infolytica
corporation

YOUR FASTEST SOLUTION TO A BETTER DESIGN

An introduction to

MagNet

for

Static 2D Modeling

J D Edwards
May 2014

We welcome your comments regarding Infolytica Corporation documents. You may send comments or corrections to the following address:

email: docs@infolytica.com
fax: Documentation Department
(514) 849-4239
post: Documentation Department
Infolytica Corporation
300 Leo Pariseau, Suite 2222
Montreal, Quebec H2X 4B3
Canada

Contents

Chapter 1

Introduction

Overview

The principal aim of this document is to introduce new users to the power of MagNet for solving 2D static magnetic field problems. A tutorial with detailed instructions takes the first-time user through the most important features of MagNet. This is followed by a series of case studies illustrating modeling techniques and introducing further features of the package. The document concludes with an introduction to advanced features that make MagNet a uniquely powerful tool.

What is MagNet?

MagNet is the most advanced package currently available for modeling electromagnetic devices on a personal computer. It provides a “virtual laboratory” in which the user can create models from magnetic materials and coils, view displays in the form of field plots and graphs, and get numerical values for quantities such as flux linkage and force. A MagNet user needs only an elementary knowledge of magnetic concepts to model existing devices, modify designs, and test new ideas.

MagNet is designed as a full 3D-modeling tool for solving electromagnetic problems that can involve static magnetic fields, time-varying fields and eddy-currents, and transient conditions with motion of parts of the device. Many devices can be represented very well by 2D models, so MagNet offers the option of 2D modeling, with a substantial saving in computing resources and solution time.

A feature of MagNet is its use of the latest methods of solving the field equations and calculating quantities such as force and torque. To get reliable results, the user does not need to be an expert in electromagnetic theory or numerical analysis. Nevertheless the user does need to be aware of the factors that govern the accuracy of the solution. One of the aims of this document is to show how the user can obtain accurate results. In 2D, problems can be solved very rapidly, so it is usually not necessary to consider the trade-off between speed and accuracy. In 3D modeling, on the other hand, this is an important consideration.

For the advanced user, MagNet offers facilities for user-defined adjustment of the model parameters, calculation of further results from the field solution, and control of the operation of the package with scripts and scripting forms. MagNet can be linked to other applications through the Windows ActiveX Automation interface; another application can send commands to MagNet to build and solve models and retrieve solution results.

Limitations

The information given in this document has been prepared for the free Trial Edition of MagNet, and for the full version where the licensed features are restricted to static magnetic fields and 2D models. The full version of MagNet includes parameterization: the automatic solution of sequences of problems with modified model parameters, which is available as a licensed option. Because parameterization is such a useful feature, the document includes examples of its use, but alternative methods are also provided for those who do not have access to this feature.

A guide to the document

The next sections in chapter 1 give some background information for first-time users of software for electromagnetics, particularly for users whose knowledge of elementary magnetism is insecure. It is helpful but not essential to read some of this before proceeding to the next chapter.

Chapter 2 is a practical introduction to MagNet in the form of a tutorial. It takes the user through all the steps of modeling a simple magnetic device, with full explanations of the operations and the interpretation of the results. This chapter is an essential prerequisite for chapters 3 and 4.

Chapters 3 and 4 contain case studies in which MagNet is applied to a variety of magnetic problems. These can be used in two ways: as reference material, and as a series of graded exercises for developing skills after completion of the tutorial.

Chapter 5 introduces scripting in MagNet, including the use of Microsoft Excel to control MagNet. Scripting is available in all versions of MagNet, and chapter 5 indicates some of the ways of using this powerful feature.

Appendix A contains further information about the magnetic field equations and the solution methods used in MagNet for 2D problems. Novice users do not require most of this material, but advanced users may find the additional insight helpful. The discussion of boundary conditions is relevant to all users, and includes the basis of the Kelvin transformation technique for open-boundary problems.

Appendix B covers energy, force and inductance calculation. This includes the derivation of some of the equations used in the case studies, and further information about the methods used in MagNet.

Revised edition

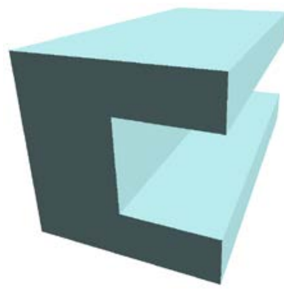
This revised edition of the document is based on MagNet 7.5, released in May 2014. A chapter on the use of the Stack Calculator, which was included in earlier editions, has been omitted because this feature has been removed from MagNet. The previous revision introduced the Kelvin transformation technique for handling some open-boundary problems; this is not yet part of the standard release of MagNet, but a manual implementation is described in chapter 3, and a scripting example is given in chapter 5.

Modeling in 2D and 3D

Some practical problems are essentially three-dimensional – examples include the rotor of a claw-pole alternator and the end-winding regions of rotating AC machines. Problems of this kind require the full 3D modeling capability of MagNet. In many cases, however, a 2D model will give useful results. There are two common types of device geometry that allow 3D objects to be modeled in two dimensions: translational geometry and rotational geometry.

Translational geometry

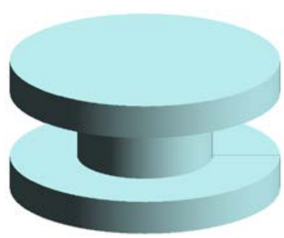
Translational geometry means that the object has a constant cross-sectional shape generated by translation – moving the shape in a fixed direction. The diagram below shows a C-core formed in this way.



With translational geometry, any slice perpendicular to the axis has the same shape. Rotating electrical machines can often be represented in this way, and so can many other devices such as transformers and actuators. Inevitably this 2D approximation neglects fringing and leakage fields in the third dimension, so the model must be used with caution. The shape is usually drawn in the XY plane, with the *z*-axis as the axis of translation.

Rotational geometry

Rotational geometry means that the object has a shape formed by rotation about an axis, like turning on a lathe. The diagram below shows an object formed in this way from the same basic C shape used in the diagram above.



Objects with rotational geometry are usually described in cylindrical polar coordinates, with the *z*-axis as the axis of rotation. The rotated shape is then defined in an RZ plane, which makes an angle θ with the 3D X axis. This geometry differs from translational geometry in two important respects. First, it is a true representation of a real 3D object, so highly accurate solutions are possible. Secondly, there are different equations to be solved, and different methods required for calculating quantities such as force and inductance. For all built-in calculations MagNet handles these differences automatically, but the user needs to be aware of the difference when interpreting flux plots for models with rotational geometry.

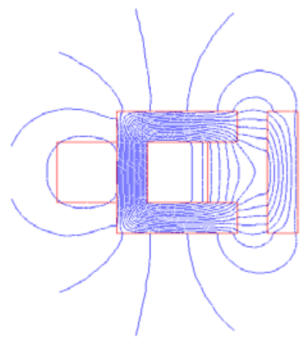
In MagNet, the 2D cross-section of a rotationally symmetric model must be drawn in the XY plane, with the Y axis as the axis of rotation. The XY coordinates then correspond to the RZ coordinates of the conventional cylindrical polar coordinate representation.

Magnetic concepts

MagNet can be used to model practical devices without knowing anything about the differential equations of electromagnetism or the numerical methods used to solve them. This section reviews some basic magnetic concepts that are required for making effective use of MagNet; more advanced topics are covered in appendix A. The system of units used is the SI or MKSA system, although other systems will be mentioned in the context of magnetic materials.

Magnetic flux density B

The fundamental magnetic concept is the magnetic field described by the vector **B**, which is termed the *magnetic flux density*. In two dimensions this field is commonly represented by curved lines, known as flux lines, which show both the direction and the magnitude of **B**. The direction of a line gives the direction of **B**, and the spacing of the lines indicates the magnitude; the closer the lines, the greater the magnitude. The diagram below shows the flux plot for a simple electromagnet where the C-shaped steel core on the left attracts the steel bar on the right. The two sides of the magnetizing coil are represented by squares.



Although the magnetic field is an abstract concept, the effects of **B** are concrete and physical. The force in a device such as this electromagnet can be expressed in terms of **B**. In simplified terms, the flux lines can be treated as elastic bands pulling the bar towards the magnet with a tensile stress (force per unit area) given by $B^2 / 2\mu_0$. In this expression, B is the magnitude of the vector **B**, and $\mu_0 = 4\pi \times 10^{-7}$ is a fundamental constant. The unit of B is the tesla (T), and the unit of μ_0 is the henry per meter (H/m).

A direct physical interpretation of **B** is given by the Lorentz equation for the magnetic force on an electric charge q moving with velocity **u**:

$$\mathbf{f} = q\mathbf{u} \times \mathbf{B} \tag{1-1}$$

If the moving charge is an electric current flowing in a conductor, then equation 1-1 leads to the familiar expression $f = Bli$ for the force on a conductor of length l carrying a current i. If the conductor itself is moving with velocity **u**, then the Lorentz force causes a displacement of charge in the conductor, leading to the expression $e = Blu$ for the induced voltage.

Frequently it is not the flux density **B** that is required, but the magnetic flux ϕ and the flux linkage λ. Flux is defined as $\phi = BA$ when the flux density **B** is constant and perpendicular to a surface of area A. If the field is not constant or perpendicular to the surface then the flux is given by an integral, but the principle is the same. Flux linkage is the sum of the fluxes for all the turns of a coil; this is $\lambda = N\phi$ for a coil of N turns where each turn links a flux ϕ. The concept of flux gains its value from Faraday's law of electromagnetic induction, which states that the voltage induced in a coil is $e = d\lambda / dt$. If the flux linkage results from current flow, either in the same coil or in a different coil, this leads to the definition of inductance as flux linkage per ampere. The calculation of inductance is discussed in appendix B.

Magnetic intensity H

Electric currents give rise to magnetic fields. The currents may flow in conductors or coils, or they may take the form of electron spin currents in the atoms of a magnetic material. In either case the problem is to define the relationship between the magnetic field described by **B**, and the currents which are the source of the field. In seeking a mathematical form for this relationship that can be used to solve practical problems, it is useful to introduce a new magnetic quantity **H,** which is related both to **B** and to the currents that are the source of **B**.

For a magnetic field in free space, set up by currents flowing in conductors, **H** is defined through the equation $\mathbf{B} = \mu_0 \mathbf{H}$. The relationship between **H** and the currents is then given by Ampère's circuital law:

$$\oint \mathbf{H}.\mathbf{dl} = \sum i \qquad (1\text{-}2)$$

where the integral on the left is taken round a closed path, and the summation on the right is the sum of all the currents enclosed by the path. This equation makes it easy to calculate the field of a simple system such as a long straight conductor or a toroidal coil, and it is the basis of the magnetic circuit concept, which is widely used for approximate calculations in electromagnetic devices. In its differential form it leads to general methods that are applicable to any problem; this point is expanded in appendix A.

The quantity **H** is known as the *magnetic intensity*; from equation 1-2 it has units of amperes per meter (A/m). For magnetic fields in free space, there would be little advantage in using **H**; equation 1-2 could be expressed in terms of **B** and μ_0. When magnetic materials are present, however, the situation is completely different.

Magnetic materials

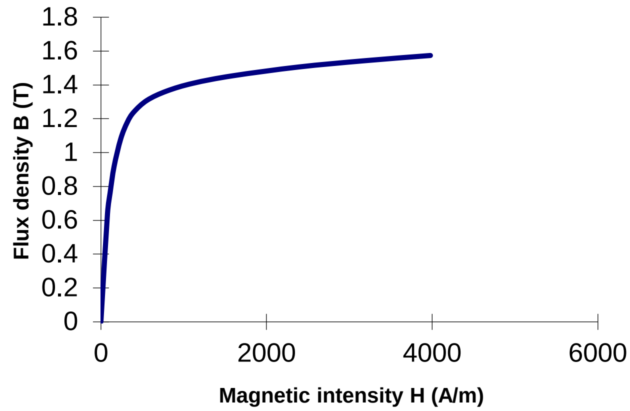
The behavior of a coil changes dramatically when it is wound on a core of magnetic material, such as iron or steel, instead of a non-magnetic material such as wood. If the core is closed, the coil has a much higher inductance. If the core is open, so that the coil behaves as a magnet, the external magnetic field is greatly increased. The material of the core has itself become the source of a magnetic field that reinforces the effects of the coil.

The behavior of magnetic materials can be described by modifying the relationship between **B** and **H**. We may put:

$$\mathbf{B} = \mu_0 (\mathbf{H} + \mathbf{M}) \qquad (1\text{-}3)$$

where **H** is the magnetic intensity given by equation 1-2, and **M** is an induced magnetization in the material which depends on **H**. Thus **H** can be regarded as the cause, which is related to currents in conductors; **B** is the effect, giving rise to forces and induced voltages.

From the point of view of the device designer, the magnetization **M** is unimportant; what matters is the relationship between **H** and the resulting **B**. This relationship can be extremely complex; the vectors may not be in the same direction, and the present value of **B** may depend on the past history as well as the present value of **H**. For many practical purposes, however, these complexities can be ignored and the properties of the material expressed by a simple curve relating B to H. This is the B-H curve or magnetization characteristic of the material. A typical example is the curve for transformer steel shown below. This curve has three distinct regions: the steep initial part of the curve, where a small increase in H produces a large increase in B; the knee of the curve; and the saturated region beyond the knee, where a large increase in H is required for any perceptible increase in B. With this material, a flux density of about 1.4 T marks the onset of saturation. MagNet provides B-H curves for a wide range of magnetic materials, and also enables the user to create new material curves.



Transformer steel magnetization characteristic

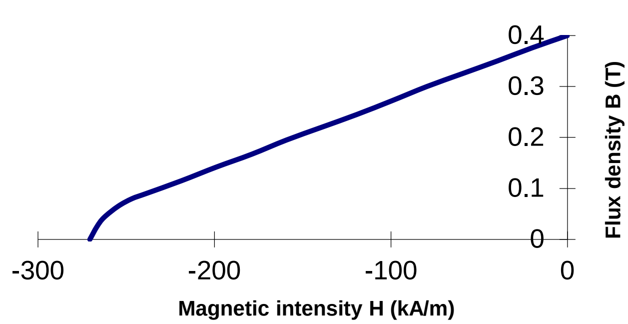
With soft magnetic materials such as transformer steel the magnetization virtually disappears when the external field is removed, so the B-H curve passes through the origin. For these materials it is convenient to express the relationship between B and H as

$$B = \mu_0 \mu_r H \tag{1-4}$$

where μ_r is a dimensionless property of the material known as the *relative permeability*. A non-magnetic material has a relative permeability of 1. For a material such as transformer steel, μ_r is not constant but varies with B or H; it has an initial value of several thousand, but may fall below 100 in the saturation region. To simplify the study of some devices, and to speed up the solution, it is useful to have a fictitious material with a constant permeability corresponding to a linear relationship between B and H. MagNet provides several linear materials, with relative permeability values ranging from 10 to 10^6.

Permanent magnets

Permanent magnets have the property that some magnetization remains in the material when the external field is removed. For these materials the important part of the B-H curve is the second quadrant, known as the demagnetization characteristic, shown below for a ceramic ferrite.



Ceramic ferrite demagnetization characteristic

Permanent-magnet materials have two distinctive parameters: the remanence B_r and the coercivity H_c, which are defined as follows. B_r is the value of B remaining in the material when the applied H is zero. H_c is the negative value of H that must be applied to reduce B to zero. For the ceramic ferrite characteristic shown above, the value of B_r is 0.4 T, and the value of H_c is –270 kA/m. An additional parameter is the recoil permeability, which specifies the behavior when the magnetic conditions change. With some materials, if the magnitude of the negative (or demagnetizing) H is increased and then reduced, the operating point does not retrace the original curve, but instead it follows a *recoil line* with a smaller slope. The recoil permeability is the slope of this line divided by μ_0; it is thus a relative permeability, and its value is often close to 1.

Some permanent-magnet materials such as samarium cobalt have a B-H characteristic which is virtually a straight line from $(0, B_r)$ to $(-H_c, 0)$, and the recoil line has the same slope. These materials are treated as linear, and are specified by the values of B_r and H_c instead of a B-H curve.

Units

Occasionally the B-H data for magnetic materials are given in units other than the SI or MKSA units of tesla and ampere/meter respectively. The following unit conversions may be required.

B: 1 tesla = 10 kilogauss = 64.516 kilolines/inch^2.

H: 1 ampere/meter = 12.566×10^{-3} oersted = 25.4×10^{-3} ampere/inch.

Static and time-varying fields

A static field does not change with time. For a given model structure, the field is determined by the constant source currents or permanent magnets. If the source currents vary with time, then eddy currents may be induced in any conducting materials in the model, and these currents will modify the magnetic field. When the source currents are sinusoidal alternating quantities, and the magnetic materials are linear, the eddy currents and the resulting magnetic field will also be sinusoidal. This is termed a time-harmonic field. For general transient problems involving time-varying fields, and the particular case of time-harmonic fields, the equations to be solved are different from those of the static field given in appendix A. The full version of MagNet provides transient and time-harmonic solvers in addition to static solvers for fields in two and three dimensions. In many applications involving alternating fields, however, eddy-current effects are small. In these cases a static field solution gives good results, as will be shown in some of the case studies.

Using MagNet effectively

This section contains a few practical pointers to getting the best out of MagNet. Some of the suggestions may not make much sense until the user has had some experience of using MagNet, at least to the extent of working through the tutorial in chapter 2.

The principle of progressive refinement

The time taken for MagNet to solve a problem will depend on the complexity of the model and the desired solution accuracy. For this reason alone it is not advisable to attempt an exceedingly detailed model of a practical device with every geometric feature faithfully copied. There is also a practical reason for avoiding complex models initially. The first model is almost certain to contain mistakes; if it is very detailed it will take a long time to solve, and an even longer time to rebuild when the solution has revealed the mistakes.

It is generally best to begin with a very simple model that preserves the essential features of the device. Shapes and dimensions can be simplified. Some parts do not need to be modeled at all. For example, real coils will have non-magnetic insulation separating the coil from the steel core of the device. There is no need to model such insulation; the coils can be drawn touching the steel without any significant error. The case studies in chapters 3 and 4 give some indication of what can be done with simple models.

For the first solution of a new model, it is desirable to get a flux plot as quickly as possible, because the flux plot is an effective tool for revealing errors in the structure of the model. At this stage, there is no need to use the powerful adaption feature of MagNet to improve the solution accuracy, or to take other measures to control the size of the finite-element mesh.

When the new model is producing a sensible flux plot, and the numerical results for forces, torques and inductances are plausible, the solver and adaption options can be used to improve the accuracy. In some cases, it may be desirable to control the size of the mesh explicitly by setting parameters in MagNet. The case studies give examples of the methods and settings that may be required.

Getting accurate results

MagNet uses the finite-element method to solve the field equations. For a 2D model, the entire region is subdivided into a mesh of triangular elements, and within each element the true field is approximated by a polynomial. The accuracy can be improved by increasing the order of the polynomial: this is one of the solver options. It can be further improved by using smaller elements in critical regions of the model, which is done automatically when the user sets the adaption options.

With any numerical method, perfect accuracy is unattainable. Even with full use of the options for improving the accuracy, the solution generated by MagNet will contain errors. In most cases these errors will be insignificant, and are likely to be smaller than the changes caused by manufacturing tolerances or variations in the magnetic properties of the materials.

Calculated values for forces and torques are particularly sensitive to errors in the field solution, so these values are likely to change significantly as the solution accuracy is improved. If these are the quantities of interest in the device, then it is sensible to continue refining until the values appear to have converged. If it is known that some torque values or force components should be zero, then refinement should continue until the values are small in comparison with the useful values. Similarly, where quantities are expected to be equal in magnitude, the difference should be a small fraction of the mean magnitude.

With certain types of problem, the automatic method of refining the mesh may not yield an accurate solution in MagNet. A typical example is the calculation of force or torque in a device where the active airgap is very small in comparison with the dimensions of the iron parts. Here the values may not converge towards a limit as the refinement level is increased. Cases like this require the user to take control of the mesh structure, which can be done by specifying a maximum element size in some parts of the model.

Getting help

MagNet is a powerful and complex package with many features that are not covered in this introductory document. Although care has been taken to make the instructions in the document clear and accurate, there may be occasions when the user is in difficulty. The first point of assistance is the comprehensive help facility in MagNet, which gives detailed explanations of the features and instructions for their use.

Further help is available from the Infolytica web site: http://www.infolytica.com, where there is extensive tutorial material and a gallery of examples of the application of MagNet to a variety of electromagnetic modeling problems.

Chapter 2

Tutorial: C-core Electromagnet

Introduction

This chapter takes the user through the complete sequence of using MagNet to model a simple electromagnetic device: the C-core electromagnet shown below. The objectives are as follows:

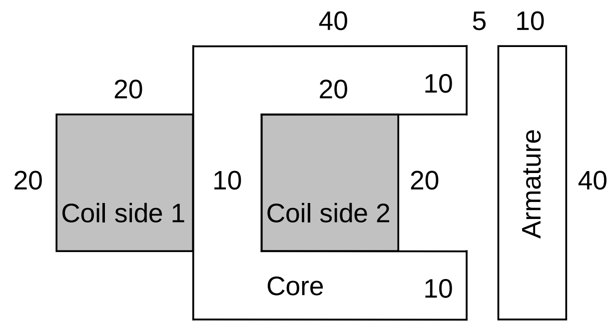
- To examine the magnetic field in various parts of the magnetic circuit.
- To determine the force on the square armature plate.
- To determine the self-inductance of the coil.
- To modify the model by changing the coil current, the core material, the shape of the core, and the position of the armature.

This is an example of a device that can be represented quite well by a 2D model with translational geometry, even though it is not very long in the direction of translation. The important leakage field between the poles is accurately modeled, and the part of the fringing field that is neglected in a 2D model has only a minor effect on the calculation of force. Fringing and leakage are discussed later, on pages 25 and 63. An important limitation of the model, however, is that the calculated inductance is very inaccurate because the leakage field in the third dimension has been neglected. If an accurate value for the inductance is required, a 3D model must be used. Results from 2D and 3D models are compared on page 40.

Device Model

Brief description

The diagram below shows the cross-section of the electromagnet with the dimensions in millimeters. Each side of the coil is a square of side 20 mm, and the core is 10 mm thick throughout. The armature and the core each have a depth of 40 mm, perpendicular to the plane of the drawing. The coil has 1000 turns, and the current is initially 2.0 A.



Since the electromagnet is surrounded by air, the magnetic field theoretically extends to infinity, making it an *open boundary* problem. In practice the field decays rapidly with distance, and is insignificant at a distance of about 10 times the magnet dimensions.

A technique is available in MagNet for exact modeling of an open boundary, and for some types of problem this is the best approach: an example is given in chapter 3. However, for many devices such as this electromagnet, good results can be obtained with the simpler technique of specifying an outer boundary at a distance of 5 to 10 times the magnet dimensions.

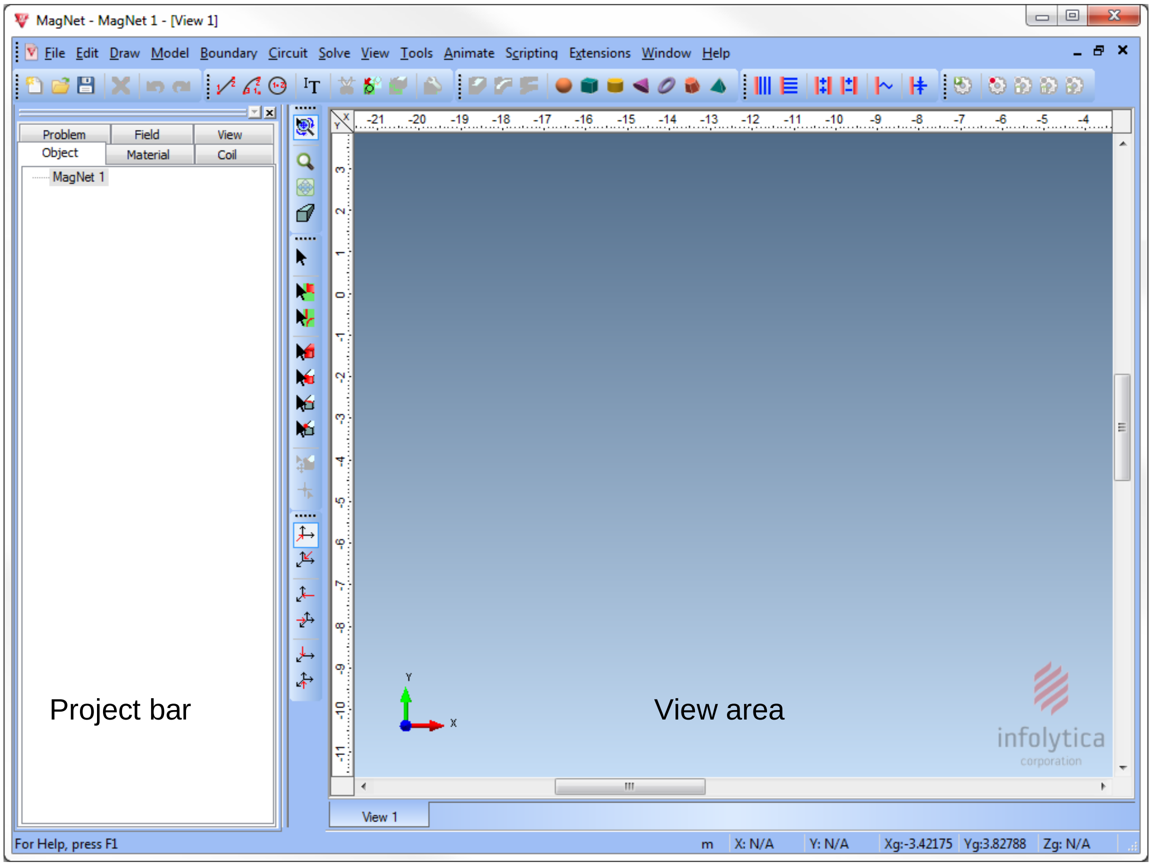
Modeling the electromagnet involves the following steps:

- Draw the cross-section of the core.
- Extend it in a straight line to form a solid body, and specify the material of the core.
- In the same way, construct the armature and the two coil sides.
- Specify the coil.
- Define the bounded region of the problem as an *air box* within which the field will be calculated.
- Instruct MagNet to solve the equations and display the results.

These steps are described in detail in the next sections.

Getting started

The instructions given below assume that the user is familiar with Microsoft Windows, that MagNet has been installed, and the application started by double-clicking the MagNet icon. The MagNet Main window, shown below, should be visible. This is the default window, which can be customized by experienced users.



1 Examine the MagNet Main window, and identify the parts listed below.

- The Project bar displays information about the model, with tabs at the top labeled Object, Material, etc. Initially, the Object page is displayed.
- The View area is the work area where the model is constructed and the results viewed. Initially, the View 1 window is displayed.
- Between the Project bar and the View area there are vertical toolbars with buttons for viewing and selecting objects.
- At the top of the Main window, there is the usual menu bar, and a row of horizontal toolbars.

2 Move the pointer over the buttons on the toolbars, pausing on each for the “tooltip” message that describes the action of the button.

- Toolbar buttons are duplicated on the menus. For example, the View menu gives access to the same viewing tools as the buttons on the vertical View toolbar.
- Other toolbars and buttons can be added, by selecting Customize Toolbars from the Tools menu.

Building the model

In MagNet, the default units of length are meters. For this model, however, it is more convenient to work with dimensions in millimeters. Proceed as follows to start a new model and change the default units and grid settings.

Initial settings

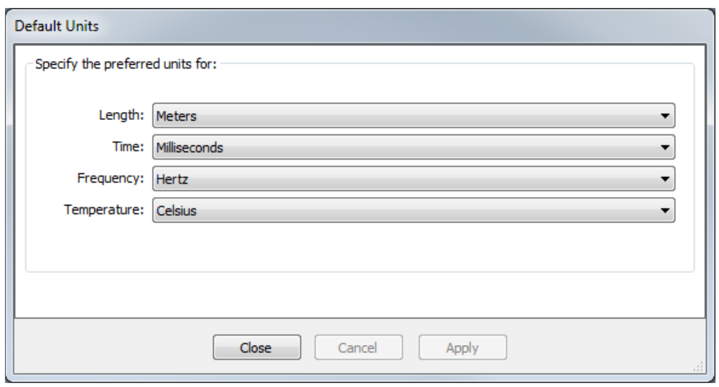
1 On the File menu, click Save. Alternatively, click the Save button.

- Browse to a suitable folder for storing the model.

2 Save the model with the name **C-core electromagnet.**

The model name in the Object page should change to C-core electromagnet. The extension will be .mn in the full version of MagNet, or .mnte in the Trial Edition.

3 On the Tools menu, click Set Units to display a dialog:

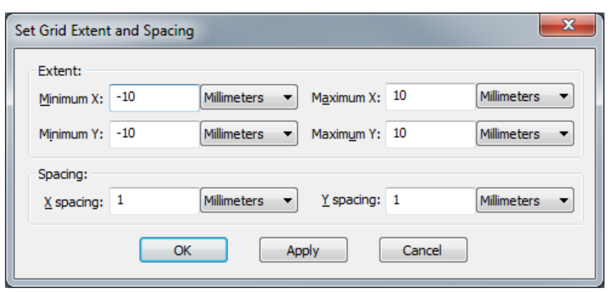


4 Click the Length drop-down list.

- Select Millimeters.

5 Click OK to close the dialog.

6 On the View menu, click Set Construction Grid to display a dialog:



7 Set the Extent and Spacing values as follows:

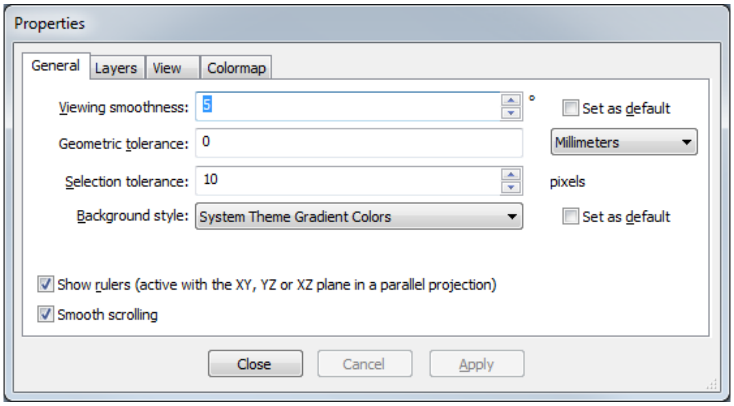
- Minimum X: **–60** Maximum X: **20**
- Minimum Y: **–20** Maximum Y: **20**
- X spacing: **5** Y spacing: **5**

8 Click OK.

Changing the background color

It may be easier to draw the model if the background color is changed:

1 Right click in the View 1 window, and select Properties to display a dialog:



2 In the Background Style drop-down list, select System Window Solid Color.

3 If this background is preferred, Click Set as Default, and click OK.

Displaying the grid

Display the whole of the construction grid as follows.

1 On the View menu, click Construction Grid.

You should see a grid of a few small points, widely spaced.

2 On the View menu, click Examine Model Dynamically, or click the Examine Model button. 

3 In the View 1 window, roll the mouse scroll wheel downwards to zoom out, until the whole of the grid is visible.

- If you go too far, roll the scroll wheel upwards to zoom in.
- To restore the original display, double-click in the View 1 window. Alternatively, on the View menu, click View All.

4 If there is no scroll wheel, proceed as follows to add a Dynamic Zoom button to the View toolbar:

- On the Tools menu, click Customize Toolbars.
- Select the Commands page of the Customize dialog.
- In the Categories list, click View Toolbar.
- Drag the Dynamic Zoom button to the View toolbar. 
- Click OK to close the Customize dialog.

5 Click the Dynamic Zoom button.

6 In the View 1 window, drag the pointer downwards to zoom out, so that more of the grid is visible, and then release the button.

- Repeat as required until the whole grid is visible. If you go too far, drag upwards to zoom in.
- To restore the original display, double-click in the View 1 window. Alternatively, on the View menu, click View All.

