Project 2: Higher Order Element

Zhipeng Yu

1. Introduction to the problem:

Solve the following boundary value problem, with domain (0, L), analytically, the conditions are given like this.

$$\frac{d}{dx}\left(E\frac{du}{dx}\right) = xk^3cos(\frac{2\pi kx}{L})$$

$$E = 0.2$$

$$k = 12$$

$$L = 1$$

$$u(0) = \Delta_1 = given\ constant = 3$$

$$u(L) = \Delta_2 = given\ constant = -1$$

What am I going to do is to get a solution of "u" without direct integral.
How am I going to do is to combine some linear simple functions and add them up such that getting as closed as possible to the true solution.
For example: function like "(x-1)x/2,(1+x)(1-x),x(x+1)/2".
Although these functions seem simple, they are really powerful if you get fairly large number of them.

2. Objective

Goals: As we are going to find an approximate solution, we are supposed to make error less equal to 0.04.
And bellow, it is how to calculate the error.

$$e^N \stackrel{\text{def}}{=} \frac{||u-u^N||_{E(\Omega)}}{||u||_{E(\Omega)}} \leq TOL = 0.04,$$

$$||u||_{E(\Omega)} \stackrel{\text{def}}{=} \sqrt{\int_\Omega \frac{du}{dx} E \frac{du}{dx}\, dx}$$

Besides, we are going to find what the smallest number of "functions" is for given "k" when p=1, p=2 or p=3. Actually, we divide a given domain into N elements. And every element has a unique function. So now the problem becomes that what is the best N for different p given k?

$$\begin{aligned} p = 1 &\Rightarrow N =? \\ p = 2 &\Rightarrow N =? \\ p = 3 &\Rightarrow N =? \end{aligned}$$

When changing p or N, what is the error going to be? We will try to figure it out soon.

3. My procedure

First we decide to use the Weak Formulation to solve it.

Find $u \in H^1(\Omega)$ $u|_{\Gamma_u} = d$ such that $\forall \nu \in H^1(\Omega), \nu|_{\Gamma_u} = 0$

$$\int_\Omega \frac{d\nu}{dx} E \frac{du}{dx}\, dx = \int_\Omega f\nu\, dx + t\nu|_{\Gamma_t}.|$$

We approximate "u" by:

$$u^h(x) = \sum_{j=1}^{N} a_j \phi_j(x).$$

If we choose "v" with the same approximation functions, but a different linear combination, we get "v" like this:

$$\nu^h(x) = \sum_{i=1}^{N} b_i \phi_i(x),$$

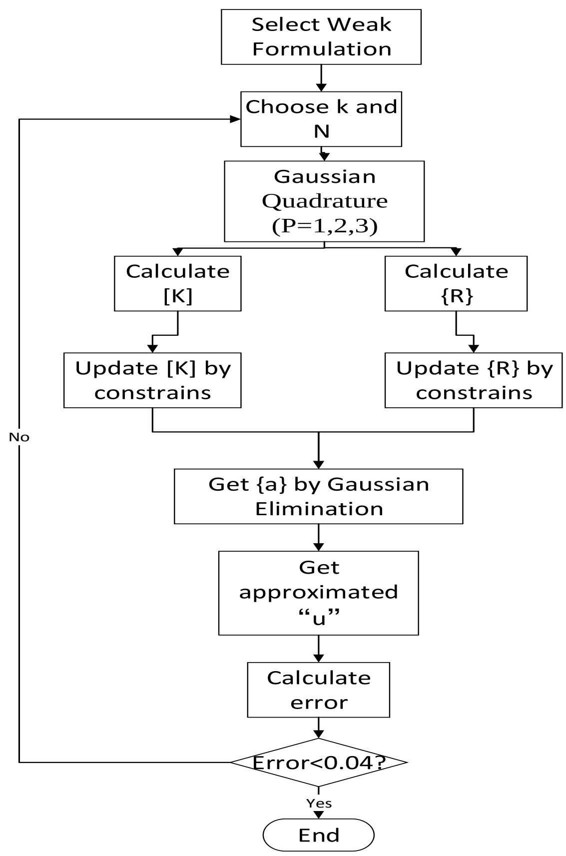
Since the “v” are arbitrary (formulation definition), the “bi” are arbitrary, therefore

$$\sum_{i=1}^{N} b_i \left(\sum_{j=1}^{N} K_{ij} a_j - R_i\right) = 0 \Rightarrow [K]\{a\} = \{R\},$$

$$K_{ij} \stackrel{\text{def}}{=} \int_{\Omega} \frac{d\phi_i}{dx} E \frac{d\phi_j}{dx}\, dx \text{ and}$$

$$R_i \stackrel{\text{def}}{=} \int_{\Omega} \phi_i f\, dx + \phi_i t|_{\Gamma_t},$$

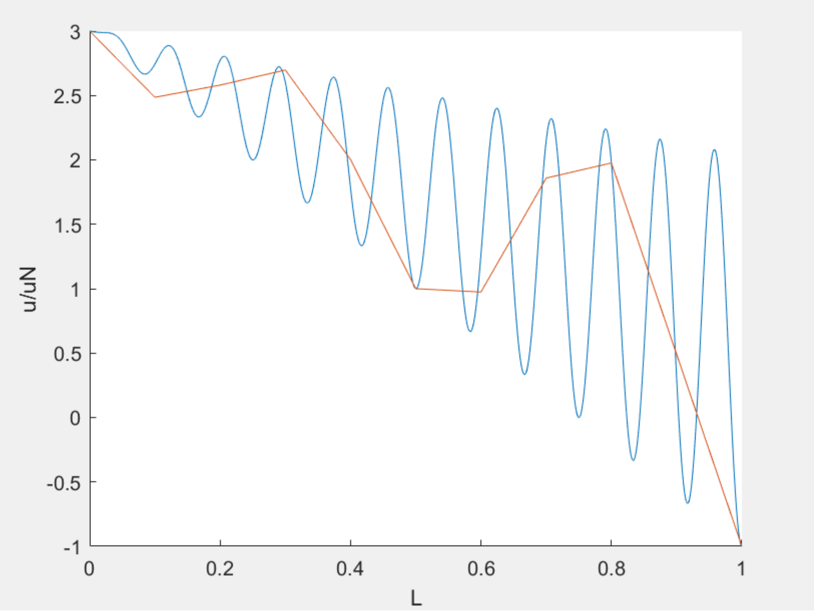
According to that, we will use some mathematical trick to simplify the integral calculation such as Gaussian quadrature (ϕi represents the simple “function” that we build by ourselves). Then will get [K] and {R} and add constrains to them. And, we will solve this linear algebra problem by Gaussian Elimination. Once we get {a}, we get approximated “u”. Based on that, we can do further test on errors and try to find the best N for each k.



