Lab Exercise 2

Finite Element Method/FSF3561

Kin Wing Wong Nuclear engineering
Yanjun Zhang Vehicle Engineering

yanjunzh@kth.se; kwwo@kth.se

November 14, 2023

1 Algorithm of 2D FEM

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

$$-\nabla(k(x)\nabla u) = f(x), x \in \Omega$$
$$u = 0, x \in \partial\Omega$$

The main structure to build FEM is

1. Derive the variational formulation
2. Construct a mesh
3. Constructing a FEM approximation, Discrete the domain
4. Compute integrals
5. Assemble the stiffness, mass matrix, and load vector.

1.

To obtain weak formulation, we multiply with v and integrate by parts as:

$$\int_\Omega fvdx = \int_\Omega -\nabla(k\nabla u)vdx$$
$$\Rightarrow \int_\Omega fvdx = \int_\Omega (k\nabla u)\nabla vdx - \int_\Omega n \cdot (k\nabla u)vds = \int_\Omega (k\nabla u)\nabla vdx$$

Then a weak formulation for this problem is: find $u \in H^1(\Omega)$ with $v \mid_{\partial\Omega}= 0$ all

$$\int_\Omega k\nabla u\nabla vdx = \int_\Omega fvdx, \quad \forall v \in V.$$

2.

Let κ be a triangulation of Ω, and let $P_1(K)$ be the space of linear function on κ, defined by

$$P_1(K) = \{v : v = C_0 + C_1x_1 + C_2x_2, \quad (x_1, x_2) \in K, c_0, c_1, c_2 \in R\}$$

3.

Define the finite-dimensional space, the space of all continuous piecewise linear polynomials V_h, defined by

$$V_h = \{v : v \in C^0(\Omega), v \mid_k \in P_1(K)\}, \forall K \in \kappa$$

$C^0(\Omega)$ denotes the space of all continuous functions on Ω. To construct a basis for V_h we first show that a function v in V_h is uniquely determined by its nodal values

$$\{v(N_j)\}_{j=1}^{n_p}$$

Define $\{\phi_j\}_{j=1}^{N_p} \subset V_h$ as the hat functions with:

$$\phi_j(N_i) = \begin{cases} 1 \; i = j \\ 0 \; i \neq j \end{cases} \quad for \; i, j = 1, 2, ..., n_p$$

Now, using the hat function basis we note that any function v in V_h can be written

$$v = \sum_{i=1}^{n_p} \alpha_i\phi_i,$$

where $\alpha_i(N_i), i = 1, 2, ..., n_p$, are the nodal values of v.

Let κ be a triangulation of Ω,and let V_h be the space of continuous piecewise linears on κ . To satisfy the boundary conditions, let also

$$V_{h,0} = \{v \in V_h : v \mid_{\partial\Omega=0}\}$$

Replacing V_0 with $V_{h,0}$, then finite element approximation is: find $u_h \in V_{h,0}$ with

$$\int_\Omega fv = \int_\Omega k\nabla u_h\nabla v$$

for all $v \in V_{h,0}$.
Since ϕ_i are basis for V_h this is the same as

$$\int_\Omega f\phi_i = \int_\Omega k\nabla u_h\nabla\phi_i \quad i = 1, 2, ..., n.$$

Since $u_h = \sum_{j=1}^{n_i} \xi_j\phi_j$, the finite element approximation could also be

$$\int_\Omega f\phi_i = \sum_{j=1}^{n_i} \xi_j \int_\Omega k\nabla\phi_j \cdot \nabla\phi_idx \;\; i = 1, 2, ...n_i$$

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

$$\begin{cases} A_{ij} & := \int_\Omega k\nabla\phi_j\nabla\phi_i d_x \\ b_i & := \int_\Omega f\phi_i d_x \end{cases} \quad i = 1, 2, ..., n-1$$