Drawing the magnet core

The default drawing mode is Snap to Grid, which means that lines, arcs and circles drawn with the mouse will lock to the grid points. Drawing tools can be activated from buttons on a horizontal toolbar, as shown in the instructions, or selected from the Draw menu. The device cross-section, with dimensions in millimeters, is reproduced below.

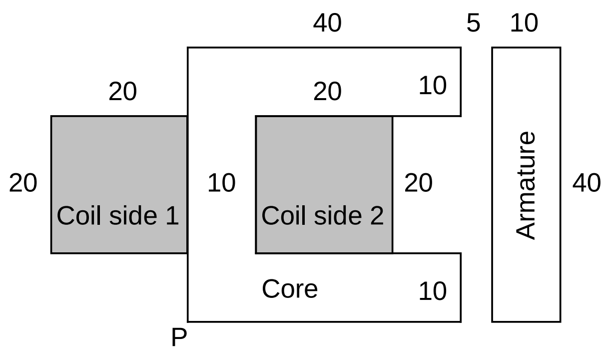
In MagNet, all drawing takes place on a 2D *construction slice*. For 2D models this is just the XY plane, but for 3D models the construction slice can be moved to other planes.

1 Click the Add Line button.

2 Draw the outline of the C-core as follows, starting at the bottom left-hand corner (point P in the diagram).

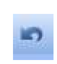
- Move the pointer. Observe the coordinates displayed on the Status bar at the bottom of the screen.
- Click near the grid point at (–40, –20).
- Move the pointer near (0, –20) and click.
- Move the pointer near (0, –10) and click.
- Continue in this way for the rest of the core.

Do not draw the lines for the coil sides or the armature at this stage. See below for the method of deleting any lines drawn in the wrong place.



3 If any lines have been drawn in the wrong place, correct them as follows.

- If the last line to be drawn was in the wrong place, undo this action with Ctrl+Z, or click the Undo button. For other lines, proceed as follows.



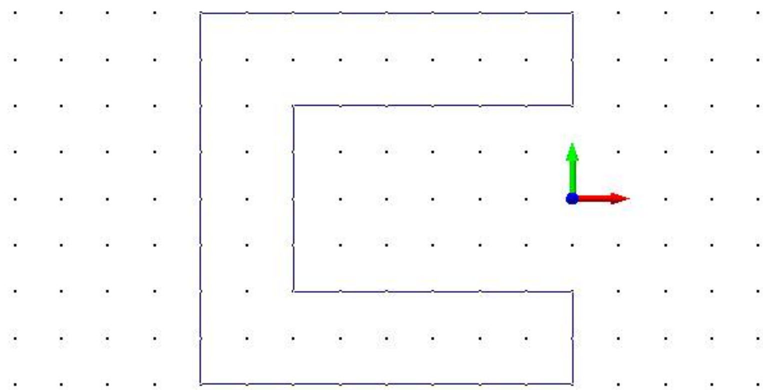
- Double-click the last point, or press Esc, to stop line drawing.
- Click the Select Construction Slice Lines/Arcs button.

- Click the line that you want to delete.

  The selected line should turn red.
- Press the Delete key.
- Click the Add Line button and redraw the line.

4 When the drawing is complete, double-click the last point, or press Esc, to terminate line drawing.

5 The finished outline of the core should look like the picture on the next page.



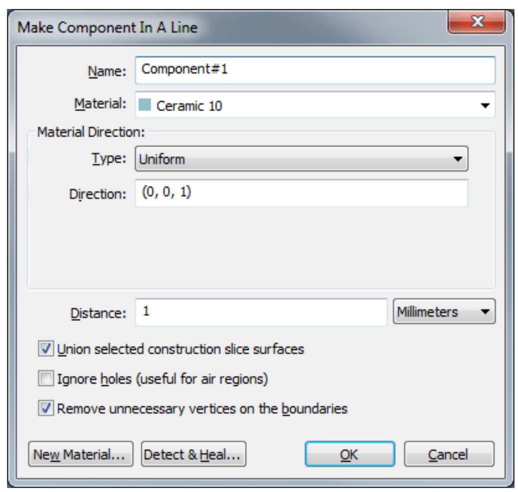
Completing the magnet core

1 Click the Select Construction Slice Surfaces button. 

2 Click anywhere inside the core.

The interior of the core should fill with a red pattern.

3 Click the Make Component in a Line button to display a dialog: 



4 Change the Name from Component#1 to **Core**.

5 Click in the Material box. Start typing **CR10: Cold rolled 1010 steel**

- When you type **CR** the display will change to the name of the required material.
- Alternatively, scroll down through the list and select the material.

6 Change the Distance to **40**.

7 Click OK.

A component named Core should be shown in the Object page of the Project bar.

Making the armature

1 Click the Add Line button, and draw the outline of the armature.

- Make sure that there is a 5 mm gap between the armature and the core.

2 Click the Select Construction Slice Surfaces button, and click inside the armature.

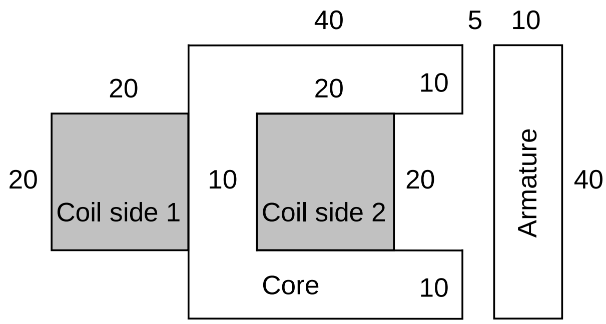
3 Click the Make Component in a Line button.

The Distance should be shown as 40 mm, and the Material as CR10: Cold Rolled 1010 steel.

- Change the Name to **Armature.**
- Click OK.

4 The Object page of the Project bar should show two components: Core and Armature.

If the name of a component is wrong, edit the name in the Object page by selecting the name and pressing F2, or by making two single clicks.



Making the two coil sides

1. Draw the outline of coil side 1, by adding three lines to the left of the core.
2. Select the coil side, and make the component with the following entries in the dialog box:
   - Change the Name to **CoilSide#1**.
   - Click in the Material box. Start typing **Copper: 5.77e7 Siemens/meter**
   - The Distance should be 40 mm.
3. In a similar way, make the second coil side, with the name CoilSide#2.
4. The Object page should show two new components for these coil sides.

Defining the coil

The next step is to link the two sides to form a coil, and to specify the number of turns and the current.

1. In the Object page, click CoilSide#1.

   The component name should be highlighted, and the left-hand coil side marked in the View 1 window.
2. Hold down the Ctrl key, and click CoilSide#2.

   Both component names should be highlighted, and the coil sides marked in the View 1 window.
3. On the Model menu, click Make Simple Coil.

   Coil#1 should appear in the Object page.
4. Select the Coil page of the Project bar, by clicking the Coil tab.

   Coil details should be displayed.
5. Click on 1 Turn (the sixth item in the list for Coil#1).
   - Click a second time, or press F2.
     The display should change to a text box displaying the number 1.
   - Change 1 to **1000** and press Enter.
6. Click on 0 A rms.
   - Click a second time, or press F2.
   - Change 0 to **2** and press Enter.

     The display should show 2 A rms
7. Select the Object page of the Project Bar, by clicking the Object tab.

   This displays the names of the model components again.

Removing selections

After defining the coil in this way, part of the model will be selected in the View window. This can interfere with subsequent displays. The following is a simple way of removing all selections:

- In the Object page of the Project bar, click the model name.

Air box

An outer boundary must be added to the model by creating a new component called an *air box,* which encloses all the other components. The default boundary condition for the air box is Flux Tangential (see appendix A), which means that the outer boundary is a flux line. This is a reasonable approximation if the boundary is sufficiently far away from the device.

Since the air box is much larger than the electromagnet, it is not convenient to draw it with the mouse. Instead, coordinates are entered with the keyboard.

If the air box is made in the same way as the other components, it will contain holes corresponding to the shapes of those components. This is undesirable, because it will cause problems later when the model is modified. To prevent holes being formed, the construction slice lines for the other components will be removed, as described in step 3 below.

1 On the View menu, click Construction Grid.

This turns off the construction grid display.

2 On the View menu, click Update Automatically. Alternatively, click the Automatic View All button.

This keeps all of the components in view in the window.

3 Remove the construction slice lines as follows.

- Click the Select Construction Slice Lines/Arcs button.

- Press Ctrl+A to select all the lines.

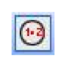
All the lines are marked in red.

- Press Delete to delete the lines.

4 On the Tools menu, click Keyboard Input Bar if there is no check mark beside it.

The Keyboard Input bar should be displayed at the bottom of the Main window, above the Status bar, with a text box for entering coordinates:

5 Click the Add Circle button on a horizontal toolbar.



The Status bar at the bottom of the window should show:
Specify the center point and a point on the radius of the circle…

6 Click in the text box of the Keyboard Input bar.

The text box should show the center coordinates as (0, 0).

- If the coordinates are not shown as (0, 0), edit the text.

7 Press Enter, or click the Enter button.

Nothing will change in the View 1 window, so it looks as though nothing has happened. However, the display next to the Enter button should have changed from (x, y) *to* (0, 0) mm, *and the coordinate display on the Status bar should show* X:0 Y:0 Xg:0 Yg:0 Zg:0.

8 Change the coordinates in the text box to **(400, 0)**, and then press Enter, or click the Enter button. The brackets and the comma can be omitted.

The display next to the Enter button should show (400, 0) mm, *and the coordinate display on the Status bar should show* X:400 Y:0 Xg:400 Yg:0 Zg:0. *A circle of radius 400 mm should be shown in the View 1 window.*

9 Click the Select Construction Slice Surfaces button. 

10 Click inside the circle.

11 Click the Make Component in a Line button to make the air box. 

- Change the Name to **AirBox**.
- Click in the Material box. Start typing **AIR**
- Alternatively, scroll down through the list and select the material.

It is essential to use AIR, which has a special function in MagNet. Do not select Virtual Air, which is a generic non-magnetic material.

- The Distance should be 40 mm.
- Click OK.

12 On the File menu, select Save, or click the Save button. 

It is good practice to save often, in case of computer problems.

Viewing the model

To view the electromagnet in more detail again, zoom in as follows.

1 On the View menu, click Examine Model Dynamically, or click the Examine Model button. 

- Position the pointer outside the electromagnet.
- Hold down the Ctrl key and the left mouse button, and drag a rectangle enclosing the electromagnet.
- Release the button.

The region contained in the rectangle will expand to fill the View 1 window.

2 If the Ctrl key was not pressed, this operation will have rotated the model instead of dragging a rectangle. If this has happened, restore the normal view as follows:

- On the View menu, click Preset Views / Positive Z axis, or click the Show XY (+Z) button. 

3 To adjust the size of the displayed view, zoom in or out:

- Roll the mouse scroll wheel upwards or downwards.
- If there is no scroll wheel, click the Dynamic Zoom button; drag the pointer upwards to zoom in or downwards to zoom out. 
- See the instructions on page 15 for adding the Dynamic Zoom button.

Solving the model

MagNet uses the finite-element method of solving the electromagnetic field equations. This subdivides a 2D model into small triangular elements, forming a mesh that covers the entire region. The true field within each element is approximated by a polynomial in terms of the field values at a small number of points, and MagNet solves for the unknown field values at these points for all the elements. For example, a first-order polynomial just gives linear interpolation between the field values at the vertices of the triangles.

The accuracy will be higher with a fine mesh or a high-order polynomial. By default, a first-order polynomial is used, which is fast but not very accurate.

Initial solution

1. Right-click in the View 1 window and select Initial 2D mesh. Alternatively, on the View menu, click Initial 2D Mesh.

   This should show the default mesh that MagNet uses to solve the field equations.

2. On the Solve menu, click Static 2D.

   The Solver Progress dialog should appear briefly. When the solution is complete, the Results window should be displayed instead of the View 1 window.

3. Return to the View 1 window by clicking the View 1 tab at the bottom of the View area.

Viewing the flux plot

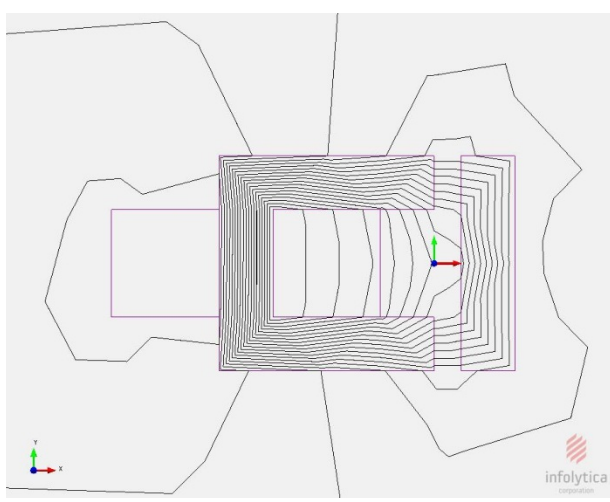
1. Select the Field page of the Project bar by clicking the Field tab.

   The Field page has tabs at the bottom for Contour, Shaded and Arrow. The Contour page is active by default, with the Flux Function selected.

2. Click the Shaded tab, and in the list of fields click None.

3. Click Update View to view a contour plot of the flux function.

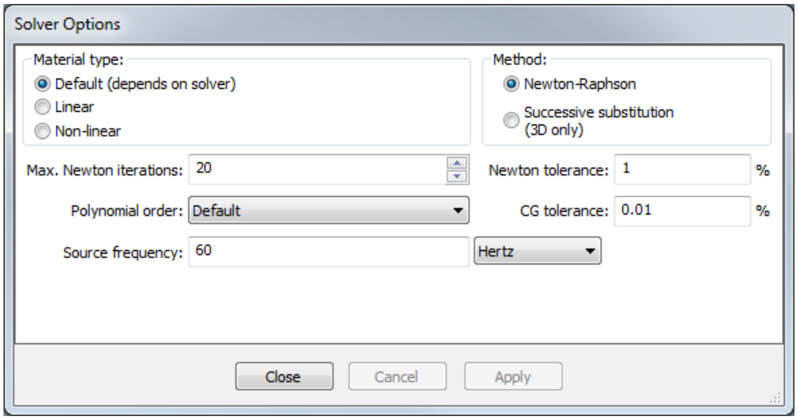
   This is the ordinary flux plot. It should be similar to the plot below. The plot is irregular because the solution is not very accurate at this stage.



Improving the solution accuracy

To improve the solution accuracy, the polynomial order of the elements can be increased as follows.

1 On the Solve menu, click Set Solver Options to display a dialog:



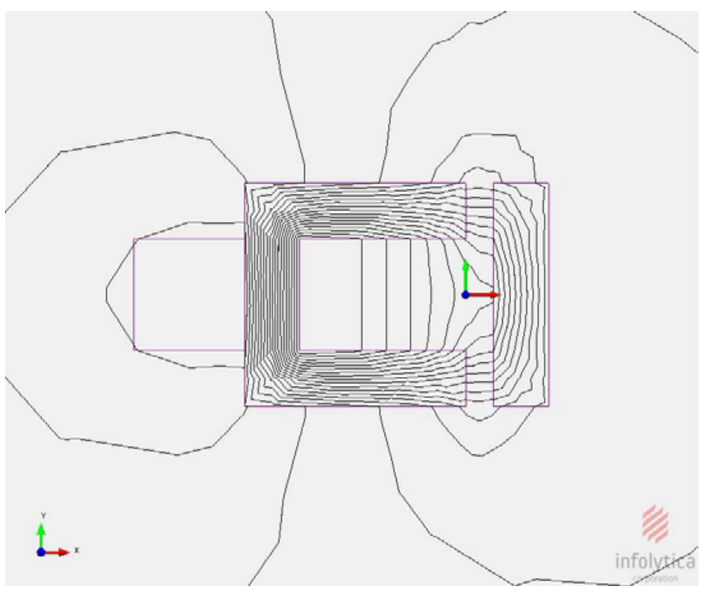
The Polynomial Order is shown as Default. For this model, the default is order 1.

- In Polynomial Order, select 2 from the drop-down list.
- Click OK.

2 On the Solve menu, click Static 2D.

When the solver finishes, the flux lines should be smoother, as shown below, indicating a more accurate solution.

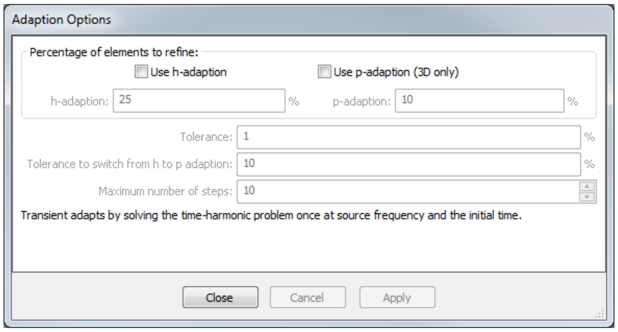
3 Save the model again.



Refining the mesh

Increasing the polynomial order has made some improvement, but the real problem is that the mesh is too coarse in parts of the model. MagNet can refine the mesh automatically – a process termed *adaption*.

1 On the Solve menu, click Set Adaption Options to display a dialog:



- Click Use h-adaption.

  *The h-adaption box should have the value **25**.*

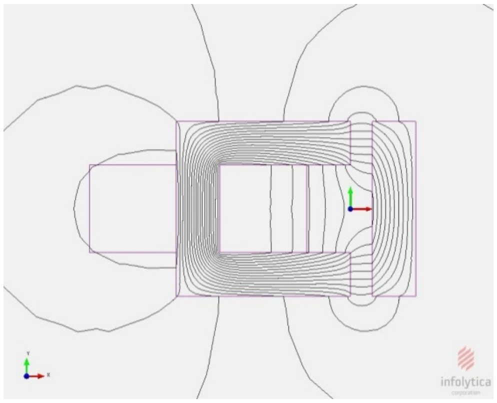
- In the Tolerance box, type **0.5**

At each adaption step, MagNet will select the worst 25% of elements and generate new elements with half their dimensions.

2 Click OK.

3 On the Solve menu, click Static 2D.

The Solver Progress dialog shows the adaption steps. The process continues until the change in the calculated value of stored energy is less than the specified tolerance of 0.5%. The resulting flux plot should resemble the diagram below.



4 Examine the change in the mesh as follows.

- Right-click in the View 1 window and select Initial 2D mesh. Alternatively, in the View menu, click Initial 2D Mesh.

  This should show the original mesh.

- Similarly, select Solution Mesh.

  This should show the refined mesh.

5 Save the model again.

Post-processing

After a field solution has been obtained, other quantities can be calculated and displayed. This is termed *post-processing*. MagNet has a Results window that displays numerical quantities such as force and flux linkage. In addition, shaded color plots and contour plots of fields can be displayed in the Field page of the View window; field values at any point in the model can then be displayed by moving the mouse pointer.

Each of the available fields can be viewed as a smoothed field. Smoothing applies averaging to remove the discontinuities that are present in the computed field. A smoothed field is continuous across mesh element boundaries where the materials are the same. Smoothed fields are not continuous across element boundaries where the materials differ (true discontinuity exists at these boundaries).

Getting flux density values

1. In the Field page of the Project bar:
   - Click the Shaded tab.
   - Click |B| smoothed.
   - Click Update View.

   This should show a color map of the flux density magnitude, superimposed on the flux plot.
2. On a vertical toolbar, click the Probe Field Values button.

   
   - Move the mouse pointer anywhere in the model region, **but do not click**.
   - The Status bar should show two values at the left-hand side: the flux function, and |B| smoothed.

     Ignore the flux function. The other number is the flux density magnitude at the position of the pointer.
   - Move the pointer without clicking, and observe the change on the Status bar.
3. Hold the mouse still, with the pointer anywhere in the model region, and click once.
   - A window should open at the bottom of the Main window, called the Text Output bar.
   - This displays the coordinates of the point, and values of the flux function and |B| smoothed.
   - Every click in the model region displays a new set of values.
4. Close the Text Output bar by clicking the Close box on the left-hand side of the bar, or clicking Text Output Bar on the Tools menu.

Notice two features of the magnetic field in the electromagnet. First, the magnetic field in the airgap between the poles and the armature is not confined to the pole region, but spreads into the surrounding air; this is termed *fringing*. Secondly, some flux takes a short cut across the space between the poles, instead of crossing the airgap and passing through the armature; this is termed *leakage*.

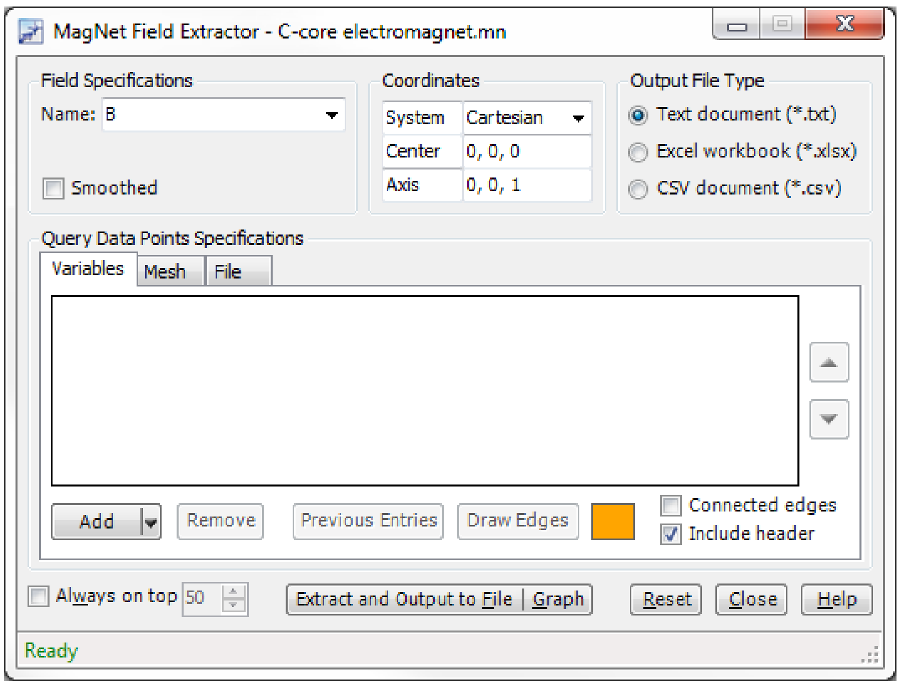
Graphs of flux density

MagNet provides items on the Extensions menu for further processing of the solution results. These include the versatile Field Extractor, which can extract field values over a grid of points, and display graphs of values along lines parallel to the X or Y axis.

Field Extractor

It is instructive to display graphs of the flux density components B_x and B_y along a line passing through the airgap between the armature and the poles, with end-point coordinates (2.5, –30) and (2.5, 30). This will show the variation of the flux density in each airgap, and the rapid decay beyond the edges of the poles.

1 On the Extensions menu, click Field Extractor to display a new window:



In the Field Name drop-down list, B should already be displayed.

2 Click Add twice to add X and Y rows to the Variables box.

3 In each row, enter the Start, End, and Number of sampling points as follows:

X - End	2.5	1
Y - End	-30,30	61

For X, entering only a Start value automatically sets the number of sampling points to 1.

Since the Y coordinate range is 60 mm, setting the number of sampling points to 61 will sample the field at increments of 1 mm.

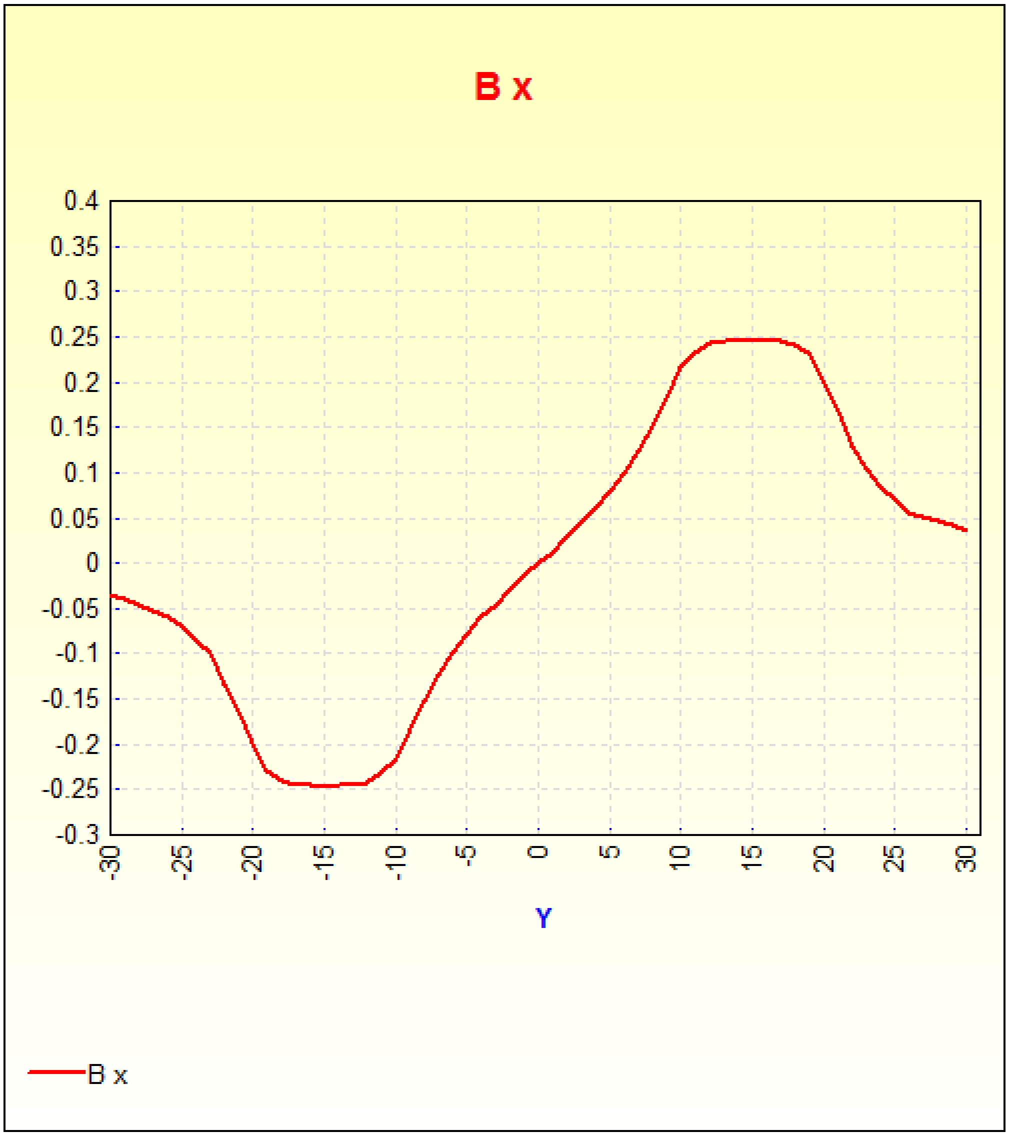
4 Click Graph.

Three new graph windows should be displayed in turn behind the Field Extractor window, replacing the View 1 window.

5 Close the Field Extractor window.

6 Click the B x or B y tab at the bottom of the View area to display the corresponding graph.

The graph of B x should resemble the following:



Global quantities

Click the Results tab to display the Results window. This has pages for displaying the calculated values of quantities including energy, force and flux linkage for the model. Numerical results given below were obtained with version 7.5 of MagNet.

The display precision can be adjusted with a control at the top of the Results window; the default precision is 3. For the results given below, the precision was set to 4.

Energy

By default, the Results window should display the Energy page. If this is not visible, click the Energy tab. The displayed values should be similar to the following:

Stored Magnetic Energy	0.3202 J
Co-energy	0.3216 J

See appendix B for a discussion of energy and co-energy. The difference between these two values is an indication of the level of saturation in the steel parts of the model.

Flux linkage and inductance

In the Results window, click the Flux Linkage tab. The displayed value of the flux linkage with the coil should be similar to the following:

Coil#1	0.3209 Wb

The self-inductance of the coil may be calculated from the flux linkage:

$$L = \frac{\lambda}{i} = \frac{0.3209}{2.0} = 0.1605 \text{ H} = 160.5 \text{ mH}$$

Alternatively, the self-inductance may be calculated from the stored magnetic energy:

$$L = \frac{2W}{i^2} = \frac{2 \times 0.3202}{(2.0)^2} = 0.1601 \text{ H} = 160.1 \text{ mH}$$

See appendix B for details of these methods of calculating inductance.

With some models, the self-inductance can be calculated automatically with one of the standard extensions to MagNet: the RLC Matrix Calculator (see page 67). However, this method is valid only for models constructed from linear magnetic materials, so that the inductance values are independent of current. It is not applicable to the C-core electromagnet, where the inductance falls with increasing current.

Forces

MagNet automatically calculates forces and torques on all bodies in the model. A body is defined as a set of connected components surrounded by the special material AIR. In this case, there are two bodies: the armature, and the set comprising the core and the two coil sides.

In the Results window, click the Force tab.

The force components and magnitudes, in newtons, should be similar to the following:

Body	X	Y	Z	Magnitude
Core + CoilSide#1 + CoilSide#2	+24.21	+0.0015	0	24.21
Armature	–24.15	-0.0018	0	24.15

- Since this is a 2D model, the Z components are zero.
- By symmetry, the Y components should also be zero. The non-zero values are an indication of numerical error in the solution, but they are very small in comparison with the X values.
- The X components should be equal and opposite. The small difference in magnitude is another indication of numerical error.
- The signs indicate a force to the left on the armature, and a force to the right on the core and coil sides. This signifies a force of attraction, as expected from the flux plot.

Torque values are also displayed, but they can be disregarded. From the symmetry of the model, the torque values should be zero; non-zero torque values are further indications of numerical error in the solution.

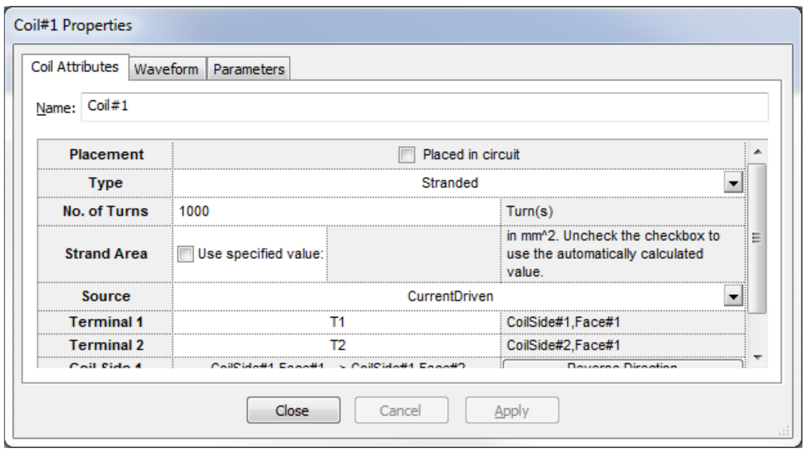
Modifying the model

Once a model has been constructed, it is a straightforward matter to make changes. Each object in the model, listed in the Object page of the Project bar, has a set of properties that can be modified. In addition, every property of the model can be *parameterized*: it can be given a list of values, and MagNet will create a corresponding set of problems. For the electromagnet, it is useful to vary the coil current and the armature position in this way. If the full version of MagNet is licensed for parameterization, the set of problems will be solved automatically. This feature is disabled in the Trial Edition of MagNet.

