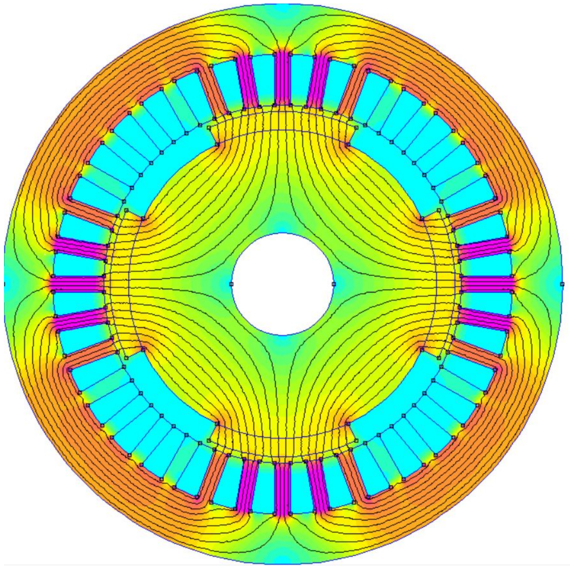
L2a: Finite-Element Method

gregheins@ieee.org

Why

- While analytical methods such as the "Reluctance Method" can be used for simple circuits, for complex analysis we need to break the problem down into more than just several rectangles or cylinders.
- Finite elements can be used to break complex shapes into many small particles.
- Increased computing power has allowed these particles to be increasingly smaller.



Energy Conservation and Functional Minimisation

Why - Energy Conservation and Functional Minimization

We need a way to formulate an equation that can be solved simultaneously at every discrete point in the problem using matrix methods .

The equation used in static structural analysis is based on Hooke's law $\mathbf{F} = \mathbf{Kx}$ where K is known as the "stiffness matrix".

We need a similar equation for electromagnetic problems.

How - Energy Conservation

- Conservation of energy
  - Energy input using current density (J)
  - Stored as flux density (B)

$$W_{\text{in}} = W_{\text{stored}} \quad (4.1)$$

$$\frac{1}{2}\int \mathbf{J} \cdot \mathbf{A} dv = \int \frac{B^2}{2\mu} dv \quad (4.2)$$

- Consider the “Functional”

$$F = W_{\text{stored}} - W_{\text{input}} \quad (4.4)$$

$$F = \int \left[\frac{B^2}{2\mu} - \frac{1}{2}\int \mathbf{J} \cdot \mathbf{A} \right] dv \quad (4.5)$$

How - Functional Minimisation

$$\frac{\partial F}{\partial A} = 0 \quad (4.6)$$

$$\frac{\partial}{\partial A} \int \frac{B^2}{2\mu} dv = \int J dv \quad (4.7)$$

Rather that solving the equation directly, we look to find where the derivative of the "Functional" is zero (minimum)

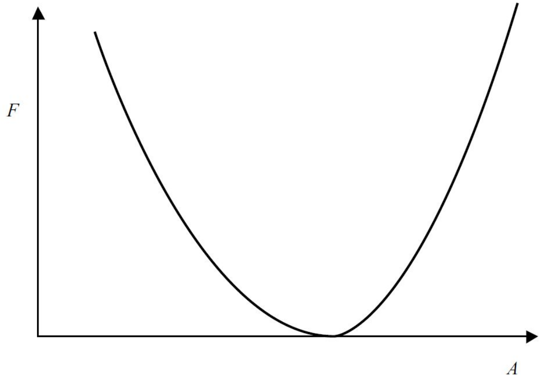
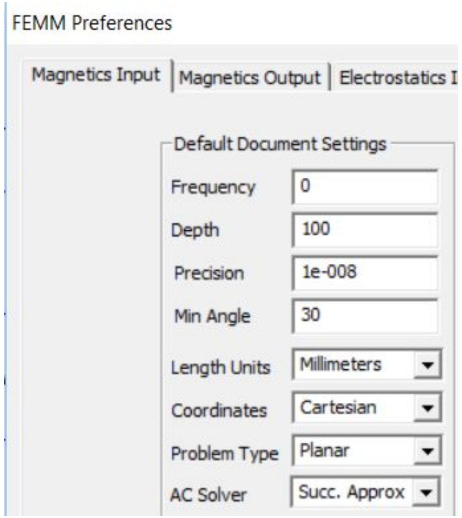


FIGURE 4.1 Functional F with a minimum having zero slope (derivative).



Test for understanding

What variable is the source of input energy?

What variable is used to define the stored energy?

