FSF3561/SF2561 **Finite Element Methods**

Lab Exercise 1

Aim: This lab concerns implementation of the finite element method (FEM) in 1D. The purpose of the lab is to get hands-on experience of implementing a FEM program. *The code should be written so that it is easy to adapt for multiple dimensions.*

Examination: The lab can be done individually or in groups of two. This lab consist of a set of compulsory problems, is graded by pass/fail (1/0), and should be submitted in time for the deadline.

Format of submissions: Each group has to submit a written report in pdf-format including a first page with: name, email address, and educational program for all group members. The report should be written such that a person taking a similar course in FEM should be able to read and understand the problem and your solution from the report. Please include references to books, websites, etc that you used to complete the lab assignment. Note that it is not allowed to copy existing code on the internet or from another group. A tar-archive of the source files needed to test the program and reproduce the results should be provided. You may write your program in Matlab, Octave, or Python.

Part A

To begin, consider the one-dimensional Poisson equation given by

$$
\begin{aligned}
-(k(x)u')' + p(x)u =& f(x), \quad x \in (x_L, x_R) & (1)\\
u(x_L) = a, & \quad u(x_R) = b. & (2)
\end{aligned}
$$

We will be constructing a general finite element code for given functions $k(x)$, $p(x)$, and $f(x)$, as well as domain and boundary conditions.

1. Derive a weak formulation for this problem and formulate the finite element formulation using continuous piecewise linear test and trial functions.
2. Write a function to construct a mesh, $\mathcal{T}_h$, defined by the set of $N+1$ points $\{x_1, x_2, \ldots, x_{N+1}\}$ for a general one-dimensional domain $[x_L, x_R]$ such that $x_1 = x_L$ and $x_{N+1} = x_L$. For simplicity, assume the N elements, $I_i = (x_i, x_{i+1})$, are uniformly sized. Our mesh is then defined as the union of those elements:
$$\mathcal{T}_h = \bigcup_{i=1}^{N} I_i.$$
3. Write a function to perform the integrations in the variational formulation using a quadrature rule. Discuss the reasons you have chosen the quadrature rule used, including operation count and computational efficiency.
4. Write a routine that assembles the stiffness matrix, A, mass matrix, M, and load vector, $\mathbf{b}$. Assume that the linear system for the approximation coefficients (degrees of freedom) can be written as
$$(A + M)\mathbf{c} = \mathbf{b}.$$

Part B

We will now test out the code that you have written on the following two problems:

$$
\begin{aligned}
-(xu')' =& 5 - 4x, \quad x \in (2, 4), & (3)\\
u(2) =& 3, \quad u(4) = 5,
\end{aligned}
$$

and

$$
\begin{aligned}
-u'' + u =& x, \quad x \in (0, 1) & (4)\\
u(0) =& u(1) = 0.
\end{aligned}
$$

1. Determine the exact solution for (3) and show that for Equation (4) the exact solution is given by $u(x) = x - \frac{\sinh(x)}{\sinh(1)}$.
2. Alter the code that you constructed in **Part A** to produce finite element approximations to Equations (3) and (4).
3. Plot the exact solution, u, as well as the finite element approximation, u_h, for $N = 4$ and $N = 16$ elements. Comment on the difference between these discretizations.

4. Plot the pointwise errors:
   (a) At the element centers for these two discretizations.
   (b) At 6 points per element for $N = 4$. For this particular plot, ensure that the approximation in each element is plotted separately.
   (c) At 6 points per element on a series of meshes with $N = 4, 8, 16, 32, 64$ elements in $\log -$scale (Matlab command: semilogy).

5. Compute the *global* L^2-norm of the error, $e_h = u - u_h$,

$$\|e_h\|_{L^2(\Omega)} = \left(\int_\Omega |e_h|^2\right)^{1/2},$$

on a series of meshes with $N = 4, 8, 16, 32, 64$ elements using 6 points per element. Report these results together with your results from Problem (7), given below.

6. Construct a $\log - \log$ plot of:
   (a) h (x-axis) vs. $\|e_h\|_{L^2(\Omega)}$. Calculate the slope of each (global) L^2-error curve linear portion (remember you are in log-log scale) and comment on what the slope represents.
   (b) Mesh size (x-axis) vs. spectral condition number of the system matrix (y-axis). How does the condition number scale with the mesh size?

7. Calculate the order of accuracy of the approximation using the formula obtained from Richardson's extrapolation:

$$\text{order} = \frac{\log\left(\frac{\text{previous error}}{\text{current error}}\right)}{\log\left(\frac{\text{current N}}{\text{previous N}}\right)}.$$

Display the results of the *global* L^2-errors and order of accuracy as a table:

N	L^2-error	Order
4		—
8		
16		
32		
64		