Varying the coil current

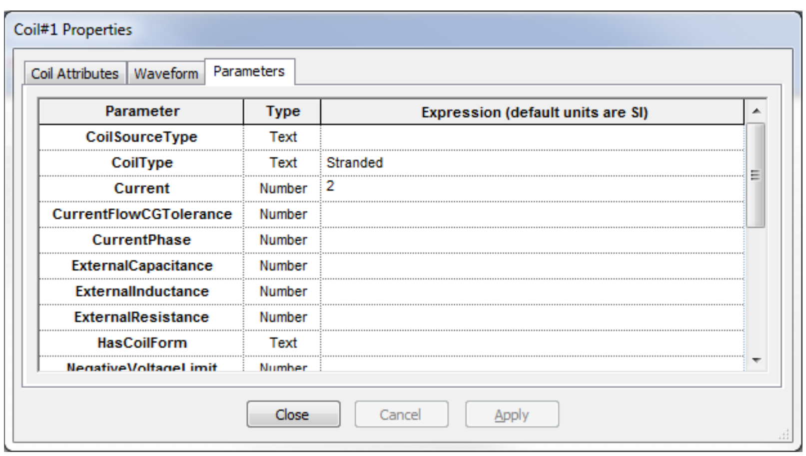
Parameterization

1 In the Object page of the Project bar, right click the Coil#1 object, and select Properties to display a dialog:



2 Click the Parameters tab.

Observe that there is a value 2 in the Expression field for the Current parameter:



3 Edit the Expression field for the Current parameter so that it contains the following list of values:
2, 4, 6, 8, 10

4 Press Enter to accept the list.

If the field turns red, there is an error in the field, which must be corrected.

5 Click OK to close the Properties dialog.

6 In the Object page of the Project Bar, click the name of the model (C-core electromagnet).

This cancels the selection of the Coil#1 object. If it is left selected, it interferes with the field plots described below.

7 Select the Problem page of the Project Bar.

- Observe that five problems have been created, each with a different coil current.
- If MagNet is licensed for parameterization, all of the problems will be marked for solution, otherwise only the first problem will be marked.

8 On the Solve menu, click Static 2D.

- If MagNet is licensed for parameterization, all five problems will be solved in succession automatically.
- With the Trial Edition of MagNet, only the first problem will be solved automatically. In this case, the others can be solved manually as follows:
- After the first solution has been inspected, open the Coil Parameters page as described on page 30, and delete the first item from the Current parameter list, so that the list becomes **4, 6, 8, 10.** Click Close.
- On the Solve menu, click Static 2D again. This solves for a current of 4 A.
- Repeat this process for currents of 6, 8 and 10 A.
- For each of the solutions in turn, inspect the solution as described below.

Post-processing – 1

The instructions below are applicable when MagNet is licensed for parameterization. For the Trial Edition, only one problem ID will be visible in any of the lists, and it will not be possible to display animations or graphs of results.

1 Select the Field page of the Project bar.

Notice that the Problem ID number is set to 1.

2 View the contour plot and the shaded plot for |B| as before.

3 Click the Problem ID drop-down list on the Field page, and select 3, corresponding to a current of 6 A.

4 Click Update View to display the plots for problem 3.

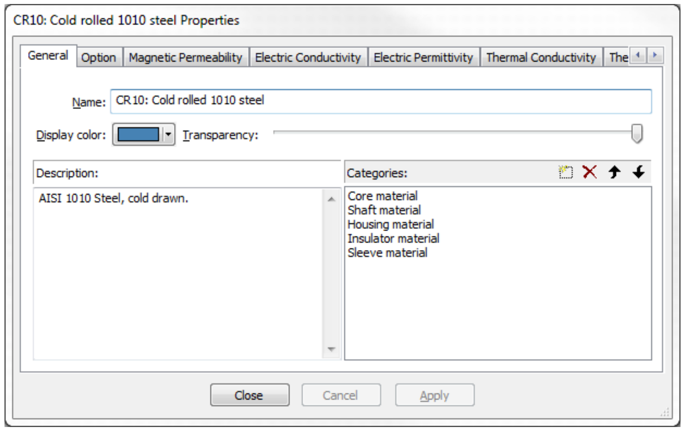
The flux lines are less smooth than with problem 1, because the solution is less accurate at the higher flux densities in this problem. It is possible to improve the accuracy by reducing the tolerance setting in the Adaption Options dialog.

5 Use the Probe Field Values tool as before to explore the flux density values in different parts of the model.

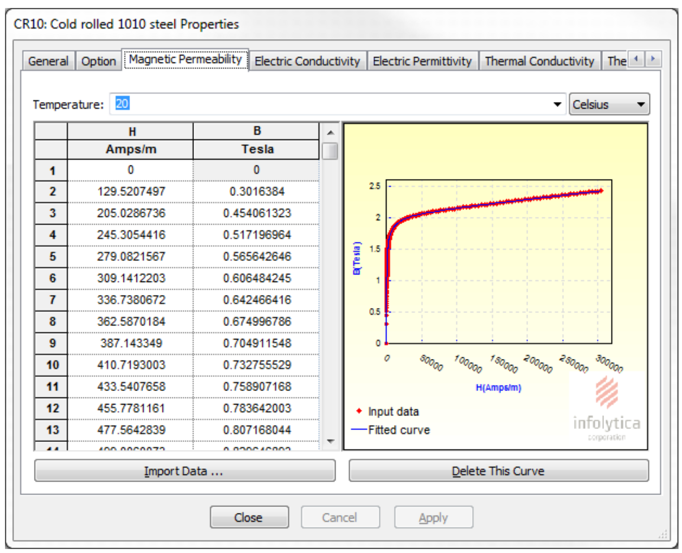
Notice that flux density values are much higher than before, exceeding 2 tesla in part of the core.

6 Display a graph of the *B-H* characteristic for the core material as follows:

- Select the Material page of the Project bar.
- In the Model Materials, right-click CR10: Cold rolled 1010 steel and select Properties to display a dialog:



- Click the Magnetic Permeability tab and resize the dialog box to display the *B-H* characteristic:



- Observe that the material saturates at about 1.8 T.
- Click Close.
- Use the Probe Field Values tool to find where the core is saturated.

7 Similarly, inspect the plots for Problem ID 5, corresponding to a current of 10 A.

8 In the Field page, click Animate.

MagNet generates an image of the plot for each problem ID, and then displays them in sequence in a new Animation window in the View area.

9 In the Animation window, use the buttons at the bottom of the display to control the speed etc.

10 Close the Animation window.

- This will display a Save Changes dialog. Click No.

Post-processing – 2

1 Display the Results window by clicking the Results tab.

2 In the Results window, click the Force tab.

Force values for problem 1 are displayed.

3 Click the Problem ID drop-down list on the Results window, and select Problem 2.

- Observe the force values for problem 2.

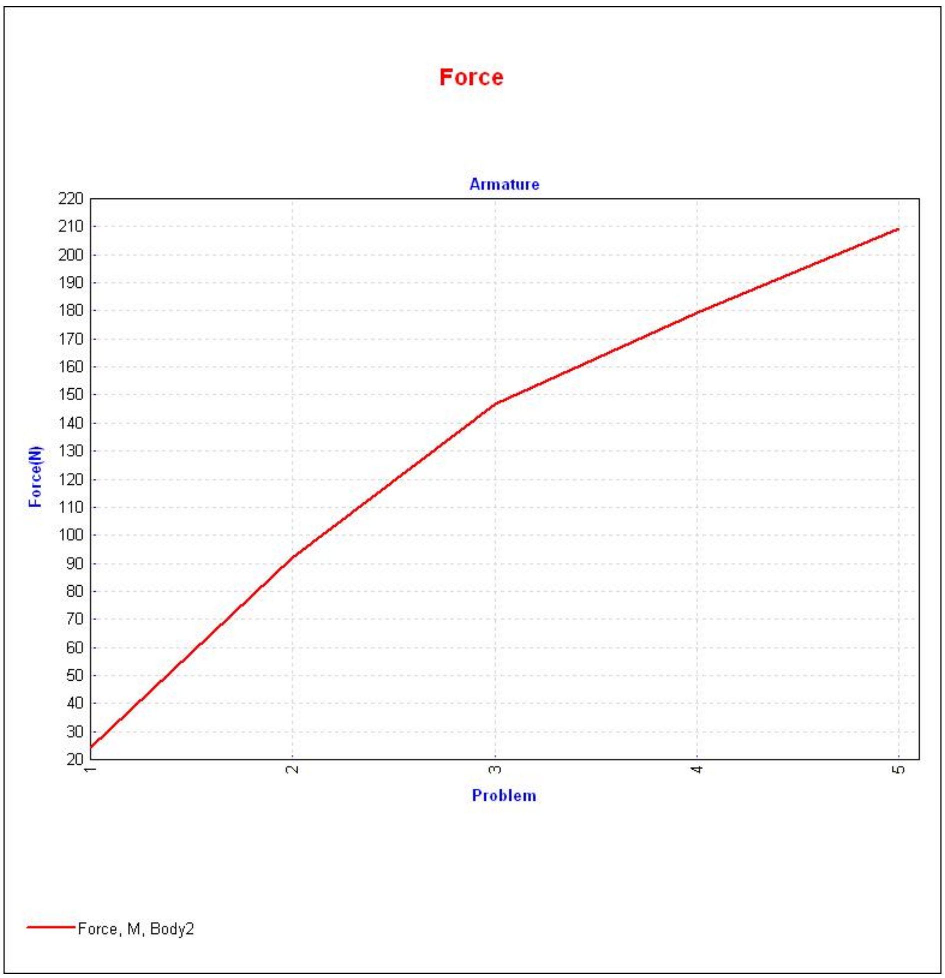
4 Similarly, observe the force values for the other problems.

5 Display a graph of force magnitude values as follows.

- For the Armature, click in the text box for the Force Magnitude.

  The Graph Selection… button should be enabled.

- Click the Graph Selection… button to display a graph:



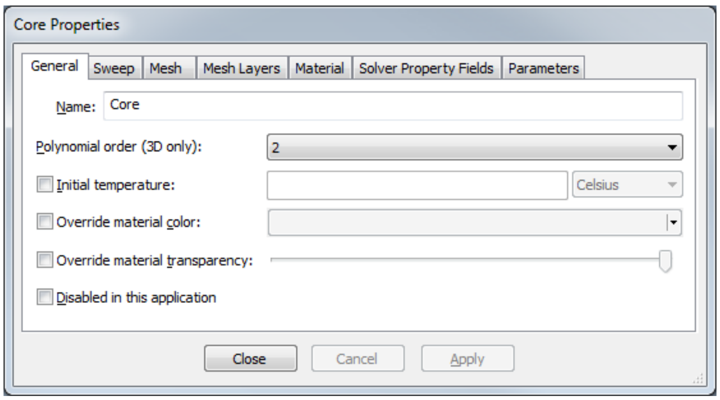
Note that the current values are not displayed in this graph, because MagNet uses this method to show the variation of force with any parameter. The current values corresponding to the problem numbers 1 to 5 are 2, 4, 6, 8 and 10 A respectively.

6. The force values might be expected to increase as the square of the current. Observe that the magnitude of the force does not increase in this way, because the core is saturating at high current values.
7. Close the graph window.

Altering the core material

The procedure below will change the core material from ordinary low-carbon steel (CR10) to a high-quality magnet iron (Remko). A similar procedure can be used to change the material of any other component in the model.

1. Display the model again:
   - On the View menu, click Solid Model.
2. Select the Object page of the Project bar.
3. Right-click the Core component, and select Properties to display a dialog:



4. Click the Material tab.
   - In the Material box, start typing **Remko: Soft pure iron**.
5. Click OK to apply this change and close the Core Properties dialog.

   The color of the core should have changed to represent the new material.
6. On the Solve menu, click Static 2D.
7. View the solution results as before.

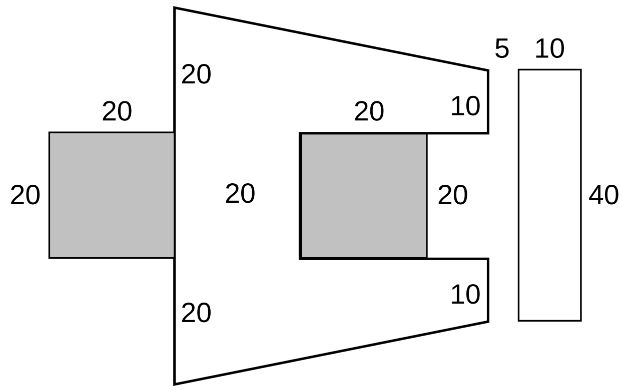
Remko has a higher saturation flux density than CR10, so a greater proportion of the core is unsaturated, resulting in a larger force on the armature. The results should be similar to the following, which were obtained with MagNet version 7.5.

Material	Force at 2 A	Force at 6 A	Force at 10 A
CR10	24.15 N	145.3 N	211.2 N
Remko	24.73 N	161.3 N	254.6 N

This change of material has made a small improvement. For significant improvement, the shape of the core must be changed.

Changing the geometry

With large values of coil current, part of the core is saturated magnetically, so the electromagnet is less effective. The diagram below shows a change to the shape of the core that will put more material in the critical areas, thereby reducing the value of the flux density in these parts of the core.



To make this change, the original model will be modified with the *shift* and *distort* facilities in MagNet. Before doing this, the original drawing lines will be deleted as described below.

1 Save the model as **C-core electromagnet mod**.

2 On the View menu, click Construction Grid.

This displays the construction grid again.

3 Click the Select Construction Slice Lines/Arcs button. 

4 Select all the lines and arcs by pressing Ctrl+A.

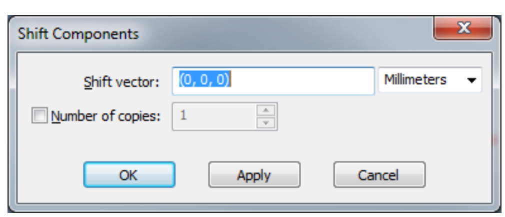
All the lines and arcs should be marked in red.

5 Press the Delete key.

All the marked lines should be deleted.

6 Move the left-hand coil side as follows:

- In the Object page, click CoilSide#1.
- On the Model menu, click Shift Components to display a dialog:



- In the Shift Vector text box, change the text to **(–10, 0, 0).**
- Click OK to apply this shift and close the dialog.

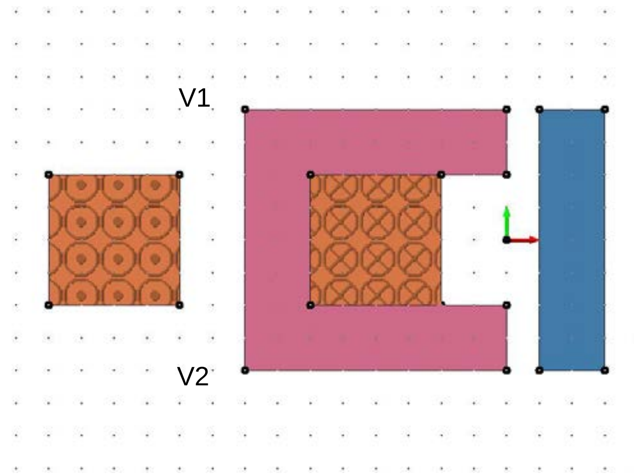
The left-hand coil side should have moved 10 mm to the left.

7 Change the shape of the core as follows:

- On the Model menu, click Distort Vertices, or click the Distort Vertices button.

  The model vertices are marked in the View 1 window:

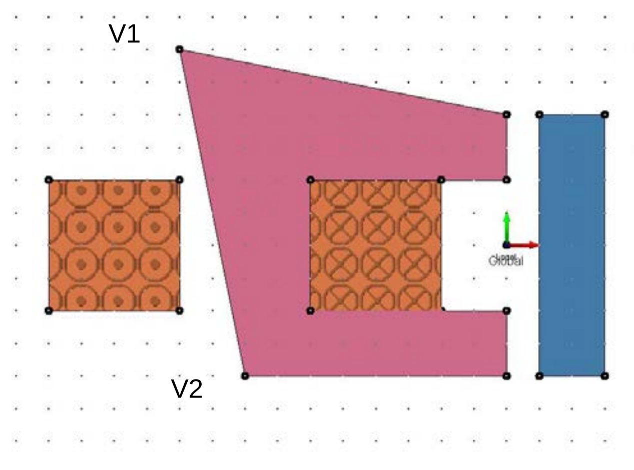
- Click the vertex labeled V1 in the diagram.

  The selected vertex is colored red.

- Move the pointer to the required position, displaced from the original position by 10 mm in both *x* and *y*. Do not press the mouse button while doing this.

  "Rubber band" lines follow the pointer.

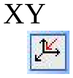
- Click again to move the vertex to the new position:

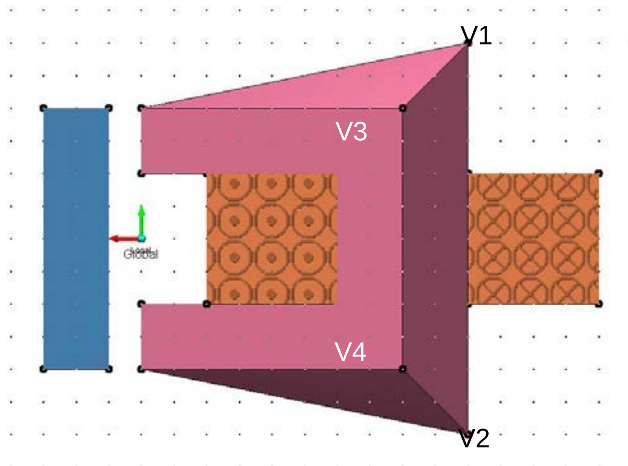
- Repeat for the vertex V2.

8 This has only changed part of the component. Complete the change as follows:

- On the View menu, click Preset Views / Negative Z axis, or click the Show XY (–Z) button.

  This shows the back view of the model (see the next page):



- Move vertices V3 and V4 to the same positions as V1 and V2.
- On the View menu, click Preset Views / Positive Z axis, or click the Show XY (+Z) button. 

  This restores the original view.

9 On the Solve menu, click Static 2D.

10 View the solution results as before.

With a core material of Remko, the results should be similar to the following, which were obtained with MagNet version 7.5.

Model	Force at 2 A	Force at 6 A	Force at 10 A
Original	24.15 N	145.3 N	211.2 N
Modified	25.26 N	223.1	444.3

In the absence of saturation, the force would increase by 9 times when the current increases from 2 A to 6 A, and by 25 times when the current increases from 2 A to 10 A. With the modified model, the force increase is 8.8 times at 6 A, and 17.6 times at 10 A. Thus, the change has almost eliminated saturation for currents of up to 6A. There is considerable improvement in the performance at 10 A, but saturation is still significant at this current.

Moving the armature

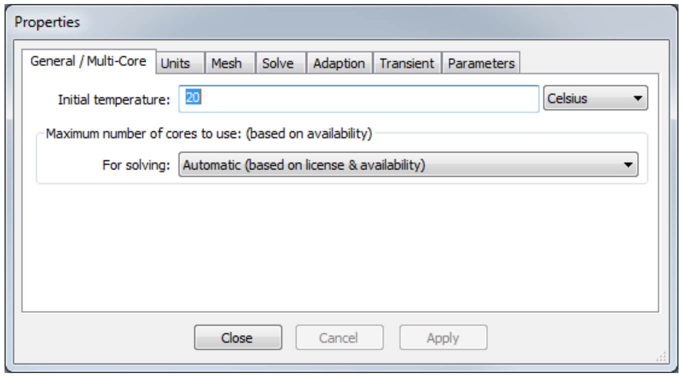
If the armature is displaced from its position of alignment with the magnet poles, there will be a restoring force that varies with the displacement. To examine this effect, another form of parameterization will be used to move the armature component relative to the rest of the model. This introduces the topic of *user-defined parameters.*

In the Model Properties dialog, there is a Parameters page where the user can create a new named parameter and give it a range of values. The new parameter can be used to modify other properties of model components, so that they depend on one parameter. In this case, a user-defined parameter will be used to vary the Y coordinate of a *shift vector* that determines the displacement of the armature.

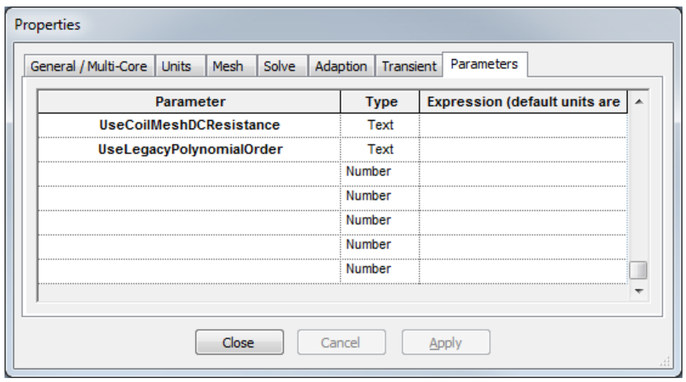
It is necessary to remove the list of values from the coil current parameter, as described in the instructions on the next page; otherwise MagNet will generate a new problem for each combination of values for the current parameter and the shift parameter.

Creating a user-defined parameter

1 In the Object page, right-click the model name C-core electromagnet mod, and select Properties to display a dialog:



2 Select the Parameters page and scroll down to the end of the list of parameters:



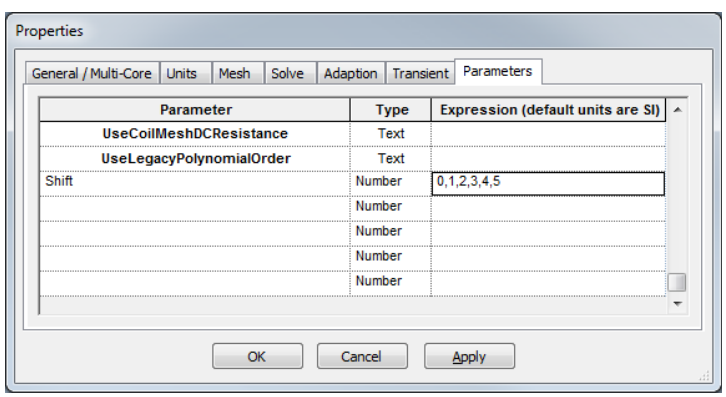
3 In the first vacant Parameter field, enter **Shift**.

The Type drop-down list for this parameter should already be set to Number.

4 In the Expression field, enter the following list of values:
0, 1, 2, 3, 4, 5

5 Press Enter to accept the list; the dialog should change as shown below.

If the field turns red, there is an error in the field, which must be corrected.



6 Click Apply to change the properties.

This leaves the Properties dialog open.

Parameterizing the model

1 In the Object page, click Coil#1.

The dialog changes to Coil#1 Properties, with Parameters displayed.

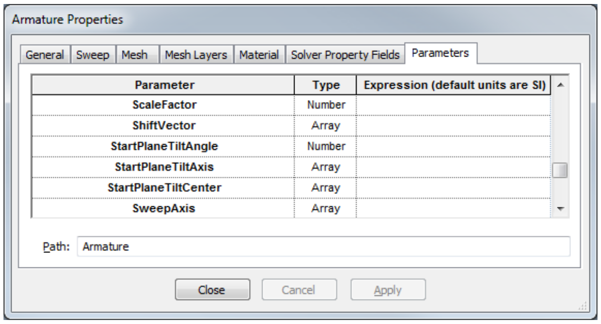
2 Edit the list of Current values so that only the value 2 remains.

3 Press Enter to accept the change, and then click Apply.

4 In the Object page, click Armature.

The dialog changes to Armature Properties, with Parameters displayed.

5 Scroll down the list of parameters to find ShiftVector:



6 In the Expression field, type the following array for the vector, including the square brackets: **[0, %Shift%mm, 0]**

The name of the user-defined parameter Shift must be preceded by a % symbol. The suffix %mm converts values from millimeters to the basic units of meters.

7 Press Enter to accept the array.

If the field turns red, there is an error in the field, which must be corrected.

8 Click OK to close the Properties dialog.

9 Select the Problem page of the Project bar.

- Observe that five problems have been created, with different shift vectors.

Solving and post-processing

1 Solve as Static 2D.

With the Trial Edition of MagNet, only the first problem will be solved automatically. See page 31.

2 In the Results window, click the Force tab.

3 Click in the text box for the X component of force on the armature.

4 Click Graph Selection.

A graph of the x component of force should be displayed.

5 Click Results, and in a similar way, display a graph of the Y component of force.

The results should be similar to the following, obtained with MagNet version 7.5.

Shift (mm)	0	1	2	3	4	5
–Fx (N)	25.26	24.95	24.53	23.78	22.82	21.73
–Fy (N)	0.00	0.91	1.70	2.50	3.18	3.83

Postscript

The table below compares the 2D results for the original C-core electromagnet with the results from a 3D model, using MagNet version 7.5, at a current of 2.0 A.

	2D	3D
Stored magnetic energy	0.3202 J	0.4869 J
Co-energy	0.3216 J	0.4939 J
Flux linkage	0.3209 Wb	0.4905 Wb
Self-inductance	160.5 mH	245.3 mH
Force on the armature	24.15 N	24.53 N

These results show that the 2D model predicts the force of attraction with good accuracy, but it seriously under-estimates the flux linkage and hence the inductance of the coil. These results may be explained by considering the 3D model shown below.

In the middle of the airgap, the magnitude of the flux density is approximately 0.249 T. If the field were confined to the area of the pole face, with a uniform value of 0.249 T, equation B-13 of Appendix B gives the force of attraction as

$$f_n = \frac{B^2 A}{2\mu_0} = 19.74 \text{ N}$$

The 2D result of 24.15 N is 22.3% higher than the idealized value. This increase can be attributed to the contribution of the fringing flux in the region of the long edges of each pole face, visible in the flux plot on page 24, which increases the effective area of the pole face. In the 3D model, there is a further small contribution from the fringing flux in the region of the short edges, giving a force value that is 1.6% higher than the 2D result.

The flux linkage with the coil, on the other hand, is increased significantly by two effects. The leakage field between the poles will extend well beyond the device in the third dimension; this will increase the flux in the yoke, giving a corresponding increase in the flux linkage with the coil. Additionally, the curved ends of the coil will generate a magnetic field which is ignored in the 2D solution, and this will further increase the flux linkage. Consequently, the 3D solution gives an inductance value that is 53% higher than the 2D result. There is a similar increase in the stored magnetic energy.

Chapter 3

Case Studies: Translational Geometry

Introduction

The case studies in this chapter cover a range of modeling problems for devices with translational geometry. Devices with rotational geometry are discussed in chapter 4. These case studies are arranged in order of increasing difficulty, progressively introducing further features of MagNet, so it is advisable to work through them in sequence. The detailed descriptions of basic MagNet operations given in chapter 2 will not be repeated, but any new operations will be fully explained.

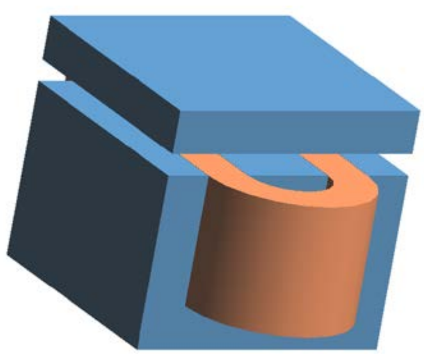
In some of the case studies, a 2D model does not always give accurate results because the device is not very long in the translational direction. Since the user needs to be aware of the limitations of 2D modeling, these case studies also include 3D results for comparison.

For all of the case studies in chapter 3 and chapter 4, the instructions assume that a new model is being started, as described in the tutorial in chapter 2. To avoid tedious repetition, this instruction is given in abbreviated form in the case studies.

Some of the case studies require shapes to be drawn from arcs and straight lines. Drawing an arc requires the user to specify the coordinates of the center, the start point and the end point. Arcs are normally drawn counter-clockwise from the start point to the end point, but the direction can be changed to clockwise by holding down the Ctrl key. If the arc drawing tool is selected from the Draw toolbar, the order of the points is center, start, end. The Draw menu has other arc tools with the points in different orders.

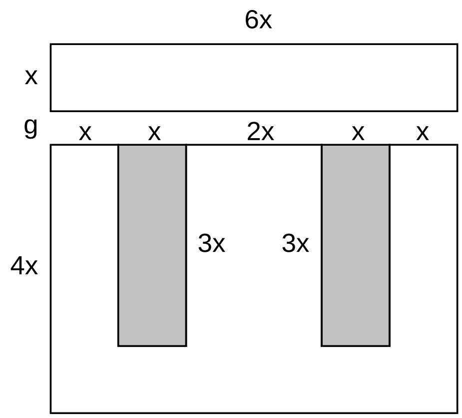
E-core electromagnet

The diagram below shows an E-core electromagnet. It is similar in principle to the C-core electromagnet of chapter 2, but it is a better magnetic design because the coil is nearer to the airgap, and both sides of the coil are active. The objectives are to determine the self-inductance of the coil and the force on the armature, and to explore the magnetic field distribution in the device.

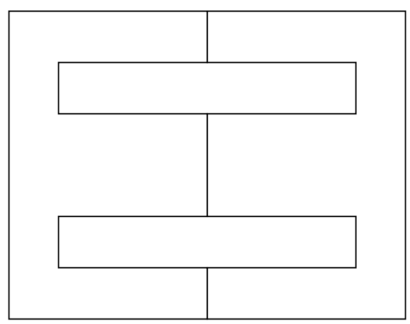


Modeling the device

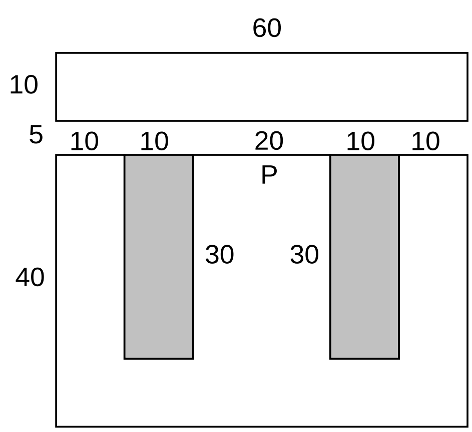
The diagram below shows the 2D model of the device, with the dimensions as multiples of a basic unit *x*; the airgap length *g* is independent of *x*. The shaded areas are the two coil regions, representing the two sides of the coil shown in the 3D view on the previous page.



The dimensions in this diagram are chosen so that the armature and the core can be made from laminations punched from sheet steel without any waste, as shown in the diagram below.



The basic dimension *x* is 10 mm, the airgap length *g* is 5 mm, and the depth of the electromagnet is 60 mm. The coil has 1000 turns, carrying a current of 2.0 A.



As with the C-core electromagnet, an air box must be drawn around the model with a radius of about 10 times the model dimensions.

Creating the model

1. Start a new model and save it as **E-core electromagnet**.
2. Set the model length units to millimeters.
3. The coordinate origin is the point P in the center of the top edge of the middle pole, show in the diagram on the previous page. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Construct components for the core and the armature:
   - Material: CR10: Cold rolled 1010 steel.
   - Sweep distance: 60 mm.
5. Construct components for the two coil sides:
   - Material: Copper: 5.77e7 Siemens/meter.
   - Sweep distance: 60 mm.
6. Make a single coil from the two components:
   - Number of turns: 1000.
   - Current: 2.0 A.
7. Construct an air box from a circle centered at the origin, with a radius of 400 mm:
   - Ignore Holes must be active in Make Component in a Line.
   - Sweep distance: 60 mm.

Solving and post-processing

The following solver settings should give an accurate 2D solution without excessive computing time. The user is invited to try the effect of different settings.

1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Solver Options:
     Polynomial order 2.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.2%.
2. Solve as Static 2D.
3. Inspect the contour plot of the flux function and the shaded plot of the |**B**| values.
4. Inspect the computed global quantities, and calculate self-inductance values as follows (see appendix B for details):
   - From the flux linkage: $L = \lambda / i$, where λ is the flux linkage for the coil, and i is the coil current.
   - From the stored energy: $L = 2W / i^2$, where W is the stored magnetic energy.

Sample results

The results below were obtained with MagNet version 7.5. For comparison, results from a 3D solution are also given.

