Lab Exercise 1

Finite Element Method/FSF3561

Kin Wing Wong   Nuclear engineering
Yanjun Zhang   Vehicle Engineering

yanjunzh@kth.se; kwwo@kth.se

October 24, 2023

1 Algorithm of 1D FEM

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

$$-(k(x)u')' + p(x)u = f(x), x \in (x_L, x_R)$$
$$u(x_L) = a, u(x_R) = b.$$

The main structure to build FEM is

1. Derive the variational formulation
2. Constructing a FEM approximation, Discrete the domain
3. Choose a space of approximating
4. Assemble the stiffness, mass matrix, and load vector.

To obtain weak formulation, we multiply with

$$v \in H^1(x_L, x_R), v(x_L) = 0, v(x_R) = 0$$

or

$$v \in H_0^1(x_L, x_R)$$

and integrate by parts as:

$$\int_{x_L}^{x_R} fv = \int_{x_L}^{x_R} -(ku')'v + puv$$

$$= \int_{x_L}^{x_R} (ku'v' + puv) - k(x_L)u'(x_L)v(x_L) + k(x_R)u'(x_R)v(x_R) = \int_{x_L}^{x_R} (ku'v' + puv)$$

Then a weak formulation for this problem is: find $u \in H_0^1(x_L, x_R)$ with all

$$\int_{x_L}^{x_R} fv = \int_{x_L}^{x_R} ku'v' + puv$$

for all $v \in H_0^1(x_L, x_R)$.

Replace u, v by piecewise linear functions $u_h.v_h$, Define space of piecewise linear function V_h with $N+1$ equidistant points, $v_h \in V_{h,0}$ is globally continuous, and it is linear on each $[x_i, x_{i+1}]$.
Define $\{\phi_j\}_{j=0}^N \subset V_h$ as the hat functions with:

$$\phi_j(x_i) = \begin{cases} 1 \; i = j \\ 0 \; i \neq j \end{cases} \quad for \; i, j = 0, ..., N$$

$$u_h(x) = \sum_{i=0}^{N} \xi_i \phi_i(x), \;\; where \, \xi_i = u_h(x_i).$$

Then finite element approximation us: find $u_h \in V_h$ with

$$\int_{x_L}^{x_R} fv_h = \int_{x_L}^{x_R} ku_h'v_h' + pu_h v_h$$

for all $v \in H_0^1(x_L, x_R)$.
Since ϕ_i are basis for V_h this is the same as

$$\int_{x_L}^{x_R} f\phi_i = \int_{x_L}^{x_R} ku_h'\phi_i' + pu_h\phi_i$$

for all $\; 0 \leq i \leq N$.
Since $u_h(x) = \sum_{i=0}^N \xi_i \phi_i(x)$, the finite element approximation could also be

$$\int_{x_L}^{x_R} f\phi_i = \sum_{i=0}^{N} \xi_j \int_{x_L}^{x_R} (k\phi_j'\phi_i' + p\phi_j\phi_i)$$

for $\; 1 \leq i \leq N-1$.

The equations can be written as a system$(A+M)c = b$, where $A \in R^{(N+1)\times(N+1)}$ are given by

$$\begin{cases} A_{ij} \; := \int_{x_L}^{x_R} k\phi_j'\phi_i' d_x \\ M_{ij} \; := \int_{x_L}^{x_R} p\phi_j\phi_i d_x \quad\quad i = 1, 2, ..., n-1 \\ b_i \; := \int_{x_L}^{x_R} f\phi_i d_x \end{cases}$$

$$\varphi_i = \begin{cases} \frac{x-x_{i-1}}{h_i} \ x \in I_i \\ \frac{x_{i+1}-x}{h_i} \ x \in I_{i+1} \\ 0 \ otherwise \end{cases} \quad \varphi_i' = \begin{cases} \frac{1}{h_i} \ x \in I_i \\ \frac{-1}{h_{i+1}} \ x \in I_{i+1} \\ 0 \ otherwise \end{cases}$$

Then, it is straightforward to calculate the entries of A, M and b.

$$A = \begin{cases} A_{ii} := \frac{k_i}{h_i} + \frac{k_{i+1}}{h_{i+1}} \\ A_{i,j+1} := -\frac{k_{i+1}}{h_{i+1}} \end{cases} \quad M = \begin{cases} M_{ii} := \frac{p_i \times h_i}{3} + \frac{p_{i+1} \times h_{i+1}}{3} \\ M_{i,j+1} := \frac{p_{i+1} \times h_{i+1}}{6} \end{cases} \quad b = \begin{cases} b_0 := \frac{f(x_0) \times h_1}{2} \\ b_i := \frac{f(x_i) \times (h_i + h_{i+1})}{2} \\ b_n := \frac{f(x_n) \times h_n}{2} \end{cases}$$

A and M matrix haven't added the boundary condition in the above equation, while we have added in matlab code. Here the mass matrix M integrations using simpson's formula, this is the third-order polynomials with higher accuracy than the mid-point rule and the Trapezoidal rule, while this needs more computational effort, but unclear how much it is more expensive than the other two.

