- One of the adjacent elements has $RegNo = 0$. This condition means that you can use non-rectangular boundaries for Solution 2.

The *Superposition* command directs **PerMag** to superimpose vector-potential values from a global solution (Solution 1) over the full space of an enclosed solution (Solution 2). The following command appears in the script for Solution 2:

SUPERPOSITION FPrefix [SScale]
SUPERPOSITION UniField

The quantity *FPrefix* (string) is the prefix of the output file for Solution 1. The optional quantity *SScale* (real) is a scaling factor applied to potential values transferred from Solution 1 to Solution 2. The default is $SScale = 1.0$.

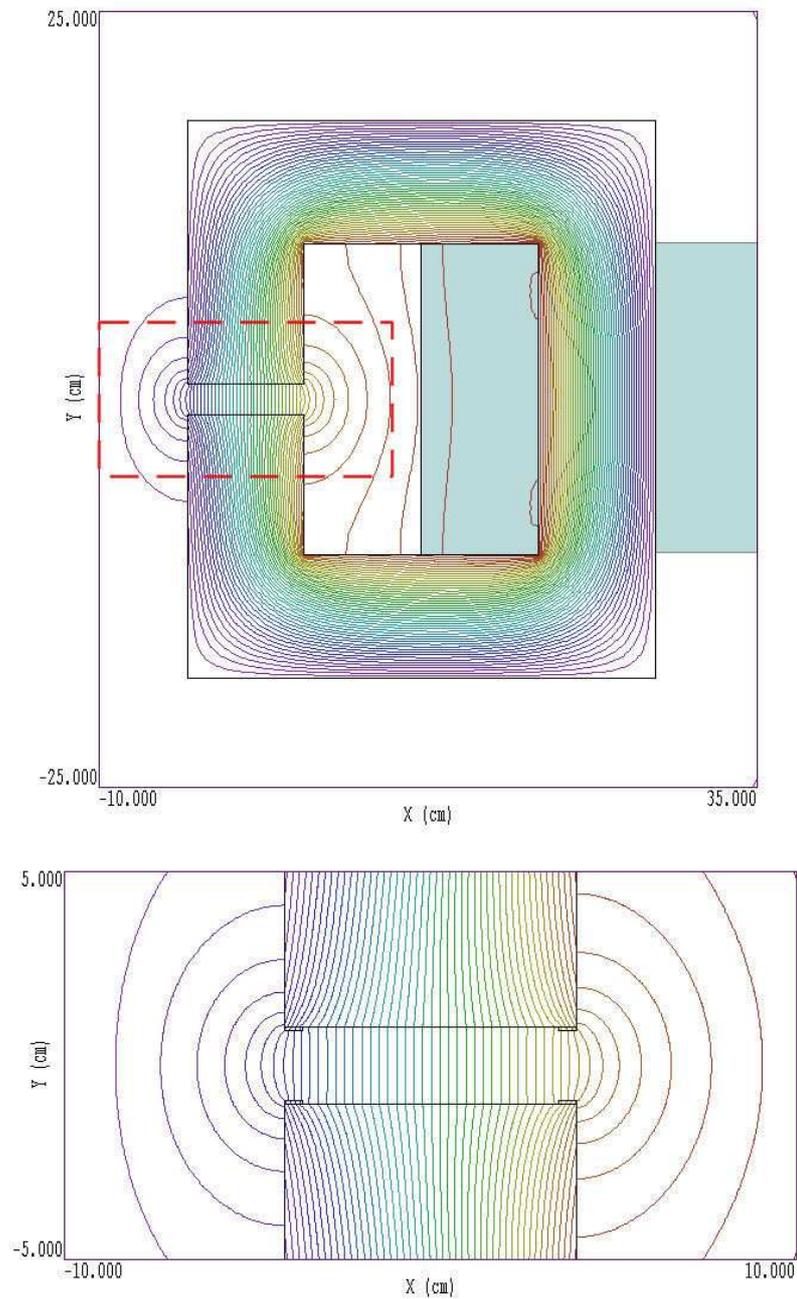


Figure 17: Illustration of the boundary method. A global solution for a C-magnet is shown at the top. The microscopic solution (bottom) has a Dirichlet boundary with values computed along the dashed line in the global solution.

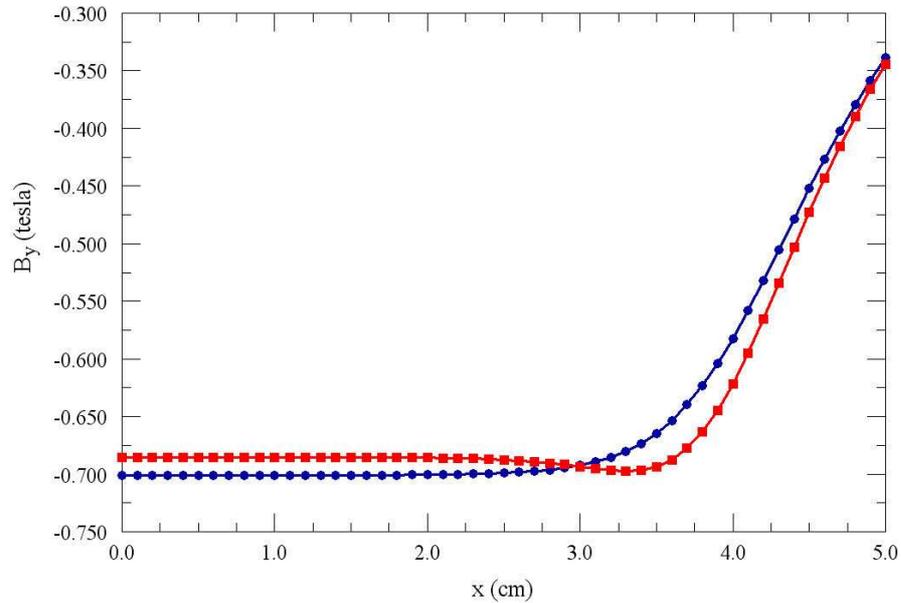


Figure 18: Variation of B_y in the plane $y = 0.0$ as a function of distance from the gap center. Blue line: no shims. Red line: shim with 0.5 cm width and 0.1 cm thickness.

In response to the command, **PerMag** opens the file `Solution1.POU` after Solution 2 has been completed. The program performs an interpolation in the space of Solution 1 to determine the A_z or ψ at each node location in the enclosed solutions and adjusts values according to

$$A_z(2) = A_z(2) + SScale \times A_z(1), \quad (13)$$

before writing the output file. You must ensure that the superposition is physically correct.

4.7 Running PerMag

PerMag can run as an interactive program in a window or as a background task. To run the program from the Command Prompt, use a command of the form:

```
[ProgPath\]PERMAG [DataPath\]FPrefix <ENTER>
```

where the file `FPrefix.PIN` and the appropriate **Mesh** output file are available in the data directory. With this capability, you set up extended autonomous **PerMag** runs using a DOS batch file or a Perl script..

The remainder of this section discusses commands in the main menu when **PerMag** runs in the interactive mode. The program enters this mode when launched from **TC** or run with no input file prefix. The following commands appear in the *File* popup menu:

EDIT SCRIPT (PIN)
EDIT LISTING (PLS)
EDIT FILE

These commands call the internal editor to inspect or to modify **PerMag** input and output

files. Choosing a file from an alternate directory does not change the working directory. The *Edit script (PIN)* command shows a list of all files with names of the form `FPREFIX.PIN` while *Edit listing (PLS)* displays files with names `FPREFIX.PLS`.

RUN ANALYSIS SCRIPT

An analysis script allows you to perform complex or repetitive analyses on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.7. Note that the analysis script must be in the same directory as the data files.

SETUP

The function of this command is to create a **PerMag** script to control a magnetostatic solution. The program first prompts for a **Mesh** output file to define the system geometry. The prefix of the file will be used as the argument of the *Mesh* script command. The program then displays the dialog shown in Fig. 11. The number of regions in the dialog is determined by the **Mesh** file. The functions of the control parameters in the upper box are described in Sect. 4.2.

RUN

Pick an input file (`FPrefix.PIN`) to start a solution. The working directory is changed if you pick a file from an alternate directory. The run begins if the file `FPrefix.MOU` or the file specified in the *Mesh* command and any required data files are present. During the solution, the screen color is blue and the progress is shown in the status bar.

STOP

This command terminates **PerMag** and saves the current state of the relaxation solution. For example, you may want to stop a relaxation solution at a moderate value of convergence to check whether the problem has been correctly defined.

SOLUTION DISPLAY

When *solution display* is active, **PerMag** plots calculated magnetic field lines during the matrix inversion to show solution convergence. The number of relaxation steps per plot is initially small and increases as the solution nears its final state. This feature is useful to verify convergence or to identify problem areas. (Note that the plot operations increase the run time.)

ANALYZE

Pick a file of the type `FPrefix.POU` and call up the analysis menu for plotting and numerical analysis.

TABLE GENERATOR

Run the utility described in the next section to create a tabular function from a mathematical expression.

PERMAG MANUAL

Show this manual in your default PDF display software. The file `permag.pdf` must be available in the same directory as `permag.exe`.

4.8 Field energy listings

For calculations that include applied currents, integrals of field energy may be used to determine inductance. Suppose that a total magnetic field energy U_m is created by a coil region that carries a net current NI (where I is the drive current and N is the number of turns in the coil). In this case, the inductance is related to the field energy by:

$$L = \frac{2U_m}{I^2}. \quad (14)$$

PerMag calculates volume integrals of field energy density u_m organized by region and records them in the listing file `RUNNAME.PLS`. The expression for u_m depends on the type of material. The field energy density for air or an isotropic material with a fixed relative magnetic permeability is:

$$u_m = \frac{B^2}{2\mu_r\mu_0}. \quad (15)$$

The energy density in an isotropic, nonlinear material characterized by a field-dependent relative magnetic permeability is

$$u_m = \int_0^B \mathbf{H}(B') \cdot d\mathbf{B}' = \int_0^B \frac{B^2}{\mu_r(B)\mu_0}. \quad (16)$$

The expression for an isotropic linear material with fixed values of relative magnetic permeability μ_1 and μ_2 along two normal axes is

$$u_m = \frac{1}{\mu_0} \left(\frac{B_1^2}{2\mu_{r1}} + \frac{B_2^2}{2\mu_{r2}} \right). \quad (17)$$

In Eq. 17, the quantities B_1 and B_2 are the components of magnetic flux density along the axis directions. Finally, an anisotropic, nonlinear material involves integrals like that of Eq. 16 along each axis.