Triangular Elements for Magnetostatics

Why - Triangular Elements

We need to break a complex problem into "discrete" elements.

There are many options for the shape of these elements and many arguments about which are the best to use.

The most versatile are triangles as they can approximate any shape. We will use these in this course.

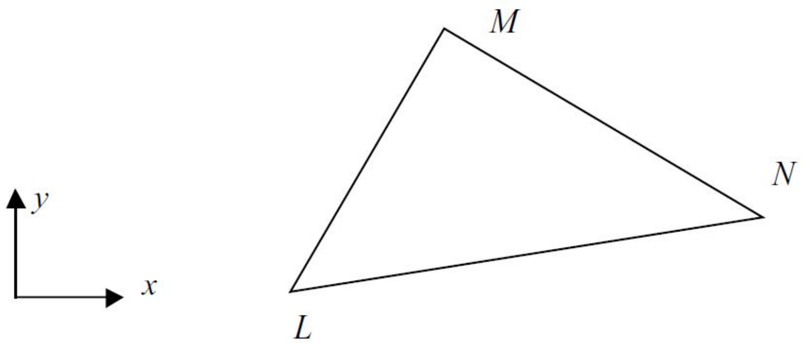


FIGURE 4.2 Triangular finite element lying in the *xy* plane.

How

Assume a "Shape Function" that will define the value of the function ($\mathbf{A}$) for any value of $\mathbf{x}$ and y within the element.

The value of choosing this particular function is that the coefficients can be found that are independent of the values at the nodes $\mathbf{A_k}$.

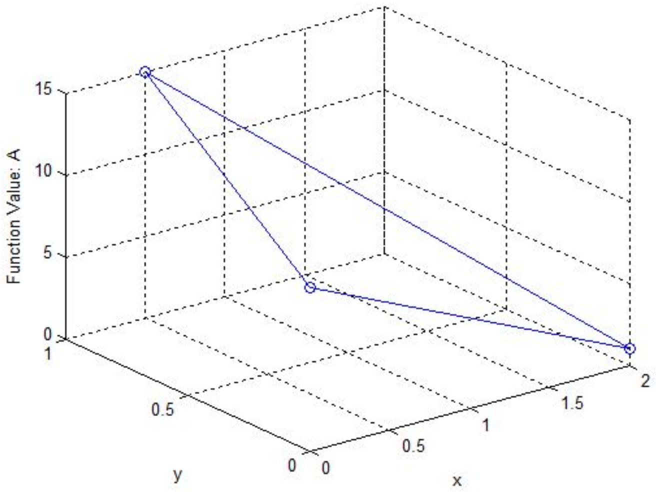
These coefficients can be found for any element by knowing the $\mathbf{x}$ and $\mathbf{y}$ coordinates.

$$A(x, y) = \sum_{k=L,M,N} [A_k(a_k + b_k x + c_k y)] \quad (4.8)$$

$$\begin{pmatrix} a_L & a_M & a_N \\ b_L & b_M & b_N \\ c_L & c_M & c_N \end{pmatrix} = \begin{pmatrix} 1 & x_L & y_L \\ 1 & x_M & y_M \\ 1 & x_N & y_N \end{pmatrix}^{-1} \quad (4.9)$$