Consider a triangle k with nodes $N_i = (x_1^{(i)}, x_2^{(i)}), i = 1, 2, 3$. To each node N_i there is a hat functions φ_i associated, what takes the value 1 at node N_i and 0 at the other two nodes. Each hat function is a linear function of K so it has the form

$$\varphi_i = a_i + b_i x_1 + c_i x_2$$

where the coefficients a_i, b_i, and c_i are determined by

$$\varphi_i(N_i) = \begin{cases} 1 \; i = j \\ 0 \; i \neq j \end{cases} \quad i, j = 1, 2, 3$$

$$a_i = \frac{x_1^{(j)}x_2^{(k)} - x_1^{(k)}x_2^{(j)}}{2 \mid K \mid}, \quad b_i = \frac{x_2^{(j)} - x_2^{(k)}}{2 \mid K \mid}, \quad c_i = \frac{x_1^{(k)} - x_1^{(j)}}{2 \mid K \mid}, \nabla\varphi_i = [b_i, c_i]^T$$

$$A_{ij}^k := \int_k k\nabla\phi_i\nabla\phi_j d_x = (b_i b_j + c_i c_j)\int_k k dx \approx \overline{k}(b_i b_j + c_i c_j) \mid k \mid$$

4.

Where

$$\overline{k(x)} = k(x)(\frac{1}{3}(N_1 + N_2 + N_3))$$

is the center of gravity value of A on K. Here, for the f, also use the simplest quadrature formula, the center of gravity rule. The reason is it is simple while the accuracy is lower than two-dimensional Mid-poin rule. But it is sufficient for this simple implementation of fem, according to page 60 of reference [1].

$$\int_k f dx \approx f(\frac{(N_1 + N_2 + N_3)}{3}) \mid k \mid$$

The above analysis object is the triangle element k, We chose the first-order polynomial to interpolate f. If the analysis object is a line element, it is also called the trapezoidal rule. As discussed in lab 1, we chose simpson's formula for the third-order polynomials with higher accuracy than the mid-point rule and the trapezoidal rule, while this needs more computational effort. The trapezoidal rule is a special type of quadrature rule.

5.

Assembly of the Stiffness Matrix:

1. Let n be the number of nodes and m the number of elements in a mesh, and let the mesh be described by its point matrix P and connectivity matrix T.
2. Allocate memory for the $n \times n$ matrix A and initialize all matrix entries to zero.
3. for $K = 1, 2, ...m$ do
4. Compute the gradients $\nabla\varphi_i = [b_i, c_i]$, $i = 1, 2, 3$ of the three hat functions φ_i on K.
5. Compute the 3×3 local element mass matrix A^k given by

$$A^K = \overline{k}\begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix} \mid K \mid$$

6. Set up the local-to-global mapping, loc2glb$= [r, s, t]$
7. for $i = 1, 2, 3$ do
8. for $j = 1, 2, 3$ do
9. $A_{loc2glb_i loc2glb_j} = A_{loc2glb_i loc2glb_j} + A_{ij}^k$
10. end for
11. end for
12. end for

Assembly of the load vector

1. Let n_p be the number of nodes and n_t the number of elements in a mesh described by its point matrix P and connectivity matrix T.
2. Allocate memory for the $n_p \times 1$ vector B and initialize all vector entries to zero.
3. For $K = 1, 2, ...n_t$ do

4. Compute the 3×1 local element load vector b^k given by

$$b^K = \frac{1}{3} \begin{bmatrix} f(N_1) \\ f(N_2) \\ f(N_3) \end{bmatrix} \mid k \mid$$

5. Set up the local to global mapping, loc2glb=$[r, s, t]$
6. for $i = 1, 2, 3$ do
7. $\quad b_{loc2glb_i} = b_{loc2glb_i} + b_i^k$
8. $\quad$ end for
9. end for

Examining the structure of the node arrangement reveals that MATLAB format arranges the boundary nodes first (i.e., 1 to N_b points are boundary only). Consequently, A_{ij} for the boundary nodes should be a diagonal matrix, and the off-diagonal terms are zero for the first N_b nodes. Similarly, the first N_b entries of the load vector are zero. This is analogous to Lab 1, where we modified the first and last entries of the load vector. No further amendment is necessary for the nodes from N_b to N, which are interior nodes.

