CAAM 336 · DIFFERENTIAL EQUATIONS

Problem Set 7 · Solutions

Posted Wednesday 25 March 2015. Due Wednesday 1 April 2015, 5pm.

General advice: You may compute any integrals you encounter using symbolic mathematics tools such as WolframAlpha, Mathematica, or the Symbolic Math Toolbox in MATLAB.

1. [40 points: 8 points each] Let $k(x)$ and $p(x)$ be two positive-valued continuous functions on $[0, 1]$.

   (a) Define
   $$V = C_D^2[0,1] = \Big\{u \in C^2[0,1] : u(0) = u(1) = 0\Big\}.$$
   Derive the weak form of the differential equation
   $$-\frac{d}{dx}\Big(k(x)\frac{du}{dx}\Big) + p(x)u = f(x), \quad 0 < x < 1,$$
   subject to the boundary conditions
   $$u(0) = u(1) = 0;$$
   that is, transform this differential equation into a problem of the form:
   $$\text{Find } u \in V \text{ such that } a(u, v) = (f, v) \text{ for all } v \in V,$$
   where $(\cdot, \cdot)$ denotes the usual inner product $(f, g) = \int_0^1 f(x)g(x)\,dx$, and $a(\cdot, \cdot)$ is some bilinear form that you should specify. Verify that $a(u, v)$ is still an inner product.

   (b) Let $p(x) = 1$, $k(x) = \epsilon$, and let the source function $f(x) = 1$. Construct the finite element system $A\alpha = b$, where
   $$A_{ij} = a(\phi_j, \phi_i), \quad b_i = \int_0^1 f(x)\phi_i(x)$$
   using the approximation space V_N given by the piecewise linear *hat functions*: for $n \geq 1$, $h = 1/(N + 1)$, and $x_k = kh$ for $k = 0, \ldots, N + 1$, we have
   $$\phi_k(x) = \begin{cases} (x - x_{k-1})/h, & x \in [x_{k-1}, x_k); \\ (x_{k+1} - x)/h, & x \in [x_k, x_{k+1}); \\ 0, & \text{otherwise.} \end{cases}$$
   Hint: for this specific choice of $p(x), k(x)$, it may be easier to show that you can express
   $$A = \epsilon K + M, \quad K_{ij} = \int_0^1 \frac{\partial \phi_j}{\partial x}\frac{\partial \phi_i}{\partial x}$$
   where the form of M is specified in Homework 6, 4(d).

   (c) This specific equation corresponds to the simplest steady-state *reaction-diffusion* equation, where $u(x)$ is the concentration of some solvent, and the choices of $p(x)$ model local chemical reactions that may occur due to that solvent (multiple chemicals interacting may be modeled using *systems* of reaction-diffusion equations).

   Solve the above system with $N = 32$ and $\epsilon = .1, .25, 1$. What do you observe about the solution as ϵ becomes smaller?

(d) Let the space V now be defined as

$$V = \Big\{ u \in C^2[0,1] : u(0) = \frac{du}{dx}(1) = 0 \Big\}.$$

Derive the weak form of the differential equation

$$-\frac{d}{dx}\Big(k(x) \frac{du}{dx} \Big) + p(x)u = f(x), \quad 0 < x < 1,$$

subject to the boundary conditions

$$u(0) = \frac{du}{dx}(1) = 0;$$

that is, transform this differential equation into a problem of the form:

$$\text{Find } u \in V \text{ such that } a(u,v) = (f,v) \text{ for all } v \in V,$$

where $(\cdot,\cdot)$ denotes the usual inner product $(f,g) = \int_0^1 f(x)g(x)\,dx$, and $a(\cdot,\cdot)$ is some bilinear form that you should specify.

(e) Show that the form $a(u,v)$ from part (d) is an inner product for $u, v \in V$.

Solution.

