CAAM 336 · DIFFERENTIAL EQUATIONS

Homework 9 · Solutions

Posted Wednesday 29 October, 2014. Due 5pm Wednesday 5 November, 2014.

Please write your name and ***residential college*** *on your homework.*

1. [30 points: 10 points each]
   Consider the following BVP with inhomogeneous boundary conditions conditions:

$$
\begin{aligned}
-((1+x^2)u')' &= x,\ 0 < x < 1,\\
u(0) &= 1,\\
u(1) &= 2.
\end{aligned}
$$

(a) Let $x_0 = 0, x_1, \ldots, x_N, x_{N+1} = 1$ be a grid of points where $x_i = ih$. Compute the finite element solution of this BVP using piecewise linear basis functions

$$
\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h} & \text{if } x \in [x_{i-1}, x_i);\\ \dfrac{x_{i+1} - x}{h} & \text{if } x \in [x_i, x_{i+1});\\ 0 & \text{otherwise;} \end{cases}
$$

Plot the Galerkin solutions with $N = 4, 8, 16, 32$ superimposed on each other. *You may wish to start with the codes from HW 8.*

(b) In general, inhomogeneous boundary conditions are treated by decomposing $u(x)$ into

$$
u(x) = w(x) + g(x)
$$

where $w(0) = w(1) = 0$ and $g(x)$ is any function satisfying inhomogeneous boundary conditions (this is referred to as the *lift*). We should make sure that the finite element solution does not depend on what lift we choose.

Let $g(x) = 1 + x$; compute what modifications must be made to the load vector in order to compute the solution in this case.

(c) Using the above modifications for $g(x) = 1 + x$, plot in MATLAB the solution $u_N(x)$ for $N = 4, 8, 16, 32$. Verify that these solutions should look identical to the solutions from (a).

Solution.

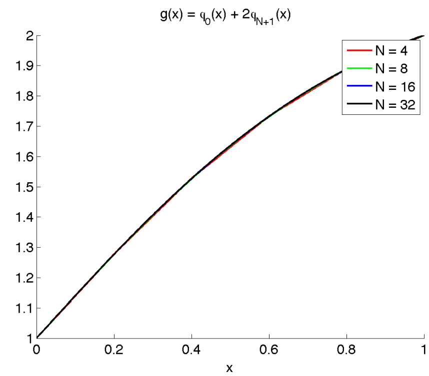
(a) In class, we showed that the resulting finite element system $K\alpha = b$ satisfied

$$
K_{ij} = a(\phi_j, \phi_i), \quad b_i = \begin{cases} (f, \phi_1) - u(0)a(\phi_0, \phi_1) & i = 1\\ (f, \phi_i) & 1 < i < N\\ (f, \phi_N) - u(1)a(\phi_{N+1}, \phi_N) & i = N. \end{cases}
$$

In Homework 8, we computed K_{ij}, so here we will focus only on computing b_i.

Since $a(u, v) = \int_0^1 (1 + x^2)\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}$, and ϕ_0 is piecewise linear and nonzero only on the interval $[x_0, x_1] = [0, h]$,

$$
a(\phi_0, \phi_1) = \int_0^h (1 + x^2)\frac{-1}{h^2} = \frac{1}{h^2}\left[x + \frac{x^3}{3}\right]_0^h = \frac{1}{h} + \frac{h}{3}.
$$



Similarly, ϕ_{N+1} is nonzero only on $[x_N, x_{N+1}] = [1-h, h]$, which gives

$$a(\phi_{N+1}, \phi_N) = \int_{1-h}^{1} (1+x^2)\frac{-1}{h^2} = \frac{1}{h^2}\left[x + \frac{x^3}{3}\right]_{1-h}^{1} = \frac{2h}{3} + \frac{4}{h} - 2.$$

Then, since $a(\phi_0, \phi_1)$ and $a(\phi_{N+1}, \phi_N)$ are now given, we can compute b_i using the above formula. The graph produced by the finite element solution for $N = 4, 8, 16, 32$ is shown below

(b) By changing $g(x)$ to $1 + x$ and having $u(x) = w(x) + g(x)$, the more general form of the finite element equation needs to hold:

$$a(w, \phi_i) = (f, \phi_i) - a(g, \phi_i), \quad i = 1, \ldots, N.$$

Then, since $\frac{\partial g}{\partial x} = 1$, we can write

$$a(g, \phi_i) = \int_0^1 (1+x^2)\frac{\partial \phi_i}{\partial x} = \frac{1}{h}\left(\int_{x_{i-1}}^{x_i} (1+x^2) - \int_{x_i}^{x_{i+1}} (1+x^2)\right)$$

Plugging in $x_i = ih$ reduces the above to $a(g, \phi_i) = -2ih^2$.
Another alternative way to solve this problem is to notice that you can represent $g(x)$ exactly using a linear combination of $\phi_0, \ldots, \phi_{N+1}$

$$g(x) = \sum_{j=0}^{N+1} g(x_j)\phi_j(x).$$

You can then use the fact that

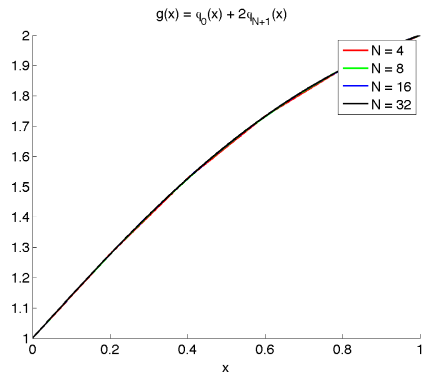
$$\begin{aligned} a(g, \phi_i) &= \sum_{j=1}^{N} g(x_j)a(\phi_j, \phi_i) + g(0)a(\phi_0, \phi_i) + g(1)a(\phi_{N+1}, \phi_N) \\ &= \sum_{j=1}^{N} g(x_j)K_{ij} + a(\phi_0, \phi_i) + 2a(\phi_{N+1}, \phi_N), \end{aligned}$$

which reduces down to using K_{ij} (from Hw 8) and the solution from part (a) for $a(\phi_0, \phi_i)$ and $a(\phi_{N+1}, \phi_i)$, which are zero unless $i = 1$ or $i = N$.

