YOUR FASTEST SOLUTION TO A BETTER DESIGN

An introduction to

ElecNet

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 ElecNet for solving 2D static electric field and current-flow problems. A tutorial with detailed instructions takes the first-time user through the most important features of ElecNet. 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 ElecNet a uniquely powerful tool.

What is ElecNet?

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

ElecNet is designed as a full 3D-modeling tool for solving static and low frequency time-varying electric field problems. Many devices can be represented very well by 2D models, so ElecNet offers the option of 2D modeling, with a substantial saving in computing resources and solution time.

A feature of ElecNet 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, ElecNet offers powerful facilities for user-defined adjustment of the model parameters, and control of the operation of the package with scripts and spreadsheets.

Limitations

The information given in this document has been prepared for the free Trial Edition of ElecNet, and for the full version where the licensed features are restricted to static electric fields and 2D models. The full version of ElecNet 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 this chapter give some background information for first-time users of software for electrostatic and current-flow field problems, 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 ElecNet in the form of a tutorial. It takes the user through all the steps of modeling a simple electrostatic 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 ElecNet is applied to a variety of electrostatic and current-flow 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 ElecNet, including the use of Microsoft Excel to control ElecNet. Scripting is now available in all versions of ElecNet, and chapter 5 indicates some of the ways of using this feature.

Appendix A contains further information about the electric field equations and the solution methods used in ElecNet 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.

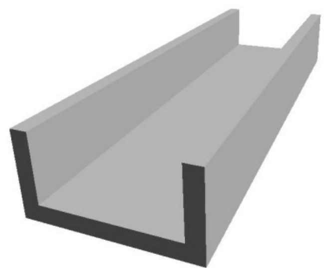
Appendix B covers energy, force and capacitance calculation. This includes the derivation of some of the equations used in the case studies, and further information about the methods used in ElecNet.

Modeling in 2D and 3D

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

Translational geometry

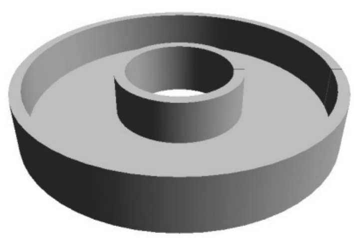
Translational geometry means that the object has a constant cross-sectional shape generated by translation – moving the shape in a fixed direction. The diagram below shows an example.



With translational geometry, any slice perpendicular to the axis has the same shape. Inevitably, this 2D approximation neglects fringing 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 U 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 capacitance. For all built-in calculations, ElecNet handles these differences automatically, so the user does not need to take account of the differences.

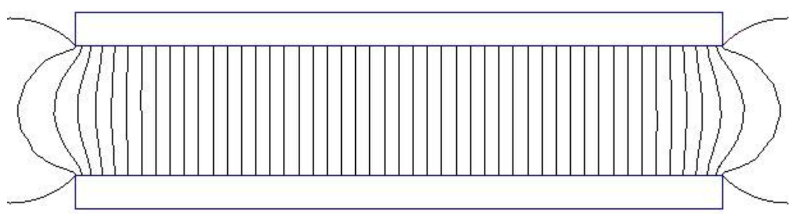
In ElecNet, the 2D cross-section of a rotationally symmetric model must be drawn in the XY plane, with the Y axis as the axis of rotation. The XY coordinates then correspond to the RZ coordinates of the conventional cylindrical polar coordinate representation.

Electric field concepts

ElecNet 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 electric field concepts that are required for making effective use of ElecNet; more advanced topics are covered in appendix A. The system of units used is the SI or MKSA system.

Electric field strength E

The fundamental concept is the electric field described by the vector **E**, which is termed the *electric field strength.* In two dimensions, this field is commonly represented by lines, known as field lines, which show both the direction and the magnitude of **E**. The direction of a line gives the direction of **E**, and the spacing of the lines indicates the magnitude; the closer the lines, the greater the magnitude. The diagram below shows the field plot for a simple air-spaced parallel-plate capacitor, where an external source maintains a constant potential difference (voltage) between the plates.



Field plot for a parallel-plate capacitor

Although the electric field is an abstract concept, the effects of **E** are concrete and physical. The mechanical force in a device such as this capacitor can be expressed in terms of **E**. In simplified terms, the field lines can be treated as elastic bands pulling the plates together with a tensile stress (force per unit area) given by $\frac{1}{2}\varepsilon_0 E^2$. In this expression, E is the magnitude of the vector **E**, and $\varepsilon_0 = 8.854 \times 10^{-12}$ is a fundamental constant. The unit of E is the volt per meter (V/m), and the unit of ε_0 is the farad per meter (F/m).

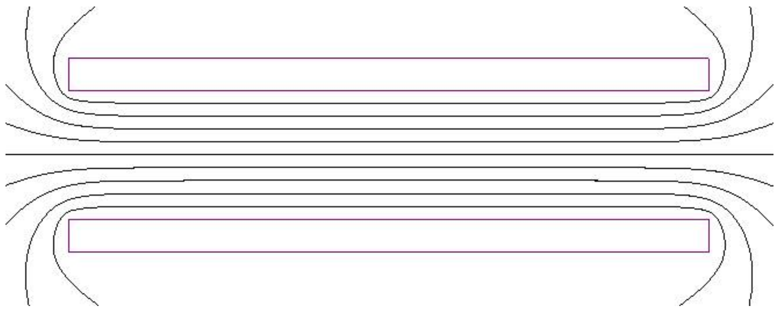
A direct physical interpretation of **E** is given by the expression for the electric force on a stationary electric charge q:

$$\mathbf{f} = q\mathbf{E} \tag{1-1}$$

where **f** is the force in newtons and q is the charge in coulombs. A consequence of this force is that high values of electric field strength E can result in electrical breakdown of an insulating material.

Electric potential V

If the electric field does not vary with time, the vector **E** can be expressed as the gradient of a scalar quantity *V*, known as the *electric potential* or *voltage*. Since scalar quantities are much easier to handle than vector quantities, it is usual to solve electric field problems by calculating the potential distribution, and to represent the field by an *equipotential plot* – contours of constant *V*. The diagram below shows the equipotential plot for the same parallel-plate capacitor.



Equipotential plot for a parallel-plate capacitor

The equipotential lines are orthogonal to the field lines of the previous plot. In the middle of the region between the plates, the field is uniform; this is shown by uniformly spaced parallel lines in both the field plot and the equipotential plot.

ElecNet can generate equipotential plots automatically, but it cannot generate conventional plots of field lines in all cases. Instead, ElecNet displays shaded color plots of quantities such as $E = |\mathbf{E}|$. In most cases, the combination of an equipotential plot and a shaded plot of *E* is at least as useful as the conventional field plot. Contour plots of the flux function in ElecNet show lines of the electric flux density **D**, which resemble lines of **E** in many cases.

Electric flux density D

When dielectric (insulating) materials are present, it is necessary to introduce a new electric quantity: the *electric flux density* **D**. This quantity is related to electric charge, which is the ultimate source of the electric field, through Gauss's law:

$$\oint \mathbf{D} \cdot \mathbf{da} = \sum q \qquad (1\text{-}2)$$

where the integral on the left is taken over a closed surface, and the summation on the right is the sum of the electric charge (measured in coulombs) enclosed by the surface. From equation 1-2, **D** has units of coulombs per square meter.

For most dielectric materials, when the field is not varying with time, there is a simple linear relationship between the electric flux density **D** and the electric field strength **E**:

$$\mathbf{D} = \varepsilon_r \varepsilon_0 \mathbf{E} \qquad (1\text{-}3)$$

where the dimensionless quantity ε_r is a property of the material known as the *relative permittivity* or the *dielectric constant.* For empty space, and to a close approximation for air, $\varepsilon_r = 1$

If the electric field is perpendicular to the boundary between two materials, and there is no surface charge at the boundary, equation 1-2 shows that the value of D will be the same in the two materials. If the relative permittivity ε_r has values ε_1 and ε_2 in the two materials, equation 1-3 shows that the corresponding values of E are related by:

$$\frac{E_1}{E_2} = \frac{\varepsilon_2}{\varepsilon_1} \tag{1-4}$$

Equation 1-4 has important consequences for voids in a dielectric material, where the value of ε_r in the material is higher than the value of ε_r in the void. In this case, the value of E in the void will be higher than in the surrounding material, and this may result in electrical breakdown in the void.

Current density J

In a conducting material, an applied electric field **E** will result in a flow of electric current represented by the vector **J**. In most conductors, these are linearly related:

$$\mathbf{J} = \sigma \mathbf{E} \tag{1-5}$$

where **J** is the *current density* in amperes per square meter (A/m^2), and σ is the *conductivity* of the material in siemens per meter (S/m). Equation 1-5 is the field equivalent of Ohm's law for conductors.

There is an important difference between current-flow fields represented by **J** in conductors, and electrostatic fields represented by **E** in devices such as capacitors. Most electrostatic fields are unbounded: they extend beyond the ends of the electrodes, as can be seen in the field plot for the parallel-plate capacitor. In principle, the field extends to infinity, although the magnitude decays rapidly with distance. When modeling such devices, it is necessary to impose an artificial boundary to limit the field to a finite region, and this is a possible source of error in the solution. This problem does not arise with current-flow fields. When a conductor is embedded in an insulating material for which the conductivity is effectively zero, the current flow is limited to the volume of the conductor. This difference between electrostatic and current-flow fields will be explored in the case studies.

Using ElecNet effectively

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

The principle of progressive refinement

The time taken for ElecNet 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. 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 field plot as quickly as possible, because this 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 ElecNet to improve the solution accuracy.

When the new model is producing a sensible field plot, and the numerical results for force, torque and capacitance are plausible, the solver and automatic mesh adaption options can be used to improve the accuracy. The case studies give examples of the settings that may be required. For the advanced user, ElecNet has methods of controlling the mesh structure that can be used as an alternative or a supplement to automatic mesh adaption: see Appendix A.

Getting accurate results

ElecNet 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 ElecNet 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 electrical properties of the materials.

Calculated values for force and torque 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 they 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 ElecNet. 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 other 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: see Appendix A.

Getting help

ElecNet 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 ElecNet, 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 ElecNet to a variety of electric field modeling problems.

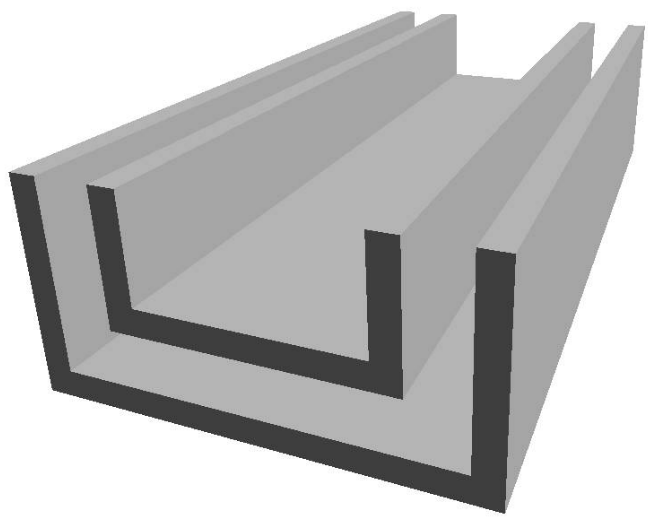
Chapter 2

Tutorial: Parallel Channels

Introduction

This chapter takes the user through the complete sequence of using ElecNet to model a simple electrostatic device: the arrangement of parallel conducting channels shown below. The objectives are as follows:

- To examine the electric field and potential in the space between the conductors.
- To determine the forces on the conductors
- To determine the capacitance of the system.
- To modify the model by changing the position of the inner conductor.



This is an example of a device that can be represented quite well by a 2D model. The fringing field above the channels 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 and capacitance. Fringing is discussed later, on page 23.

Device Model

Brief description

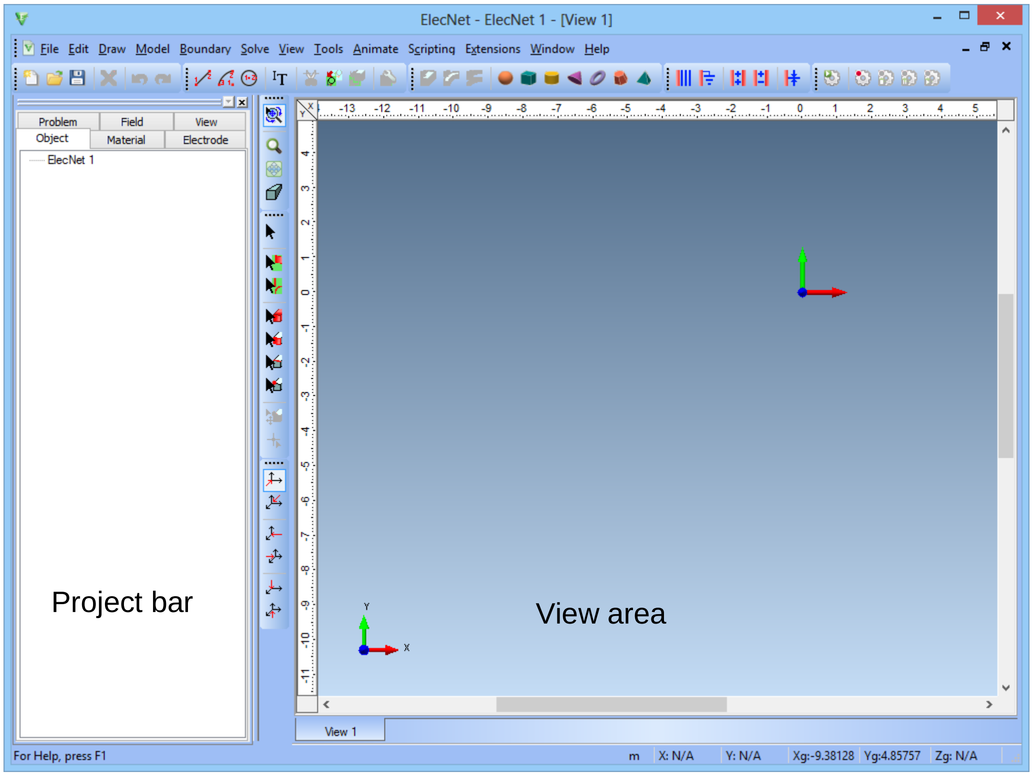
Modeling the system involves the following steps:

- Draw the cross-section of one channel.
- Extend it in a straight line to form a solid body, and specify the material.
- In the same way, construct the second channel.
- Specify the electrodes.
- Define the bounded region of the problem as an *air box* within which the field will be calculated.
- Instruct the program 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 you are familiar with Microsoft Windows, that you have installed ElecNet, and started the application (usually by double-clicking the ElecNet icon). The ElecNet Main window, shown below, should be visible. This is the default window, which can be customized by experienced users.



1 Examine the ElecNet 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 View1 window is displayed.
- Between the Project bar and the View window 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 ElecNet, the default units of length are meters. It is more convenient to work with dimensions in millimeters for this tutorial. Proceed as follows to change the default units and grid settings.

Initial settings

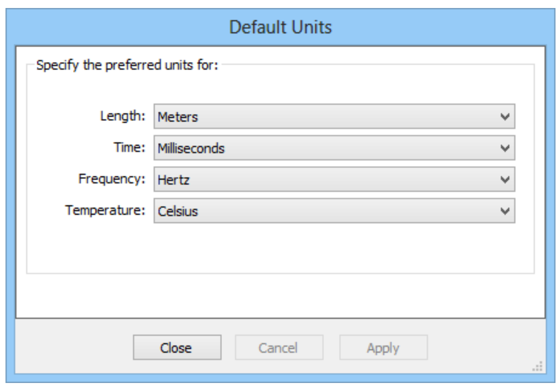
1 On the File menu, click Save. Alternatively, click the Save button. 

- Browse to a suitable folder for storing the model.

2 Save the model with the name **Channels.**

The model name in the Object page should change to Channels. The extension will be .en in the full version of ElecNet, or .ente in the Trial Edition.

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

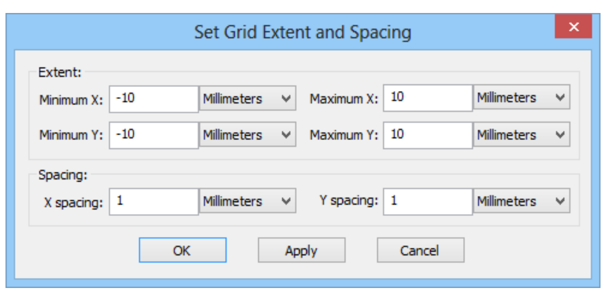


4 Click the Length drop-down list.

- 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:

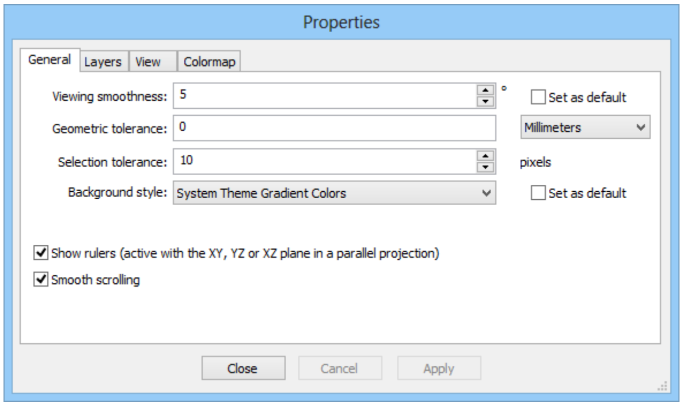
- Minimum X: **–40** Maximum X: **40**
- Minimum Y: **–40** Maximum Y: **0**
- X spacing: **5** Y spacing: **5**

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 outer channel

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 the vertical toolbar, as shown in the instructions, or selected from the Draw menu. The device cross-section, with dimensions in millimeters, is shown below.

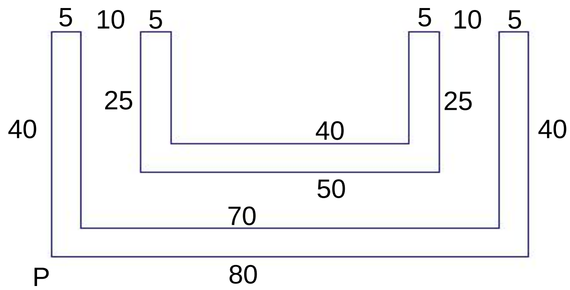
In ElecNet, all drawing takes place on a 2D *construction slice*. For 2D models, this is just the XY plane, but for 3D models, the construction slice can be moved to other planes.