Here is an example of an energy table for a simple solution involving air (Region 1) and soft iron (Rregion 2):

FIELD ENERGY INTEGRALS (EXCLUDING PERMANENT MAGNET REGIONS)				
NReg	Um (J)	Volume (m3)	Avg(um) (J/m3)	Material model
1	1.06716E-02	4.71239E-03	2.26457E+00	Linear, isotropic
2	1.77859E+00	1.57080E-03	1.13229E+03	Nonlinear, isotropic

When a solution contains a nonlinear material, **PerMag** also lists values of the integral of Eq. 16:

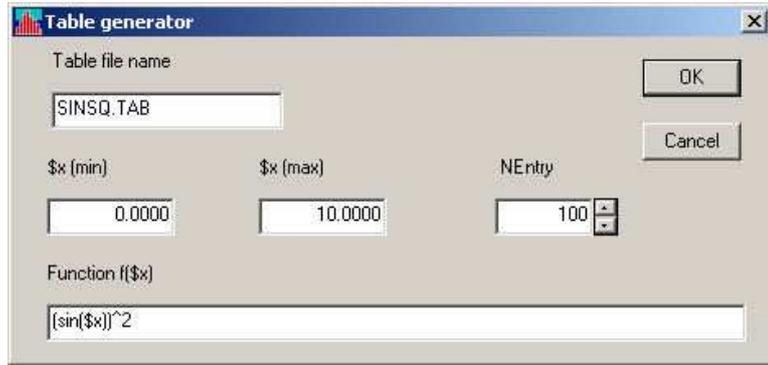


Figure 19: Table Generator Tool

```

Energy density table, Axis 1, Region No: 2
NEntry      B      Int [BdB/(Mu0*MuR)]
            (tesla)      (J/m3)
=====
  1  0.00000E+00  0.00000E+00
  2  1.00000E-01  7.95775E+00
  3  2.00000E-01  3.18310E+01
  ...
 22  2.50000E+00  4.97359E+03
 23  3.00000E+00  7.16197E+03

```

Regions of permanent magnet materials are not included in the field-energy list. Instead, the code computes the average value of *energy product*, determined by a volume integral over the region:

$$\frac{\int \int \int dV B_1 H_1}{\int \int \int dV}. \quad (18)$$

In Eq. 18, the field components B_1 and H_1 point along the direction of magnetization. Energy product is the standard figure-of-merit to characterize the effective use of permanent magnet materials. The data are summarized in a table like the following:

```

--- ENERGY PRODUCT ANALYSIS OF PERMANENT MAGNET REGIONS ---
NReg      Int (BH)      Volume      Avg (BH)
            (J)            (m3)            (J/m3)
=====
  3  2.85051E+01  1.13097E-04  2.52040E+05

```

4.9 Computational utilities

The *Help* menu contains three utilities for preparing input for **PerMag**: *Table Generator Tool*, *Mu Table Tool* and *Field Index Tool*. The *Table Generator Tool* provides a quick way to create tabular function files for non-linear materials from mathematical functions. Click on the command *Help/Table Generator* to bring up the dialog of Fig. 19. Supply information in the following four fields:

- **Table file name.** Enter a name for the file, such as `FERRITE.MUR`. The string may contain up to 36 characters.
- **$\$x(\text{min})$ and $\$x(\text{max})$.** The range of the independent variable (real numbers).
- **Function $f(\$x)$.** An algebraic function of the variable $\$x$ following the rules discussed below. The string may contain up to 255 characters.
- **NEntry.** The number of data lines in the table. The maximum value is 256.

When you click *OK*, the table generator opens a text file, writes a heading and adds *NEntry* data lines, $[\$x, f(\$x)]$. The values are evenly spaced along $\$x$ from $\$x(\text{min})$ to $\$x(\text{max})$. Finally, the table generator appends *ENDFILE*, closes the file and then closes the dialog.

A function is a string (up to 255 characters) that may include the following entities:

- The independent variable $\$x$.
- Real and/or integer numbers in any valid format (*e.g.*, `3.1415`, `476`, `1.367E23`, `6.25E-02`, `8.92E+04`,...). Integers are converted to real numbers for evaluation.
- Binary operations: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division) and `^` (exponentiation).
- Functions: `abs` (absolute value), `sin` (sine), `cos` (cosine), `tan` (tangent), `ln` (normal logarithm), `log` (base 10 logarithm), `exp` (normal exponent) and `sqrt` (square root).
- Up to 20 sets of parentheses to any depth.
- Any number of space delimiters.

The parser conforms to the standard algebraic rules and features comprehensive error checking. Errors may include unbalanced parentheses, unrecognized characters and sequential binary operations. To illustrate a valid example, the expression

$$1 - \exp(-1.0*((\$z^2 + \$r^2)/24))$$

corresponds to

$$1 - \exp\left[-\left(\frac{z^2 + r^2}{24}\right)\right]. \quad (19)$$

The entries in the dialog shown in Fig. 19 give the file shown in Table 8.

The *Mu Table Tool* converts hysteresis curves for magnetic materials (H versus B) to the standard form used in **PerMag**, B 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*. **PerMag** 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 referece listing of $\mu_0 H$ and B in units of tesla in the form of comment lines. The data section of the file consists

Table 8: Extract from the file SINSQ.DAT

```
* Field Precision Table Generation Tool
*   Number of entries: 100
*   XMin:  0.0000E+00
*   XMax:  1.0000E+01
*   Function: (sin($x))^2
    0.0000E+00  0.0000E+00
    1.0101E-01  1.0168E-02
    2.0202E-01  4.0260E-02
    3.0303E-01  8.9051E-02
    4.0404E-01  1.5456E-01
    5.0505E-01  2.3411E-01
    ...
    9.7980E+00  1.3293E-01
    9.8990E+00  2.0852E-01
    1.0000E+01  2.9596E-01
ENDFILE
```



Figure 20: Mu Table Tool

of lines containing B and $\mu_r = B/\mu_0 H$. A cubic spline interpolation is used to find μ_r at $B = 0.0$.

The *Field Index Tool* is useful working with magnets for weak-focusing circular particle accelerators. The field index,

$$n = -\frac{B_z}{r} \frac{dB_z}{dr}, \quad (20)$$

plays an important role in vertical and horizontal focusing. To use the tool, pick a data file `FPREFIX.SUF` that contains a radial scan of axial magnetic field. Each line of the file should contain a value of r and the corresponding value of $B_z(0, r)$ in any units. The may contain comment lines and should terminative with the *EndFile* command. **PerMag** applies a cubic spline interpolation to calculate the derivative and creates a data file `FPREFIX.OUT` with line entries r , B_z and n .

4.10 Format of the PerMag output file

The **PerMag** output file `FPrefix.POU` is in text format. The file has three sections:

- Header with general information on the run
- Node and element information
- Region information

The header section consists of a title line and 9 data lines:

```

--- Run parameters ---
XMin: -1.270003E-01
XMax:  1.270003E-01
KMax:  101
YMin:  0.000000E+00
YMax:  2.540005E-01
LMax:  101
DUnit:  3.937000E+01
NReg:   5
ICylin:  1

```

Lines 2 and 3 list x_{min} and x_{max} , the limits along horizontal axis (x or z) of the solution volume. Dimensions are given in meters. The quantity k_{max} in Line 4 is the number of nodes along the horizontal direction. Lines 5-7 describe the vertical axis (y or r). Line 8 contains the quantity $DUnit$, the conversion factor from dimensions used in **Mesh** to meters. Line 9 gives the number of regions in the solution, while Line 10 specifies the symmetry (0: planar, 1: cylindrical).

The node section consists of 4 title lines and $k_{max} \times l_{max}$ data lines, one for each node of the solution space. Each data line contains the following quantities:

- The indices of the node (K, L)

- The region number of the node ($RgNo$) and region numbers for two associated elements ($RgUp$ and $RgDn$). The upper element lies above the line between nodes (K,L) and $(K+1,L)$ and the lower element lies below the line.
- The coordinates of the node in meters, (x,y) or (z,r) .
- The vector potential A_z (tesla-m) or stream function ψ (tesla-m²) at the node.
- Value of relative magnetic permeability for Axis 1 and Axis 2 in the upper and lower elements at the node.

The node section has the following appearance:

```

--- Nodes ---
  k   1  RgNo RgUp RgDn      x          y          Pot
=====
  1   1    8   1   0 -1.000000E-01 -5.000000E-02  0.000000E+00
  2   1    8   1   0 -9.878124E-02 -5.000000E-02  7.378424E-05
  3   1    8   1   0 -9.780469E-02 -5.000000E-02  1.329054E-04
  4   1    8   1   0 -9.679882E-02 -5.000000E-02  1.938010E-04
  5   1    8   1   0 -9.580030E-02 -5.000000E-02  2.542518E-04

          Mu1Up      Mu1Dn      Mu2Up      Mu2Dn
=====
  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00
  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00
  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00
  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00

```

The region section consists of four title lines following by $NReg$ data lines, one for each region. An entry "1" in the second column designates a fixed-potential region (electrode), while "1" in the third column designates an anisotropic material. The recorded region values are μ_r along Axis 1 and Axis 2 for linear materials, the angle associated with anisotropic materials or permanent magnetics, the total region current, the fixed vector potential, the cross-section area and the coercive magnetic flux density. The region section has the following appearance:

--- Regions ---

RegNo	Fix	Aniso				Mu1	Mu2	RegAngle
1	0	0	0	0	0	1.000000E+00	1.000000E+00	0.000000E+00
2	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00
3	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00
4	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00
5	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00
6	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00
7	0	0	0	0	0	1.000000E+03	1.000000E+03	0.000000E+00

RegCurrent	Potential	Area	Bc
0.000000E+00	0.000000E+00	1.357824E-02	0.000000E+00
0.000000E+00	0.000000E+00	3.199844E-03	0.000000E+00
0.000000E+00	0.000000E+00	3.200132E-03	0.000000E+00
0.000000E+00	0.000000E+00	4.999999E-06	0.000000E+00
0.000000E+00	0.000000E+00	5.000001E-06	0.000000E+00
0.000000E+00	0.000000E+00	4.999999E-06	0.000000E+00
0.000000E+00	0.000000E+00	5.000000E-06	0.000000E+00

5 TriComp analysis functions

To create plots and to perform numerical calculations, click the *Analyze* command in the main menu and choose a solution file. The *Analysis* menu contains the following main entries: *File*, *Plots*, *Analysis*, *Scans*, *Export* and *Return*. The commands of the *Export* menu (which generate hardcopy output and plot files) are similar to those in **Mesh**. The *Return* command restores the main menu where you can run additional solutions. This chapter gives a general description of post-processing capabilities for all **TriComp** programs. The following chapter covers specific plot quantities and analysis capabilities of **EStat**.