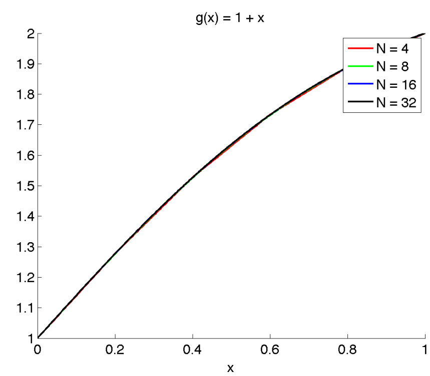
(c) The figure produced using $g(x) = 1 + x$ is as follow:

The code to compute both part (a) and (c) is below as well.

```
% demo of the finite element method for the problem
%   -d/dx((1+x^2) du/dx)  = x, 0 < x < 1, u(0) = u(1) = 0.
```



(a) $g(x) = \phi_0(x) + 2\phi_{N+1}(x)$



(b) $g(x) = 1 + x$

```
Nvec = [4 8 16 32];   % vector of N values we shall use

color ='rgbk';

% each pass of the following loop handles a new N value...
for j=1:length(Nvec)
    N = Nvec(j);
    h = 1/(N+1);
    x = [1:N]*h;

    % construct the stiffness matrix (integrals done by hand)
    maindiag = 2/h + 2*h/3 + 2*h*([1:N].^2);
    offdiag  = -1/h - h*(([1:N-1].^2) + [1:N-1] + 1/3);
    K = diag(maindiag) + diag(offdiag,1) + diag(offdiag,-1);

    % construct the load vector (integrals done by hand)
    f = h^2*[1:N]';
    f(1)=f(1) + 1/h + (h/3);
    f(N)=f(N) + 2*h/3 + 4/h - 2;

    % solve for expansion coefficients of Galerkin approximation
    c = K\f;

    % plot the true solution
    xx = linspace(0,1,1000)';     % finely spaced points between 0 and 1.

    % plot the approximation solution
    uN = zeros(size(xx));
    for k=1:N
        uN = uN + c(k)*hat(xx,k,N);
    end
    uN = uN + (xx < h).*(h-xx)/h;
    uN = uN + 2*(xx > (1-h)).*(xx-(1-h))/h;

    figure(1);hold on
    plot(xx, uN, color(j),'linewidth',2)
    set(gca,'fontsize',16)
    xlabel('x')
    axis([0 1 1 2])

    % ==================================================================

    % use now g(x) = 1+x, recompute f and add a(g,\phi_i) terms
    f = h^2*[1:N]';
    f = f + 2*[1:N]'*h^2;

    % solve for expansion coefficients of Galerkin approximation
```

```
    c = K\f;

    % plot the approximation solution
    uN = zeros(size(xx));
    for k=1:N
        uN = uN + c(k)*hat(xx,k,N);
    end
    g = @(x) 1+x;
    uN = uN + g(xx);

    figure(2);hold on
    plot(xx, uN, color(j),'linewidth',2)
    set(gca,'fontsize',16)
    xlabel('x')
    tag{j} = sprintf('N = %d', N);
    axis([0 1 1 2])
end
figure(1)
legend(tag)
title('g(x) = \phi_0(x) + 2\phi_{N+1}(x)')
print('-dpng',gcf,'p1a')
figure(2)
title('g(x) = 1 + x')
legend(tag)
print('-dpng',gcf,'p1c')
```

---

2. [40 points: 10 points each]
Consider the linear differential equation,

$$\begin{aligned} -u'' + u &= f \qquad 0 < x < 1 \\ u(0) &= 0 \\ u'(1) &= 0 \end{aligned}$$

Let $f(x) = x(1-x)$. Suppose that N is a positive integer and define $h = \dfrac{1}{N+1}$ and $x_i = ih$ for $i = 0, 1 \ldots, N+1$. Consider the hat functions $\phi_i \in C[0,1]$, defined as

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h} & \text{if } x \in [x_{i-1}, x_i); \\ \dfrac{x_{i+1} - x}{h} & \text{if } x \in [x_i, x_{i+1}); \\ 0 & \text{otherwise;} \end{cases}$$

for $i = 1, \ldots, N+1$.
Let the stiffness matrix K be defined as

$$K_{ij} = \int_0^1 \phi_j'(x)\phi_i'(x)dx$$

Likewise, let the mass matrix M be defined as

$$M_{ij} = \int_0^1 \phi_j(x)\phi_i(x)dx$$

(a) Show that the finite element matrix A for the weak form of the equation

$$-u'' + u = f$$

can be defined as $A = K + M$. Specify the entries M_{ij} and K_{ij} (you may use the results of previous homework).

