AEM-ADV11 Finite Element Method Assignment

Name: Wenze Liu Chen**, CID:** 01526073

Consider the Helmholtz problem

$$\frac{\partial}{\partial x}\left(\sigma\frac{\partial u}{\partial x}\right) - \lambda u = f(x). \tag{1}$$

1. Analytically construct the Galerkin weak form of equation (1) in the region a ≤ x ≤ b describing how to impose the Neumann and Dirichlet boundary conditions

$$u(a) = \alpha, \frac{du}{dx}(b) = \beta, \tag{2}$$

where α and β are known constants. [15%]

The Galerkin weak form can be constructed by following the formulation of the method of weighted residuals. The differential equation (1) is first multiplied by a test function $v(x)$ on both sides, and then integrate over the solution domain $a \leq x \leq b$:

$$\int_a^b v(x)\left(\frac{\partial}{\partial x}\left(\sigma\frac{\partial u}{\partial x}\right) - \lambda u\right)dx = \int_a^b v(x)f(x)\,dx$$

Since σ is assumed to be a constant, so the integral can be simplified somewhat:

$$\int_a^b v(x)\left(\sigma\frac{\partial^2 u}{\partial x^2} - \lambda u\right)dx = \int_a^b v(x)f(x)\,dx$$

Perform the integral by parts on the term which contains the second order derivative:

$$\int_a^b \sigma v(x)\frac{\partial^2 u}{\partial x^2}dx = \left[\sigma v(x)\frac{\partial u}{\partial x}\right]_a^b - \int_a^b \sigma\frac{\partial v(x)}{\partial x}\frac{\partial u}{\partial x}dx$$

Substitute back into the original big integral:

$$\left[\sigma v(x)\frac{\partial u}{\partial x}\right]_a^b - \int_a^b \sigma\frac{\partial v(x)}{\partial x}\frac{\partial u}{\partial x}dx - \int_a^b \lambda v(x)u\,dx = \int_a^b v(x)f(x)\,dx$$

Rearrange the integral:

$$\int_a^b \sigma\frac{\partial v(x)}{\partial x}\frac{\partial u}{\partial x}dx + \int_a^b \lambda v(x)u\,dx = \left[\sigma v(x)\frac{\partial u}{\partial x}\right]_a^b - \int_a^b v(x)f(x)\,dx$$

The expression above is the Galerkin weak form of the differential equation (1). Note that $u(x)$ denotes the exact solution of the differential equation, when performing finite element method, the exact solution is approximated by a finite expansion which is denoted by $u^\delta(x)$. This also applies to the test function $v(x)$, its finite expansion is represented by $v^\delta(x)$. Thus, the integral above becomes:

$$\int_a^b \sigma\frac{\partial v^\delta(x)}{\partial x}\frac{\partial u^\delta}{\partial x}dx + \int_a^b \lambda v^\delta(x)u^\delta\,dx = \left[\sigma v^\delta(x)\frac{\partial u^\delta}{\partial x}\right]_a^b - \int_a^b v^\delta(x)f(x)\,dx$$

Since a Dirichlet boundary condition need to be applied on boundary a, $u(a) = \alpha$, a test function with boundary condition that gives zero on both boundaries is chosen, that is, $v(a) =$

0 and $\frac{\partial v(b)}{\partial x} = 0$. And it is known that $\frac{du}{dx}(b) = \beta$. The first term on the right-hand side can be therefore written as:

$$\left[\sigma v^{\delta}(x)\frac{\partial u^{\delta}}{\partial x}\right]_a^b = \left[\sigma v^{\delta}(b)\frac{\partial u}{\partial x}(b)\right] - \left[\sigma v^{\delta}(a)\frac{\partial u}{\partial x}(a)\right]$$
$$= \left[\sigma v^{\delta}(b)\frac{\partial u}{\partial x}(b)\right] - 0$$
$$= \left[\sigma\beta v^{\delta}(b)\right]$$