5.1 File menu commands

LOAD SOLUTION FILE

Read a different solution file for analysis without returning to the main menu. Pick a new file in the dialog. Changing the directory in the dialog changes the program working directory.

LOAD CONFIGURATION FILE

A configuration file is a text file of information that controls operation in the *Analysis* menu. The file defines quantities for plots, interpolations, volume integrals and surface integrals. The program loads a default configuration the first time you run it. This file contains a broad spectrum of useful quantities for the solution type. Some programs may automatically switch between two default configurations, depending on the solution type (*e.g.*, dielectric or conductive solutions in **EStat**). The default quantities are sufficient for most applications. Use this command if you want to load a specialized configuration or one that you have prepared. Chapter 7 gives detailed information on the format of configuration files and how to prepare them.

LOAD FROM SERIES

Initial-value programs like **TDiff** may produce several solutions files in a run at different simulation times. Use this command to load other files in a series of solutions. The program displays the dialog of Fig. 21 which lists the full set of output files created by the run along with the simulation times. Pick the desired file and click *OK* to load the solution. Note that this command does not appear in the menus of boundary-value programs like **EStat**.

OPEN DATA RECORD

Commands such as *Point calculation* and *Line scan* generate quantitative information. You can automatically record the data generated during an analysis session by opening a data file. Pick a file in the dialog or accept the default. The text-format data file will be stored in the working directory. You can use an editor to view the file or to extract information to send to mathematical analysis programs or spreadsheets. The suggested suffix for data records is **DAT**.

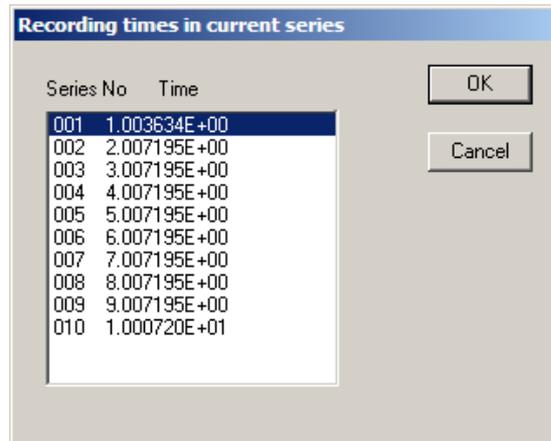


Figure 21: Dialog to pick a solution in a series generated by an initial-value program.

CLOSE DATA RECORD

Close the current data file. Use this command if you want to start a new file. You must close the data file before opening it with the internal editor.

RUN SCRIPT

A analysis script allows you to perform complex or repetitive calculations on a set of similar solutions. This command displays a dialog listing files with the suffix `SCR`. Pick a file and click *OK*. The script can load data files, open and close data records, and perform any of the quantitative analysis functions described in this chapter. The script command language is described in Sect. 5.7. Note that the analysis script must be in the same directory as the data files.

CREATE SCRIPT

Use this command to create an analysis script with the internal text editor. Supply a file prefix *S**Prefix* in the dialog – the resulting script will be saved with the name *S**Prefix*.`SCR`. The program opens the file in the editor and writes a reference list of allowed commands. The list follows the *EndFile* command and will be ignored by the script parser. Enter commands above the *EndFile* command.

EDIT FILES

Use this command to view or to modify an existing file with the internal program editor. Use the dialog to choose a text file. Changing directories in the dialog does not change the working directory of the program. You must exit the editor to resume normal operation of the program.

EXIT

Exit the program.

5.2 Plot menu commands

Spatial plots show variations of quantities over the two-dimensional space of the simulation. The following plot types are available:

- **Mesh.** Element facets of the computational mesh.
- **Region.** Computational mesh with elements color-coded by region number.
- **Filled contour.** Discrete bands of color coded according to values of the current plot quantity.
- **Contour lines.** Lines that follow constant values of the current plot quantity.
- **Element.** Elements of the solution space color-coded by values of the current plot quantity.
- **Surface.** A three-dimensional plot representing the current quantity as height over a region in the x - y or z - r plane. The spatial limits of the plot correspond to the current view window for *Mesh*, *Region*, *Filled contour*, *Contour* or *Element* plots.

The *Settings* popup menu contains the following commands.

PLOT TYPE

Change the plot type.

PLOT QUANTITY

A dialog shows a list of the available quantities defined in the INTERPOLATION section of the current configuration file.

PLOT LIMITS

In the default *Autoscale* mode the program picks limits in *Filled contour*, *Contour lines*, *Element* and *Surface* plots that span the full range of the current quantity. With this command you can set specific limits. In the dialog uncheck the *Autoscale* box and supply minimum and maximum values. Note that the program does not check the consistency of the values. Check the box to return to *Autoscale* mode.

TOGGLE GRID DISPLAY

Use this command to activate or to suppress the display of grid lines in *Mesh*, *Region*, *Contour lines*, *Filled contour* and *Element* plots. Grid lines corresponding to the axes ($x = 0.0$ or $y = 0.0$) are plotted as solid lines.

GRID CONTROL

This command displays the dialog of Fig. 22 to set properties of the grid. In the default *Autoscale* mode, the program automatically chooses intervals and positions so that lines occur at convenient values of x or y (for example, 0.01 rather than 0.01153). The grid intervals change as the view is zoomed. To set the grids manually, uncheck the *Autoscale* box and enter values for the intervals in x and y .

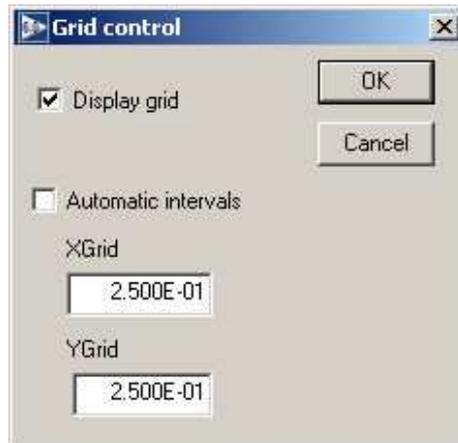


Figure 22: Grid control dialog

MOUSE/KEYBOARD

By default the program uses mouse entry of coordinates for commands like *Line scan* and *Zoom*. This command toggles between mouse and keyboard input. Enter keyboard coordinates in the distance units used in **Mesh**. In other words, if the solution program has $DUnit = 1.0 \times 10^6$, then enter dimensions in microns. You can temporarily switch to keyboard entry when entering coordinates with the mouse by pressing the *F1* key.

TOGGLE SNAP MODE

When snap mode is active, the mouse returns the coordinate values closest to an integer multiple of the quantity $DSnap$. In other words, if $DSnap = 0.5$ and the mouse position is (5.4331, -2.6253), the returned coordinates are (5.500, -2.500). Exercise caution when using commands like *Point calculation*. If snap mode is active, the calculation is performed at the snap point rather than the current location of the mouse cursor. You can also toggle the snap mode by pressing the *F2* key during coordinate entry.

SNAP DISTANCE

Set the distance scale $DSnap$ for the mouse snap mode.

TOGGLE ELEMENT OUTLINE

This command determines whether element facets are displayed.

TOGGLE FIXED POINT DISPLAY

In the default mode, the programs generate contour line, filled contour and element plots from element information. Therefore, isolated nodes (representing structures like fixed-potential grids or sheets) do not appear. In response to this command, the program plots circles at fixed-potential nodes that are surrounded by material elements.

CONTOUR STYLE

This command is active only when the current plot type is *Contour lines*. There are four

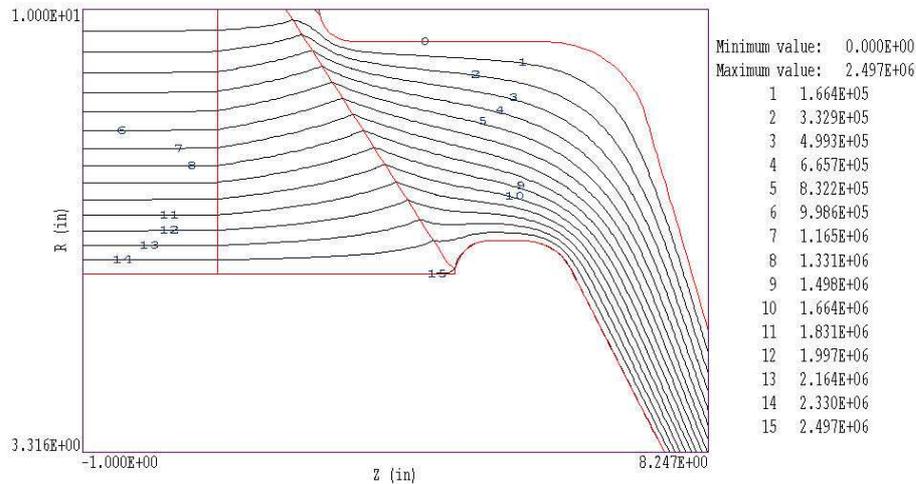


Figure 23: Monochrome contour plot with labels

choices: monochrome, monochrome with labels, colored and colored with labels. In the colored mode, the lines are color-coded according to the value of the plotted quantity. A legend is included in the information window to the right of the plot. In the labeled modes, contour lines are numbered according to their values (Fig. 23). Overlapping labels on closely-spaced lines may look better in a zoomed view.

NUMBER OF CONTOURS

Change the number of plotted contour lines. This command is active only when the current plot type is *Filled contour* or *Contour lines*.

TOGGLE NUMBER FORMAT

The program automatically chooses real-number notation for axis labels when the range of dimensions is close to unity. Use this command to switch to scientific notation.

The following commands (described in the **Mesh** manual), change the view limits in *Mesh*, *Region*, *Filled contour*, *Contour line* and *Element* plots. The current view limits of the two-dimensional plots are used when creating three-dimensional *Surface* plots.

ZOOM WINDOW

ZOOM IN

EXPAND VIEW

GLOBAL VIEW

PAN

The following commands control the appearance of *Surface* plots. The commands are active only when a *Surface* plot is displayed.

ROTATE 3D IMAGE

Rotate the *Surface* plot by 90° in the spatial plane.

VIEW ANGLE 3D

Set the elevation angle for the view point.

SET GRID 3D

Change the resolution of the surface plot. To create a *Surface* plot, a quantity is mapped to a rectangular grid with dimensions $N_x \times N_y$. The numbers also determine the total number of grid lines in the *Surface* plot. The default values are $N_x = N_y = 75$.