(b) Show that A is positive definite. *Hint: Use the weak form.*

(c) Write MATLAB code to solve the finite element system

$$A\alpha = b$$

for the approximate solution $u_N(x) = \sum_{j=1}^{N} \alpha_j \phi_j(x)$ of the differential equation using the finite element method. Produce a plot that compares the approximate solution u_N for $N = 4$ and $N = 8$ with the true solution

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

Hint: If you'd like, you can use the Matlab code called posted on the course webpage. You may also use the Matlab function quad for numerical integration.

(d) Describe what modifications to the load vector b are necessary to compute the solution to the problem with inhomogeneous Neumann boundary condition

$$\begin{aligned} -u'' + u &= f \qquad 0 < x < 1 \\ u(0) &= 0 \\ u'(1) &= 1. \end{aligned}$$

Modify your Matlab code to accommodate these changes, and produce a plot of the solution for $N = 4, 8, 16$. *You do not need to compare against the exact solution for this problem.*

Solution.

(a) Let $V = \{v \in C^2[0,1] : v(0) = 0\}$ For $\forall v \in V$ Multiply both side of BVP by the test function v

$$(-u'' + u)v = fv$$

Now integrate both sides over 0 to 1

$$\int_0^1 (-u'' + u)v dx = \int_0^1 fv dx$$

Apply integration by parts to second derivative

$$\begin{aligned}
\int_0^1 (-u'')v dx + \int_0^1 uv dx &= [-u'v]_0^1 + \int_0^1 u'v' dx + \int_0^1 uv dx \\
&= (-u'(1)v(1) + u'(0)v(0)) + \int_0^1 u'v' dx + \int_0^1 uv dx \\
&= \int_0^1 u'v' dx + \int_0^1 uv dx \text{ (by the mixed boundary condition and } v(0) = 0) \\
&= a(u,v) + (u,v)
\end{aligned}$$

Thus the variational form is

$$a(u,v) + (u,v) = (f,v) \qquad \forall v \in V$$

Now define a finite element space $V_n = \{\phi_1, \phi_2, \cdots \phi_{n+1}\}$ and define the finite element solution $u_n = \sum_{j=1}^{n+1} c_j\phi_j$ then we get

$$a(\sum_{j=1}^{n+1} c_j\phi_j, \phi_i) + (\sum_{j=1}^{n+1} c_j\phi_j, \phi_i) = (f, \phi_i) \qquad i = 1, \cdots n+1$$

Above system can be written also as follows

$$\sum_{j=1}^{n+1} \left(\underbrace{a(\phi_j, \phi_i)}_{K_{ij}} + \underbrace{(\phi_j, \phi_i)}_{M_{ij}} \right) c_j = (f, \phi_i) \qquad i = 1, \cdots n+1.$$

This is a linear system of equation $Ac = f$ where $A = K + M$ and A is $(N+1) \times (N+1)$ matrix and f is $(N+1) \times 1$ vector.

Now let us specify entries of K and M. Here we should note that the hat function ϕ_i defined in the problem for $i = 1, 2 \cdots, n$. However ϕ_{n+1} is only half of the other hat functions. Then

$$\phi_{n+1}(x) = \begin{cases} \dfrac{x - x_n}{h} & \text{if } x \in [1-h, 1); \\ 0 & \text{otherwise;} \end{cases}$$

By the definition of ϕ_i we have

$$\phi_i'(x) = \begin{cases} \dfrac{1}{h} & \text{if } x \in [x_{i-1}, x_i); \\ \dfrac{-1}{h} & \text{if } x \in [x_i, x_{i+1}); \\ 0 & \text{otherwise;} \end{cases}$$

and

$$\phi'_{n+1}(x) = \begin{cases} \dfrac{1}{h} & \text{if } x \in [x_{i-1}, x_i); \\ 0 & \text{otherwise;} \end{cases}$$

When the hat functions ϕ_i and ϕ_j are not neighbors, i.e., |i-j|>1,

$$\int_0^1 \phi_i \phi_j = 0 \text{ and } \int_0^1 \phi'_i \phi'_j = 0$$

Thus we only need to consider the following cases:

$$\begin{aligned} K_{ii} = a(\phi_i, \phi_i) &= \int_0^1 \phi'_i \phi'_i \\ &= \int_{x_{i-1}}^{x_i} \phi'_i \phi'_i + \int_{x_i}^{x_{i+1}} \phi'_i \phi'_i \\ &= \int_{x_{i-1}}^{x_i} (\frac{1}{h})(\frac{1}{h}) + \int_{x_i}^{x_{i+1}} (-\frac{1}{h})(-\frac{1}{h}) \\ &= \frac{2}{h} \end{aligned}$$

Since $[x_i, x_{i+1}]$ is the only interval where ϕ_i and ϕ_{i+1} and their derivative are not 0 we have

$$\begin{aligned} K_{i(i+1)} = a(\phi_i, \phi_{i+1}) &= \int_0^1 \phi'_i \phi'_{i+1} \\ &= \int_{x_i}^{x_{i+1}} \phi'_i \phi'_{i+1} \\ &= \int_{x_i}^{x_{i+1}} (-\frac{1}{h})(\frac{1}{h}) \\ &= -\frac{1}{h} \end{aligned}$$

By symmetry $K_{i(i+1)} = K_{(i+1)i}$.

$$\begin{aligned} K_{(n+1)(n+1)} = a(\phi_{n+1}, \phi_{n+1}) &= \int_0^1 \phi'_{n+1} \phi'_{n+1} \\ &= \int_{1-h}^1 \phi'_{n+1} \phi'_{n+1} \\ &= \int_{1-h}^1 (\frac{1}{h})(\frac{1}{h}) \\ &= \frac{1}{h} \end{aligned}$$