1 Click the Add Line button. 

2 Draw the outline of the outer channel as described below, 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, –40) for point P.
- Move the pointer near (40, –40) for the other end of the line, and click.
- Move the pointer near (40, 0) for the end of the next line, and click.
- Continue in this way for the rest of the shape.

Do not draw the lines for the inner channel at this stage. See below for the method of deleting any lines drawn in the wrong place.



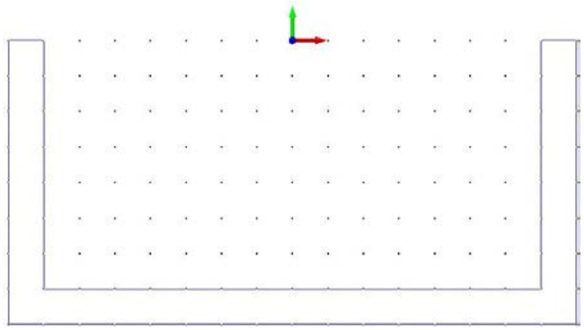
3 If any lines have been drawn in the wrong place, correct them as follows.

- 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 To terminate line drawing, use a double click to complete the last line, or press Esc when the drawing is complete.

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



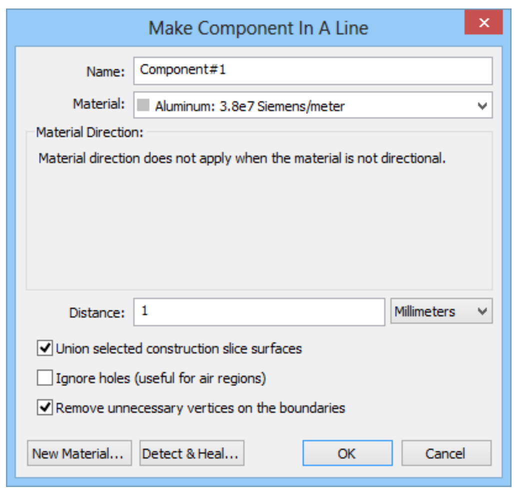
Completing the outer channel

1 Click the Select Construction Slice Surfaces button. 

2 Click anywhere inside the channel.

The interior of the channel 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 **Outer**.

5 Click in the Material box. Start typing **Aluminum: 3.8e7 Siemens/meter**

- When you type **3.8e7** 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 **160**.

7 Click OK.

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

Making the Inner channel

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

- Make sure that that there is a 10 mm gap between the two channels.

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

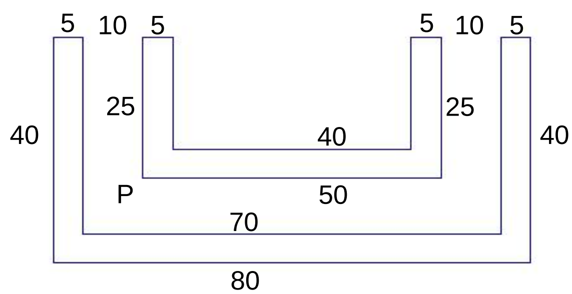
3 Click the Make Component in a Line button. 

The Distance should be shown as 160 mm, and the Material as Aluminum: 3.8e7 Siemens/meter.

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

4 The Object page of the Project bar should show two components: Outer and Inner (in that order).

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



Defining the electrodes

The next step is to specify that the channel surfaces are equipotentials, and to specify the voltage values. This is accomplished by defining *electrodes*, which can be surfaces of components or entire components. In this example, the entire volume of each channel is an equipotential, so there is no electric field inside the channel. However, if the entire channel is defined as an electrode, ElecNet excludes the channel from the finite-element mesh, and it will not find the force on the channel. Since the forces are required, the electrodes will be defined in terms of the channel surfaces, which are termed *faces* in ElecNet.

1. In the Object page, click the + box next to Outer.

   This expands a branch of the tree directory, displaying the list of faces (surfaces) of the Outer component.

2. Click Face#1 (Start Face).
3. Hold down the Shift key and click Face#10

   This should select all the faces from Face#1 to Face#10, which comprise the surfaces of the component. The surfaces should be highlighted in the View window.

4. On the Model menu, click Make Electrode.

   Electrode#1 should appear in the Object page.

5. In the same way, make the electrode for the inner channel.

   Electrode#2 should appear in the Object page.

6. Select the Electrode page of the Project bar, by clicking the Electrode tab.

   Electrode details should be displayed.

7. In the list for Electrode#2, click on 0 V rms
   - Click a second time, or press F2.
     The display should change to an edit box displaying the number 0.
   - Change 0 to **10000** and press Enter.

     The display should show 10000 V rms

8. 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 electrodes 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 is 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 electric field is constrained to be parallel to this boundary. This is a reasonable approximation if the boundary is sufficiently far away from the device. It is usually satisfactory to make the radius of the air box about 10 times the radius of the rest of the model.

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

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.

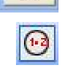
All the lines are marked in red.

- 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 (Center, Radius) 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 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 ElecNet. Do not select Virtual Air, which is a generic insulator material.

- The Distance should be 160 mm.
- Click OK.

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

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

Viewing the model

To view the channels 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 channels.
- Hold down the Ctrl key and the left mouse button, and drag a rectangle enclosing the channels.
- 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 13 for adding the Dynamic Zoom button.

Solving the model

ElecNet uses the finite-element method of solving the electric 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 ElecNet 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 ElecNet 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. The tab for this window is labeled Results.

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

Viewing the equipotential plot

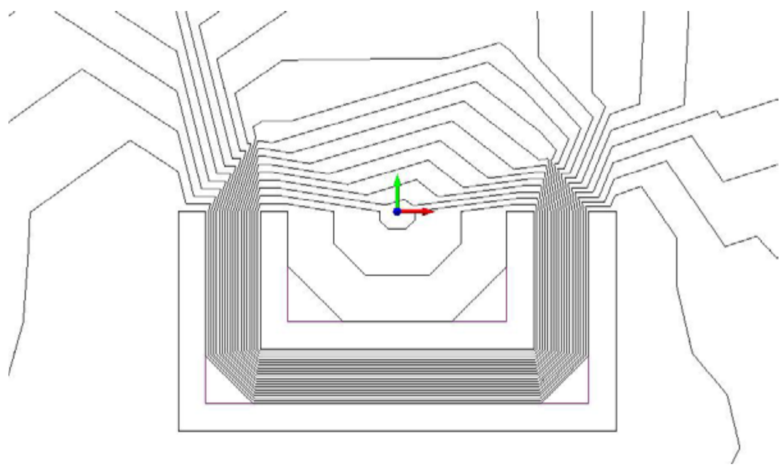
1 Select the Field page of the Project bar by clicking the Field tab.

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

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

3 Click Update View.

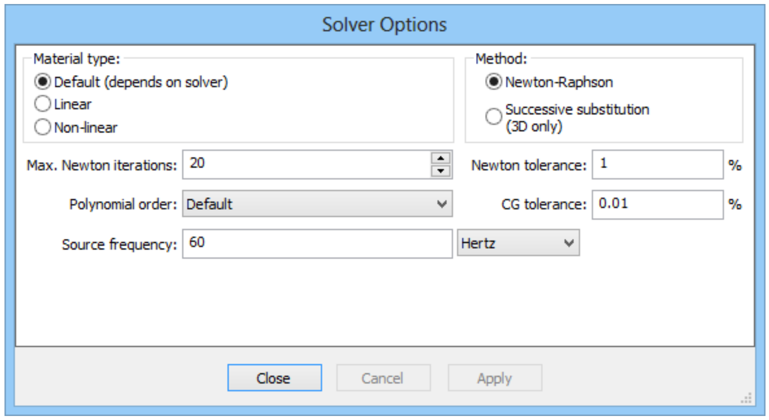
   The window should show contours of constant V – an equipotential plot. It should be similar to the plot below. The plot is not smooth because the solution is not very accurate at this stage.



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:



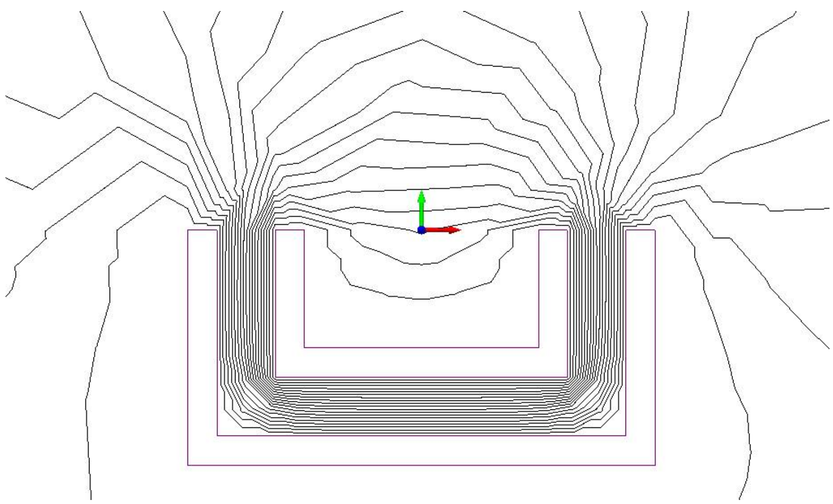
The Polynomial Order is shown as Default. For this model, the default is order 1.

- 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 equipotential lines should be smoother, as shown below, indicating a more accurate solution.

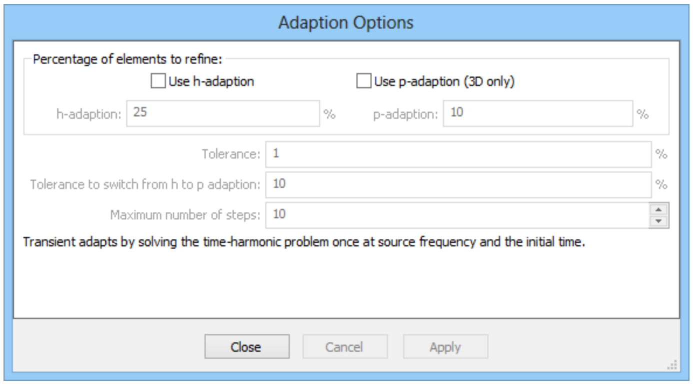
3 Save the model again.



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. ElecNet 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**.*

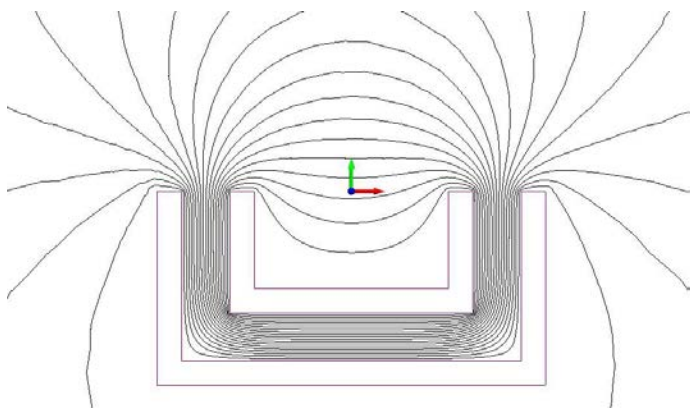
- In the Tolerance box, type **0.5**

At each adaption step, ElecNet 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 plot should resemble the diagram below.



4 Examine the change in the mesh as follows.

- On the View menu, click Initial 2D Mesh.

  This should show the original mesh.

- On the View menu, click 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*. ElecNet has a Results window that displays numerical values for quantities such as force and charge; these are termed global quantities. 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 electric field strength values

1 In the Field page of the Project bar:

- Click the Shaded tab.
- Click |E| smoothed.
- Click Update View.

This should show a color map of the electric field magnitude, superimposed on the equipotential plot.

2 On a vertical toolbar, click the Field Probe 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 voltage, and |E| Smoothed.

  These numbers are the values at the position of the mouse 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 voltage and |E| 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 electric field in this device. First, the electric field is not confined to the region between the channels, but spreads into the surrounding air; this is termed *fringing*. Secondly, the electric field magnitude is very high near the convex corners at the bottom of the inner channel and at the tops of both channels. These corner points are the locations of singularities where, theoretically, the magnitude of the electric field is infinite.

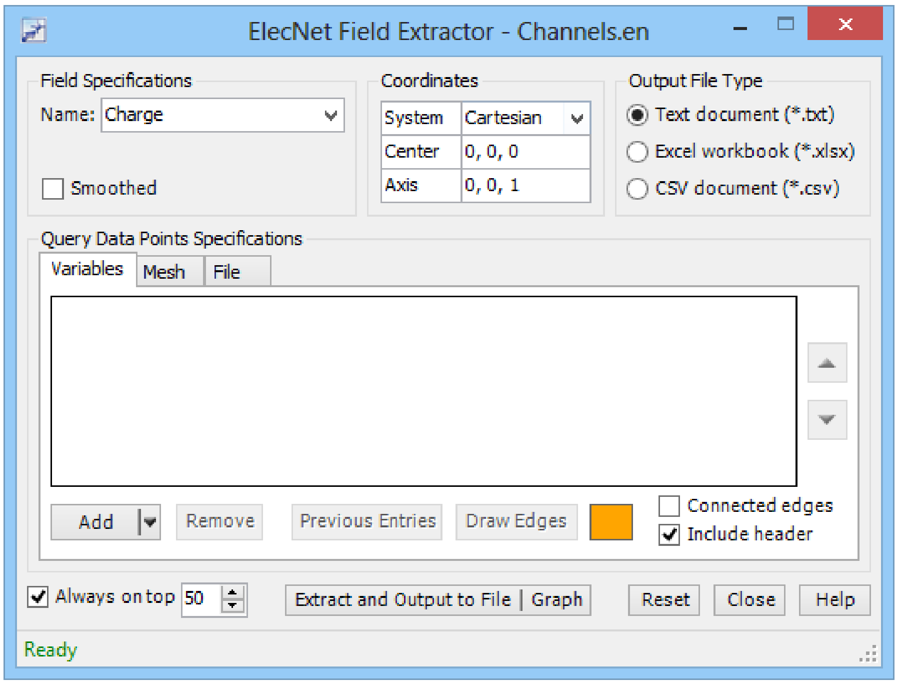
Graphs of electric field strength

ElecNet 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 a graph of the electric field strength magnitude along a line just below the inner channel, with end-point coordinates (–35, –25.5) and (35, –25.5). This will show the high field values near the corners of the channel.

1 On the Tools menu, click Field Extractor to display a dialog:



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	-35, 35	141
Y - End	-25.5	1

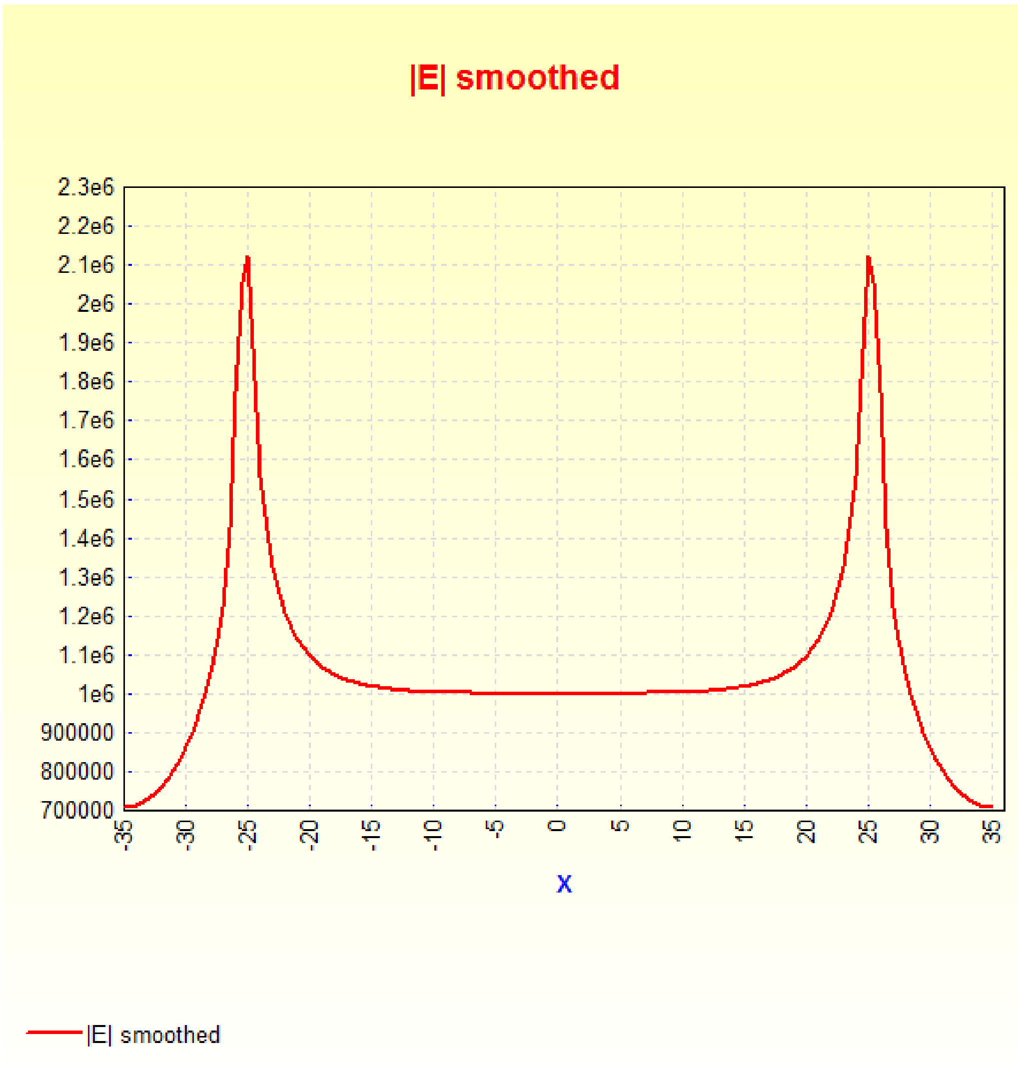
For Y, entering only a Start value automatically sets the number of sampling points to 1.