	2D	3D
Stored magnetic energy	0.7535 J	0.8948 J
Co-energy	0.7564 J	0.8984 J
Force on core and coils	+64.09 N	+66.95 N
Force on armature	–63.91 N	–67.25 N
Flux linkage	0.7549 Wb	0.8981 Wb
Inductance from flux linkage	377.5 mH	449.0 mH
Inductance from stored energy	376.8 mH	447.4 mH

Discussion

2D model

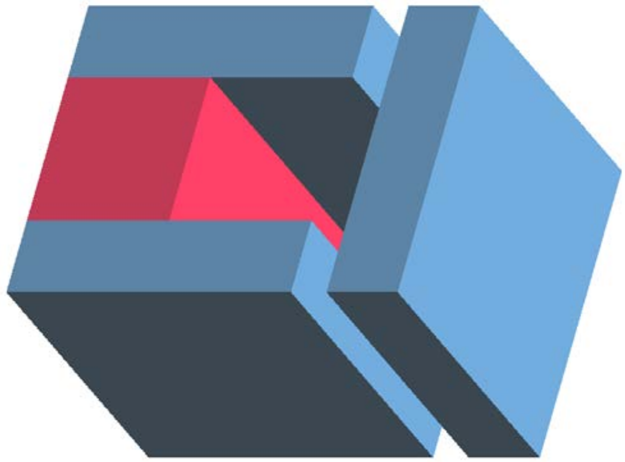
Since the airgap length and the coil ampere-turns are the same as for the C-core electromagnet, the airgap flux density values are expected to be similar. The total pole-face area is three times as great, so the force of attraction and the self-inductance are also expected to be about three times as great, which is the case. If the coil ends are neglected, the coil resistance will be twice that of the C-core electromagnet, because the coil cross-sectional area is halved. Thus, the E-core electromagnet has a lower power loss per unit force, and a larger value of time constant L / R, than the C-core electromagnet.

3D model

As with the C-core electromagnet, the 3D solution gives a significantly higher value for the self-inductance: in this case 19% higher than the 2D result. Similarly, the 3D solution gives a higher value for the force: 5.2% higher than the 2D result for the force on the armature. For comparison, with the C-core electromagnet the 3D solution gives an inductance value that is 53% higher than the 2D result, and a force value that is 1.6% higher.

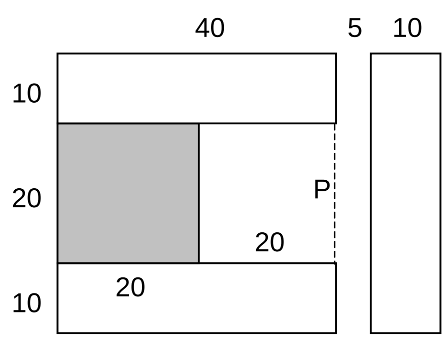
Magnetic latch with a permanent magnet

The diagram below shows a magnetic latch, which takes the form of a C-core permanent magnet with steel poles and a steel armature. This is similar to the C-core electromagnet of chapter 2, except that the excitation for the magnetic circuit is provided by a permanent magnet. The dimensions of the poles, armature and airgap are the same as for the C-core electromagnet, and the thickness of the permanent magnet is 20 mm.



Modeling the device

The diagram below shows the cross-section of the device with the dimensions in mm. All parts have a depth of 40 mm. The poles and the armature are made from 1010 cold-rolled steel, and the magnet from Alnico-9NB.



As with the C-core electromagnet, an air box must be drawn around the model with a radius of about 10 times the model dimensions.

Creating the model

1. Start a new model and save it as **PM latch**.
2. Set the model length units to millimeters.
3. The coordinate origin is the point P between the poles. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Construct components for the poles and the armature:
   - Material: CR10: Cold rolled 1010 steel.
   - Sweep distance: 40 mm.
5. Construct the component for the magnet:
   - Material: AL9N: Alnico-9NB.
   - Type: uniform.
   - Direction: (0, 1, 0).
   - Sweep distance: 40 mm.

   The direction data specifies that the block is uniformly magnetized, with the magnetization vector in the positive Y direction.
6. Construct an air box from a circle centered at the origin, with a radius of 400 mm:
   - Ignore Holes must be active in Make Component in a Line.
   - Sweep distance: 40 mm.

Solving and post-processing

The following solver settings should give an accurate 2D solution without excessive computing time. The user is invited to try the effect of different settings.

1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Solver Options:
     Polynomial order 2,
     Newton Tolerance 0.5%.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.1%.
2. Solve as Static 2D.
3. Inspect the contour plot of the flux function and the shaded plot of the |**B**| values.
4. Inspect the computed global quantities.

Sample results

The results below were obtained with MagNet version 7.5. For comparison, results from a 3D solution are also given.

	2D	3D
Stored magnetic energy	–0.6129 J	–0.9170 J
Co-energy	0.6129 J	0.9188 J
Force on poles and magnet	+37.57 N	+31.72 N
Force on armature	–37.56 N	–31.62 N

Discussion

Energy

The stored magnetic energy values are negative in this model because **B** and **H** are in opposite directions in the permanent-magnet material, so the energy density calculated from equation B-1 (Appendix B) is negative in this part of the model. When the total energy is calculated by integrating the energy density over the volume of the model, the negative energy in the permanent-magnet material exceeds the positive energy in the rest of the model. See the MagNet help for further information.

2D model

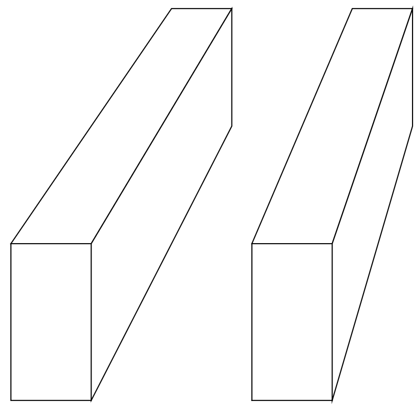
The airgap flux density is somewhat higher than in the C-core electromagnet, giving an increased force of attraction, and the maximum flux density in the steel is also higher. Because the permanent magnet has a recoil relative permeability close to 1, there is much more leakage flux at the back of the magnet than with the coil in the C-core electromagnet.

3D model

In this 3D model the force on the armature is 16% lower than the value from the 2D model. In contrast, with the C-core electromagnet, the force from a 3D model is 1.6% larger. The reduction in force can be explained as follows. There is significant flux leakage between the poles at the ends, which is ignored in the 2D model. Thus, more of the magnet flux is diverted into leakage paths, so that less flux crosses the airgap, giving a reduction in the force of attraction. This effect is greater if the width of the permanent magnet is reduced.

Busbar forces

The diagram below shows two long non-magnetic busbars, where the force due to the currents is to be calculated in two cases: currents in the same direction, and currents in opposite directions. For this case study, the bars are 0.2 m apart, 0.2 m high and 0.1 m wide. The current density is 5 A/mm^2, so the current in each bar is 100 kA.



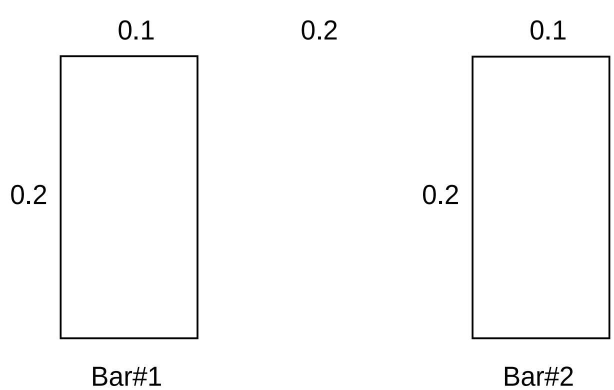
There is a simple analytical expression for the force per unit length between two infinitely long busbars in free space:

$$f = \frac{2 \times 10^{-7} k i_1 i_2}{d} \quad \text{[N/m]} \tag{3-1}$$

where i_1 and i_2 are the currents in the bars, k is a constant that depends on the shape of the conductors and their separation, and d is the distance between centers of the busbars. This is based on the formula given by Bewley [1] and Steele [2]. The value of k in this case study can be calculated from the analytical formula to be 0.94980 when d = 0.3 m, so the theoretical force is 6.3320 kN/m; this serves as a check on the results from MagNet.

Modeling the device

The diagram below shows the 2D model of the busbars, with dimensions in meters.



Theoretically the field extents to infinity, but an accurate representation of this open-boundary condition in MagNet depends on the directions of the currents. If the currents are in opposite directions, so that the net current is zero, then the Kelvin transformation method described in Appendix A can be used to represent the open boundary exactly. However, if the currents are in the same direction, this method cannot be used. The magnitude of the force will be the same as in the other case, but the field distribution will be different. If values of the flux density are required, a closed boundary must be used in the form of an air box with its outer surface sufficiently far from the busbars to model the open-boundary condition accurately.

Currents in opposite directions

At present, the Kelvin transformation method is not available as a standard feature of MagNet, so this section describes a manual implementation of the method. Scripting, described in chapter 5, can be used to automate this operation.

The open boundary representation has two parts: (a) a circular boundary surrounding the model, enclosing an air space; (b) a second small circular boundary, enclosing another air space, outside the first. Corresponding points on the two boundaries are linked so that the field values will be identical, by applying an *even periodic* boundary condition.

Creating the basic model

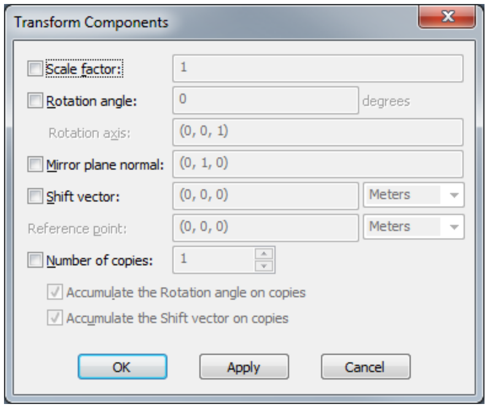
1. Save the new model as **Busbars 1**.
2. Set the model length units to meters.
3. The coordinate origin is in the center of the 2D model, with the busbars disposed symmetrically about the origin. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Construct a component for each busbar:
   - Name the components **Bar#1** and **Bar#2**.
   - Material: Copper: 5.77e7 Siemens/meter.
   - Sweep distance: 1 m.
5. Make a single coil from the two components:
   - Number of turns: 1.
   - Current: 1.0e5 A.
6. Select and delete all the construction-slice lines.

Adding the open boundary

1 Create an air box of radius 0.4 m, centered at the origin, named **AirSpace**.

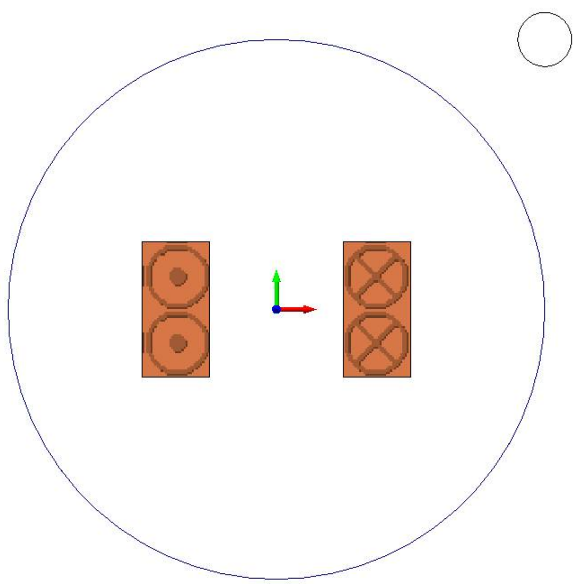
2 In the Object page, select the Airspace component.

3 On the Model menu, click Transform Components to display a dialog:



- Click Scale Factor and enter the value **0.1** in the text box.
- Click Shift Vector and edit the text box: (**0.4**, **0.4**, 0).
- Click Apply the Transformation to a Copy of the Selection.
- Click OK.

A new air box should appear in the View 1 window, as shown below, and there should be a new component in the Object page.



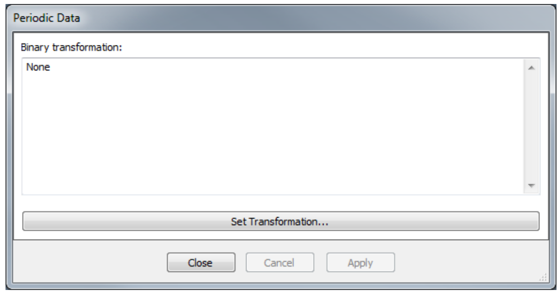
4 In the Object page, change the name of the new component to **Exterior**.

5 Display the tree directory for the Airspace component.

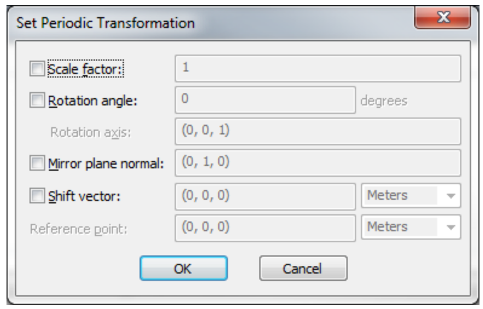
6 Hold down the Ctrl key, and select Face#3 and Face#4.

These are the half-cylinders which form the curved surface of the Airspace component, as can be seen by rotating the view of the model.

7 On the Boundary menu, click Even Periodic to display the Periodic Data dialog:



8 Click Set Transformation to display the Set Periodic Transformation dialog:



- Click Scale Factor and enter the value **0.1** in the text box.
- Click Shift Vector and edit the text box: (**0.4**, **0.4**, 0).

  These settings must be identical to those used in the Transform Components dialog to create the Exterior component.
- Click OK to close the Set Periodic Transformation dialog.
- Click OK to close the Periodic Data dialog.

A new item BoundaryCondition#1 (EP) should appear in the Object page. The curved surfaces of the two components will be marked with patterns representing the boundary condition, which can be seen by rotating the view of the model.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 2,
  CG Tolerance 0.001%.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.01%.

2 Solve as Static 2D.

3 Inspect the resulting solution mesh as well as the force values.

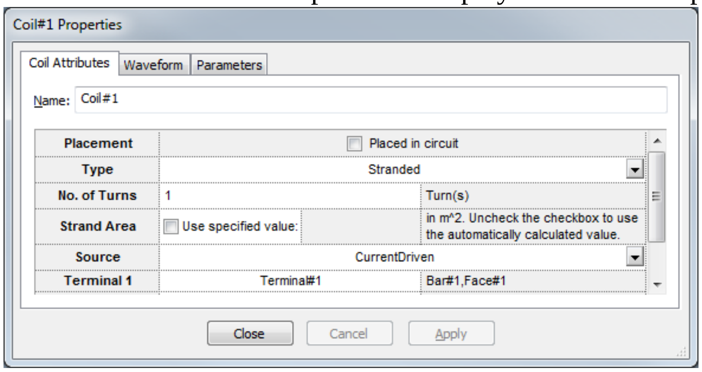
4 Compare the force with the theoretical value of 6.332 kN.

Currents in the same direction

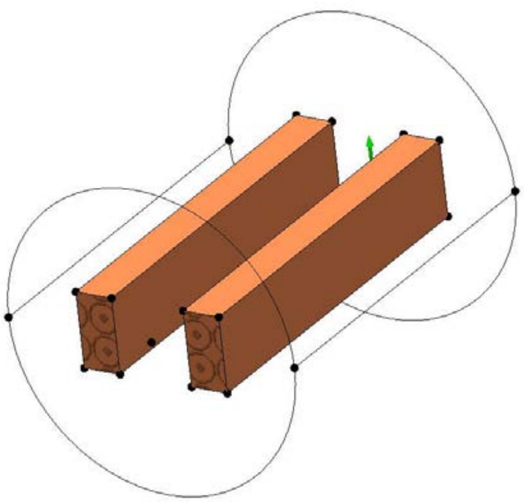
When the bars carry current in the same direction, the field lines will approximate to circles at large distances from the conductors. A circular air box is therefore a suitable choice for the closed boundary. The default Flux Tangential boundary condition is appropriate in this case, but the problem is to choose a suitable size for the air box. A range of sizes will be examined by making the radius of the air box a parameter.

Modifying the model

1. Open the model Busbars 1 and save the model as **Busbars 2**.
2. In the Object page, select and delete the following open boundary objects:
   - BoundaryCondition#1 (EP),
   - Exterior.
3. Right-click Coil#1 and select Properties to display the Coil#1 Properties dialog:

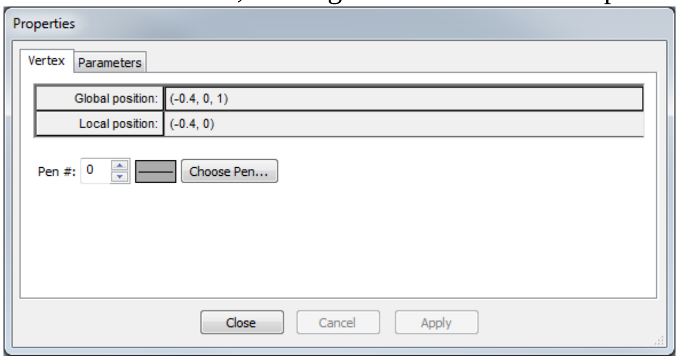


4. Scroll down and click Reverse Direction for CoilSide#2. Click OK.
5. Display the model Properties dialog. In the Parameters page, define a new numeric parameter **BoxRadius** with values:
   0.4, **1**, **2**, **4**
6. On the vertical View toolbar, click the Examine Model button. 
7. In the View 1 window, drag the pointer to rotate the model so that the front and back faces of the air box are visible:



8 Click the Select Component Vertices button.
The air box has four vertices, shown as black dots at the ends of the lines in the diagram on the previous page. These dots are not visible in the MagNet View window unless the Select and Distort Component Vertices button is selected. Be sure to use the Select Component Vertices button.

9 Click a vertex of the air box, then right-click and select Properties to display a dialog:



10 In the Vertex page, note the sign in the first Local Position value, e.g. (–0.4, 0)

11 Open the Parameters page.

12 In the Expression field for the Position parameter, enter an expression such as: **[–%BoxRadius, 0]** with the same sign as in the first Local Position value.

13 Click Apply.

14 Click the next vertex and repeat the procedure. Do this for all four vertices.

Viewing the parameterized model

1 In the Project bar, click the Problem tab.

2 Click problem 2, then click Update View.
Observe the increased size of the air box in relation to the busbars.

3 Similarly, view problems 3 and 4.
Observe the progressive increase in the radius of the air box.

Solving and post-processing

1 Changes to the default solving options are the same as those used for currents in opposite directions:

- Solver Options:
  Polynomial order 2,
  CG Tolerance 0.001%.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.01%.

2 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

3 Note the values of force obtained with different sizes of air box.

Sample results

The results below were obtained with MagNet version 7.5.

Currents in opposite directions

With the Kelvin open boundary, the computed force values are as follows:

Force on Bar#1 –6332.0 N

Force on Bar#2 +6332.0 N

Currents in the same direction

Values for the mean force magnitude on the two bars are as follows, for different air box sizes.

Air box radius	Force
0.4 m	6803.2 N
1 m	6344.0 N
2 m	6334.4 N
4 m	6332.3 N

Discussion

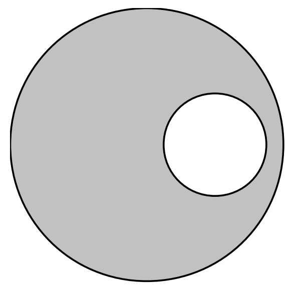
The analytical value for the force is 6332.0 N. With currents in opposite directions, the open boundary method gives a result very close to this value. With currents in the same direction, where a closed boundary is required, the force is over-estimated by 7.4% when the air box radius is 0.4 m. An air box radius of 2 m, which is about 10 times the radius of the model, is sufficient to give high accuracy in the force calculation. To achieve this accuracy, the default CG tolerance has been reduced to 0.001%, and the adaption tolerance has been set to a very low value of 0.01%. Nevertheless, this 2D model solves rapidly because there are no magnetic materials present.

Field in a cylindrical conductor

The magnetic field of a long solid cylindrical conductor is well known; the flux lines are circles centered on the axis. Outside the conductor, the magnitude of the flux density varies as $1/r$, where r is the radial distance from the axis. Inside the conductor, the magnitude varies directly with r, so the field vanishes on the axis.

If a cylindrical hole is bored in the conductor from one end to the other, there will be no magnetic field in this hole, provided the hole is coaxial with the conductor. In other words, there is no field inside a hollow cylindrical conductor.

A remarkable result, which is not widely known, is the nature of the field in the hole when it is bored off-center, so that its axis is displaced from the axis of the cylinder, as shown in the diagram below. This field is perfectly uniform; the flux lines are parallel straight lines in a direction normal to the plane containing the two axes [3].



Modeling the device

Although the dimensions of the cylinders are not critical, the size and position of the hole require a little care. In theory, the optimum size is half the diameter of the cylinder, positioned between the axis and the circumference. The hole then just breaks through the surface of the cylinder, but this is difficult to model in MagNet. If the circles are drawn so that the hole is a tangent to the cylinder, the material will be too thin and the model will not give accurate results. A satisfactory arrangement is shown in the diagram above, where the minimum thickness of material in the cylinder is 10 mm, the radius of the solid cylinder is 80 mm and the radius of the hole is 30 mm. A current of 69 kA will then give a flux density close to 0.1 T in the hole.

Creating the model

1 Start a new model and save it as **Cylindrical conductor**.

2 Set the model length units to millimeters.

3 The coordinate origin is in the center of the solid cylinder. Set the display to show a corresponding range of *x* and *y* values with a construction grid.

4 Construct the single component for this model from the area between two circles, with the center of the hole at (40, 0):

- Ignore Holes **must not be active** in Make Component in a Line.
- Material: Copper: 5.77e7 Siemens/meter.
- Sweep distance: 100 mm.

5 Make a coil from this component:

- Number of turns: 1.
- Current: 69000 A.

6 Construct an air box from a circle centered at the origin, radius 500 mm:

- Ignore Holes **must be active** in Make Component in a Line.
- Sweep distance: 100 mm.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 2,
  CG Tolerance 0.0001%.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.001%.

2 Solve as Static 2D.

3 Inspect the contour plot of the flux function and the shaded plot of $|B|$ smoothed.

Observe the nature of the flux lines in the hole. Use the Probe Field Values tool to get values of the flux density at several points in the hole.

4 Use the Field Extractor on the Extensions menu to obtain values for B_y over a grid of nine equally spaced points in the hole as follows:

- In the Field Name drop-down list, select B y.
- Check the *Smoothed* box immediately underneath.
- Click Add twice to add X and Y rows to the Variables box.
- In each row, change from the End method to the Spacing method.
- Enter the Start and Spacing values, and Numbers of sampling points, as follows:

X - Spacing	20,20	3
Y - Spacing	-20,20	3

- Click Extract and Output to File.

A set of nine values of B_y should be output to a .txt file, which should be opened automatically using the preferred application associated with this file type.

Sample results

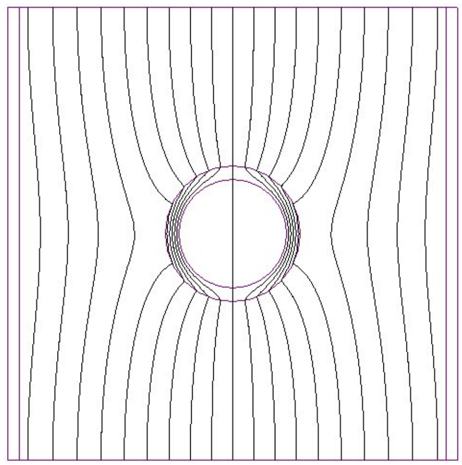
The table below shows the deviation of B_y from the mean value B_{y0} of 0.100001 T over a grid of nine equally spaced points in the hole. The quantity displayed is the percentage deviation $\Delta B_y / B_{y0}$. These results were obtained with MagNet version 7.5.

x (mm)→ *y* (mm)↓	20	40	60
–20	+0.00107%	–0.00003%	–0.00103%
0	+0.00117%	–0.00003%	–0.00119%
+20	+0.00087%	–0.00003%	–0.00083%

Cylindrical screen in a uniform field

Description of the problem

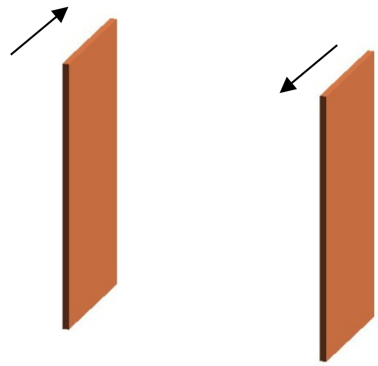
The diagram below shows the flux plot of a hollow iron cylinder placed in a transverse magnetic field. In the absence of the cylinder the field is uniform, so the diagram demonstrates the screening effect of the cylinder.



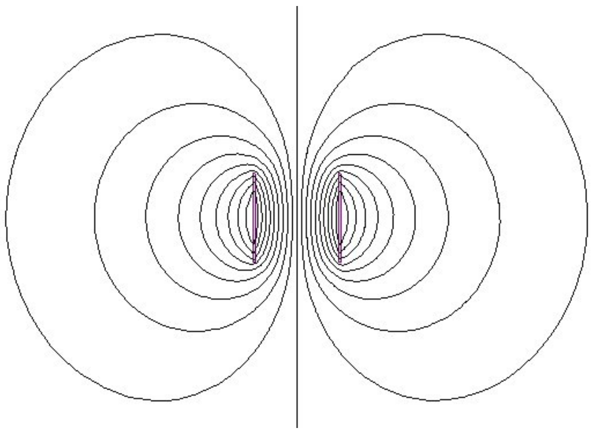
This problem makes an interesting case study for MagNet, firstly because the results can be compared with an analytical solution if the magnetic material is linear [3], and secondly because it illustrates a technique for producing a uniform magnetic field. For this case study, the inner radius of the cylinder is 24 mm, the outer radius is 30 mm, the relative permeability of the iron is 1000, and the magnitude of the applied field is 0.1 T.

Modeling the device

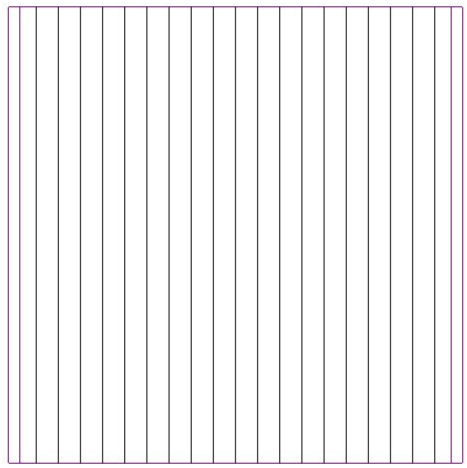
To produce a uniform magnetic field in MagNet, we use a “coil” formed from two slabs, as shown in the diagram below. The arrows show the direction of current flow, and the coil is assumed to be very long in this direction, giving a 2D field.



If this coil is surrounded by air, the field takes the form shown below.



As this flux plot shows, the field in the middle of the coil is approximately uniform. It can be made exactly uniform by a simple modification: the coil is enclosed in a close-fitting air box with a Field Normal boundary condition on four surfaces. The diagram below shows the resulting field pattern.



Boundary conditions are discussed in appendix A. This use of the Field Normal condition effectively embeds the device in a material of infinite permeability. Flux lines have to enter the top and bottom bounding surface at right angles, and they find return paths round the sides (not visible in MagNet) of zero reluctance. The result is a perfectly uniform field of finite extent, which is a useful approximation to the theoretical uniform field of infinite extent. When an iron cylinder is placed in this field, the result is shown in the flux plot on page 57.

The relationship between the coil current and the flux density is found from Ampère's circuital law – equation 1-2 of Chapter 1. The result is:

$$B_0 = \frac{\mu_0 i}{l} \quad [\mathrm{T}] \tag{3-2}$$

where B_0 is the flux density in the absence of an iron cylinder, i is the coil current, and l is the length of each coil side in the direction of the field. The result does not depend on the separation between the coil sides, or on their thickness.

For the screening problem, we need to generate a uniform field over a region that is significantly larger than the cylinder. A region 240 mm square should be satisfactory for a cylinder with an outer diameter of 60 mm. The required current is therefore:

$$i = \frac{B_0 l}{\mu_0} = \frac{0.1 \times 0.24}{4\pi \times 10^{-7}} = 19.10 \text{ kA} \tag{3-3}$$

Creating the model

1 Start a new model and save it as **Cylindrical screen 1**.

2 Set the model length units to millimeters.

3 The coordinate origin is in the center of the hollow cylinder, so the display should be adjusted to show a corresponding range of x and y values. Use a grid spacing of 6 mm to simplify drawing the component outlines.

4 Construct the cylinder from two concentric circles, of radii 24 mm and 30 mm.
- Material: MU3: Relative permeability 1000.
- Sweep distance: 100 mm

5 Construct the left-hand coil side from a rectangle 240 mm long and 6 mm wide:
- Left-hand edge: 120 mm from the origin.
- Name of the component: **CoilSide#1**.
- Material: Copper: 5.77e7 Siemens/meter.
- Sweep distance: 100 mm.

6 Make the right-hand coil side by copying, as follows.
- Select the left-hand coil side by clicking CoilSide#1 in the Object page.
- On the Model menu, click Shift Components.
- Click the check box for making a copy.
- Enter the following value for the shift vector:
  (**234, 0, 0**)
- Click OK.
- Rename the new component **CoilSide#2**.

7 Make a coil from the two coil sides:
- Number of turns: 1.
- Current: 19100 A.

Air box and boundary conditions

1. Construct a square air box that encloses the coil, with a side length of 240 mm:
   - Ignore Holes must be active in Make Component in a Line.
   - Sweep distance: 100 mm.
2. In the Object page, click the + symbol beside the air box component to display its tree directory.
3. Select four faces as follows:
   - Select Face#3 by clicking.
   - Hold down the Shift key and click Face#6.

   Four faces should be highlighted.
4. On the Boundary menu, click Field Normal.

   One boundary condition should appear in the Object page.
5. Use the Examine Model tool to inspect the faces of the air box.

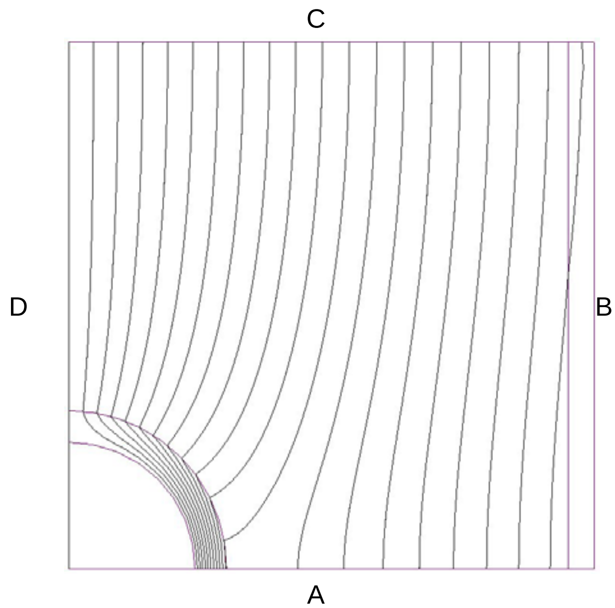
   Observe the pattern for the Field Normal boundary condition on four faces.

Solving and post-processing

1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Solver Options:
     Polynomial order 2,
     CG Tolerance 0.01%.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.01%.
2. Solve as Static 2D.
3. Inspect the flux plot and the shaded plot of |B|.
   - Use the Probe Field Values tool to explore the flux density values inside and outside the cylinder, and in the material of the cylinder.
   - Confirm that the value inside the cylindrical cavity is about 1% of the value far away from the cylinder.
4. Use the Field Extractor to explore B_x and B_y over a square grid of points, ranging from –15 mm to +15 mm in both X and Y directions, with a spacing of 5 mm. See page 56 for the procedure.
   - The values of B y smoothed should all be close to 1.165 mT.
   - The values of B x smoothed should be about 1000 times smaller.
   - These results show that the field is almost uniform in the hole.

Exploiting symmetry

It is possible to exploit the symmetry of this problem by modeling only one quarter of the device, as shown below. The coil current is half of the value for the full model, and the required boundary conditions are Field Normal on faces A, B and C, and Flux Tangential on face D. This is a useful technique for more complex models.



