ET4375 Course Introduction to Finite Elements

One Dimensional Model of a Fault Current Limiter

The objective of this assignment is to implement a one-dimensional model of the fault current limiter.

Problem Definition

To build the one-dimensional model, we consider solving the Poisson equation on the computational domain $\Omega = (-0.2\,\mathrm{m}, 0.2\mathrm{m})$ subdivided into the following four parts

$$\Omega = \Omega_{air} \cup \Omega_{core} \cup \Omega_{coil-up} \cup \Omega_{coil-down} \tag{1}$$

where Ω_{air}, Ω_{core}, $\Omega_{coil-up}$ and $\Omega_{coil-down}$ denote the domain representing the air, ferromagnetic core, coil with upward facing current and coil with downward facing current, respectively. These individual subdomains are defined as follows

$$\Omega_{air} = (-0.2\,\mathrm{m}, -0.057\,\mathrm{m}) \cup (-0.012\,\mathrm{m}, 0.012\,\mathrm{m}) \cup (0.057\,\mathrm{m}, 0.2\,\mathrm{m}) \tag{2}$$

$$\Omega_{core} = (-0.037\,\mathrm{m}, -0.012\,\mathrm{m}) \cup (0.012\,\mathrm{m}, 0.037\,\mathrm{m}) \tag{3}$$

$$\Omega_{coil-up} = (-0.057\,\mathrm{m}, -0.037\,\mathrm{m}) \tag{4}$$

$$\Omega_{coil-down} = (0.037\,\mathrm{m}, 0.057\,\mathrm{m})\,. \tag{5}$$

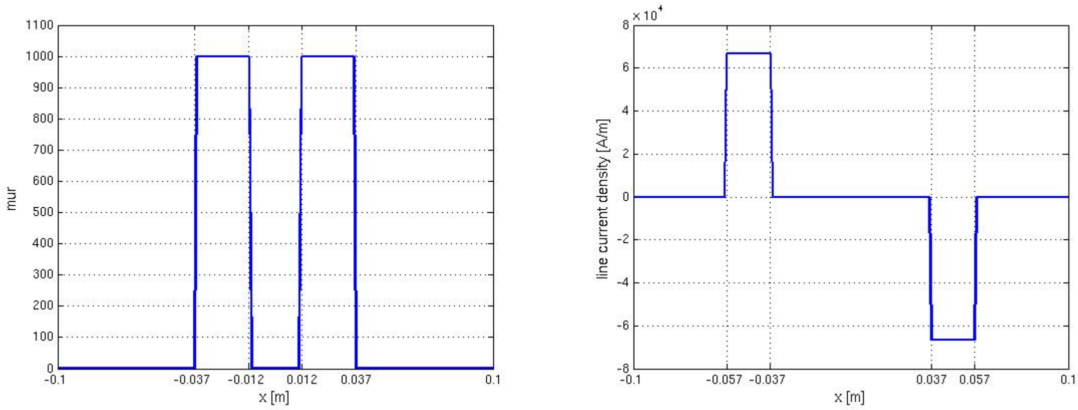
This geometry definition allows us to define the relative magnetic permeability $\mu_r(x)$ (dimensionless) and the line current density $g(x)$ (in A/m) as follows

$$\mu_r(x) = \begin{cases} 1000 \text{ if } x \in \Omega_{core} \\ 1 \quad\;\; \text{if } x \in \Omega_{air} \cup \Omega_{coil-up} \cup \Omega_{coil-down} \end{cases} \tag{6}$$

and

$$g(x) = \begin{cases} 7 \cdot 10^4 \mathrm{A/m} & \text{if } x \in \Omega_{coil-up} \\ -7 \cdot 10^4 \mathrm{A/m} & \text{if } x \in \Omega_{coil-down} \\ 0 & \text{if } x \in \Omega_{air} \cup \Omega_{core} \end{cases} \tag{7}$$

The magnetic permeability is then defined as $\mu(x) = \mu_0 \mu_r(x)$ (in N/A^2). The functions $\mu_r(x)$ and $g(x)$ are shown in Figure 1.



(a) Relative permeability $\mu_r(x)$

(b) Line current density $g(x)$.

Figure 1: Definition of the relative magnetic permeability (left) and the line current density (right).

Our objective is then to solve the differential equation

$$-\frac{d}{dx}\left(\frac{1}{\mu}\frac{du}{dx}\right) = g(x) \text{ for } x \in \Omega \tag{8}$$

supplied with the Dirichlet boundary conditions

$$u(x = -.2\,\mathrm{m}) = 0 \text{ and } \quad u(x = .2\,\mathrm{m}) = 0. \tag{9}$$

The unknown function $u(x)$ here represents the z-component of the magnetic vector potential. The boundary conditions insulate the computational domain for the magnetic flux. The y-component of the magnetic flux B_y and the magnetic field H_y can then be computed as

$$B_y = \frac{d\,u}{dx} \text{ and } H_y = \mu\, B_y\,, \tag{10}$$

respectively.

Assignments

This part guides the student to the development of a finite element code for solving the above differential equation supplied with boundary conditions by guiding in the construction of a mesh, the assembly of the linear system on the mesh and solution of the linear system.