Since the X coordinate range is 70 mm, setting the iteration value to 141 will sample the field at increments of 0.5 mm.

4 In the Field Name drop-down list, select |E| smoothed.

5 Click Graph.

6 Close the Field Extractor window.



Global quantities

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

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 5.

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 Electric Energy 0.00091326 J

See appendix B for a discussion of energy.

Charge and capacitance

In the Results window, click the Charge tab. The displayed value of the charge should be similar to the following:

Electrode#1 –1.8263e–007 coulombs

Electrode#2 1.8263e–007 coulombs

These values are expected to be equal and opposite; the small difference is an indication of numerical error in the field solution. The capacitance of the system may be calculated from the average magnitude of the charge on the electrodes and the voltage between them:

$$\mathrm{C} = \frac{\mathrm{Q}}{\mathrm{V}} = \frac{1.8263 \times 10^{-7}}{10000} = 1.8263 \times 10^{-11}\mathrm{F} = 18.263\ \mathrm{pF}$$

Alternatively, the capacitance may be calculated from the stored electric energy:

$$\mathrm{C} = \frac{2\mathrm{W}}{\mathrm{V}^2} = \frac{2 \times 0.00091326}{(10000)^2} = 1.8265 \times 10^{-11}\mathrm{F} = 18.265\ \mathrm{pF}$$

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

Forces

ElecNet automatically calculates forces 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 inner and the outer channel components.

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
Outer	4.1101e-005	0.041537	0	0.041537
Inner	-1.5676e-005	–0.041488	0	0.041488

- Since this is a 2D model, the Z components are zero.
- By symmetry, the X components should also be zero. The non-zero values are an indication of numerical error in the solution, but they are small in comparison with the Y values.
- The Y components should be nearly equal and opposite. The small difference in magnitude is an indication of numerical error in the solution. Theoretically, there is a small difference between the magnitudes of the force values corresponding to a residual force on the outer boundary, but this is extremely small.

The signs indicate a downward force on the inner conductor, and an upward force on the outer conductor. This signifies a force of attraction, as expected from the opposite signs of the charges.

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.

Solution accuracy

There are small discrepancies between the force values on the two components, which indicate errors in the numerical solution. To reduce the errors, the solution can be repeated after making the following change:

- In the Solver Options dialog, reduce the CG tolerance from 0.01% to 0.001%.

Reducing the CG tolerance will reduce the discrepancy between the magnitudes of the charges on the two electrodes. It is generally advisable to make this change from the default CG tolerance of 0.01%. A value of 0.001% is used for the rest of this tutorial.

Note that reducing the CG tolerance does not eliminate the discrepancy between the magnitudes of the force values. The problem here is the singularity in the value of E at each external corner, which affects the accuracy of the force calculation. To some extent, the problem can be overcome by reducing the adaption tolerance, but this creates a large number of mesh elements. A better solution is to surround each channel with a thin layer of Virtual Air – a material that behaves as air for the field solution, but it is treated as part of the channel for determining the forces; see Appendix B. However, in the present case, the force discrepancy is small, so the Virtual Air shell is not required.

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 ElecNet will create a corresponding set of problems. For the parallel channels, it is useful to vary the relative position in this way. If the full version of ElecNet is licensed for parameterization, the set of problems will be solved automatically. This feature is disabled in the Trial Edition of ElecNet.

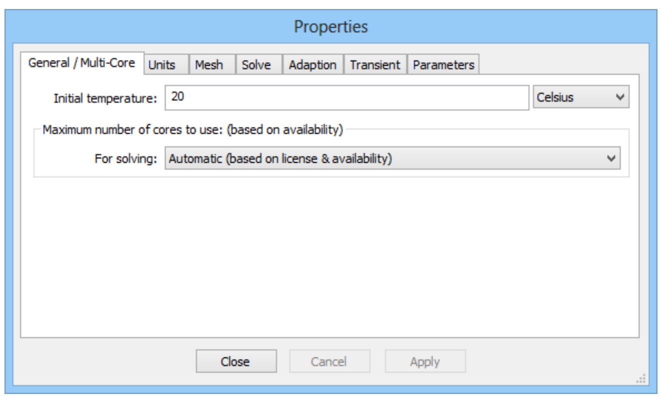
Moving the inner channel

If the inner channel is displaced from its symmetrical position, there will be a lateral force that varies with the displacement. To examine this effect, a new user-defined parameter will be used to move the armature component relative to the rest of the model.

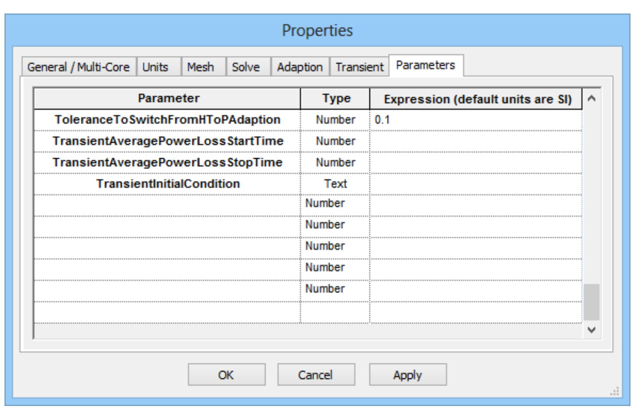
In the Model Properties dialog, there is a Parameters page where the user can create a new named parameter and give it a range of values. The new parameter can be used to modify other properties of model components, so that they depend on one parameter. In this case, a user-defined parameter will be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

Creating a user-defined parameter

1 In the Object page, right-click the model name Channels, and select Properties to display a dialog:



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



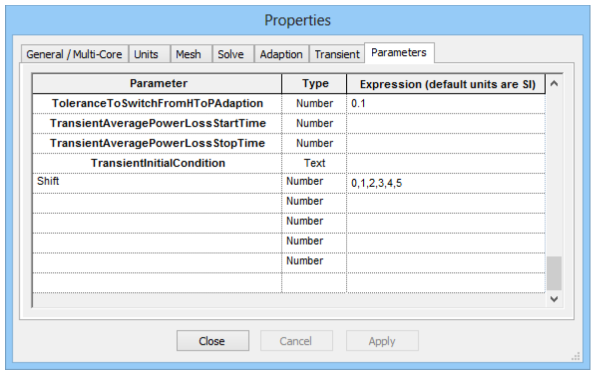
3 In the first vacant Parameter field, type **Shift**.

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

4 In the Expression field, type 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.

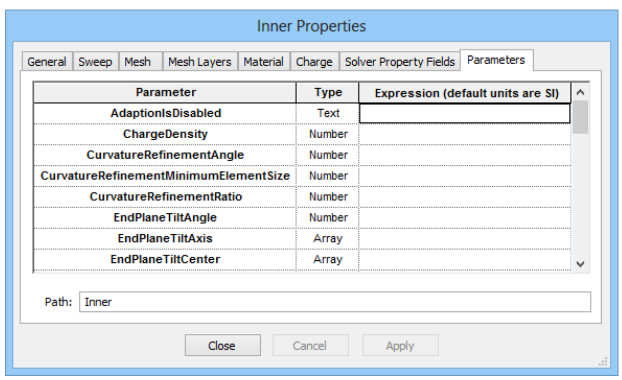


6 Click Apply to change the properties.

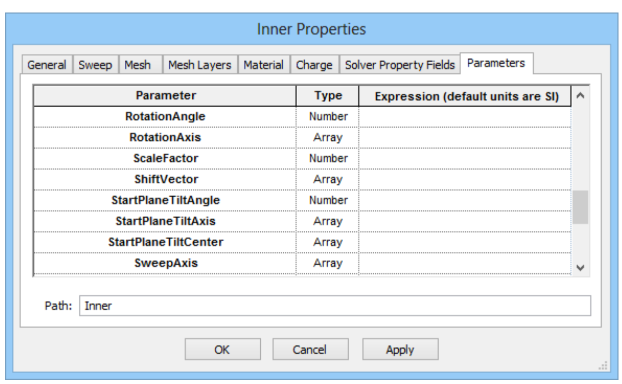
This leaves the Properties dialog open.

Parameterizing the model

1 In the Object page, click Inner. The dialog should change:



2 Scroll down the list of parameters to find ShiftVector:



3 In the Expression field, type the following array for the vector, including the square brackets:
[%Shift%mm, 0, 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.

4 Press Enter to accept the array.

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 Select the Problem page of the Project bar.

- Observe that six problems have been created, each with a different shift vector for the inner channel.
- If ElecNet is licensed for parameterization, all of the problems will be marked for solution, otherwise only the first problem will be marked.

7 On the Solve menu, click Static 2D.

- If ElecNet is not licensed for parameterization, only the first problem will be solved automatically. In this case, the others can be solved manually as follows:
- Open the Parameters page of Model Properties, and delete the first item from the Shift parameter list, so that the list becomes **1, 2, 3, 4, 5.** Click Close.
- On the Solve menu, click Static 2D again. This solves for a displacement of 1 mm.
- Repeat this process for displacements of 2, 3, 4 and 5 mm.
- For each of the solutions in turn, inspect the solution as described below.

The instructions below for post-processing are applicable when ElecNet is licensed for parameterization. With 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.

Post-processing – 1

1 Select the Field page of the Project bar.

Notice that the Problem ID number is set to 1.

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

3 Click the Problem ID drop-down list on the Field page, and select 2, corresponding to a displacement of 1 mm.

4 Click Update View to display the plots for problem 2.

5 Use the Field Probe as before to explore the field magnitude values in different parts of the model.

6 In the Field page, click Animate.

ElecNet generates an image of the plot for each problem ID, and then displays them in sequence in the View window, with a tab labeled Untitled Animation 1.

7 After viewing the animation, close the window by clicking its Close box.

- 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.

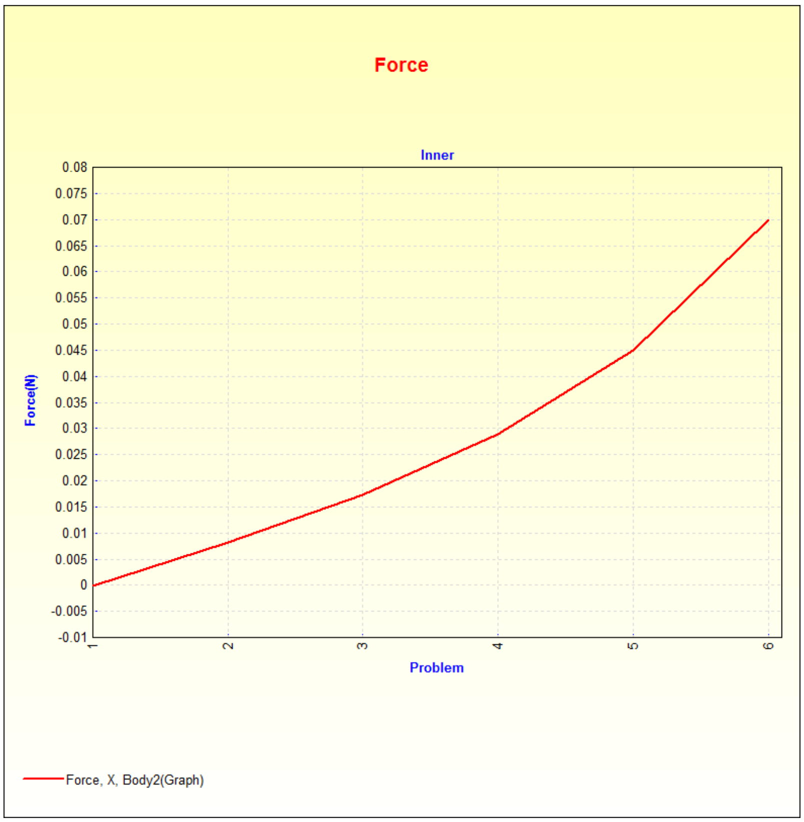
4 Similarly, observe the force values for the other problems.

5 Display a graph of force values as follows.

- For the Inner channel, click in the text box for the X component of force.

  The Graph Selection button should be enabled.

- Click the Graph Selection button to display a graph:

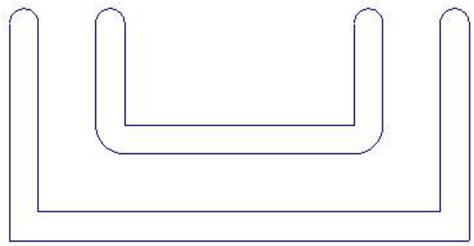


Note that the displacement values are not displayed in this graph, because ElecNet uses this method to show the variation of force with any parameter. The displacement values corresponding to the problem numbers 1 to 6 are 0, 1, 2, 3, 4 and 5 mm respectively.

6 Close the graph window.

Changing the geometry

The field plots have shown that the electric field magnitude is very high in regions close to convex corners. This effect can be reduced if the corners are rounded, as shown in the diagram below. The model will be generated by modifying the construction-slice shapes for the original model and making new components. Proceed as follows.



1 On the file menu, click Save As.

- Save the model as **Channels modified**

2 On the View menu, select Solid Model.

3 Select the Problem page of the Project bar.

- Click Problem 1.
- Click Update View.

4 Select the Object page of the Project bar, then select both channels as follows:

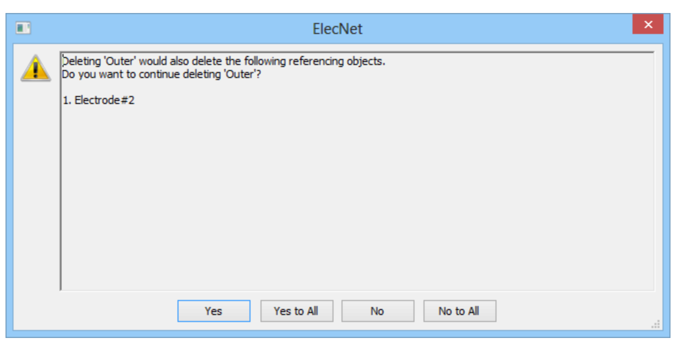
- Click Outer
- Hold down the Ctrl key.
- Click Inner.

5 On the Draw menu, click Extract Edges

This creates lines on the construction slice corresponding to the edges of the components.

6 Delete the channels as follows:

- Select both channels in the Object page.
- Press Delete; this should display a dialog:



- Click Yes to All.

The channels and electrodes should be deleted, but the construction slice lines should be visible.

7 On the View menu, click Set Construction Grid.

- Set the X Spacing and the Y Spacing to 2.5 mm.
- Click OK.

8 On the View menu, click Construction Grid.

9 Click the Add Arc (Center, Start, End) button.

The Status bar at the bottom of the window should show:
Specify the center point, the start point and the end point…

10 Add a semicircular arc to the top of each flange as follows:

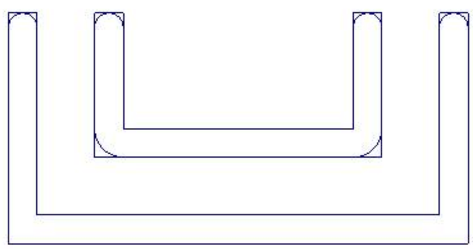
- Click a point 2.5 mm below the centre of the horizontal top line for the center of the arc.
- Click the right-hand vertical line, 2.5 mm below the top.
- Click the left-hand vertical line, 2.5 mm below the top.

The order is important. If you click the left-hand line before the right-hand line, the arc will be drawn the wrong way.

11 Add quadrant arcs to the bottom corners of the inner channel as follows:

- Click the inner corner of the channel.
- Click successive points on the outside of the channel that will form the start and end points of the arc, noting that the arc will be drawn counter-clockwise from the start point to the end point.

The outline of the channels should resemble the following:



12 Select all the lines and arcs:

- Click the Select Construction Slice Lines/Arcs button.
- Press Ctrl+A.

All nine lines and arcs should be marked in red.

13 On the Draw menu, click Segment Edges

This will divide the lines into segments at the points where they touch the arcs.

14 Select and delete the unwanted line segments next to all of the arcs.

15 Make components for the inner and outer channel as before.

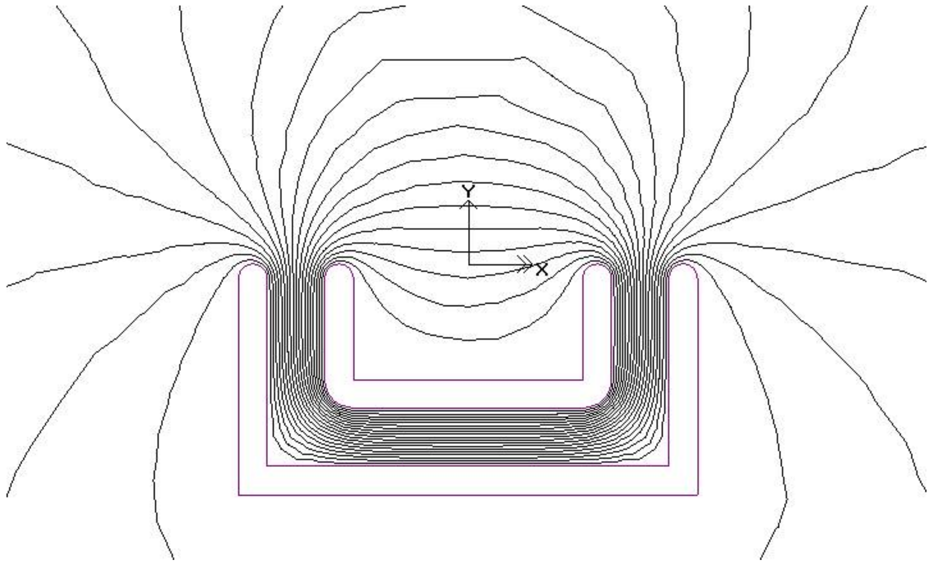
16 Make electrodes for the inner and outer channel surfaces as before, and set the voltage of the inner channel to 10000 V.

17 Modify the ShiftVector parameter of the Inner component as before.

18 On the Solve menu, click Static 2D.

19 View the solution results as before.

The equipotential plot for problem 1 should resemble the following:



With the inner channel in a symmetrical position (problem 1), the maximum value of |E| smoothed is 1.663 MV/m. This is the maximum value displayed in the color legend of the shaded plot. The corresponding value for the original model is 2.512 MV/m, so it appears that the modification has made a small reduction to the maximum field value. In reality, the reduction is much greater than this. Theoretically, the maximum value of |E| in the original model is infinite at the external corners. If the mesh size in the original model is reduced, the computed maximum value of |E| will increase, apparently without limit. However, if the external corners are given a small radius of 0.5 mm, the value of |E| does not increase. These effects can be tested by progressively reducing the adaption tolerance; the table below shows results obtained with ElecNet version 7.5.

