ME 280A
Homework 1
Ahmad Zareei

Introduction

In this problem set, we want to solve 1-D conservation equation using finite element method. Conservation equation could be written as:

$$\frac{d}{dx}\left(A_1\frac{du}{dx}\right) = f\left(x\right)$$

where A_1 could be a function of x. Conservation equation appears in many branches of physics. The most general form of conservation equation could be written as

$$\dot{V} + \nabla.\mathbf{F} = f \tag{1}$$

where V is some parameter changing with time whose flux is $\mathbf{F}$. Moreover, f is some external function that is forcing the change of V. As some example consider the Euler equations in fluid mechanics

$$\frac{\partial\rho}{\partial t} + \nabla.\left(\rho\mathbf{u}\right) = 0 \tag{2}$$

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla.\left(\mathbf{u}\times\rho\mathbf{u}\right) + \nabla p = 0 \tag{3}$$

$$\frac{\partial E}{\partial t} + \nabla.\left(\mathbf{u}\left(E+p\right)\right) = 0 \tag{4}$$

$$\tag{5}$$

where first equation is the mass conservation, second one is the momentum conservation and the third one is the energy conservation. All these equations are written in conservational form. Now if we suppose that we are solving for steady state in only onedimension, then the time derivative term i.e. $\partial/\partial t$ is equal to zero and we have the exact form of equation that we are solving in this problem set. Another famous example is the equation of static mechanical conservation equilibrium as

$$\nabla.\sigma + f = 0 \tag{6}$$

where σ is the stress tensor and f is the external force on the object. In this problem set we will solve these types of equations in just 1 dimension.

Objectives

1. Solving a 1-D steady state conservation equation with a driving force using Finite element method with equally sized elements.

2. Understanding the error analysis and comparing the result with exact solution.

3. Finding the relation between the error and number of the elements

Problem and Procedure

1. Solve the following boundary value problem, with domain $\Omega = (0, L)$, analytically :

$$\frac{d}{dx}\left(A_1 \frac{du}{dx}\right) = k^2 \sin\left(\frac{\pi k x}{L}\right) + 2x \tag{7}$$

where $A_1 = \text{const.} = 0.2$ and $k = \text{const.}$, $L = 1$ and $u(0) = \Delta_1 = 0$ and $u(L) = \Delta_2 = 1$.

Solution:

Integrating the equation twice, we end up with

$$u(x) = -\frac{L^2}{A_1 \pi^2} \sin\left(\frac{\pi k x}{L}\right) + \frac{x^3}{3A_1} + Cx + D \tag{8}$$

Introducing the boundary conditions of $u(0) = 0$ and $u(L) = 1$, we will have

$$D = 0 \qquad C = \left(1 - \frac{L^3}{3A_1}\right)\frac{1}{L} \tag{9}$$

So the solution, satisfying the boundary condition will be:

$$u(x) = -\frac{L^2}{A_1 \pi^2} \sin\left(\frac{\pi k x}{L}\right) + \frac{x^3}{3A_1} + \left(1 - \frac{L^3}{3A_1}\right)\frac{x}{L} \tag{10}$$

plots of the solution of the equation for different values of k are as in figures 1 and 2.

2. Now solve this with the finite element method using linear equal-sized elements. How many elements (N) are required to achieve the error criteria. The error is defined as

$$e^N = \frac{||u - u^N||_{A_1(\Omega)}}{||u||_{A_1(\Omega)}} \tag{11}$$

where

$$||u||_{A_1(\Omega)} = \sqrt{\int_\Omega \frac{du}{dx} A_1 \frac{du}{dx} dx}. \tag{12}$$

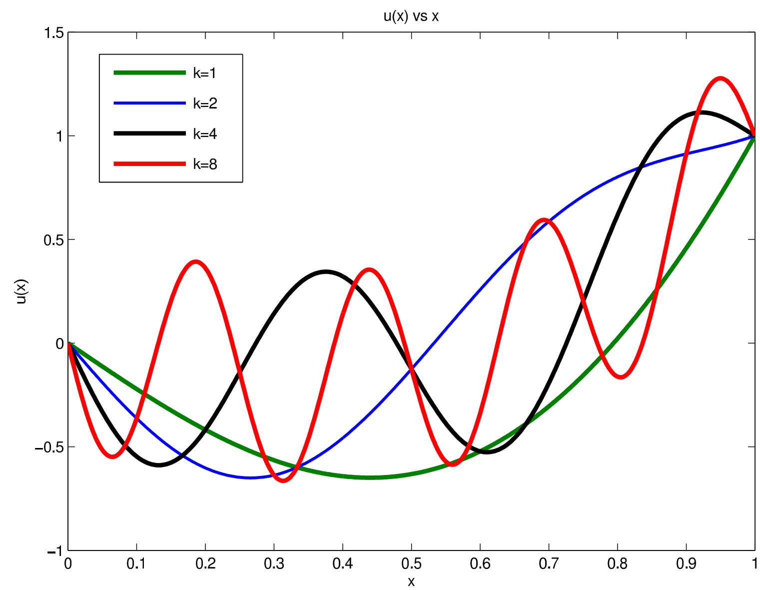

Figure 1: Exact solution of $u(x)$ for different values of $k = \{1, 2, 4, 8\}$

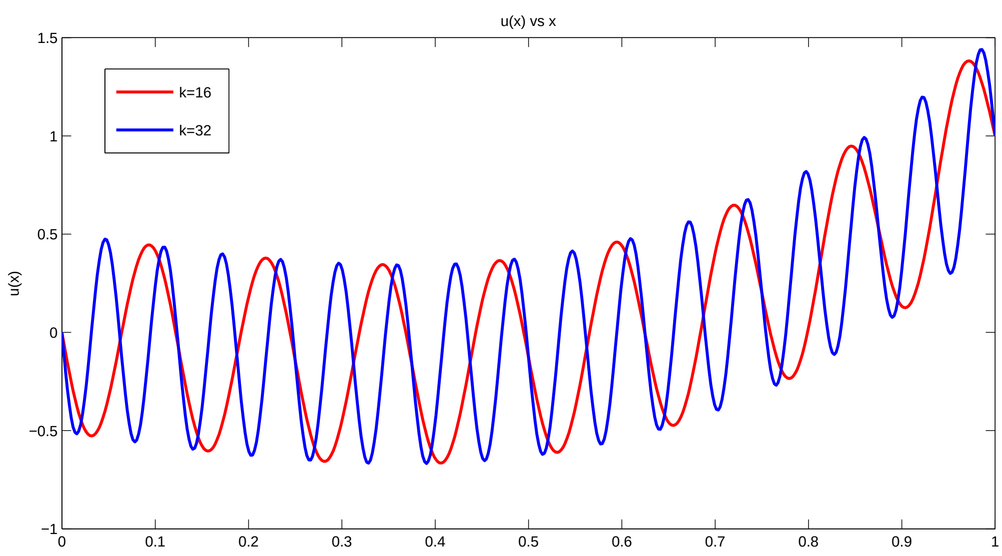

Figure 2: Exact solution of $u(x)$ for different values of $k = \{16, 32\}$

Solution:

For Solving the conservation equation using finite element method, we follow the procedure of (i) finding the weak form of the equation, (ii) discretizing and finding the system of equations, and (iii) solving for the problem. First we define $r(x)$ as

$$r(x) = \frac{d}{dx}\left(A_1 \frac{du}{dx}\right) - f(x) \tag{13}$$

where

$$f(x) = k^2 \sin\left(\frac{\pi k x}{L}\right) + 2x \tag{14}$$

In order for $u(x)$ to be a solution, the following integral

$$\int_\Omega \left(\frac{d}{dx}\left(A_1 \frac{du}{dx}\right) - f(x)\right) v \mathrm{d}x = 0 \tag{15}$$

should be true for every test function $v(x)$ in the domain Ω. With integrating by parts, we get

$$A_1 \frac{du}{dx} v|_{\partial\Omega} - \int_\Omega A_1 \frac{du}{dx}\frac{dv}{dx} dx - \int_\Omega f(x) v \mathrm{d}x = 0 \tag{16}$$

we choose $v(x)$ to be zero at the boundaries. So the equation will become

$$\int_\Omega A_1 \frac{du}{dx}\frac{dv}{dx} \mathrm{d}x = -\int_\Omega f(x) v \mathrm{d}x \tag{17}$$