Therefore

$$K = \frac{1}{h} \begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & \cdots & & -1 & 2 & -1 \\ 0 & & \cdots & & -1 & 1 \end{bmatrix}$$

Similar way the matrix M, can be defined and using the result of previous homework we can find the integrals.

$$\begin{aligned}
M_{ii} = (\phi_i, \phi_i) &= \int_0^1 \phi_i \phi_i \\
&= \int_{x_{i-1}}^{x_i} \phi_i \phi_i + \int_{x_i}^{x_{i+1}} \phi_i \phi_i \\
&= \int_{x_{i-1}}^{x_i} (\frac{x - x_{i-1}}{h})(\frac{x - x_{i-1}}{h}) + \int_{x_i}^{x_{i+1}} (\frac{x_{i+1} - x}{h})(\frac{x_{i+1} - x}{h}) \\
&= \frac{2h}{3} \qquad \text{by the previous homework}
\end{aligned}$$

Since $[x_i, x_{i+1}]$ is the only interval where ϕ_i and ϕ_{i+1}

$$\begin{aligned}
M_{i(i+1)} = (\phi_i, \phi_{i+1}) &= \int_0^1 \phi_i \phi_{i+1} \\
&= \int_{x_i}^{x_{i+1}} \phi_i \phi_{i+1} \\
&= \int_{x_i}^{x_{i+1}} (\frac{x_{i+1} - x}{h})(\frac{x - x_i}{h}) \\
&= \frac{h}{6} \qquad \text{by the previous homework}
\end{aligned}$$

By symmetry $M_{i(i+1)} = M_{(i+1)i}$.

$$\begin{aligned}
M_{(n+1)(n+1)} = (\phi_{n+1}, \phi_{n+1}) &= \int_0^1 \phi_{n+1} \phi_{n+1} \\
&= \int_{1-h}^1 \phi_{n+1} \phi_{n+1} \\
&= \int_{1-h}^1 \frac{x - x_n}{h} \frac{x - x_n}{h} \\
&= \frac{h}{3}
\end{aligned}$$

Therefore

$$M = \begin{bmatrix}
\frac{2h}{3} & \frac{h}{6} & 0 & 0 & \cdots & 0 \\
\frac{h}{6} & \frac{2h}{3} & \frac{h}{6} & 0 & \cdots & 0 \\
0 & \frac{h}{6} & \frac{2h}{3} & \frac{h}{6} & \cdots & 0 \\
\vdots & & & \ddots & & \vdots \\
0 & \cdots & & \frac{h}{6} & \frac{2h}{3} & \frac{h}{6} \\
0 & & \cdots & & \frac{h}{6} & \frac{h}{3}
\end{bmatrix}$$

(b) We would like to show A is positive definite by showing $x^T A x > 0$. It is obvious $A^T = A$. From part (a) we have

$$\sum_{j=1}^{n+1} \left(a(\phi_j, \phi_i) + (\phi_j, \phi_i)\right) c_j = (f, \phi_i) \qquad \text{for} i = 1, \cdots n+1$$

Then we can say that

$$A_{ij} = a(\phi_j, \phi_i) + (\phi_j, \phi_i)$$

Let $0 \neq x \in R^{n+1}$

$$
\begin{aligned}
x^T A x &= \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} x_i A_{ij} x_j \\
&= \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} x_i (a(\phi_i, \phi_j) + (\phi_i, \phi_j)) x_j \\
&= a(\sum_{i=1}^{n+1} x_i \phi_i, \sum_{j=1}^{n+1} x_j \phi_j) + (\sum_{i=1}^{n+1} x_i \phi_i, \sum_{j=1}^{n+1} x_j \phi_j)
\end{aligned}
$$

Since $x \neq 0$ and ϕ_i is linearly independent

$$\sum_{i=1}^{n+1} x_i \phi_i \neq 0$$

Thus

$$a(\sum_{i=1}^{n+1} x_i \phi_i, \sum_{j=1}^{n+1} x_j \phi_j) > 0 \text{ and } (\sum_{i=1}^{n+1} x_i \phi_i, \sum_{j=1}^{n+1} x_j \phi_j) > 0$$

Therefore $x^T A x > 0$ for any $x \neq 0$. Thus A is positive definite.

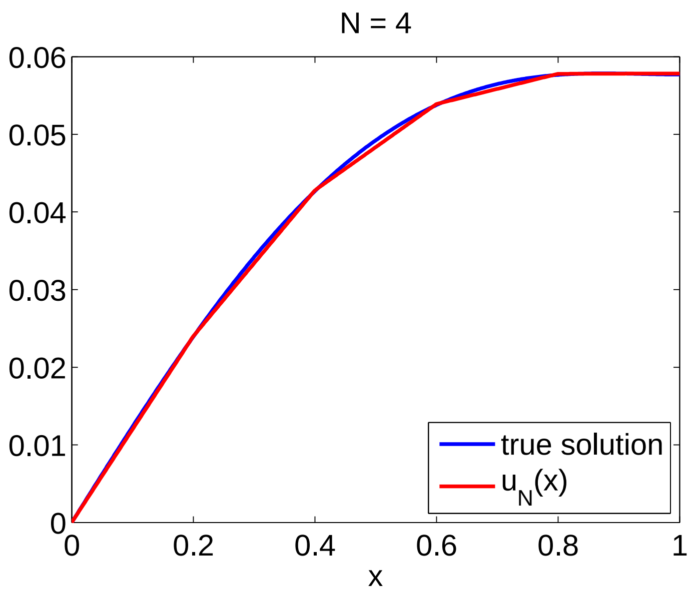
(c) To solve the system first we need to find (f, ϕ_i)