6.

a

As shown in Figure 1, with different mesh, the finite element approximation results u_h are different. With a small mesh size h, u_h is more accurate, since the amplitude(around 2) is closer to the exact solution(2).Also, the 2D plot is more accurate.

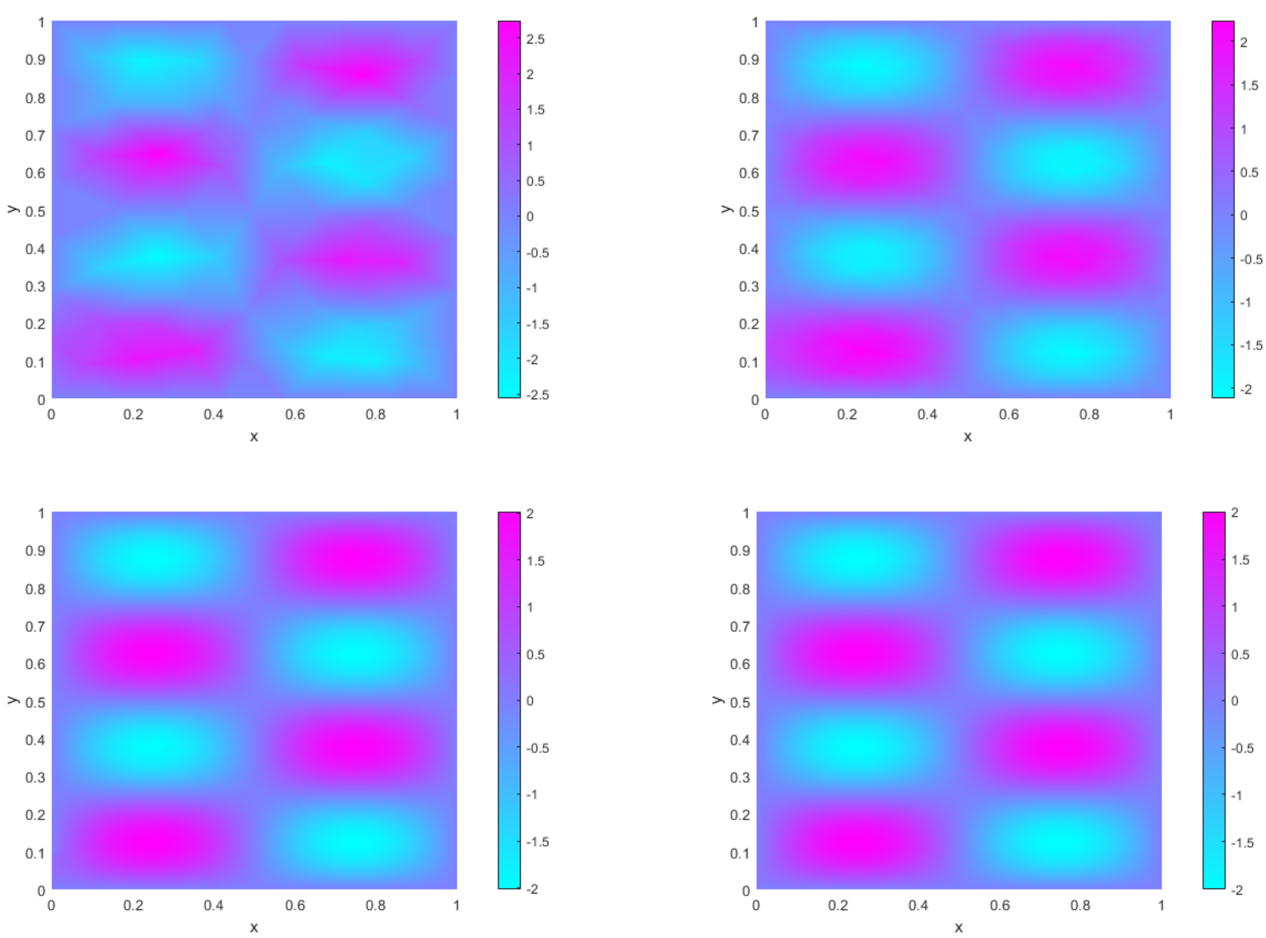

Figure 1: Finite element approximation uh with different mesh, uh for h=0.25/0.125/0.0625/0.03125.