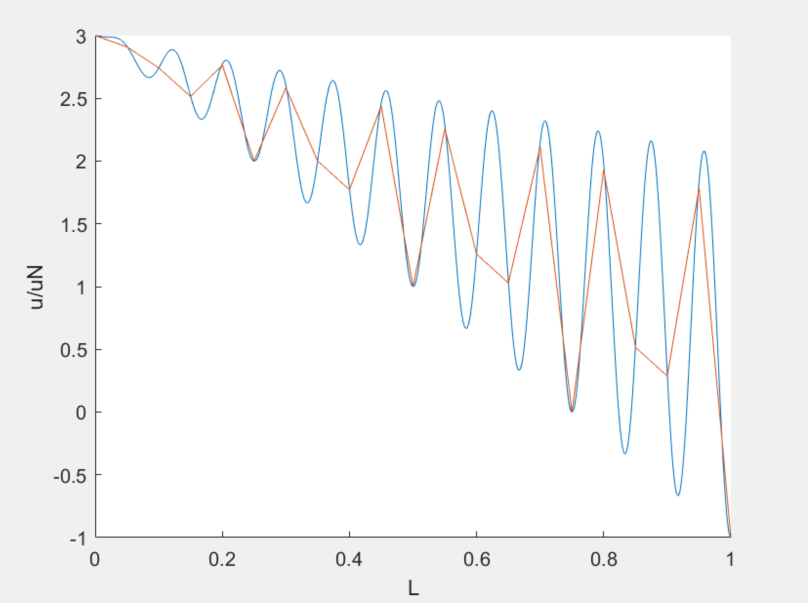
4. Findings

1) Best N for each p:
   p=1, N=542;
   p=2, N=78;
   p=3, N=48

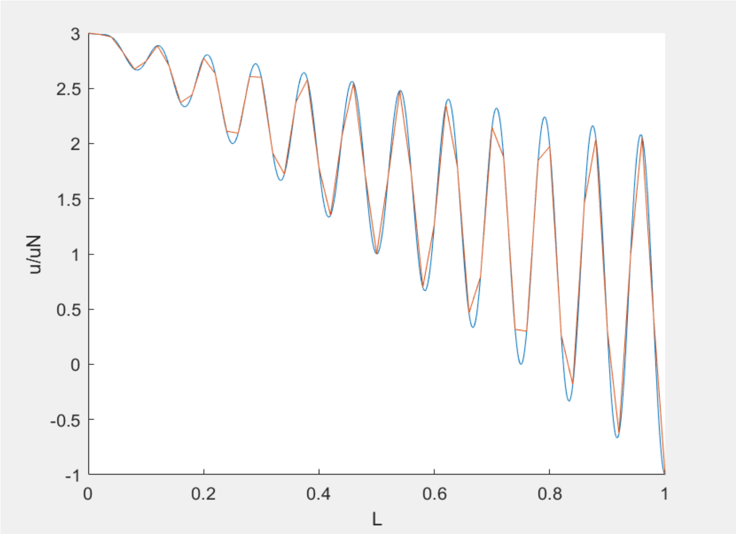
2) **p=1, N=10**



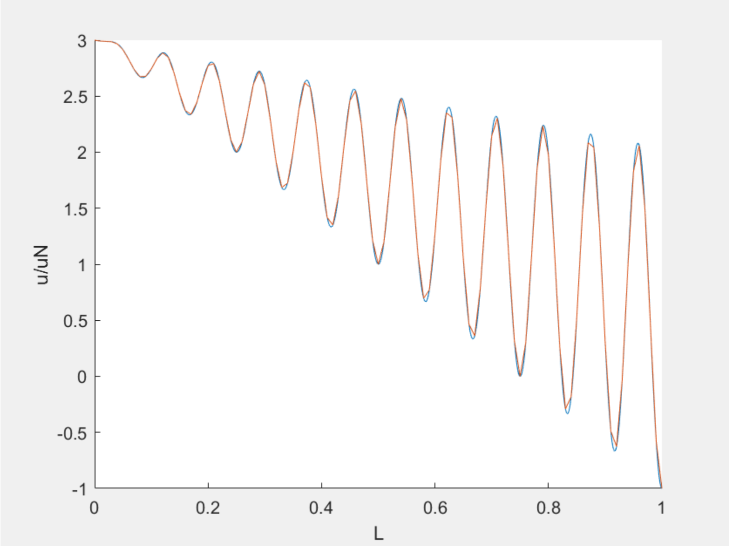
p=1, n=20



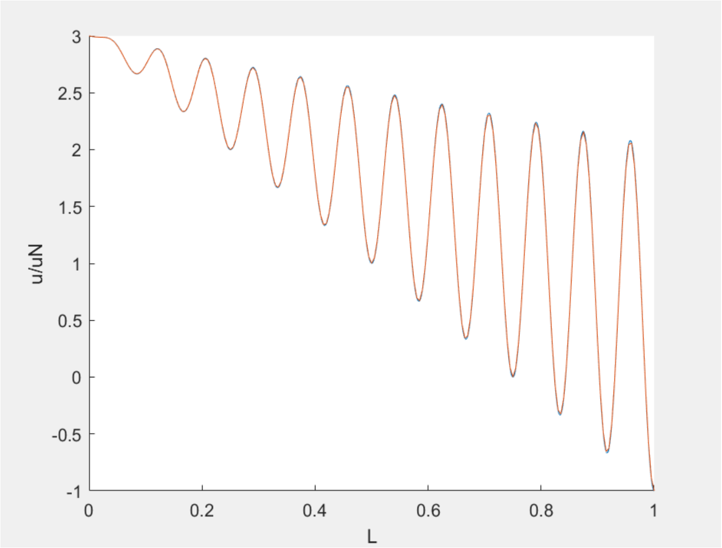