Example - Triangular Elements

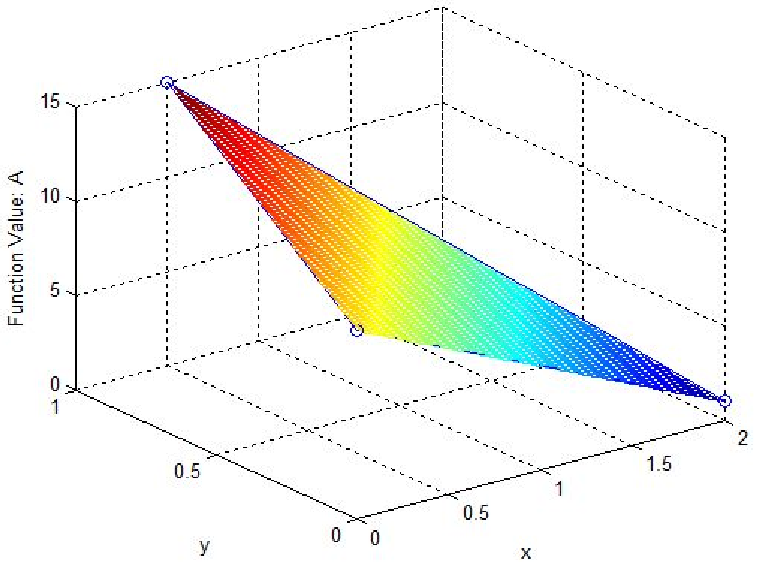
$$\begin{pmatrix} a_L & a_M & a_N \\ b_L & b_M & b_N \\ c_L & c_M & c_N \end{pmatrix} = \begin{pmatrix} 1 & x_L & y_L \\ 1 & x_M & y_M \\ 1 & x_N & y_N \end{pmatrix}^{-1} \quad (4.9)$$




```
close all; clear all; clc;
xL = 0; yL = 0; xM = 0.5; % defin coords
yM = 1; xN = 2; yN = 0; % define coords
AL = 10; AM = 15; AN = 1; % define function

xv = [xL xM xN xL];
yv = [yL yM yN xL];
Av = [AL AM AN AL]; % for plotting

figure
plot3(xv, yv,  Av,'-o');
grid on; hold on
xlabel('x');
ylabel('y');
zlabel('Function Value: A')

Mxy = [ 1 xL yL; 1 xM yM; 1 xN yN];
Mab = inv(Mxy);  % equation 4.9
aL = Mab(1,1); aM = Mab(1,2); aN = Mab(1,3);
bL = Mab(2,1); bM = Mab(2,2); bN = Mab(2,3);
cL = Mab(3,1); cM = Mab(3,2); cN = Mab(3,3);
```

Example - Triangular Elements

Using the coefficients the value of the function can be defined at any point in the element




```
xPlot = linspace(min([xL xM xN]), max([xL xM
xN]),80);
yPlot = linspace(min([yL yM yN]), max([yL yM
yN]),90);

for xIndex = 1:length(xPlot)
    for yIndex = 1:length(yPlot)
        x = xPlot(xIndex);
        y = yPlot(yIndex);
        if inpolygon(x,y,xv,yv);
            A(yIndex,xIndex) = AL*(aL+bL*x+cL*y)+ ...
                AM*(aM+bM*x+cM*y)+AN*(aN+bN*x+cN*y);
        else
            A(yIndex,xIndex) = nan;
        end
    end
end
mesh(xPlot, yPlot, A)
```

Test for understanding

Why are we using triangular elements in this course?

Were the coefficients we determined for the shape function dependant on:

A. The function values at the nodes
B. The x and y coordinates at the nodes
C. Both the function values and the x and y coordinates

Matrix Equation

Why - Matrix Equation

Now that we have determined a shape function and a “Functional” that we are going to minimise we can define a matrix equation.

This matrix equation is what we will solve to ensure that the equations are valid at every element and node.

How

Using the “Functional” and the “Shape Function” we can determine an equation relating the input current density (J) to the magnetic vector potential (A).

See the reference in the book for the derivation

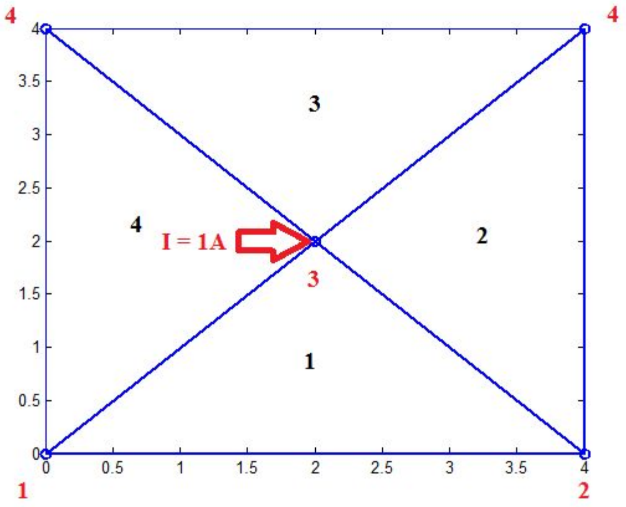
$$(S/\mu)\begin{pmatrix} (b_L b_L + c_L c_L) & (b_L b_M + c_L c_M) & (b_L b_N + c_L c_N) \\ (b_M b_L + c_M c_L) & (b_M b_M + c_M c_M) & (b_M b_N + c_M c_N) \\ (b_N b_L + c_N c_L) & (b_N b_M + c_N c_{ML}) & (b_N b_N + c_N c_N) \end{pmatrix} \begin{bmatrix} A_L \\ A_M \\ A_N \end{bmatrix} = (S/3)\begin{bmatrix} J \\ J \\ J \end{bmatrix} (4.12)$$