Part B

1.

The exact solution for $-(xu')' = 5 - 4x, x \in (2,4), u(2) = 4, u(4) = 5$ is

$$-(xu')' = 5 - 4x$$
$$-x'u' - xu'' = 5 - 4x$$

make $u' = v$, then

$$-x'v - xv' = 5 - 4x$$

First, we solve

$$-x'v - xv' = 0$$

this equation is equal to

$$-x'v = xv'$$
$$\frac{-dx}{x} = \frac{dv}{v}$$

The solution is

$$\ln \mid v \mid = -\ln \mid x \mid + C_1$$

Since $x \in (2,4)$, so above equation could be writen as

$$\ln \mid v \mid = -\ln x + C_1$$

Then the generation solution is

$$\mid v_g \mid = \frac{1}{x} e_1^C$$

Now, let's consider the right side $5 - 4x$. Assume the v is in the form of $v_p = ax + b$, then

$$v_p' = a$$

take this into

$$-x'v - xv' = 5 - 4x$$

Then get

$$-(ax + b) - xv_p' = 5 - 4x$$
$$-(ax + b) - xa = 5 - 4x$$
$$-2ax - b = 5 - 4x$$

Then get

$$a = 2 \qquad b = -5$$

so the general solution for v is a combination of the homogeneous and particular solutions:

$$v = v_g + v_p = \frac{1}{x} e^{C_1} + 2x - 5$$

Now calculate u according to v through integration

$$u(x) = \int (\frac{1}{x} e^{C_1} + 2x - 5) dx$$

$$u(x) = e^{C_1} \ln x + x^2 - 5x + C_2$$

Since $u(2) = 3, \ u(4) = 5$, then

$$\begin{cases} u(2) = 3 = e^{C_1} \ln 2 + 2^2 - 5 \times 2 + C_2 \\ u(4) = 5 = e^{C_1} \ln 4 + 4^2 - 5 \times 4 + C_2 \end{cases} \Rightarrow \begin{cases} C_1 = 0 \\ C_2 = 9 \end{cases}$$

So the exact solution is

$$u(x) = x^2 - 5x + 9$$

For the equation $-u'' + u = x, x \in (0,1)$, the exact solution is $u(x) = x - \frac{sinh(x)}{sinh(1)}$, where $sinh(x) = \frac{e^x - e^{-x}}{2}$, $sinh(1) = \frac{e^1 - e^{-1}}{2}$, the show is below

$$\begin{cases} u(x) = x - \frac{e^x - e^{-x}}{2sinh(1)} \\ u(x)' = 1 - \frac{e^x + e^{-x}}{2sinh(1)} \\ u(x)'' = -\frac{e^x - e^{-x}}{2sinh(1)} \end{cases}$$

Then take above series to $-u'' + u = x, x \in (0,1)$, get $\begin{cases} -u'' + u = x, x \in (0,1) \\ \frac{e^x - e^{-x}}{2sinh(1)} + x - \frac{e^x - e^{-x}}{2sinh(1)} = x \end{cases}$

2.

The Dirichlet condition needs to be adjusted. The stiffness matrix's first and last rows' coefficients should be set to 1.0 for $i = 1$ and $i = N$, while the mass matrix should have zero coefficients at these locations. The boundary values should be included in the first and last rows of the load vector. More details are shown in the attachment, Matlab files.

3.

Figure 1 show the comparison of the FEM solutions for equations 3 and 4 respectively. The case with smaller discretization(N=4) leads to higher error than the case with 16 discretization(N-16).