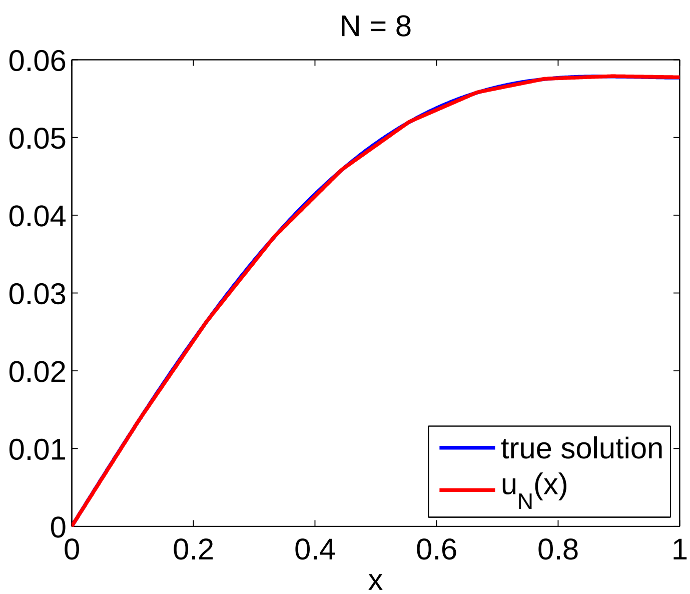
$$
\begin{aligned}
(f, \phi_i) = \int_0^1 f \phi_i = \int_{x_{i-1}}^{x_{i+1}} f \phi_i &= \int_{x_{i-1}}^{x_i} f \phi_i + \int_{x_i}^{x_{i+1}} f \phi_i \\
&= \int_{x_{i-1}}^{x_i} x(1-x) \frac{x - x_{i-1}}{h} \int_{x_i}^{x_{i+1}} x(1-x) \frac{x_{i+1} - x}{h} \\
&= \frac{-h^2}{12} (2h - 12i + 12hi^2)
\end{aligned}
$$

for $i = 1, \cdots, n$. Now, for $i = n+1$ we have

$$
\begin{aligned}
(f, \phi_{n+1}) = \int_0^1 f \phi_{n+1} &= \int_{1-h}^1 f \phi_{n+1} \\
&= \int_{1-h}^1 x(1-x) \frac{x - x_n}{h} \\
&= \frac{-h^2}{12} (h - 2)
\end{aligned}
$$

The requested plot is below.





The above plot was produced using the following MATLAB code.

```
% demo of the finite element method for the problem
%   -u'' + u = x(1-x), 0 < x < 1, u(0) = 0, u'(1) = 0


Nvec = [4 8];    % vector of N values we shall use
maxerr = zeros(size(Nvec));           % vector to hold the max errors for each N

% each pass of the following loop handles a new N value...
for j=1:length(Nvec)
   N = Nvec(j);
   h = 1/(N+1);
   x = [1:N]*h;

% construct the stiffness matrix (integrals done by hand)
   K = (2/h)*eye(N+1) - (1/h)*diag(ones(N,1),1) - (1/h)*diag(ones(N,1),-1);
   K(N+1,N+1) = 1/h;
```

```
   K;

   % construct the mass matrix (integrals done by hand)
   M = (2*h/3)*eye(N+1) + (h/6)*diag(ones(N,1),1) + (h/6)*diag(ones(N,1),-1);
   M(N+1,N+1) = h/3;
   M;
   % construct the matrix A
   A=K+M;

% Compute vector f containtin (f,phi_j), where f(x) = x(1-x)

 f = zeros(N+1,1);
 i = [1:N]';

  f = [-((h^2)*(2*h - 12*i + 12*h*(i.^2)))/12;
    -((h^2)*(h - 2))/12];

   c = A\f;
   c;

% plot the true solution
   xx = linspace(0,1,500)';     % finely spaced points between 0 and 1.
   %u = (1-xx.^2)/2;      % true solution
   u = -xx.^2+ xx-2 + (exp(1)*(2*exp(1)-1)*exp(-xx))/(1+exp(2)) +((exp(1)+2)*exp(xx))
       /(1+exp(2));      % true solution

   figure(1), clf

   plot(xx, u, 'b-','linewidth',2)
   hold on

% plot the approximation solution
   uN = zeros(size(xx));
   for k=0:N
      uN = uN + c(k+1)*hat(xx,k+1,N);
   end
   plot(xx, uN, 'r-','linewidth',2)
   set(gca,'fontsize',16)
   xlabel('x')
   legend('true solution', 'u_N(x)',4)
   title(sprintf('N = %d', N))

% plot the error in the solution for this N
   figure(2)
   plot(xx, u-uN, 'r-','linewidth',2)
   set(gca,'fontsize',16)
   xlabel('x')
   ylabel(' u(x) - u_N(x)')

% approximate the maximum error for this value of N
   maxerr(j) = max(abs(u - uN));

   input('hit return to continue')
end

% plot the maximum error
  figure(3), clf
  loglog(Nvec, maxerr, 'r.-', 'linewidth',2, 'markersize',20)
  hold on
  loglog(Nvec, Nvec.^(-2), 'b--', 'linewidth',2)
  legend('computed error', 'O(h^2) error')
  set(gca,'fontsize',16);
  xlabel('N')
  ylabel('Maximum error in u_N')
```