(a) Multiply the differential equation with some function v from the space V and integrate from $x = 0$ to $x = 1$ to obtain

$$\int_0^1 \Big(-\frac{d}{dx}\Big(k(x)\frac{du}{dx}(x)\Big)v(x) + p(x)u(x)v(x) \Big)\,dx = \int_0^1 f(x)v(x)\,dx.$$

Break the integral on the left into pieces to obtain

$$\int_0^1 \Big(-\frac{d}{dx}\Big(k(x)\frac{du}{dx}(x)\Big)\Big)v(x)\,dx + \int_0^1 \Big(p(x)u(x)\Big)v(x)\,dx = \int_0^1 f(x)v(x)\,dx.$$

Integrate the first integral by parts to obtain

$$-\Big[\kappa(x)\frac{du}{dx}(x)v(x)\Big]_0^1 + \int_0^1 k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x)\,dx + \int_0^1 \Big(p(x)u(x)\Big)v(x)\,dx = \int_0^1 f(x)v(x)\,dx.$$

The boundary terms vanish due to the fact that $v(0) = v(1) = 0$ if $v \in V = C_D^2[0,1]$. We consolidate the integrals on the left to arrive at the weak problem:

$$\text{Find } u \in V \text{ such that} \quad a(u,v) = (f,v) \quad \text{for all } v \in V,$$

where

$$a(u,v) = \int_0^1 \Big(k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x) + p(x)u(x)v(x) \Big)\,dx.$$

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

- **Symmetry** is apparent by inspection:

$$\begin{aligned} a(u,v) &= \int_0^1 \Big(k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x) + p(x)u(x)v(x) \Big)\,dx \\ &= \int_0^1 \Big(k(x)\frac{dv}{dx}(x)\frac{du}{dx}(x) + p(x)u(x)v(x) \Big)\,dx \;=\; a(v,u). \end{aligned}$$

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

$$
\begin{aligned}
a(\alpha u+\beta v,w) &= \int_0^1 \Big(k(x)\frac{d(\alpha u(x)+\beta v(x))}{dx}(x)\frac{dw}{dx}(x)+p(x)(\alpha u(x)+\beta v(x))w(x)\Big)\,dx \\
&= \int_0^1 \Big(k(x)\Big(\alpha\frac{du(x)}{dx}+\beta\frac{dv(x)}{dx}\Big)\frac{dw}{dx}(x)+p(x)(\alpha u(x)+\beta v(x))w(x)\Big)\,dx \\
&= \alpha\int_0^1 \Big(k(x)\frac{du(x)}{dx}\frac{dw}{dx}(x)+p(x)u(x)w(x)\Big)\,dx \\
&\quad +\beta\int_0^1 \Big(k(x)\frac{dv(x)}{dx}\frac{dw}{dx}(x)+p(x)v(x)w(x)\Big)\,dx \\
&= \alpha a(u,w)+\beta a(v,w).
\end{aligned}
$$

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

$$
\begin{aligned}
a(u,u) &= \int_0^1 \Big(k(x)\frac{du}{dx}(x)\frac{du}{dx}(x)+p(x)u(x)u(x)\Big)\,dx \\
&= \int_0^1 \Big(k(x)\Big(\frac{du}{dx}(x)\Big)^2+p(x)\big(u(x)\big)^2\Big)\,dx.
\end{aligned}
$$

  Since $k(x)$ and $p(x)$ are both positive for all $x\in[0,1]$, each 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 $x\in[0,1]$, and $du(x)/dx=0$ for all $x\in[0,1]$, which is only possible if $u(x)=0$ for all $x\in[0,1]$, i.e., $u=0$.

(b) If $p(x)=1$ and $k(x)=\epsilon$, our formulation reduces down to $a(u,v)=(f,v)$ where

$$a(u,v)=\int_0^1 u(x)v(x)dx+\epsilon\int_0^1\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}dx.$$

Then, if $A_{ij}=a(\phi_j,\phi_i)$, we have

$$A_{ij}=\int_0^1\phi_j(x)\phi_i(x)dx+\epsilon\int_0^1\frac{\partial\phi_j}{\partial x}\frac{\partial\phi_i}{\partial x}dx=M_{ij}+\epsilon K_{ij}$$

where $M_{ij}=\int_0^1\phi_j(x)\phi_i(x)dx$ is the Gram matrix for hat functions using the L^2 inner product. The entries of M and K are known to be

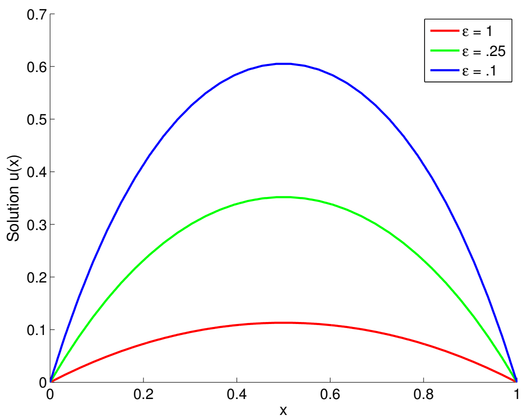
$$M_{ij}=\begin{cases} M_{i,i} &= 2h/3 \\ M_{i+1,i} &= h/6 \\ M_{ij} &= 0, \quad |i-j|>1 \end{cases}, \qquad K_{ij}=\begin{cases} K_{i,i} &= 2/h \\ K_{i+1,i} &= -1/h \\ K_{ij} &= 0, \quad |i-j|>1 \end{cases}$$