p=1, n=50



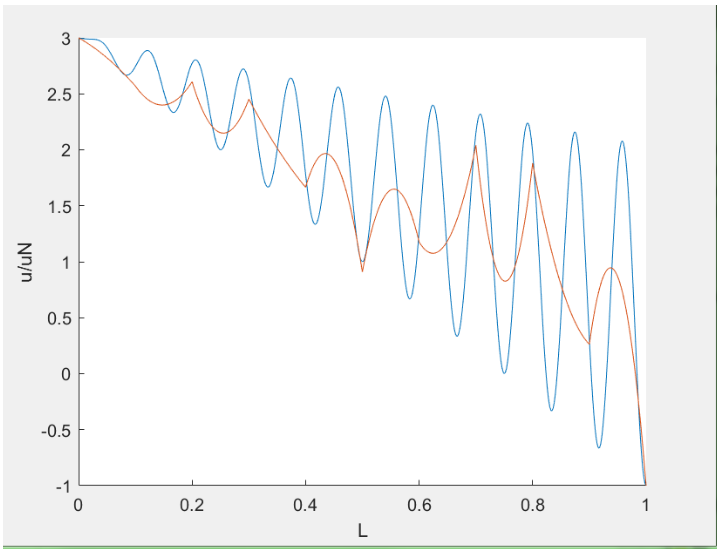
p=1, n=100



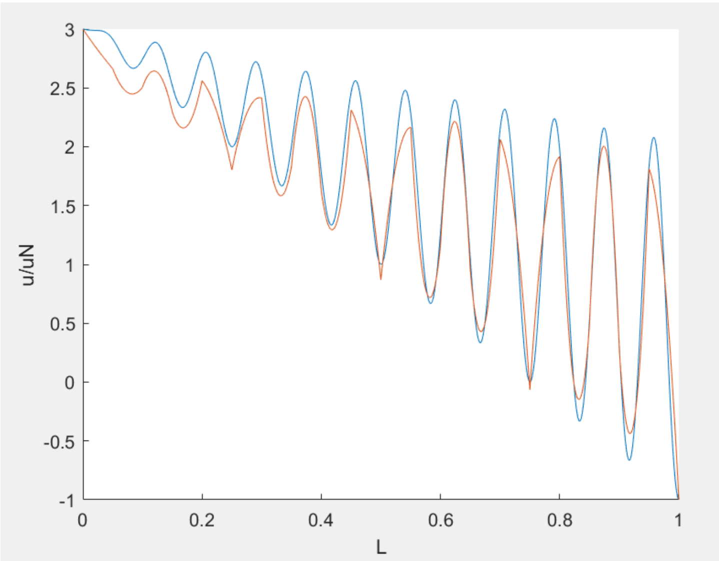
p=1, n=200



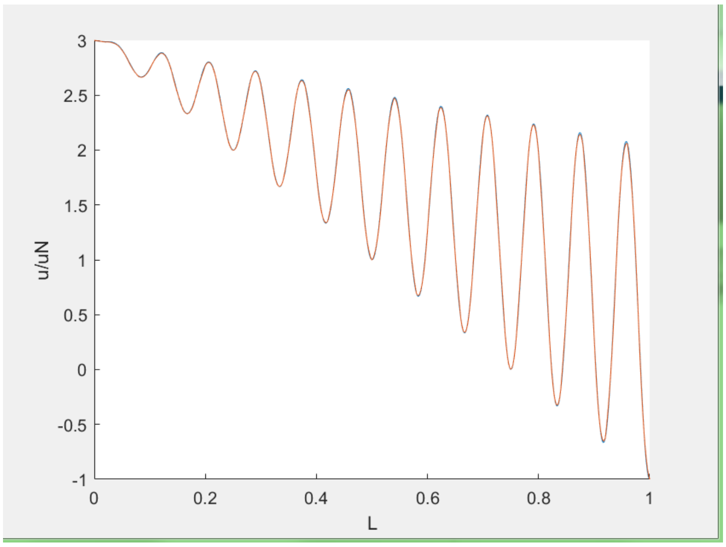
When p=2, n= 10, 20 , 50 , 100



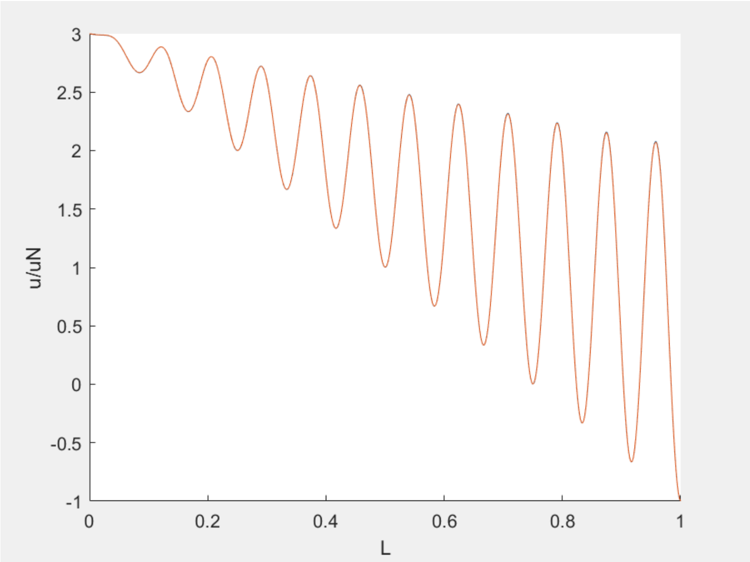
u/uN