	Original			**Modified**			**Corner rad. 0.5 mm**		
Adaption tolerance, %	0.5	0.05	0.005	0.5	0.05	0.005	0.5	0.05	0.005
Max. |E| smoothed, MV/m	2.387	3.149	6.262	1.647	1.647	1.623	3.563	3.587	3.600

Postscript

The table below compares the 2D results for the modified channels with the results from a 3D model, using ElecNet version 7.5.

	2D	**3D**
Stored electric energy	0.0008706 J	0.0009524 J
Charge	1.741×10^{-7} C	1.905×10^{-7} C
Capacitance	17.41 pF	19.05 pF
Force on inner channel	0.03948 N	0.04044 N

For the 3D model, the enclosing air box is a cuboid with dimensions 800 mm × 800 mm × 960 mm; the solution parameters are the same as for the 2D model, except that the h-adaption percentage is reduced from 25% to 5%.

The 3D solution gives a capacitance value that is 9.4% higher than the 2D result, whereas the force value is only 2.6% higher. An explanation for this difference is that the capacitance per unit area depends on E, whereas the force per unit area depends on E^2. In the fringing field beyond the ends of the channels, the value of E will diminish more slowly with distance than does the value of E^2. Consequently, this fringing field, which is ignored in a 2D solution, will have a greater effect on the capacitance than it has on the force.

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 ElecNet, so it is advisable to work through them in sequence. The detailed descriptions of basic ElecNet operations given in chapter 2 will not be repeated, but any new operations will be fully explained.

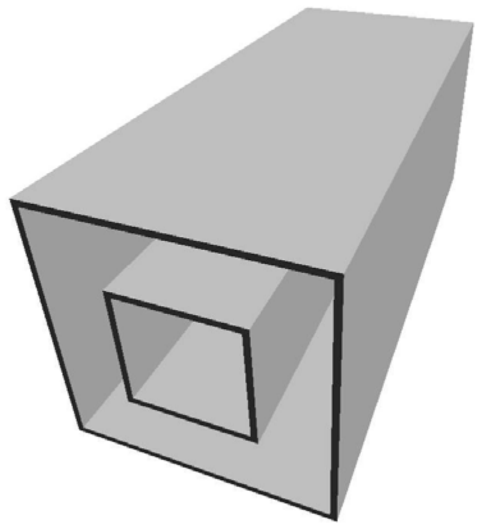
For all of the case studies in chapter 3 and chapter 4, the instructions assume that a new model is being started, as described in the tutorial in chapter 2. To avoid tedious repetition, this instruction is given in abbreviated form in the case studies.

In all modeling studies, it is necessary to ensure that an accurate solution has been obtained. There are two principal sources of inaccuracy: approximating an open boundary by a boundary at a finite distance from the device, and numerical errors in the finite-element solution. Errors introduced by the open-boundary approximation will be examined in some of the case studies. Errors in the numerical solution can be minimized by adjusting the solution tolerance values as described in chapter 2, using the value of the conduction current in place of charge for current-flow models.

In some models, there is only one electrode, so it is not possible to check the accuracy by comparing values that should be equal. In these cases, the value of the charge or conduction current can be monitored and the tolerance values altered until the change in the value is acceptably small. The tolerance values given in the case studies have been determined in this way. In cases where ElecNet does not calculate forces automatically from a simple model of the material parts, results can be obtained by adding layers of the material Virtual Air: see Appendix B.

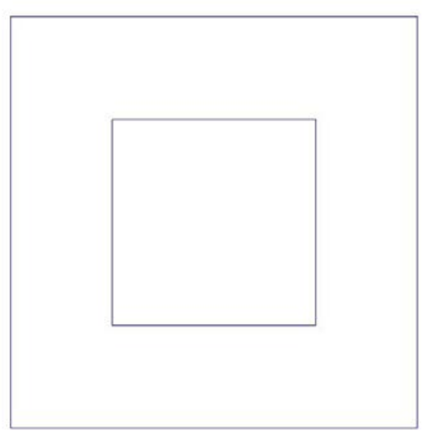
Coaxial square conductors – 1

The diagram below shows two long coaxial square conductors. The objectives are (a) to examine the field, and (b) to determine the capacitance of the system per meter length and compare the value obtained from ElecNet with a known result obtained by analytical solution of the field equations.



Modeling the device

Since the device is long, a 2D model will be an accurate representation. Unlike the case study, the electric field is confined to the region between the two conductors, so it is not necessary to add an artificial outer boundary. Since the conductors are equipotentials in this electrostatic problem, they do not need to be modeled. All that is required is a square-section tube representing the air region between the conductors, with the inner and outer surfaces defined as electrodes. The cross section of the model is shown below.



The inner and outer squares have side lengths of 20 mm and 40 mm respectively.

Creating the model

1 Start a new model and save it as **Coaxial square full**.

2 Set the model length units to millimeters.

3 Set and display the construction grid.

4 Draw the two squares, with the coordinate origin at the center.

5 Construct a component for the air region:

- Sweep distance: 1000 mm.
- Material: AIR.

6 Turn off the construction grid.

7 Make the outer electrode:

- In the Object page, click the + sign next to the component name to display the faces.
- Click the faces in turn, and observe which lines are highlighted in the View window.
- Identify the four faces that form the outer square.
- Click the first of these faces.
- Hold down the Ctrl key and click the remaining three faces in turn.

All four sides of the outer square should be highlighted in the View window.

- On the Model menu, click Make Electrode.

8 In the same way, make the inner electrode, using the four surfaces that form the inner square.

9 In the Electrode page, set the voltage of the inner electrode (Electrode#2) to 1 V.

Solving and post-processing

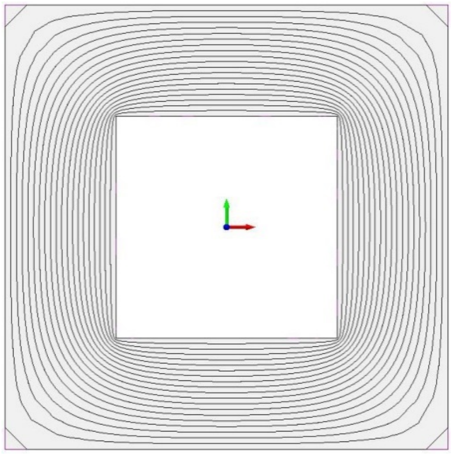
1 Set the options for solving, with the following changes from the default settings:

- Solver Options:
  CG tolerance 0.001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.002%.

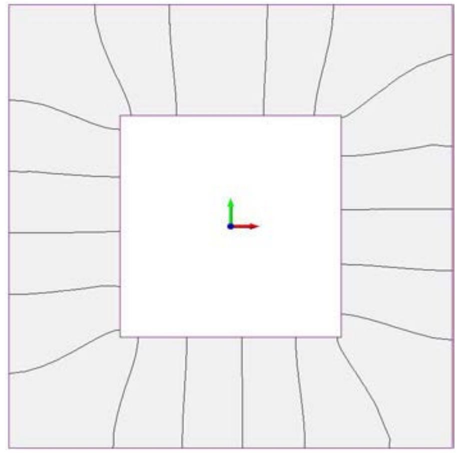
2 Solve as Static 2D.

3 Inspect the contour plots of V and the flux function.

The contour plot of V displays equipotentials, and the contour plot of the flux function displays field lines, as shown below.



Contour plot of V



Contour plot of flux function

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

- From the charge: $C = Q / V$, where Q is the average of the magnitude of the charge on each electrode, and V is the voltage difference (1 V in this case).
- From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

The results below were obtained with ElecNet version 7.5. For comparison, the analytical value of the capacitance per meter length [1, 2] is 90.61 pF.

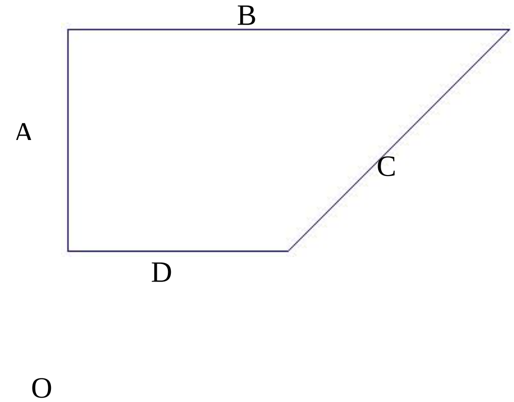
Stored electric energy:	4.5308×10^{-11} J
Charge on outer electrode:	-9.0618×10^{-11} C
Charge on inner electrode:	9.0613×10^{-11} C
Capacitance from charge:	90.62 pF
Capacitance from stored energy:	90.62 pF

Coaxial square conductors – 2

The model used in the previous case study does not exploit the symmetry of the structure. Consequently, the finite-element mesh is unnecessarily large, with a correspondingly long solution time. It is advantageous to take advantage of symmetry to model only a portion of the device.

Modeling the device

It is only necessary to model one eighth of the device, shown below.



Surfaces represented by lines B and D are the electrodes; surfaces represented by lines A and C are planes of symmetry, which can be represented by the default Flux Tangential boundary condition. Lines A and D are 10 mm long; line B is 20 mm.

Creating the model

1 Start a new model and save it as **Coaxial square 8th**.

2 Set the model length units to millimeters.

3 Set and display the construction grid, with the minimum X and Y values set to zero.

4 Draw the outline of the quarter model, with the coordinate origin (point O in the diagram above) at the center of the complete square.

5 Construct a component for the air region:

- Sweep distance: 1000 mm.
- Material: AIR.

6 Turn off the construction grid.

7 Make the outer electrode:

- In the Object page, click the + sign next to the component name to display the faces.
- Identify and select the face for the outer electrode (corresponding to B).
- On the Model menu, click Make Electrode.

8 In the same way, make the inner electrode, using the face corresponding to D in the diagram.

9 In the Electrode page, set the voltage of the inner electrode (Electrode#2) to 1 V.

Solving and post-processing

1 Set the options for solving, with the following changes from the default settings:

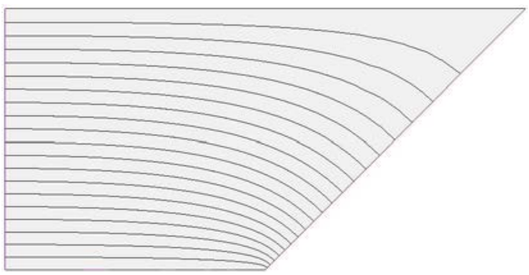
- Solver Options:
  CG tolerance 0.001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.02%.

For similar accuracy, a larger adaption tolerance is used for this model than for the full model.

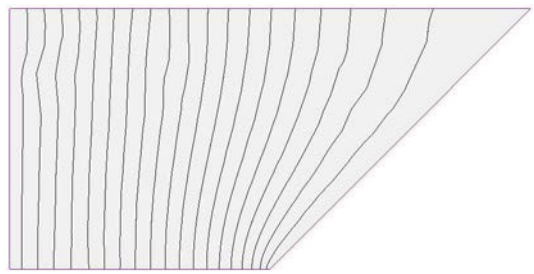
2 Solve as Static 2D.

3 Inspect the contour plots of V and the flux function.

The contour plot of V displays equipotentials, and the contour plot of the flux function displays field lines, as shown below.



Contour plot of V



Contour plot of flux function

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

- From the charge: $C = Q / V$, where Q is the average of the magnitude of the charge on each electrode, and V is the voltage difference (1 V in this case).
- From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

This value is one eighth of the capacitance of the complete model.

Sample results

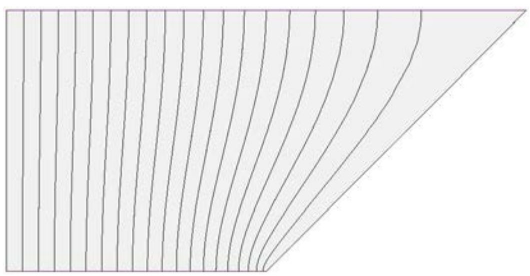
The results below were obtained with ElecNet version 7.5. For comparison, the analytical value of the capacitance per meter length [1, 2] is 90.61 / 8 = 11.33 pF.

Stored electric energy:	5.6641×10^{-12} J
Charge on outer electrode:	-1.1328×10^{-11} C
Charge on inner electrode:	1.1328×10^{-11} C
Capacitance from charge:	11.33 pF
Capacitance from stored energy:	11.33 pF

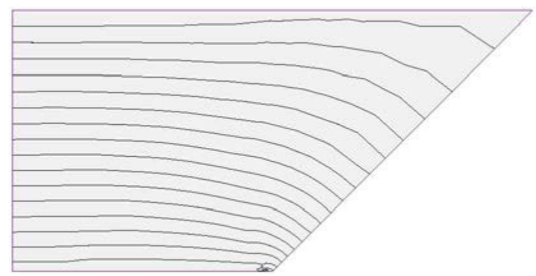
The capacitance of the complete device is 8 × 11.33= 90.64 pF, which is close to the value of 90.62 pF from the full model.

Duality

It is instructive to interchange the electrode and symmetry boundaries in the model. Thus, the faces represented by lines A and C become the electrodes, and the other faces have the default Flux Tangential boundary condition. The result is the dual of the original quarter model, where field lines and equipotentials are interchanged. The diagrams below shows the contour plots for the dual model. Observe that the contour plot of V gives a better plot of field lines than does the contour plot of the flux function in the original model; similarly, the contour plot of V in the original model gives a better plot of equipotentials than does the contour plot of the flux function in the dual model.



Contour plot of V

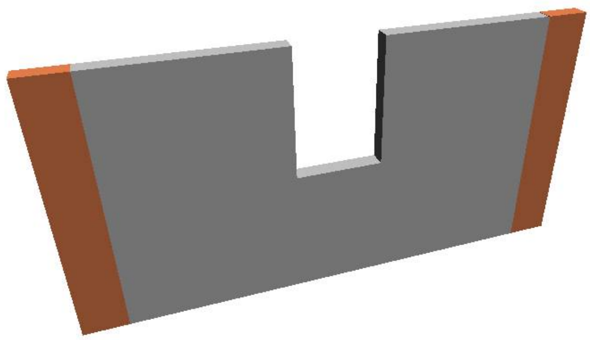


Contour plot of flux function

This technique gives an alternative method of displaying the field lines in ElecNet for any model where it is possible to interchange the electrodes and Flux Tangential boundaries. In electrostatic problems, this generally requires the device to have some form of symmetry so that the model represents only part of the device.

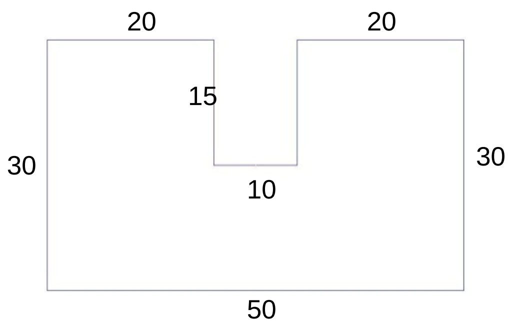
Trimmed resistor

The diagram below shows a resistor made from a flat sheet of resistive material, with electrodes at the ends. The value of the resistance is adjusted by making a cut in the material. It is required to determine the resulting value of the resistance, by finding the current that flows when the voltage difference between the electrodes is 1 V.

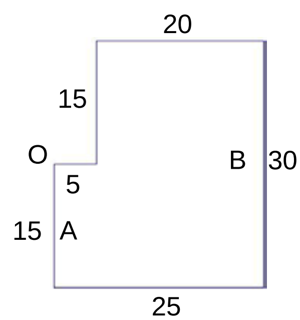


Modeling the device

This is another example of a problem that is well represented by a 2D model. It is a current-flow problem where the field is confined to the region of the resistive material, so it is not necessary to add an artificial outer boundary. The electrodes at the ends do not need to be modeled; the end faces of the resistive material just need to be defined as electrodes. Dimensions in mm of the cross section of the full model are shown below; the depth in the third dimension is 2 mm, and the material is nichrome with a resistivity of 110×10^{-8} Ωm. In this case, because of the symmetry of the device, a half model can be used.



Full model



Half model

The material nichrome does not exist in the standard material database supplied with ElecNet. If desired, a new material can be defined, based on an existing material such as Aluminum: 3.8e7 Siemens/meter. See the ElecNet Help for the method of doing this. A simpler method, which is used in this case study, is to use an existing material and scale the result.

Creating the model

1. Start a new model and save it as **Trimmed resistor half**.
2. Set the model length units to millimeters.
3. Set and display the construction grid.
4. Draw the outline of the half model, with the coordinate origin at corner O.

5 Construct a component for the resistor:
- Sweep distance: 2 mm.
- Material: Aluminum: 3.8e7 Siemens/meter.

6 Make the left-hand electrode:
- In the Object page, click the + sign next to the component name to display the faces.
- Identify and select the left-hand face corresponding to line A in the diagram.
- On the Model menu, click Make Electrode.

7 In the same way, make the electrode for the right-hand face corresponding to B.

8 In the Electrode page, set the voltage of the right-hand electrode (Electrode#2) to 0.5 V.

This corresponds to a voltage difference of 1 V between the electrodes of the full model.

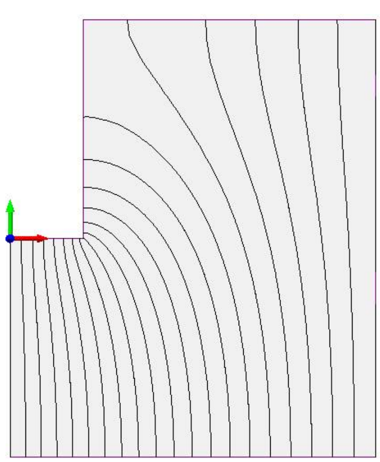
Solving and post-processing

1 Set the options for solving, with the following changes from the default settings:
- Solver Options:
  CG tolerance 0.0001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.01%.