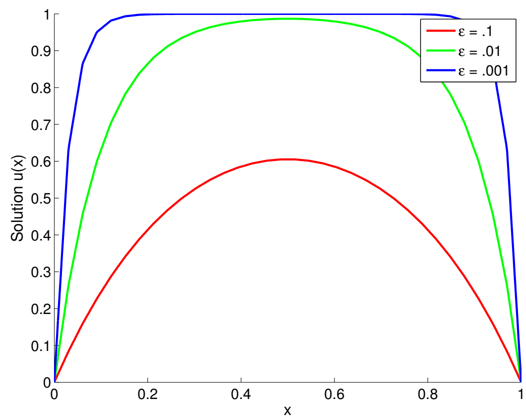
from class and from previous homeworks.

(c) The code to generate the figures for this problem are given below.

```
iter = 1;
C = hsv(3);
for ep = [1 .25 .1];
    N = 32;
    h = 1/(N+1);
    x = [1:N]*h;

    M = (2/3)*diag(ones(N,1)) + (1/6)*diag(ones(N-1,1),1) + (1/6)*diag(ones(N-1,1),-1);
    M = h*M;

    K = 2*diag(ones(N,1)) - diag(ones(N-1,1),1) - diag(ones(N-1,1),-1);
```

```
    K = (1/h)*K;

    A = M + ep*K;

    b = h*ones(N,1);

    c = A\b;

    xx = linspace(0,1,500)';      % finely spaced points between 0 and 1.
    hold on
    % plot the approximation solution
    uN = zeros(size(xx));
    for j=1:N
        uN = uN + c(j)*hat(xx,j,N);
    end
    plot(xx,uN,'color',C(iter,:),'linewidth',2)
    iter = iter + 1;
end
legend('\epsilon = 1','\epsilon = .25','\epsilon = .1')
set(gca,'fontsize',14)
xlabel('x','fontsize',15)
ylabel('Solution u(x)','fontsize',15)
print(gcf,'-depsc','../ep1.eps')
```





As ϵ decreases, the temperature in the bar increases. It is difficult to see with $\epsilon \geq .1$, but as ϵ gets small, the solution actually develops additional characteristics called *boundary layers*, where the solution becomes very steep near the boundaries. *Graders: please give credit just for noting the temperature increases, as the boundary layer phenomena was not visible for the range of* ϵ *specified in the problem.*

(d) The process is very similar to part (a). Multiply the differential equation with some function v from the space V and integrate from $x = 0$ to $x = 1$ to obtain

$$\int_0^1 \left(-\frac{d}{dx}\Big(k(x)\frac{du}{dx}(x)\Big)v(x) + p(x)u(x)v(x) \right) dx = \int_0^1 f(x)v(x)\,dx.$$

Break the integral on the left into pieces to obtain

$$\int_0^1 \left(-\frac{d}{dx}\Big(k(x)\frac{du}{dx}(x)\Big)\right)v(x)\,dx + \int_0^1 \Big(p(x)u(x)\Big)v(x)\,dx = \int_0^1 f(x)v(x)\,dx.$$

Integrate the first integral by parts to obtain

$$-\left[\kappa(x)\frac{du}{dx}(x)v(x)\right]_0^1 + \int_0^1 k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x)\,dx + \int_0^1 \Big(p(x)u(x)\Big)v(x)\,dx = \int_0^1 f(x)v(x)\,dx.$$

The first term disappears because of the boundary conditions $v(0) = 0$ and $du(1)/dx = 0$. We consolidate the integrals on the left to arrive at the weak problem:

$$\text{Find } u \in V \text{ such that} \quad a(u,v) = (f,v) \quad \text{for all } v \in V,$$

where

$$a(u,v) = \int_0^1 \Big(k(x)\frac{du}{dx}(x)\frac{dv}{dx}(x) + p(x)u(x)v(x) \Big)\, dx.$$

(e) The proof for (e) is identical to the proof for (a). *Graders: please give full credit if the student notices this.*