The weak form with Neumann boundary condition imposed is:

$$\int_a^b \sigma\frac{\partial v^{\delta}(x)}{\partial x}\frac{\partial u^{\delta}}{\partial x}dx + \int_a^b \lambda v^{\delta}(x)u^{\delta}\,dx = \left[\sigma\beta v^{\delta}(b)\right] - \int_a^b v^{\delta}(x)f(x)\,dx$$

In order to impose the Dirichlet boundary condition, the approximated solution u^{δ} can be considered as the sum of a function $u^{\mathcal{D}}$ and a homogenous function $u^{\mathcal{H}}$. The know function $u^{\mathcal{D}}$ is known and it satisfies the Dirichlet boundary conditions; the homogenous function $u^{\mathcal{H}}$ is unknown and is zero on the boundaries:

$$u^{\delta} = u^{\mathcal{H}} + u^{\mathcal{D}}$$
$$u^{\mathcal{H}}(a) = 0$$
$$u^{\mathcal{D}}(a) = \alpha$$

Incorporate them into the integral:

$$\int_a^b \sigma\frac{\partial v^{\delta}}{\partial x}\left(\frac{\partial u^{\mathcal{H}}}{\partial x} + \frac{\partial u^{\mathcal{D}}}{\partial x}\right)dx + \int_a^b \lambda v^{\delta}(u^{\mathcal{H}} + u^{\mathcal{D}})\,dx = \left[\sigma\beta v^{\delta}(b)\right] - \int_a^b v^{\delta}f(x)\,dx$$

Rearrange the equation by moving the known terms to the right-hand side:

$$\int_a^b \sigma\frac{\partial v^{\delta}}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x}dx + \int_a^b \lambda v^{\delta}u^{\mathcal{H}}\,dx$$
$$= \left[\sigma\beta v^{\delta}(b)\right] - \int_a^b v^{\delta}f(x)\,dx - \int_a^b \sigma\frac{\partial v^{\delta}}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x}dx - \int_a^b \lambda v^{\delta}u^{\mathcal{D}}\,dx$$

In a more concise form:

$$\int_a^b \left[\sigma\frac{\partial v^{\delta}}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x} + \lambda v^{\delta}u^{\mathcal{H}}\right]dx = \left[\sigma\beta v^{\delta}(b)\right] - \int_a^b \left[v^{\delta}f(x) + \sigma\frac{\partial v^{\delta}}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x} + \lambda v^{\delta}u^{\mathcal{D}}\right]dx$$

The equation above is the final expression of Helmholtz problem in Galerkin weak form with both Dirichlet and Neumann boundaries condition incorporated.

2. Discuss how to enforce the alternative mixed boundary condition

$$\sigma \frac{du}{dx}(a) + u(a) = \gamma. \tag{3}$$

where γ is a constant. [10%]

Consider the integral or weak form of the differential equation, omitting the derivation process which is already explained in previous section:

$$\int_a^b \sigma \frac{\partial v^\delta(x)}{\partial x}\frac{\partial u^\delta}{\partial x}\,dx + \int_a^b \lambda v^\delta(x) u\,dx = \left[\sigma v^\delta(x)\frac{\partial u^\delta}{\partial x}\right]_a^b - \int_a^b v^\delta(x) f(x)\,dx$$

Rearrange the given Robin or mixed boundary condition:

$$\sigma \frac{du}{dx}(a) + u(a) = \gamma \rightarrow \sigma \frac{du}{dx}(a) = \gamma - u(a)$$

Assume the boundary condition for b remains the same. And substitute the above rearranged Robin boundary condition into the integral gives:

$$\int_a^b \sigma \frac{\partial v^\delta(x)}{\partial x}\frac{\partial u^\delta}{\partial x}\,dx + \int_a^b \lambda v^\delta(x) u\,dx = \left[\sigma\beta v^\delta(b)\right] - \left[v^\delta(a)\big(\gamma - u(a)\big)\right] - \int_a^b v^\delta(x) f(x)\,dx$$

Rearrange the integral:

$$\int_a^b \sigma \frac{\partial v^\delta(x)}{\partial x}\frac{\partial u^\delta}{\partial x}\,dx + \int_a^b \lambda v^\delta(x) u\,dx - \left[v^\delta(a)\,u(a)\right] = \left[\sigma\beta v^\delta(b)\right] - \left[v^\delta(a)\,\gamma\right] - \int_a^b v^\delta(x) f(x)\,dx$$

The expression above shows how the Robin boundary condition is imposed in the weak form of the differential equation.