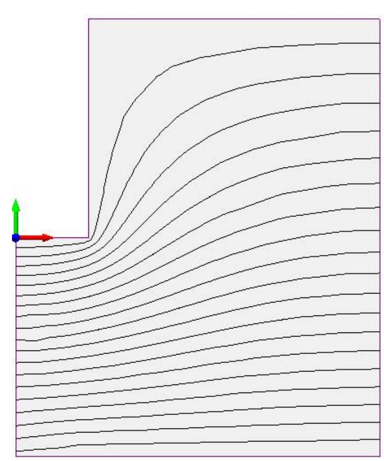
2 Solve as Current Flow 2D.

3 Inspect the contour plots of *V* and the flux function.

The contour plot of V displays equipotentials, and the contour plot of the flux function displays current flow lines, as shown below.



Contour plot of V



Contour plot of flux function

4 Inspect the computed global quantities, and calculate the resistance value as follows:

- From the current: $R = V / I$, where I is the average of the magnitude of the conduction current at each electrode, and V is the voltage difference (1 V in this case).
- From the ohmic loss: $R = V^2 / P$, where P is the ohmic loss.

5 Select the Material page of the Object bar.

- Right click on Aluminum: 3.8e7 Siemens/meter and select Properties.
- In the Properties dialog, click the Electric Conductivity tab.
- Note the value of the conductivity at 20ºC.

The value should be 3.8×10^7 S/m

- Close the Properties dialog.

Sample results

The results below were obtained with ElecNet version 7.5.

Ohmic loss:	15304 W
Conduction current at outer electrode:	–30608 A
Conduction current at inner electrode:	30609 A
Resistance from current:	16.34 μΩ
Resistance from ohmic loss:	16.34 μΩ

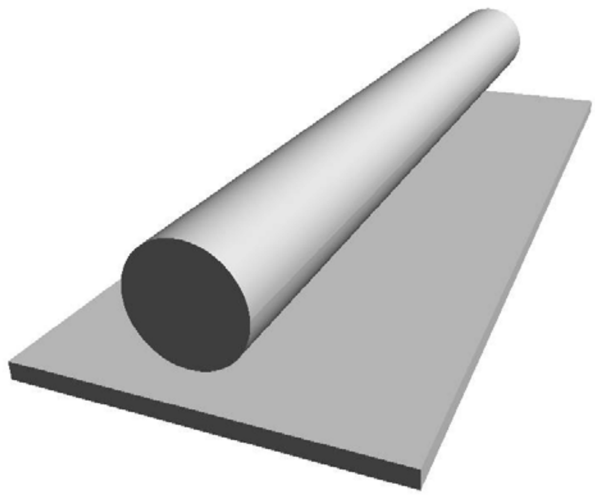
The conductivity of aluminum is 3.8×10^7 S/m, so the resistivity is 2.632×10^{-8} Ωm. If the material is changed from aluminum to nichrome, the resistance from the half model will be:

$$R_{nich} = \frac{\rho_{nich} R_{alum}}{\rho_{alum}} = \frac{110\times10^{-8}\times16.34\times10^{-6}}{2.632\times10^{-8}}\ \Omega = 682.9\ \mu\Omega$$

The resistance of the complete device is thus 2×682.9 μΩ = 1.366 mΩ.

Cylinder above a plane – 1

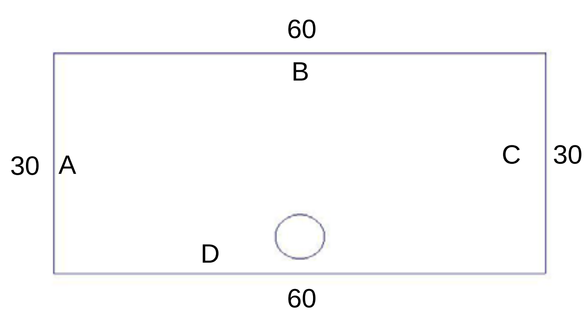
The diagram below shows a long conducting cylinder above a conducting ground plane. The objectives are (a) to examine the field, and (b) to determine the capacitance of the system per meter length and compare the value obtained from ElecNet with a known result obtained by analytical solution of the field equations.



Modeling the device

Since the cylinder is long, a 2D model is a good approximation. The cylinder is 6 mm in diameter, there is a gap of 2 mm between the cylinder and the plane, and the cylinder is maintained at a voltage of 1 kV relative to ground. This is an open-boundary problem, where the electric field extends to infinity. The Kelvin transformation method (Appendix A) is often an effective way of handling an open boundary, but it cannot be used in this case study because the ground plane is of infinite extent. It is therefore necessary to impose an artificial boundary with an air box, as was done in the tutorial of chapter 2. A suitable initial choice for the height of the air box is 10 times the radius of the cylinder, and the width is set equal to twice the height. However, the choice of air box size will affect the accuracy of the capacitance calculation, so the effect of increasing the size will be investigated.

The diagram below shows the cross-section of the complete model, with dimensions in millimeters. The bottom side D of the rectangle is drawn along the X axis, with its mid-point at the origin. The cylinder is drawn with its center at (0, 5), vertically above the mid-point of side D. Sides A, B and C have the default Flux Tangential boundary condition, and side D is assigned the Ground boundary condition. Since we are not interested in the force on the cylinder, this component will be defined as an electrode, so the interior of the cylinder is excluded from the finite-element mesh.



Creating the model

1. Start a new model and save it as **Cylinder plane full**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum Y value set to zero, and a spacing of 1 mm.
4. Draw the circle, and construct a component for the cylinder:
   - Material: Aluminum: 3.8e7 Siemens/meter.
   - Sweep distance: 1000 mm.
5. Select and delete the construction slice arcs.
6. Make an electrode from the cylinder component:
   - Select the entire component in the Object page, instead of a set of faces.
   - Set the voltage to 1000 (1 kV).
7. Draw the rectangle, and construct a component for the air box:
   - Material: AIR.
   - Sweep distance: 1000 mm.
8. Turn off the construction grid.
9. Identify and select the face of the AirBox component that corresponds to line D in the diagram.
10. On the Boundary menu, click Ground.

    BoundaryCondition#1 (G) should appear in the Object page.

Solving and post-processing

1. Set the options for solving, with the following changes from the default settings:
   - Solver Options:
     CG tolerance 0.001%,
     Polynomial order 2.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.1%.
2. Solve as Static 2D.
3. Inspect the contour plot of V and the shaded plot of the smoothed $|\mathbf{E}|$ values.
4. Inspect the computed global quantities, and calculate the capacitance value as follows (see appendix B for details):
   - From the charge: $C = Q / V$, where Q is the charge on Electrode#1, and V is the voltage difference (1 kV in this case).
   - From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

The results below were obtained with ElecNet version 7.5. For comparison, the analytical value of the capacitance per meter length [1] is 50.64 pF.

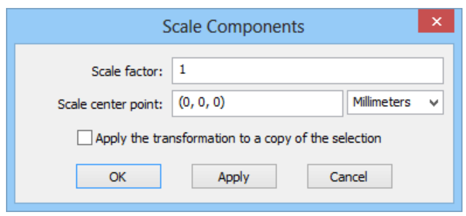
Stored electric energy:	2. 4639×10^{-5} J
Charge on Electrode#1:	4.9299×10^{-8} C
Capacitance from charge:	49.30 pF
Capacitance from stored energy:	49.28 pF
Maximum value of |**E**| smoothed:	610.3 kV/m

Thus, the value of capacitance from the model is 2.6% lower than the analytical value.

Effect of air box size

To explore the effect of the air box size, the dimensions will be doubled and then doubled again.

1. View the solid model.
2. Select and delete the construction slice lines.
3. In the Object page, select the AirBox component.
4. On the Model menu, select Scale Components… to display a dialog:



5. Change the scale factor to 2 and click OK.
6. Solve the model and inspect the results.
7. Repeat steps 3 to 5
8. Reduce the adaption tolerance to 0.05%
9. Solve the model and inspect the results.

The table below shows typical results of this process. For the 120 mm box, the adaption tolerance was reduced from 0.1% to 0.05%.

Air box height (mm):	30	60	120
Adaption tolerance (%)	0.1	0.1	0.05
Stored electric energy (J):	2.4639×10^{-5}	2.5150×10^{-5}	2.5281×10^{-5}
Charge on Electrode#1 (C):	4.9299×10^{-8}	5.0298×10^{-8}	5.0561×10^{-8}
Capacitance from charge (pF):	49.30	50.30	50.56
Capacitance from stored energy (pF):	49.28	50.30	50.56
Maximum value of |**E**| smoothed (kV/mm):	610.3	609.0	609.4
Capacitance error:	2.6%	0.69%	0.16%

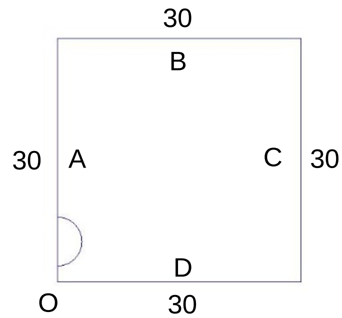
These results show that the rule of making the air box radius 10 times the radius of the device is insufficient in this case for high accuracy in the determination of the capacitance. A factor of 20 is required for an error of less than 1%.

Cylinder above a plane – 2

Because of symmetry, it is only necessary to model half of the system. In complex problems, particularly 3D, this can save a good deal of computing time. In the present example the solution is obtained very quickly anyway, but it is instructive to compare the results from a half model.

Modeling the device

The half model of the device is shown below, with dimensions in millimeters. Point O is the coordinate origin. The radius of the semicircle is 3 mm, and its center is at (0, 5).



The default Flux Tangential boundary condition is correct for the surface represented by side A for the symmetry boundary, so it is only necessary to assign a boundary condition to the surface represented by side D as before.

Creating the model

1. Start a new model and save it as **Cylinder plane half**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum X and Y values set to zero, and a spacing of 1 mm.
4. Draw the square boundary of the half model, with the coordinate origin at O.
5. Draw the semicircle, and make a component for the half cylinder:
   - Material: Aluminum: 3.8e7 Siemens/meter.
   - Sweep distance: 1000 mm.
6. Make an electrode from this component:
   - Select the entire component in the Object page, instead of a set of faces.
   - Set the voltage to 1000 (1 kV).
7. Delete the construction-slice arc for the half cylinder, but leave the lines for the square.
8. Make the air box:
   - Material: AIR.
   - Sweep distance: 1000 mm.
9. Turn off the construction grid.
10. Apply the Ground boundary condition to the face that corresponds to line D in the diagram.

Solving and post-processing

1 Set the options for solving, with the following changes from the default settings:

- Solver Options:
  CG tolerance 0.001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

3 Inspect the contour plot of V and the shaded plot of the smoothed $|\mathbf{E}|$ values.

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

- From the charge: $C = Q / V$, where Q is the charge on Electrode#1, and V is the voltage difference (1 kV in this case).
- From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

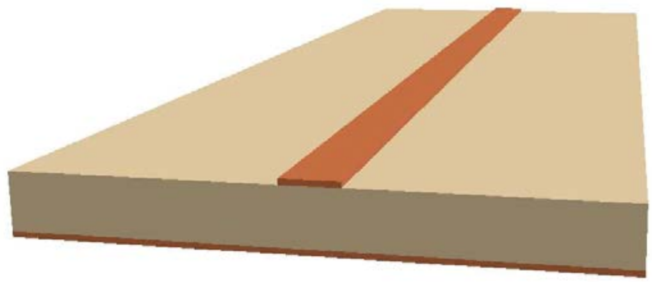
The results below were obtained with ElecNet version 7.5.

Stored electric energy:	1.2321×10^{-5} J
Charge on Electrode#1:	2.4614×10^{-8} C
Capacitance from charge:	24.61 pF
Capacitance from stored energy:	24.64 pF
Maximum value of $|\mathbf{E}|$ smoothed:	609.3 kV/m

As expected, the capacitance values are half of those obtained with the full model, and the maximum value of $|\mathbf{E}|$ is virtually unchanged.

Single PCB track

A printed-circuit board (PCB) has a copper sheet on one side and a copper track on the other side, as shown in the diagram below. The problem is to determine the capacitance per meter length between the track and the copper sheet.

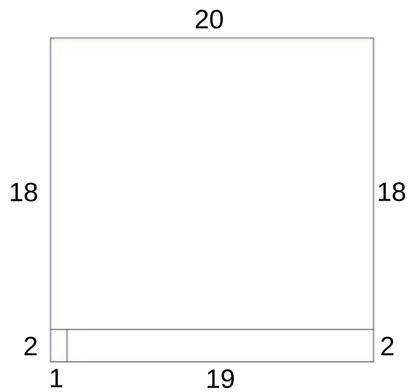


Modeling the device

The copper track is 2 mm wide, the board is 2 mm thick, and the relative permittivity of the board material is 5.0. Both the copper sheet and the copper track are very thin, so it is not necessary to model these parts as solid objects. Instead, the component representing the PCB is divided into three regions: one region underneath the track, and adjoining regions on each side. The upper surface of the track region will be specified as an electrode, and the lower surface will have the Ground boundary condition applied.

This problem has a similar geometry to that of the Cylinder and Plane case study, so the previous results can be used as a guide. For an accurate value of the capacitance, it is necessary to make the effective radius of the air box at least 20 times the effective radius of the device, and symmetry can be exploited by modeling only half of the device.

The diagram below shows the cross-section of the half model, with dimensions in millimeters. The small rectangle measuring 2 mm × 1 mm represents half of the track region of the PCB, and the adjacent small rectangle represents the remainder of the PCB. A Ground boundary condition is assigned to the bottom surfaces of both of these components, and the top surface of the track region is defined as an electrode. All other outer boundaries have the default Flux Tangential boundary condition.



We require a dielectric material for the components representing the PCB. For this case study, the specified relative permittivity matches a standard material in the ElecNet library: EP05: Relative permittivity 5. For most practical problems, it would be necessary to create a new user-defined material with the required properties.

Creating the model

1. Start a new model and save it as **Single track half**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum X and Y values set to zero, and a spacing of 1 mm.
4. Draw the square boundary of the half model, with the coordinate origin at the bottom left-hand corner.
5. Add two lines for the PCB components.
6. Make the two PCB components in a single operation as follows:
   - Select the smallest rectangle.
   - Hold down the Ctrl key, and select the larger rectangle.
   - Click Make Component in a Line.
   - In the Make Component dialog, uncheck the box for Union Selected Construction Slice Surfaces. The Name text box will then be grayed out.
   - Select the material as **EP05: Relative permittivity 5**.
   - Type the sweep distance as **1000** mm.
   - Click OK.

   This should create two objects named Component#1 and Component#2.
7. Delete two lines for the PCB component, leaving the square for the air box.
8. Turn off the construction grid.
9. Make the air box:
   - Material: AIR.
   - Sweep distance: 1000 mm.
10. Apply the Ground boundary condition to the air box face that corresponds to the bottom line in the diagram.
11. Make an electrode for the track component:
    - Select the face for the upper surface of the small PCB component.
    - Make the electrode.
    - Set the voltage to 1 V.

Solving and post-processing

1. Set the options for solving, with the following changes from the default settings:
   - Solver Options:
     CG tolerance 0.001%,
     Polynomial order 2.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.1%.
2. Solve as Static 2D.
3. Inspect the contour plot of V and the shaded plot of the smoothed $|\mathbf{E}|$ values.
4. Inspect the computed global quantities, and calculate the capacitance value as follows (see appendix B for details):
   - From the charge: $C = Q / V$, where Q is the charge on Electrode#1, and V is the voltage difference (1 V in this case).
   - From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

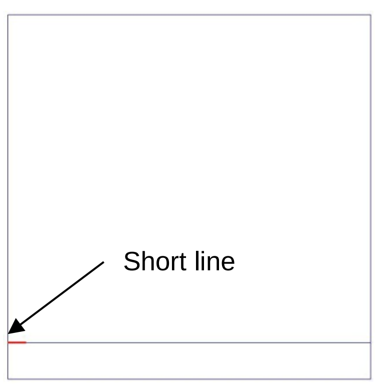
The results below were obtained with ElecNet version 7.5.

Stored electric energy:	2. 3414×10^{-11} J
Charge on Electrode#1:	4.6820×10^{-11} C
Capacitance from charge:	46.82 pF
Capacitance from stored energy:	46.83 pF
Maximum value of $|\mathbf{E}|$ smoothed:	5.024 kV/m

Since this is a half model, the capacitance between the track and ground is double the value given above: 93.64 pF from the charge, or 93.66 pF from the stored energy, giving a mean value of 93.65 pF.

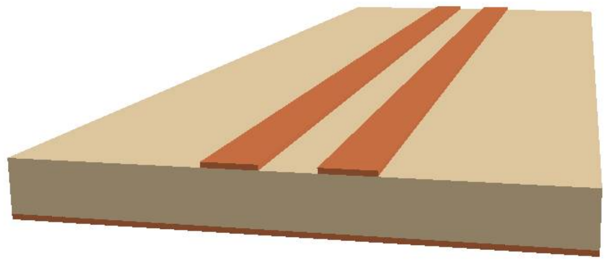
Alternative model

A single component can be used for the PCB if an extra face is created for the track region. This can be done by drawing two lines for the upper surface as shown below. In the Make Component dialog, the option Remove Unnecessary Vertices on the Boundary must be unchecked. Two faces will be created for the upper surface, one of which can be used to define an electrode.



Twin PCB tracks

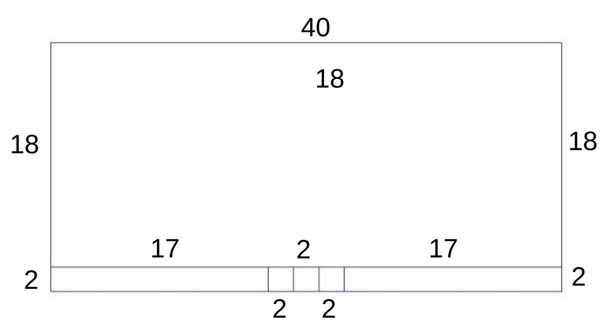
This is an extension of the previous case study. A printed-circuit board (PCB) has a copper sheet on one side and two copper tracks on the other side, as shown in the diagram below. The problem is to determine the capacitance per meter length between the tracks, and between each track and the grounded copper sheet.