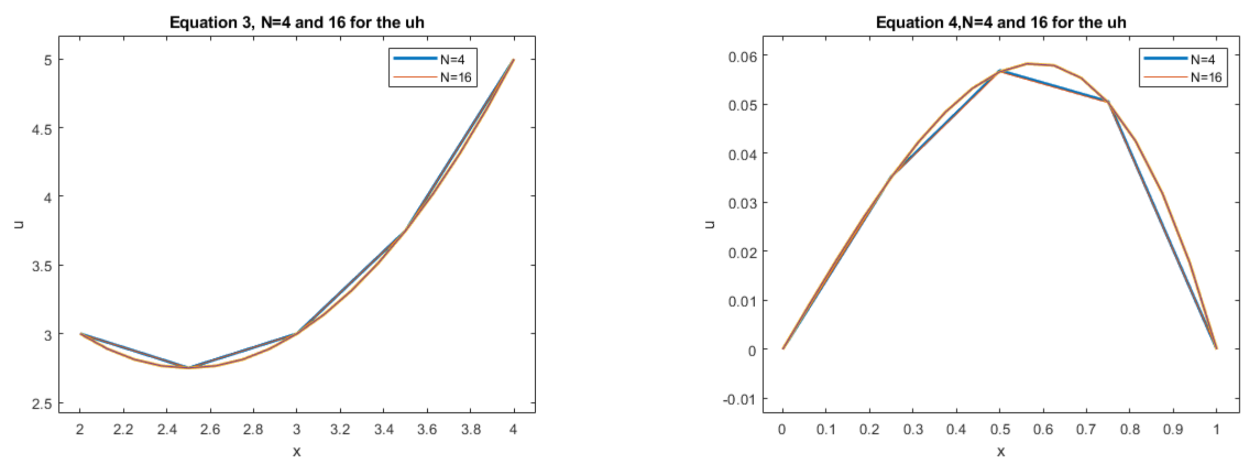

Figure 1: Comparison of FEM solution of Equation 3/4 with N=4 and 16.

4 (a).

Figure 2 illustrate the comparison of errors at the center of the element. It is evident that the error is lower when the discretization is increased. One interesting finding is for equation 3, which does not have reaction term, the pointwise errors in midpoint do not change, it is near constant. While for the equation 4, which has reaction term, the error has large difference when N=4. (Not totally understand why this happened? Maybe the reason is exact solution of equation 3 is more linear than equation 4?)

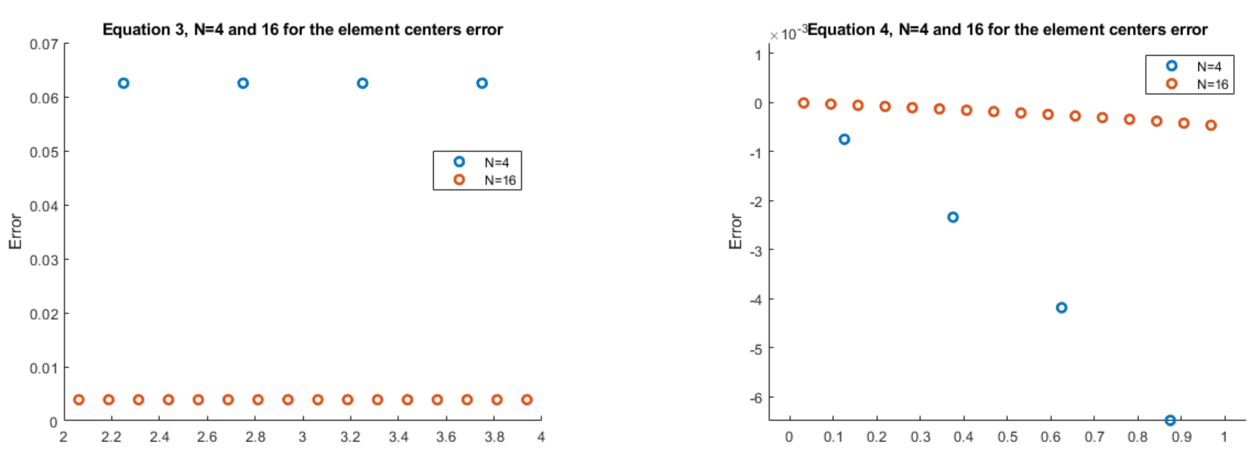

Figure 2: Midpoint Error of Equation 3/4 with N=4 and 16.

4 (b).

Figure 4 shows the pointwise errors at 6 points per element for N=4. The mid of these 6 points per element has the largest error. The reason is in each element, the result in stored in nodes, between the nodes, the error has increased. The closer to the node, the accurate it is.