L



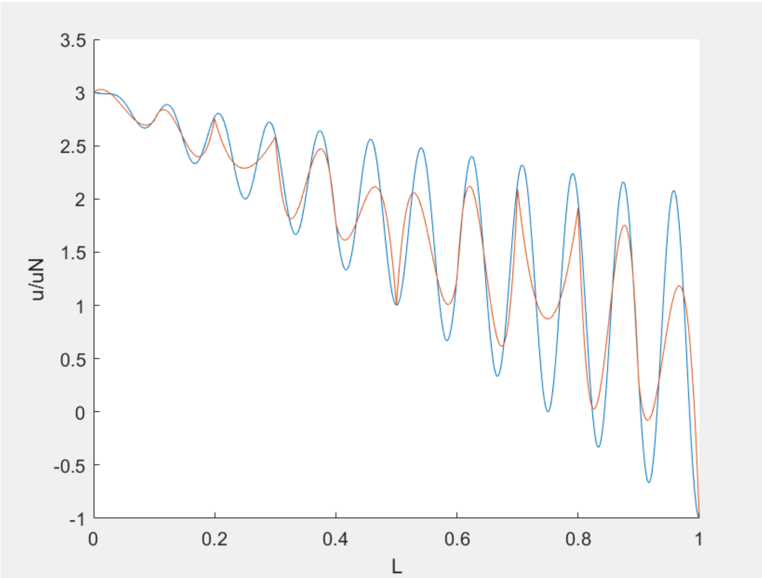
u/uN
L

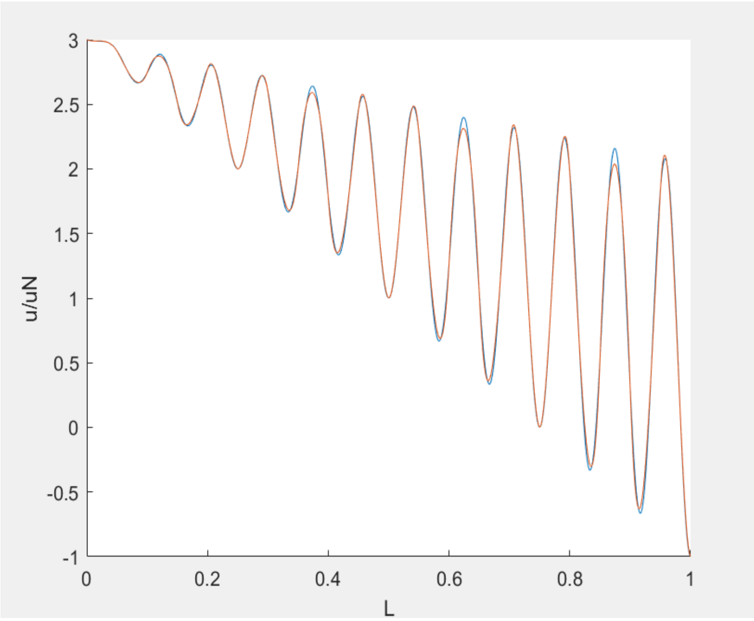


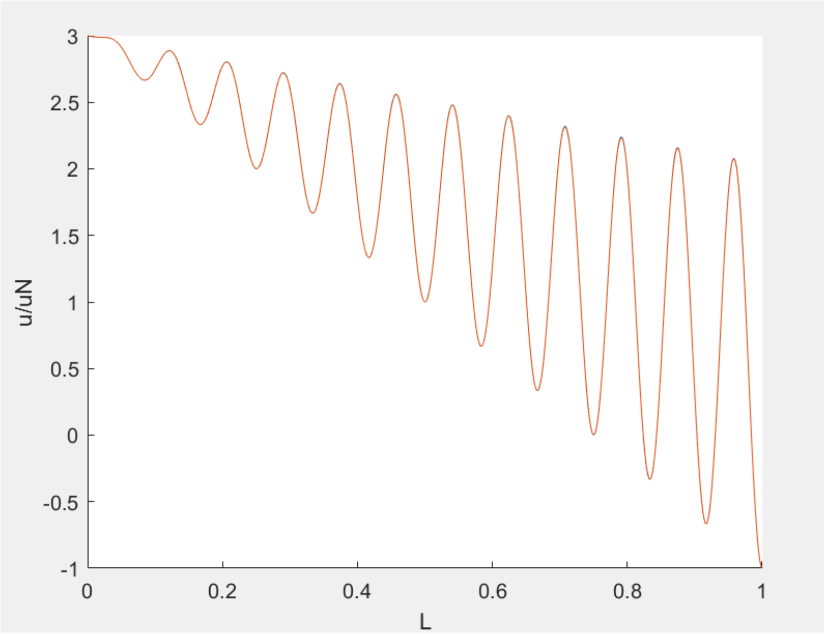
u/uN
L



When p=3, n=10, 20 ,50

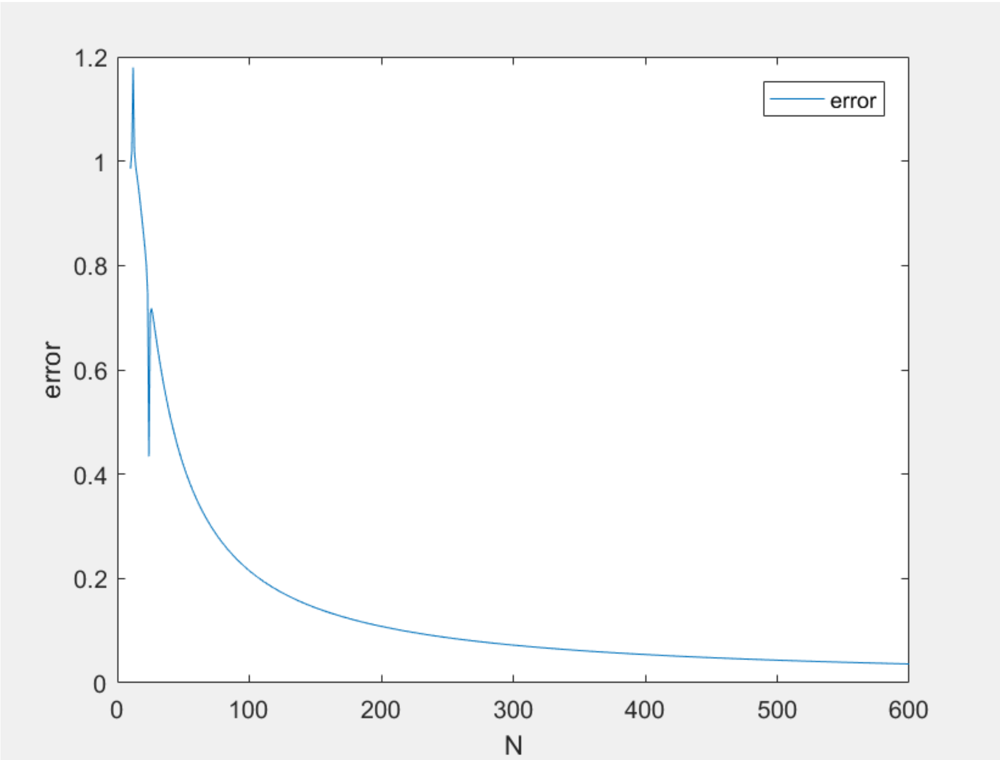




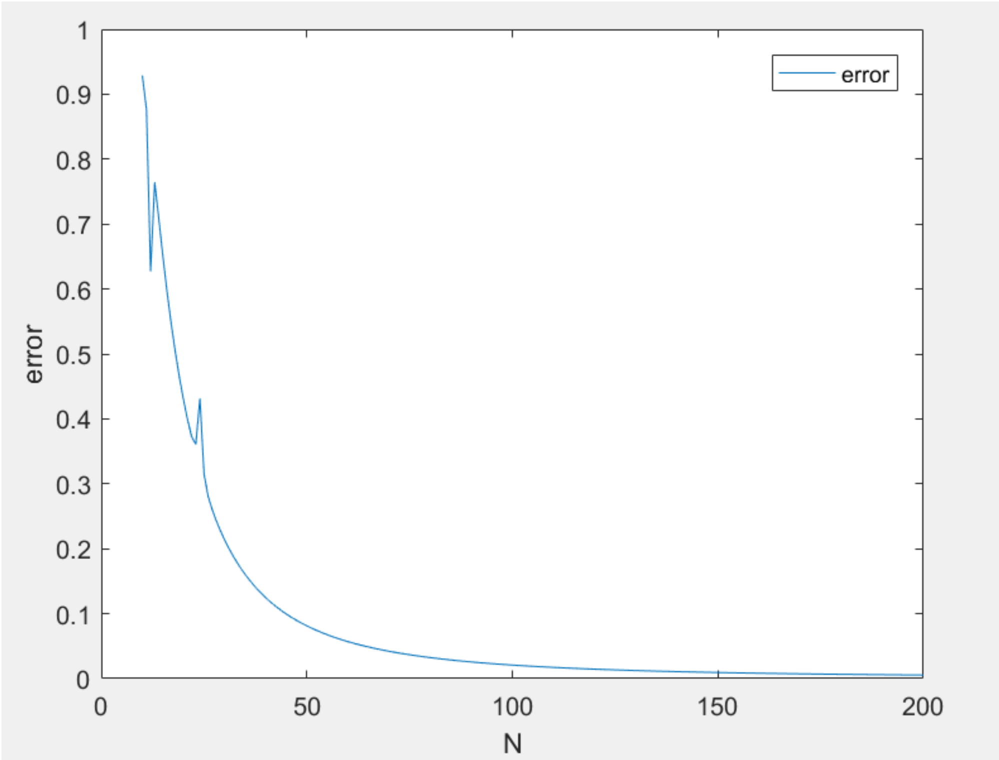