Now we suppose that $u = \sum a_j \phi_j$ and $v = \sum b_i \phi_i$, for some basis functions ϕ_i. We should note that $\{b_i\}$ are arbitrary since the integral should be true for any function of $v(x)$. The goal in here is to find system of equations for $\{a_i\}$ such that $u = \sum a_j \phi_j$ be the solution. Subistituting $u(x)$ and $v(x)$ in our equation, we obtain

$$\sum_i b_i \left(\sum_j \left(\int_\Omega A_1 \frac{d\phi_i}{dx}\frac{d\phi_j}{dx} \mathrm{d}x\right) a_j + \int_\Omega f(x)\phi_i(x) \mathrm{d}x\right) = 0 \tag{18}$$

Since the equation should be true for any choice of the test function i.e. different b_i, we should have

$$\sum_j \left(\int_\Omega A_1 \frac{d\phi_i}{dx}\frac{d\phi_j}{dx} \mathrm{d}x\right) a_j + \int_\Omega f(x)\phi_i(x) \mathrm{d}x = 0 \tag{19}$$

where this equation could be written as

$$\sum_j M_{ij} a_j = L_i \tag{20}$$

where

$$M_{ij} = \int_\Omega A_1 \frac{d\phi_i}{dx}\frac{d\phi_j}{dx} \mathrm{d}x \qquad L_i = -\int_\Omega f(x)\phi_i(x) \mathrm{d}x \tag{21}$$

we now need to solve the linear system of equation

$$[M]\{a\} = [L] \tag{22}$$

Now suppose we divide the interval (i.e. Ω) by N equal size elements. So we will have

$$0 = x_0 < x_1 < ... < x_i = ih < ... < x_N = 1 \tag{23}$$

where $h = \frac{1-0}{N} = \frac{1}{N}$. As we want to solve, we choose linear basis functions i.e. ϕ_i, as follows

$$\phi_i(x) = \begin{cases} \frac{x-x_{i-1}}{h} & x_{i-1} < x < x_i \\ \frac{-x+x_{i+1}}{h} & x_i < x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

using these elements, we will have

$$\frac{\mathrm{d}\phi_i(x)}{\mathrm{d}x} = \begin{cases} \frac{1}{h} & x_{i-1} < x < x_i \\ \frac{-1}{h} & x_i < x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

Using these basis functions we will find stiffness matrix ($[M]$) and loading vector ($\{L\}$). We will use stamp method i.e. on each element ($x_i < x < x_{i+1}$) we compute the contribution of the element to the stiffness matrix and the loading vector and then we sum over all the elements to find the complete stiffness matrix and loading vector. Contribution of each element, say Ω_i where $x_i < x < x_{i+1}$, to the stiffness matrix is computed as follows:

$$[M] = \begin{pmatrix} \int_{\Omega_i} A_1 \frac{d\phi_i}{dx}\frac{d\phi_i}{dx}\mathrm{d}x & \int_{\Omega_i} A_1 \frac{d\phi_i}{dx}\frac{d\phi_{i+1}}{dx}\mathrm{d}x \\ \int_{\Omega_i} A_1 \frac{d\phi_{i+1}}{dx}\frac{d\phi_i}{dx}\mathrm{d}x & \int_{\Omega_i} A_1 \frac{d\phi_{i+1}}{dx}\frac{d\phi_{i+1}}{dx}\mathrm{d}x \end{pmatrix} \tag{26}$$

where since elements are linear and A_1 is a constant and the intervals are equally distributed with length of h, we obtain

$$[M]_i = \frac{A_1}{h}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}. \tag{27}$$

Contribution of each element, Ω_i, to the loading vector is as

$$\{L\}_i = \begin{Bmatrix} -\int_{\Omega_i} f(x)\phi_i(x)\mathrm{d}x \\ -\int_{\Omega_i} f(x)\phi_{i+1}(x)\mathrm{d}x \end{Bmatrix} \tag{28}$$

The integrals in the loading vector are mapped to the master element and then computed using quadrature points. As an example consider one of the integrals as

$$\int_{\Omega_i} f(x)\phi_i(x)\mathrm{d}x = \int_{\Omega_i'} f(x(\zeta))\phi_i(x(\zeta))\frac{\mathrm{d}x}{\mathrm{d}\zeta}\mathrm{d}\zeta \tag{29}$$

where $\Omega_i'(\zeta)$ is the master element where is $-1 < \zeta < 1$ and the mapping function is

$$x = x_i + \frac{\zeta+1}{2}\left(x_{i+1} - x_i\right). \tag{30}$$

So we will have

$$\int_{\Omega_i'} f(x(\zeta))\phi_i(x(\zeta))\frac{\mathrm{d}x}{\mathrm{d}\zeta}\mathrm{d}\zeta = \int_{\Omega_i'} f(x(\zeta))\hat{\phi}_i(\zeta)\,\frac{h}{2}\,\mathrm{d}\zeta \tag{31}$$

where $\hat{\phi}_i(\zeta)$ and $\hat{\phi_{i+1}}(\zeta)$ in Ω_i' are

$$\hat{\phi}_i(\zeta) = \frac{1+\zeta}{2} \qquad \hat{\phi_{i+1}}(\zeta) = \frac{1-\zeta}{2} \tag{32}$$

Using these mapping we found contribution of each element to the stiffness matrix and the loading vector.

For computing the integrals, we used Guass quadrature method with two nodes. This method for an arbitrary function $f(\zeta)$ states that

$$\int_{-1}^{1} f(\zeta)\,\mathrm{d}\zeta = f\left(+\frac{1}{\sqrt{3}}\right) + f\left(-\frac{1}{\sqrt{3}}\right) \tag{33}$$

At the boundaries we have dirichlet boundary condition, which would mean $a_1 = \Delta_1$ and $a_{N+1} = \Delta_2$ where Δ_1, Δ_2 are given constants at the boundaries (here $\Delta_1 = 0$ and $\Delta_2 = 1$). For implementing the boundary condition, we removed the very first and last row in the stiffness matrix and replaced it by obvious equations of $a_1 = \Delta_1$ and $a_{N+1} = \Delta_2$.

Error as defined in the question is defined as

$$e^N = \frac{||u - u^N||_{A_1(\Omega)}}{||u||_{A_1(\Omega)}} \tag{34}$$

where

$$||u||_{A_1(\Omega)} = \sqrt{\int_\Omega \frac{du}{dx} A_1 \frac{du}{dx} dx}. \tag{35}$$

We evaluated these integrals in the error by mapping to the master element and then computed the integrals with quadrature method.

We wrote a program called *fem1_v2* that gives the error (output) as a function of number of elements (N) and the constant value of k as inputs. Using this function, we did a binary search to find the minimum number of elements that gives relative error less than 0.01. In the binary search, we start from $N = 2$ and each time multiply N by two untill we find the error to be less than 0.01. Now we know that the number of elements is in the interval of $[N_{left} = N/2, N_{right} = N]$. We then find the error for the the middle point i.e. $N_{mid} = (N_{left} + N_{right})/2$ and compute the error for this point. Then we update the interval based on the error of middle point. If it is larger than 0.01 it means that the minimmum number of elements is between $[N_{left}, N_{mid}]$ and if the error is less than 0.01, it means that the minimmum number of elements is between $[N_{mid}, N_{right}]$. Using this procedure, we decrease the length of the interval untill we reach to the point that the length of this interval be 1. This means that the for the left numer the error is larger than 0.01 and for the right number the error is less than 0.01.

The result of number of elements required for error of less than 0.01 is summrized in table 1.

Findings

The result of the minimum number of elements, such that the defined error be less than the tolerance of 0.01, are tabulated in table (1). As we increase the value of k, the number of elements needed will increase. In the discussion, I have talked more about the relation between k and number of elements needed.

k	1	2	4	8	16	32
N	92	146	338	712	1444	2899

Table 1: Minimmum number of elements to achieve relative error of 0.01

Figures for convergence of the solution to the exact solution (found analatycally) for different values of k are plotted in figures: 3, 6, 9, 12, 15, 18. Since it is hard to tell the difference between plots for different N, I have zoomed in some part of the figure to show the difference between different values of N. These figures for different values of k could be found in figures: 4, 7, 10, 13, 16, 19. Figures of error versus number of elements for different values of k are plotted in figures: 5, 8, 11, 14, 17, 20.