b

As shown in Table 1, the L^2 error was shown with the order calculated by the formula mentioend in the Lab 1 for order calcualtion. Error is reducing with fine mesh, small element size h.

h	N	L^2-error	Order
0.4	17	23.1031	–
0.3	27	7.7055	3.81
0.2	52	3.0973	2.24
0.1	185	1.4118	1.13

Table 1: Equation 3

c

Figure 2 shows the order convergence of the approximation u_h with respect to the number of mesh points N and the mesh size h. It is found that the order of convergence reduces with the decrease in the mesh size. This might be due to some approximation error accumulating at large numbers of mesh points.

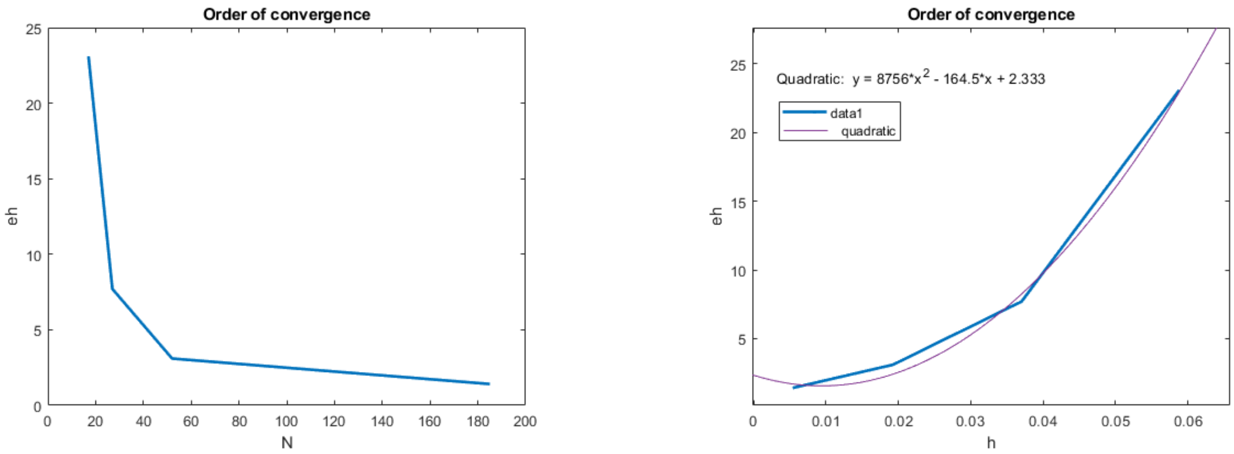

Figure 2: Mesh points N with error e_h, $h = 1/N$, Mesh size h with error e_h

7.

a, b

Figure 3 shows the finite element approximation and the error e_h. We can see that the error decreases with the decreasing mesh size, as expected.