Assignment 1 Write a Matlab (or Python) routine, called *GenerateMesh.m* (or GenerateMesh.py), that returns as output a vector of size $n + 1$ of grid points x_i on Ω. Make sure that the point separating the subdomains, i.e., the points $\{-0.057, -0.037, -0.02, 0.02, 0.037, 0.057\}$ are mesh points. The mesh is thus in general non-uniform.

Assignment 2 Write the routines *SourceFct.m* and *Diffusion.m* that given the coordinate $x \in \Omega$ return as output the source function $g(x)$ and magnetic permeability $\mu(x)$, respectively. For *SourceFct.m* for instance, one can use, given a value for x as input, a logical expression of the form

$$\text{value} = \text{amplitude} * (-0.057 \leq x \leq -0.037) + (-\text{amplitude}) * (0.037 \leq x \leq 0.057) \tag{11}$$

to compute the value as output of the *SourceFct.m* in x. As similar construction can be used for *Diffusion.m*.

Assignment 3 Write a routine, called *GenerateTopology.m*, that generates an $n \times 2$ matrix called *elmat* such that the ith row of *elmat* contains the indices of the left and right node of the ith element in the global enumeration on mesh on Ω, that is

$$\begin{aligned} elmat(i,1) &= i \quad \text{for } i = 1, \ldots, n \\ elmat(i,2) &= i+1 \end{aligned} \tag{12}$$

Assignment 4 Write a routine, called *GenerateElementMatrix.m*, that given the coordinates x_i and x_{i+1} of the end points of element e_i with midpoint $x_{i+1/2} = (x_i + x_{i+1})/2$, generates the 2×2 element matrix S_{e_i} such that

$$S_{e_i} = \frac{1}{x_{i+1} - x_i} \frac{1}{\mu_{i+1/2}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \tag{13}$$

where

$$\mu_{i+1/2} = \mu(x_{i+1/2})\,. \tag{14}$$

The diffusion coefficient is thus evaluated in the midpoint of the element e_i.

Assignment 5 Write a routine, called *GenerateElementVector.m*, that given the coordinates x_i and x_{i+1} of the end point of element e_i generates the 2×1 element vector f_{e_i} such that

$$f_{e_i} = \frac{x_{i+1} - x_i}{2} \begin{pmatrix} g(x_i) \\ g(x_{i+1}) \end{pmatrix}. \tag{15}$$

Assignment 6 Write a routine, called *AssembleMatrix.m*, that assembles the element matrices S_{e_i} on each element into the global matrix $(n+1) \times (n+1)$ matrix S. To so by first initializing S to be an empty $(n+1) \times (n+1)$ matrix and subsequently performing a loop over the elements. In this loop the element matrices are generated and added to the global matrix. In this addition the connectivity of the mesh defined by the matrix *elmat* needs to be taken into account. Write therefore a triple for-loop in which

- the outermost loop indexed by $i = 1, \ldots, n$ traverses the elements;
- on each element e_i the element matrix S_{e_i} on the element e_i is computed;
- the innermost two loops indexed by $j, k = 1, 2$ traverse the nodes on ith element;
- the following statement is placed in the innermost loop

$$S(elmat(i,j), elmat(i,k)) = S(elmat(i,j), elmat(i,k)) + S_{e_i}(j,k)\,. \tag{16}$$

Assignment 7 Write a routine, called *AssembleVector.m*, that assembles the element vectors f_{e_i} on each element into the global matrix $(n+1) \times 1$ vector f. To so by first initializing f to be an empty $(n+1) \times 1$ vector and subsequently performing a loop over the elements. In this loop the element vectors are generated and added to the global vector. In this addition the connectivity of the mesh defined by the matrix *elmat* needs to be taken into account. Write therefore a double for-loop in which

- the outermost loop indexed by $i = 1, \ldots, n$ traverses the elements;
- on each element e_i the element matrix f_{e_i} on the element e_i is computed;
- the innermost loop indexed by $j = 1, 2$ traverse the nodes on ith element;
- the following statement is placed in the innermost loop

$$f(elmat(i,j)) = f(elmat(i,j)) + f_{e_i}(j)\,. \tag{17}$$

Assignment 8 Modify the first and last equation of the linear system $S\,u^u = f$ in such a way that the finite element solution $u^h(x)$ satisfied the Dirichlet boundary conditions at $x = -0.02\,\mathrm{m}$ and $x = 0.02\,\mathrm{m}$. This can be accomplished by modifying the first equation of the linear system in the following way. Modify the first row of the matrix S by setting $S(1,1) = 1$, $S(1,2) = 0$, $S(n+1,n) = 0$ $S(n+1,n+1) = 1$ and modify the first and last element of f by setting $f(1) = 0$ and $f(n+1) = 0$.

Assignment 9 Compute the finite element solution u^h using $u^h = S \backslash f$ in Matlab for various values of the number of elements n.

Assignment 10 Compute and plot the derivative $d\,u^h(x)/dx$ of the finite element solution. Give a physical interpretation of the picture you obtain. Obverse that the derivative is discontinuous in those points x in which μ is discontinuous.

Assignment 11 Compute and plot the quantity $(1/\mu)d\,u^h(x)/dx$ obtained from the the finite element solution. Give a physical interpretation of the picture you obtain. Observe that the quantify considered in continuous.