3. For the region $0 \leq x \leq 1$, show that the solution $u^{ex}(x) = cos(2\pi x)$ satisfies equation (1) when $f(x) = -(4\sigma\pi^2 + \lambda)\cos(2\pi x)$ and determine the values of α, β and γ in the boundary conditions in equations (2) and (3). [10%]

For ease of reference, the equation (1) is:

$$\frac{\partial}{\partial x}\left(\sigma\frac{\partial u}{\partial x}\right) - \lambda u = f(x)$$

The simplest way to proof that the given solution is exact solution of equation 1 when $f(x) = -(4\sigma\pi^2 + \lambda)\cos(2\pi x)$ is substitute it into the differential equation. So, compute the first and second derivative of $u^{ex}(x)$ is:

$$\frac{\partial u^{ex}(x)}{\partial x} = -2\pi\sin(2\pi x)$$

$$\frac{\partial^2 u^{ex}(x)}{\partial x^2} = -4\pi^2\cos(2\pi x)$$

Substitute the second derivative into the left-hand side of equation (1):

$$-4\sigma\pi^2\cos(2\pi x) - \lambda cos(2\pi x)$$

Simplify the expression gives:

$$-(4\sigma\pi^2 + \lambda)\cos(2\pi x)$$

The simplified expression is exactly the same as the $f(x)$ given. This implies that the $u^{ex}(x) = cos(2\pi x)$ is the exact solution of equation (1). Now compute the boundary conditions:

Dirichlet boundary condition

$$u(a) = cos(2\pi \cdot a) = \alpha \rightarrow \alpha = cos(2\pi \cdot 0) = 1$$

Neumann boundary condition

$$\frac{du}{dx}(b) = -2\pi\sin(2\pi \cdot b) = \beta \rightarrow \beta = -2\pi\sin(2\pi \cdot 1) = 0$$

Robin boundary condition

$$\sigma\frac{du}{dx}(a) + u(a) = -2\pi\sigma\sin(2\pi \cdot a) + cos(2\pi \cdot a) = \gamma \rightarrow$$
$$\gamma = -2\pi\sigma\sin(2\pi \cdot a) + cos(2\pi \cdot 0) = 1$$

Based on the solution domain $0 \leq x \leq 1$, the Dirichlet, Neumann and Robin boundary condition are $\alpha = 1$, $\beta = 0$ and $\gamma = 1$ respectively.

4. Set up the linear and quadratic finite element approximation of the Helmholtz problem (1) in the interval $0 \leq x \leq 1$ with the boundary conditions given by equation (2) and $\lambda = 1$. For the linear expansion use the standard elemental expansion basis

$$\boldsymbol{\phi_0(\xi) = \frac{1-\xi}{2}}$$

$$\boldsymbol{\phi_1(\xi) = \frac{1+\xi}{2}}$$

where $-1 \leq \xi \leq 1$. For the quadratic finite element approximation use the linear elemental modes $\phi_0(\xi)$, $\phi_1(\xi)$ with the additional quadratic elemental expansion mode

$$\boldsymbol{\phi_2(\xi) = \frac{(1-\xi)(1+\xi)}{4}}.$$

a) Discuss how integration, differentiation and global matrix assembly are performed in your implementation explaining alternative methods if appropriate [15%]

Remember the weak form of the differential equation is:

$$\int_a^b \sigma \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}dx + \int_a^b \lambda v^\delta u^\delta \, dx = \left[\sigma\beta v^\delta(b)\right] - \int_a^b v^\delta f \, dx$$