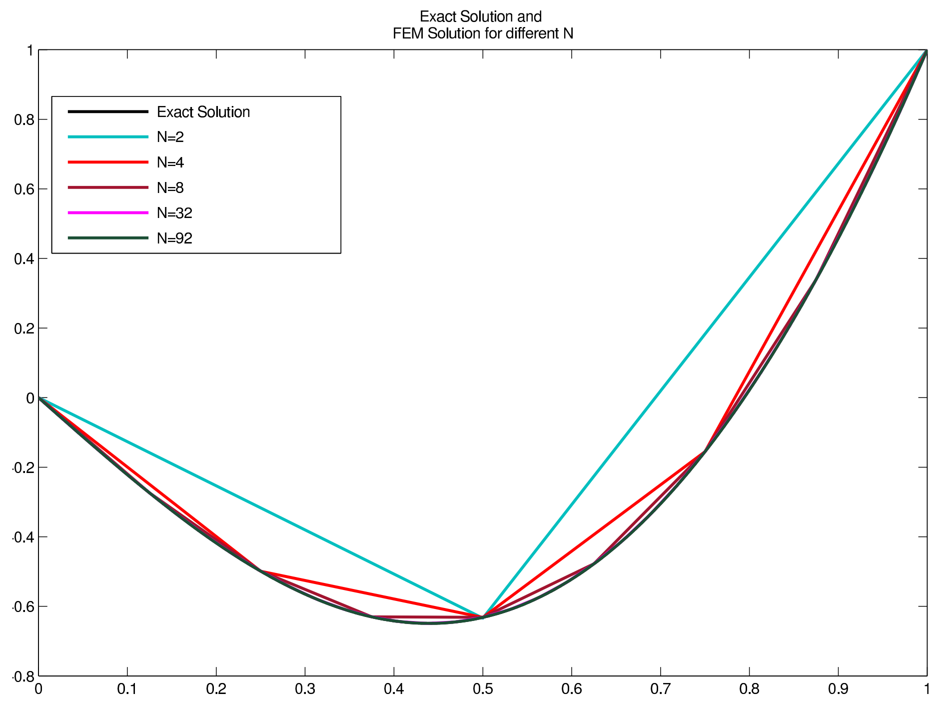

Figure 3: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 1$

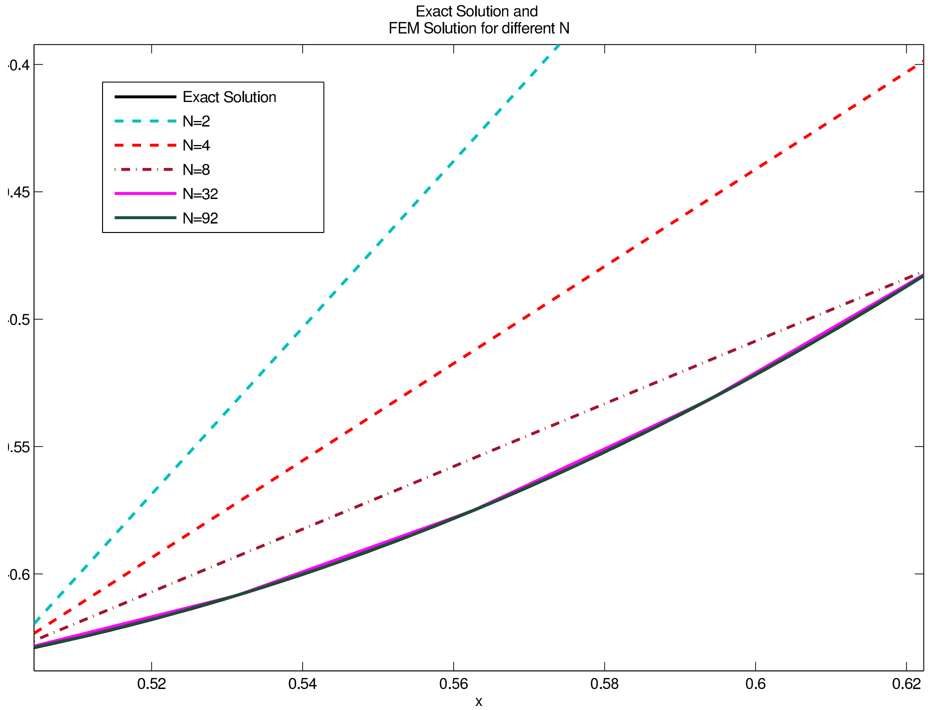

Figure 4: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 1$

Discussion

As it is shown in table (1), number of elements increases as we increase the number k in the forcing term. Finite element method that we used in here is same as using second order finite difference method for the differential part and an integration on the loading part. Suppose we write the equation of FEM method on on general element, we have

$$A_1 \frac{u_{i-1} - 2u_i + u_{i+1}}{h} = \int_{x_{i-1}<x<x_{i+1}} f(x)\phi_i(x)\mathrm{d}x \tag{36}$$

and if f on the r.h.s is linear, $\int f(x)\phi_i(x)\mathrm{d}x = 1/2 f(x_i)2h = f(x_i)h$. So the equation will become:

$$A_1 \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = f(x_i) \tag{37}$$

So when $f(x)$ is linear, this FEM method is like a second order finite difference method. With this approximation, the error of this finite difference method is relative to the maximmum of the absolute value of second derivative of the solution $u''(x)$ and h^2. For this problem that we have, since the right hand side is some $\sin(kx)$ function, the solution will have this form, so the second derivative will be relative to k^2. This will mean that the error of our numerical solution is proportional to Ck^2h^2, where C is some constant. since $h = L/N$, this means that

$$|u - u^N| \sim CL^2 \frac{k^2}{N^2} \tag{38}$$

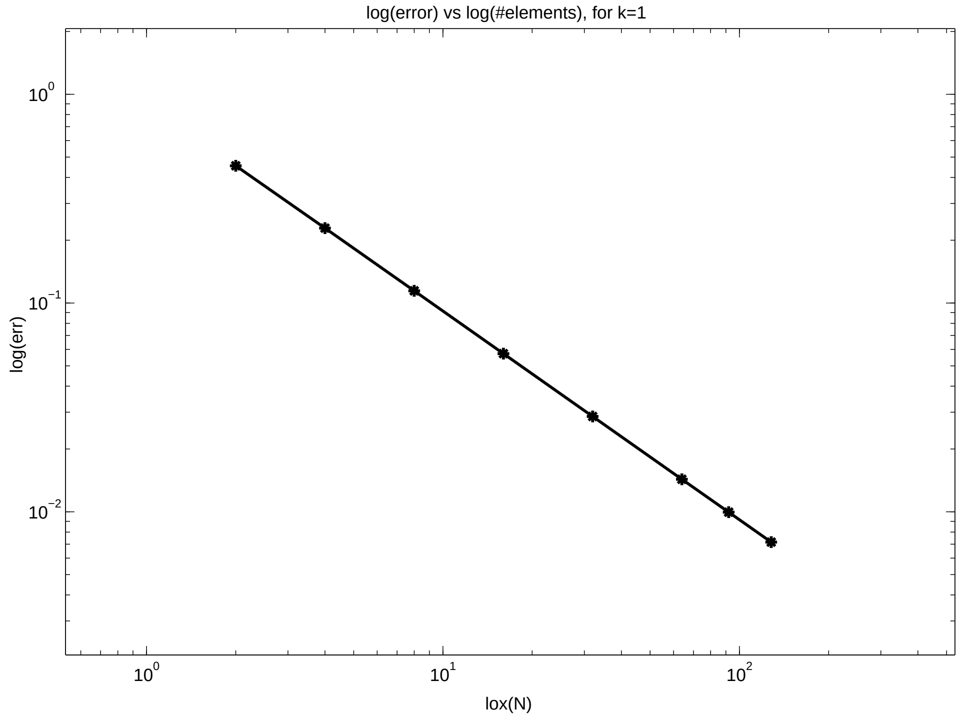

Figure 5: Error vs number of elements in logarithmic scale, for $k = 1$

This means that if we wish to have the same error as we increase k, we should increase number of elements linearly. In other words, relation between number of elements N and k is linear. I have plotted this relation and it is linear (see figure 21).