3) **p =1, 2, 3… the relationship between error and N**

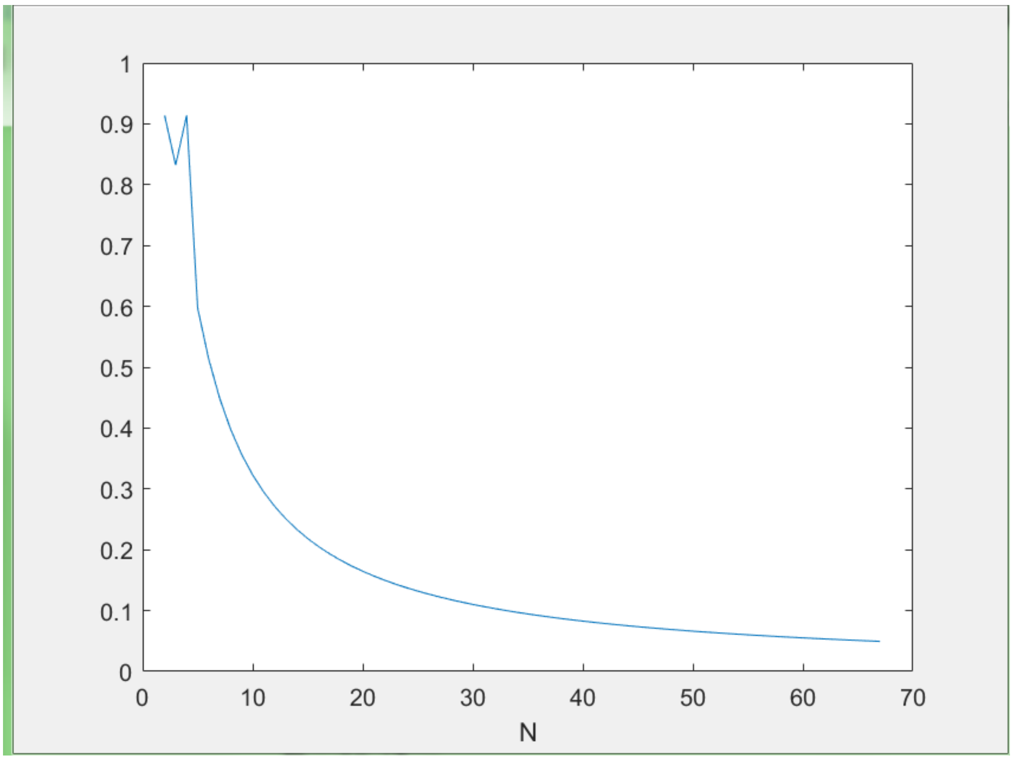
p=1



p=2

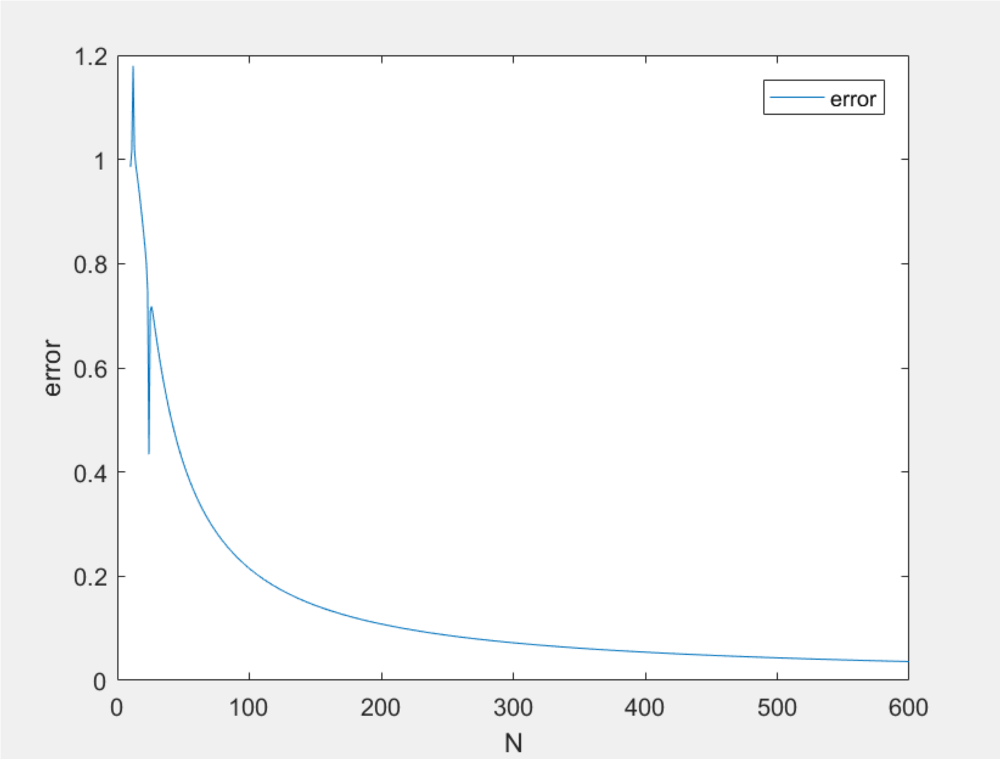


p=3



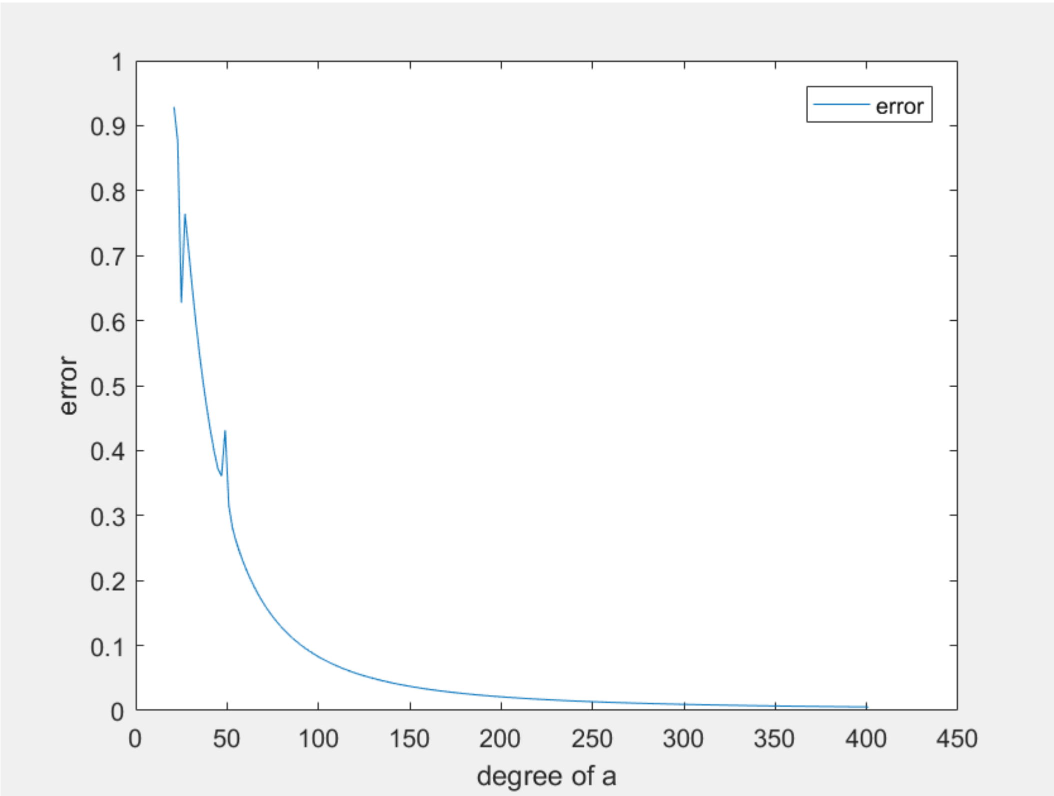
4) p=1,2,3 the relationship between error and degree