Substitute $\beta = 0$ and $\lambda = 1$ into the integral:

$$\int_a^b \sigma \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}dx + \int_a^b v^\delta u^\delta \, dx = -\int_a^b v^\delta f \, dx$$

The term $\int_a^b \sigma \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}dx$ and $\int_a^b v^\delta u^\delta \, dx$ on the left-hand side will lead to the so-called Laplacian matrix and Mass matrix respectively. Now consider the global matrix assembly of the Mass matrix, it is known that the approximated solution u^δ is given by the expansion of the following form:

$$u^\delta = \sum_{j=0}^{N} \hat{u}_j \Phi_j$$

Similarly, the test function is given by:

$$v^\delta = \sum_{i=0}^{N} \hat{v}_i \Phi_i$$

Substitute them into the integral:

$$\int_a^b v^\delta u^\delta dx = \sum_{i=0}^{N} \hat{v}_i \int_a^b \Phi_i \left[\sum_{j=0}^{N} \hat{u}_j \Phi_j\right] dx$$

The above expression can be written in matrix form:

$$\sum_{i=0}^{N} \hat{v}_i \int_a^b \Phi_i \left[\sum_{j=0}^{N} \hat{u}_j \Phi_j\right] dx \rightarrow \boldsymbol{\hat{v}^T M \hat{u}}$$

$$\boldsymbol{M} = \int_a^b \Phi_i \Phi_j dx$$

$$\hat{\boldsymbol{v}} = [\hat{v}_0, \dots, \hat{v}_N]^T$$

$$\hat{\boldsymbol{u}} = [\hat{u}_0, \dots, \hat{u}_N]^T$$

The global mass matrix $\boldsymbol{M}$ are constructed from the local elemental mass matrix $\boldsymbol{M^e}$, and this can be done via various methods. It could be done by a numerical operation which maps the DOF (degree of freedom) numbers between the local numbering and global numbering. Or alternately, it is also possible to assemble the global mass matrix by using an assembly matrix $\mathcal{A}$. Another way to construct the global mass matrix is to put the elemental matrices directly to the corresponding position in the global matrix, and this method is one used in this coursework:

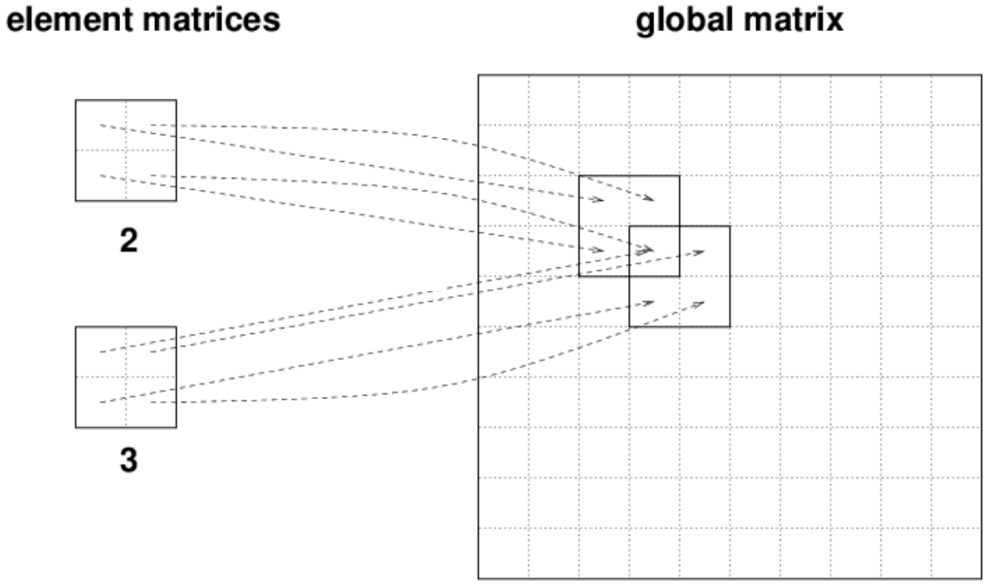

Figure 1: Global matrix assembly scheme

$$\boldsymbol{M} = \begin{bmatrix} {M_{i,j}}^1 & {M_{i,j}}^1 & 0 & 0 & 0 & 0 \\ {M_{i,j}}^1 & {M_{i,j}}^1 + {M_{i,j}}^2 & {M_{i,j}}^2 & 0 & 0 & 0 \\ 0 & {M_{i,j}}^2 & {M_{i,j}}^2 + {M_{i,j}}^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & {M_{i,j}}^N + {M_{i,j}}^{N+1} & {M_{i,j}}^{N+1} \\ 0 & 0 & 0 & 0 & {M_{i,j}}^{N+1} & {M_{i,j}}^{N+1} \end{bmatrix}$$

The algorithm to perform such matrix assembly in MATLAB is shown below:

```
M_glob = zeros(N+1, N+1);
for i = 0:nm-1:(N-1)*(nm-1)
    M_glob(1+i:nm+i, 1+i:nm+i) = M_glob(1+i:nm+i, 1+i:nm+i) + M_elem;
end
```

Once known how to assemble the global matrix the elemental matrix need to be defined. The construction of elemental mass matrix involves evaluation of integral. The integral can be evaluated analytically or numerically. In this coursework, the Gaussian quadrature was implemented to compute the elemental mass matrix. Depending the choice of points, there are three types of quadrature, which are Gauss, Gauss-Radau and Gauss-Lobatto. And Gauss-

Lobatto quadrature was performed. So, start by transforming the integral of mass matrix from solution (global) domain to elemental domain and then transform again to standard domain:

$$\boldsymbol{M^e} = \int_{\Omega_e} \phi_i \phi_j dx = \int_{-1}^{1} \phi_i \phi_j J^e d\xi$$

where J^e is the Jacobian, and it is given by:

$$J^e = \frac{dx^e}{d\xi}$$

Express the integral in discrete form:

$$\boldsymbol{M^e}[i,j] = \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)J^e d\xi = \sum_{k=1}^{Q} \phi_i(\xi_k)\phi_j(\xi_k)J^e w_k$$

where w_k is the weight. Three points was chosen to perform the quadrature ($Q = 3$), which is able to compute the integral of polynomials of order $P \leq 2Q - 3 = 3$ exactly. The Gaussian-Lobatto quadrature includes both end points, therefore the chosen points in standard domains and the corresponding weights are:

Table 1: Choice of points and weights

$\boldsymbol{\xi_k}$	$\boldsymbol{w_k}$
-1	1/3
0	4/3
1	1/3

Based on all this information and the process explained the global mass matrix was constructed in MATLAB. A very similar process was followed to construct the global Laplacian matrix. Note that the Laplacian matrix contained two derivatives, so there is an extra step to construct the Laplacian matrix. The same as integration, the differentiation also can be evaluated analytically or numerically. To make the life easier, an analytical differentiation was performed. Remember the Laplacian matrix is given by:

$$\boldsymbol{L} = \int_a^b \frac{\partial \Phi_i}{\partial x}\frac{\partial \Phi_j}{\partial x} dx$$

The global Laplacian matrix is constructed from elemental Laplacian matrix which is defined by:

$$\boldsymbol{L^e} = \int_{\Omega_e} \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} dx = \int_{-1}^{1} \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} J^e d\xi = \int_{-1}^{1} \frac{\partial \phi_i}{\partial \xi}\frac{\partial \phi_j}{\partial \xi}\left(\frac{\partial \xi}{\partial x}\right)^2 J^e d\xi$$

$$\boldsymbol{L^e}[i,j] = \sum_{k=1}^{Q} \frac{\phi_i}{\partial \xi}(\xi_k)\frac{\phi_j}{\partial \xi}(\xi_k)\left(\frac{\partial \xi}{\partial x}\right)^2 J^e w_k$$

