Department of Mechanical Engineering
University of Bath

ME40064: System Modelling & Simulation

Assignment 1: Static MATLAB-Based FEM Modelling

Summary

You must develop a FEM-based simulation tool to solve the static diffusion-reaction equation in a 1D mesh, which your consultancy firm can re-use for future modelling projects. This tool must be developed following good software development practice and must be validated against standard test cases, accompanied by appropriate documentary proof.

The specific tasks of this deliverable are as follows:

Part 1: Software Verification & Analytical Testing

a. Create a MATLAB function to calculate the local 2-by-2 element matrix for the diffusion operator, for any element, e_N, in the finite element mesh. **[2%]**

   The inputs to the function are:
   - the diffusion coefficient, *D*
   - the local element number, e_N
   - the mesh data structure (through which you can access each element's Jacobian, J and nodal positions x_0 and x_1)

   Create the mesh using the function `OneDimLinearMeshGen.m`, and store it using the mesh data structure, as provided in Tutorial 2.

   Your function should pass the unit test, `CourseworkOneUnitTest.m`, which is available from the course's Moodle page. In this script you will see a suggested name for your function, as well as the required arguments for your function.

   Include the following in your report:
   - a legible screenshot that shows your function passes the test
   - the function's source code

b. Create a MATLAB function to calculate the local 2-by-2 element matrix for the linear reaction operator, for any element, e_N, in the finite element mesh. **[3%]**

   The inputs to the function are:
   - the reaction coefficient, *lambda*

- the local element number, e_N
- the mesh data structure (through which you can access each element's Jacobian, J and nodal positions x_0 and x_1)

Create your own unit test for this function and explain in your report what your unit test has been designed to check and why these tests are sufficient to show that your function works correctly.

Include the following in your report:

- a legible screenshot that shows your function passes the test
- the function's source code
- the unit test's source code

c. Using these functions, develop a finite element solver for the static reaction-diffusion equation, using additional unit tests to verify any new functions as appropriate.

Demonstrate that the code is correct using the following two analytical test cases:

<u>Analytical Test One - Laplace's equation</u>:

Solve Laplace's equation:

$$\frac{\partial^2 c}{\partial x^2} = 0$$

for the four-element mesh of the domain between x=0 and x=1, shown in Figure 1.

i. Solve for the case of the two Dirichlet boundary conditions **[25%]**:

$$c = 2 \quad at \quad x = 0$$

$$c = 0 \quad at \quad x = 1$$

Compare your numerical solution to the analytical solution for this equation & boundary conditions, which is:

$$c = 2(1 - x)$$

How accurate is your solution? Why is this?

ii. Run your code again for Laplace's equation, this time with the following Dirichlet boundary condition and Neumann boundary condition **[5%]**:

$$\frac{\partial c}{\partial x} = 2 \quad at \quad x = 0$$

$$c = 0 \quad at \quad x = 1$$

Explain the effect that changing this boundary condition has had on the solution **[5%].**

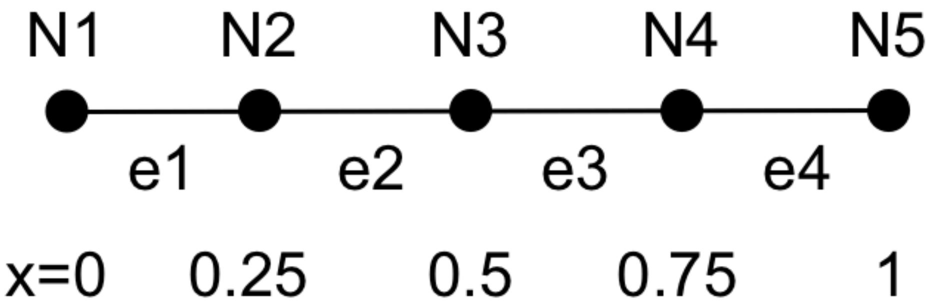

Figure 1: Uniform 4-element mesh defined between x=0 and x=1.

<u>Analytical Test Two – Diffusion-Reaction equation</u>

iii. Now check your code solves reaction terms correctly, by using it to solve the diffusion-reaction equation **[10%]**:

$$D\frac{\partial^2 c}{\partial x^2} + \lambda c = 0$$

for the following parameters:

$$D = 1, \quad \lambda = -9$$

and Dirichlet boundary conditions:

$$c = 0 \quad at \quad x = 0$$

$$c = 1 \quad at \quad x = 1$$

For these parameters & boundary conditions that equation has the following analytical solution:

$$c(x) = \frac{e^3}{e^6 - 1}\left(e^{3x} - e^{-3x}\right)$$

Use several different mesh resolutions and check that your solution approaches the analytical value as you increase the number of elements.

Part 2: Modelling & Simulation Results

Thermophoresis is a phenomenon in which particles of different types diffuse differently through a medium depending on the local temperature gradient that they are exposed to. The particles diffuse at a rate that *is negatively proportional to the local temperature gradient* i.e. the steeper the negative temperature gradient, the faster the particles will diffuse. In biomaterial processing this effect can be harnessed to create a spatially varying concentration of these particles, which causes a spatially varying stiffness to be created in the biomaterial. This is useful for growing cell cultures under biologically realistic conditions.