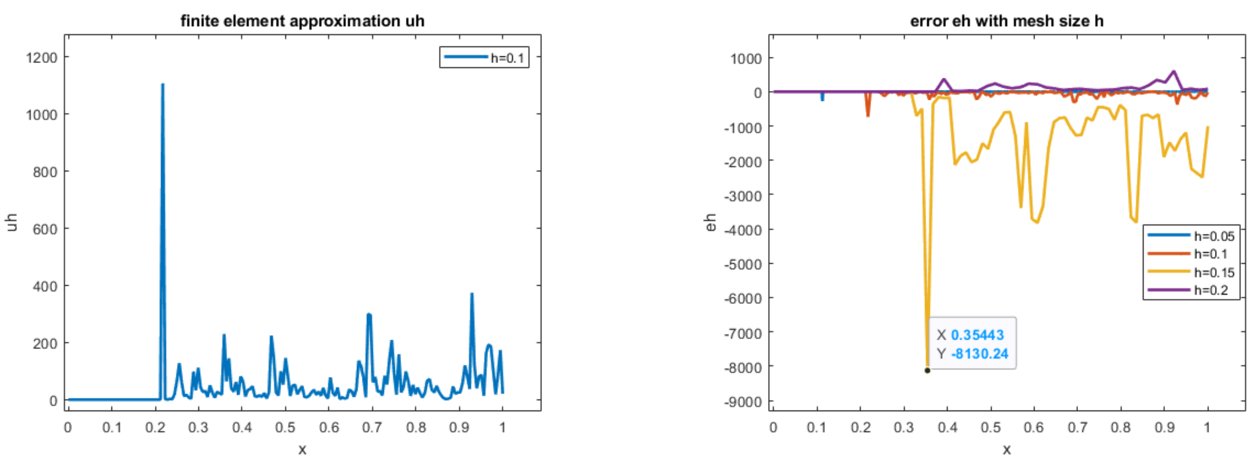

Figure 3: Mesh points N with error e_h, $h = 1/N$, Mesh size h with error e_h

c

As shown in 3, for the $h = 0.15$, the maximum error occurs at $x = 0.35$. This function is achieved by 'find' in Matlab, more detail could be found in Q7.m.

d

i: Figure 4 displays the mesh before and after refinement. The refinement was only applied to the cells within a circle centered at (0.5, 0.5) with a radius of 0.05. Consequently, a small area near the center of the mesh was refined.
ii: Residual is computed by function 'pdejmps':
iii.
Five local mesh refinement, error L^2 norm

Red-green mesh Error	Red-green mesh element	Three uniform mesh Error	Three uniform mesh element
1.5e4	130	1.4e4	130
7.7e3	253	1.8e3	582
8.4e3	325	292	2414
9.7e3	413	94	9770

Table 2: Error and element

As shown in Table 3, for the local mesh refinements with red-green mesh refinements, the L2 norm of error reduced slowly, only one order for 3 times refinements. While for the three uniform mesh refinements, after 3 times refinements, the L2 norm of error reduced 4 orders, which is significantly. At the same time, the elements increases nearly two orders for three uniform mesh, while for Red green mesh element, it still in the same order.

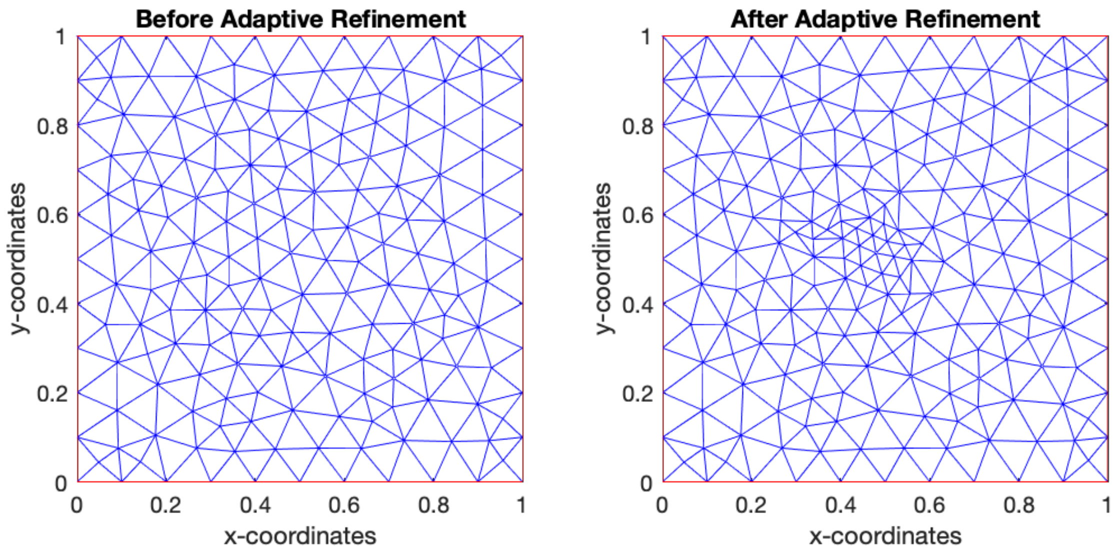

Figure 4: Comparison of the meshes before and after refinement.