Modeling the device

The copper tracks are 2 mm wide, with a 2 mm gap between them, the board is 2 mm thick, and the relative permittivity of the board material is 5.0. As before, the copper sheet and the copper tracks are very thin, so it is not necessary to model these parts as solid objects. Instead, the component representing the PCB is divided into five regions: one region underneath each track, and adjoining regions on each side. The upper surface of each track region will be specified as an electrode, with impressed voltages of +1 V and –1 V, and the lower surface will have the Ground boundary condition applied.

Since the tracks have equal and opposite voltages, the device has odd symmetry, so a half model could be used with a Ground boundary condition on the symmetry plane. However, a full model is required if the voltage magnitudes are different. A full model also enables the ElecNet RLC Matrix Calculator to be used for calculating the capacitance values. The diagram below shows the cross-section of the full model, with dimensions in millimeters. The outer small squares of side 2 mm represent the track regions of the PCB; the central small square and the small rectangles represent the remainder of the PCB. A Ground boundary condition is assigned to the bottom surfaces of all of these components, and the top surface of each track region is defined as an electrode. All other outer boundaries have the default Flux Tangential boundary condition.



As before, a standard material in the ElecNet library: EP05: Relative permittivity 5 is used for the board material.

An alternative model could be constructed using a single component for the PCB, as in the previous case study, with extra faces created for the two electrodes.

Creating the model

1. Start a new model and save it as **Twin track**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum Y value set to zero, and a spacing of 1 mm.
4. Draw the rectangular boundary of the half model, with the coordinate origin at the at the mid-point of the bottom line in the diagram.
5. Add five lines for the PCB components.
6. Make the five PCB components in a single operation as follows:
   - Select the left-hand rectangle.
   - Hold down the Ctrl key, and select the three small squares and the right-hand rectangle.
   - Click Make Component in a Line.
   - In the Make Component dialog, uncheck the box for Union Selected Construction Slice Surfaces. The Name text box will then be grayed out.
   - Select the material as **EP05: Relative permittivity 5**.
   - Type the sweep distance as **1000** mm.
   - Click OK.

   This should create five objects named Component#1 to Component#5.
7. Delete five lines for the PCB component, leaving the rectangle for the air box.
8. Turn off the construction grid.
9. Make the air box:
   - Material: AIR.
   - Sweep distance: 1000 mm.
10. Apply the Ground boundary condition to the air box face that corresponds to the bottom line in the diagram.
11. Make an electrode for each PCB component that has a track:
    - Select the face for the upper surface.
    - Set the voltage to –1 V for one electrode and +1 V for the other.

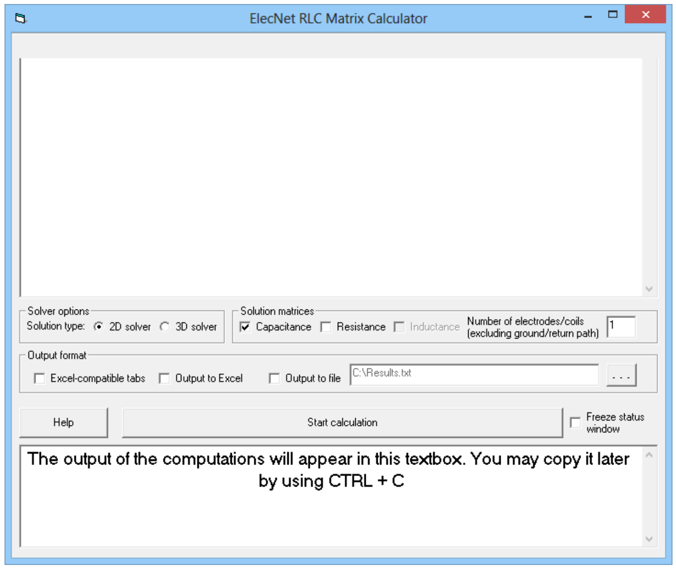
Solving and post-processing

1. Set the options for solving, with the following changes from the default settings:
   - Solver Options:
     CG tolerance 0.0001%,
     Polynomial order 2.
   - Adaption Options:
     Use *h*-adaption,
     Tolerance 0.1%.
2. Solve as Static 2D.
3. Inspect the contour plot of *V* and the shaded plot of the smoothed |**E**| values.

Capacitance values

Capacitance values can be obtained with the ElecNet RLC Matrix Calculator as follows:

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



2 Set options as follows:

- Solution Type: 2D
- Solution Matrices: Capacitance

The Number of Electrodes should be ignored; they are defined in the model.

3 Click Start Calculation.

- In the Save Now dialog, click Yes.
- When the calculation is complete, capacitance values are displayed in the upper part of the dialog.

Sample results

The results below were obtained with ElecNet version 7.5.

Stored electric energy:	1. 0430×10^{-10} J
Charge on Electrode#1:	-1.0428×10^{-10} C
Charge on Electrode#2:	1.0431×10^{-10} C
Maximum value of $\lvert\mathbf{E}\rvert$ smoothed:	11.70 kV/m
Capacitance 1 to ground:	85.39 pF
Capacitance 2 to ground:	85.40 pF
Capacitance 1 to 2:	9.450 pF

The values for the capacitance to ground are different from the value of 93.59 pF obtained for a single track in the previous study, because the Matrix Calculator returns the capacitance values in an equivalent network. See Appendix B for an explanation of how the capacitance values are defined and calculated. To get the corresponding capacitance for a single track, the capacitance 1 to 2 must be added to the capacitance to ground, giving a value of 94.85 pF.

Open boundary problem

Several of the case studies are open boundary problems where a modeling issue is the required size of the air box for accurate results. With some problems, the open boundary can be modeled exactly in ElecNet by the Kelvin transformation technique (Appendix A). For this to be possible, the model must have translational symmetry, and it must be possible to enclose the entire 2D model in a circle of finite radius. Thus, problems with infinite ground planes cannot be handled in this way. The parallel channel problem of chapter 2 is an open-boundary problem that can be handled with the Kelvin transformation. In this case study, the results obtained with an exact open boundary representation are compared with the results obtained with a large air box.

Modeling the device

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

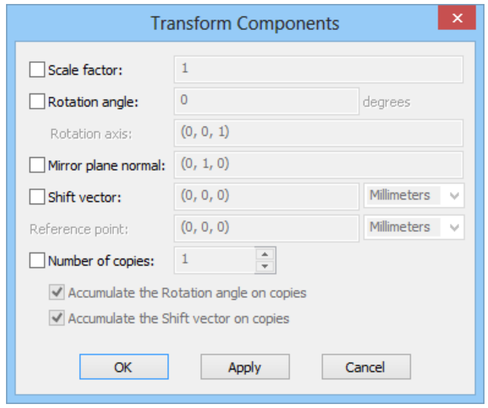
The open boundary representation has two parts: (a) a circular boundary surrounding the model, enclosing an air space; (b) a second small circular boundary, enclosing another air space, outside the first. Corresponding points on the two boundaries are linked so that the field values will be identical,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

Creating the basic model

1. Open the model Channels that was created in the tutorial of chapter 2.
2. Check the options for solving; changes from the default settings should be:
   - Solver Options:
CG tolerance 0.001%,
Polynomial order 2.
   - Adaption Options:
Use *h*-adaption,
Tolerance 0.05%.
3. Solve the model as Static 2D, and note the values of stored energy, force and charge in the Results window.
4. Save the model to store the results, then use File | Save As to save the model as a new file **Channels open**.
5. In the Object page, select the AirBox component and delete it by pressing Delete.
6. In the View window, select and delete the air box construction slice circle.

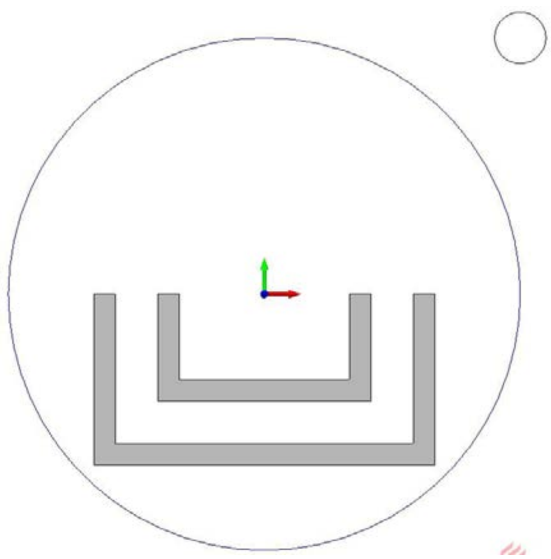
Adding the open boundary

1. Create a new air box of radius 60 mm, centered at the origin, with a sweep distance of 160 mm, 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 type the value **0.1** in the text box.
- Click Shift Vector and edit the text box: (**60**, **60**, 0).
- Click Apply the Transformation to a Copy of the Selection.
- Click OK.

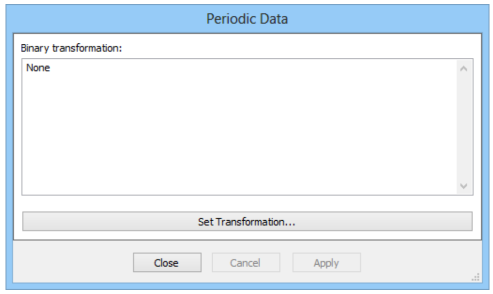
A new air box should appear in the View 1 window, as shown below, and there should be a new component in the Object page.



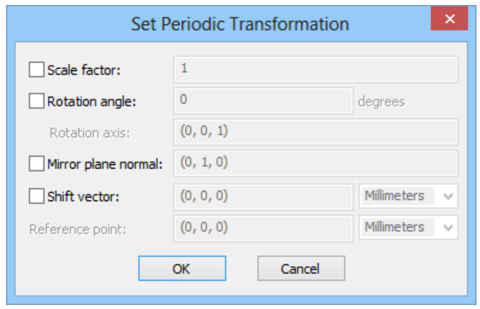
4. In the Object page, change the name of the new component to **Exterior**.
5. Display the tree directory for the AirSpace component.
   - Select Face#3
   - Hold down the Ctrl key, and select Face#4.

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

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



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



- Click Scale Factor and type the value **0.1** in the text box.
- Click Shift Vector and edit the text box: (**60**, **60**, 0).

  These settings are identical to those in the Transform Components dialog that was used 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.

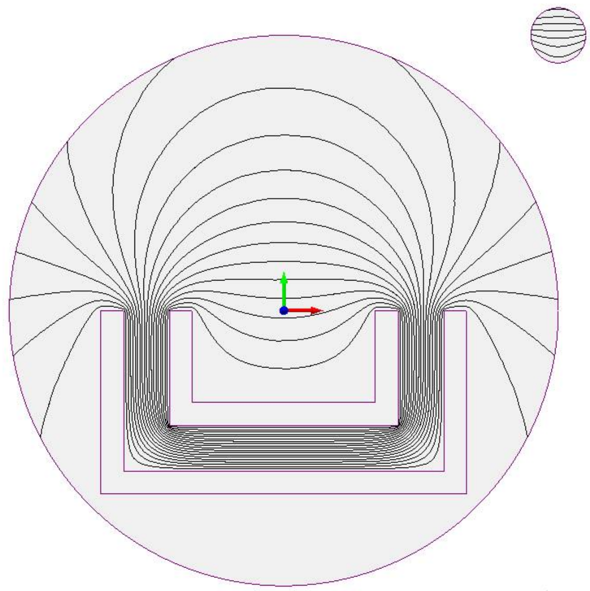
Solving and post-processing

1 Check the options for solving; changes from the default settings should be:

- Solver Options:
  CG tolerance 0.001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

3 Display the equipotential plot for Problem 1:



Equipotential lines inside the larger circle show the true field for this portion of the open boundary problem. Those inside the smaller circle are a transformed representation of the field outside the larger circle.

4 Note the values in the Results window and compare them with the previous values obtained with a large air box.

Sample results

The results below were obtained with ElecNet version 7.5:

	Large air box	Open boundary
Stored electric energy (J):	0.00091326	0.00091253
Charge on Electrode#1 (C):	-1.8265×10^{-7}	-1.8250×10^{-7}
Charge on Electrode#2 (C):	$+1.8265\times10^{-7}$	$+1.8251\times10^{-7}$
Force on Outer channel (N):	+0.041531	+0.041476
Force on Inner channel (N):	–0.041487	–0.041488

In this example, the differences between the large air box results and the open boundary results are small, indicating that the size of the large air box has been correctly chosen.

Chapter 4

Case Studies: Rotational Geometry

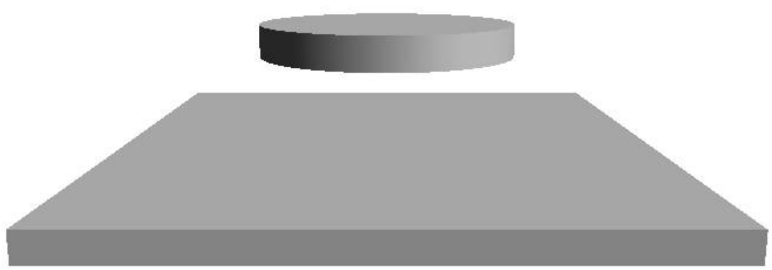
Introduction

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 ElecNet default angle of 90º will be used for all of the case studies. This construction technique is described in detail in the first case study: Disc above a Plane.

Disc above a plane

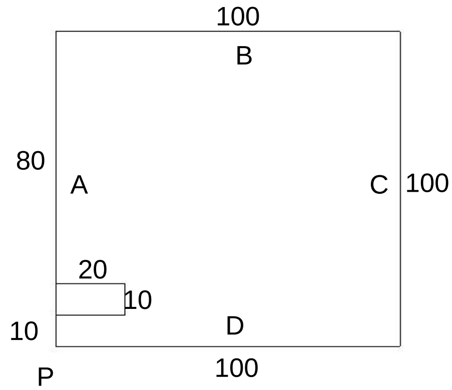
The diagram below shows a conducting disc above a conducting plane. The objectives are to examine the electric field distribution, and to determine the capacitance of the system.



The disc has a diameter of 40 mm and a depth of 10 mm, and there is a uniform gap of 10 mm between the disc and the plane. The plane is grounded and the disc is maintained at a voltage of 10 kV with respect to ground. A cross-sectional view is shown below; the plane is considered to be of infinite extent.

Modeling the device

This is an open-boundary problem, where the electric field extends to infinity. An artificial boundary must be applied in the form of an air box, since the Kelvin transformation method (Appendix A) cannot be used with rotational geometry. A suitable initial choice for the radius of the air box is 5 times the radius of the disc, and the height is set equal to the radius. However, the choice of air box size will affect the accuracy of the capacitance calculation, so the effect of increasing the size will be investigated. The diagram below shows the cross-section in the XY plane that is rotated about the Y axis to form the system, with dimensions in millimeters. The left-hand side of the square is drawn along the Y axis, with the point P at the origin. Surfaces corresponding to sides A, B and C have the default Flux Tangential boundary condition, and the surface corresponding to side D is assigned the Ground boundary condition. Since we are not interested in the force on the disc, the entire disc component is defined as an electrode; the interior of the disc is thereby excluded from the finite-element mesh.



Creating the model

1. Start a new model and save it as **Disc plane**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum X and Y values set to zero.
4. Draw the square boundary of the model, with the coordinate origin at the bottom left-hand corner (point P in the diagram).
5. Draw the cross-section of the disc component as a rectangle 10 mm × 20 mm as shown in the diagram and make the component as follows:
   - Select the interior of the rectangle.
   - Click the Make Component in an Arc button. 
   - Follow the usual procedure for setting the component name and the material (Aluminum: 3.8e7 Siemens/meter).
   - Leave the sweep parameters set at their default values: angle 90º, center (0, 0), axis vector (0, –1).
   - Click OK.
6. Delete three lines for the disc component, leaving the square for the air box.
7. Turn off the construction grid.

8 Make the air box by selecting the interior of the square and sweeping in an arc, with the default sweep parameters.

9 Apply the Ground boundary condition to the face of the air box that corresponds to line D in the diagram.

10 Make an electrode from the disc component:

- Select the entire component in the Object page, instead of a set of faces.
- Set the voltage to 10000 (10 kV).

Solving and post-processing

For models with rotational symmetry, the ElecNet 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, with the following changes from the default settings:

- Solver Options:
  CG tolerance 0.001%,
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.1%.

2 Solve as Static 2D.

3 Inspect the contour plot of V and the shaded plot of the smoothed |E| values.

Observe the high value of electric field strength at the edges of the disc, corresponding to the corners in the 2D representation. Theoretically, the field is infinite at these edges. To reduce the maximum value of E, it would be necessary to alter the shape of the electrode – see below.

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

- From the charge: $C = Q / V$, where Q is the charge on the electrode, and V is the voltage relative to ground (10 kV in this case).
- From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

The results below were obtained with ElecNet version 7.5:

Stored electric energy:	0.00016795 J
Charge on Electrode#1:	3.3582×10^{-8} C
Capacitance from charge:	3.358 pF
Capacitance from stored energy:	3.359 pF

Effect of air box size

The table below shows the effect of changing the size of the air box. In each case, the cross section of the air box in the drawing plane is a square. For the 200 mm and 400 mm boxes, the adaption tolerance was reduced from 0.1% to 0.05%.

Air box radius and height (mm):	100	200	400
Stored electric energy (J):	0.00016795	0.00017073	0.00017109
Charge on Electrode#1 (C):	3.3582×10^{-8}	3.4144×10^{-8}	3.4213×10^{-8}
Capacitance from charge (pF):	3.358	3.414	3.421
Capacitance from stored energy (pF):	3.359	3.415	3.422

From these results, the capacitance appears to be converging on a limit as the size of the air box is increased. Relative to the value for a 400 mm air box, the capacitance values for 200 mm and 100 mm air boxes are lower by 0.2% and 1.8% respectively. These results confirm the general rule that, for high accuracy, the radius of the air box should be about 10 times the radius of the device.

Effect of electrode shape

The table below shows the effect of rounding the corners of the disc electrode in the same way as the channel flanges in chapter 2, with a radius of 5 mm; the air box radius is 200 mm.

Electrode corners:	square	rounded
Stored electric energy (J):	0.00017073	0.00015572
Charge on Electrode#1 (C):	3.4144×10^{-8}	3.1148×10^{-8}
Capacitance from charge (pF):	3.414	3.115
Capacitance from stored energy (pF):	3.415	3.114
Maximum value of |**E**| smoothed (kV/mm):	>3.87*	1.65

* The value depends on the size of mesh elements, and is theoretically infinite.