5.3 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 zoom limits, plotted quantity and plot type (spatial or scan) are saved. 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 either a spatial or scan plot.

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

```
Program: TriComp
PlotStyle: Spatial
Outline: ON
Grid: ON
Scientific: OFF
FixedPoint: OFF
Vectors: OFF
XGMin: -1.000000E+00
XGMax: 1.000000E+01
YGMin: 0.000000E+00
YGMax: 1.000000E+01
PlotType: Elem
NPQuant: 1
...
```

If a specific solution file is loaded, the plot will be restored exactly. The saved view feature has two useful features if a different mesh is 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 plot type or quantity) are applicable to any solution, but others (like zoom limits) depend on the geometry. The program 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, the programs use 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.

5.4 Analysis menu commands

The commands in the *Analysis* popup menu generate numerical data. Most of the functions require coordinate input from the user, usually through the mouse. Therefore, the analysis menu is active only when a *Mesh*, *Region*, *Contour*, *Element* or *Vector* plot is displayed.

POINT CALCULATION

Click on the command and move the mouse cursor to a position inside the solution volume. Current coordinates are displayed at the bottom of the screen. Press *F2* if snap mode is active and you want to specify an intermediate point. Press *F1* to type in the coordinates. The program writes a subset of quantities to the information area below the plot. If a data file is open, the program also records complete information on all quantities defined in the *Interpolation* section of the configuration file.

LINE SCAN

After clicking on the command, supply two points with the mouse or keyboard to define a scan line. The snap mode is useful in this application (for example, you may want a 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 at equal intervals along the line. The program makes a screen plot of the currently-selected quantity versus distance along the scan and activates the commands in the *Scan plot* menu (Sect.5.5). The information is recorded if a data file is open. The default is to include all computed quantities. Use the *Set recorded quantities* command to limit the information.

VOLUME INTEGRALS

Determine volume integrals of quantities defined in the **VOLUME** section of the configuration file and record the results. The program will prompt if a data file is not currently open. The program first records the global and regional volumes (or areas for planar solutions) and then integrals of defined quantities organized by region.

SURFACE INTEGRALS

Calculate surface integrals of quantities defined in the **SURFACE** section of the configuration

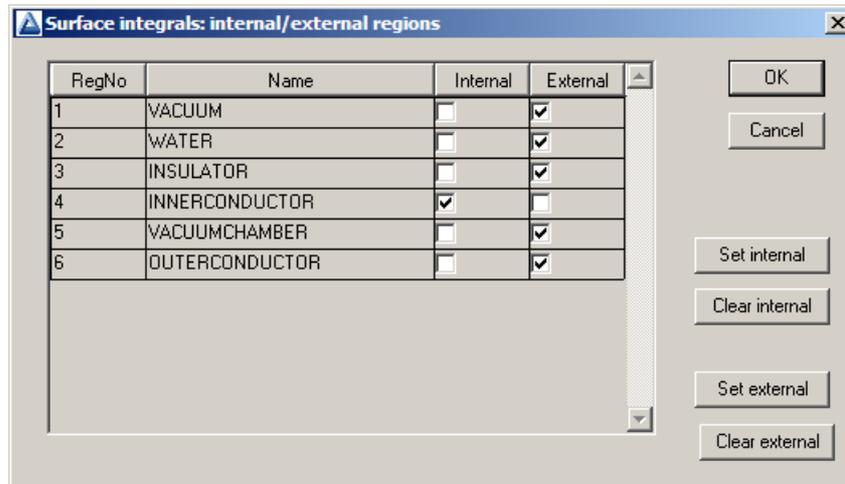


Figure 24: Surface integral dialog

file over region boundaries. In response to the command, the program displays the dialog of Fig. 24. Pick one or more regions as internal regions and bounding regions as external. The program computes the integral over the surfaces of the internal region(s) in contact with the external regions.

LINE INTEGRALS

Take integrals of vector quantities specified in the **SURFACE** section of the current configuration file over a line in the solution volume. Specify the line with the mouse or keyboard as in the *Line scan* command. The program reports both parallel and normal integrals in the data file (*i.e.*, the vector quantity is taken both parallel and normal to the differential line segments).

MATRIX FILE

The program can write matrix files of field values to help you create your own analysis routines. Although information is available in the output file of the solution program, it may be difficult to deal with the conformal triangular mesh. The *Matrix file* command uses the interpolation capabilities of the program to create a text data file of field quantities on a rectangular grid in x - y or z - r . The command displays the dialog of Fig. 25. Here you can set the matrix file name, the dimensions of the box and the number of intervals along x and y (or z and r). The program creates the file in the current directory. The default is to include all computed quantities. Use the *Set recorded quantities* command to limit the information.

The *Analysis settings* popup menu contains the following entries.

INTERPOLATION METHOD

The default interpolation method for plots and analysis commands such as *Point calculation* and *Line scan* is a second-order least-squares fit with a selective choice of data points. For example, only points on the side of a material boundary that contains the target point are included to give the correct field discontinuity. The least-squares fit may fail in very small

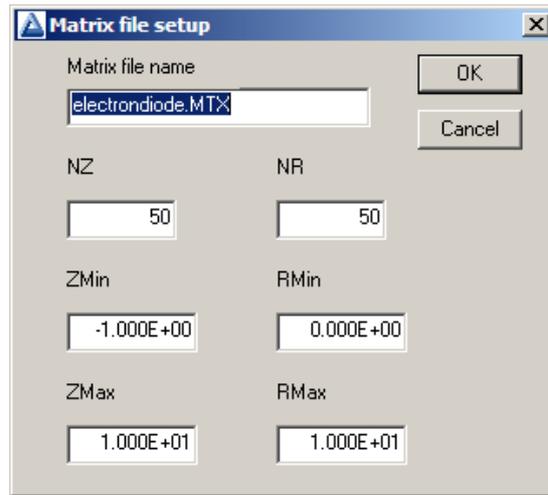


Figure 25: Dialog to generate a matrix file.

regions or enclosed areas if the program cannot identify enough data points. In this case, toggle to the linear mode. Here, field values are determined by a first order fit in the element that contains the target point. The status bar reports the current interpolation type.

SCAN PLOT QUANTITY

With this command you can choose the quantity to display in the screen plots of line scans. Pick the quantity from the list box and click *OK*. All quantities defined in the `INTERPOLATION` section of the current configuration file are available. This setting has no effect on the data file records which may include all field quantities.

NUMBER OF SCAN POINTS

This command sets the number of line scan points in plots and data file records. The default is 50 and the maximum number is 500.

SET RECORDED QUANTITIES

The default for data records of line scans and matrix files is to include all quantities defined in the `INTERPOLATION` section of the current configuration file. This state may result in large files with unnecessary information. Use this command to set the quantities that will be included. In the dialog of Fig. 26, activate the check box to include a quantity. There are a few rules:

- Quantity selection is available when the program runs interactively in a window. It does not apply when the program is run in the background. In this case, you can limit recorded quantities by editing the `INTERPOLATION` section of a custom configuration file.
- In the interactive mode, quantity selection applies to direct calls to the *Line scan* or *Matrix file* commands and indirect calls from an analysis script.
- All quantities are active when the program starts.
- The state of active quantities is preserved when a new data file is loaded.

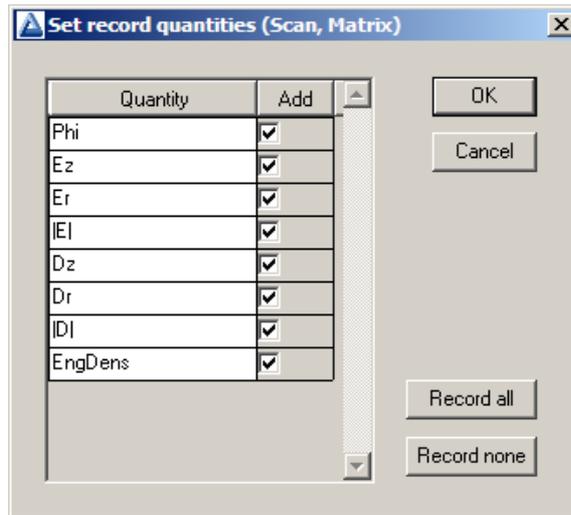


Figure 26: Dialog to set quantities recorded in response to the *Line scan* and *Matrix commands*.

5.5 Scan plot menu

The commands of the *Scan* menu become active following the *Line scan* command.

OSCILLOSCOPE MODE

In the oscilloscope mode, a scan plot assumes characteristics of a digital oscilloscope (Fig. 27). The program superimposes a cross-hair pattern on the graph. Plot values at the intersection are displayed in the information window. Move the marker along the plot by moving the mouse. If you click the left mouse button at a point, the program displays the plot values along with the numerical derivative and integral of the curve. The definite integral is taken from the left-hand side of the plot to the current point. Values are displayed on the screen and written to the data file if open. Press the right mouse button to exit the oscilloscope mode.

TOGGLE SCAN SYMBOLS

The setting determines whether plot symbols are added to the scan plot showing calculated points.

TOGGLE GRID

The setting determines whether grid lines are added to the screen and hardcopy scan plots.

CLOSE SCAN PLOT

Close the scan plot plot and return to the *Analysis* menu.

5.6 Vector tools

The analysis menu includes useful tools to display directional quantities in the VECTOR section of the current configuration file. Figure 28 shows the corresponding entries on the toolbar.