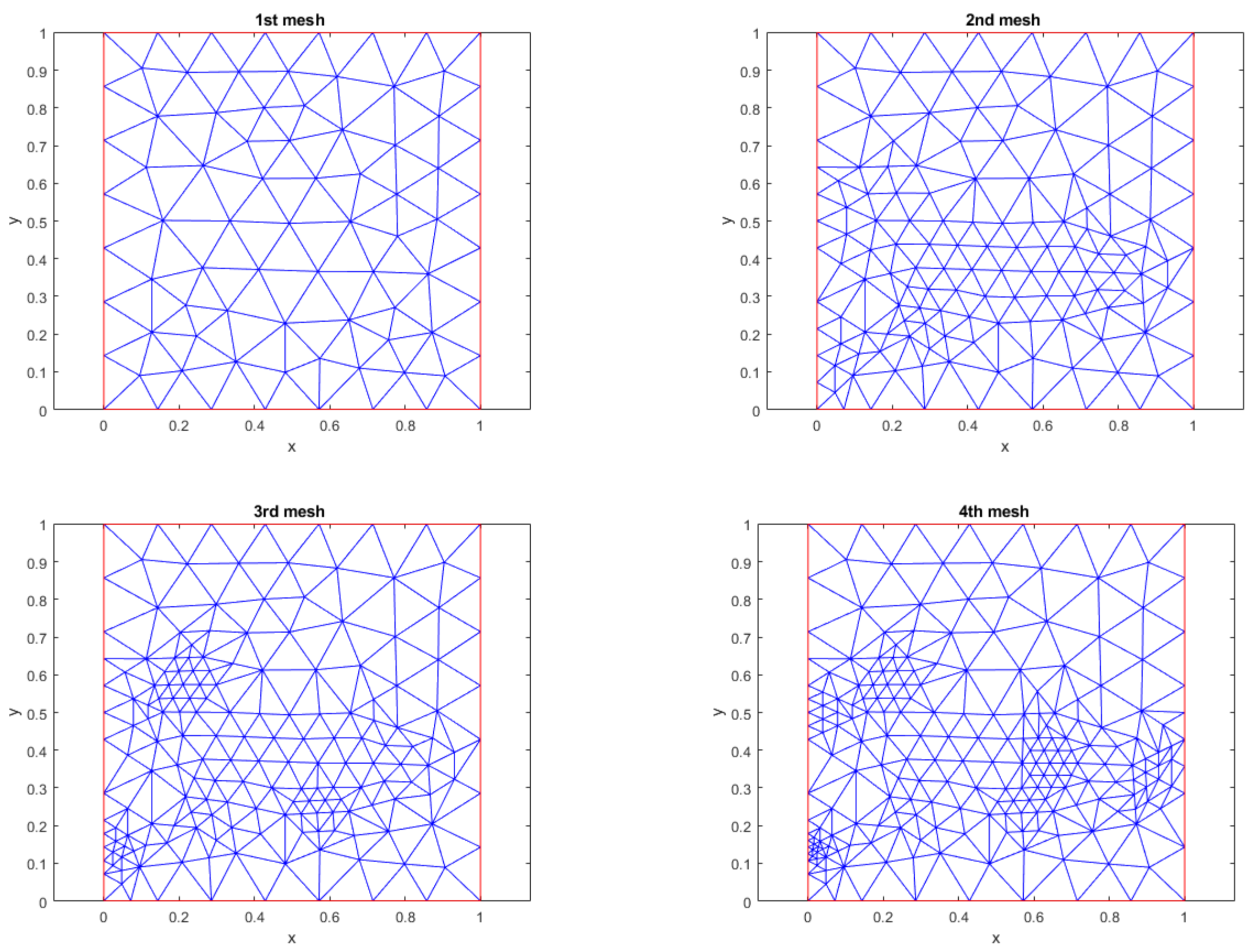

Figure 5: Red-green mesh methods for 3 times refinements.