$$[K]\{A\} = \{J\}(4.13)$$

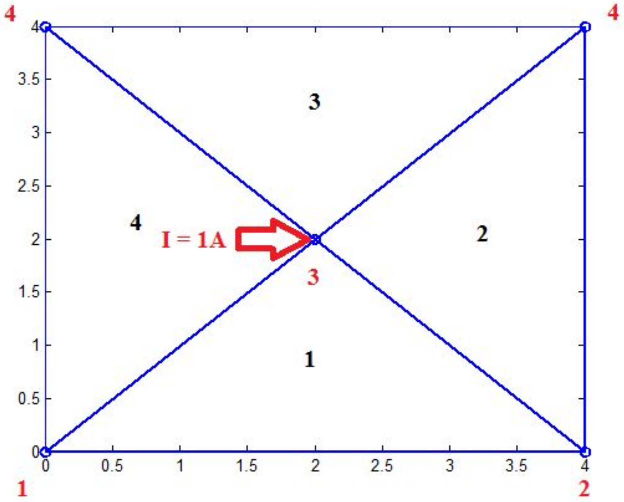
Example - Matrix equation

For a square with a point current I = 1A in the centre solve for A.

1. Find the K matrix for each element
2. Combine the individual element matrices into a total K matrix
3. Solve the matrix equation: $[K]\{A\} = \{J\}(4.13)$
4. Plot A along the diagonal contour (1,3,4)



Matrix equation for each element




```
%% Four element input data
x = [0 4 2 0 4];
y = [0 0 2 4 4];
J = [0 0 1 0 0]';
T = [1 2 3; 2 3 5;  3 4 5; 1 3 4];
Node_diag = [1,3,5];
mu = 1.257e-6;  % material properties


%% in function
S    =    abs(xL*(yM-yN)+xM*(yN-yL)+xN*(yL-yM))/2;
    Mxy = [ 1 xL yL; 1 xM yM; 1 xN yN];
    Mab = inv(Mxy);  % equation 4.9
    bL = Mab(2,1); bM = Mab(2,2); bN = Mab(2,3);
    cL = Mab(3,1); cM = Mab(3,2); cN = Mab(3,3);
    K = [   bL*bL+cL*cL     bL*bM+cL*cM     bL*bN+cL*cN;
            bM*bL+cM*cL     bM*bM+cM*cM     bM*bN+cM*cN;
            bN*bL+cN*cL     bN*bM+cN*cM     cN*cN+cN*cN];
    K = K*S/mu;
```

Combine element K matrices

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} 1352 & 1750 & 2148 & 0 & 0 \\ 1750 & 2307 & 2864 & 0 & 0 \\ 2148 & 2864 & (3580 + 2704) & 3500 & 4296 \\ 0 & 0 & 3500 & 4614 & 5728 \\ 0 & 0 & 4296 & 5728 & 7160 \end{pmatrix} \end{array} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} = \begin{bmatrix} 1.333 \\ 1.333 \\ 4 \\ 2.667 \\ 2.667 \end{bmatrix} \quad \text{(E4.1.5)}$$

```
T = [1 2 3; 2 3 5;  3 4 5; 1 3 4];

% assemble K matrix
for t = 1:length(T)
    K_tot(T(t,:),T(t,:)) = K_tot(T(t,:),T(t,:)) + FEA_Find_K_Fn(x(T(t,:)),y(T(t,:)),mu);
end
```

Apply boundary conditions

Without applying boundary conditions, the matrix will be ill conditioned and unable to be solved.

A simple method is to zero out the rows and columns corresponding to the nodes which are set to zero and put a 1 on the diagonal for these equations [1]

	1	2	3	4	5
1	1	0	-7.9554e+05	0	0
2	0	1	-7.9554e+05	0	0
3	-7.9554e+05	-7.9554e+05	3.9777e+06	-7.9554e+05	-7.9554e+05
4	0	0	-7.9554e+05	1	0
5	0	0	-7.9554e+05	0	1

[1] Finite Element Analysis of Electrical Machines, Sheppard Salon (1995)