Creating the model

1 Start a new model and save it as **Cylindrical screen 2**.

2 Set the model length units to millimeters.

3 The coordinate origin is in the center of the hollow cylinder. Set the display to show a corresponding range of *x* and *y* values with a construction grid. Use a grid spacing of 6 mm to simplify drawing the component outlines.

4 Construct the quarter cylinder from two concentric arcs, of radii 24 mm and 30 mm, as follows:

- On the horizontal Draw toolbar, click the Add Arc button. 
- Click near the coordinate origin for the center of the arc.
- Click near the point (24, 0) for the first point on the arc.
- Click near the point (0, 24) for the second point on the arc.
- Draw the second arc of radius 30 mm in a similar way.
- Add two lines to complete the outline.
- Make the component, using the material MU3: Relative permeability 1000 and a sweep distance of 100 mm.

5 Construct the coil side from a rectangle 120 mm long and 6 mm wide:

- Right-hand edge: 120 mm from the origin.
- Material: Copper: 5.77e7 Siemens/meter.
- Sweep distance: 100 mm.

6 Make a coil from the single coil side:

- Number of turns: 1.
- Current: 9550 A (this is half of 19100 A for the full model).

Air box and boundary conditions

1. Select and delete all the construction-slice lines and circles.
2. Construct a square air box that encloses the coil:
   - Side 120 mm.
   - Sweep distance: 100 mm.
3. In the Object page, click the + symbol beside the air box component to display its tree directory.
4. Select three faces corresponding to A, B and C in the diagram as follows:
   - Select Face#3 by clicking.
   - Hold down the Shift key and click Face#5.

   Three faces should be highlighted.
5. On the Boundary menu, click Field Normal.
6. In a similar way, apply the Flux Tangential boundary condition to Face#6.
7. Use the Examine Model tool to rotate the model; inspect the faces of the air box, to check that the boundary conditions have been applied correctly.

Solving and post-processing

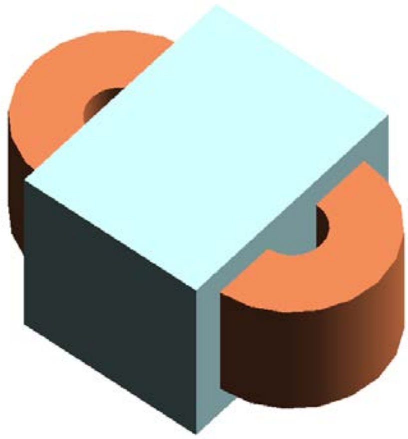
1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Solver Options:
     Polynomial order 2,
     CG Tolerance 0.01%.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.01%.
2. Solve as Static 2D.
3. Inspect the flux plot and the shaded plot of |B|.
4. Check that the flux density values are similar to those obtained with the full model of the cylinder and coil.

Discussion

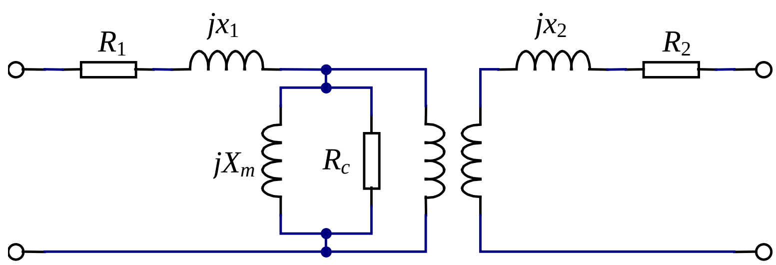
The theoretical value [3] for the flux density inside the cylindrical cavity is 1.101 mT for the cylinder used in this case study, when it is placed in a uniform field of 100 mT, of infinite extent. When averaged over a square grid of 49 points in the hole, the value in the full model is 1.165 mT. The difference of 5.6% can be attributed to the approximate representation of an infinite field. In practice this error is unimportant, because the screening effect is strongly dependent on the relative permeability of the screen, which is likely to be highly variable in practice. To see this effect, try replacing the linear material MU3 with a non-linear material such as CR10.

Transformer equivalent circuit

The diagram below shows a conventional shell-type transformer with the secondary wound over the top of the primary. For simplicity, the windings have equal numbers of turns in this case study.



The full equivalent circuit for the transformer is shown below, where x_1 and x_2 are the primary and secondary leakage reactances respectively, and X_m is the magnetizing reactance.



These reactances are related to the magnetic field through the inductances:

$$x_1 = \omega l_1, \;\; x_2 = \omega l_2, \;\; X_m = \omega M \qquad (3\text{-}4)$$

where ω is the angular frequency. Although this appears to be a time-harmonic problem, a static solution will be sufficient for determining these inductances because transformers are normally designed to minimize eddy-currents in the core.

When the windings have equal numbers of turns, the leakage inductances l_1 and l_2 are defined as follows:

$$l_1 = L_1 - M, \;\; l_2 = L_2 - M \qquad (3\text{-}5)$$

where M is the mutual inductance, and L_1 and L_2 are the self-inductances of the two windings. These leakage inductances represent flux produced by one winding which fails to link with the second winding, but “leaks” into the surrounding air (or other non-magnetic material) instead. From appendix B, the mutual inductance is given by

$$M = \frac{\lambda_{21}}{i_1} = \frac{\lambda_{12}}{i_2} \qquad (3\text{-}6)$$

where λ_{21} is the flux linkage with winding 2 when winding 1 carries a current i_1, and λ_{12} is the flux linkage with winding 1 when winding 2 carries a current i_2.

MagNet calculates flux linkages for each coil, so it is possible in principle to determine L_1, L_2 and M, and hence to find the values for the reactances in the equivalent circuit. A potential difficulty with this approach, however, is that the leakage inductances in equation 3-5 are the small differences between large quantities. Any errors in the calculation of L_1, L_2 and M will be magnified enormously in the resulting values of l_1 and l_2. A similar problem occurs in experimental work, where it is not possible to measure L_1, L_2 and M with sufficient accuracy to get good results for l_1 and l_2 by subtraction.

Experimentally, the reactance parameters are determined from open-circuit and short-circuit tests. In the open-circuit test, the secondary is open-circuited so that $i_2 = 0$, and the normal voltage is applied to the primary winding. Measurements at the primary terminals give the value of the total primary reactance $x_1 + X_m$.

In the short-circuit test, the secondary terminals are short-circuited and a low voltage is applied to the primary, sufficient to circulate the normal full-load current. The secondary current is then close to its normal full-load value. Under these conditions the current flowing in the magnetizing reactance X_m is negligible, so measurements at the primary terminals give the total leakage reactance $x = x_1 + x_2$. It is not possible to determine x_1 and x_2 separately by this method, so it is usually assumed that each is equal to half of the total.

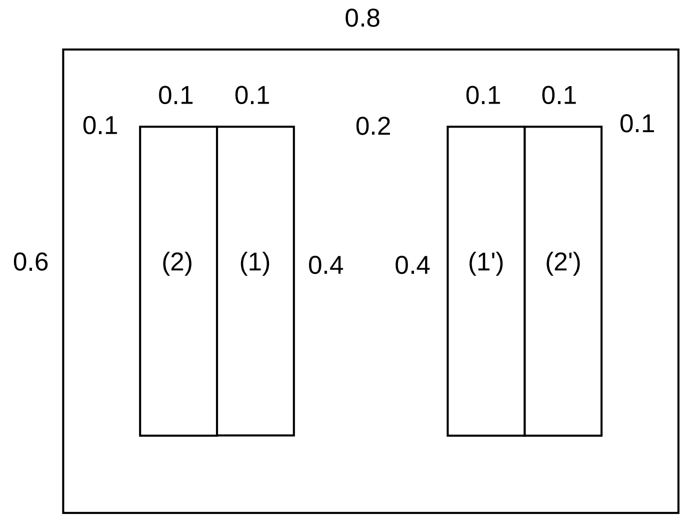
To simulate the conditions of a short-circuit test in MagNet, the windings are supplied with equal and opposite currents. From appendix B, the sum of the leakage inductances is given by

$$l_1 + l_2 = \frac{2W_s}{i_s^2}, \qquad (3\text{-}7)$$

where i_s is the current in each winding and W_s is the stored energy.

Modeling the device

The diagram below shows the model for the transformer, with the dimensions in meters. The core depth, perpendicular to the plane of the diagram, is 0.6 m. The primary coil sides are labeled 1 and 1'; the secondary coil sides are labeled 2 and 2'.



Each coil has 1000 turns. Under no-load conditions, when only one coil carries current, the current is 2.0 A. For the simulated short-circuit condition where the coils carry equal and opposite currents, the current is 200 A, giving a current density of 5 MA/m^2 (or 5 A/mm^2).

For this problem it is necessary to use a linear material, because equations 3-5 to 3-7 are valid for a linear system only. A suitable choice is MU3, with a constant relative permeability of 1000, which is reasonably representative of transformer steel under normal operating conditions.

Because there is virtually no external field, the air box can be quite close to the core of the transformer.

Creating the model

1. Start a new model and save it as **Transformer**.
2. Set the model length units to meters.
3. The coordinate origin is in the center of the transformer. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Construct a component for the core:
   - Ignore Holes **must not be active** in Make Component in a Line.
   - Material: MU3: Relative permeability 1000.
   - Sweep distance: 0.6 m.
5. Construct components for the four coil sides:
   - Material: Copper: 5.77e7 Siemens/meter.
   - Sweep distance: 0.6 m.

6 For the primary winding, make a single coil from two components by selecting the Start Face of side 1 and the End Face of side 1'.

- Number of turns: 1000.
- Current: 2.0 A.

7 For the secondary winding, make a single coil from two components by selecting the end face of side 2 and the start face of side 2'. This is the opposite of the primary winding.

- Number of turns: 1000.
- Current: 0 A.

8 Construct an air box from a rectangle with vertices at the following points: (–0.5, –0.4), (0.5, –0.4), (0.5, 0.4), (–0.5, 0.4).

- Ignore Holes **must be active** in Make Component in a Line.
- Sweep distance: 0.6 m.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

This solution is for the no-load condition with the primary energized.

3 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

4 Inspect the computed global quantities and calculate inductances as follows (see appendix B for details):

- Self-inductance: $L_1 = \lambda_{11} / i_1$, where λ_{11} is the flux linkage with the primary, and i_1 is the coil current.
- Mutual inductance: $M = \lambda_{21} / i_1$, where λ_{21} is the flux linkage with the secondary.

5 Change the primary current to 0 A and the secondary current to 2.0 A. Solve again and calculate the inductances:

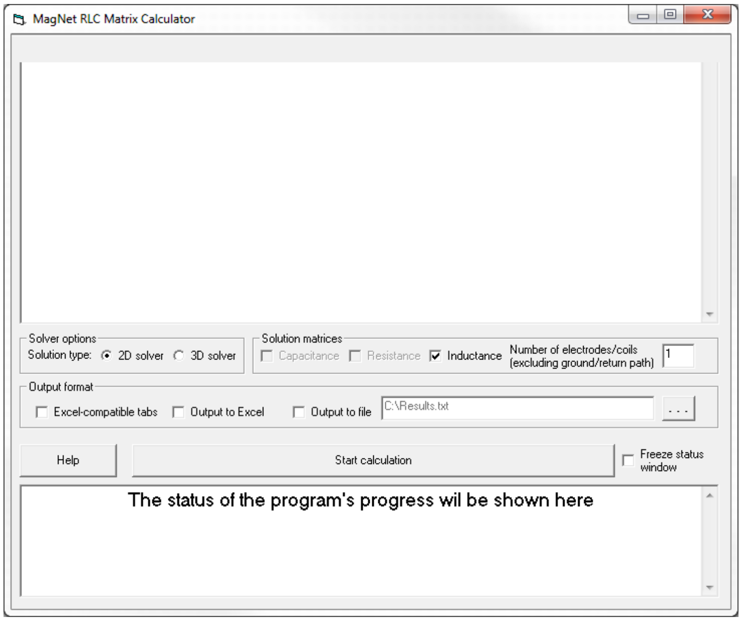
- Self-inductance: $L_2 = \lambda_{22} / i_2$, where λ_{22} is the flux linkage with the secondary, and i_2 is the coil current.
- Mutual inductance: $M = \lambda_{12} / i_2$, where λ_{12} is the flux linkage with the primary.

6 Change the primary current to 200 A and the secondary current to 200 A. Solve again and calculate the total leakage inductance from equation 3-7

Using the RLC Matrix Calculator

In this linear model, the inductance values are independent of the currents. For this case, the inductances can be determined automatically with one of the standard extensions to MagNet.

1 On the Extensions menu, click RLC Matrix Calculator to display the Calculator dialog:



- Context-sensitive help will be displayed in the status window at the bottom.
- For this model, all of the default settings will be satisfactory.

2 Click the Go button to start the calculation.

- If necessary, click Yes to save the model.

3 The program will solve the model with a current of 1 A in each coil in turn, and display the inductance values in the results window.

- The results given by the Calculator should be identical to those found from the current and flux linkage values.

Sample results

The results below were obtained with MagNet version 7.5. For comparison, results are also given for a 3D model of the same device.

$i_1 = 2.0$ A, $i_2 = 0$	**2D**	**3D**
Stored magnetic energy:	212.1877 J	212.9218 J
Flux linkage λ_{11}:	212.1877 Wb	212.9218 Wb
Flux linkage λ_{21}:	–211.9368 Wb	–212.5080 Wb
Self-inductance L_1:	106.0939 H	106.4609 H
Mutual inductance M:	105.9684 H	106.2540 H
Leakage inductance l_1:	0.1255 H	0.2069 H
$i_1 = 0$, $i_2 = 2.0$ A		
Stored magnetic energy:	212.1884 J	212.9792 J
Flux linkage λ_{22}:	212.1884 Wb	212.9793 Wb
Flux linkage λ_{12}:	–211.9373 Wb	–212.5077 Wb
Self-inductance L_2:	106.0942 H	106.4896 H
Mutual inductance M:	105.9687 H	106.2538 H
Leakage inductance l_2:	0.1255 H	0.2358 H
$i_1 = 200$ A, $i_2 = 200$ A		
Stored magnetic energy:	5020 J	8838 J
Leakage inductance $l_1 + l_2$:	0.2510 H	0.4419 H

Discussion

2D results

The first two solutions give very similar values for self-inductance, mutual inductance and leakage inductance. At first sight this is surprising because the primary winding occupies the space between the secondary and the center limb of the core, so the two are not obviously equivalent. But in a 2D model, if the core were infinitely permeable the current and flux patterns would have a symmetry that results in equal inductance values. Evidently a permeability of 1000 is sufficient to give very similar results.

The result from the energy calculation is very close to the sum of the two leakage reactances computed from the separate flux linkages. This is useful confirmation of the accuracy of the flux calculation, and shows that it is sometimes possible to obtain a good value for the leakage reactance by subtraction.

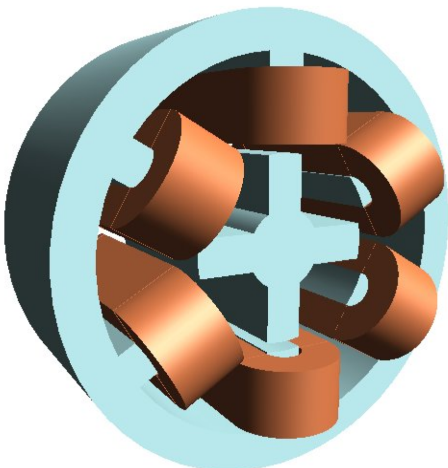
3D results

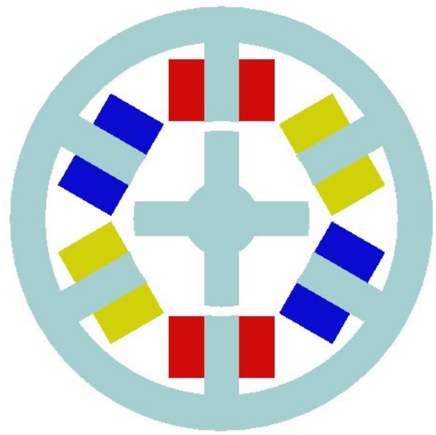
These results are significantly different from the 2D solution. Both of the leakage inductances are larger than their 2D counterparts because of the end-winding field, and as expected, the secondary value is considerably larger than the primary value.

The leakage inductance from the energy calculation is slightly smaller than the sum of the two leakage reactances computed from the separate flux linkages, which is an indication of numerical errors in the 3D results.

Variable Reluctance Stepper Motor

The diagrams below show 3D and 2D models of a simple variable-reluctance stepper motor.





The motor has a 3-phase stator, with two coils in each phase shown by the colors in the 2D model. Here, the red phase is energized, pulling one pair of rotor poles into alignment with the corresponding stator poles. Successively energizing the red, yellow and blue coils will make the rotor move through successive rotational steps of 30º.

Modeling the device

Although it would be possible to exploit the symmetry of the structure and model only half of the device, it is simpler to model the complete motor. Solutions are obtained quickly in 2D, so there is only a small penalty in computing time.

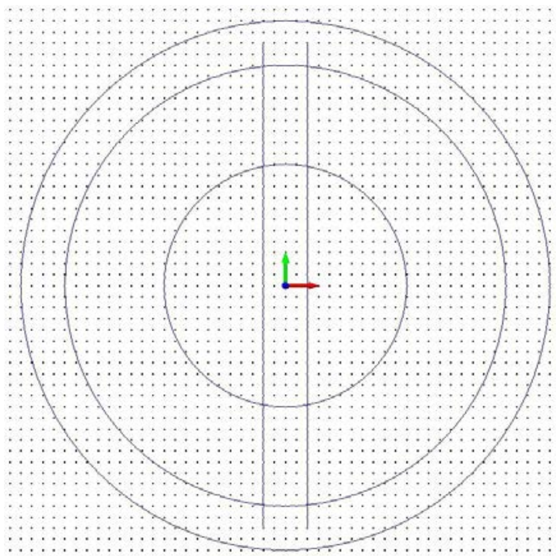
Key dimensions of the motor are as follows:

- Stator outer radius 120 mm
- Stator core radial depth 20 mm
- Stator inner radius 55 mm
- Stator pole width 20 mm
- Stator coil width 60 mm
- Stator coil depth 45 mm
- Rotor outer radius 50 mm
- Rotor hub radius 20 mm
- Rotor pole width 20 mm
- Motor axial length 100 mm

Building the stator core and poles

The stator core and poles will be made by drawing intersecting lines and circles, and deleting the unwanted parts.

1. Start a new model and save it as **Stepper motor**.
2. Set the model length units to millimeters.
3. The coordinate origin is in the center of the motor, so the display should be adjusted to show a corresponding range of *x* and *y* values. Use a grid spacing of 5 mm to simplify drawing the component outlines.
4. Draw circles of radius 120 mm, 100 mm, and 55 mm.
5. Draw two vertical lines of length 220 mm, spaced 20 mm apart, which will form the two vertical poles. At this stage, the drawing should resemble the following:



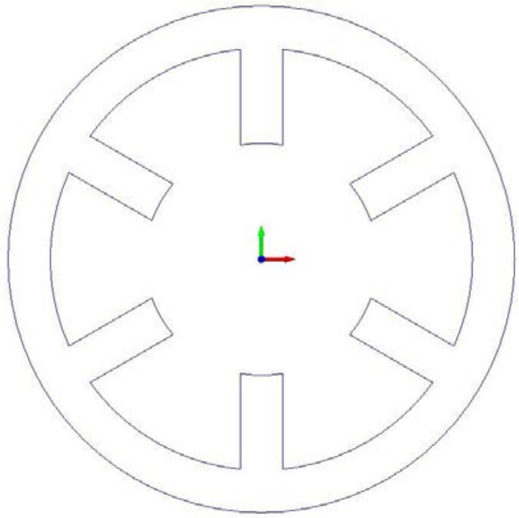
6. Select the two lines.
7. On the Draw menu, click Rotate Edges.
8. In the Rotate Edges dialog, enter the rotation angle as 60 degrees and click the check box for Apply the Transformation to a Copy of the Selection.

   This should activate Accumulate Rotation Angle on Apply.
9. Click Apply, and then click OK.

   This should create two more sets of lines for the other poles.
10. Select all the lines and circles with Ctrl+A.
11. On the Draw menu, click Segment Edges.

    This divides the lines and arcs at the intersections.

12 Select and delete the unwanted line and circle segments, to leave the stator core and poles:



13 Select the stator surface and make the component in a line:

- Material: MU3: Relative permeability 1000.
- Sweep distance: 100 mm.

Building the stator coils

One coil will be made and then copied to the other locations.

1 Draw the two sides of the top stator coil, as rectangles of 20 mm by 35 mm.

2 Make the two components named CoilSide#1a and CoilSide#1b:

- Material: Copper: 5.77e7 Siemens/meter.
- Sweep distance: 100 mm.

If desired, new conductor materials with red, yellow and blue colors can be defined for the coils. See the next case study on page 76.

3 In the Object page, select both coil components.

4 On the Model menu, click Rotate Components.

5 In the Rotate Components dialog, enter the rotation angle as 60 degrees, and click the check box for Apply the Transformation to a Copy of the Selection.

This should activate Accumulate Rotation Angle on Apply.

6 Click Apply four times, then click OK.

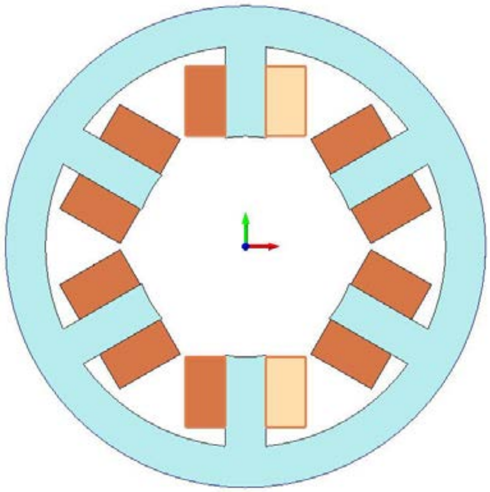
7 Rename the copies CoilSide#2a, CoilSide#2b, etc.

8 Identify the four coil sides on the vertical poles and display the tree directories.

These should be CoilSide#1a, CoilSide#1b, CoilSide#4a, and CoilSide#4b.

9 Starting with CoilSide#1a, select Face#1.

10 Hold down the Ctrl key and select the appropriate face (Face#1 or Face#2) of each of the other three coil sides, so that the two coils will act in the same direction when they carry the same current. The display in the View 1 window should resemble the following:



11 Make a simple coil:

- Number of turns: 1000
- Current: 2 A.

12 In a similar way, make the coils for the other two phases, but with the current set to 0.

Building the rotor and the air box

1 Build the rotor in a similar way to the stator, starting with two circles and two sets of straight lines:

- Material: MU3: Relative permeability 1000.
- Sweep distance: 100mm.

Since the rotor pole sides are parallel to the coordinate axes, it is not necessary to use the Rotate Edges facility to draw the second set of lines.

2 Select and delete the construction slice lines and arcs.

3 Construct an air box from a circle of radius 120 mm.

- Material: AIR
- Sweep distance: 100 mm.

There is negligible flux leakage outside the stator, so there is no need to leave a gap between the air box and the stator.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

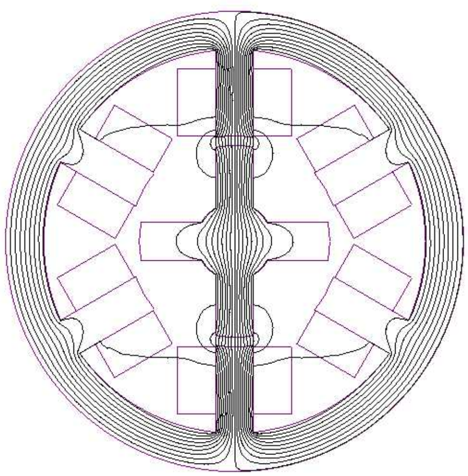
- Solver Options:
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.1%.

2 Solve as Static 2D.

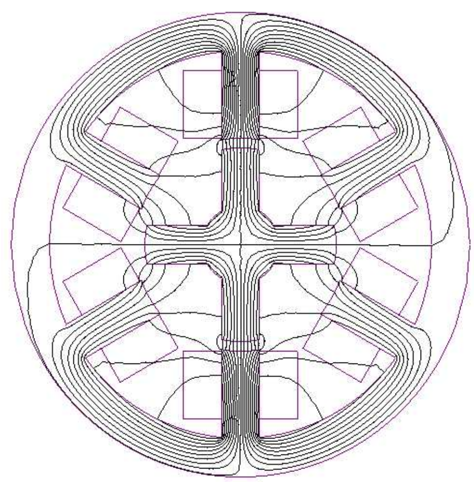
3 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

- The flux plot should resemble the diagram below on the left.
- The diagram on the right shows the effect of connecting a coil the wrong way round.
- If the coil connections are wrong, open the Coil page, right-click on a coil side, and click Reverse Coil Side Direction. Repeat for the other coil side.

4 Examine the effect of switching the coil currents by setting the current in Coil#1 to 0 and the current in Coil#2 to 2.0 A.



Correct coil connections



Incorrect coil connections

Airgap field components

It is useful to display the radial component of flux density in the airgap, which can be done with the Field Extractor extension as follows. First, it is necessary to draw the contour as a circle in the middle of the airgap:

1 On the View menu, click Solid Model.

2 Draw a circle of radius 52.5 mm.

- Use the Keyboard Input bar, or set the Construction Grid spacing to 2.5 mm.

3 Select the upper half of the circle.

4 On the Extensions menu, click Field Extractor to display a new window.

5 Check the *Connected edges* box to include the lower half of the circle.

6 In the Add drop-down list, select Edge.

This uses the circle drawn on the Construction Slice to define two arcs in the Variables box as follows:

Arc	52.5,0, 180, 0,0	100
Arc	-52.5,0, 180, 0,0	100

7 In the Field drop-down list, select |B contour normal|.

8 Click Graph.

This should display a graph of the radial component of B in the airgap; the value is small except in the regions of the energized poles.

9 Close the Field Extractor window.

Variation of torque with rotor displacement

The variation of torque with the angular position of the rotor can be determined by parameterization.

1 In the Parameters page of the model Properties, define a new numeric parameter **Angle** as a number with the list of values: **0, 5, 10, 15, 20, 25, 30**.

2 For the Rotor component, display the Parameters page of the Properties dialog and enter the following expression for the RotationAngle parameter:
%Angle%deg

The rotor will be rotated in the anticlockwise direction for successive values of the RotorAngle parameter in degrees.

3 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

4 In the Results window, open the Force page and examine the magnitude of the torque on the rotor for each problem.

- If MagNet is licensed for parameterization, display a graph of the magnitude of the torque on the rotor.

Animation

If MagNet is licensed for parameterization, it is instructive to display an animation of the flux plot as the rotor moves through successive angles.

1. Open the Field page of the Project bar.
2. In the Shaded page, select None.
3. In the Contour page, select Flux Function.
4. Click Animate.

   MagNet generates an image of the plot for each problem ID, and then displays them in sequence in a new Animation window in the View area.
5. In the Animation window, use the buttons at the bottom of the display to control the speed etc.
6. Close the Animation window.
   - This will display a Save Changes dialog. Click No.

Sample results

The values below were obtained for the magnitude of the torque on the rotor, with MagNet version 7.5. The rotor angle represents the displacement from a position of alignment with the stator poles.

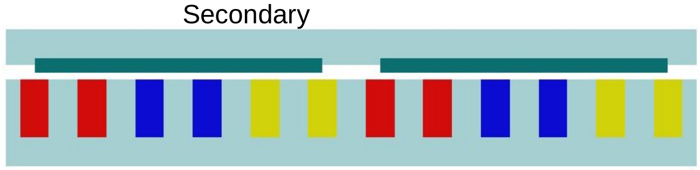
Rotor angle (º)	**Torque magnitude (Nm)**
0	0.012
5	2.112
10	3.142
15	3.621
20	3.706
25	2.845
30	1.714

Discussion

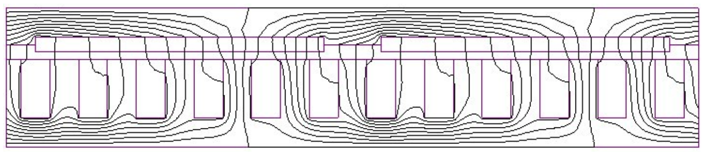
The dimensions of this simple stepper motor have been chosen to make the model easy to build, with no attempt at an optimal design. In particular, the radial depth of the core is excessive, with a correspondingly low flux density. The stator and rotor poles are narrow, so the restoring torque decreases rapidly when the rotor displacement angle exceeds 20º. Consequently, the torque developed at each switching transition is quite low.

Linear synchronous motor

The diagram below shows a 2D model of part of a simple linear synchronous motor with surface-mounted permanent magnets.



The primary has a full-pitched 3-phase winding in a laminated steel core, with 6 slots per pole, and the secondary comprises blocks of permanent-magnet material partially embedded in a steel backing. Alternating currents in the primary winding will produce a traveling magnetic field, and at one instant of time the flux plot is shown below for a load angle of 90º.



This model represents two pole pitches of an infinitely long machine. Since the field pattern repeats every two pole pitches, the field values will be identical at corresponding points on the two ends. To represent this condition in MagNet, a *periodic boundary condition* is specified for the two ends. Because of the symmetry of this device, it is possible to model just one pole pitch, since the field values repeat with opposite signs at intervals of a pole pitch.

If a field has identical values at corresponding points, an *even periodic boundary condition* is used; if it has equal and opposite values, an *odd periodic boundary condition* is used.

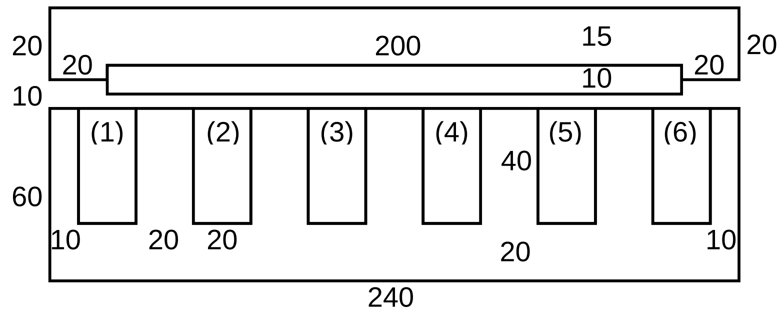
The case study will consider two ways of simulating the machine operation:

- For a fixed position of the secondary, the phase angle of the currents will be varied. This will show how the tractive force varies with load angle, for one position of the secondary.
- Both the phase of the currents and the position of the secondary will be varied, so as to hold the load angle constant. This will show the cogging effect that will be present during normal operation.

In this machine, getting accurate results requires control of the mesh density in the airgap. This is best accomplished by defining two air regions for the airgap, and specifying the maximum size for elements in these regions. Automatic mesh refinement by adaption is not used in this model.

Modeling the device

The diagram below shows one pole pitch of the device, with dimensions in millimeters. Numbers in brackets are the primary winding slot numbers. The coils have 100 turns per slot, the peak value of the current is 11 A, and the current phase angle range is –30º to –165º, corresponding to a load-angle range of 0 to –135º.



To speed up the solution, linear materials will be used for the model: MU3 for the primary and secondary cores, and PM04 for the permanent magnet. These are reasonably representative of ordinary steels and ceramic ferrite, respectively.

For the coils, new materials will be defined with the same properties as copper, but with distinctive red, yellow and blue colors to identify the phases. The color sequence for the slots will be red, blue, yellow, where the blue slots will carry reversed current from the blue phase. This gives the normal 60º phase progression between coil groups for an AC machine winding.

The model for this machine will use separate air boxes for the primary and the secondary, with a common surface in the middle of the airgap. A separate air box enables the secondary to be moved relative to the primary.