It could concluded that Red mesh method is more efficient since with same order of element, the mesh error reduced one order.

e

Use second order element, where they have points on the middle of the two nodes. Also, adaptive meshing can help to further improve the error.

8.

a

The load vector b is the modified such that the boundary points which lie on $x_1 = 0$ and $x_1 = 1$, will not be set as zero as in the Dirichlet boundary condition. We identified the points lying on the vertical line between $x_1 = 0$ and $x_1 = 1$ and avoid modifying such entries. Figure 7 shows that the comparison between exact U and finite element approximation U_h and the order of convergence follows second order convergence and then reduces at smaller mesh sizes.

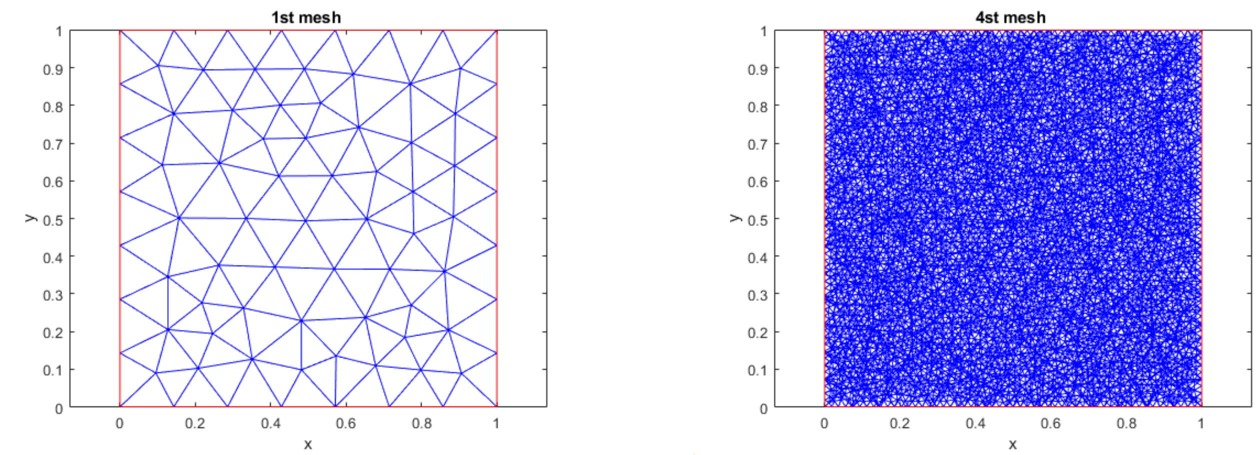

Figure 6: Three uniform mesh methods for 3 times refinements.