Solve Matrix Equation

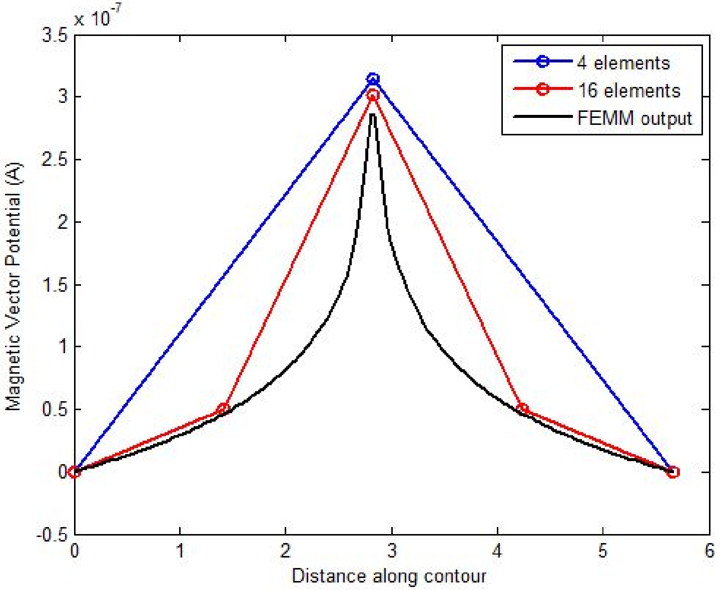
We now know the matrix K_tot

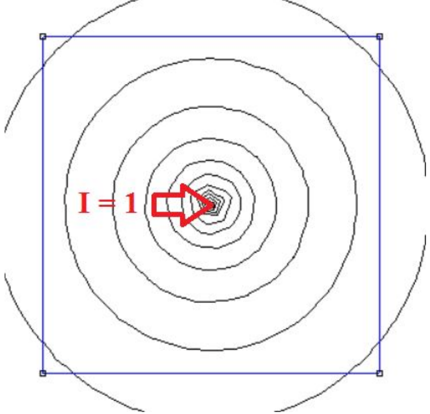
And we know the vector J

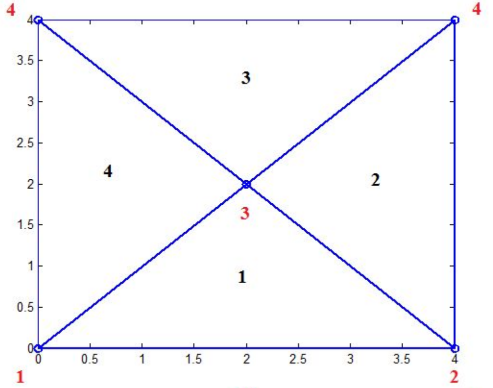
We can solve for A at every node

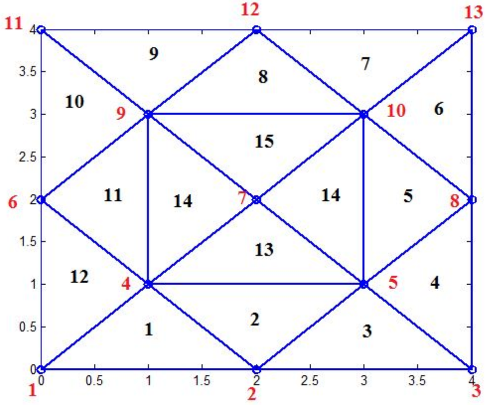
```
A = K_tot\J;
```

Plot A along the diagonal









Test for understanding

Do you have different coefficients for the shape function for every element?

What error will you get if you have not applied boundary conditions?

Finite-Element Models

Why

It is important to have a structured approach to finite element modelling

it is usually easier to ensure each step is completed rather than searching for issues once errors arise

How

(1) **Geometry, subdivided into finite elements.** All 2D or 3D geometry must be specified, and the user and/or the software must subdivide it into finite elements. Geometry specification is difficult for many real-world devices of complicated shape. For example, Figure 4.3 shows a magnetic actuator with 2D geometry. If the geometry already exists in drawing software, then some finite-element software will accept it. However, either the finite-element software or the user must subdivide all geometric regions, including air for magnetic fields, into finite elements, such as those shown in Figure 4.4.

(2) **Materials.** For electromagnetic fields, the three material properties of Chapter 2 may be required: permeability, conductivity, and permittivity. Recall that for nonlinear magnetic materials, the *B–H* curve is required. For magnetostatics, (4.12) shows that conductivity and permittivity are not required.

(3) **Excitations.** For magnetic fields, the excitations or sources input are current densities and/or currents according to (4.12). As will be seen in later chapters, voltages and/or permanent magnets can be used instead if desired.