p=1,



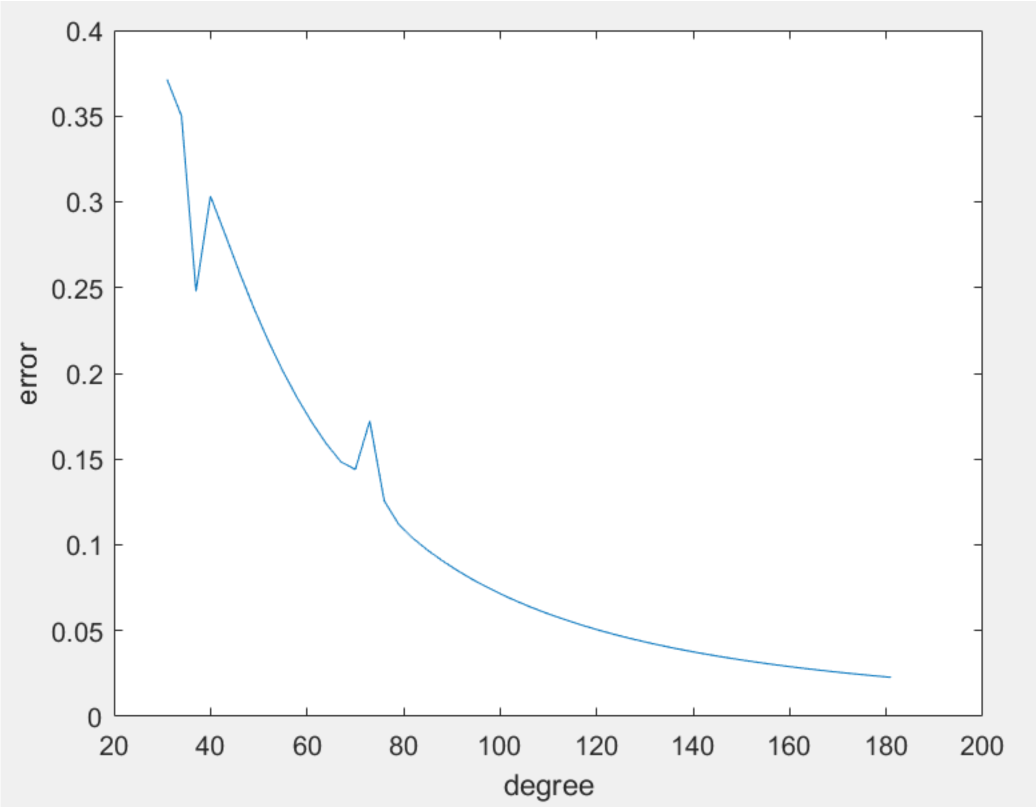
error
N
error
0
100
200
300
400
500
600
0
0.2
0.4
0.6
0.8
1
1.2

p=2,



error
degree of a
error
0
50
100
150
200
250
300
350
400
450
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

p=3,



0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
error
20
40
60
80
100
120
140
160
180
200
degree

5. Observations and discussion

According to the plots and tables,

a) I find when p increases, the Best N decreases significantly.

b) And errors drop down rapidly at the beginning. When N become larger, it is really hard to converge. When require higher accuracy, the cost will become much larger.

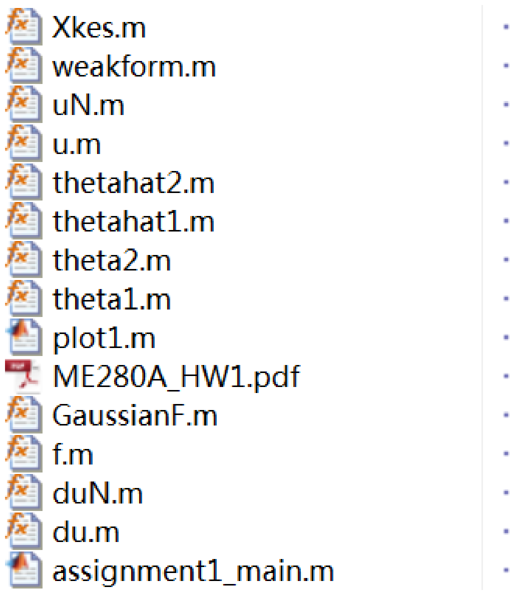
c) But for the same N, higher order require more run-time.

d) Besides, I found there is a Matlab command “sparse” that could be helpful to speed up.

e) In this case, 5 Guassian points would be enough to get accurate solution.

6. Appendix

Structure:



Weak formulation_p=1:

```
function output=weakform(N,k)
Gaussian=[
0.00,0.888;
0.774,0.555;
-0.774,0.555];
E=0.2;
L=1;
k=12;
he=L/N;
J=he/2;
Ke=zeros(2,2,N);
K=zeros(N+1,N+1);
Re=zeros(2,N);
a=zeros(N+1,1);
R=zeros(N+1,1);
%uN=zeros(N,1);
for i=1:N
    Ke(:,:,i)=[E/(J*2),-1*E/(J*2);
        -1*E/(J*2),E/(J*2)];
```

```
        Re(1,i)=J*Gaussian(1,2)*thetahat1(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
        Re(1,i)=Re(1,i)+J*Gaussian(2,2)*thetahat1(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
        Re(1,i)=Re(1,i)+J*Gaussian(3,2)*thetahat1(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
        Re(2,i)=J*Gaussian(1,2)*thetahat2(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(2,2)*thetahat2(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(3,2)*thetahat2(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
end
for i=1:N+1
        if i==1
                K(i,1)=Ke(1,1,i);
                K(i,2)=Ke(1,2,i);
                R(i)=Re(1,i);
                continue
        end
        if(i==N+1)
                K(i,N)=Ke(2,1,i-1);
                K(i,N+1)=Ke(2,2,i-1);
                R(i)=Re(2,i-1);
                continue
        else
                K(i,i-1)=Ke(2,1,i-1);
                K(i,i)=Ke(2,2,i-1)+Ke(1,1,i);
                K(i,i+1)=Ke(1,2,i);
                R(i)=Re(1,i)+Re(2,i-1);
        end
end
for i=1:N+1
end

Kc=K(2:N,2:N);
Rc=R(2:N);
Rc(N-1)=Rc(N-1)+K(N,N+1);
Rc(1)=Rc(1)-3*K(2,1);
Kc=sparse(Kc);
a=Kc\Rc;
a=[3;a;-1];
uE=@(x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))+k/(4*pi*pi*E)-4)-
(a(floor(x*N)+2)-a(floor(x*N)+1))/he)^2;
duE=@(x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))+k/(4*pi*pi*E)-4)^2);
e=(integral(uE,0,L,'ArrayValued',true))^0.5;
uu=(integral(duE,0,L,'ArrayValued',true))^0.5;
eN=e/(integral(duE,0,L,'ArrayValued',true))^0.5;
output=eN;
end
```