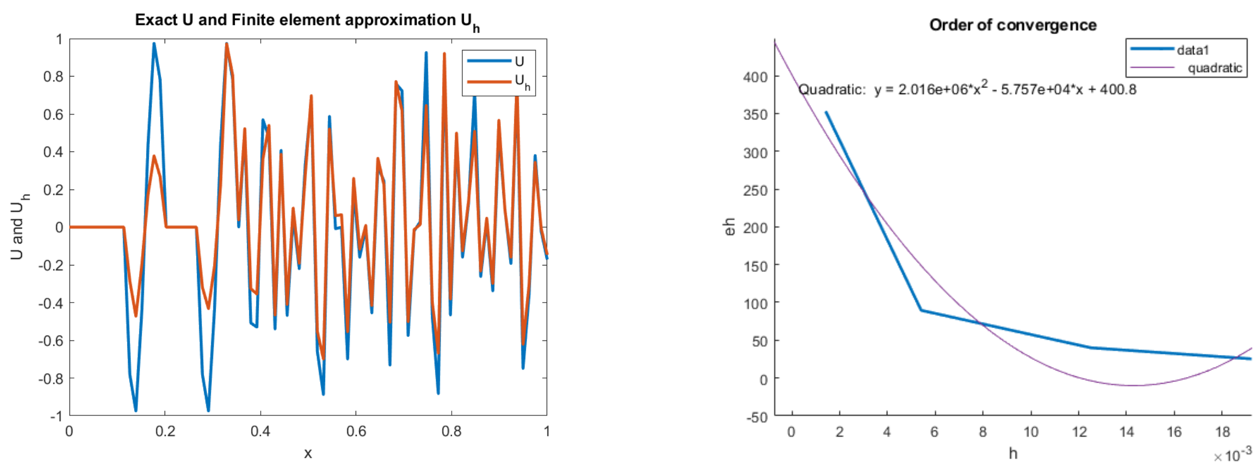

Figure 7: Exact U and Finite element approximation U_h

9.

For this question, a little confused about α and k, the first version is $\alpha = 9$, and $u(x) = 1/16$, the second version of lab2 description is $k = 9$, and $u(x) = 1/16$, then what is the α in the second version. We assume $k = 9$, and $\alpha = 1/16$.

a

Figure 8 shows the finite element mesh of the circle and the comparison between the finite element approximation U_h and the exact solution U for this mesh size.

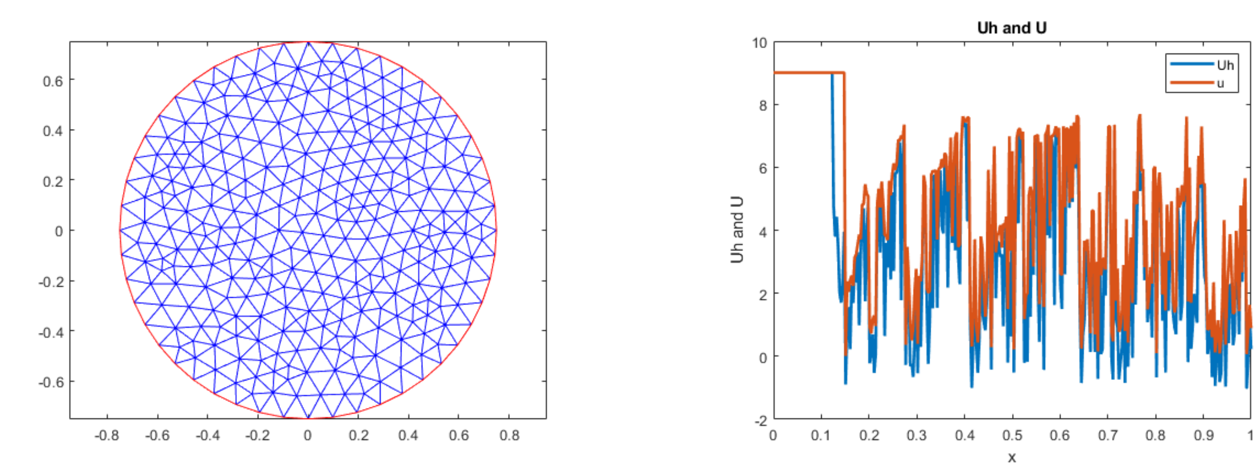

Figure 8: Exact U and Finite element approximation U_h

b

The L^2 norm that we calculated are, It seems that there might be some programming error here, as the error

Refinement Level 1	Refinement Level 2	Refinement Level 3	Three Refinement Level 4
28.2870583512373	55.0666816966912	111.943518671780	223.962786895082

Table 3: L^2 error calculated for this mesh.

increases with refinement levels.

c

We can either uses high order element or/and increases the accuracy of quadurate rule.

References

[1] Arif Wicaksana and Tahar Rachman. *FEM_TIA*. Angewandte Chemie International, 2018.