Note that, If we find values of u up to second order, it means that values of $\mathrm{d}u/\mathrm{d}x$ are correct up to order one, and then with the error defined in this problem set, e^N will be dependent on h and then $1/N$. This would mean that for any k, the slope of error versus number of elements in a logarithmic scale will be -1. This can be seen in all figures of 5, 8, 11, 14, 17, 20 for different values of k.

Appendix

Codes are as follows:

1. The core of the program:

```
function error = fem1_v2(N,k)
%clear all;
%close all;

%N=20; %Number of elements
%k=4; % constant in the equation
l=1.0; %Length of the domain
```

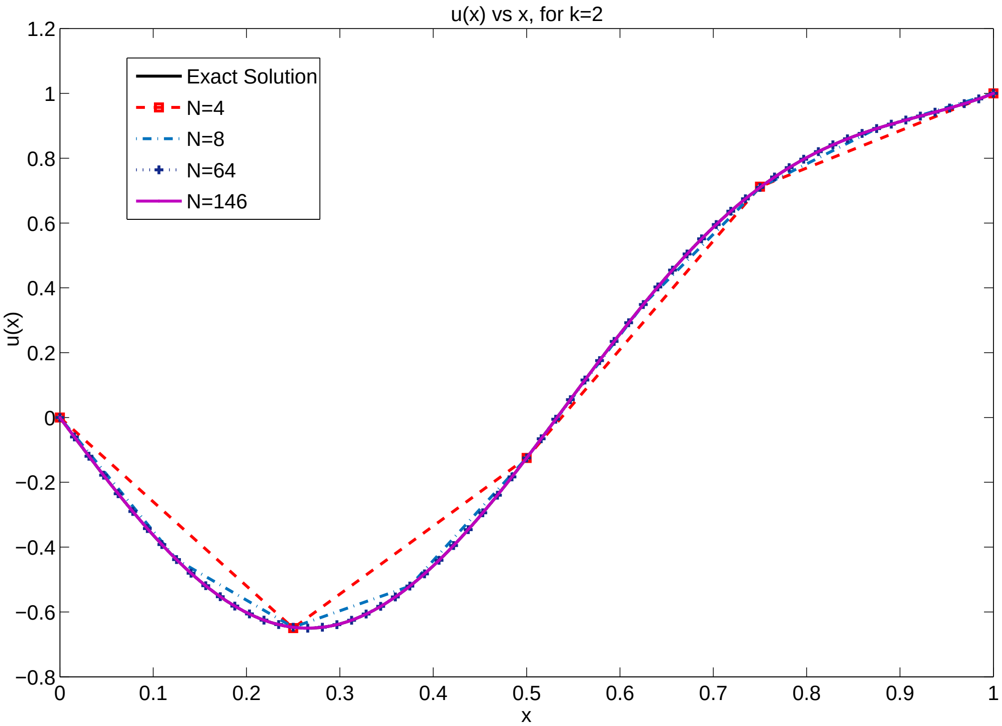

Figure 6: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 2$

```
A1=0.2; %constant in the equation

K=sparse(N+1,N+1);
L=zeros(N+1,1);
p = (1/N)*[0:N]';
e=[[1:N]',[2:N+1]'];
boundary=[1,N+1];

for element = 1:size(e,1)
    points = p(e(element,:));
    h=points(2)-points(1);
    K_stamp = A1/h*[1,-1;-1,1];
    f = @(w) (k^2*sin(pi*(points(1) + (w+1)*h/2)*k/1)+2*(
        points(1) + (w+1)*h/2)).*(1-w)/2;
    g = @(w) (k^2*sin(pi*(points(1) + (w+1)*h/2)*k/1)+2*(
        points(1) + (w+1)*h/2)).*(1+w)/2;
    L_stamp = [-(f(1/sqrt(3))+f(-1/sqrt(3)));-(g(1/sqrt(3))+g
        (-1/sqrt(3)))]*h/2;
    K(e(element,:),e(element,:)) = K(e(element,:),e(element,:)
        ) + K_stamp;
    L(e(element,:)) = L(e(element,:))+L_stamp;
```

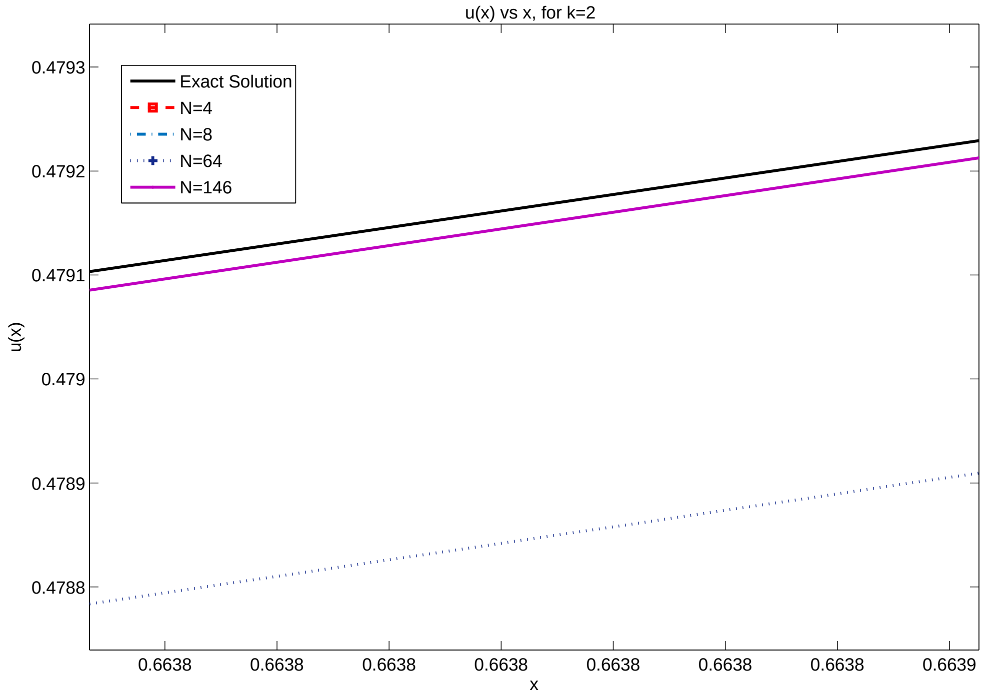

Figure 7: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 2$

```
end
K(1,:)= 0; K(N+1,:) = 0;
K(1,1) = 1; K(N+1,N+1) = 1;
L(1) = 0;  L(N+1) = 1;
u = K\L;
%exact solution repetition
C = (1-l^3/3/A1)/l;
uu = -l^2/A1/pi^2*sin(pi*k*p/l) + p.^3/3/A1 + C*p;
%plot(p,u,'-*'); hold on; plot(p,uu,'.- b');

%computing the error
error = 0;
error_denom = 0;
for element = 1:size(e,1)
    points = p(e(element,:));
    h=points(2)-points(1);
    ff = @(w) (-l*k/(A1 * pi)*cos(pi*k*(points(1) + (w+1)*h/2)
        /l) + (points(1) + (w+1)*h/2).^2/A1 + C);
    gg = @(w) -u(element)/h + u(element+1)/h;
    error = error + quad(@(w) A1*(ff(w)-gg(w)).^2,-1,1)*h/2;
```