Weak formulation_p=2:

```
%function output=weakform(N,k)
Gaussian=[
0.000,0.568;
0.538,0.478;
0.906,0.236;
-0.538,0.478;
-0.906,0.237];
E=0.2;
N=30;
L=1;
k=12;
he=L/N;
J=he/2;
%f=@(x)-1*k^2*sin(2*pi*k*x/L);
%u=@(x)(-1*L/(4*E*pi^2))*sin(2*pi*k*x/L)+L*x;
%du=@(x)(-1*L*k/(2*E*pi))*cos(2*pi*k*x/L)+L;
%Xkes=@(x,i)J*x+(2*i-1)*L/N;
%GaussianF=@(x)x;
Ke=zeros(3,3,N);
K=zeros(2*N+1,2*N+1);
Re=zeros(3,N);
a=zeros(2*N+1,1);
R=zeros(2*N+1,1);
dthetahat1=@ (x) x-0.5;
dthetahat2=@ (x) -2*x;
dthetahat3=@ (x) x+0.5;
for i=1:N
    Ke(1,:,i)=[integral(@ (x)(x-0.5)*(x-0.5)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(x-0.5)*(-2*x)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(x-0.5)*(x+0.5)*E/J,-1,1,'ArrayValued',true)];
    Ke(2,2:3,i)=[integral(@ (x)(-2*x)*(-2*x)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(-2*x)*(x+0.5)*E/J,-1,1,'ArrayValued',true)];
    Ke(3,3,i)=integral(@ (x)(x+0.5)*(x+0.5)*E/J,-1,1,'ArrayValued',true);
    Ke(2,1,i)=Ke(1,2,i);
    Ke(3,1,i)=Ke(1,3,i);
    Ke(3,2,i)=Ke(2,3,i);
    Re(1,i)=J*Gaussian(1,2)*thetahat1(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(2,2)*thetahat1(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(3,2)*thetahat1(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(4,2)*thetahat1(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
```

```
        Re(1,i)=Re(1,i)+J*Gaussian(5,2)*thetahat1(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
        Re(2,i)=J*Gaussian(1,2)*thetahat2(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(2,2)*thetahat2(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(3,2)*thetahat2(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(4,2)*thetahat2(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
        Re(2,i)=Re(2,i)+J*Gaussian(5,2)*thetahat2(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
        Re(3,i)=J*Gaussian(1,2)*thetahat3(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
        Re(3,i)=Re(3,i)+J*Gaussian(2,2)*thetahat3(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
        Re(3,i)=Re(3,i)+J*Gaussian(3,2)*thetahat3(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
        Re(3,i)=Re(3,i)+J*Gaussian(4,2)*thetahat3(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
        Re(3,i)=Re(3,i)+J*Gaussian(5,2)*thetahat3(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
    end
    for i=2:2*N
        if i==2
            K(1,1)=Ke(1,1,1);
            K(1,2)=Ke(1,2,1);
            K(1,3)=Ke(1,3,1);
            R(1)=Re(1,1);
            K(2,1)=Ke(2,1,1);
            K(2,2)=Ke(2,2,1);
            K(2,3)=Ke(2,3,1);
            R(2)=Re(2,1);
            continue
        end
        if(i==2*N)
            K(2*N,2*N-1)=Ke(2,1,N);
            K(2*N,2*N)=Ke(2,2,N);
            K(2*N,2*N+1)=Ke(2,3,N);
            R(2*N)=Re(2,N);
            K(2*N+1,2*N-1)=Ke(3,1,N);
            K(2*N+1,2*N)=Ke(3,2,N);
            K(2*N+1,2*N+1)=Ke(3,3,N);
            R(2*N+1)=Re(3,N);
            continue
        else
            if(rem (i,2)==1)
                K(i,i-2)=Ke(3,1,floor(i/2));
                K(i,i-1)=Ke(3,2,floor(i/2));
                K(i,i)=Ke(3,3,floor(i/2))+Ke(1,1,floor(i/2)+1);
                K(i,i+1)=Ke(1,2,floor(i/2)+1);
                K(i,i+2)=Ke(1,3,floor(i/2)+1);
                R(i)=Re(3,floor(i/2))+Re(1,floor(i/2)+1);
            end
            if(rem (i,2)==0)
```

```
            K(i,i-1)=Ke(2,1,floor(i/2));
            K(i,i)=Ke(2,2,floor(i/2));
            K(i,i+1)=Ke(2,3,floor(i/2));
            R(i)=Re(2,floor(i/2));
        end
    end
end
for i=1:N+1
end

Kc=K(2:2*N,2:2*N);
Rc=R(2:2*N);
Rc(2*N-1)=Rc(2*N-1)+K(2*N,2*N+1);
Rc(2*N-2)=Rc(2*N-2)+K(2*N-1,2*N+1);
Rc(1)=Rc(1)-3*K(2,1);
Rc(2)=Rc(2)-3*K(3,1);
Kc=sparse(Kc);
a=Kc\Rc;
a=[3;a;-1];
x=0:0.002:1;
y1=[];
y2=[];
uE=@ (x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))+k/(4*pi*pi*E)-4-
((a(2*(floor(x*N)+1)+1)*((2*x-(2*(floor(x*N)+1)-1)*he)/he+0.5)+a(2*(floor(x*N)+1))*(-2*((2*x-
(2*(floor(x*N)+1)-1)*he)/he))+a(2*(floor(x*N)+1)-1)*((2*x-(2*(floor(x*N)+1)-1)*he)/he-0.5))/J))^2);
duE=@ (x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))-k/(4*pi*pi*E)-4)^2);
e=(integral(uE,0,L,'ArrayValued',true))^0.5;
uu=(integral(duE,0,L,'ArrayValued',true))^0.5;
eN=e/uu;
output=eN ;
%end
```