The term $\frac{\partial \xi}{\partial x}$ appears because of the chain rule, this term is equal to the reciprocal of the Jacobian J^e, or:

$$\xi = 2\left(\frac{x - x_{lower}}{x_{upper} - x_{lower}}\right) - 1 \rightarrow \frac{\partial \xi}{\partial x} = \frac{2}{x_{upper} - x_{lower}}$$

Up to this point, all the LHS terms are evaluated. The sum of Laplacian and mass matrix is known as the stiffness matrix, so they can be expressed as:

$$Stiff = \sigma \boldsymbol{L} + \lambda \boldsymbol{M}$$

Now consider the RHS, the only term that needs to evaluate is the source function:

$$\int_a^b v^\delta f(x)\, dx = \sum_i v_i \int_a^b \Phi_i f(x)\, dx = \widehat{\boldsymbol{v}}^{\boldsymbol{T}} \boldsymbol{f}$$

Where $\boldsymbol{f}$ is the global source vector:

$$\boldsymbol{f} = \int_a^b \Phi_i f(x)\, dx$$

And this is constructed from the elemental source vector $\boldsymbol{f^e}$:

$$\boldsymbol{f^e} = \int_{\Omega_e} \phi_i f(x)\, dx = \int_{-1}^{1} \phi_i f(\xi)\, J^e d\xi$$

Express in discrete form:

$$\boldsymbol{f^e}[\mathrm{i}] = \sum_{k=1}^{Q} \phi_i f(\xi)\, J^e w_k$$

Now all the integral terms are evaluated, the next step is to impose the boundary conditions. The Neumann boundary condition is naturally incorporated in the integral as was explained in previous section, and in this case $\frac{\partial u}{\partial x}(b) = \beta = 0$, so the term disappears on RHS. For Dirichlet boundary condition, there exist two ways to implement it. One way is to express the solution as a sum of homogenous part with known part (Dirichlet boundary condition), so $u^\delta = u^{\mathcal{H}} + u^{\mathcal{D}}$, and then move the known part to the RHS. This will lead to a linear system of $(N_{node} - 1) \times (N_{node} - 1)$. Another way is known as the big bring method, and it is the one implemented in the code. This method substitutes the entry of $Stiff$ matrix that corresponds to the coefficient of Dirichlet boundary by a very huge number, and the corresponding entry on RHS matrix by the multiplication of that very huge number with Dirichlet boundary condition, that is:

$$Stiff[1,1] = 10^{10}$$

$$RHS[1,1] = u(1) \cdot 10^{10} = 10^{10}$$

Everything done before leads to the linear system below:

$$[\sigma \boldsymbol{L} + \lambda \boldsymbol{M}] u^\delta = RHS$$

Solving this system will give the approximated solution u^δ. Remember that solving this system gives the expansion coefficients and solution is given by $u^\delta = \sum_{j=0}^{N} \hat{u}_j \Phi_j$. For the linear finite element expansion, the solution of the differential equation is the expansion coefficient itself since the basis function on the nodes are 1. However, for the quadratic expansion (modal p-type expansion), the points between the nodes need to be computed by multiplying the coefficients with basis function at corresponding point. A plot of linear and quadratic expansion is shown below.