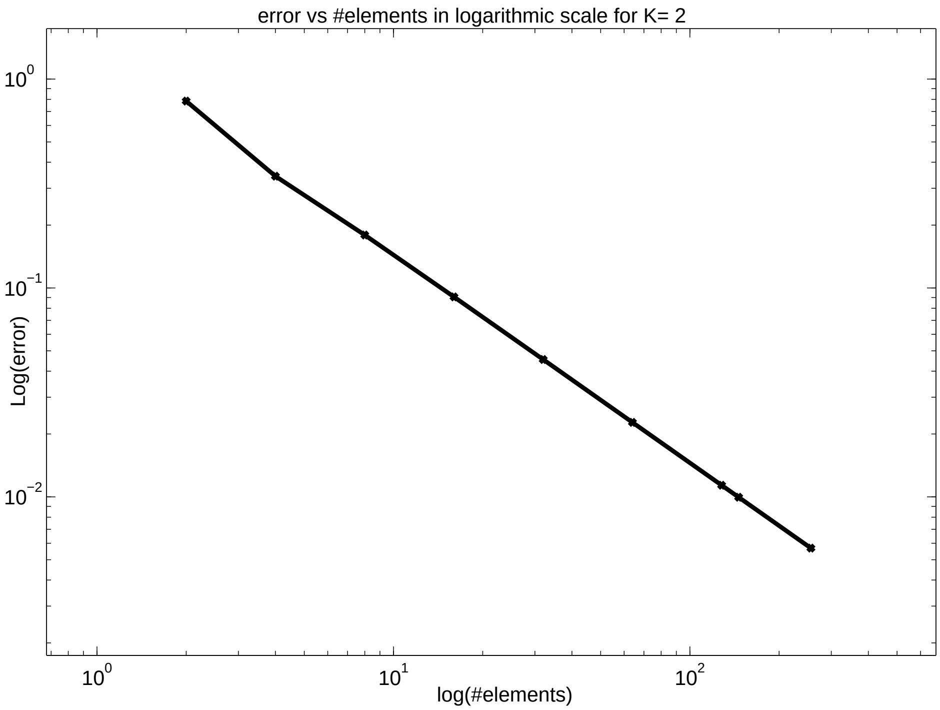

Figure 8: Error vs number of elements in logarithmic scale, for $k = 2$

```
    error_denom = error_denom + quad(@(w) A1*ff(w).^2,-1,1)*h/2;
end
error = sqrt(error)/sqrt(error_denom);
```

2. The code finding the minimmum number of elements for error less than 0.01:

```
function fem1_main

k = 2.^[0,1,2,3,4,5];


for i=0:5
    k=2^i;
    N=1;
    error =1;
    while error >0.01
        N=2*N;
        error = fem1_v2(N,k);
    end
    M = N/2;
    while abs(N-M)>1
```

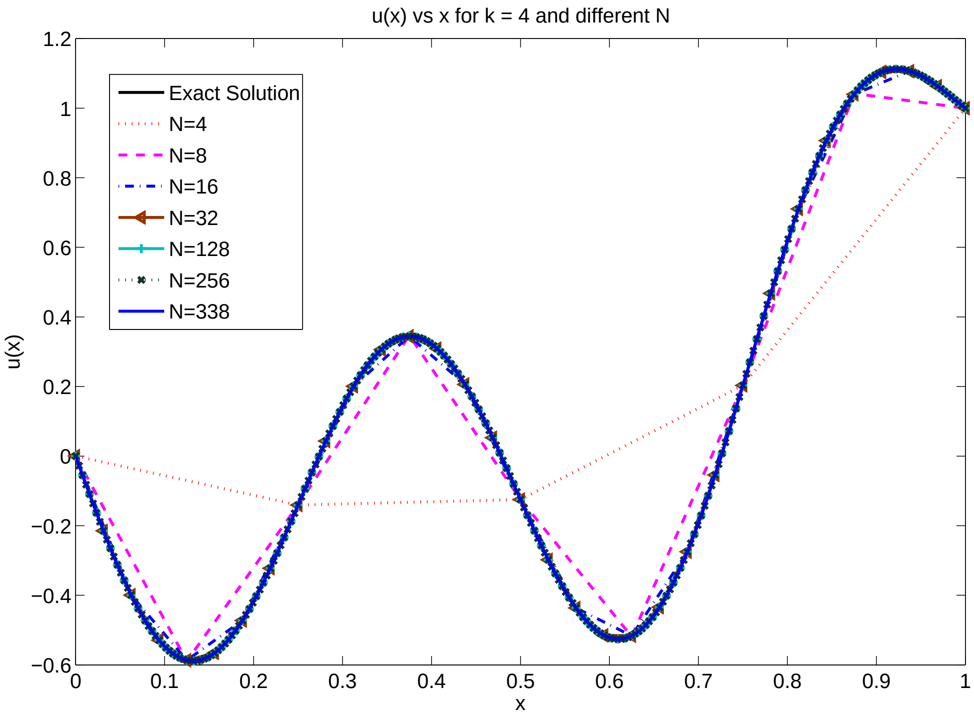

Figure 9: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 4$

```
            mid = floor((N+M)/2);
            error = fem1_v2(mid,k);
            if(error>0.01)
                M=mid;
            else
                N=mid;
            end
        end
        %mid = floor((N+M)/2);
        error = fem1_v2(N,k);
        fprintf('k = %d, N = %d, error = %5.3f\n',k,N,error);

end
```

3. The code for plotting of error and solution convergence to exact solution:

```
function fem1_main2

Nmax = [92,146,338,712,1444,2899];
l=1.0; %Length of the domain
A1=0.2; %constant in the equation
C = (1-l^3/3/A1)/l;
```

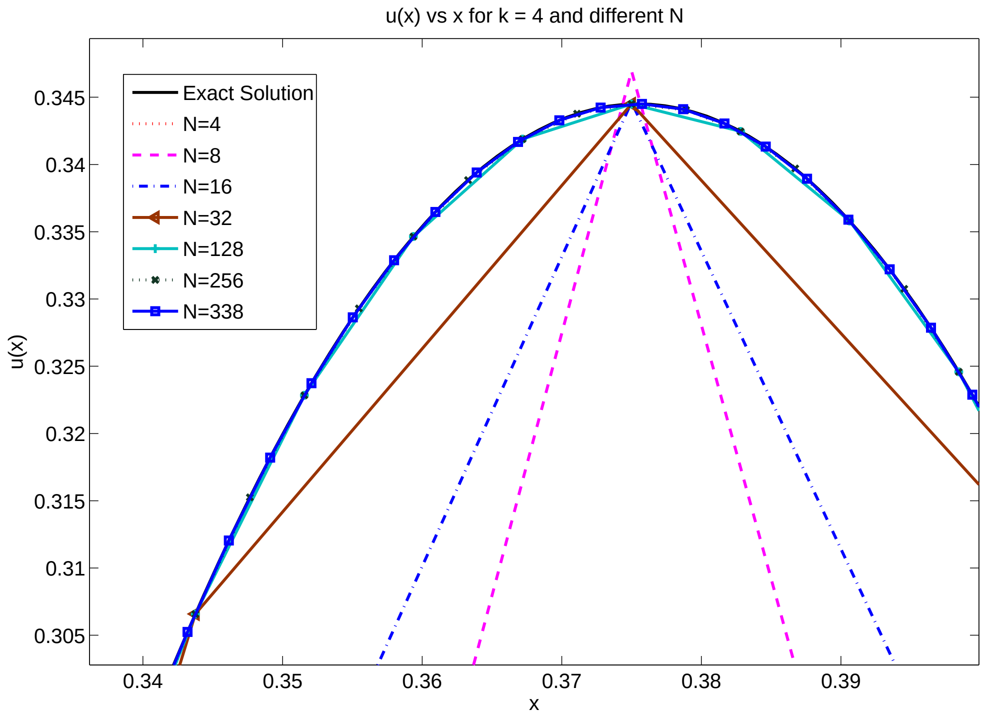

Figure 10: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 4$

```
x=linspace(0,1,5000);
err = 0;


for i=0:5
%for i=0
    index = 1;
    k=2^i;
    N=1;
    error =1;
    figure
    uu = -l^2/A1/pi^2*sin(pi*k*x/l) + x.^3/3/A1 + C*x;
    plot(x,uu);
    hold on;
    while N < Nmax(i+1)
        N=2*N;
        [err(index),u] = fem1_v3(N,k);
        plot([0:N]/N,u);
        index = index+1;
    end
```