In Part 2 you will use your code to model steady-state heat transfer through a material that is filled with small diameter heating channels. By varying the temperature of the heating liquid, the flow rate, or the spatial distribution of temperature in these channels, different temperature profiles can be obtained for the thermophoresis-based manufacturing process. Your task is to characterise the behaviour of this system for its different parameters.

Figure 2 shows a cross-section of the material:

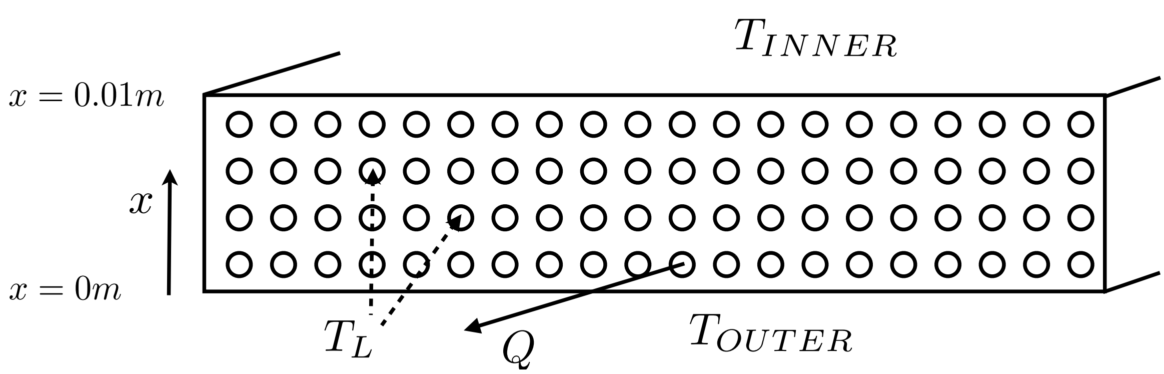

Figure 2: Cross-section of the material

Given that the boundary conditions are uniform in all directions, we can reduce the problem to a 1D approximation for heat transfer through the material:

$$k\frac{\partial^2 T}{\partial x^2} + Q(T_L - T) = 0$$

a. Use your code to solve this model and hence compute the temperature distribution T (in Kelvin) for different combinations of values of Q (the liquid flow rate) and T_L (liquid temperature in Kelvin) in the ranges **[20%]**:

   Q = 0.5 to 1.5
   T_L = 294.15 K to 322.15 K (i.e. 21 to 49 deg C)

   The other material parameters and boundary conditions are fixed as:

   k = 1.01e-05

   Dirichlet BC at *x* = 0 m is: T_{OUTER} = 323.15 K (50 deg C)
   Dirichlet BC at *x* = 0.01 m is: T_{INNER} = 293.15 K (20 deg C).

   i. Plot the results from your parameter space study and explain the effect each parameter has on both the temperature distribution and on the gradients of the temperature.

   ii. If some particles are initially placed at *x*=0 and are allowed to diffuse for a short amount of time, describe how you think their distribution, and hence the variation in stiffness, might change with Q and T_L, based on the effect these have on the temperature gradients in the material?

   iii. Assess the effect of different mesh resolutions and explain how this affects the accuracy of the results and your conclusions.

b. It is suggested that varying the temperature of the liquid from one channel to the next could provide greater control of the temperature distribution. This variation in temperature and its effect on heating performance can be represented by the following linear function of *x* for the source term **[10%]**:

$$T_L(1 + 4x)$$

   This means the equation to be solved is now:

$$k\frac{\partial^2 T}{\partial x^2} + Q(T_L(1 + 4x) - T) = 0$$

Remember that *x* can be written as the sum of the two basis functions, within each local element:

$$x = x_0\psi_0 + x_1\psi_1$$

i. Derive the analytical expression for the local element vector that represents this modified source term and include it in your report
ii. Implement the modified source term in your code and investigate & explain how it changes the behaviour of this system

Presentation [10%]

- Clear graphical presentation of equations, text, and results
- Quality of English, proper use of references, figures, and tables

Code Quality [10%]

- Correctness, elegance, and readability, of MATLAB code
- Evidence of software verification and other good development practice

SUBMISSION GUIDELINES

Structure your report as a set of answers to these questions – there is no requirement to write this in a lab report format. You also do not need to re-explain the entirety of the finite element method. However, your report must be self-contained and therefore must not assume that the reader knows the content in this document.

- You **must** include all your Matlab source code as **text** in the Appendices – failure to do so will cause you to lose marks. **Do not** paste your code into the document as an OLE item or as an image.
- **Do not** upload archived/zipped/compressed folders of these source files.
- **Do not** use MATLAB's symbolic algebra toolbox.
- Your code should use meaningful variable names and include comments, in line with good practice.
- Word limit of 2000 words (not including source code).

Submit your work using the online submission function on the unit's Moodle page.

Deadline: 4pm on Friday 8th November 2019.