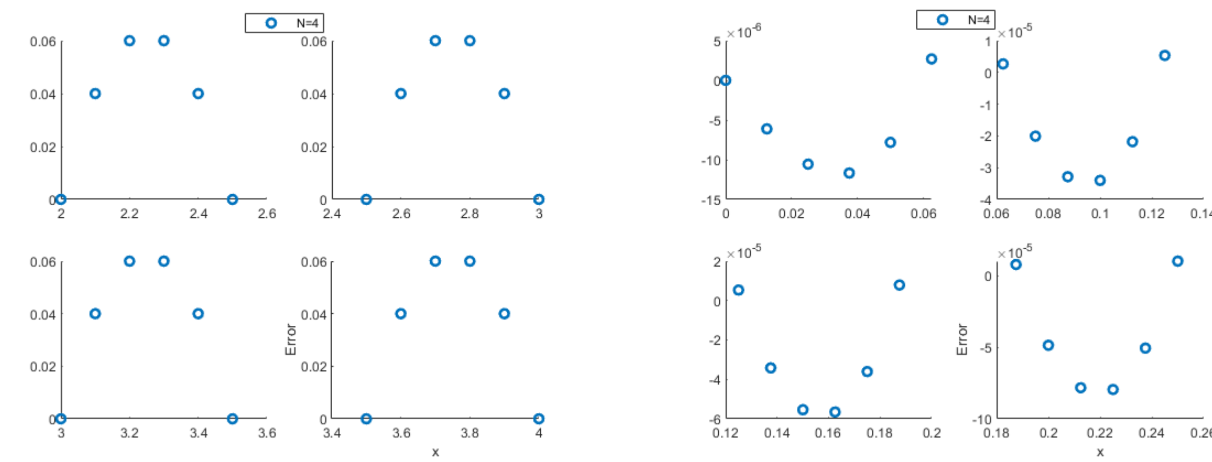

Figure 3: Pointwise errors of 6 points per element of Equation 3/4 with N=4 and 16.

4 (c).

Figure 4 shows that the pointwise element. Currently, we are only plotting the first 4 element error for N = 8, 16, 32, and 64. Maybe it is enough to plot the first 4, as the conclusion is simialr for other remianing elements. We generate six points within each element by linearly interpolating between the element's edges. We also perform a similar interpolation on the FEM solution. Figure 4 illustrates the pointwise error when using a six-point per element approach to evaluate the error. Equations 3 and 4 both demonstrate that the error decreases as the number of elements (N) increases. There is a sharp decrease at the edges of the element, but not at the interior points.

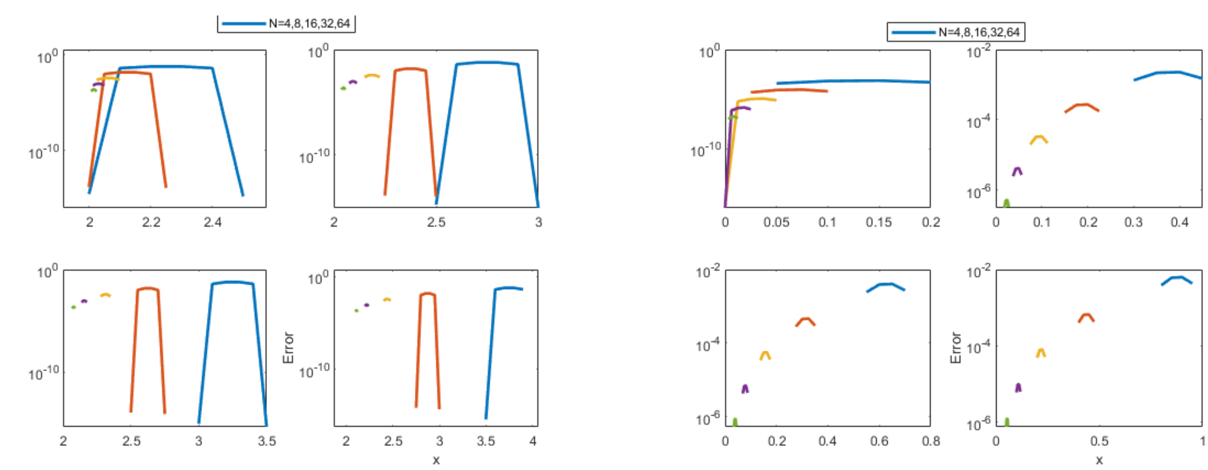

Figure 4: Log scale pointwise errors of 6 points per element of Equation 3/4 with N=4,8,16,32,64

5.

Please see question 7.

6.

The log-log plot in Figure 5 displays the absolute error (e_h) in relation to the element size (h) for equations 3 and 4 with $N = 4$ and 16. The order of equation 3 is approximately 1.226 and that of equation 4 is approximately 1.026. The slope of the graph indicates the rate at which the error decreases as the element size is reduced.