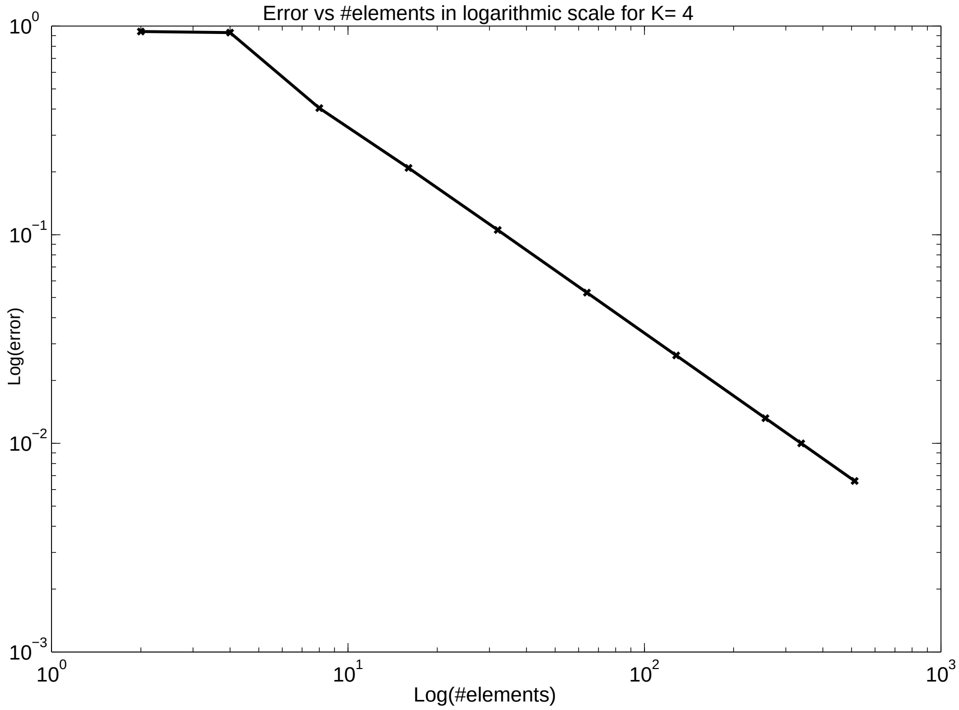

Figure 11: Error vs number of elements in logarithmic scale, for $k = 4$

```
        N=Nmax(i+1);
        [err(index),u] = fem1_v3(N,k);
        plot([0:N]/N,u);
        hold off
        M = [2.^(1:index-2),Nmax(i+1),2.^(index-1)];
        err = [err(1:index-2),err(index), err(index-1)];
        figure
        loglog(M,err);
        caption = sprintf('K= %d',k);
        title(caption);
end
```