(4) **Boundary conditions.** On the outer boundary of the region analyzed, boundary conditions are required. For example, since Figure 4.4 is a one-half model

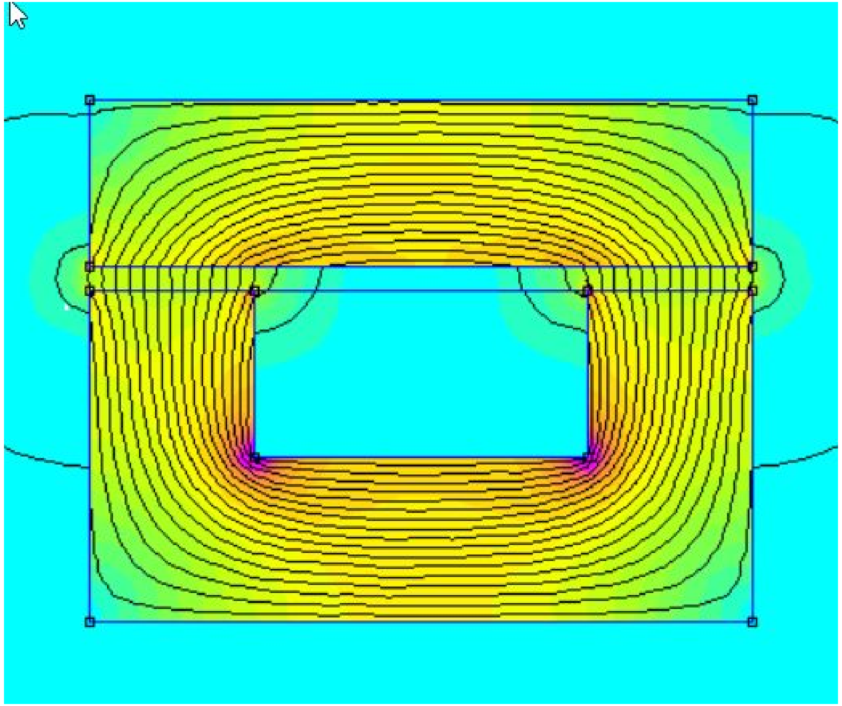
Mesh Control

Why

One of the main challenges of FEA is to balance accuracy with execution time.

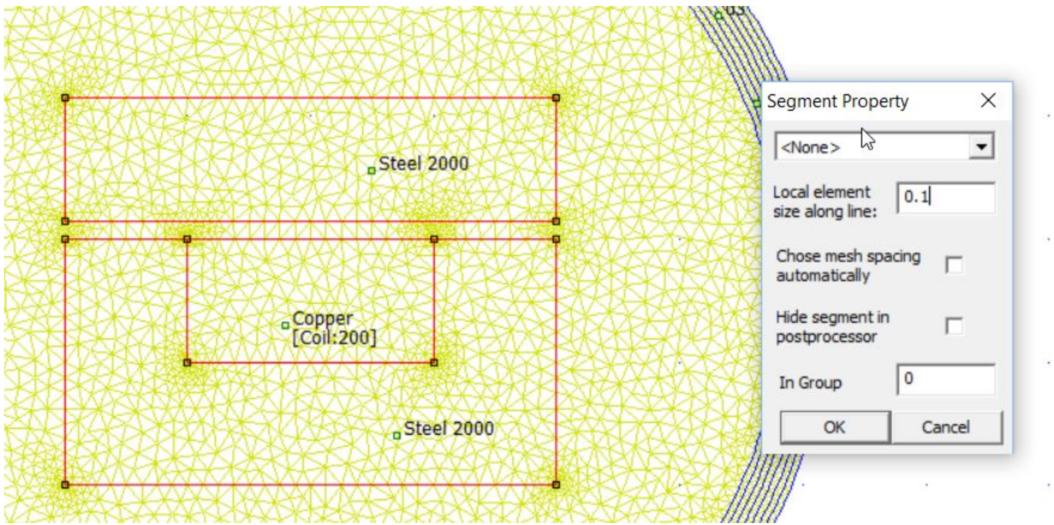
Controlling the mesh allows you to achieve this balance

Careful mesh controls allow a different accuracy/ time balance in different parts of the problem.



How

Run multiple studies with different meshes and compare results.