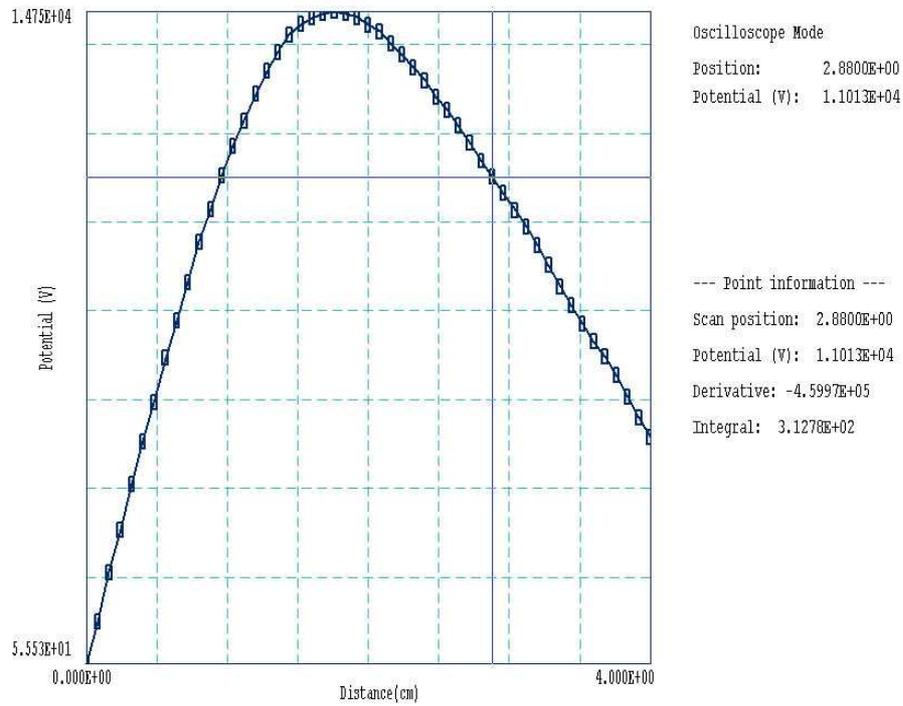


Figure 27: Scan plot in the oscilloscope mode

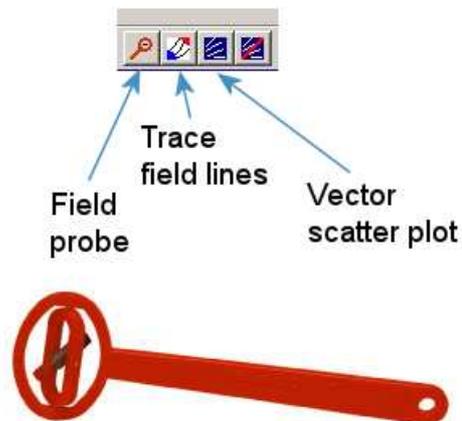


Figure 28: Tools to display vector information in the slice plot menu.

PROBE

This feature was inspired by the familiar Magnaprobe illustrated in Fig. 28. When you click on the tool and move the mouse cursor into the plot area, it changes to a semi-transparent probe that rotates about a pivot point to show the local direction of the vector quantity. The status bar shows the coordinates and the magnitude of the quantity.

FIELD LINES AT POINTS

Use this command to add lines of the vector quantity to any two-dimensional plot. The program enters coordinate mode when you click the command. Move the mouse to a point in the solution volume and click the left button. The program calculates and plots the path of the vector line that passes through the point. You may continue to add any number of lines. Click the right mouse button or press *Esc* to exit coordinate mode.

VECTOR SCATTER PLOT

Superimpose a uniform distribution of vector arrows pointing in the direction of the vector quantity. Vectors may be added to any of the two-dimensional plot types, including *Region*.

REMOVE VECTORS

Use this command to turn off the vector display. The program removes vector arrows and calculated field lines from the current plot.

PICK VECTOR QUANTITY

Choose a quantity defined in the VECTOR section of the current configuration file for display with the vector tools.

5.7 Analysis script commands

Scripts to control analysis sessions have a name of the form `FPREFIX.SCR`. They should be in the same directory as the data files. Scripts are text files that follow the **TriComp** syntax conventions. The program ignores blank lines and indentations. Data lines use the standard delimiters and comment lines begin with an asterisk [`*`]. Processing ends when the *EndFile* command is encountered.

To run a script, choose the *Run script* command in the *File* menu. The program shows a list of available scripts. Pick a file and click *OK*. The script operates on the presently-loaded data file or you can load other files from within the script. You may also sequentially open one or more data output files.

The program can perform analyses autonomously under script file control from the command prompt. If the file `GTEST.SCR` is in the data directory, then use a command of the form:

```
[ProgPath]\ProgName GTEST.SCR <Enter>
```

In the command prompt mode, the analysis functions of the program may be invoked from batch files, Perl or Python scripts or you own programs. The main application of the command prompt mode is to generate data files and to perform extended analyses under batch file control.

The following commands may appear in a script:

INPUT FileName**INPUT Switch1.EOU**

Close the current data file and load a file for analysis. The parameter is the full name of the data file. You can load several files for sequential analysis.

OUTPUT FileName**OUTPUT SW02.DAT**

Close the current data file and open an output file **FileName**.

CONFIGURATION [datapath\] FileName**CONFIGURATION = C:\FieldP\TriComp\ESTAT_FORCE.CFG**

Load a new 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 during the last run.

SCAN Xs Ys Xe Ye**SCAN Zs Rs Ze Re****SCAN = (0.00, 0.00) (10.00, 0.00)**

Write the results of a line scan between the specified points to the data file. The four real number parameters are the starting and end coordinates in **Mesh** units.

GENSCAN

Perform any number of calculations along an arbitrary path in space. At each point, write values of 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
  xp2 yp2
  ...
  xpn ypn
END
```

Each data line contains two real numbers separated by spaces to define a point in the solution volume (x - y or z - r). Enter coordinates in **Mesh** units.

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 current data file. If a region number does not appear, integrals are taken over all regions in the solution volume. Otherwise, the integral extends over elements with region number *NReg*. If quantities in the **VOLUME** section of the configuration file have units of C/m^3 , then the output quantities will have units of C for cylindrical solutions and C/m for planar calculations.

SURFACEINT Reg1 Reg2 ... RegN**SURFACEINT 5 7 -9 -12**

Perform a surface integral of vector quantities defined in the **SURFACE** section of the configuration file and write the results to the data file. Positive integer values define the set of *Internal* regions and negative values define the *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 the program does not include facets on the boundary of the solution volume in surface integrals.

MATRIX FName Nx Ny Xs Ys Xe Ye**MATRIX FPrefix Nz Nr Zs Rs Ze Re****MATRIX = Switch1 (10, 20) (0.00, 0.00, 5.00, 10.00)**

Open a matrix file and record values. The command requires seven parameters: 1) The name of the matrix file (string), 2) the number of intervals along the x or z direction (integer). 3) the number of intervals along the y or r direction (integer), 4-7) coordinates of the corners of a box in the solution volume (real). The suggested suffix for matrix files is **MTX**.

NSCAN NScan**NSCAN = 150**

Set the number of points in a line scan. The default is 50 and the maximum number is 500.

INTEPOLATION [LSQ,LINEAR]**INTERPOLATION = Linear**

Set the interpolation method for subsequent *Point*, *Line scan* and *Matrix* commands. The options are *LSQ* (least-squares fit) and *Linear*.

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. 5.3). 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 execution of the script. You may add descriptive text in any format after this command.

The following is an example of a script to compare field values along the axes of four different solutions and to write the results to the file `COMP.DAT`.

```
NSCAN 200
OUTPUT COMP.DAT
INPUT SWITCH01.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH02.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH03.EOU
SCAN 0.00 -50.00 0.00 50.00
INPUT SWITCH04.EOU
SCAN 0.00 -50.00 0.00 50.00
ENDFILE
```

Table 9: Calculated quantities in `permag_standard.cfg`.

Section	Quantity	Units
Interpolation	A_z or rA_θ (vector potential)	T-m or T-m ²
	B_x or B_z (magnetic flux density)	T
	B_y or B_r (magnetic flux density)	T
	$ \mathbf{B} $ (flux density magnitude)	T
	H_x or H_z (magnetic field)	A/m
	H_y or H_r (magnetic field)	A/m
	$ \mathbf{H} $ (magnetic field magnitude)	A/m
	μ_r (relative permeability)	
Vector	\mathbf{B} (magnetic flux density)	T
	\mathbf{H} (magnetic field)	A/m
Volume	U (field energy density)	J or J/m
	F_x or F_z	N or N/m
	F_y or F_r	N or N/m
Surface	F_x or F_z	N or N/m
	F_y or F_r	N or N/m

6 Standard PerMag calculated quantities

The standard configuration file `permag_standard.cfg` is included with the **PerMag** package. Table 9 shows quantities defined in the file. There are some restrictions:

- The volume integral of magnetostatic field energy applies only to linear materials. Energy quantities for non-linear materials and permanent magnets are tabulated in the **PerMag** listing file (`FPREFIX.PLS`).
- The volume integral of force has non-zero values only for current-carrying regions.
- The surface integral of force applies to an assembly of regions surrounded by air/vacuum elements ($\mu_r = 1.0$). The internal regions may have any material characteristics, including nonlinear iron and permanent magnets.
- Special quantities have been included as comment lines. The torque T_z is meaningful only for planar solutions. The origin for torque calculations is given by the variables `$xt` and `$yt` (with dimensions in meters). For a fixed origin, define the variables in the `PROPARAM` section. To enable user setting of the origin, include the variables in the `RUNPARAM` section.

6.1 PerMag variables

PerMag supplies values of the following field quantities at the current location for use in expressions of the **INTERPOLATION**, **VECTOR**, **VOLUME** and **SURFACE** sections of the configuration file. Values are calculated using the current interpolation method (least-squares fit or linear).

- **&A**: the vector potential A_z in tesla-m for planar solutions and stream function rA_θ in tesla-m² for cylindrical solutions.
- **&Bxz**: the x (planar) or z (cylindrical) component of magnetic flux density in tesla.
- **&Eyr**: the y (planar) or r (cylindrical) component of magnetic flux density in tesla.
- **&BMag**: the magnitude of the magnetic flux density in tesla.
- **&Hxz**: the x or z component of the magnetic field in A/m.
- **&Hyr**: the y or r component of the magnetic field in A/m.
- **&HMag**: The magnitude of the magnetic field in A/m.
- **&Mu1**: The relative magnetic permeability for isotropic materials and the value along the primary axis for anisotropic materials.
- **&Mu2**: The relative magnetic permeability along the normal axis for anisotropic materials. Equal to **&Mu1** for isotropic materials.

The following region variables are available for use in expressions. The value corresponds to the region that contains the current location.

- **\$RegMu1**: the constant value of relative magnetic permeability along the primary axis of anisotropic materials
- **\$RegMu2**: the constant value of relative magnetic permeability along the normal axis of anisotropic materials
- **\$RegAng**: the angle (in degrees) of the primary axis (relative to the x or z axis) for anisotropic materials.
- **\$RegSinAng**: the sine of the angle of the primary axis for anisotropic materials.
- **\$RegCosAng**: the cosine of the angle of the primary axis.
- **\$RegCurrent**: the total current I_z or I_θ of a region in A.
- **\$RegVecPot**: the vector potential A_z or stream function rA_θ for fixed regions.
- **\$RegArea**: the cross section area of the region in m^2 .

6.2 Standard configuration

The configuration file `permag_standard.cfg` (Table 10) is loaded by default when **PerMag** first runs. The two program parameters are μ_0 and $1/\mu_0$. Several quantities in the `INTERPOLATION` section are simply the special variables generated by **PerMag** (A, B_x, \dots). Note the command form for defining magnetic flux density components:

```
Bx/Bz = &Bxz
```

The value is given by the variable `&Bxz`. The name that will be displayed in plots has two forms separated by a slash. The first form is used if the solution has planar symmetry while the second form applies to cylindrical solutions.

The vector tools can display either \mathbf{B} or \mathbf{H} . Note that the plots will differ only if the solution includes anisotropic materials. The volume integral of the field energy density gives the total energy of regions, a useful quantity to determine inductance. The field energy is the volume integral of the quantity,

$$u = \frac{\mathbf{B} \cdot \mathbf{H}}{2}. \quad (21)$$

The expression holds for anisotropic as well as isotropic materials. Units are J/m for planar solutions and J for cylindrical solutions. The volume integral to determine force from the Lorentz expression has a non-zero value only for regions that carry current. The surface integrals of force employ the Maxwell stress tensor. They give the magnetic force in N/m or N on any assembly of iron, coils or permanent magnets surrounded by air ($\mu_r = 1.0$). See the following tutorials for a derivation of the expressions:

```
http://www.fieldp.com/documents/stresstensor.pdf  
http://www.fieldp.com/documents/forceandtorque.pdf
```

The `SURFACE` section of the configuration file contains two additional expressions that can be activated by removing the comment symbol (asterisk):

```
TzSurf =  
  $x $xt - &Byr &Bxz * * $y $yt - &Bxz 2 ^ &Bmag 2 ^ 2.0 / - * - $IMu0 *;  
  $x $xt - &Byr 2 ^ &Bmag 2 ^ 2.0 / - * $y $yt - &Bxz &Byr * * - $IMu0 *  
  
IEnclosed = &Bxz $IMu0 *;&Byr $IMu0 *
```

The torque expression applies to planar solutions (the torque on objects in a system with cylindrical symmetry is identically zero). The unit is N-m/m. Note that the quantity is calculated with respect to an origin at ($\$xt, \yt). The origin variables (in meters) must be defined in either the `PROGPARAM` or `RUNPARAM` sections. The final expression is for use with the *Line integral* command. The right hand side is the vector \mathbf{B}/μ_0 . The parallel components of a set of line integrals in air that define a circuit give the total enclosed current according to Ampere's law.

Table 10: Contents of the file permag_standard.cfg.

```

PROGPARAM
  $Mu0 = 1.256637E-6
  $IMu0 = 7.957748E5
END
RUNPARAM
END
REALTIMEPARAM
* Origin for torque calculations (in meters)
* $xt = 0.00
* $yt = 0.00
END
INTERPOLATION
  Az/rAThet = &A
  Bx/Bz = &Bxz
  By/Br = &Byr
  |B| = &BMag
  Hx/Hz = &Hxz
  Hy/Hr = &Hyr
  |H| = &HMag
  MuR = &Mu1
END
VECTOR
  BVect = &Bxz;&Byr
  HVect = &Hxz;&Hyr
END
VOLUME
  Energy = &Bxz &Hxz * &Byr &Hyr * + 0.5 *
  FVolX/FVolZ = $RegCurrent &Byr * $RegArea / @CHS
  FVolY/FVolR = $RegCurrent &Bxz * $RegArea /
END
SURFACE
  FxSurf/FzSurf = &Bxz 2 ^ &Byr 2 ^ - $IMu0 * 0.5 *;&Bxz &Byr * $IMu0 *
  FySurf/FrSurf = &Byr &Bxz * $IMu0 *;&Byr 2 ^ &Bxz 2 ^ - $IMu0 * 0.5 *
END

ENDFILE

```

6.3 Magnetic force calculations

Using quantities in the standard configuration file, **PerMag** can perform two types of calculations to find magnetic forces on objects in the solution space.

- Volume integrals of the Lorentz force density in regions with non-zero values of current.
- Surface integrals of the Maxwell stress tensor for all types of regions.

The first method applies only to drive coil regions, while the second is valid for any region. There are two ways to initiate a surface integral:

- Use the *Surface integral* command and specify sets of internal regions and external air regions to define a boundary.
- Use several instances of the *Line integral* command to define a path that encircles the object of interest.

The second method is more accurate if the surface of an iron or permanent-magnet region has sharp edges with field discontinuities. Alternately, you can define a special air region that surrounds the object and perform the integral over this boundary where field variations are smoother. In this case, the internal region set includes the object and the air region.

A surface integral of the Maxwell stress tensor represents the sum of forces on all currents (both applied and material) in the enclosed volume [see **Field Solutions on Computers** (Sect. 10.5)]. The procedure is invalid if the integral is taken over a region that shares a boundary with a permanent magnet or a region with $\mu_r \neq 1.0$. In this case, the integral encloses the material current of the second region along the shared boundary. The resulting values will represent the force on the first region plus a portion of the force on the second region. You must plan solutions around this limitation. The paths of line integrals must pass through air/vacuum regions.

The `FORCE_CALCULATION` example illustrates the meaning of force quantities and typical accuracy levels for the different methods. The geometry of the planar simulation is illustrated in Fig. 29. Two copper rods ($\mu_r = 1.0$) each carry a uniformly-distributed current of 1000 A in the $+z$ direction. The 2.0 cm diameter rods have centers at positions (-2.50, 0.00) and (2.50, 0.00). A composite structure of ferromagnetic materials has the shape of a cylinder of diameter 4.0 cm with center at position (1.00, 5.00). The relative magnetic permeability is $\mu_r = 500.0$ on the left side and $\mu_r = 10.0$ on the right side. Figure 29 shows calculated magnetic field lines in the central region of the solution volume.

Table 11 shows the **PerMag** analysis script `FORCE_CALCULATION.SCR`. In response to the *VolumeInt* commands, **PerMag** carries out a Lorentz force integrals for Regions 2 and 3 (left and right copper rods). The *SurfaceInt* commands take Maxwell stress tensor integrals around the left rod (Region 2), the right rod (Region 3) and the iron assembly (Regions 4 and 5). The four line integrals define a path that encompasses the composite iron structure.

Force calculation results are listed in Table 12. The volume and surface integrals for the force on the rod differ by about 2.2%. Note that the accuracy of surface integrals is strongly dependent on the element size. The total force on the rods is approximately equal in magnitude and opposite in direction to the total force on the iron. We can apply analytic results to check the validity of the calculation. If we remove the iron pieces and average over the radius of the

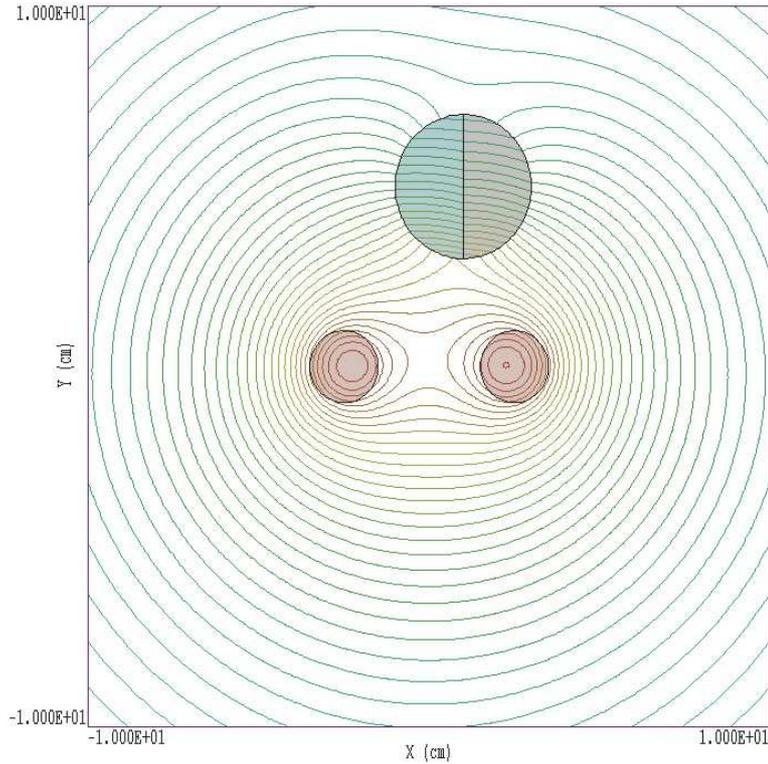


Figure 29: Central portion of the solution volume for example FORCECALC. Current-carrying rods shown in tan, composite iron structure at the top. Dimensions in cm.

rod, the force per length on one rod carrying current I from another rod with the same current at a spacing L is approximately equal to:

$$F_x = \frac{\mu_0 I^2}{2\pi L} \tag{22}$$

The values $I = 1000.0$ A and $L = 0.05$ m give $F_x = \pm 4.00$ N/m. If we set $\mu_r = 1.0$ for the iron regions and repeat the calculation, the Lorenz methods gives rod forces of $+4.0009$ N/m and -4.0010 N/m.

The SOLENOID_FORCE example illustrates how to interpret force quantities in a cylindrical calculation. The system (illustrated in Fig. 30) consists of two solenoids and an external iron pipe ($\mu_r = 500.0$) to carry the return flux. The solenoids have inner radius 4.50 cm, outer radius 5.50 cm and length 15.0 cm. They are separated along z by a 1.0 cm gap. Each coil carries current 15.0 kA-turn. The iron shield localizes the magnetic field so it is not necessary to use extended boundaries. The volume integral method gives the following forces on coils:

Force on coils by region		
NReg	ForceZ (N)	ForceR (N)
2	2.919E+01	2.606E+02
3	-2.919E+01	2.606E+02

Table 11: Analysis script FORCE_CALCULATION.SCR