Constructing the model

1. Start a new model and save it as **LSM 1**.
2. Set the model length units to millimeters.
3. The coordinate origin is the top left-hand corner of the primary core. Set the display to show a corresponding range of *x* and *y* values with a construction grid, using a grid spacing of 10 mm in *x* and 5 mm in *y*.
4. Construct components for the primary core and the secondary core:
   - Material: MU3: Relative permeability 1000.
   - Sweep distance: 100 mm.
5. Construct a component for the permanent magnet:
   - Material: PM04: Br 0.4 mur 1.0
   - Type: Uniform.
   - Direction: (0, 1, 0).
   - Sweep distance: 100 mm.

6 Define a new red conductor material as follows.

- In the Project bar, select the Material page.
- Right-click, and select New User Material.
- In the Defaults Based On drop-down list, select Copper: 5.77e7 Siemens/meter from the Commonly Used Materials section.
- In the Name box, enter **Copper red**.
- In the Display Color drop-down list, select red.
- Click Finish.

Copper red should be shown as a new User Material.

7 Similarly, define new materials Copper yellow and Copper blue.

8 Construct the first two coil sides in slots 1 and 2:

- Names: Slot#1, Slot#2.
- Material: Copper red.
- Sweep distance: 100 mm.

9 Similarly, construct the remaining coil sides:

- Slots 3 and 4: Copper blue.
- Slots 5 and 6: Copper yellow.

Coils and currents

Coils will be constructed from pairs of coil sides. It is necessary to select individual faces of the coil sides, to ensure that the current directions are correct.

1 Make the first coil as follows:

- Display the tree directories for Slot#1 and Slot#2.
- Click Face#1 (Start Face) for Slot#1.
- Hold down the Ctrl key, and click Face#1 (Start Face) for Slot#2.
- On the Model menu, click Make Simple Coil.
- Change the coil name to RedPhase.
- Set the number of turns to 100. Ignore the current.

2 In the same way, make the YellowPhase using the coil sides in Slot 5 and Slot 6.

3 For the BluePhase, follow a similar procedure, but select Face#2 (End Face) for the coil sides in Slot#3 and Slot#4. This reverses the direction for the BluePhase.

4 In the Parameters page of the model Properties, create two user-defined parameters:

- **Magnitude** as a number with the value **11**.
- **Phase** as a number with a list of values:
  –30, –15, 0, 15, 30, 45, 60, 75, 90, 105.

5 In the Parameters page of the Properties for each of the three phase coils, enter the following Current expressions:

- RedPhase: %Magnitude*cos(%Phase%deg).
- YellowPhase: %Magnitude*cos((%Phase–120)%deg).
- BluePhase: %Magnitude*cos((%Phase+120)%deg).

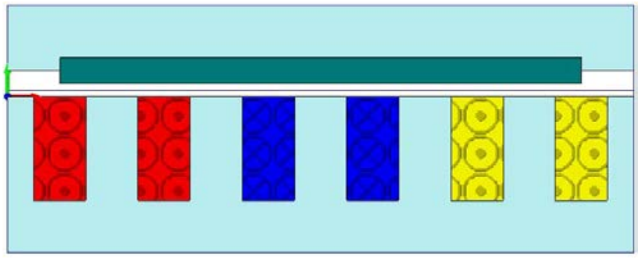
  %deg converts degrees to radians.

Air regions and boundary conditions.

1 Delete all the construction slice lines.

2 Draw a rectangle that just encloses the model.

3 Draw a line through the middle of the airgap, as shown below, dividing the rectangle into two parts.

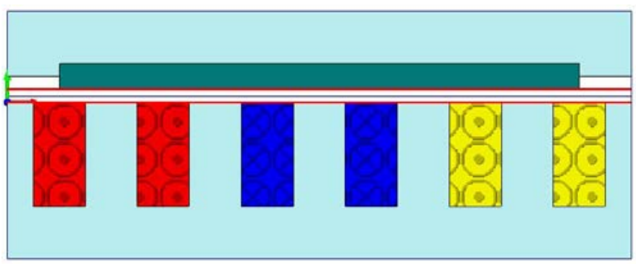


4 Construct an air box for the primary from the lower rectangle:

- Name: PrimaryAirBox
- Material: AIR.
- Sweep distance: 100 mm.

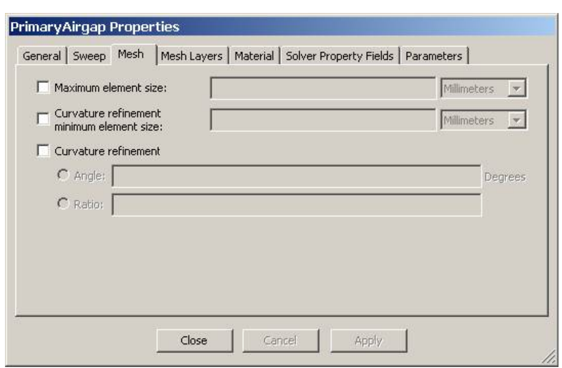
5 Similarly, construct an air box for the secondary from the upper rectangle.

6 Draw two more lines in the airgap, across the whole width of the model, touching the primary and secondary surfaces as shown below.



7 Construct air regions from the resulting narrow rectangles, named PrimaryAirgap and SecondaryAirgap.

8 Display the Properties dialog for the PrimaryAirgap, and select the Mesh page:



9 Click Maximum Element Size, and in the text box enter **2.5**.

10 Similarly, set the maximum element size for the SecondaryAirgap to 2.5 mm.

11 Apply an Odd Periodic boundary condition to the model as follows.

- Display the tree directories for PrimaryAirBox and the SecondaryAirBox.
- Hold down the Ctrl key, and click Face#6 for each of these components, to select both of the left-hand end faces.
- On the Boundary menu, click Odd Periodic.
- In the dialog, Click Set Transformation.
- In the dialog, Click Shift Vector, and enter **(240, 0, 0).**

  This represents the relative position of the other ends of the components.
- Click OK to close each dialog.

12 Check the boundary condition as follows.

- In the Object page, there should be an item BoundaryCondition#1(OP).
- Rotate the model so that the left-hand end is visible.
- Check that the faces of both air boxes are marked with the + and – symbols of the odd periodic boundary condition.
- Check the right-hand end in the same way.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 2.
- Adaption Options:
  not used.

2 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

3 Inspect the flux plot and the shaded plot of |B| for each problem.

4 In the Results window, open the Force page and examine the force components on the two bodies for each problem.

- If MagNet is licensed for parameterization, display a graph of the X component of force on the secondary.

Sample results

The following values were obtained for the X component of force on the secondary, with MagNet version 7.5:

Phase (º)	**Load angle (º)**	**Force (N)**
–30	0	0.0
–15	15	17.96
0	30	34.99
15	45	50.02
30	60	62.18
45	75	70.49
60	90	74.27
75	105	73.01
90	120	66.48
105	135	55.01

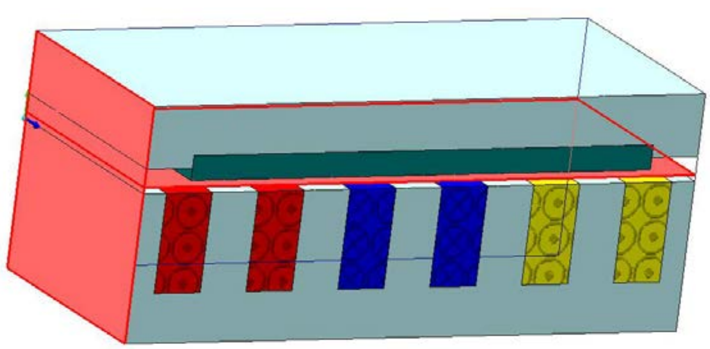
Moving the secondary

To examine cogging effects, it is necessary to move the secondary relative to the primary. One new parameter is required, which will control the shift of all the secondary components and change the phase of the primary current. Motion of the secondary is simulated by giving successive values to this shift parameter. To cope with the displacement of the secondary, the periodic boundary condition must be applied to the top surface of the primary air box as well as the ends of the two air boxes, as described below.

Boundary conditions

1. Open the model LSM 1 and save it as **LSM 2.**
2. Delete the periodic boundary condition.
3. Display the tree directories for the PrimaryAirBox and the SecondaryAirBox.
4. Hold down the Ctrl key and click the following faces:
   - PrimaryAirBox Face#5
   - PrimaryAirBox Face#6
   - SecondaryAirBox Face#6.

   The model view should resemble the following:



5 Apply an Odd Periodic boundary condition as follows.

- On the Boundary menu, click Odd Periodic.
- In the dialog, Click Set Transformation.
- In the dialog, Click Shift Vector, and enter **(240, 0, 0).**
- Click OK to close each dialog.

6 View the whole of the model to see the boundary condition.

Observe that the top surface of the Primary air box has been extended by 240 mm in the positive x direction. It is now possible to move the secondary relative to the primary, and the extended surface will apply the correct constraints to the field values.

Parameterization

1 In the Parameters page of the model Properties, change the list of values for the Phase parameter to the single value **30**.

This will give a load angle of 60º.

2 Define a new numeric parameter **Shift** with the list of values: **0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90**.

3 For each Secondary component (core, magnet, airgap and air box), open the Parameters page and enter the following expression for the Shift Vector: **[%Shift%mm, 0, 0]**

All of the secondary components will be moved in the positive x direction for successive values of the Shift parameter.

4 Change the coil parameters as follows, to introduce a phase shift corresponding to the movement of the secondary:

- Red phase: %Magnitude*cos((%Phase+0.75*%Shift)%deg).
- Yellow phase: %Magnitude*cos((%Phase+0.75*%Shift –120)%deg).
- Blue phase: %Magnitude*cos((%Phase+0.75*%Shift +120)%deg).

Solving and post-processing

Solve the model and inspect the results as before. If MagNet is licensed for parameterization, it is instructive to create an animation of the field plots, which shows the travelling magnetic field. A graph of the X component of force on the secondary shows the cogging effect as the secondary magnets move past the primary slot openings.

Sample results

The following values were obtained for the X component of force on the secondary, with MagNet version 7.5:

Shift (mm)	**Force (N)**
0	62.18
5	58.61
10	67.05
15	77.64
20	83.13
25	86.23
30	84.45
35	73.54
40	61.96
45	58.67
50	66.94
55	78.13
60	84.23
65	87.50
70	85.16
75	73.49
80	61.85
85	58.23
90	66.36

Discussion

For the simple linear synchronous motor used in this case study, the technique of shifting the secondary shows clearly the cogging effect of the primary slot openings on the force characteristic.

To simplify the modeling, a full-pitched winding has been used, so that each slot carries current from only one phase. However, it is a straightforward matter to define upper and lower coil sides in each slot, and it is then possible to model two-layer chorded windings.

The methods of using periodic boundary conditions can be applied to other devices such as rotating electrical machines, where it is only necessary to model one pole pitch of the physical device. In spite of their shape, rotating machines must also be modeled using translational geometry.

Chapter 4

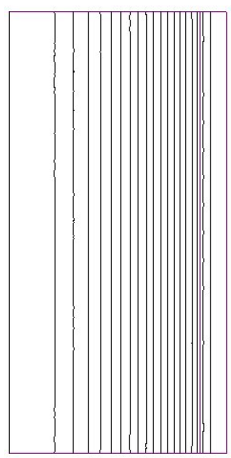
Case Studies: Rotational Geometry

Introduction

This chapter follows the same pattern as Chapter 3; it covers a variety of modeling problems for devices with rotational geometry, arranged in order of increasing difficulty.

Constructing a model with rotational geometry is different from the procedures used in chapters 2 and 3. The model is constructed as part of a solid of revolution by rotating shapes about an axis. This axis must be the Y axis of the normal XY drawing plane. Components are formed by sweeping in an arc instead of sweeping in a line. The subtended angle of the arc is unimportant for a 2D model, so the MagNet default angle of 90º will be used for all of the case studies. This construction technique is described in the first case study: Inductance of a Brooks coil.

Flux plots are not as easy to interpret with rotational geometry as they are with translational geometry. Consider the case of an infinitely long straight solenoid. The magnetic field is confined to the interior of the solenoid, where it is perfectly uniform and parallel to the axis. This device can be modeled in MagNet by setting boundary conditions – see the case study on a cylindrical screen in a uniform field on page 57. The diagram below shows the flux plot generated by MagNet, with the axis of the solenoid on the left. This diagram represents the flux in the right-hand half of a cross-section of the device.



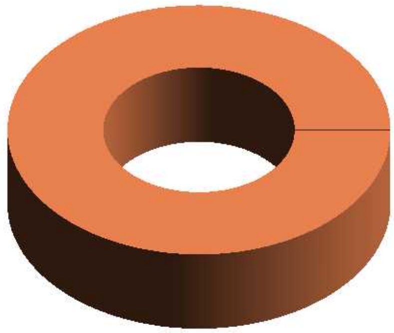
The spacing between the flux lines decreases with increasing distance from the axis. For users who are familiar with ordinary 2D flux plots for translational geometry in the XY plane, the plot gives the impression that the flux density increases with distance from the axis. This is an illusion. If Δr is the radial distance between two successive flux lines, and B is the axial flux density, then the increment of flux is:

$$\Delta\phi = B\Delta S = B \cdot 2\pi r \Delta r \qquad (4\text{-}1)$$

There are equal increments of flux between the lines, so $\Delta\phi$ is constant. If B is constant, it follows that Δr must be inversely proportional to r.

Self-inductance of a Brooks coil

The diagram below shows a Brooks coil. It is a short solenoidal coil with a square cross-section, having an inner radius equal to the length of one side of the square. This shape is generated by rotating a square about a vertical axis, as shown in the diagram on the next page.



Brooks coils have a simple shape, and are close to the optimum of a coil that has the largest inductance for wire of a given length and cross-sectional area. The inductance can be calculated analytically, so this is another useful test problem for checking the accuracy of the results produced by MagNet. From the formula in Grover [5] for solenoidal coils, the inductance is:

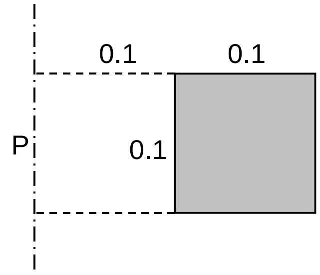
$$L = 1.6994 \times 10^{-6} rN^2 \quad \text{[H]} \tag{4-2}$$

where r is the mean radius of the coil in meters and N is the number of turns. The coil to be analyzed has the following parameters: inner radius = 0.1 m, outer radius = 0.2 m, height = 0.1 m, and N = 100. The mean radius is r = 0.15 m, so the inductance from equation 4-2 is 2.549 mH.

A value for the coil current must be specified before the problem can be solved in MagNet, although a value is not required in the description of the problem. There are no magnetic materials present in this device, so it is a linear problem in which the flux is directly proportional to the current. Since the inductance is the flux linkage per unit current, it must be independent of the current, so any reasonable value of current may be used; a value of 1 A is chosen arbitrarily.

Modeling the device

The diagram below shows the cross-section in the XY plane that is rotated about the Y axis to form the coil, with dimensions in meters. This is an open boundary problem, but the Kelvin transformation method cannot be used with rotational geometry, so the coil must be enclosed in an air box that is large enough to give an accurate inductance value. A suitable size is an axial length of 2 m and a radius of 2 m – see the “Discussion” on page 88.



Creating the model

1 Start a new model and save it as **Brooks coil**.

2 Set the model length units to meters.

3 The coordinate origin is the center of the coil, which is the point P on the vertical axis in the diagram above. Set the display to show a corresponding range of *x* and *y* values with a construction grid.

4 Draw the cross-section of the coil component as a square of side 0.1 m.

5 Make the component as follows.

- Click the Select Construction Slice Surfaces button. 
- Click anywhere inside the square.
- Click the Make Component in an Arc button. 
- Change the component name.
- Select the material: Copper: 5.77e7 Siemens/meter.
- Leave the default numerical values unchanged:
  Angle 90º, Center at (0, 0) Axis vector (0, –1).
- Click OK.

6 Make a coil from this component.

- Number of turns: 100
- Current: 1.0 A.

7 Select and delete the construction slice lines.

8 Draw the cross-section of an air box with vertices at (0, –1), (2, –1), (2, 1), (0, 1) and sweep it in an arc:

- Material: AIR.

Solving and post-processing

For models with rotational symmetry, the MagNet solver uses polynomial order 2 by default. There will be no need to change the polynomial order from that default value.

The suggested solver settings should give an accurate 2D solution without excessive computing time. The user is invited to try the effect of different settings.

1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.01%.
2. Solve as Static 2D.
3. Inspect the contour plot of the flux function and the shaded plot of the |B| values.
4. Inspect the computed global quantities, and calculate self-inductance values as follows (see appendix B for details):
   - From the flux linkage: $L = \lambda / i$, where λ is the flux linkage for the coil, and i is the coil current.
   - From the stored energy: $L = 2W / i^2$, where W is the stored magnetic energy.
5. As an alternative, use the RLC Matrix Calculator (see page 67) to determine the self-inductance of the coil.

Sample results

The results below were obtained with MagNet version 7.5:

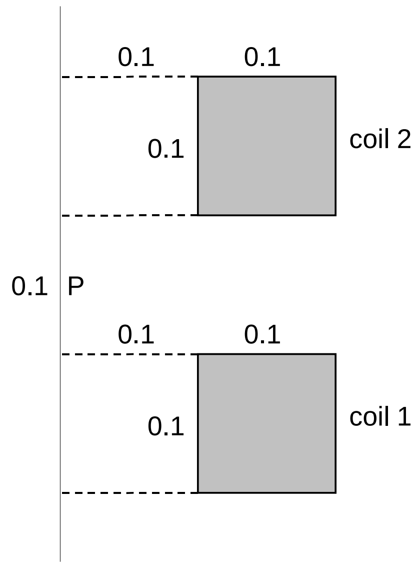
Stored magnetic energy:	0.001273 J
Co-energy:	0.001273 J
Flux linkage:	0.002546 Wb
Inductance from flux linkage:	2.546 mH
Inductance from stored energy:	2.546 mH

Discussion

The computed value of the self-inductance is only 0.1% lower than the theoretical value, which indicates that the surfaces of the air box have been taken sufficiently far away from the coil. In this case the radius of the box is 10 times the outer radius of the coil, and the axial length is 20 times the axial length of the coil. The contour plot of the flux function shows no apparent “squashing” of the flux pattern near the box surfaces. These ratios of air box dimensions to coil dimensions may be taken as guides for other problems where the inductance of an air-cored coil is to be computed but there is no known analytical solution.

Mutual inductance of coaxial coils

The diagram below shows the cross-section of a pair of Brooks coils with a common axis. This view is rotated about the vertical axis to generate the coil shape. Some of the flux produced by one coil will link the other coil, so the coils are magnetically coupled; a mutual inductance M may be defined in the same way as for the transformer on page 63. As with the self-inductance of a single Brooks coil, the mutual inductance can be calculated analytically.



The coils have the same dimensions as for the single Brooks coil (page 86), and the number of turns is $N = 100$. The spacing is 0.1 m, which is equal to the height of one coil. From Grover [4], the mutual inductance of this configuration is

$$L = 3.2736 \times 10^{-7} rN^2 \quad [\mathrm{H}] \tag{4-3}$$

Thus, the theoretical value of the mutual inductance is 0.4910 mH.

There are two methods of determining the mutual inductance with MagNet. The first method is to use the relationships based on the flux linkage with one coil when current flows in the other coil. For this linear system, appendix B gives:

$$M = \frac{\lambda_{21}}{i_1} = \frac{\lambda_{12}}{i_2} \tag{4-4}$$

where λ_{21} is the flux linkage with coil 2 when coil 1 carries a current i_1, and λ_{12} is the flux linkage with coil 1 when coil 2 carries a current i_2.

The second method of determining mutual inductance is based on the stored energy. Consider two cases: (a) equal currents, so that $i_1 = i_2 = i$; (b) equal and opposite currents, so that $i_1 = -i_2 = i$. From appendix B, the mutual inductance is given by

$$M = \frac{W_a - W_b}{2i^2}, \tag{4-5}$$

where W_a and W_b are the stored energy values in the two cases.

Modeling the device

This device is modeled in a similar way to the single Brooks coil on page 86.

Creating the model

1. Start a new model and save it as **Two Brooks coils**.
2. Set the model length units to meters.
3. The coordinate origin is the center of the coil pair, which is the point P on the vertical axis in the diagram on page 89. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Draw the cross-section of each coil component as a square of side 0.1 m, and sweep it in an arc about the Y axis:
   - Material: Copper: 5.77e7 Siemens/meter.
5. Make two coils:
   - Number of turns in each coil: 100.
   - Current in Coil#1: 1.0 A
   - Current in Coil#2: 0.
6. Select and delete the construction slice lines.
7. Draw the cross-section of an air box with vertices at (0, –1), (2, –1), (2, 1), (0, 1) and sweep it in an arc:
   - Material: AIR.

Solving and post-processing

1. Set the options for solving, using the Solve menu, with the following changes from the default values:
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.01%.
2. Solve as Static 2D.
3. Inspect the contour plot of the flux function and the shaded plot of the |B| values.
4. Inspect the computed global quantities and calculate inductances as follows (see appendix B for details):
   - Self-inductance: $L_1 = \lambda_{11} / i_1$, where λ_{11} is the flux linkage with coil 1, and i_1 is the coil current.
   - Mutual inductance: $M = \lambda_{21} / i_1$, where λ_{21} is the flux linkage with coil 2.

5 Change the current in coil 1 to 0 A and the current in coil 2 to 1.0 A. Solve again and calculate the inductances:

- Self-inductance: $L_2 = \lambda_{22} / i_2$, where λ_{22} is the flux linkage with coil 2, and I_2 is the coil current.
- Mutual inductance: $M = \lambda_{12} / i_2$, where λ_{12} is the flux linkage with coil 1.

6 Change the current in coil 1 to 1.0 A. Solve again and record the value of the stored energy; this is W_a.

7 Change the current in coil 2 to –1.0 A. Solve again and record the value of the stored energy; this is W_b.

8 Calculate the mutual inductance from equation 4-5.

9 As an alternative, use the RLC Matrix Calculator (see page 67) to determine the self-inductances and the mutual inductance of the coils.

Sample results

The results below were obtained with MagNet version 7.5:

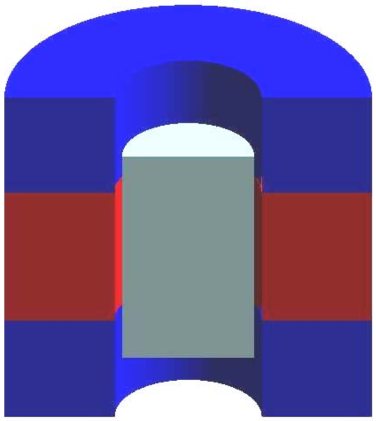
$i_1 = 1.0$ A, $i_2 = 0$	
Stored magnetic energy:	0.001273 J
Flux linkage λ_{11}:	0.002546 Wb
Flux linkage λ_{21}:	0.0004895 Wb
Self-inductance L_1:	2.546 mH
Mutual inductance M:	0.4895 H
$i_1 = 0$, $i_2 = 1.0$ A	
Stored magnetic energy:	0.001273 J
Flux linkage λ_{22}:	0.002546 Wb
Flux linkage λ_{12}:	0.0004895 Wb
Self-inductance L_2:	2.546 mH
Mutual inductance M:	0.4895 mH
$i_1 = 1.0$ A, $i_2 = 1.0$ A	
Stored magnetic energy:	0.00303554 J
$i_1 = 1.0$ A, $i_2 = –1.0$ A	
Stored magnetic energy:	0.00205666 J
Mutual inductance M:	0.4894 mH

Discussion

The computed value of the self-inductance is the same as for a single Brooks coil (see page 86), which is about 0.1% lower than the theoretical value. All three methods of calculation give the similar values for the mutual inductance, which is about 0.3% lower than the theoretical value. These results indicate that the air box is large enough to give an accurate result, and that the solver options are correctly chosen.

LVDT displacement transducer

The linear variable differential transformer (LVDT) is a device for measuring small displacements – typically a few millimeters. It comprises three fixed coils and a moving iron slug as shown in the sectional view below. The central coil is energized with alternating current; the two outer coils are connected in series opposition and act as sensing coils.

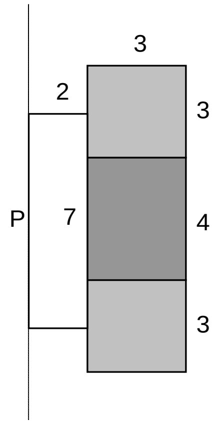


When the slug is centrally positioned, as shown above, the voltages induced in the sensing coils are equal, so the output from the transducer is zero. If the slug is displaced from this position, the voltage in one sensing coil increases while the voltage in the other decreases; the differential voltage will vary from zero in an almost linear manner. The designer has the problem of ensuring that the voltage/displacement characteristic is a linear one.

Although the device uses alternating current, it can be modeled as a static problem because it does not depend on eddy currents in the slug. The voltage depends on the flux linkage with the sensing coils, which can be determined from a static field solution.

Modeling the device

The diagram below shows the cross-section of the transducer in the XY plane, with the iron slug in its mid position. In practice there is a small clearance between the slug and the coils, but this is ignored in the model. It is required to find the differential flux linkage with the sensing coils in successive positions of the slug, with displacement increments of 0.5 mm. This will be done automatically by defining a set of values for a displacement parameter and using this parameter in a shift vector for the slug component.



The slug has a radius of 2 mm and a length of 7 mm; the dimensions of the coils are shown in millimeters in the diagram, where the inner radius is 2 mm. The central coil has 100 turns and a current of 0.1 A; the outer coils each have 200 turns, with no current.

Creating the model

1. Start a new model and save it as **LVDT**.
2. Set the model length units to millimeters.
3. The coordinate origin is the center of the coil system, which is the point P on the vertical axis in the diagram on page 92. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Draw the cross-section of each coil component, and sweep it in an arc about the Y axis:
   - Material: Copper: 5.77e7 Siemens/meter.
5. Make a coil from the middle component:
   - Number of turns: 100.
   - Current: 0.1 A
6. Make a single coil from the two outer components by selecting the start face of one component and the end face of the other component.
   - Number of turns: 200.
   - Current: 0 A
7. Draw the cross-section of the slug and sweep it in an arc:
   - Material: MU3: Relative permeability 1000.
8. Select and delete the construction slice lines.
9. Draw the cross-section of an air box with vertices at (0, –30), (60, –30), (60, 30), (0, 30) and sweep it in an arc:
   - Material: AIR.

Parameterization

1. In the Parameters page of the model Properties, define a new numeric parameter **Shift** with the following values **–1.5, –1.0, –0.5, 0, 0.5, 1.0, 1.5**.
2. In the slug Component Parameters page, specify the shift vector as **[0, %Shift%mm, 0]**.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

3 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

4 Inspect the computed global quantities.

If MagNet is licensed for parameterization, display a graph of the differential flux linkage against displacement. The differential flux linkage is the flux linkage with the complete sensing coil, which is the difference between the flux linkages of the individual coil components.

Sample results

The results below were obtained with MagNet version 7.5:

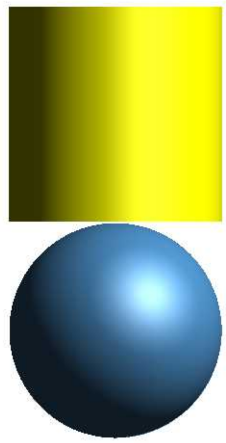
Displacement (mm)	Differential flux linkage (μWb)
–1.5	–6.497
–1.0	–4.731
–0.5	–2.482
0.0	0.0
+0.5	+2.482
+1.0	+4.730
+1.5	+6.498

Discussion

From the symmetry of the device, the differential flux linkage should be an odd function of the displacement, which is confirmed by the results. There is noticeable non-linearity as the displacement approaches the extremes of ±1.5 mm.

Magnetic pull-off force

The diagram below shows a steel sphere attracted to a cylindrical bar magnet. The problem is to calculate the force required to pull the sphere away from the magnet, and to estimate the accuracy of the force calculation.



Modeling the device

MagNet calculates forces on bodies automatically, but there must be an air space around the body. Therefore, an artificial gap must be introduced between the sphere and the magnet. Accurate calculation is difficult if the gap is very small. Also, the user needs to know how small to make the gap and the likely error that results from it. One solution is to solve for a number of different gap lengths and extrapolate the force values to obtain the pull-off force. Parameterization is a convenient way of tackling this problem.

In this case study, the sphere has a diameter of 20 mm; the magnet has a diameter of 20 mm and a length of 20 mm. Initially, the model will have the sphere in contact with the magnet. A controlled gap will be introduced by defining a set of values for a displacement parameter and using this parameter in a shift vector for the sphere.

Creating the model

1. Start a new model and save it as **Magnet and sphere**.
2. Set the model length units to millimeters.
3. The coordinate origin is the center of the sphere. Set the display to show a corresponding range of *x* and *y* values with a construction grid.
4. Draw the cross-section of the sphere as a closed semicircle of radius 10 mm and sweep it in an arc:
    - Material: CR10: Cold rolled 1010 steel.
5. Draw the cross-section of the magnet as a rectangle of width 10 mm and height 20 mm, and sweep it in an arc:

- Material: Ceramic ferrite.
    - Type: Uniform.
    - Direction: (0, 1, 0).

6. Select and delete the construction slice lines and arcs.
7. Draw the cross-section of an air box as a closed semicircle of radius 160 mm and sweep it in an arc:
    - Material: AIR.

Parameterization

1. In the Parameters page of the model Properties, define a new numeric parameter **Shift** with the following values: **0.1, 0.2, 0.3, 0.4, 0.5**.
2. In the sphere Component Parameters page, specify the shift vector as **[0, –%Shift%mm, 0]**.

    Note the negative sign for the y *shift. If this is omitted, the sphere will intersect the magnet, and the solution will fail.*

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 4.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.01%.

2 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

3 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

4 Inspect the computed global quantities. Take the force of attraction to be the mean of the magnitudes of the Y components of force on the sphere and the magnet.

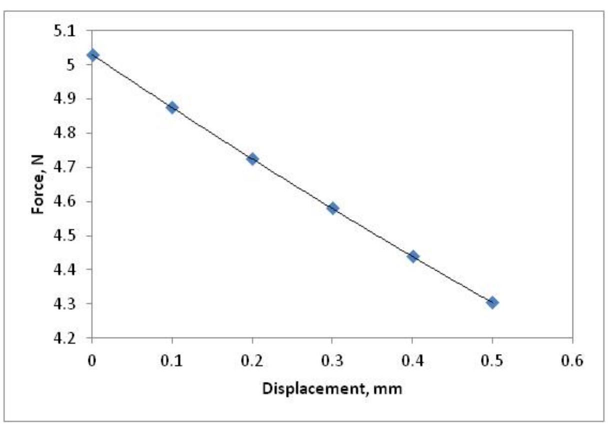
5 Repeat the study with polynomial orders 2 and 3 for the same model, and with polynomial order 4 for new models with air box radii of 80 mm and 320 mm.

Sample results

The results below were obtained with MagNet version 7.5 for an air box radius of 160 mm and solver polynomial order 4:

Displacement (mm)	Axial force on sphere (N)	Axial force on magnet (N)	Mean force (N)
0.1	4.8740	4.8744	4.8742
0.2	4.7242	4.7246	4.7244
0.3	4.5792	4.5797	4.5795
0.4	4.4385	4.4402	4.4394
0.5	4.3032	4.3039	4.3036

Fitting a smooth curve to the mean force values and extrapolating to zero displacement gives a value of 5.03 N for the pull-off force. The graph below was generated with Microsoft Excel, using a second-order polynomial trendline, and the first point adjusted to lie on the curve.