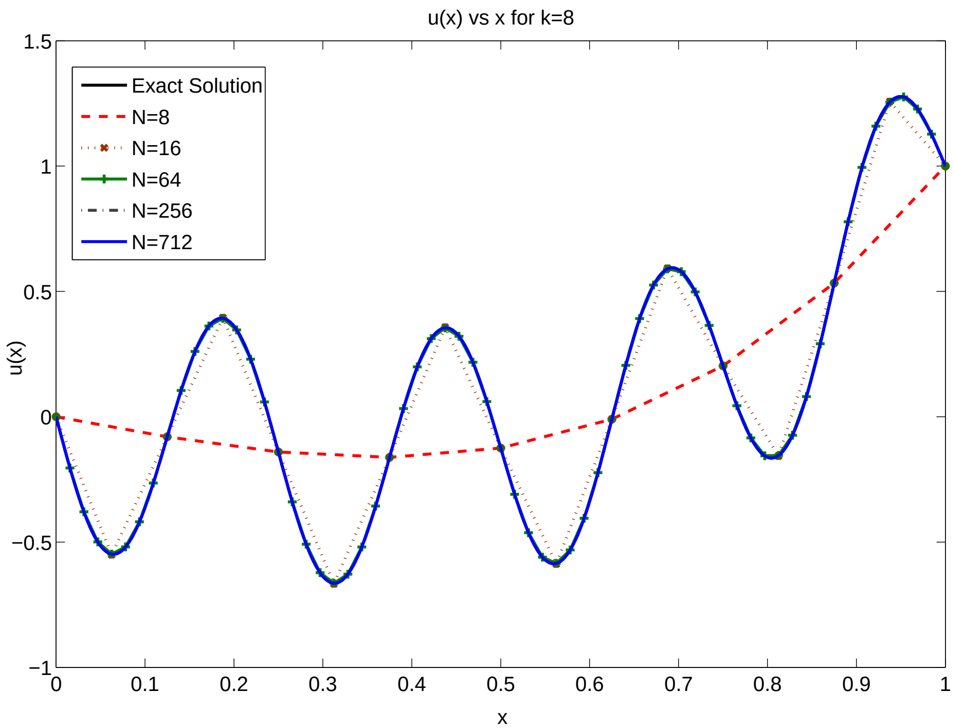

Figure 12: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 8$

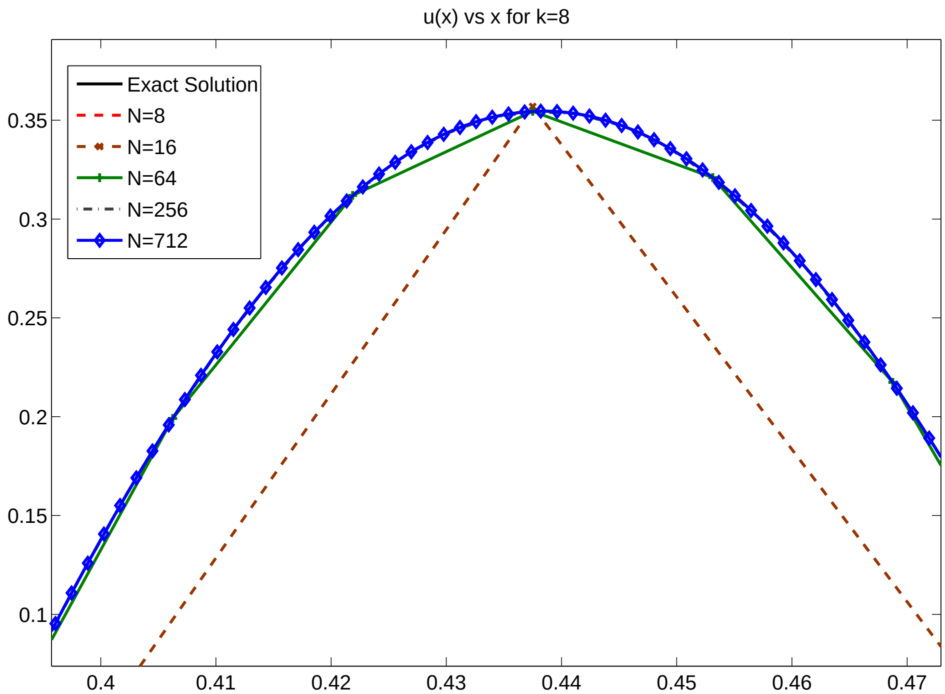

Figure 13: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 8$

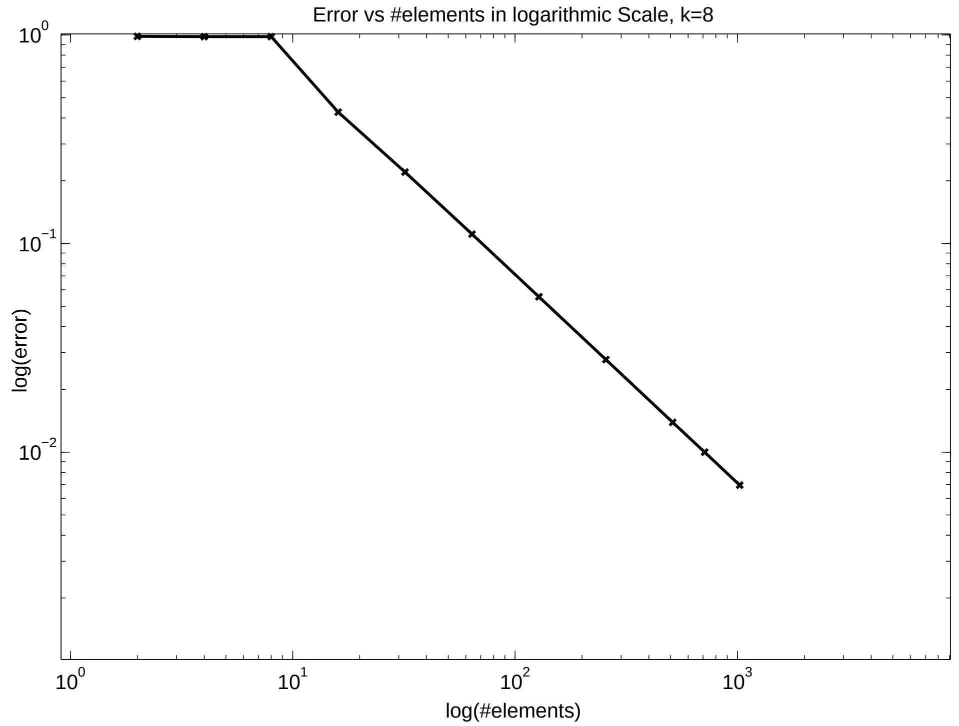

Figure 14: Error vs number of elements in logarithmic scale, for $k = 8$

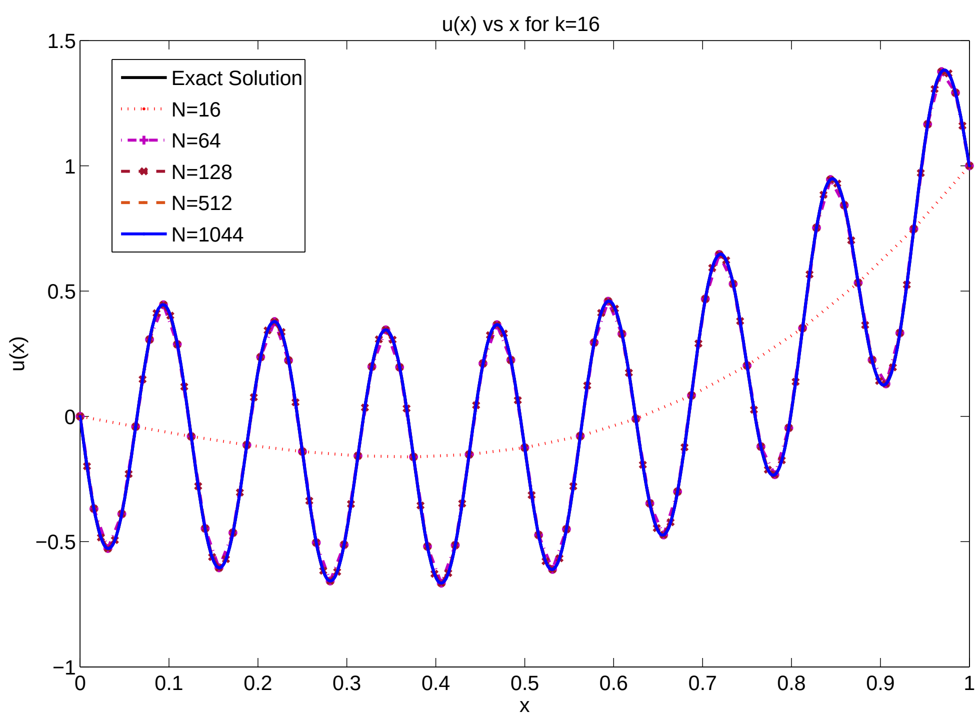

Figure 15: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 16$

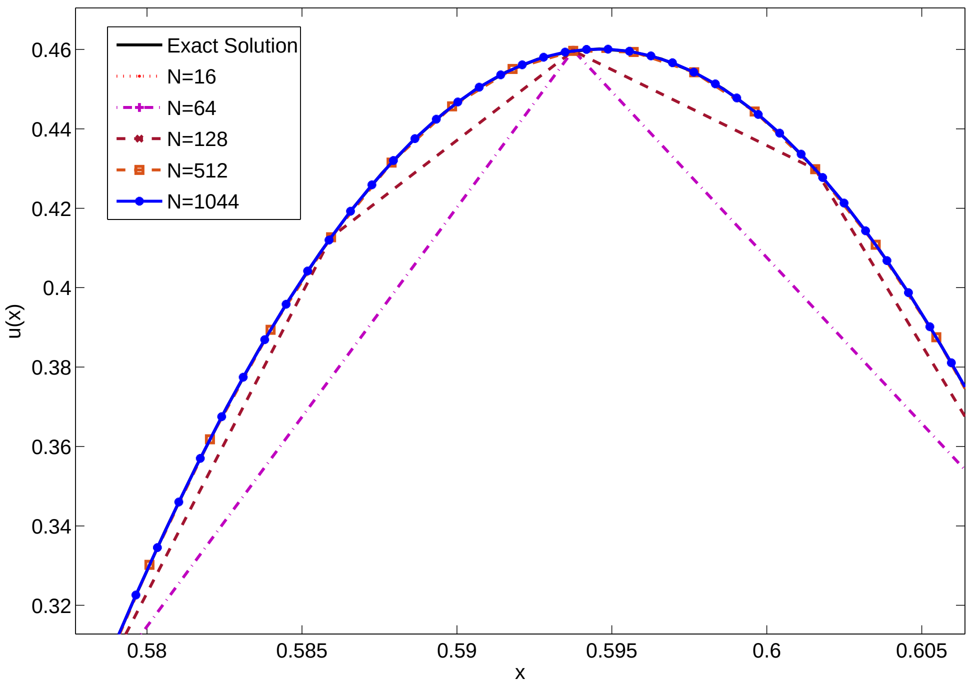

Figure 16: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 16$

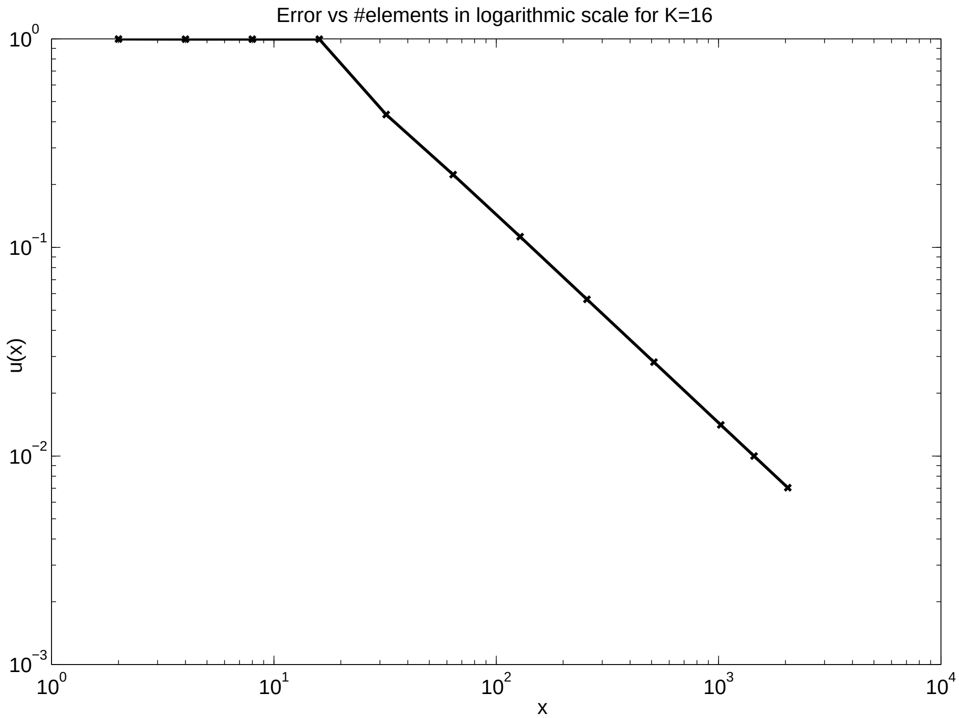