(d) With the non homogeneous Neumann Boundary conditions our weak formulation becomes as follows

$$\begin{aligned}\int_0^1 (-u'')v dx + \int_0^1 uv dx &= [-u'v]_0^1 + \int_0^1 u'v' dx + \int_0^1 uv dx \\ &= (-u'(1)v(1) + u'(0)v(0)) + \int_0^1 u'v' dx + \int_0^1 uv dx \\ &= -1.v(1) + \int_0^1 u'v' dx + \int_0^1 uv dx \text{ (by u'(1)=1 condition and } v(0) = 0) \\ &= -1.v(1) + a(u,v) + (u,v)\end{aligned}$$

Thus the variational form is

$$a(u,v) + (u,v) = (f,v) + v(1) \qquad \forall v \in V$$

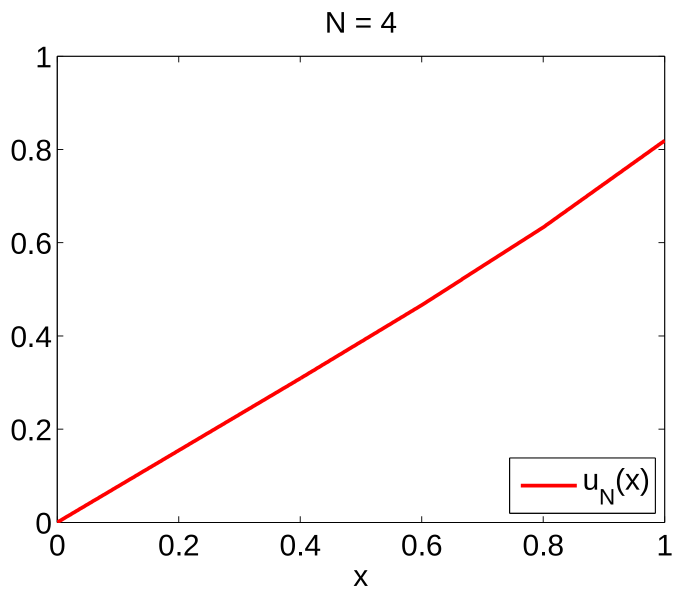
Now define the finite the finite element space V_n then our weak form becomes

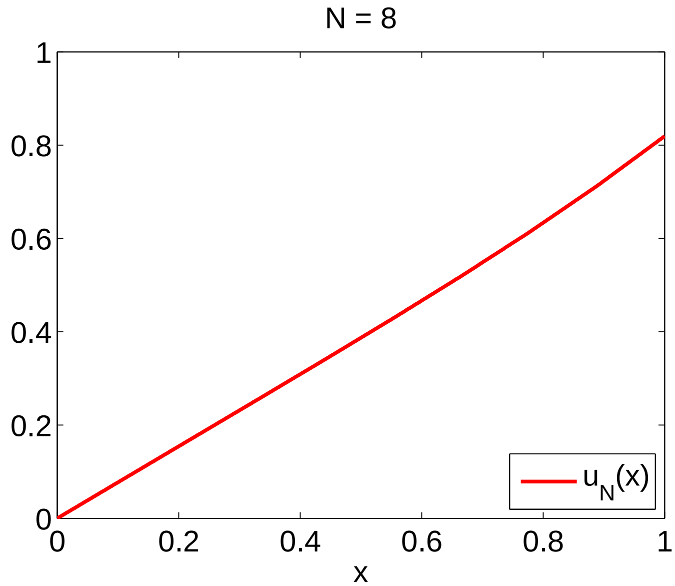
$$\sum_{j=1}^{n+1} \left(\underbrace{a(\phi_j, \phi_i)}_{K_{ij}} + \underbrace{(\phi_j, \phi_i)}_{M_{ij}} \right) c_j = (f, \phi_i) + \phi_i(1) \qquad i = 1, \cdots n+1$$

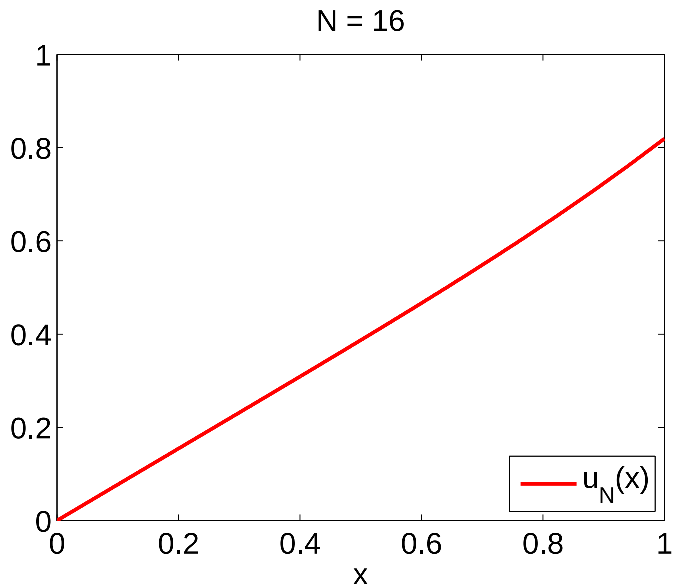
However $\phi_i(1) = 0$ for $i = 1, \cdots n$. It is only nonzero when $\phi_{n+1}(1) = 1$. So the matrix A and the load vector f are same with the part (d). Only we need to modify the last column of the f

$$f_{n+1} = (f, \phi_{n+1}) + 1$$

The requested plot is below.