```

* Region      Region
* number      name
* =====
*      1      AIR
*      2      LEFTWIRE
*      3      RIGHTWIRE
*      4      IRONLEFT
*      5      IRONRIGHT
*      6      BOUNDARY
OUTPUT FORCE_CALCULATION.DAT
INPUT FORCE_CALCULATION.POU
VOLUMEINT 2
SURFACEINT 2 -1
SURFACEINT 3 -1
SURFACEINT 4 5 -1
* Outline iron regions
LINEINT -2.0 2.0 4.0 2.0
LINEINT 4.0 2.0 4.0 8.0
LINEINT 4.0 8.0 -2.0 8.0
LINEINT -2.0 8.0 -2.0 2.0
ENDFILE

```

Table 12: Results for the FORCE_CALCULATION example

Method	Object	F_x (N/m)	F_y (N/m)
Lorenz force integral	Left rod	4.6604	0.3094
	Right rod	-4.6787	0.7737
	Both rods	-0.0183	1.0833
Region surface integral	Left rod	4.5740	0.3020
	Right rod	-4.5631	0.7563
	Iron sum	0.0189	-1.059067
Enclosing line integral	Iron structure	0.0193	-1.0561

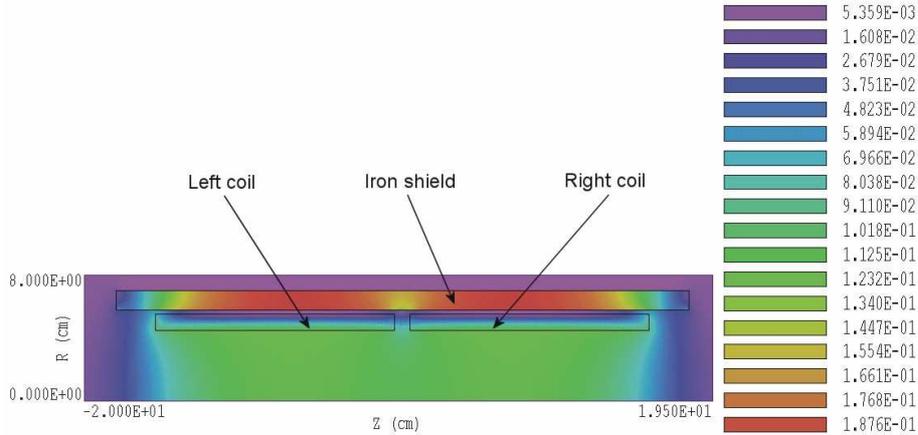


Figure 30: Geometry and values of $|\mathbf{B}|$ for the SOLENOIDFORCE example. Dimensions in cm.

The surface integral method gives values that agree to within 2.6%. The attractive rigid-body force on each coil has magnitude 29.19 newtons. Because the system is symmetric in θ , there can be no net force transverse to z . Instead, the quantity $ForceR$ represents the hoop stress on the coil, the net inward or outward force in the radial direction. We can understand the quantity by making a comparison to theory. Using the *Point analysis* command in **PerMag**, we find that the field outside the solenoid is close to zero and that the value at the inner radius is $B_0 = 0.122$ tesla. The magnetic pressure on the inner surface is given by:

$$p_m \cong \frac{B_0^2}{2\mu_0}, \quad (23)$$

so that $p_m = 5903$ N/m². The inner surface of each solenoid has area $A = 4.24 \times 10^{-2}$ m². Multiplying pressure by the area gives a total radial expansive force of 250.4 N, close to the code calculation. The Maxwell stress tensor integral over the iron shield surface gives approximately zero force in z and a compressive radial force of -28.2 newtons.

The third example, PMAGFORCE, demonstrates a force calculation for a permanent-magnet material and the use of a diagnostic surface for a Maxwell stress tensor integral. Figure 31 shows the geometry. We want to find the force between a cylindrical permanent magnet and an iron plate. The magnet (with 2.0 cm radius and 2.0 cm length) is located a distance 2.0 cm from the plate. The boundaries are well removed from the objects so that they do not contribute to the force calculations. We model the permanent magnet as a straight-line material with $B_r = 1.2$ tesla. (Note that the force-calculation method holds for all permanent magnet models because the Maxwell stress tensor depends only on fields external to the material.) An additional diagnostic region with the properties of air ($\mu_r = 1.0$) surrounds the permanent magnet.

We can use the *Surface integral* command in the interactive *Analysis* menu to find the force on the iron plate. After issuing the command, set the internal region to iron (Region 4) and the external to air (Region 1). The routine displays the following result and records it in a data file file: $F_z = -17.943$ N. Now set the internal region set to include the permanent magnet (Region 3) and the diagnostic region (Region 2). The axial force is $F_z = 17.977$ N. The force magnitudes agree to within 0.2%. We could also use the *Line integral* command along a plane at $z = 0.0$ cm from the axis to the radial boundary. Because of the Dirichlet boundary, all flux crosses this line and the integral gives the total force between the permanent magnet and iron

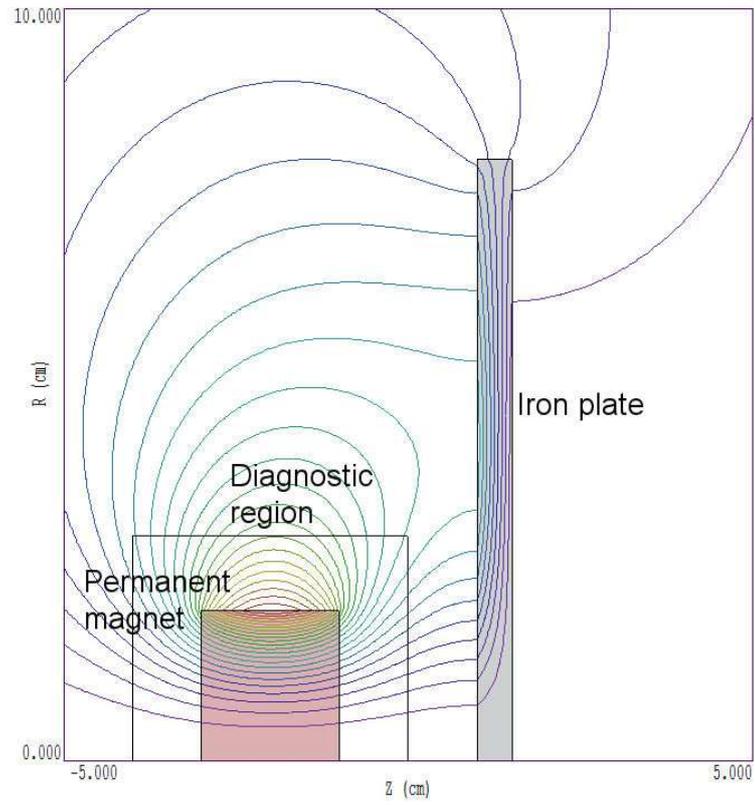


Figure 31: Portion of the solution volume for the PMAGFORCE example. Dimensions in cm.

plate. The result is $F_z = 17.960$ N.

7 Building custom analysis configurations

You can create configuration files to customize analysis operations for your application. You can then switch between configurations for different types of solutions. To build a configuration, copy and rename one of the standard files supplied with the program to act as a template. You can put your new configuration file anywhere, but we suggest you store all configurations in the same directory as the executable programs (`c:\fieldp\tricomp`). Note that the program remembers the last configuration used and attempts to reload it at the next run. This chapter describes the format and function of configuration files and how you can define calculated quantities.

7.1 Configuration file structure

An analysis configuration is a text file with a name of the form `FPREFIX.CFG`. Table 7.1 shows the contents of `estat_dielectric.cfg`. The file includes seven sections with the following functions:

- **PROGPARAM**: define fixed numerical parameters to use in the expressions of the **INTERPOLATION**, **VECTOR**, **VOLUME** and **SURFACE** sections.
- **RUNPARAM**: numerical parameters passed by the program (such as the RF frequency in **REF2** or the dump time in **TDiff**). **Do not change this section**. Doing so could cause errors reading solution files.
- **REALTIMEPARAM**: numerical parameters used in the expressions of the **INTERPOLATION**, **VECTOR**, **VOLUME** and **SURFACE**. The difference from a **PROGPARAM** is that these values may be changed by the user while the program is running. An example is the reference phase in **RFE2**.
- **INTERPOLATION**: definition of scalar quantities for plots and scans.
- **VECTOR**: definition of vector quantities for use with the vector tools.
- **VOLUME**: definition of scalar quantities used for volume integrals.
- **SURFACE**: definition of vector quantities for surface integrals.

You may define custom quantities for display and analysis in the last four sections. In plotting, run speed is a critical issue. The generation of a single element 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). The program 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 7.3 covers parameters and variables that may appear in expressions.

Table 13: Configuration file estat_dielectric.cfg.

```

PROGPARAM
  $Epsi0 = 8.854187E-12
END
RUNPARAM
  $CondFlag = 0.0
END
REALTIMEPARAM
END
INTERPOLATION
  Phi = &Phi
  Ex/Ez = &Exz
  Ey/Er = &Eyr
  |E| = &EMag
  Dx/Dz = $Epsi0 &Exz0 *
  Dy/Dr = $Epsi0 &Eyr0 *
  |D| = $Epsi0 &EMag0 *
  EngDens = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
END
VECTOR
  EVect = &Exz;&Eyr
  DVect = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
VOLUME
  Energy = &Exz &Exz0 * &Eyr &Eyr0 * + 0.5 * $Epsi0 *
  Charge = &Rho
END
SURFACE
  Charge = &Exz0 $Epsi0 *;&Eyr0 $Epsi0 *
END
ENDFILE

```

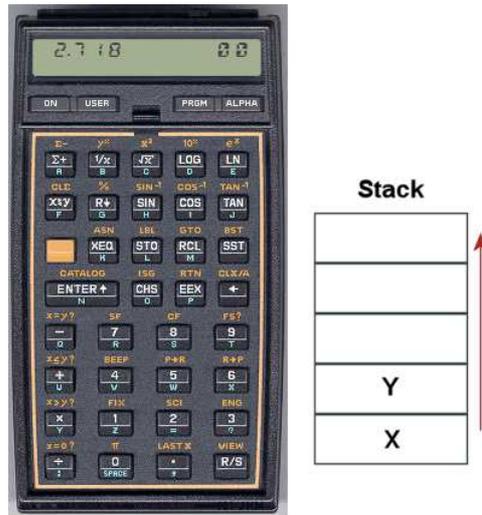


Figure 32:

7.2 Introduction to RPN

Consider the expression

$$5.0 \times 4.5^3 + \left[9.2 + 0.6 \times 0.9^{(2+0.67)} \right]. \quad (24)$$

Equation 24 could be represented in algebraic notation by the string:

`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. 32. 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(+): $Y + X$

Table 14: 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	$X = -X$	

- 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.

7.3 Operators, parameters and variables

Expressions to define quantities may contain numbers, unary and binary operations, parameters, standard variables and special variables for the program. Unary operators have names that begin with '@'. Table 14 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 -\$Epsi0 and -&grady[3] that mix a minus sign with a variable are invalid. Instead, use forms like \$Epsi0 @CHS.

You may define up to 10 constants in the PROGPARAM section that can be used in your expressions. For example:

```

PROGPARAM
  $Epsi0 = 8.854187E-12
  $Epsi0Inv = 1.1294E11
END

```

A data line contains a parameter name, an equal sign and a value in any valid real-number formal. A parameter name must begin with a dollar sign (\$) and may contain a maximum of 14 characters. Depending on the program, additional parameters may be defined in the RUNPARAM section:

```

RUNPARAM
  $CondFlag = 0.0
END

```

Here, a data line consists of a name, an equal sign and a default value. The actual value depends on the properties of the finite-element solution and is recorded in the output solution file. The parameter is set when a data file is loaded. Although you may use run variables in your expressions, the form of the RUNPARAM section should not be changed. Be sure that the form in a custom configuration is that same as that in the standard files.

The standard variables are

\$X, \$Y, \$Z, \$NREG, \$DUNIT

Note that the names of variables 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. The analysis program 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 **Mesh** 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 *
```

The special program variables give interpolated values of field quantities at the current location. As an example, the following quantities are available for use in expressions in **EStat**: **&Phi** (electrostatic potential), **&Exz** (the *x* or *z* component of electric field), **&Eyr** (the *y* or *r* component of electric field), **&EMag** (magnitude of the electric field), **&Exz0** (the *x* or *z* component of the normalized electric displacement, **D**/ ϵ_0), **&Dyr** (the *y* or *r* component of the normalized electric displacement), **&EMag0** (the magnitude of the normalized electric displacement), **&EpsSig1** and **&EpsSig2** (the relative dielectric constant or the electrical conductivity), and **&Rho** (the space-charge density for dielectric solutions). In addition, region variables may be defined. These are quantities that have fixed values over a region. The current value corresponds to the region that contains the current location. As an example, **EStat** supports the following region variables:

`$RegEps1` (the relative dielectric constant or electric conductivity along the primary axis for anisotropic materials), `$RegRho` (the space-charge density for uniform materials in dielectric solutions), `$RegPot` (the voltage of a fixed-potential region), `$RegEps2` (the relative dielectric constant or electric conductivity along the normal axis for anisotropic materials), `$RegAng` (the angle of the primary axis relative to the x or z axis for anisotropic materials), `$RegSinAng` and `$RegCosAng` (the sine and cosine of the angle of primary axis).

7.4 Building expressions

You can add your own expressions to the INTERPOLATION, VECTOR, VOLUME and SURFACE sections of the configuration file, one expression per line. Lines for a scalar quantities in the INTERPOLATION and VOLUME sections have three components:

- A name for display in plots and listings.
- An equal sign (=).
- A valid RPN expression.

For a quantity that has the same name in planar and cylindrical solutions, the name is a string up to 14 characters in length. Alternatively, the name may have the form:

`RectName/CylinName`

Here, two strings (each up to 14 characters in length) are separated by a slash (/). The string *RectName* is used for labels when a planar solution has been loaded, and the string *CylinName* is used for cylindrical solutions. For example:

```
FVolX/FVolZ = $RegCurrent &Byr * $RegArea / @CHS
```

Expression lines in the VECTOR and SURFACE sections have the following form:

```
NAME = EXPRESSION01;EXPRESSION02
```

A line consists of a name, an equal sign and two valid RPN expressions separated by a semicolon. The first expression gives the x or z component of the vector, while the second expression gives the y or r component.

It is best to build a configuration one expression at a time. Modify and save the file, then use the *Load configuration file* command. The routine will report the location of the first syntax error encountered. If the configuration file loads successfully, test the new expression using plot and point calculation commands. At shutdown, the program stores the current configuration file name in the registry and attempts to load it at the next session. If the file is missing or has a syntax error, the program displays the *Load configuration* dialog so that you can pick an alternative. An error message is displayed if the alternate file is invalid.

Index

- algebraic parser syntax, 40
- Analysis commands
 - 3D angle, 51
 - 3D grid, 51
 - Close data record, 47
 - Contour style, 49
 - Create analysis script, 47
 - Field lines at points, 57
 - Grid control, 48
 - Interpolation method, 53
 - Line integrals, 53
 - Line scan, 52
 - Load configuration file, 46
 - Load from series, 46
 - Load named view, 51
 - Load solution file, 46
 - Matrix file, 53
 - Number display format, 50
 - Number of contours, 50
 - Number of scan points, 54
 - Open data record, 46
 - Oscilloscope mode, 55
 - Pick vector quantity, 57
 - Plot limits, 48
 - Plot quantity, 48
 - Plot type, 48
 - point calculation, 52
 - Probe, 57
 - Remove vectors, 57
 - Rotate 3D, 50
 - Run analysis script, 47
 - Save named view, 51
 - Scan plot quantity, 54
 - Set recorded quantities, 54
 - Snap distance, 49
 - Surface integrals, 52
 - Toggle fixed point, 49
 - Toggle grid display, 48
 - Toggle keyboard, 49
 - Toggle outline, 49
 - Toggle snap mode, 49
 - Vector scatter plot, 57
 - Volume integrals, 52
- Analysis script
 - Configuration, 58
 - GenScan, 58
 - Input, 58
 - Interpolation, 59
 - Matrix, 59
 - NScan, 59
 - Output, 58
 - Plot, 59
 - Scan, 58
 - SurfaceInt, 59
 - VolumeInt, 59
- analysis script commands, 57
- analysis script example, 60
- Analysis script format, 57
- analysis, batch file control, 57
- analytic solutions, 6
- autoscale, 48
- batch operation
 - PerMag, 37
- boundary conditions, 10
 - symmetry, 16
- boundary solution technique, 35
- calculated quantities
 - standard variables, 74
- coercive field, 12
- comment lines, 27
- computational mesh, 6
- configuration file, 70
 - calculated quantities, 70
 - expressions, 75
 - load errors, 75
 - structure, 70
- conformal mesh, 6
- cubic spline, 32
- current
 - drive, 9
 - material, 9
 - permanent magnet, 9
- data export, 53
- delimiters, 26

- demagnetization curve, [12](#)
 - commercial examples, [12](#)
 - example, [33](#)
 - ideal, [12](#)
- Dirichlet boundary, [10](#)
- Dirichlet condition, [16](#)
- easy axis, [33](#)
- element, [6](#)
- energy product, [24](#)
- facet, [6](#)
- fiducials, [48](#)
- field
 - definition, [6](#)
 - equations, [6](#)
- field energy density, [63](#)
- Field index tool, [43](#)
- field line plots, [57](#)
- field probe, [57](#)
- finite-element method, [6](#)
 - basis, [7](#)
- force calculation methods, [65](#)
- free-space condition, [10](#)
- grids, [48](#)
- H-magnet, [16](#)
- interpolation methods, [54](#)
- keyboard/mouse control, [49](#)
- line integral, [53](#)
- magnetic circuit, [12](#)
- magnetic field
 - currents, [9](#)
 - force, [65](#)
- magnetic field lines, [10](#)
- magnetic flux, [9](#)
- magnetic flux density, [31](#)
 - relation to vector potential, [9](#)
- magnetic force
 - examples, [65](#)
- magnetic permeability, [31](#)
- magnetization curve, [31](#)
- magnetodynamic calculations, [5](#)
- mathematical functions, [41](#)
- matrix file, [53](#)
- Maxwell stress tensor, [65](#)
- microscopic solution, [34](#)
- mouse controls, [52](#)
- Mu table tool, [41](#)
- Neumann boundary, [10](#)
- Neumann condition, [16](#)
- node, [6](#)
- node equations, [6](#), [7](#), [20](#)
- non-linear materials, [31](#)
- number notation, [50](#)
- numerical solutions, [6](#)
- periodic permanent-magnet array, [21](#)
- PerMag
 - analysis screen, [20](#)
 - analysis script, [21](#)
 - boundary conditions, [10](#)
 - command format, [27](#)
 - command prompt, [37](#)
 - file types, [8](#)
 - finite-element solution, [20](#)
 - function, [4](#)
 - instructions, [39](#)
 - listing file, [24](#), [32](#), [34](#)
 - output file, [20](#)
 - output file format, [43](#)
 - Parallel, [30](#)
 - parameter dialog entry, [19](#)
 - running in a window, [37](#)
 - script, [20](#), [26](#)
 - script dialog, [17](#), [26](#), [38](#)
 - Solution display, [38](#)
 - solution procedure, [7](#)
 - steps in a solution, [17](#)
 - Table generator, [38](#)
 - two-point model, [14](#)
 - working environment, [20](#)
- PerMag analysis variables, [62](#)
- PerMag commands
 - Avg, [19](#), [29](#)
 - Boundary, [19](#), [35](#)
 - Current, [30](#)
 - DUnit, [19](#), [28](#)

- Edit script, [37](#)
- GamTarg, [29](#)
- GamTarget, [19](#)
- Geometry, [19](#), [28](#)
- Interp, [29](#)
- MaxCycle, [19](#)
- Maxcycle, [28](#)
- Mesh, [28](#)
- Mu, [30](#), [32](#)
- NDelay, [19](#)
- NUpdate, [19](#)
- Omega, [19](#), [28](#)
- Permag, [33](#)
- Potential, [30](#)
- Restarget, [19](#), [29](#)
- Run, [38](#)
- Run analysis script, [38](#)
- Setup, [38](#)
- Stop, [38](#)
- Superposition, [19](#), [35](#)
- Update, [29](#)
- permanent magnet
 - calculation example, [21](#)
 - circuit, [12](#)
 - demagnetization curve, [12](#), [33](#)
 - material parameters, [14](#)
 - models, [33](#)
 - properties, [12](#)
 - straight-line model, [33](#)
 - two-point model, [14](#), [33](#)
- plot control, [48](#)
- plot types, [48](#)
- plot view control, [50](#)
- Plot views, saving, [51](#)
- Poisson equation, [7](#), [9](#)

- recorded quantities, [52](#)
- recorded quantities, rules, [54](#)
- remanence magnetic flux, [12](#)
- reverse Polish notation, [72](#)
- rpn
 - binary operators, [72](#)
 - parsing rules, [72](#)

- scan plot menu, [55](#)
- script
 - syntax, [26](#)
- script control of analyses, [47](#)
- snap mode, [49](#)
- soft magnetic materials, [31](#)
- standard analysis functions, [46](#)
- standard solution, [63](#)
- stream function, [10](#)
 - relation to magnetic flux, [11](#)
- superposition technique, [35](#)
- surface integral, [53](#)
- surface plot control, [50](#)
- symmetry
 - cylindrical, [6](#)
 - planar, [6](#)
- symmetry boundaries, [16](#)

- Table generation tool, [40](#)
- text editor
 - internal, [37](#), [47](#)
- two-point permanent magnet model, [14](#)

- vector display, [57](#)
- vector potential, [9](#)
 - relation to magnetic flux density, [9](#)
- vector tools, [55](#)
- volume integral, [52](#)