Repeating the solution with different polynomial orders, and with different sized boundaries, gave the following results for the force on the sphere:

Axial force (N) with a boundary radius of 160 mm:

Displacement (mm)	Polynomial order 2	Polynomial order 3	Polynomial order 4
0.1	4.8781	4.8746	4.8742
0.2	4.7272	4.7246	4.7244
0.3	4.5817	4.5795	4.5795
0.4	4.4436	4.4414	4.4394
0.5	4.3049	4.3025	4.3036

Axial force (N) with polynomial order 4:

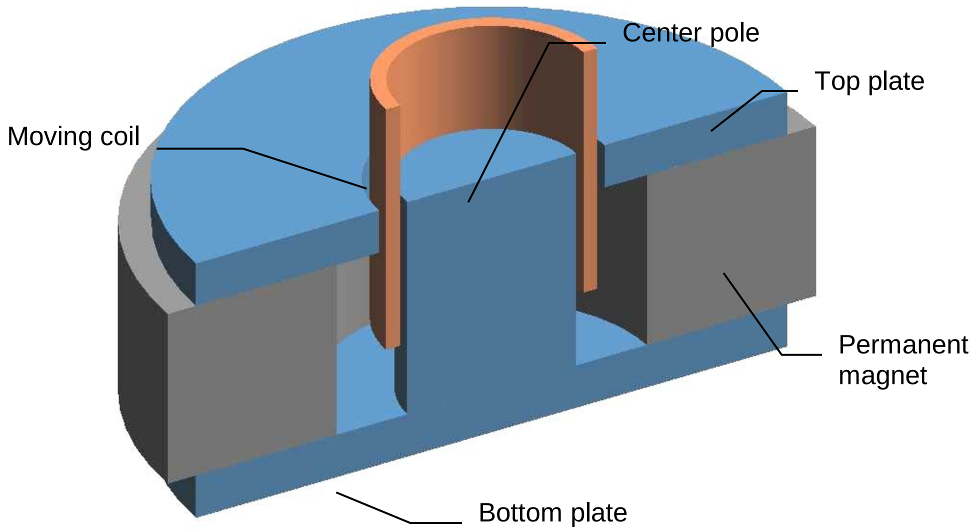
Displacement (mm)	Boundary radius 80 mm	Boundary radius 160 mm	Boundary radius 320 mm
0.1	4.8261	4.8742	4.8807
0.2	4.6767	4.7244	4.7309
0.3	4.5335	4.5795	4.5858
0.4	4.3930	4.4394	4.4453
0.5	4.2597	4.3036	4.3100

Discussion

The reference model, with an air box radius of 160 mm and polynomial order 4, gives almost identical results for the magnitudes of axial forces on the magnet and the sphere. At a displacement of 0.3 mm, reducing the boundary radius to 80 mm changes the force by 1%, but increasing the radius to 320 mm changes the force by only 0.14%. The polynomial order has very little effect on the results when adaption is used with a small tolerance of 0.01%. It appears that the computed results for a radius of 320 mm are close to convergence on the true values, and the numerical error is unlikely to exceed 1%. Since the force depends on the square of the flux density, a change of only 0.5% in the remanence of the magnetic material would give a force change of 1%. The accuracy of the computed force value is therefore limited by the variability of material properties rather than numerical errors.

Moving-coil transducer

The diagram below shows a half section of a 3D model of a moving-coil transducer. It comprises a permanent magnet that creates a radial magnetic field in the airgap between the center pole and the top plate, and a solenoidal coil that is free to move in the axial direction. When current is passed through the moving coil, an axial force is developed. This structure is found in loudspeakers, headphones, vibration generators, moving-coil actuators and chemical balances.

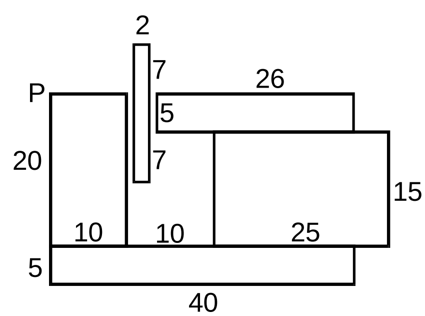


It is required to find the coil inductance and explore the linearity of the force/current relationship under the following conditions:

- variable coil current, with the coil in its mid position;
- variable coil position, with the current at its rated value.

Modeling the device

The diagram below shows the shape that must be rotated about the vertical axis to form the device, with dimensions in millimeters. The moving coil is 2 mm thick and 19 mm deep; it is positioned in the center of the airgap, which has a radial length of 4 mm. The material for the center pole, top plate and bottom plate is 1010 cold-rolled steel, and the permanent-magnet material is ceramic ferrite. The moving coil has 300 turns and the rated current is 1 A.



To determine the inductance of the coil, it is necessary to allow for the contribution of the permanent magnet to the total flux linkage with the coil. If λ_i is the flux linkage when the coil carries a current i, and λ_0 is the flux linkage with no current, the coil inductance is given by

$$L = \frac{\lambda_i - \lambda_0}{i} \tag{4-6}$$

The force on the coil will be calculated for the following conditions:

- The coil current is varied in steps of 0.2 A with the coil in its mid position.
- The coil position is varied in steps of 1 mm with the current at its rated value.

Constructing the model

1. Start a new model and save it as **Moving coil**.
2. Set the model length units to millimeters.
3. The coordinate origin is the center of the top face of the center pole, which is the point P in the diagram on page 99. Set the display to show a corresponding range of x and y values with a construction grid.
4. Draw the cross-section of each steel component and sweep it in an arc:
   - Material: CR10: Cold rolled 1010 steel.
5. Draw the cross-section of the permanent-magnet component, and sweep it in an arc:
   - Material: Ceramic ferrite.
   - Type: Uniform.
   - Direction: (0, –1, 0).
6. Draw the cross-section of the coil component and sweep it in an arc:
   - Material: Copper: 5.77e7 Siemens/meter.
7. Make the coil:
   - Number of turns: 300
   - Current: 1 A.
8. Select and delete the construction slice lines.
9. Draw the cross-section of an air box as a closed semicircle of radius 300 mm, centered at the origin, and sweep it in an arc:
   - Material: AIR.

Parameterization – current

1. In the Coil Parameters page, enter the following list of values for the Coil Current parameter:
   –1.0, –0.8, –0.6, –0.4, –0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0

Solving and post-processing – current

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.5%.

2 Solve as Static 2D.

With the Trial Edition of MagNet, solutions of successive problems can be obtained by the method described in chapter 2 (page 31).

3 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

4 Inspect the computed global quantities for the different current values.

If MagNet is licensed for parameterization, display a graph of the Y component of force on the coil.

5 Determine the coil inductance for a current of –1 A from equation 4-5 on page 100.

- Note that the RLC Matrix Calculator extension to MagNet (see page 67) will not give the correct value for the inductance in this case.

Parameterization – displacement

1 In the Coil Parameters page, change the current parameter values from a list to the single number: **–1.0**

2 In the Parameters page of the model Properties, define a new numeric parameter **Shift** with the following values: **–5, –4, –3, –2, –1, 0, 1, 2, 3, 4, 5**

3 In the Coil Component Parameters page, specify the shift vector as **[0, %Shift%mm, 0]**

Solving and post-processing – displacement

1 Solve as Static 2D.

2 Inspect the contour plot of the flux function and the shaded plot of the |B| values.

3 Inspect the computed global quantities for the different displacement values.

If MagNet is licensed for parameterization, display a graph of the Y component of force on the coil.

Sample results

The results below were obtained with MagNet version 7.5:

Coil in mid position:		Coil current = –1.0 A	
Coil current (A)	Axial force on coil (N)	Displacement (mm)	Axial force on coil (N)
–1.0	–7.910	–5.0	–7.727
–0.8	–6.284	–4.0	–7.926
–0.6	–4.680	–3.0	–8.033
–0.4	–3.096	–2.0	–8.055
–0.2	–1.537	–1.0	–8.001
0.0	0.0	0.0	–7.910
0.2	1.512	1.0	–7.771
0.4	3.000	2.0	–7.558
0.6	4.463	3.0	–7.311
0.8	5.899	4.0	–7.006
1.0	7.309	5.0	–6.616

Coil in mid position, coil current = –1.0 A:

Flux linkage λ_t: –0.081578 Wb

Flux linkage λ_0: – 0.076166 Wb

Self-inductance L: 5.412 mH

Discussion

There is noticeable non-linearity in the force/current characteristic, which is not symmetrical for positive and negative current values. Its origin is the flux generated by the coil current itself, which aids or opposes the permanent-magnet flux in the airgap. The origin of the effect can be demonstrated by substituting a non-magnetic material such as aluminum for the ceramic ferrite material, since the recoil permeability of ceramic ferrite is close to 1. A current magnitude of 1.0 A in the coil then gives an axial force of –0.389 N, independent of the direction of the current. This force is in the negative Y direction, drawing the coil in to the magnet, and it is proportional to the square of the current.

The force/displacement characteristic is also non-linear, and the asymmetry in this characteristic reflects the asymmetry of the fringing field above and below the top plate of the magnet. Both of these non-linear effects are sources of distortion in moving-coil loudspeakers.

The inductance calculated from equation 4-6 on page 100 is a static value, which ignores eddy currents in the magnetic circuit, particularly the center pole, when the coil carries alternating current.

Chapter 5

Scripting

Introduction

Up to this point, MagNet has been used interactively, with the mouse and the keyboard, to build models and analyze the results. MagNet can also be controlled by *scripts* and *scripting forms*.

Scripts are text files containing commands that control MagNet. A script can be recorded during a MagNet session. When this script is run, all the operations that were carried out during the recording session will be repeated automatically. Scripts created in this way can be edited to change the operations, and scripts that are more powerful can be created with the VBScript programming language.

Recording a script is often an effective way of finding out how to use the MagNet scripting commands, in preparation for writing special-purpose user scripts.

Scripting forms take scripting a stage further by providing a graphical user interface for the user to interact with the script. A form can have text boxes for entering values, buttons for starting actions, and areas for displaying results. Scripting forms are not covered in this document, but sources of information are given below.

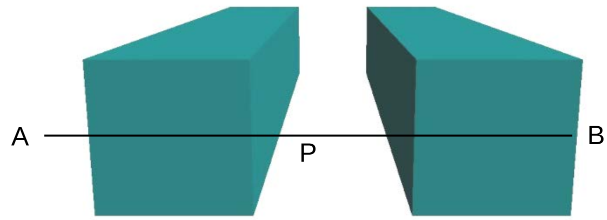
The most advanced kind of scripting uses another application to communicate with MagNet through the Microsoft ActiveX Automation Interface. Microsoft Excel, for example, can be used in this way. A script in the form of an Excel macro can command MagNet to build and solve a model, using data entered on the spreadsheet, and then get results back from MagNet to display on the spreadsheet. An example of this form of scripting is given in the section “Automation with Excel” on page 109.

Further information

The MagNet help gives full particulars of the MagNet scripting commands. Guidance on writing scripts is available from the Support area of the Infolytica website, www.infolytica.com.

Example model

To illustrate the principles of scripting, a simple model will be used. The diagram below shows two permanent-magnet blocks, magnetized in a direction parallel to the line AB. They will attract each other if they are magnetized in the same direction, but they will repel each other if the direction is reversed in one block. The blocks are deep enough for a 2D model to be used.



The cross-section of each block is a square of side 20 mm, the gap between the blocks is 20 mm, and the blocks are 100 mm deep. The material is ceramic ferrite.

This is an open boundary problem, which is best handled by the open boundary technique described in the busbar forces case study on page 48. However, for simplicity, a simple circular boundary will be used in this example.

Creating the model

1. Save the file as **PM blocks**.
2. Set the model length units to millimeters.
3. The coordinate origin is mid-way between the blocks, which is the point P in the diagram on the previous page. Set the display to show a corresponding range of X and Y values with a construction grid.
4. Construct a component for each block by sweeping in a line:
   - Material: Ceramic Ferrite.
   - Type: uniform.
   - Direction: (1, 0, 0).
   - Sweep distance: 100 mm.

   The material parameters specify that the block is uniformly magnetized, with the magnetization vector in the positive x direction.
5. Select and delete all the construction-slice lines.
6. Create a circular air box with a radius of 200 mm.
   - Material: AIR.
   - Sweep distance: 100 mm.

Solving and post-processing

1 Set the options for solving, using the Solve menu, with the following changes from the default values:

- Solver Options:
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

3 Inspect the contour plot of the flux function.

4 Inspect the forces on the blocks.

Sample results

The results below were obtained with MagNet version 7.5.

X component of force on left block:	9.376 N
X component of force on right block:	–9.402 N

Script for the model

Creating a User Script Log file

Recording a script is similar to recording a macro in applications like Microsoft Word or Excel. Once you have opened a User Script Log file, commands representing the operations carried out with the mouse and the keyboard will be recorded in the file. This continues until you pause or stop the User Script. The following steps will create a script file for building and solving the model.

1. Save the PM blocks model.
2. On the File menu, click New.
3. On the Scripting menu, select Start Recording User Script.

   A Save As dialog box is displayed.
4. Enter the file name as **PM blocks.vbs** and click Save.
5. Construct the model again, using the same procedure as before, but omitting the initial step of saving the model.
6. Set the solver and adaption options as before, and solve the model.
7. When the solution is complete, on the Scripting menu, click Stop Recording User Script.
   - In the message box “Do you want to edit the user script file … immediately?”, click No.

Running the script

1. On the File menu, click Close.
   - Do not save the current model.
2. On the Scripting menu, click Run Script.

   An Open dialog box is displayed.
3. Select the file PM blocks.vbs and click Open.

   The model should be re-created and solved. Check that everything is the same as before.

   An alternative way of running this script is to select it from the bottom of the list in the File menu.

Editing the script

The recorded script should have re-created the original model without any changes. It can be edited to change the model, for example to change the size of the blocks. Try the following.

1 In Windows, open the Notepad text editor.

2 On the Notepad File menu, click Open.

An Open dialog box is displayed.

3 In the File Name box, change the entry from *.txt to ***.vbs** and press Enter.

This enables script files to be displayed in the dialog box.

4 Navigate to the folder containing the script file PM blocks.vbs, and open the file.

This is the folder used for saving the MagNet model.

5 Save the file as **PM blocks modified.vbs**.

6 Examine the file contents. There should be some commands similar to the following:

```
Call getDocument().getView().newLine(-10, -10, -30, -10)
Call getDocument().getView().newLine(-30, -10, -30, 10)
Call getDocument().getView().newLine(-30, 10, -10, 10)
Call getDocument().getView().newLine(-10, 10, -10, -10)
```

…

…

```
Call getDocument().getView().newLine(10, -10, 10, 10)
Call getDocument().getView().newLine(10, 10, 30, 10)
Call getDocument().getView().newLine(30, 10, 30, -10)
Call getDocument().getView().newLine(30, -10, 10, -10)
```

These commands draw lines to form squares of side 20 mm. The order may be different, depending on the way the squares were drawn when the script was recorded.

7 Edit the lines so that all the Y coordinate values are doubled in value:

```
Call getDocument().getView().newLine(-10, -20, -30, -20)
Call getDocument().getView().newLine(-30, -20, -30, 20)
Call getDocument().getView().newLine(-30, 20, -10, 20)
Call getDocument().getView().newLine(-10, 20, -10, -20)
```

…

…

```
Call getDocument().getView().newLine(10, -20, 10, 20)
Call getDocument().getView().newLine(10, 20, 30, 20)
Call getDocument().getView().newLine(30, 20, 30, -20)
Call getDocument().getView().newLine(30, -20, 10, -20)
```

8 On the File menu, click Save.

9 Return to MagNet.

10 Start a new model.

11 Run this modified script. It should construct a different model.

12 Check the result.

Creating a new script

Some operations are possible with scripts, but are not possible when using MagNet interactively. For example, the Probe Field Values tool only works with the mouse – it is not possible to enter the coordinates with the keyboard. The script listed below will wait for the user to enter *x* and *y* coordinate values, and then display the flux density magnitude at the point. It does this repeatedly until the user clicks the No button.

Line **Script**

```
1  'Script to get values of |B| at points in a 2D field
2
3  Set Mesh = getDocument.getSolution.getMesh(1)
4  Set Field = getDocument.getSolution.getSystemField(Mesh,"|B|")
5  ReDim Value(0)
6  Do
7      X = InputBox("Enter the X co-ordinate:",,0)
8      Y = InputBox("Enter the Y co-ordinate:",,0)
9      Call Field.getFieldAtPoint (X, Y, 0, Value)
10     Response = MsgBox("The value of |B| is " & Value(0) & Chr(10) _
11     & "Enter another point?", VbYesNo)
12 Loop Until (Response = VbNo)
```

A brief explanation of the script is given below.

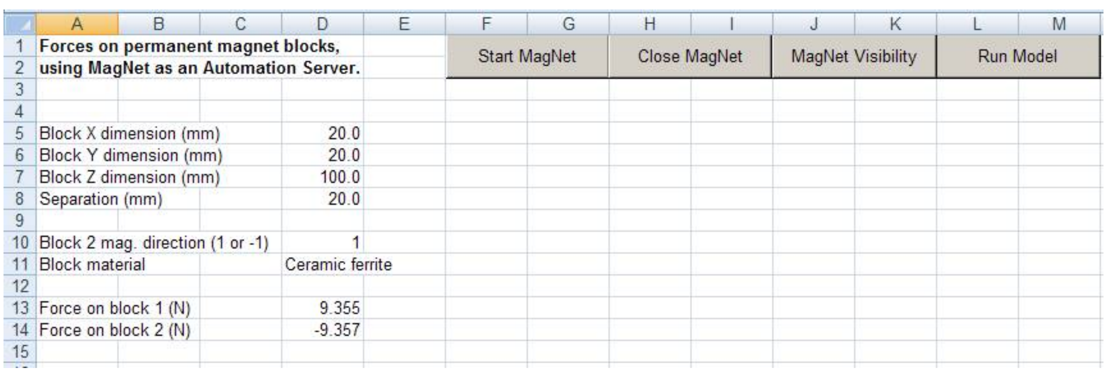
Line	Comment
1	Any line starting with a single quote character (') is a comment, which is ignored when the script runs.
2	Blank lines are ignored when the script runs.
3	This gets the solution mesh and creates an object handle `Mesh`, required in line 5.
4	This gets the required field and creates an object handle `Field`, required in line 9. An underscore preceded by a space character means the statement continues on the next line.
5	This is the continuation of line 4.
6	An array with one element is created, for use in lines 10 and 11.
7	This is the start of a repeat loop that ends at line 13.
8	The VBScript InputBox function is used to get the X co-ordinate value entered by the user.
9	The VBScript InputBox function is used to get the Y co-ordinate value entered by the user.
10	The required field value is returned in the first element of the array `Value`.
11	The VBScript MsgBox is used to display the result and get a yes/no response from the user. Strings are enclosed between double quote (") characters. The & operator joins strings and converts numbers to strings. Chr(10) is a special character, used to start a new line in the message box.
12	This is the continuation of line 11. Returned values are VbYes or VbNo.
13	This is the end of the repeat loop that started at line 7.

Creating and using the script

1. Use Notepad to create the script, and enter the lines exactly as listed on the previous page **except that the line numbers must be omitted.**
2. Check each line very carefully. There must be no mistakes, or the script will probably fail.
3. Save the script in a file named **GetFieldValues.vbs**.
4. In MagNet, open a file for any existing model.
5. Run the GetFieldValues script.
6. If the script fails with an error message, note the line number listed in the message box, and edit the script file with Notepad to correct the error.
7. When the script is working, display a shaded plot of |B| smoothed. Confirm that the values are similar to those obtained with the Probe Field Values tool.
8. To get values of another field, change the quantity specified in line 4 to one of the fields listed in the Field page, for example `"B y"`.

Automation with Excel

This section shows how Microsoft Excel can communicate with MagNet via the ActiveX Automation Interface to set up and analyze the model of two permanent-magnet blocks. The diagram below shows part of the Excel worksheet.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Forces on permanent magnet blocks,**					Start MagNet		Close MagNet		MagNet Visibility		Run Model	
2	**using MagNet as an Automation Server.**												
3													
4													
5	Block X dimension (mm)			20.0									
6	Block Y dimension (mm)			20.0									
7	Block Z dimension (mm)			100.0									
8	Separation (mm)			20.0									
9													
10	Block 2 mag. direction (1 or -1)			1									
11	Block material			Ceramic ferrite									
12													
13	Force on block 1 (N)			9.355									
14	Force on block 2 (N)			-9.357									
15													

The core of the Excel implementation is a set of *macros* or subroutines, written in Visual Basic, which are similar in principle to the scripts discussed above. There are four buttons on the worksheet, each linked to a macro, which do the following:

- Start MagNet: starts the MagNet application and set the length units.
- Close MagNet: closes the model file, close the application and release resources.
- MagNet Visibility: makes Magnet visible or invisible.
- Run Model: gets data values from cells D5 to D11, sends commands to MagNet to build and solve the model, gets the *x*-components of force on the blocks from MagNet, and displays the results in cells D13 and D14.

The macro activated by the Run Model button is the core of the implementation. For clarity, this macro calls other subroutines that carry out specific tasks such as creating one magnet block.

The sections below describe how to set up the Excel worksheet and create the subroutines. This is an advanced topic, which assumes some familiarity with Microsoft Excel and the Visual Basic for Applications (VBA) macro language used in Excel.

The instructions are based on Microsoft Excel 2007; other versions may behave differently.

Excel worksheet – 1

1. Start Excel.
2. Ensure that there is access to Visual Basic as follows:
   - Click the Microsoft Office button, and then click Excel Options.
   - Click Trust Center, and then click Trust Center Settings.
   - Click Macro Settings.
   - In Macro Settings, click Disable all macros with notification.
   - In Developer Macro Settings, set the check box for Trust access to the VBA project object model.
   - Click OK to close the Trust Center window.
   - Click Popular, and set the check box for Show Developer tab in the Ribbon.
   - Click OK to close the Excel Options window.
3. Save the current blank workbook as **PM Blocks.xls**
4. Enter the text and numerical values exactly as shown on the previous page, except for cells D13 and D14.
   - In cells D13 and D14, enter 0 (numeral zero).
   - Set the formatting for cells D5 to D9 to one decimal place.
   - Set the formatting for cells D13 and D14 to three decimal places.
   - All other cells can use the default formatting.

Visual Basic module – 1

Before creating buttons on the worksheet, it is helpful to create some of the Visual Basic subroutines they will use.

1. Click the Developer tab of the Ribbon, and in the Code group click Visual Basic.

   This should display the Visual Basic Editor in a new blank window.
2. On the Insert menu, click Module

   This should open a new Code window named Module1, with an insertion point for text entry.
3. Type the text listed on the next two pages, taking care to copy it accurately. See the section “Comments on the Code” for an explanation of the content.

```
' Permanent magnet blocks with MagNet.

Option Explicit

Dim Mag As Object       ' MagNet application
Dim Doc As Object       ' New document
Dim Cur As Object       ' Current view
Dim Con As Object       ' Infolytica constant
Dim Sol As Object       ' Solution
Dim Visible As Boolean  ' MagNet is visible
Dim Running As Boolean  ' MagNet is running
'
' Data values extracted from Sheet1 (see GetData)
'
Dim Lx As Double        ' Block X dimension
Dim Ly As Double        ' Block Y dimension
Dim Lz As Double        ' Block Z dimension
Dim Lg As Double        ' Gap between blocks
Dim Mx As Double        ' Magnetization direction: +1 or -1.
Dim Material As String  ' Material name

Sub StartMagNet()
' Subroutine called by the Start MagNet button.
' Starts MagNet and sets variables.
    If Running Then
        Call MsgBox("MagNet is already running.", vbOKOnly)
        Exit Sub
    End If
    Set Mag = CreateObject("Magnet.Application")
    Visible = True
    Mag.Visible = Visible
    Set Doc = Mag.newDocument
    Set Con = Mag.getConstants
    Set Cur = Doc.getCurrentView
    Running = True
End Sub

Sub CloseMagNet()
' Subroutine called by the Close MagNet button.
' Closes MagNet and resets variables.
    If Not Running Then
        Call MsgBox("MagNet is not running.", vbOKOnly)
        Exit Sub
    End If
    Call Doc.Close(Con.infoFalse)
    Call Mag.Exit
    Set Mag = Nothing
    Running = False
    Visible = False
End Sub
```

```
Sub Visibility()
' Subroutine called by the MagNet Visibility button.
' Toggles the MagNet visibility flag.
    If Not Running Then
        Call MsgBox("MagNet is not running.", vbOKOnly)
        Exit Sub
    End If
    If Visible Then
        Visible = (MsgBox("MagNet is visible. Change to invisible?", vbYesNo) = vbNo)
        If Not Visible Then
            Mag.Visible = False
        End If
    Else
        Visible = (MsgBox("MagNet is invisible. Change to visible?", vbYesNo) = vbYes)
        If Visible Then
            Mag.Visible = True
        End If
    End If
End Sub
```

Excel worksheet – 2

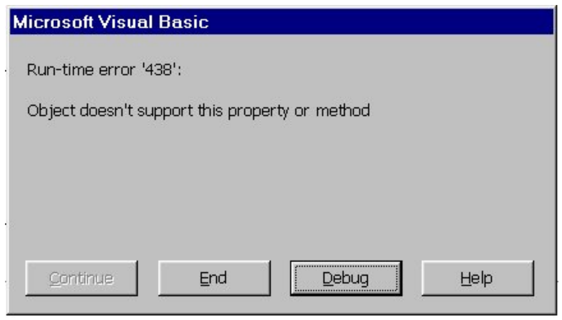
1. Return to the Excel window by clicking the Excel icon on a left-hand toolbar at the top of the Visual Basic Editor window.
2. In the Controls group, click Insert.
   - In the Form Controls area, click the Button icon.
3. Click on worksheet cell F1 to insert a button.

   A button and a dialog box should appear.
4. In the Macro Name drop-down list, select StartMagNet.
5. Click OK.
6. Click inside the new button, and change the text to **Start MagNet**.
7. Hold down the Alt key, and adjust the size and position of the button by dragging the border.

   The button will snap to the worksheet cells.
8. Click outside the button, so that the button is no longer marked with a border.
   - After this, avoid clicking the button. Two more buttons are required before testing can begin.
9. Save the file.
10. In a similar way, insert two more buttons:
    - A button named Close MagNet, linked to the CloseMagNet subroutine.
    - A button named MagNet Visibility, linked to the Visibility subroutine.

Testing – 1

1. Click Start MagNet.
    - If the macro is working correctly, the MagNet application should start, with the MagNet icon shown on the Windows task bar, and the MagNet window visible.
    - If the macro contains fatal errors, there will be a Visual Basic dialog similar to the following.



    - In this case, click Debug.

      This should take you to the line in the Visual Basic code where the error occurred.
    - Correct the error, and press F5 to continue.
    - Continue in this way until the macro appears to be working.
2. Click MagNet Visibility
    - If there are errors, correct them as before.
    - Click Yes to make MagNet invisible.
    - The MagNet window should close, and the MagNet icon should disappear from the Windows task bar.
3. Click MagNet Visibility again, to make MagNet visible.
4. Click Close MagNet.
    - MagNet should close, and the icon should disappear from the task bar.
    - If there are errors, correct them as before.
5. Check that all three buttons work correctly.
    - A message box should be displayed if you try to start MagNet again when it is already running, or use either of the other two buttons when MagNet is not running.
6. Ensure that MagNet is closed before continuing with the development.
    - If you start a second instance of MagNet, it may not work correctly.
    - If this happens, press Ctrl+Alt+Delete and use the Windows Task Manager to close all instances of MagNet.

Completion

1. Return to the Visual Basic editor.
2. Type in the remainder of the code listed on the next three pages.

```
Sub RunModel()
' Subroutine called by the Run Model button.
' Builds and solves the model.

Dim Fx, Fy, Fz

    If Not Running Then
        Call MsgBox("MagNet is not running.", vbOKOnly)
        Exit Sub
    End If
' Build the PM block model
    NewModel
    GetData
    Call MakeBlock(Lx, Ly, Lz, -(Lx + Lg) / 2, 0, "Block#1", True)
    Call MakeBlock(Lx, Ly, Lz, (Lx + Lg) / 2, 0, "Block#2", Mx > 0)
    Call CreateBoundary(Lx, Ly, Lz, Lg)
' Solve the model and get results from MagNet.
    Call Doc.solveStatic2D
    With Sheets("Sheet1")
        Call Sol.getForceOnBody(1, 1, Fx, Fy, Fz)
        .Cells(13, 4).Value = Fx
        Call Sol.getForceOnBody(1, 2, Fx, Fy, Fz)
        .Cells(14, 4).Value = Fx
    End With
End Sub

Private Sub NewModel()
' Closes the current model and starts a new model.
    Call Doc.Close(Con.infoFalse)
    Set Doc = Mag.newDocument
    Set Con = Mag.getConstants
    Set Cur = Doc.getCurrentView
    Set Sol = Doc.getSolution
    Call Doc.setDefaultLengthUnit("Millimeters")
    Call Doc.setScaledToFit(True)
    Call Doc.setPolynomialOrder("", 2)
    Call Doc.useHAdaption(True)
    Call Doc.setHAdaptionRefinement(0.25)
    Call Doc.setAdaptionTolerance(0.001)
End Sub

Private Sub GetData()
' Gets data values from column D on Sheet1.
    With Sheets("Sheet1")
        Lx = .Cells(5, 4)
        Ly = .Cells(6, 4)
        Lz = .Cells(7, 4)
        Lg = .Cells(8, 4)
        Mx = .Cells(10, 4)
        Material = .Cells(11, 4)
    End With
End Sub

Private Sub MakeBlock(Lx, Ly, Lz, Ox, Oy, Name, Positive)
' Makes a magnet block.
' Lx, Ly, Lz are the block dimensions
' Ox, Oy are the center coordinates.
' Name is the component name.
' Positive = True for +ve, False for -ve magnetization direction.

Dim X1 As Double, X2 As Double
Dim Y1 As Double, Y2 As Double
Dim ArrayOfValues(0)
```

```
' Calculate coordinates of opposite corner points.
    X1 = Ox - Lx / 2
    Y1 = Oy - Ly / 2
    X2 = X1 + Lx
    Y2 = Y1 + Ly
' Draw lines to form a rectangle.
    Call Cur.newLine(X1, Y1, X2, Y1)
    Call Cur.newLine(X2, Y1, X2, Y2)
    Call Cur.newLine(X2, Y2, X1, Y2)
    Call Cur.newLine(X1, Y2, X1, Y1)
' Select the surface.
    ArrayOfValues(0) = Con.infoSliceSurface
    Call Cur.selectAt(Ox, Oy, Con.infoSetSelection, ArrayOfValues)
' Make the component.
    ArrayOfValues(0) = Name
    If Positive Then
        Call Cur.makeComponentInALine(Lz, ArrayOfValues, _
        "Name=" + Material + ";Type=Uniform;Direction=[1,0,0]")
    Else
        Call Cur.makeComponentInALine(Lz, ArrayOfValues, _
        "Name=" + Material + ";Type=Uniform;Direction=[-1,0,0]")
    End If
End Sub

Private Sub OpenBoundary(Lx, Ly, Lz, Lg)
' Builds the air space and exterior regions, and applies an even
' periodic boundary condition for the Kelvin transformation.
' Lx, Ly, Lz are the block dimensions, Lg is the gap.

Const Krad = 1.5    ' Radius factor for air space
Const Kmag = 0.1    ' Scale factor for exterior
Const Air = "AirSpace"
Const Ext = "Exterior"
Const Bnd = "BoundaryCondition"

Dim Rb As Double    ' Radius of air space boundary.
Dim ArrayOfValues(), ShiftVec(2), Center(2)

' Delete construction slice lines.
    ReDim ArrayOfValues(0)
    ArrayOfValues(0) = Con.InfoSliceLine
    Call Cur.SelectAll(Con.infoSetSelection, ArrayOfValues)
    Call Cur.deleteSelection
' Build the air space.
    Rb = Krad * Sqr((Lx + Lg) ^ 2 + Ly ^ 2)
    Call MakeCylinder(0, 0, Rb, Lz, Air)
' Build the exterior component.
    Call MakeCylinder(Rb, Rb, Kmag * Rb, Lz, Ext)
' Apply the even periodic boundary condition.
    ReDim ArrayOfValues(1)
    ArrayOfValues(0) = Air & ",Face#3"
    ArrayOfValues(1) = Air & ",Face#4"
    ShiftVec(0) = Rb
    ShiftVec(1) = Rb
    ShiftVec(2) = 0
    Center(0) = 0
    Center(1) = 0
    Center(2) = 0
    Call Doc.createBoundaryCondition(ArrayOfValues, Bnd)
    Call Doc.setEvenPeriodic(Bnd, Kmag, Null, Null, Null, ShiftVec, Center)
End Sub
```

```
Private Sub MakeCylinder(Ox, Oy, Rb, Lz, Name)
' Makes a boundary air cylinder component.
' Ox, Oy are center coordinates, Rb is the radius, Lz the depth.
' Name is the component name.

Dim ArrayOfValues(0)

' Build the cylinder.
    Call Cur.newCircle(Ox, Oy, Rb)
    ArrayOfValues(0) = Con.infoSliceSurface
    Call Cur.selectAt(Ox, Oy, Con.infoSetSelection, ArrayOfValues)
    ArrayOfValues(0) = Name
    Call Cur.makeComponentInALine(Lz, ArrayOfValues, "Name=AIR")
End Sub
```

3 Return to Excel.

4 Insert a new button named Run Model, linked to the Visual Basic subroutine RunModel.

5 Test this button.

- If necessary, correct errors as before.