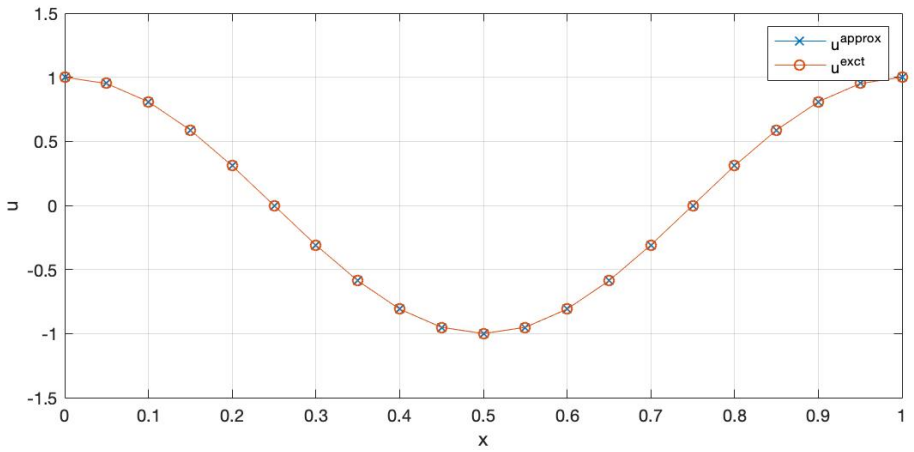

Figure 2: Solution using linear basis function

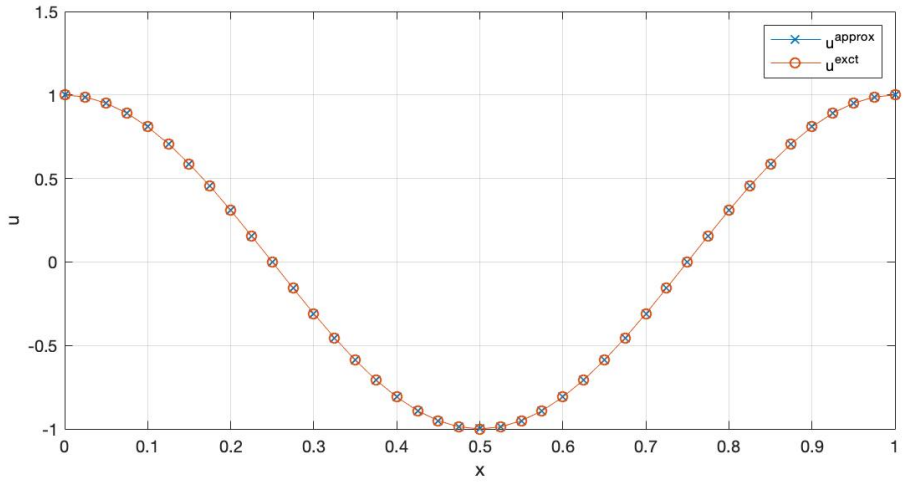

Figure 3: Solution using quadratic basis function

b) Using a mesh of $N_{el} = 5, 10, 20, 50, 100$ equispaced elements determine the L_2 error $\epsilon(x_i) = u^{ex}(x_i) - u^{\delta}(x_i)$ between the exact $u^{ex}(x)$ and numerical solution $u^{\delta}(x)$ defined as

$$L_2 = \sqrt{\sum_{i=0}^{N} \frac{[u^{ex}(x_i) - u^{\delta}(x_i)]^2}{(N+1)}}$$

where

$$x_i = i/N, 0 \leq i \leq N,$$

and

$$\begin{cases} N := N_{el} & \textit{for linear finite element expansion} \\ N := 2N_{el} & \textit{for quadratic finite element expansion} \end{cases}$$

Plot the error, for both your linear and quadratic finite element expansions, versus the size of the element size $1/N_{el}$ on a log-log graph [35%]