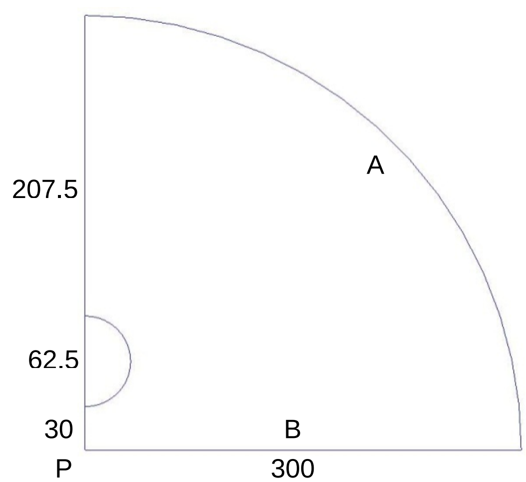
Two spheres

The diagram below shows two conducting spheres as used for high voltage measurement. The objectives are to examine the electric field distribution, to determine the capacitance of the system, and compare the results with values obtained by analytical solution of the field equations.

Each sphere has a diameter of 62.5 mm, the gap between the spheres is 60 mm, and the voltage between the spheres is 185 kV.

Modeling the device

Since the configuration has a plane of symmetry, it is only necessary to model one sphere, at a voltage of 185/2 = 92.5 kV with respect to a ground plane; the other sphere, which is not modeled, would be at a voltage of –92.5 kV. This is an open-boundary problem, so an artificial boundary will be applied in the form of a hemispherical air box, with a radius of about 10 times the radius of the sphere. However, the choice of air box size will affect the accuracy of the capacitance calculation, so the effect of increasing the size will be investigated. The diagram below shows the cross-section in the XY plane that is rotated about the Y axis to form the system, with dimensions in millimeters. The left-hand side is drawn along the Y axis, with the point P at the origin. The surface corresponding to arc A has the default Flux Tangential boundary condition, and the surface corresponding to B is assigned the Ground boundary condition. Since we are not interested in the force on the sphere, the entire component is defined as an electrode; the interior of the sphere is thereby excluded from the finite-element mesh.



Creating the model

1. Start a new model and save it as **Two spheres**.
2. Set the model length units to millimeters.
3. Set and display the construction grid, with the minimum X and Y values set to zero.
4. Let the coordinate origin be the point representing the center of the gap between the spheres, which is the point P in the diagram above.
5. Draw the boundary of the model as a quadrant of radius 300 mm as shown in the diagram.
6. Using the keyboard input bar, draw the cross-section of the sphere as an arc with its center at (0, 61.25) and end-points at (0, 30) and (0, 92.5).
7. Make the sphere component, using the material Aluminum: 3.8e7 Siemens/meter.
8. Delete the construction-slice arc for the sphere.
9. Turn off the construction grid.
10. Make the air box from the quadrant outer boundary.
11. Apply the Ground boundary condition to the face that corresponds to line B in the diagram.
12. Make an electrode from the sphere component, and set the voltage to 92500 V (92.5 kV)

Solving and post-processing

1. Set the options for solving, with the following changes from the default settings:
   - Solver Options:
CG tolerance 0.0001%,
   - Adaption Options:
Use *h*-adaption,
Tolerance 0.1%.
2. Solve as Static 2D.
3. Inspect the contour plot of V and the shaded plot of the smoothed |E| values.
4. Inspect the computed global quantities, and calculate capacitance values as follows for the half model:
   - From the charge: $C = Q / V$, where Q is the charge on the electrode, and V is the voltage relative to ground (92.5 kV in this case).
   - From the stored energy: $C = 2W / V^2$, where W is the stored electric energy in the half model.

Sample results

The results below were obtained with ElecNet version 7.5:

Stored electric energy:	0.019665 J
Charge on Electrode#1:	4.2517×10^{-7} C
Capacitance from charge:	4.596 pF
Capacitance from stored energy:	4.597 pF
Maximum value of |**E**| smoothed:	5.351 kV/mm

The capacitance for the full model is half of the value for the half model: 2.298 pF. The analytical values [3] are: capacitance = 2.349 pF, maximum |**E**| = 5.352 kV/mm.

Effect of air box size

The table below shows the effect of changing the size of the air box. In each case, the cross section of the air box in the drawing plane is a quadrant. For the 600 mm and 1200 mm boxes, the adaption tolerance was reduced from 0.1% to 0.02%.

Air box radius (mm):	300	600	1200
Stored electric energy (J):	0.019665	0.020041	0.020090
Charge on Electrode#1 (C):	4.2517×10^{-7}	4.3332×10^{-7}	4.3436×10^{-7}
Capacitance from charge (pF):	4.596	4.685	4.696
Capacitance from stored energy (pF):	4.597	4.685	4.696
Maximum value of |**E**| smoothed (kV/mm):	5.351	5.367	5.368

From these results, the capacitance appears to be converging on a limit as the size of the air box is increased. Relative to the analytical value, the capacitance values for 300 mm, 600 mm and 1200 mm air boxes are lower by 2.1%, 0.3% and 0.04% respectively. The corresponding errors for the maximum |**E**| values are 0.6%, 0.4% and 0.7% respectively. It appears that an air box radius of 600 mm gives acceptably low error values for this problem.

Chapter 5

Scripting

Introduction

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

Scripts are text files containing commands that control ElecNet. A script can be recorded during an ElecNet 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 ElecNet 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.

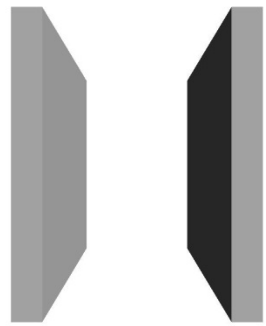
The most advanced kind of scripting uses another application to communicate with ElecNet through the Microsoft ActiveX Automation Interface. Microsoft Excel, for example, can be used in this way. A script in the form of an Excel macro can command ElecNet to build and solve a model, using data entered on the spreadsheet, and then get results back from ElecNet to display on the spreadsheet. An example of this form of scripting is given in the section “Automation with Excel” on page 77.

Further information

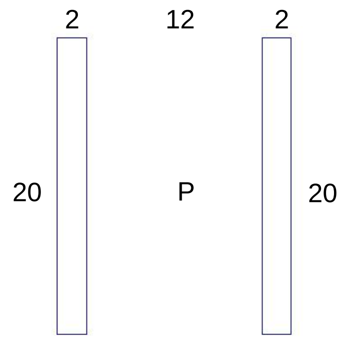
The ElecNet Help gives full particulars of the ElecNet scripting commands. Guidance on writing scripts is available from the Support area of the Infolytica website, www.infolytica.com.

Example model

To illustrate the principles of scripting, a simple model will be used. The diagram below shows two parallel plates, maintained at different voltages. The plates are deep enough for a 2D model to be used.



The cross section of the system is shown below, with dimensions in millimeters.



This is an open boundary problem, which could be handled by the open boundary technique described in the case study on page 58. However, for simplicity, a simple circular boundary will be used.

Creating the model

1. Save the file as **Parallel plates**.
2. Set the model length units to millimeters.
3. Display the construction grid.

   The default grid extent and spacing are satisfactory.
4. Let the coordinate origin be the point P mid-way between the plates, so that the plates are disposed symmetrically about the origin.
5. Construct a component for each plate:
   - Material: Aluminum: 3.8e7 Siemens/meter.
   - Sweep distance: 1000 mm.
6. Delete the construction slice lines.
7. Create a circular air box with a radius of 400 mm.
   - Material: AIR.
   - Sweep distance: 1000 mm.

8 Make electrodes for the plate components:

- Select all the faces for the component.
- Set the voltage to 6 kV for the left plate and 0 V for the right plate.

Solving and post-processing

1 Set the options for solving, with the following changes from the default settings:

- Solver Options:
  CG tolerance 0.001%,
  Polynomial order 2.
- Adaption Options:
  Use *h*-adaption,
  Tolerance 0.05%.

2 Solve as Static 2D.

3 Inspect the contour plot of V and the shaded plot of the smoothed $|\mathbf{E}|$ values.

4 Inspect the computed global quantities, and calculate capacitance values as follows:

- From the charge: $C = Q / V$, where Q is the charge on the electrode, and V is the voltage between the plates (6 kV in this case).
- From the stored energy: $C = 2W / V^2$, where W is the stored electric energy.

Sample results

The results below were obtained with ElecNet version 7.5.

Stored electric energy:	0.00049236 J
X component of force on left plate:	0.02659 N
X component of force on right plate:	–0.02663 N
Charge on left plate:	1.6411×10^{-7} C
Charge on right plate:	-1.6413×10^{-7} C
Capacitance from charge:	27.35 pF
Capacitance from stored energy:	27.35 pF

Script for the model

Creating a User Script Log file

Recording a script is similar to recording a macro in applications such a Word or Excel. After a User Script Log file has been opened, commands representing the operations carried out with the mouse and the keyboard will be recorded in the file. This continues until User Script recording is paused or stopped. The following steps will create a script file for building and solving the model.

1. Save the Parallel plates model.
2. On the File menu, click New.
3. On the Scripting menu, click Start Recording User Script.

   A Save As dialog box is displayed.
4. Enter the file name as **Parallel plates.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 Tools menu, click Scripting and select Stop User Script.
   - In the dialog box for editing the user script file, click NO.

Running the script

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

   An Open dialog box is displayed.
3. Select the file Parallel plates.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 plates. 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 Parallel plates.vbs, and open the file.

This is the folder used for saving the ElecNet model.

5 Save the file as **Parallel plates modified.vbs**.

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

```
Call getDocument().getView().newLine(-6, 10, -8, 10)
Call getDocument().getView().newLine(-8, 10, -8, -10)
Call getDocument().getView().newLine(-8, -10, -6, -10)
Call getDocument().getView().newLine(-6, -10, -6, 10)
```

…
…

```
Call getDocument().getView().newLine(6, 10, 8, 10)
Call getDocument().getView().newLine(8, 10, 8, -10)
Call getDocument().getView().newLine(8, -10, 6, -10)
Call getDocument().getView().newLine(6, -10, 6, 10)
```

These commands draw lines to form rectangles 2 mm × 20 mm. The order may be different, depending on the way the lines 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(-6, 20, -8, 20)
Call getDocument().getView().newLine(-8, 20, -8, -20)
Call getDocument().getView().newLine(-8, -20, -6, -20)
Call getDocument().getView().newLine(-6, -20, -6, 20)
```

…
…

```
Call getDocument().getView().newLine(6, 20, 8, 20)
Call getDocument().getView().newLine(8, 20, 8, -20)
Call getDocument().getView().newLine(8, -20, 6, -20)
Call getDocument().getView().newLine(6, -20, 6, 20)
```

8 On the File menu, click Save.

9 Return to ElecNet.

10 Start a new model.

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

12 Check the results.

Creating a new script

Some operations are possible with scripts, but are not possible when using ElecNet interactively. For example, the Field Probe 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 electric field magnitude at the point. It does this repeatedly until the user clicks the No button.

Line / Script

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

Automation with Excel

This section shows how Microsoft Excel can communicate with ElecNet via the ActiveX Automation Interface to set up and analyze the model of two parallel plates. It includes an implementation of the Kelvin transformation for an open boundary. The picture below shows part of the Excel worksheet.

	A	B	C	D	E	F	G	H	I	J	K	L
1					Start ElecNet		Close ElecNet		ElecNet Visibility		Run Model	
2	**Analysis of parallel plates,**											
3												
4												
5	Plate x dimension			2.0	mm							
6	Plate y dimension			20.0	mm							
7												
8	Separation			12.0	mm							
9												
10	Voltage difference			6000.0	V							
11	Plate material			Aluminum: 3.8e7 Siemens/meter								
12												
13	Force on plate 1			-0.026629	N							
14	Force on plate 2			0.026594	N							
15	Charge on plate 1			1.64112	μC							
16	Charge on plate 2			-1.64131	μC							
17	Capacitance			27.35	pF							

The core of the Excel implementation is a set of *macros* or subroutines, written in Visual Basic for Applications (VBA), 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 ElecNet: start the ElecNet application and set the length units.
- Close ElecNet: close the model file, close the application and release resources.
- ElecNet Visibility: make ElecNet visible or invisible.
- Run Model: get data values from cells D5 to D11, send commands to ElecNet to build and solve the model, get the X components of force on the plates and the charge values from ElecNet, and display the results in cells D13 to D17.

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 plate.

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.

Registered users can download a copy of the complete Excel workbook, containing the worksheet and the macro module, by clicking here.

The instructions below relate to Microsoft Office Excel 2013. Other versions of Excel may behave differently.

Excel worksheet – 1

1. Start Excel.
2. If the Developer tab in the Ribbon is not visible, do the following:
    - Click the Office button, and select Excel Options.
    - On the Popular page, click Show Developer Tab in the Ribbon.
    - Click OK.
3. Start a new Excel workbook.
4. Save the file as **Parallel plates.xlsm**

    This is an Excel macro enabled workbook.
5. Enter the text and numerical values exactly as shown on the previous page, except for cells D13 to D17, which should be left blank.
6. Set the numerical format of cells as follows:
    - D5 to D10: one decimal place.
    - D13 and D14: six decimal places.
    - D15 and D16: five decimal places.
    - D17: two 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. On the Developer tab, click the Visual Basic button.

    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 page, taking care to copy it accurately. See the section "Comments on the Code" for an explanation of the content.

```
' Parallel plates with ElecNet.

Option Explicit

Dim Elec As Object, Doc As Object, Con As Object
Dim Cur As Object, Sol As Object
Dim Lx As Double, Ly As Double, Lz As Double
Dim Lg As Double, Vd As Double
Dim Visible As Boolean, Running As Boolean
Dim Material

Public Sub StartElecNet()
' Subroutine called by the Start ElecNet button.
' Start ElecNet and set variables.
    If Running Then
        Call MsgBox("ElecNet is already running.", vbOKOnly)
    Else
        Set Elec = CreateObject("Elecnet.Application")
        Visible = True
        Elec.Visible = Visible
        Set Doc = Elec.newDocument
        Set Con = Elec.GetConstants
        Set View = Doc.getView
        Running = True
    End If
End Sub

Public Sub CloseElecNet()
' Subroutine called by the Close ElecNet button.
' Close ElecNet and reset variables.
    If Not Running Then
        Call MsgBox("ElecNet is not running.", vbOKOnly)
    Else
        Call Doc.Close(Con.infoFalse)
        Call Elec.Exit
        Set Elec = Nothing
        Running = False
    End If
End Sub

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

Excel worksheet – 2

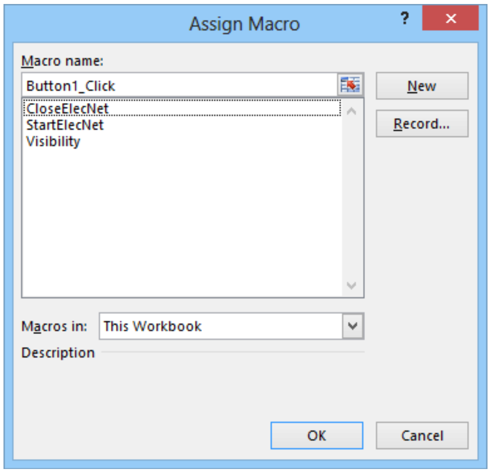
1 Return to the Excel window by clicking the Excel icon on the Windows task bar.

2 On the Developer tab, click Insert and select the Button from Form Controls.

This is the small rectangle in the top left-hand corner of the group of controls.

3 Click on worksheet cell E1 to insert a button.

A dialog box should appear:



4 In the Macro Name drop-down list, select StartElecNet.

5 Click OK.

6 Click inside the new button, and change the text to **Start ElecNet**.

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.

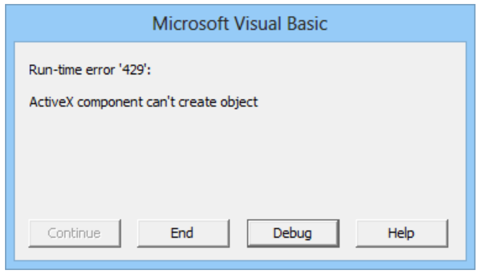
10 In a similar way, insert two more buttons:

- A button named Close ElecNet, linked to the CloseElecNet subroutine.
- A button named ElecNet Visibility, linked to the Visibility subroutine.

Testing – 1

1 Click Start ElecNet.

- If the macro is working correctly, the ElecNet window should open.
- 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 ElecNet Visibility

- If there are errors, correct them as before.
- Click Yes to make ElecNet invisible.

The ElecNet window and task-bar icon should disappear.

3 Click ElecNet Visibility again.

- Click Yes to make ElecNet visible.

The ElecNet window and task-bar icon should reappear.

4 Click Close ElecNet.

- The ElecNet application should close.
- 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 ElecNet again when it is already running, or use the other two buttons when ElecNet is not running.