6 If it is difficult to find the errors, proceed as follows.

7 Place the insertion point anywhere in the RunModel() subroutine.

8 Begin single-step debugging as follows.

- Press F8.

The subroutine header line is marked in yellow.

- Press F8 three times.

The line NewModel is marked in yellow.

- Press F8 again.

The subroutine NewModel is entered.

9 Continue in this way to step through successive lines of the subroutine.

- If you get an error dialog, click Debug to continue.
- To help locate the error, inspect variable values by pausing the pointer over each variable name.
- Correct the error and press F8 to continue.

10 If Visual Basic has to be reset, it is important to close MagNet before continuing.

- If you start a second instance of MagNet, it may not work correctly.
- If this happens, press Ctrl+Alt+Delete and use the Windows Task Manager to close all instances of MagNet.

11 When all the errors have been corrected, single-stepping will reach the end of the RunModel() subroutine, and the result will be the same as if the Run Model button had been clicked on the Excel worksheet.

12 Check that the force values are similar to those displayed in the view of the Excel worksheet on page 109. The values will be slightly different, because the model uses a Kelvin open boundary.

- If there are significant differences, single-step through the macro again.
- Carefully check the numerical values of variables in the macro, and the details of the MagNet model, to find the error.

Using the worksheet

1 Try reversing the direction of magnetization, by entering –1 in cell D10, and click Run Model.

The signs of the force values should reverse, and the magnitudes should be slightly different from the previous values.

2 Change the material to a similar linear material by entering **PM04: Br 0.4 mur 1.0** in cell D11, and click Analyze.

- This text must be entered accurately, or the model will fail because the material is unknown.
- The solution will be very much faster, because no Newton steps are required.

3 Try changing the dimensions of the blocks and the gap between the blocks.

Comments on the code

The commands differ in some respects from those found in a User Script Log file and listed in the MagNet help. In particular, a named constant such as **`infoSetSelection`** has to be given in the form **`Mag.GetConstants.infoSetSelection`**. In the macro, this is simplified to **`Con.infoSetSelection`** by the object variable assignment **`Set Con = Mag.GetConstants`**. The first four subroutines are visible outside the module; the others are private subroutines, which a are only visible within the module.

Subroutine StartMagNet

This subroutine starts the MagNet application, and assigns values to some object variables for convenience in the rest of the module. The value of the global variable **`Running`** indicates whether MagNet is already running. It has the value **`False`** initially, but is set to **`True`** when MagNet is started, and reset to **`False`** when MagNet is closed.

Subroutine CloseMagnet

This subroutine closes the current model if one has been built, then closes the MagNet application and resets variable**`s`**.

Subroutine Visibility

This subroutine determines whether the MagNet window should be visible, by testing the value of the global variable **`Visible`**. The value (**`True`** or **`False`**) is set by the user's response to a question in a dialog box.

Subroutine RunModel

This subroutine calls other subroutines described below to get data from the worksheet and build the model. It solves the model, and displays results for force on the worksheet.

Subroutine NewModel

This subroutine closes the current model and starts a new model. It also resets some object variables so that they refer to the new model, and sets the MagNet solving parameters.

Subroutine GetData

This subroutine gets data from designated cells on the first worksheet, which is assumed to have the Excel default name Sheet1. The subroutine does no data checking, and simply assumes that valid data have been entered in the worksheet. A production version of the subroutine should include checks for valid data.

Subroutine MakeBlock

This subroutine makes a block of permanent-magnet material, using data supplied in the parameter list. It is called twice by the subroutine SolveModel to make the two blocks. The MagNet commands are similar to those found in a User Script Log file, apart from the string manipulation to use the name of the material from the worksheet.

Subroutine OpenBoundary

This subroutine implements the Kelvin transformation to represent an open boundary. It creates an interior air space component surrounding the model, and a small exterior air component. The components are linked with an even periodic boundary condition.

Subroutine MakeCylinder

This subroutine makes a cylindrical air region, using data supplied in the parameter list. It is called twice by the subroutine OpenBoundary to make the air space and exterior regions.

Appendix A

Field Equations and Solution

Field equations

The elementary concepts of electromagnetism covered in Chapter 1 are sufficient for using MagNet to solve practical problems. It is helpful, however, if the user also has some understanding of the basic theory given below. See Cheng [5] for a good introductory account of electromagnetic theory.

The static magnetic field is described by the magnetic flux density **B** and the magnetic intensity **H**, which satisfy the equations

$$\mathrm{curl}\mathbf{H} = \mathbf{J}\,, \tag{A-1}$$

$$\mathrm{div}\mathbf{B} = 0\,, \tag{A-2}$$

$$\mathbf{B} = \mu\mathbf{H} = \mu_0\mu_r\mathbf{H}\,, \tag{A-3}$$

where **J** is the current density, which is the source of the magnetic field. Equation A-1 is the differential form of Ampère’s circuital law – equation 1-2 of Chapter 1 – and equation A-2 embodies the fact that there are no free magnetic poles, so magnetic flux lines are closed curves.

To solve these equations we use equation A-2 to express the flux density **B** in terms of another vector **A** through the equation

$$\mathbf{B} = \mathrm{curl}\mathbf{A}\,, \tag{A-4}$$

where **A** is known as the magnetic vector potential. Equation A-1 then becomes

$$\mathrm{curl}\left(\frac{1}{\mu}\mathrm{curl}\mathbf{A}\right) = \mathbf{J}\,, \tag{A-5}$$

which is solved numerically by the finite-element method to determine the magnetic field in the device (see the section “Numerical Solution” on page 123).

In two dimensions, the magnetic vector potential has some very useful properties. For the XY plane of translational geometry, the current and therefore the vector potential is in the *z* direction. The components of **B** are then given by

$$B_x = \frac{\partial A}{\partial y},\ \ B_y = -\frac{\partial A}{\partial x} \tag{A-6}$$

where $A = A_z$ is the magnitude of the vector potential. The scalar quantity A will be termed the magnetic potential. From equation A-6 it may be shown that equipotentials – lines of constant A – are flux lines. Moreover, the magnetic flux ϕ between any two points in the XY plane, for a depth d in the z direction, is given by

$$\phi = d\,(A_1 - A_2)\,, \tag{A-7}$$

where A_1 and A_2 are the values of A at those points. Thus A is the flux per meter depth, with units of Wb/m. The flux plots for translational geometry in MagNet are contours of constant A.

With the RZ plane of rotational geometry, a modified potential is required for numerical stability and accuracy [2]; MagNet uses the quantity

$$U = rA\,. \tag{A-8}$$

It may be shown that lines of constant U are flux lines in rotational geometry, and that the flux between two points is given by

$$\phi = 2\pi(U_1 - U_2)\,, \tag{A-9}$$

where U_1 and U_2 are the values of U at those points. Thus U is the flux per radian. The flux plots for rotational geometry in MagNet are contours of constant U.

The components of **B** are not given by equation A-6 in rotational geometry with r and z substituted for x and y. Instead, in cylindrical polar coordinates, we have:

$$B_r = -\frac{\partial A}{\partial z} = -\frac{1}{r}\frac{\partial U}{\partial z},\;\; B_z = \frac{1}{r}\frac{\partial}{\partial r}(rA) = \frac{1}{r}\frac{\partial U}{\partial r} \tag{A-10}$$

Boundary conditions and symmetry

Boundary conditions

To solve the field equations, it is necessary to specify what happens to the field beyond the device. Theoretically, the field extends to infinity, which implies an "open boundary" – a boundary at infinity. For the restricted case of some 2D static problems in translational geometry, an open boundary can be modeled exactly with the Kelvin transformation technique [7], which is used in some electromagnetics packages. MagNet does not use this technique automatically, but it can be applied to specific problems in a simple way. Its use is described below on page 122, and an example is given in the busbar forces case study on page 48. If the Kelvin transformation technique is not applicable, it is necessary to specify an artificial far-field boundary.

In 2D, an artificial boundary takes the form of a closed curve along which a property of the field is specified. The field property is the *boundary condition.* Two kinds of boundary condition are relatively easy to implement when solving the field equations:

- The Dirichlet, or *flux tangential*, boundary condition. The flux function (A or U) is constant over any portion of the boundary with this condition, so the portion becomes part of a flux line.
- The Neumann, or *field normal*, boundary condition. The direction of **B** is at right angles to any portion of the boundary with this condition, so the flux lines enter the portion at right angles.

The default boundary condition in MagNet is flux tangential, so the entire outer boundary will become a flux line unless the user specifies otherwise. For most problems this is the best choice. It is equivalent to putting the model in a cavity of a material with zero permeability, so that no flux can escape from the model. If the boundary is taken sufficiently far away from the components of the model, it is a good approximation to an open boundary. A radius of about 10 times the model dimensions will be sufficient in many applications.

If the field normal boundary condition is applied to the entire boundary, the effect is equivalent to putting the model in a cavity of a material with infinite permeability. This has the opposite effect to the flux tangential boundary: it draws flux away from the model. This boundary condition is usually not as good as the flux tangential boundary for representing an open boundary, but it has a number of uses:

- It is a simple way of simulating the effect of a magnetic screen around the model.
- It can be used for creating artificial field patterns, such as the uniform field of finite extent in the case study on the cylindrical screen (page 57).
- It can be used to reduce the size of a model by exploiting symmetry (see below).

For any given artificial boundary shape, it may be shown that the true field in the model lies between two extremes: the result with a flux tangential boundary condition, and the result with a field normal boundary condition. A method of checking the size of the boundary is therefore to repeat the solution using the other boundary condition and compare the results.

Symmetry and periodicity

A line of symmetry in a device is frequently either a flux line or a line normal to flux lines. These conditions can be represented by drawing a closed boundary round part of the model and imposing the flux tangential or field normal conditions as required. In this way, only a portion of the device needs to be modeled. See the case study on the cylindrical screen (page 57) for an example.

Devices such as rotating electrical machines have a periodic structure, where the magnetic field conditions in one part of the device are similar to the conditions in another part. It is possible to represent the device by a model of a small part, for example one or two pole pitches in a rotating machine. The periodic nature of the device is represented by a constraint between the field values at the two ends of the representative portion. This constraint is termed a *periodic boundary condition.* If the field values at the two ends are equal in magnitude and sign, the constraint is an *even periodic* boundary condition. A model of two complete pole pitches of a machine would require this constraint. If the field values are equal in magnitude but opposite in sign, the constraint is an *odd periodic* boundary condition. A model of one pole pitch of a machine would require this constraint, and an example is given in the case study of linear synchronous motors on page 76.

Open boundaries

Consider a 2D static problem with translational symmetry that has an open boundary. Let the 2D model be enclosed in a circle of radius *R*, so that all the material parts of the model are inside the circle. The field region comprises two parts: the finite interior of the circle, and the infinite space outside the circle. Since the infinite space region is current-free, so that **J** = 0, it follows from equation A-1 that the magnetic intensity **H** can be represented by the gradient of a scalar potential:

$$\mathbf{H} = -\mathrm{grad}\,\Omega \qquad \text{(A-11)}$$

There is a further restriction, however: for equation A-11 to hold, we must also have:

$$\oint \mathbf{H} \cdot \mathbf{d}\boldsymbol{l} = 0 \qquad \text{(A-12)}$$

The integral form of equation A-1 is Ampere’s circuital law:

$$\oint \mathbf{H} \cdot \mathbf{d}\boldsymbol{l} = \sum i \qquad \text{(A-13)}$$

where the right-hand side is the sum of all currents passing through the contour of integration. Equation A-12 therefore implies that the model cannot contain any unbalanced currents.

Provided equation A-11 holds, the Kelvin transformation [7] gives the following useful result: the infinite space outside the circle of radius *R* is equivalent to the interior of another circle of any finite radius *kR*, if the potentials of all corresponding points on the two circles are identical. To implement this in MagNet, it is necessary to construct two air boxes: a normal circular air box surrounding the model, and a second circular air box to represent the infinite external space. An even periodic boundary condition is imposed on corresponding curved surfaces of the two air boxes, forcing the potentials to have the same values at corresponding points. An example of a manual implementation is given in the case study on busbar forces (page 48), and a scripting example is given on page 109.

Other techniques can be used to simulate open boundaries in 2D models with rotational symmetry, and in 3D models, but they will not be considered in this document.

Numerical solution

Introduction

The core of MagNet is a powerful technique for solving the electromagnetic field equations numerically. Most of this process is automatic and virtually transparent to the user, but it is necessary to control the process by setting the solver and adaption options. As with the fundamental equations, it is helpful if the user has some understanding of the method.

MagNet employs the finite-element method [8] to solve the 2D form of equation A-5 for the magnetic potential. With this method, the region of the problem is divided into a mesh of triangular elements, and the potential in each element is approximated by a simple function of the x and y (or r and z) coordinates. The simplest function is a linear variation with position; this gives first-order elements, where the potential inside a triangular element is obtained from the potentials at the three vertices or nodes. High-order elements use high-order polynomials and additional nodes to represent the potential. The problem of solving equation A-5 then reduces to the solution of a set of linear equations for the unknown potentials at all the nodes. This must be repeated several times if the model contains non-linear magnetic materials.

The accuracy of the finite-element solution depends on three factors: the nature of the field, the size of the elements, and the element order. In regions where the direction or magnitude of the field is changing rapidly, high accuracy requires small elements or a high element order. In addition, the methods used to find the finite-element solution are iterative, with an adjustable error criterion for terminating the process.

Solver

When non-linear magnetic materials are present, the permeability μ depends on the local value of B. Equation A-5 is solved as follows:

- Constant values of permeability are chosen for each element, from the initial slope of the material B-H curve.
- The resulting linear equations are solved numerically for the magnetic potential, using the semi-iterative *conjugate gradient* method.
- The flux density values are calculated from the magnetic potential, and these results are used to calculate new values for the permeability of each element.
- The process is repeated until the element permeability values have converged.

CG steps

At each step in the conjugate gradient process, the change in the solution is monitored. The process continues until the change is less than the CG Tolerance. For most static problems, the default value of 0.01% should be satisfactory. Time-harmonic eddy-current problems frequently require a value at least 10 times smaller for accurate results.

Method of permeability calculation

For 3D problems, MagNet gives a choice of the Newton-Raphson method or successive substitution for calculating updated element permeability values. The default Newton-Raphson method normally converges more rapidly, but there can be convergence problems with some material characteristics, in which case the successive substitution method is required. Only the Newton-Raphson method is available for 2D problems.

Newton steps

At each step in the permeability calculation process, the change in the solution is monitored. The process continues either until the change is less than the Newton Tolerance, or until the limit of Maximum Newton Iterations is reached. For most problems, the default values of 1% and 20 iterations should be satisfactory. Examples of problems where these settings are not sufficient are (a) permanent-magnet materials in contact with non-linear soft magnetic materials, in a device where the flux density values are very high, and (b) closed magnetic circuits in iron surrounding a current (like a closed slot in an electrical machine). For these kinds of problem, the tolerance should be reduced to 0.01% and the iteration limit increased to 50 or more.

Polynomial order

The solver polynomial order setting is a global value that applies throughout the model. If its setting is left at Default, MagNet will use order 1 in translational geometry, and order 2 in rotational geometry. Order 1 gives a fast solution of low accuracy, and is useful for initial tests on a complex model, but it is not satisfactory in rotational geometry. For 3D models the polynomial order of elements in particular components can be specified separately (see “Control of the mesh structure” on page 125), but for 2D models the solver polynomial order option sets the value that will be used for the entire model.

With some models, increasing the polynomial order is as effective as using adaption to improve the solution accuracy. In most cases, however, good results will be obtained by setting the polynomial order to 2 and using adaption as described below. In special cases, a high order must be used in conjunction with adaption: see the case study on magnetic pull-off force (page 95).

Adaption

Adaption is the process of automatic refinement of the mesh to improve the solution accuracy. For 3D models there is a choice of two adaption methods: h-type adaption, where element sizes are halved, and p-type adaption, where the element polynomial order is increased. For 2D models, only h-type adaption is available.

A consequence of the finite-element approximation to the true field is a discontinuity in the value of B from one element to the next. MagNet determines which elements to refine by calculating the discontinuity error values. At each adaption step, elements with the largest error are refined first. The adaption option Percentage of Elements to Refine determines the percentage of the total number of elements that will be refined at each step. The default value of 25% is generally satisfactory for 2D models. For 3D models, where the number of elements increases very rapidly at each step, a lower value is appropriate.

After each adaption step, the change in the calculated value of stored magnetic energy is monitored. Adaption continues until this change is less than a specified tolerance, or the specified number of steps has been reached. As the case studies demonstrate, the default tolerance of 1% is generally too large for a high-accuracy solution. If the quantity of interest is the force or torque, rather than an energy-related quantity such as inductance, a more accurate solution may be required, and the change in the stored magnetic energy may not be a good indicator. In such cases it is often advantageous to set the tolerance to a very low (but non-zero) value, and control the mesh refinement by adjusting the maximum number of adaption steps. The optimum setting can be determined by changing the number of steps and monitoring the change in the force or torque value.

Control of the mesh structure

If the user takes no action, MagNet will determine the initial mesh automatically. Adaption can then be used to refine the mesh to get an accurate solution. For most 2D problems, this should be satisfactory. In cases where this process fails, or gives very long solution times, the user can exercise control of the mesh structure by specifying the following quantities.

- Maximum element size: the maximum element edge length. This can be increased to force adaption to start with a coarse mesh, or reduced to give a fine mesh.
- Curvature refinement ratio: a measure of the maximum deviation when a curved part of the model is approximated by the straight-line edge of an element.
- Curvature refinement minimum element size: limits over-discretization of tight curves when the elements are refined.

These properties can be set for the entire model, or on individual components, surfaces and edges. In addition, the mesh can be controlled by *edge subdivision*. This feature of MagNet enables the user to specify the number of segments on a given component edge (line or arc) when the initial mesh is generated. The subdivisions can be linear or logarithmic. Details of the procedures for setting the mesh properties are given in the MagNet help. For most purposes, the user is recommended to avoid using edge subdivision because the other methods are more effective.

The case study on the linear synchronous motor (page 76) gives an example of setting the maximum element size in the small airgap of the machine. In this case, the method is more effective than adaption for getting accurate force values.

Appendix B

Energy, Force and Inductance

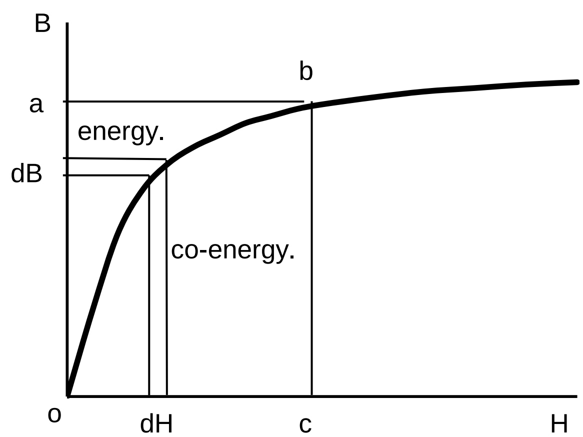
Stored energy and co-energy

Definitions

When a magnetic field is established in a device by increasing the currents in the coils from zero to some final value, energy must be supplied in addition to the i^2R ohmic loss in the conducting material. This energy is considered to be stored in the magnetic field; if hysteresis is neglected, it can be recovered when the currents are reduced to zero.

The diagram below shows part of the *B-H* curve for a magnetic material. If point *b* represents the final magnetic condition in a particular part of the material, then it may be shown [9] that the energy density, or energy stored per unit volume, is given by:

$$w = \int \mathbf{H} \cdot \mathbf{dB} \quad [\mathrm{J/m^3}] \tag{B-1}$$



The energy density is the area between the *B-H* curve and the *B* axis, which is the area *oab* in the diagram.

A related quantity is the co-energy density given by

$$w' = \int \mathbf{B} \cdot \mathbf{dH} \quad [\mathrm{J/m^3}]. \tag{B-2}$$

The co-energy density is the area between the curve and the *H* axis; this is equal to the area *ocb* in the diagram. Since the sum of the two densities, $w + w'$, must equal the area of the rectangle *oabc*, we have

$$w' = \int \mathbf{B} \cdot \mathbf{dH} = \mathbf{B} \cdot \mathbf{H} - w = \mathbf{B} \cdot \mathbf{H} - \int \mathbf{H} \cdot \mathbf{dB} \quad [\mathrm{J/m^3}]. \tag{B-3}$$

If the material is linear, then the energy density and the co-energy density are both equal to half the area of the rectangle *oabc*. For a non-linear material the energy density is less than this area, and the co-energy density is greater than this area; we therefore have

$$w < \tfrac{1}{2}\mathbf{B}.\mathbf{H} < w' \tag{B-4}$$

The total energy stored in the device is just the integral of the energy density over the volume:

$$W = \int w\,dv = \int \left(\int \mathbf{H} \cdot \mathbf{dB} \right) dv \quad [\mathrm{J}]. \tag{B-5}$$

Similarly, the total co-energy is the integral of the co-energy density over the volume:

$$W' = \int w'\,dv = \int \left(\mathbf{B} \cdot \mathbf{H} - \int \mathbf{H} \cdot \mathbf{dB} \right) dv \quad [\mathrm{J}]. \tag{B-6}$$

When a device is solved in MagNet, the stored energy and co-energy values displayed in the Energy tab of the Post-processing bar are the values of W and W' given by equations B-5 and B-6.

Applications

We have seen that the stored energy and the co-energy are identical for linear materials, but for non-linear materials the co-energy is greater than the stored energy. The difference between the two values is a measure of the non-linearity, and hence the degree of saturation, of the magnetic materials in the device.

The usual definition of inductance is the flux linkage per ampere (see page 130). If the stored energy is calculated in the usual way as $\tfrac{1}{2}Li^2$, this will differ from both the stored energy W and the co-energy W'; in fact it is equal to the equivalent linear energy

$$W_{lin} = \int \tfrac{1}{2}\mathbf{B} \cdot \mathbf{H}\,dv \quad [\mathrm{J}] \tag{B-7}$$

which is greater than W and less than W'. If a precise value of the stored energy is required, for example when energy is dumped from a highly saturated inductor, then the value of the stored energy W will be a better estimate than the value of $\tfrac{1}{2}Li^2$.

The co-energy finds its application in the virtual work method of calculating forces and torques; this is quite different from the method used in MagNet and described on page 129. If the current is held constant and a part of the device is given a displacement s in any direction, it may be shown that the component of force in the direction of s is given by

$$F_s = \frac{\partial W'}{\partial s} \approx \frac{\Delta W'}{\Delta s} \tag{B-8}$$

If two models are constructed which differ by Δs in a distance s, then the force may be calculated from equation B-8 by taking the difference of the co-energy values. A similar equation holds for torque in terms of an angular displacement θ:

$$T_\theta = \frac{\partial W'}{\partial \theta} \approx \frac{\Delta W'}{\Delta \theta} \tag{B-9}$$

Force calculation

Lorentz force

The force per unit volume on a current-carrying conductor in a magnetic field is

$$\mathbf{f} = \mathbf{J} \times \mathbf{B} \tag{B-10}$$

which is another form of the Lorentz equation given in equation 1-1 of chapter 1. The total force is therefore given by the integral

$$\mathbf{F} = \int \mathbf{J} \times \mathbf{B}\, dv \tag{B-11}$$

and the total torque about a point P is

$$\mathbf{T} = \int \mathbf{r} \times \mathbf{J} \times \mathbf{B}\, dv \tag{B-12}$$

where $\mathbf{r}$ is the radius vector from P to the volume element, and the integral is taken over the volume of the conductor. These equations are known to give good results when the flux density $\mathbf{B}$ is determined numerically by the finite-element method, so MagNet implements equations B-11 and B-12 to calculate the force and the torque on current-carrying regions in the model, provided the material has a relative permeability $\mu_r = 1$.

Maxwell stress

Equations B-11 and B-12 can be used only for current-carrying conductors; they do not give the force and torque on magnetic materials. If a device includes magnetic materials as well as current-carrying conductors, MagNet uses a method of force calculation based on the Maxwell stress concept [9, 10] which gives the stress, or force per unit area, directly in terms of the magnetic flux density. If B_n and B_t are the components of flux density normal and tangential to a surface, and σ_n and σ_t are the corresponding components of stress, then:

$$\sigma_n = \frac{B_n^2 - B_t^2}{2\mu_0} \tag{B-13}$$

$$\sigma_t = \frac{B_n B_t}{\mu_0} \tag{B-14}$$

Provided that the surface is closed, and passes entirely through air, the total force and torque may be determined by integrating the stresses over the surface. This result is completely general; it is independent of the nature of the objects inside the surface, which may include currents, soft magnetic materials or permanent magnets.

If the Maxwell stress method is used to calculate forces from a standard numerical solution for the field, it is difficult to get accurate results. The integral for the force or torque may be unreliable if it comprises terms which alternate in sign, leading to an accumulation of numerical errors. It is also very sensitive to the accuracy of the numerical solution. Users require considerable skill and experience to get good results by this method.

MagNet avoids this difficulty with the conventional Maxwell stress method by implementing a novel tunable method [11]. This requires no skill on the part of the user, and it gives accurate values for the force and the torque.

Forces on objects

MagNet calculates the forces and torques on all the bodies in a device automatically, using the tunable Maxwell stress method. A body is defined as a set of connected regions completely surrounded by the special material AIR. These quantities are displayed in the Results window of the View area, and they are also given in the Solver Log file that is generated automatically when the model is solved. This file can be accessed from the Help menu.

The Lorentz $\mathbf{J} \times \mathbf{B}$ forces and torques are calculated for any current-carrying regions where the material has a relative permeability $\mu_r = 1$. These quantities are not displayed in the Results window, but are given in the Solver Log file. Since these forces and torques are included in the total force and torque on a body containing the current-carrying region, the force and torque on the magnetic material alone may be determined by subtraction.

Inductance calculation

Flux linkage

The magnetic properties of any pair of coupled coils may be described in circuit terms by self-inductances L_1 and L_2 and mutual inductances M_{12} and M_{21}. If coil 1 carries a current i_1, the flux linkages with coils 1 and 2 are given by

$$\lambda_1 = N_1\phi_1 = L_1 i_1 , \tag{B-15}$$

$$\lambda_{12} = N_2\phi_{12} = M_{12} i_1 , \tag{B-16}$$

where N_1 and N_2 are the numbers of turns on the coils. Similarly, if coil 2 carries a current i_2, the flux linkages are

$$\lambda_2 = N_2\phi_2 = L_2 i_2 , \tag{B-17}$$

$$\lambda_{21} = N_1\phi_{21} = M_{21} i_2 . \tag{B-18}$$

The inductances are the flux linkages per ampere, and are given by:

$$L_1 = \frac{\lambda_1}{i_1} \tag{B-19}$$

$$L_2 = \frac{\lambda_2}{i_2} \tag{B-20}$$

$$M_{12} = \frac{\lambda_{12}}{i_1} \tag{B-21}$$

$$M_{21} = \frac{\lambda_{21}}{i_2} \tag{B-22}$$

For a linear system, all four coefficients are independent of current, and $M_{12} = M_{21} = M$. When $N_1 = N_2$, leakage inductances l_1 and l_2 may be defined as follows:

$$l_1 = L_1 - M , \;\; l_2 = L_2 - M \tag{B-23}$$

These leakage inductances represent flux produced by one winding which fails to link with the second winding, but “leaks” into the surrounding air (or other non-magnetic material) instead.

For devices in which the magnetic materials are linear and soft, there is an alternative method of calculation based on energy.

Energy methods

For a linear system, the stored magnetic energy in a pair of coils is given by

$$W = \tfrac{1}{2}L_1 i_1^2 + \tfrac{1}{2}L_2 i_2^2 + M i_1 i_2 . \qquad \text{(B-24)}$$

Alternative methods of calculating mutual inductance and leakage inductance can be derived by equating this expression to the stored energy given by equation B-5.

To calculate mutual inductance, consider two situations: (a) identical currents, so that $i_1 = i_2 = i$; (b) equal and opposite currents, so that $i_1 = -i_2 = i$. The stored energy values are:

$$W_a = \tfrac{1}{2}L_1 i^2 + \tfrac{1}{2}L_2 i^2 + M i^2 \qquad \text{(B-25)}$$

$$W_b = \tfrac{1}{2}L_1 i^2 + \tfrac{1}{2}L_2 i^2 - M i^2 \qquad \text{(B-26)}$$

from which it follows that

$$M = \frac{W_a - W_b}{2i^2} . \qquad \text{(B-27)}$$

Equation B-23 for the leakage inductances may be unsatisfactory when the values are given by the small differences between large quantities. An energy method can be based on a simulation of the transformer short-circuit test, where the two coils carry equal and opposite currents. In terms of the leakage inductances, equation B-24 becomes

$$W = \tfrac{1}{2}l_1 i_1^2 + \tfrac{1}{2}l_2 i_2^2 + \tfrac{1}{2}M(i_1 + i_2)^2 . \qquad \text{(B-28)}$$

For a simulated short-circuit test where $i_1 = -i_2 = i_s$, the stored energy is

$$W_s = \tfrac{1}{2}(l_1 + l_2) i_s^2 , \qquad \text{(B-29)}$$

which gives

$$l_1 + l_2 = \frac{2W_s}{i_s^2} . \qquad \text{(B-30)}$$

As with the physical short-circuit test, this method can determine only the sum of the leakage inductances, not their individual values; but it may give more precise results for the sum than equation B-23.

References

1 L.V. Bewley, Two-Dimensional Fields in Electrical Engineering. New York: Dover, 1948.

2 C.W. Steele, Numerical Computation of Electric and Magnetic Fields. New York: Van Nostrand, 1987.

3 B. Popovic, Introductory Engineering Electromagnetics. Addison Wesley, 1971.

4 L. Grover, Inductance Calculations – Working Formulas and Tables. New York: Dover, 1962.

5 D.K. Cheng, Fundamentals of Engineering Electromagnetics. Addison-Wesley, 1993.

6 D.A. Lowther and P.P. Silvester, Computer-Aided Design in Magnetics. Springer-Verlag, 1985.

7 E.M. Freeman and D.A. Lowther, 'A novel mapping technique for open boundary finite element solutions to Poisson's equation', *IEEE Transactions on Magnetics*, **24**, 1988, pp. 1934-2936.

8 P.P. Silvester and R.L. Ferrari, Finite Elements for Electrical Engineers, 2nd ed. Cambridge University Press, 1990.

9 J. A. Stratton, Electromagnetic Theory. New York: McGraw Hill, 1941.

10 C. J. Carpenter, 'Surface integral methods of calculating forces on magnetised iron parts', *Proc. IEE*, **107C**, 1960, pp. 19-28.

11 S. McFee, J. P. Webb and D. A. Lowther, 'A tunable volume integration formulation for force calculation in finite-element based computational magnetostatics', *IEEE Transactions on Magnetics*, **24,** 1988, pp. 439-42.