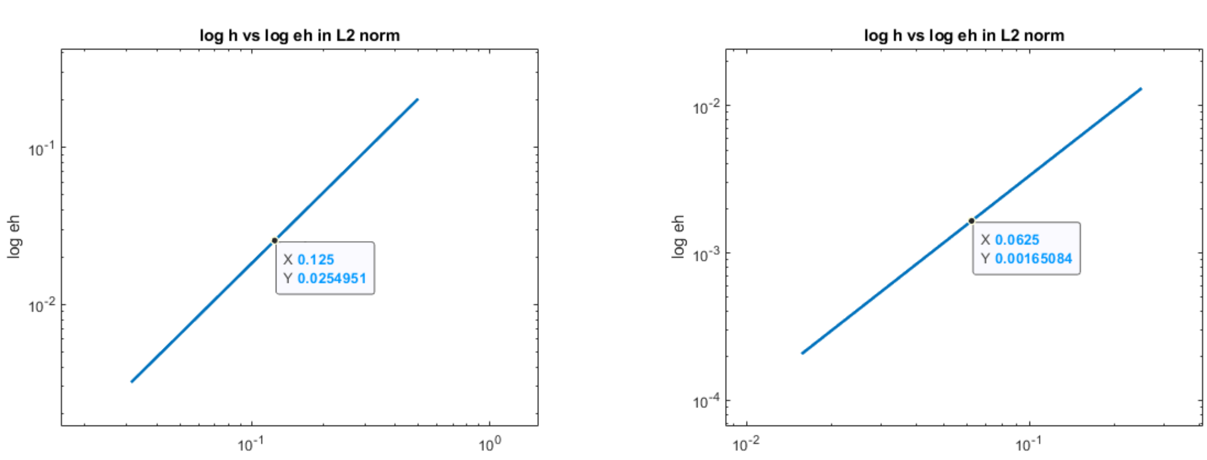

Figure 5: mesh of Equation 3/4 with N=4 and 16.

The figure 6 shows a log-log plot of the condition number in relation to the element size (h) for equations 3 and 4 with $N = 4$ and 16. It can be seen that the condition number decreases as the element size increases. The higher the spectral condition number, the closer the system matrix is to being singular. This implies that the solution for the lower mesh size is more sensitive to errors in the input.

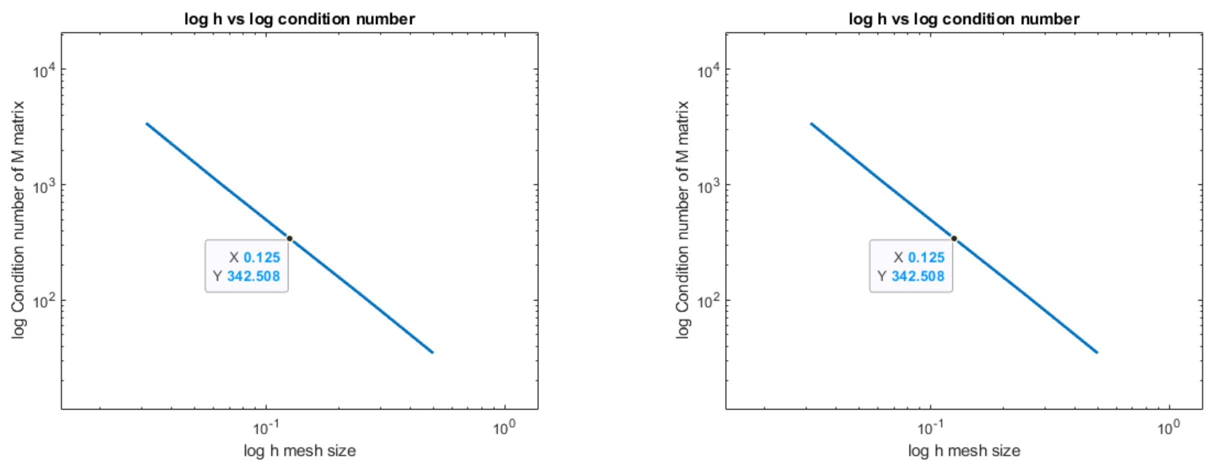

Figure 6: mesh of Equation 3/4 with N=4 and 16.

7.

Tables 1 and 2 present the L^2-error and order values for N ranging from 4 to 64. The L^2-error was determined using the MATLAB command $norm(\mathbf{A})$, where $\mathbf{A}$ is the error matrix for 6 points per element. We can observe an order of approximately 1.5 for equations 3 and 4, however, there is a slight decrease in order when N is equal to 64 for equation 4.

N	L^2-error	Order
4	0.2039	–
8	0.0721	1.4998
16	0.0254	1.5052
32	0.0090	1.4968
64	0.0031	1.5377

Table 1: Equation 3

N	L^2-error	Order
4	0.0131	–
8	0.0046	1.5099
16	0.0016	1.5236
32	0.0005	1.6781
64	0.0002	1.3219

Table 2: Equation 4