The above plot was produced using the following MATLAB code.

```
% demo of the finite element method for the problem
%   -u'' + u = x(1-x), 0 < x < 1, u(0) = 0, u'(1) = 1


Nvec = [4 8 16];    % vector of N values we shall use
maxerr = zeros(size(Nvec));           % vector to hold the max errors for each N

% each pass of the following loop handles a new N value...
for j=1:length(Nvec)
   N = Nvec(j);
   h = 1/(N+1);
   x = [1:N]*h;

% construct the stiffness matrix (integrals done by hand)
   K = (2/h)*eye(N+1) - (1/h)*diag(ones(N,1),1) - (1/h)*diag(ones(N,1),-1);
   K(N+1,N+1) = 1/h;
```

```
   K;

   % construct the mass matrix (integrals done by hand)
   M = (2*h/3)*eye(N+1) + (h/6)*diag(ones(N,1),1) + (h/6)*diag(ones(N,1),-1);
   M(N+1,N+1) = h/3;
   M;
   % construct the matrix A
   A=K+M;

% Compute vector f containtin (f,phi_j), where f(x) = x(1-x)

 f = zeros(N+1,1);
 i = [1:N]';

  f = [-((h^2)*(2*h - 12*i + 12*h*(i.^2)))/12;
     -((h^2)*(h - 2))/12+1];

   c = A\f;
   c;

% plot the approximation solution
   uN = zeros(size(xx));
   for k=0:N
      uN = uN + c(k+1)*hat(xx,k+1,N);
   end
   figure(1), clf
   plot(xx, uN, 'r-','linewidth',2)
   set(gca,'fontsize',16)
   xlabel('x')
   legend( 'u_N(x)',4)
   title(sprintf('N = %d', N))


   input('hit return to continue')
end
```

3. [30 points: 10 points each]
Consider the *Euler Bernoulli beam equation,*

$$(k(x)u''(x))'' = f(x), \qquad 0 < x < 1,$$

with boundary conditions describing a beam that is *clamped* at both ends:

$$u(0) = u(1) = 0; \qquad u'(0) = u'(1) = 0$$

Here $k(x)$ is a positive-valued function that describes the material properties of the beam. With these boundary conditions, the eigenvalues and eigenvectors of this operator are diffcult to compute even if $k(x) = 1$. We will consider finite element solutions of this problem.

(a) Derive the weak form of the beam equation with the above boundary conditions, i.e., derive the weak problem

$$a(u, v) = (f, v); \qquad \text{for all } v \in V = C_D^4[0, 1],$$

where

$$C_D^4[0, 1] = \{u \in C^4[0, 1] : u(0) = u(1) = u'(0) = u'(1) = 0\}.$$

Specify the bilinear form $a(u, v)$, and show that it is an inner product on $C_D^4[0, 1]$
Note: for the problem $-(ku')' = f$, we do not explicitly impose Neumann boundary conditions, they follow 'naturally'. For the beam equation, we must impose all four boundary conditions on the space of test functions, $V = C_D^4[0, 1]$.

(b) Suppose that $V_n = span\{\phi_1, \cdots, \phi_n\}$ is an n-dimensional subspace of $C_D^4[0, 1]$. (Do not assume a particular form for the functions $\phi_1, \cdots, \phi_n$ at this point.)
Show how the Galerkin problem

$$a(u_n, v) = (f, v), \qquad \text{for all } v \in V_n$$

leads to the linear system $Ku = f$. Be sure to specify the entries of $K, u,$ and f .

(c) Now suppose we take for $\phi_1, \cdots, \phi_n$ the standard piecewise linear 'hat' functions used, for example, in Problem 2. Are these functions suitable for this problem? If so, describe the location of the nonzero entries of the matrix K. If not, roughly describe a better choice for the functions $\phi_1, \cdots, \phi_n$ and the explain which entries of K are nonzero for that choice.

Solution.

(a) For all $v \in V$ multiply both side of BVP with the test function v

$$(k(x)u''(x))''v = f(x)v, \qquad \forall v \in V,$$

Now integrate over 0 to 1

$$\int_0^1 (k(x)u''(x))''v(x)dx = \int_0^1 f(x)v(x)dx \qquad \forall v \in V$$

Apply integration by parts the left hand side

$$
\begin{aligned}
\int_0^1 (k(x)u''(x))''v(x)dx &= [(k(x)u''(x))'v(x)]_0^1 - \int_0^1 (k(x)u''(x))'v'(x)dx \\
&= ((k(1)u''(1))'v(1) - (k(0)u''(0))'v(0)) - \int_0^1 (k(x)u''(x))'v'(x)dx \\
&= -\int_0^1 (k(x)u''(x))'v'(x)dx \quad \text{by imposing } v(0) = 0, v(1) = 0 \\
&= [-(k(x)u''(x))v'(x)]_0^1 + \int_0^1 (k(x)u''(x))v''(x)dx \qquad \text{by integration by parts} \\
&= (-(k(1)u''(1))v'(1) + (k(0)u''(0))v'(0)) + \int_0^1 (k(x)u''(x))v''(x)dx \\
&= \int_0^1 (k(x)u''(x))v''(x)dx \quad \text{by imposing } v'(0) = 0, v'(1) = 0 \qquad \forall v \in V
\end{aligned}
$$