Weak formulation_p=3:

```
%function output=weakform(N,k)
Gaussian=[
0.000,0.568;
0.538,0.478;
0.906,0.236;
-0.538,0.478;
-0.906,0.237];
E=0.2;
N=20;
L=1;
```

```
k=12;
he=L/N;
J=he/2;
Ke=zeros(4,4,N);
K=zeros(3*N+1,3*N+1);
Re=zeros(4,N);
a=zeros(3*N+1,1);
R=zeros(3*N+1,1);
dthetahat1=@ (x) (-9/16)*(3*x*x-2*x-1/9);
dthetahat2=@ (x) (9/16)*(3*x*x-x*2/3-1);
dthetahat3=@ (x) (-9/16)*(3*x*x+x*2/3-1);
dthetahat4=@ (x) (9/16)*(3*x*x+2*x-1/9);
for i=1:N
    Ke(1,:,i)=[integral(@ (x)(-9/16)*(3*x*x-2*x-1/9)*(-9/16)*(3*x*x-2*x-1/9)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(-9/16)*(3*x*x-2*x-1/9)*(9/16)*(3*x*x-x*2/3-1)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(-9/16)*(3*x*x-2*x-1/9)*(-9/16)*(3*x*x+x*2/3-1)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(-9/16)*(3*x*x-2*x-1/9)*(9/16)*(3*x*x+2*x-1/9)*E/J,-1,1,'ArrayValued',true)];
    Ke(2,2:4,i)=[integral(@ (x)(9/16)*(3*x*x-x*2/3-1)*(9/16)*(3*x*x-x*2/3-1)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(9/16)*(3*x*x-x*2/3-1)*(-9/16)*(3*x*x+x*2/3-1)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(9/16)*(3*x*x-x*2/3-1)*(9/16)*(3*x*x+2*x-1/9)*E/J,-1,1,'ArrayValued',true)];
    Ke(3,3:4,i)=[integral(@ (x)(-9/16)*(3*x*x+x*2/3-1)*(-9/16)*(3*x*x+x*2/3-1)*E/J,-1,1,'ArrayValued',true),
        integral(@ (x)(-9/16)*(3*x*x+x*2/3-1)*(9/16)*(3*x*x+2*x-1/9)*E/J,-1,1,'ArrayValued',true)];
    Ke(4,4,i)=integral(@ (x)(9/16)*(3*x*x+2*x-1/9)*(9/16)*(3*x*x+2*x-1/9)*E/J,-1,1,'ArrayValued',true);
    Ke(2,1,i)=Ke(1,2,i);
    Ke(3,1,i)=Ke(1,3,i);
    Ke(4,1,i)=Ke(1,4,i);
    Ke(3,2,i)=Ke(2,3,i);
    Ke(4,2,i)=Ke(2,4,i);
    Ke(4,3,i)=Ke(3,4,i);
    Re(1,i)=J*Gaussian(1,2)*thetahat1(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(2,2)*thetahat1(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(3,2)*thetahat1(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(4,2)*thetahat1(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
    Re(1,i)=Re(1,i)+J*Gaussian(5,2)*thetahat1(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
    Re(2,i)=J*Gaussian(1,2)*thetahat2(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
    Re(2,i)=Re(2,i)+J*Gaussian(2,2)*thetahat2(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
    Re(2,i)=Re(2,i)+J*Gaussian(3,2)*thetahat2(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
    Re(2,i)=Re(2,i)+J*Gaussian(4,2)*thetahat2(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
    Re(2,i)=Re(2,i)+J*Gaussian(5,2)*thetahat2(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
    Re(3,i)=J*Gaussian(1,2)*thetahat3(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
    Re(3,i)=Re(3,i)+J*Gaussian(2,2)*thetahat3(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
    Re(3,i)=Re(3,i)+J*Gaussian(3,2)*thetahat3(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
    Re(3,i)=Re(3,i)+J*Gaussian(4,2)*thetahat3(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
    Re(3,i)=Re(3,i)+J*Gaussian(5,2)*thetahat3(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
```

```
        Re(4,i)=J*Gaussian(1,2)*thetahat4(Gaussian(1,1))*f(Xkes(Gaussian(1,1),i,J,L,N),k,L);
        Re(4,i)=Re(4,i)+J*Gaussian(2,2)*thetahat4(Gaussian(2,1))*f(Xkes(Gaussian(2,1),i,J,L,N),k,L);
        Re(4,i)=Re(4,i)+J*Gaussian(3,2)*thetahat4(Gaussian(3,1))*f(Xkes(Gaussian(3,1),i,J,L,N),k,L);
        Re(4,i)=Re(4,i)+J*Gaussian(4,2)*thetahat4(Gaussian(4,1))*f(Xkes(Gaussian(4,1),i,J,L,N),k,L);
        Re(4,i)=Re(4,i)+J*Gaussian(5,2)*thetahat4(Gaussian(5,1))*f(Xkes(Gaussian(5,1),i,J,L,N),k,L);
    end
    for i=3:3*N-1
        if i==3
            K(1,1)=Ke(1,1,1);
            K(1,2)=Ke(1,2,1);
            K(1,3)=Ke(1,3,1);
            K(1,4)=Ke(1,4,1);
            R(1)=Re(1,1);
            K(2,1)=Ke(2,1,1);
            K(2,2)=Ke(2,2,1);
            K(2,3)=Ke(2,3,1);
            K(2,4)=Ke(2,4,1);
            R(2)=Re(2,1);
            K(3,1)=Ke(3,1,1);
            K(3,2)=Ke(3,2,1);
            K(3,3)=Ke(3,3,1);
            K(3,4)=Ke(3,4,1);
            R(3)=Re(3,1);
            continue
        end
        if(i==3*N-1)
            K(3*N-1,3*N-2)=Ke(2,1,N);
            K(3*N-1,3*N-1)=Ke(2,2,N);
            K(3*N-1,3*N)=Ke(2,3,N);
            K(3*N-1,3*N+1)=Ke(2,4,N);
            R(3*N-1)=Re(2,N);
            K(3*N,3*N-2)=Ke(3,1,N);
            K(3*N,3*N-1)=Ke(3,2,N);
            K(3*N,3*N)=Ke(3,3,N);
            K(3*N,3*N+1)=Ke(3,4,N);
            R(3*N)=Re(3,N);
            K(3*N+1,3*N-2)=Ke(4,1,N);
            K(3*N+1,3*N-1)=Ke(4,2,N);
            K(3*N+1,3*N)=Ke(4,3,N);
            K(3*N+1,3*N+1)=Ke(4,4,N);
            R(3*N+1)=Re(4,N);
            continue
        else
            if(rem (i,3)==1)
```

```
            K(i,i-3)=Ke(4,1,floor(i/3));
            K(i,i-2)=Ke(4,2,floor(i/3));
            K(i,i-1)=Ke(4,3,floor(i/3));
            K(i,i)=Ke(4,4,floor(i/3))+Ke(1,1,floor(i/3)+1);
            K(i,i+1)=Ke(1,2,floor(i/3)+1);
            K(i,i+2)=Ke(1,3,floor(i/3)+1);
            K(i,i+3)=Ke(1,4,floor(i/3)+1);
            R(i)=Re(4,floor(i/3))+Re(1,floor(i/3)+1);
        end
        if(rem (i,3)==2)
            K(i,i-1)=Ke(2,1,floor(i/3));
            K(i,i)=Ke(2,2,floor(i/3));
            K(i,i+1)=Ke(2,3,floor(i/3));
            K(i,i+2)=Ke(2,4,floor(i/3));
            R(i)=Re(2,floor(i/3));
        end
        if(rem (i,3)==0)
            K(i,i-2)=Ke(3,1,floor(i/3));
            K(i,i-1)=Ke(3,2,floor(i/3));
            K(i,i)=Ke(3,3,floor(i/3));
            K(i,i+1)=Ke(3,4,floor(i/3));
            R(i)=Re(3,floor(i/3));
        end
    end
end
for i=1:N+1
end

Kc=K(2:3*N,2:3*N);
Rc=R(2:3*N);
Rc(3*N-1)=Rc(3*N-1)+K(3*N,3*N+1);
Rc(3*N-2)=Rc(3*N-2)+K(3*N-1,3*N+1);
Rc(3*N-3)=Rc(3*N-3)+K(3*N-2,3*N+1);
Rc(1)=Rc(1)-3*K(2,1);
Rc(2)=Rc(2)-3*K(3,1);
Rc(3)=Rc(3)-3*K(4,1);
Kc=sparse(Kc);
a=Kc\Rc;
uE=@ (x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))+k/(4*pi*pi*E)-4-
((a(2*(floor(x*N)+1)+1)*((2*x-(2*(floor(x*N)+1)-1)*he)/he+0.5)+a(2*(floor(x*N)+1))*(-2*((2*x-
(2*(floor(x*N)+1)-1)*he)/he))+a(2*(floor(x*N)+1)-1)*((2*x-(2*(floor(x*N)+1)-1)*he)/he-0.5))/J))^2);
duE=@ (x)E*(((k*L/(4*pi*pi*E))*(2*pi*k*x*sin(2*pi*k*x/L)+L*cos(2*pi*k*x/L))-k/(4*pi*pi*E)-4)^2);
e=(integral(uE,0,L,'ArrayValued',true))^0.5;
uu=(integral(duE,0,L,'ArrayValued',true))^0.5;
```

```
eN=e/uu;
output=eN ;
%end
```