Department of Mechanical Engineering
University of Bath

ME40064: Systems Modelling & Simulation

Assignment 1: Static Matlab-Based FEM Modelling

Summary

You must develop a finite element method based simulation tool can 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 to demonstrate the successful completion of this deliverable are as follows:

Part 1: Software Verification & Analytical Testing

a. Write a function to calculate the local 2-by-2 element matrix for the diffusion operator, which can calculate this matrix for an arbitrary element e_N defined between the points x_0 and x_1. The mesh should be created using the function, and mesh data structure, provided in Tutorial 2: `OneDimLinearMeshGen.m`.

   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)

   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
- the mathematical derivation of this element matrix

b. Write a unit test for a function to calculate the local element matrix for the linear reaction operator. Explain in your report the conditions that your unit test has been designed to check and why these are sufficient to show that your function works correctly.

Now write a function to calculate the local element matrix for the linear reaction operator. Again include the source code, mathematical derivation of this element matrix, and a screenshot showing that your function passes the unit test in your report.

c. Finish writing a finite element solver for the static reaction-diffusion equation, using additional unit tests to verify functions where appropriate. Check that the code is working overall using the following two analytical test cases.

Use your code to solve Laplace's equation:

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

in the domain between x=0 and x=1.

- Run your code for the four element mesh shown in Figure 1, for the case of two Dirichlet boundary conditions:

$$c = 2 \quad at \quad x = 0$$
$$c = 0 \quad at \quad x = 1$$

- Compare your numerical solution to the analytical solution for this equation, which is:

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

  Comment on the accuracy of your solution and the reason for this.

- Run your code again for Laplace's equation, this time with one Dirichlet boundary condition, and one Neumann boundary condition, defined as:

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

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

  Explain the effect that changing this boundary condition has had on the solution.

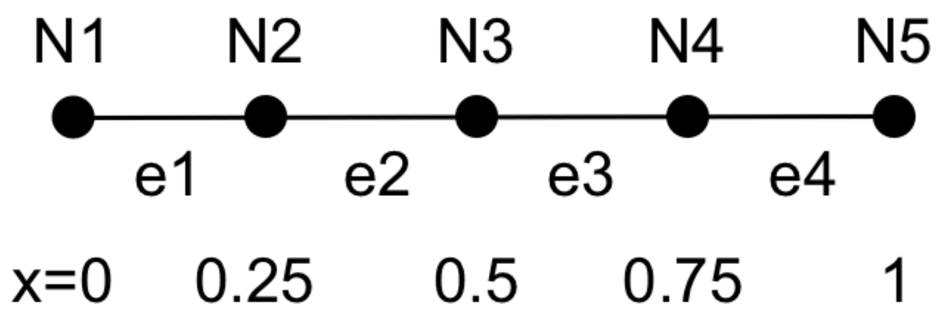

Figure 1: Equispaced, 4-element mesh between x= 0 and x=1.

d. Now check your code solves reaction terms correctly, by using it to solve the diffusion-reaction equation:

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

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 settings the 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. By controlling temperature distributions in a material, this process is harnessed to create spatially varying material properties, for example in semi-conductor manufacturing.

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, a particular temperature profile can be obtained suitable for the manufacturing process. Your task is to characterise the performance of this system for the different parameters.

Figure 2 shows a cross-section of the material:

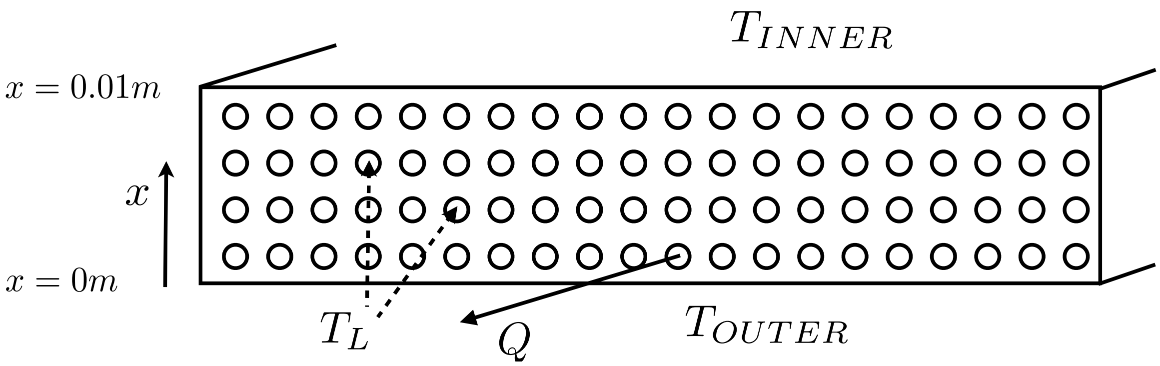

Figure 2: Cross-section of the material

Given that the boundary conditions are uniform in all directions, we reduce the problem to a 1D approximation for heat transfer through the material. The equation to describe this is:

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

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 following ranges:

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

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

Make sure to try several different mesh resolutions and explain how this affects the accuracy of the results and conclusions.

b. It is suggested that varying the temperature of the liquid in the different channels could provide greater control of the temperature distribution. This variation in temperature and its effect on heating performance can be represented by the following polynomial function of x for the source term:

$$T_L(1+4x)$$

This means the governing equation is now:

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

Remembering that *x* can be written as the sum of the two basis functions, within each local element, i.e:

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

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

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. 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. **Do not** paste your code into the document as an OLE or as an image.
- **Do not** upload zipped or compressed folders of these source files.
- Your code should use meaningful variable names and include comments, in line with good practice.
- Word limit of 1500 words (not including source code)

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

Deadline: 4pm on Friday 9th November 2018.

Marks will be awarded based on the following general criteria:

- Correctness of numerical results
- Clear presentation of results
- Correctness & readability of Matlab code and evidence of software verification
- Level of insight, thoroughness of investigation, and clarity of explanation of the results