Thus we get

$$
\int_0^1 (k(x)u''(x))v''(x)dx = \int_0^1 f(x)v(x)dx
$$

where

$$
a(u,v) = \int_0^1 (k(x)u''(x))v''(x)dx \qquad \text{and} \qquad (f,v) = \int_0^1 f(x)v(x)\,dx \qquad \forall v \in V
$$

To show that the form $a(u,v)$ in part is an inner product, we must verify the three basic properties:

- **Symmetry** is apparent by inspection:

$$
\begin{aligned}
a(u,v) &= \int_0^1 k(x)u''(x)v''(x)\,dx \\
&= \int_0^1 k(x)v''(x)u''(x)\,dx = a(v,u).
\end{aligned}
$$

- **Linearity** follows from the linearity of differentiation and integration:

$$
\begin{aligned}
a(\alpha u + \beta v, w) &= \int_0^1 k(x)(\alpha u + \beta v)''(x)w''(x)\,dx \\
&= \int_0^1 k(x)(\alpha u''(x) + \beta v''(x))w''(x)\,dx \\
&= \int_0^1 \alpha k(x)u''(x)w''(x) + \beta k(x)v''(x)w''(x)\,dx \\
&= \alpha \int_0^1 k(x)u''(x)w''(x) + \beta \int_0^1 k(x)v''(x)w''(x)\,dx \\
&= \alpha a(u,w) + \beta a(v,w).
\end{aligned}
$$

- **Positivity** requires that $a(u,u) \geq 0$ and $a(u,u) = 0$ only when $u = 0$. Note that

$$
\begin{aligned}
a(u,u) &= \int_0^1 k(x)u''(x)u''(x)\,dx \\
&= \int_0^1 k(x)\Big(u''(x)\Big)^2\,dx.
\end{aligned}
$$

Since $k(x)$ is positive for all $x \in [0,1]$, integrand is non-negative, and hence $a(u,u) \geq 0$. To have $a(u,u) = 0$,we must have $u''(x) = 0$ for all $x \in [0,1]$, which is only possible if $u(x) = bx + c$ by BC $b = c = 0$ then $u(x) = 0$ for all $x \in [0,1]$, i.e., $u = 0$.

(b) Let $V_n = span\{\phi_1, \cdots, \phi_n\}$ is an n-dimensional subspace of $C_D^4[0,1]$. We would like to shot that

$$a(u_n, v) = (f, v), \qquad \text{for all } v \in V_n$$

leads to a linear system.

The Galerkin solution can be defined by the linear combination of basis function $u_n = \sum_{j=1}^{n} u_j \phi_j(x)$ for coefficients u_j. Then

$$a(\sum_{j=1}^{n} u_j \phi_j(x), \phi_i(x)) = (f, \phi_i(x)), \qquad \text{for } i = 1, \cdots, n$$

By the linearity of bilinear form

$$\sum_{j=1}^{n} a(\phi_j(x), \phi_i(x)) u_j = (f, \phi_i(x)), \qquad \text{for } i = 1, \cdots, n$$

This leads to a linear system of N equations with N unknown. If we define that system as $Ku = f$ then each component of K and f will be

$$K_{ij} = a(\phi_j(x), \phi_i(x)) = \int_0^1 k(x)\phi_j''(x)\phi_i''(x)\,dx \qquad \text{for } i, j = 1, \cdots, n$$

and

$$f_i = (f(x), \phi_i(x)) = \int_0^1 f(x)\phi_i(x)\,dx \qquad \text{for } i = 1, \cdots, n$$

u is our solution vector with $u = [u_1, u_2, \cdots, u_n]^T$.

(c) Now if we take for $\phi_1, \cdots, \phi_n$ the standard piecewise linear 'hat' functions as in Problem 2. We will get $\phi_i''(x) = 0$ for $i = 1, \cdots, n$. That leads our stiffness matrix $K = 0$. Therefore, this hat function is not suitable for Euler Bernoulli beam problem.

By the same idea, we want to define a hat function satisfy following property

$$\phi_i(x_j) = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if} i \neq j; \end{cases}$$

Also for

$$K_{ij} = \int_0^1 k(x)\phi_j''(x)\phi_i''(x)\,dx$$

we need second derivative of the hat function. For simplify, let pick a hat function second derivative is constant.

In that case , the new hat function

$$\phi_i(x) = \begin{cases} \left(\dfrac{x - x_{i-1}}{h}\right)^2 & \text{if } x \in [x_{i-1}, x_i]; \\ \left(\dfrac{x_{i+1} - x}{h}\right)^2 & \text{if } x \in [x_i, x_{i+1}); \\ 0 & \text{otherwise}; \end{cases}$$

With these choose we will get again tridiagonal matrix K such that

$$K_{ii} = \int_0^1 k(x)\phi_i''(x)\phi_i''(x)\,dx = \int_{x_{i-1}}^{x_{i+1}} k(x)\phi_i''(x)\phi_i''(x)\,dx$$

$$K_{i(i+1)} = \int_0^1 k(x)\phi_i''(x)\phi_{i+1}''(x)\,dx = \int_{x_i}^{x_{i+1}} k(x)\phi_i''(x)\phi_{i+1}''(x)\,dx$$

By symmetry $K_{ij} = K_{ji}$.
Finally, for $|i-j| > 1$,

$$K_{ij} = \int_0^1 k(x)\phi_i''(x)\phi_j''(x)\,dx = 0$$