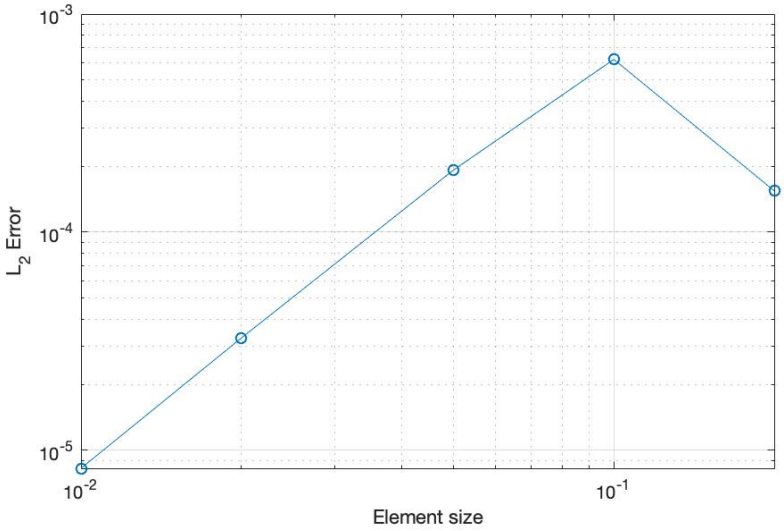

Figure 4: L_2 Error for linear finite element expansion

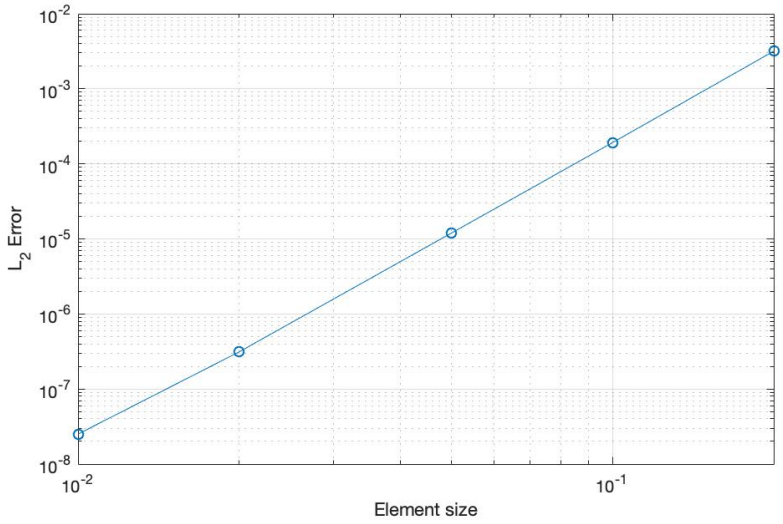

Figure 5: L_2 Error for quadratic finite element expansion

c) Comment on the slopes of error curves on the plot and whether there is an alternative way to express the solution error. [15%]

Figure 4 and *Figure 5* shows the L_2 error of the linear finite element expansion and quadratic finite element expansion. In general, the error tends to decrease when reducing element size, except the linear finite element expansion of five elements. It seems that the approximated solution at those five points match coincidentally good with the exact solution, and this result in a smaller error. It is expected to see smaller element size lead to smaller error, since smaller element size implies finer mesh which should result in more accurate result.

Looking at both figures, the error curves are nearly straight, note that the plots are in log-log scale, which means the error decreases exponentially when reducing the element size. With the same element size, the quadratic finite element expansion gives smaller error then the linear one, so quadratic approximation gives more accurate results. The slope of the error for linear expansion and quadratic expansion is about -2 and -4 respectively, that is, by reducing the element size 10^1 times smaller will lead to a reduction of error 10^2 and 10^4 times smaller for each expansion. Thus, the error is reducing faster for the quadratic expansion, which implies the quadratic expansion gives more accurate results than the linear expansion when using the same element numbers.

It is expected to have a slop of -1 for linear finite expansion, but the actual slope is about -2, which is quite different from what is expecting. In fact, this not a problem of the implementation, this mainly caused by the use of L_2 norm. To express the error more accurately, there is an alternative way called the energy norm. Unlike the L_2 norm, which only takes into account the error at the nodes, the energy norm computes the error between interpolant $\mathbb{I}u$ and u. So, energy norm considers the error in the area between the nodes which gives better representation of the error between approximated solution and exact solution. The energy norm is given by:

$$\|u\|_E{}^2 = \int_0^l \left[\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + \lambda u u\right] dx$$

and in practice the energy norm can be calculated by:

$$\|\epsilon\|_E = \sqrt{\sum_i [\sigma L + \lambda M]\,\epsilon}$$