6 Ensure that ElecNet is closed before continuing with the development.

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

Completion

1. Return to the Visual Basic editor.
2. Type in the remainder of the code listed below.

```
Public Sub RunModel()
' Subroutine called by the Run Model button.
' Build and solve the model.
Dim Q1 As Double, Q2 As Double
Dim Fx, Fy, Fz
    If Not Running Then
        Call MsgBox("ElecNet is not running.", vbOKOnly)
    Else
        ' Build the parallel plate model
        NewModel
        GetData
        Call MakePlate(Lx, Ly, Lz, (Lx + Lg) / 2, 0, "Plate#1")
        Call MakePlate(Lx, Ly, Lz, -(Lx + Lg) / 2, 0, "Plate#2")
        MakeElectrodes
        OpenBoundary
        ' Solve the model and get results from ElecNet.
        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
            Q1 = Sol.getChargeOnElectrode(1, 1)
            .Cells(15, 4).Value = Q1 * 1000000#
            Q2 = Sol.getChargeOnElectrode(1, 2)
            .Cells(16, 4).Value = Q2 * 1000000#
            .Cells(17, 4).Value = (Q1 - Q2) / (2 * Vd) * 1000000000000#
        End With
    End If
End Sub

Private Sub NewModel()
' Close the current model and start a new model.
    Call Doc.Close(Con.infoFalse)
    Set Doc = Elec.newDocument
    Set Con = Elec.GetConstants
    Set View = Doc.getView
    Set Sol = Doc.getSolution
    Call Doc.setDefaultLengthUnit("Millimeters")
    Call Doc.setScaledToFit(True)
    Call Doc.setPolynomialOrder("", 2)
    Call Doc.setCGTolerance(0.00001)
    Call Doc.useHAdaption(True)
    Call Doc.setHAdaptionRefinement(0.25)
    Call Doc.setAdaptionTolerance(0.0005)
End Sub
```

```
Private Sub GetData()
' Get data values from column D on Sheet1.
    With Sheets("Sheet1")
        Lx = .Cells(5, 4)  ' Plate x dimension
        Ly = .Cells(6, 4)  ' Plate y dimension
        Lz = .Cells(7, 4)  ' Plate z dimension
        Lg = .Cells(8, 4)  ' Gap between plates
        Vd = .Cells(10, 4) ' Voltage difference.
        Material = .Cells(11, 4) ' Material name
    End With
End Sub

Private Sub MakePlate(Dx As Double, Dy As Double, Dz As Double, _
Ox As Double, Oy As Double, Name As String)
' Make a plate, dimensions Dx by Dy by Dz, center at (Ox, Oy).
' Name is the component name.
Dim I As Integer
Dim X(4) As Double, Y(4) As Double
    X(1) = Ox - Dx / 2
    Y(1) = Oy - Dy / 2
    X(2) = X(1) + Dx
    Y(2) = Y(1)
    X(3) = X(2)
    Y(3) = Y(2) + Dy
    X(4) = X(1)
    Y(4) = Y(3)
    For I = 1 To 3
        Call View.newLine(X(I), Y(I), X(I + 1), Y(I + 1))
    Next I
    Call View.newLine(X(4), Y(4), X(1), Y(1))
    ReDim ArrayOfValues(0)
    ArrayOfValues(0) = Con.infoSliceSurface
    Call View.selectAt(Ox, Oy, Con.infoSetSelection, ArrayOfValues)
    ArrayOfValues(0) = Name
    Call View.makeComponentInALine(Dz, ArrayOfValues, "Name=" + Material)
End Sub

Private Sub MakeElectrodes()
' Make electrodes from plate surfaces.
    Dim I As Integer
    Dim ArrayOfValues(5)
    For I = 0 To 5
        ArrayOfValues(I) = "Plate#1,Face#" & (I + 1)
    Next I
    Call Doc.makeElectrode(ArrayOfValues)
    For I = 0 To 5
        ArrayOfValues(I) = "Plate#2,Face#" & (I + 1)
    Next I
    Call Doc.makeElectrode(ArrayOfValues)
    Call Doc.setElectrodeVoltage("Electrode#1", Vd, 0)
End Sub
```

```
Private Sub OpenBoundary()
' Make the air space and exterior regions.
' Apply boundary conditions for the Kelvin transformation.
Const Krad = 1.5, Kmag = 0.1
' Krad: radius factor for air space
' Kmag: scale factor for exterior
' Ndiv: number of edge subdivisions
Dim Rb As Double, Subdiv As String
Dim ArrayOfValues(0), ShiftVec(2), Center(2)
' Delete construction slice lines.
    ArrayOfValues(0) = Con.infoSliceLine
    Call View.SelectAll(Con.infoSetSelection, ArrayOfValues)
    Call View.deleteSelection
' Calculate radius of interior air region and build the component.
    Rb = Krad * Sqr((Lx + Lg) ^ 2 + Ly ^ 2)
    Call View.newCircle(0, 0, Rb)
    ArrayOfValues(0) = Con.infoSliceSurface
    Call View.selectAt(0, 0, Con.infoSetSelection, ArrayOfValues)
    ArrayOfValues(0) = "Air space"
    Call View.makeComponentInALine(Lz, ArrayOfValues, "Name=AIR")
' Build the exterior component.
    Call View.newCircle(Rb, Rb, Kmag * Rb)
    ArrayOfValues(0) = Con.infoSliceSurface
    Call View.selectAt(Rb, Rb, Con.infoSetSelection, ArrayOfValues)
    ArrayOfValues(0) = "Exterior"
    Call View.makeComponentInALine(Kmag * Lz, ArrayOfValues, "Name=AIR")
' Apply boundary conditions to each half of the air component.
    ArrayOfValues(0) = "Air space,Face#3"
    Call Doc.createBoundaryCondition(ArrayOfValues, "BC1")
    ShiftVec(0) = Rb
    ShiftVec(1) = Rb
    ShiftVec(2) = 0
    Center(0) = 0
    Center(1) = 0
    Center(2) = 0
    Call Doc.setEvenPeriodic("BC1", Kmag, Null, Null, Null, ShiftVec, Center)
    ArrayOfValues(0) = "Air space,Face#4"
    Call Doc.createBoundaryCondition(ArrayOfValues, "BC2")
    Call Doc.setEvenPeriodic("BC2", Kmag, Null, Null, Null, ShiftVec, Center)
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 ElecNet before continuing.

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

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 81. 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 ElecNet model, to find the error.

Using the worksheet

Try changing the dimensions of the plates and the gap between the plates.

Comments on the code

The commands differ in some respects from those found in a User Script Log file and listed in the ElecNet help. In particular, a named constant such as **`infoSetSelection`** has to be given in the form **`Elec.GetConstants.infoSetSelection`**. In the macro, this is simplified to **`Con.infoSetSelection`** by the object variable assignment
`Set Con = Elec.GetConstants`.

Subroutine StartElecNet

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

Subroutine CloseElecNet

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

Subroutine Visibility

This subroutine determines whether the ElecNet 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 display results 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 ElecNet 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 numeric form in the worksheet. A production version of the subroutine should include checks for the data format, and conversion from strings to numbers if required.

Subroutine MakePlate

This subroutine creates a plate, using data supplied in the parameter list. It is called twice by the subroutine SolveModel to create the two plates. The ElecNet 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 MakeElectrodes

This subroutine makes the two electrodes by selecting component surfaces, and sets the voltage on one electrode to the specified voltage difference between the plates.

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.

Appendix A

Field Equations and Solution

Field equations

The elementary concepts of electrostatics covered in Chapter 1 are sufficient for using ElecNet to solve practical problems. It is helpful, however, if the user also has some understanding of the basic theory given below. See Cheng [4] for a good introductory account of electrostatic theory.

The static electric field is described by the electric field strength **E,** the electric flux density **D**, and for current-flow problems, the current density **J**, which satisfy the equations

$$\mathrm{curl}\,\mathbf{E} = 0 \tag{A-1}$$

$$\mathrm{div}\,\mathbf{D} = \rho \tag{A-2}$$

$$\mathrm{div}\,\mathbf{J} = -\frac{\partial \rho}{\partial t} \tag{A-3}$$

$$\mathbf{D} = \varepsilon \mathbf{E} = \varepsilon_0 \varepsilon_r \mathbf{E} \tag{A-4}$$

$$\mathbf{J} = \sigma \mathbf{E} \tag{A-5}$$

where ρ is the charge density (charge per unit volume) which is the source of the electric field. Equation A-2 is the differential form of Gauss's law – equation 1-2 of Chapter 1.

In electrostatic problems, if charges are confined to the surfaces of conductors and insulators, so that there is no free charge within the volume of a material, equation A2 reduces to:

$$\mathrm{div}\,\mathbf{D} = 0 \tag{A-6}$$

To solve this equation, we use equation A-1 to express the electric field strength **E** in terms of the electric potential V through the equation

$$\mathbf{E} = -\mathrm{grad}\,V \tag{A-7}$$

Equation A-6 then becomes

$$\mathrm{div}\left(\varepsilon\,\mathrm{grad}\,V\right) = 0 \tag{A-8}$$

which is solved numerically by the finite-element method to determine the electric field in the device (see the section "Numerical Solution" on page 90). At the surface of an electrode, the integral form of equation A2 relates the charge to the surface integral of **D**.

For current-flow problems, within the volume of the conductor equation A-3 becomes:

$$\mathrm{div}\,\mathbf{J} = 0 \tag{A-9}$$

giving a similar equation to A-8:

$$\mathrm{div}\left(\sigma\,\mathrm{grad}\,V\right) = 0 \tag{A-10}$$

At the surface of the conductor, the integral form of equation A3 relates the conduction current, which is the rate of flow of charge, to the surface integral of **J**.

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 with the Kelvin transformation technique [5]. ElecNet 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 89, and an example is given in the case study on page 58. 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 lines of **D** or **E** are parallel to the boundary.
- The Neumann, or *field normal*, boundary condition. The direction of **E** is at right angles to any portion of the boundary with this condition. Since there is no component of **E** parallel to the boundary, there is no change in V, so the boundary is an equipotential. This is implemented in ElecNet by applying a Ground boundary ($V = 0$) or defining an electrode on the boundary.

The default boundary condition in ElecNet is flux tangential, so the entire outer boundary will become a field line unless the user specifies otherwise. For most problems, this is the best choice. For an electrostatic problem, it is equivalent to putting the model in a cavity of a material with zero permittivity, so that no electric flux can escape from the model. For a current-flow problem, it means there is no escape of electric current from the model. For an electrostatic problem, 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 (Ground) boundary condition is applied to the entire boundary, the effect is equivalent to putting the model in a conductive screening box. 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 an electric screen around the model.
- 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 field line or a line normal to field lines. These conditions can be represented by drawing a closed boundary round part of the model and imposing the flux tangential or field normal (Ground) conditions as required. In this way, only a portion of the device needs to be modeled. See the case study on the coaxial square conductors (page 41) for an example.

Devices may have a periodic structure, where the electric field conditions in one part of the device are similar to the conditions in another part. It is possible to represent such a device by a model of a small part. 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. If the field values are equal in magnitude but opposite in sign, the constraint is an *odd periodic* boundary condition.

Open boundaries

Consider a 2D electrostatic or current-flow 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. In both regions, we have:

$$\mathbf{E} = -\operatorname{grad} V \qquad \text{[A-7]}$$

For this problem, the Kelvin transformation [5] 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 ElecNet, 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 channels with an open boundary (page 58), and a scripting example is given on page 77.

Numerical solution

Introduction

The core of ElecNet is a powerful technique for solving the electrostatic 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.

ElecNet employs the finite-element method [6] to solve the 2D form of equation A-8 or A-9 for the electric 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-8 or A-10 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 dielectric 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 dielectric materials are present, the permittivity ε depends on the local value of E. Equation A-8 or A-10 is solved as follows:

- Constant values of permittivity are chosen for each element, from the initial point of the material ε / E curve.
- The resulting linear equations are solved numerically for the electric potential, using the semi-iterative *conjugate gradient* method.
- The field strength values are calculated from the potential, and these results are used to calculate new values for the element permittivities.
- The process is repeated until the element permittivity 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% appears to be too large. A value at least 10 times smaller may be required for accurate results.

Method of permittivity calculation

For 3D problems, ElecNet gives a choice of the Newton-Raphson method or successive substitution for calculating updated element permittivity 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 permittivity 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.

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 92), 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.

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 E from one element to the next. ElecNet 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 electric 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 capacitance, a more accurate solution may be required, and the change in the stored electric 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, ElecNet 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 ElecNet 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 ElecNet help. For most purposes, the user is recommended to avoid using edge subdivision because the other methods are more effective.

Appendix B

Energy, Force and Capacitance

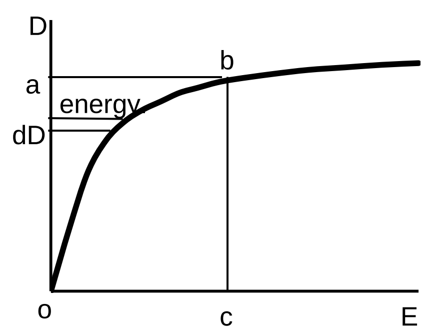
Stored energy

Definitions

When an electric field is established in a device by increasing the potentials on electrodes from zero to some final value, energy must be supplied. This energy is considered to be stored in the electric field; if hysteresis is neglected, it can be recovered when the potentials are reduced to zero.

The diagram below shows part of the *D*-*E* curve for a non-linear dielectric material. If point *b* represents the final electric condition in a particular part of the material, then it may be shown [7] that the energy density, or energy stored per unit volume, is given by:

$$w = \int \mathbf{E} \cdot \mathbf{dD} \quad [\mathrm{J/m^3}] \tag{B-1}$$



The energy density is the area between the *D*-*E* curve and the *D* axis, which is the area *oab* in the diagram. The total energy stored in the device is just the integral of the energy density over the volume:

$$W = \int w\,dv = \int\left(\int \mathbf{E} \cdot \mathbf{dD}\right)dv \quad [\mathrm{J}]. \tag{B-2}$$

When a device is solved in ElecNet, the stored energy value displayed in the Energy tab of the Post-processing bar is the values of *W* given by equations B-2.

Applications

The usual definition of capacitance is the charge per volt (see page 95). If the stored energy is calculated in the usual way as $½CV^2$, this will differ from the stored energy *W*; in fact, it is equal to the equivalent linear energy

$$W_{lin} = \int ½\mathbf{D} \cdot \mathbf{E}\,dv \quad [\mathrm{J}] \tag{B-3}$$

which is greater than *W*. If a precise value of the stored energy is required, for example when energy is dumped from a capacitor, then the value of the stored energy *W* will be a better estimate than the value of $½CV^2$.

Force calculation

Maxwell stress

In principle, the electrostatic force can be calculated from the Maxwell stress concept [7] which gives the stress, or force per unit area, directly in terms of the electric field strength. If E_n and E_t are the components of field strength normal and tangential to a surface, and σ_n and σ_t are the corresponding components of stress, then:

$$\sigma_n = ½\varepsilon_0(E_n^2 - E_t^2) \quad \text{(B-4)}$$

$$\sigma_t = \varepsilon_0 E_n E_t \quad \text{(B-5)}$$

If 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 conductors and dielectric materials.

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 that 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.

ElecNet avoids this difficulty with the Maxwell stress method by implementing a method based on virtual work. This requires no skill on the part of the user, and it usually gives accurate values for the force and the torque.

Forces on bodies

ElecNet calculates the forces and torques on all the bodies in a device automatically. A body is defined as a set of connected regions completely surrounded by the special material AIR.

Virtual air shells

Two problems can arise in the calculation of force: ElecNet does not calculate the force on a component when the entire component has been defined as an electrode, and the singularity in the value of E at external corners (see page 35) may affect the accuracy of force calculation. A solution is to surround such components with a thin layer of Virtual Air. This material behaves as air for the field solution, but it is treated as part of the component for determining the force. The technique can be used to determine the force on the boundary, which enables force balance to be verified when there is a single electrode.

Capacitance calculation

Two terminals

If a system has two electrical terminals, specified by two electrodes or a single electrode and ground, the magnitude of the charge on each electrode is given by

$$Q = CV \tag{B-6}$$

where C is the capacitance and V is the voltage between the terminals. The capacitance is therefore given by

$$C = \frac{Q}{V} \tag{B-7}$$

where Q is the charge calculated from the field solution with equation 1-2:

$$Q = \oint \mathbf{D} \cdot \mathbf{da} \tag{B-8}$$

The energy stored in the capacitance is given by

$$W = \tfrac{1}{2} C V^2 \tag{B-9}$$

Thus, the capacitance can also be determined from

$$C = \frac{2W}{V^2} \tag{B-10}$$

where W is the stored energy calculated from the field solution with equation B-2:

$$W = \int w\,dv = \int \left(\int \mathbf{E} \cdot \mathbf{dD} \right) dv \quad \text{[J].} \qquad \text{[B-2]}$$

Multiple electrodes and ground

If a system has several different electrodes maintained at different voltages with respect to ground, it is necessary to consider the capacitance between each electrode and ground, and between pairs of terminals. The charges on the electrodes will be a linear combination of the electrode voltages, so we may put

$$\mathbf{Q} = \mathbf{c}\,\mathbf{V} \tag{B-11}$$

where $\mathbf{V}$ is the column vector of voltages of the electrodes with respect to ground, $\mathbf{Q}$ is the column vector of charge values, and $\mathbf{c}$ is a square matrix of coefficients. For example, with two electrodes and ground, we have:

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{bmatrix} \tag{B-12}$$

$$\mathbf{Q} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} c_{11}V_1 + c_{12}V_2 \\ c_{12}V_1 + c_{22}V_2 \end{bmatrix} \tag{B-13}$$

The elements of $\mathbf{c}$ can be determined by finding the values of the elements of $\mathbf{Q}$ when the elements of $\mathbf{V}$ are successively set to 1, with all other elements set to zero. In the above example, we have:

$$c_{11} = [Q_1]_{V_1=1}, \ c_{12} = [Q_2]_{V_1=1}, \ c_{21} = [Q_1]_{V_2=1}, \ c_{22} = [Q_2]_{V_2=1} \tag{B-14}$$

The RLC calculator determines the capacitance values in an equivalent network where there is a capacitance C_{i0} between electrode i and ground, and a capacitance C_{ij} between electrode i and electrode j. In the above example, the charges are given by

$$\mathbf{Q} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = \begin{bmatrix} C_{10}V_1 + C_{12}(V_2 - V_1) \\ C_{20}V_2 + C_{12}(V_1 - V_2) \end{bmatrix} = \begin{bmatrix} (C_{10} - C_{12})V_1 + C_{12}V_2 \\ C_{12}V_1 + (C_{20} - C_{12})V_2 \end{bmatrix} \quad \text{(B-15)}$$

The coefficients in B-13 and B-15 are related by

$$C_{12} = c_{12} = c_{21},\ C_{10} = c_{11} - c_{12},\ C_{20} = c_{22} - c_{21} \quad \text{(B-16)}$$

References

1 W.R. Hayt, Engineering Electromagnetics. New York: McGraw Hill, 4th edition, 1981.

2 M.A.R. Gunston, Microwave Transmission Line Impedance Data. London: Van Nostrand, 1972.

3 B. Popovic, Introductory Engineering Electromagnetics. Addison Wesley, 1971.

4 D.K. Cheng, Fundamentals of Engineering Electromagnetics. Addison-Wesley, 1993.

5 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.

6 P.P. Silvester and R.L. Ferrari, Finite Elements for Electrical Engineers, 2nd ed. Cambridge University Press, 1990.

7 J. A. Stratton, Electromagnetic Theory. New York: McGraw Hill, 1941.