In FEMM:

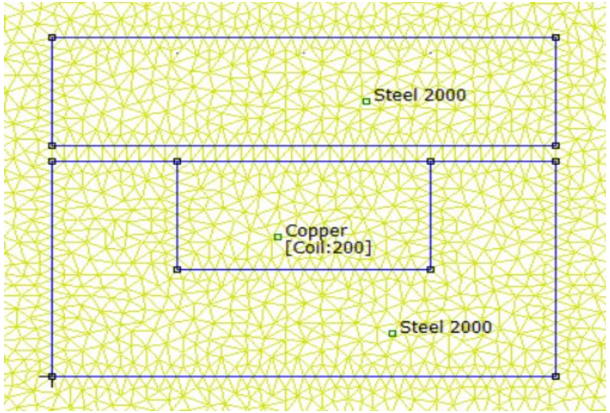
- Change the “Local element size along line”
- It can be useful in FEMM to use Edit > select region to select multiple lines at the same time

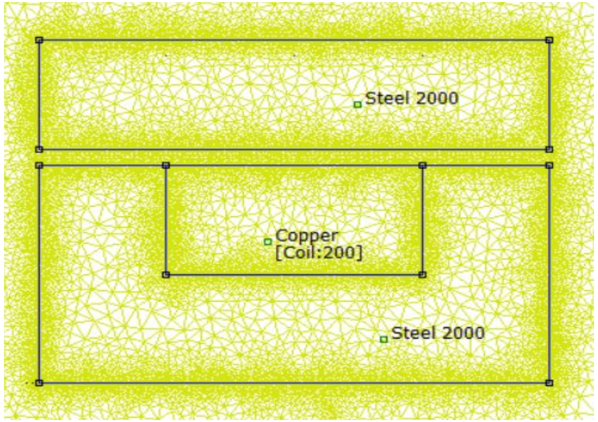
Example: Mesh controls

Using an example with a “clapper” solenoid:

Change the “Local element size along line” from 1mm to 0.01mm

Compare the results for the force on the clapper and the flux density at a particular point

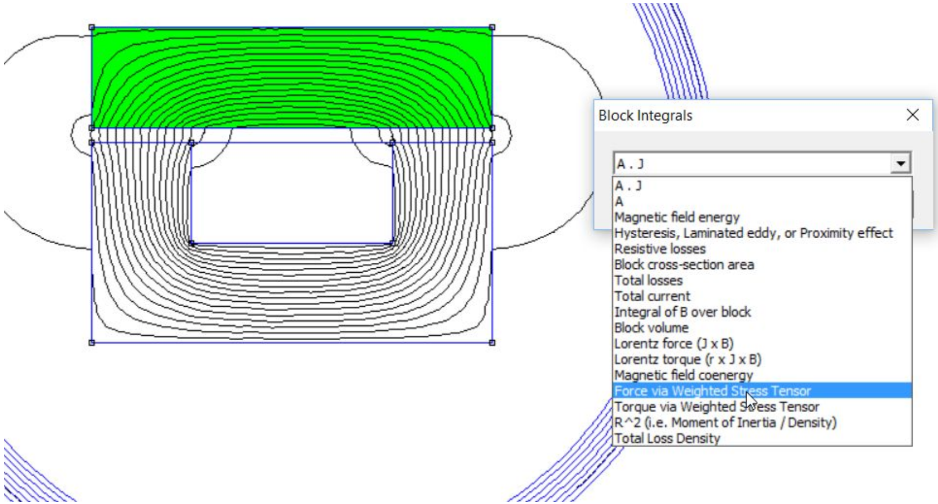


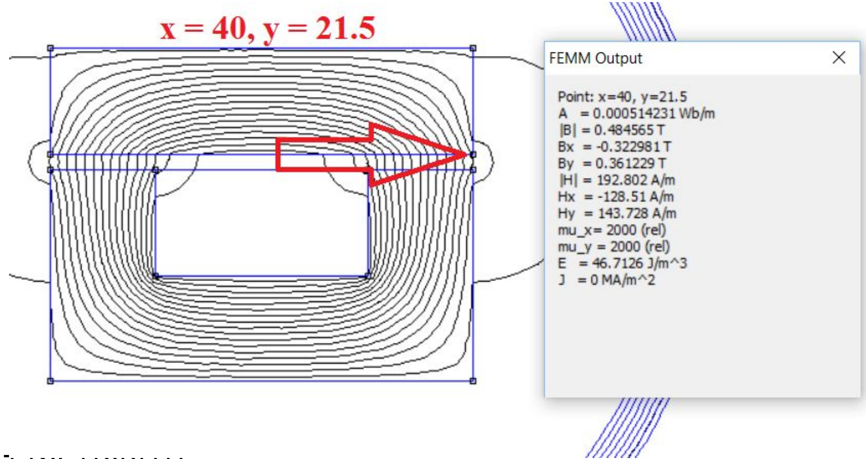


Example: Mesh controls

Each time the element length was changed:

1. The Force on the “clapper” was measured
2. The Flux density at the bottom right hand corner of the clapper was measured (using “tab” in the and typing in the numbers). This also requires: view> output window