Figure 17: Error vs number of elements in logarithmic scale, for $k = 16$

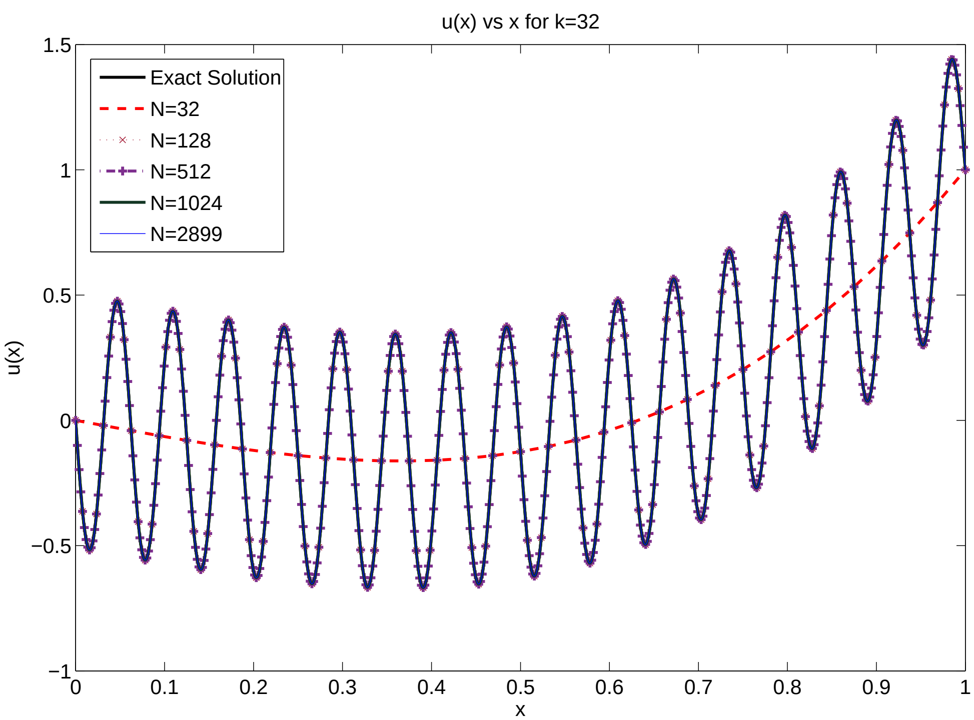

Figure 18: $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 32$

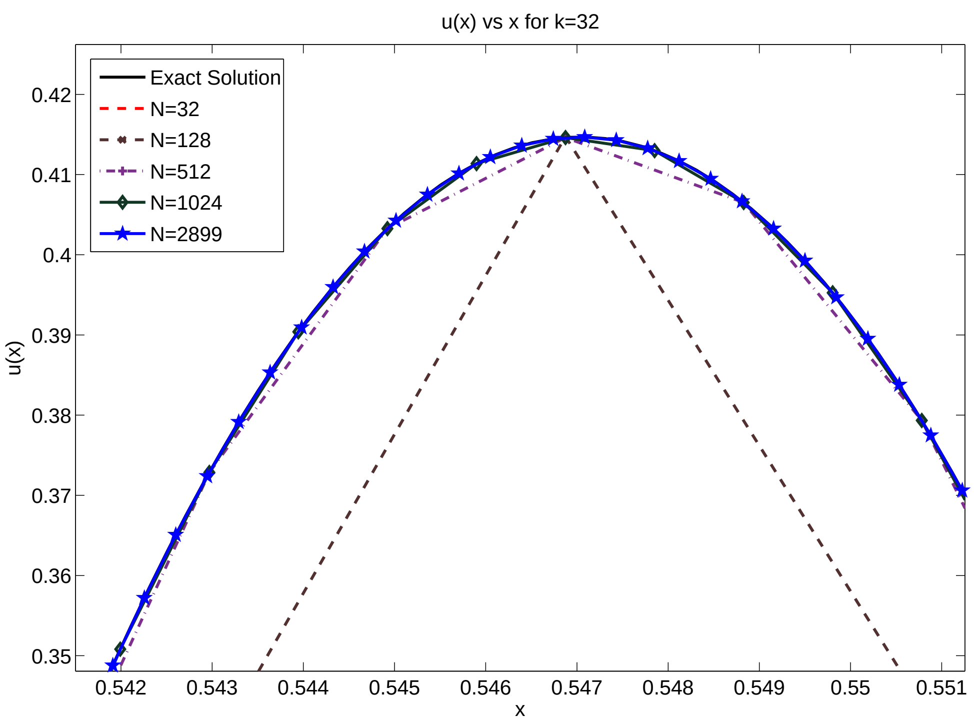

Figure 19: Zoomed plot of $u(x)$ vs x, Exact Solution and FEM solutions for differenet N, as $k = 32$

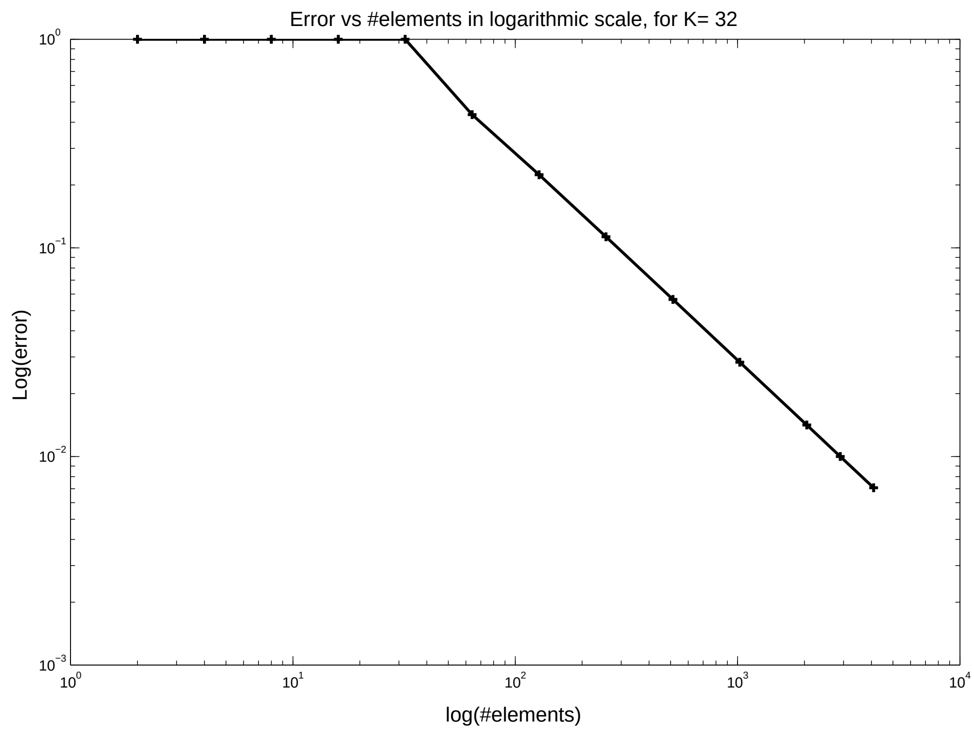

Figure 20: Error vs number of elements in logarithmic scale, for $k = 32$

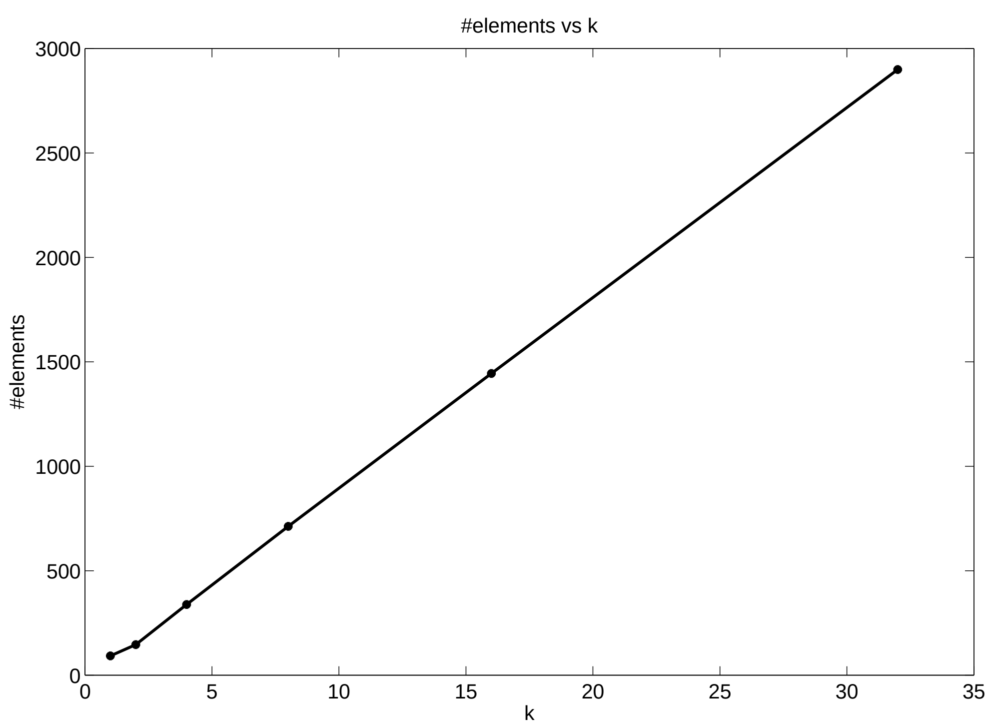

Figure 21: N versus k for error less than 0.01, Linear depence is discussed in discussion part