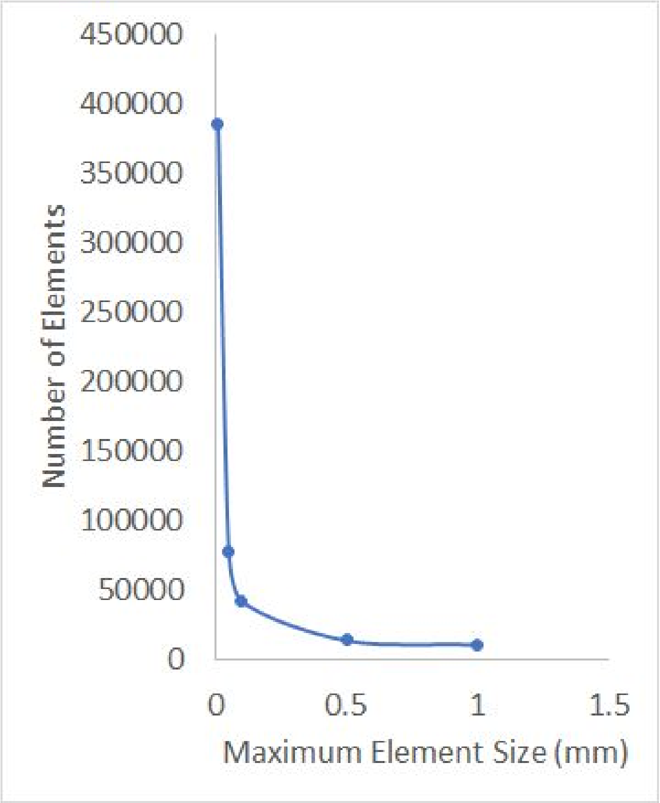
Results: Mesh convergence

As the element length decreases, there is a sharp rise in the number of elements.

About 400000 elements was the most that my computer could cope with for FEMM.

In 3D FEA, the number of elements increases much more quickly.

The number of elements is a good indication of execution time.



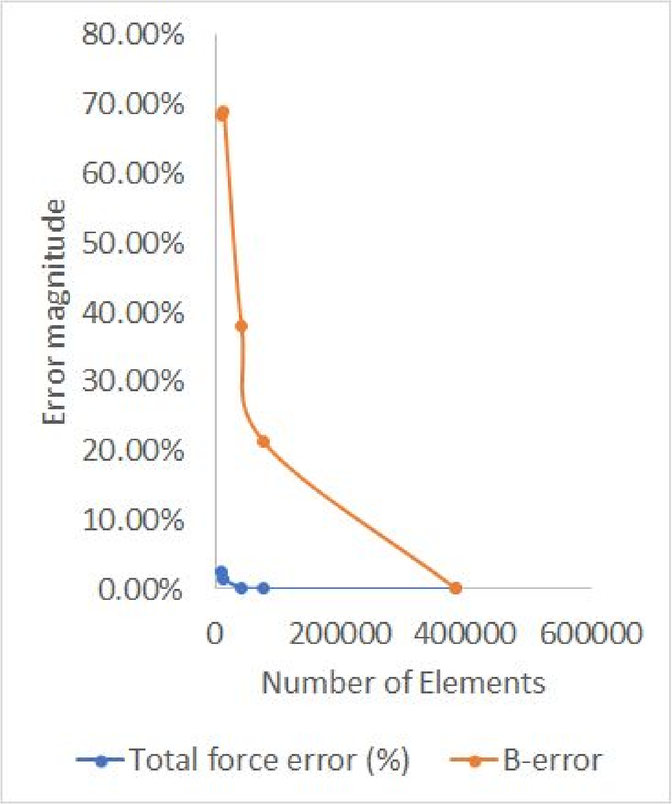
Results: Mesh Convergence

We assume that the solution with the most elements is the correct one.

That allows us to determine the "relative error" when we use less elements.

Of the two measurements we were interested in, the total force converged much more quickly that the B at a point.

The force is an "integral" measurement so much less susceptible to mesh variation than a point value.



Test for understanding

Write down the four things that will need to define for a standard FEA problem.

Will you need a finer mesh to find an accurate value for the total force on a solenoid or to find the value of B at a particular point?

Summary

FEA is a powerful tool for both simple and complex geometries.

Regardless of the software used, you will need to define:

- Geometry
- Materials
- Excitation
- Boundary Conditions

Careful use of Mesh controls is required to balance execution time and accuracy.