---

2. [40 points: 10 points each] Use the finite element method to solve the differential equation

$$-(u'(x)\kappa(x))' = 2x, \qquad 0 < x < 1$$

for $\kappa(x) = 1 + x^2$, subject to homogeneous Dirichlet boundary conditions,

$$u(0) = u(1) = 0,$$

with the approximation space V_N given by the piecewise linear *hat functions* that featured on earlier problem sets: For $n \geq 1$, $h = 1/(N+1)$, and $x_k = kh$ for $k = 0, \ldots, N+1$, we have

$$\phi_k(x) = \begin{cases} (x - x_{k-1})/h, & x \in [x_{k-1}, x_k); \\ (x_{k+1} - x)/h, & x \in [x_k, x_{k+1}); \\ 0, & \text{otherwise.} \end{cases}$$

(a) Write MATLAB code that constructs the stiffness matrix $\mathbf{K}$ for a given value of N, with $\kappa(x) = 1 + x^2$.
[You may edit the `fem_demo1.m` code from the class website. You should compute all necessary integrals (by hand or using a symbolic package) so as to obtain clean formulas that depend on h and the index of the hat functions involved (e.g.,$a(\phi_j, \phi_j)$ can depend on j).]

(b) Write MATLAB code that constructs the load vector $\mathbf{f}$ for a given value of N, with $f(x) = 2x$.

(c) For $N = 7$ and $N = 15$, produce plots comparing your solution u_N to the true solution

$$u(x) = (4/\pi)\tan^{-1}(x) - x.$$

(Note that you can compute $\tan^{-1}(x)$ as `atan(x)` in MATLAB.)

(d) Produce a `loglog` plot showing how the error

$$\max_{x \in [0,1]} |u_N(x) - u(x)|$$

decreases as N increases. (For example, take $N = 8, 16, 32, 64, 128, 256, 512$.) On the same plot, show N^{-2} for the same values of N. If your code from parts (a) and (b) is working, your error curve should have the same slope as the N^{-2} curve. (Consult the `fem_demo1.m` code on the website for a demonstration of the style of plot we intend for part (d); edit this code as you like.)

Solution.

(a) First we compute the energy inner product of the basis functions. Note that

$$\frac{d\phi_k}{dx}(x) = \begin{cases} 1/h, & x \in [x_{k-1}, x_k); \\ -1/h, & x \in [x_k, x_{k+1}); \\ 0, & \text{otherwise.} \end{cases}$$

Thus we have

$$\begin{aligned} a(\phi_j, \phi_j) &= \int_0^1 (1+x^2)\Big(\frac{d\phi_j}{dx}(x)\Big)^2\,dx \\ &= \int_{x_{j-1}}^{x_j} (1+x^2)\Big(\frac{1}{h}\Big)^2\,dx + \int_{x_j}^{x_{j+1}} (1+x^2)\Big(-\frac{1}{h}\Big)^2\,dx \\ &= \frac{1}{h^2}\int_{x_{j-1}}^{x_{j+1}} (1+x^2)\,dx \;=\; \frac{1}{h^2}\Big[x + \frac{x^3}{3}\Big]_{x_{j-1}}^{x_{j+1}} \;=\; \frac{2}{h} + \frac{2h}{3} + 2hj^2, \end{aligned}$$

$$\begin{aligned} a(\phi_j, \phi_{j+1}) &= \int_0^1 (1+x^2)\Big(\frac{d\phi_j}{dx}(x)\Big)\Big(\frac{d\phi_{j+1}}{dx}(x)\Big)\,dx \\ &= \int_{x_j}^{x_{j+1}} (1+x^2)\Big(-\frac{1}{h}\Big)\big(\frac{1}{h}\Big)\,dx \\ &= -\frac{1}{h^2}\int_{x_j}^{x_{j+1}} (1+x^2)\,dx \;=\; -\frac{1}{h^2}\Big[x + \frac{x^3}{3}\Big]_{x_j}^{x_{j+1}} \;=\; -\frac{1}{h} - h\Big(j^2 + j + \frac{1}{3}\Big), \end{aligned}$$

and for $|j-k| > 1$,

$$a(\phi_j, \phi_k) = 0$$

since $\big(d\phi_j(x)/dx\big)\big(d\phi_k(x)/dx\big) = 0$ for all $x \in [0,1]$ (except at the nodes x_ℓ, where strictly speaking these derivatives are not defined—but these single isolated points do not add anything to the integral). The stiffness matrix is given by

$$\mathbf{K} = \begin{bmatrix} a(\phi_1,\phi_1) & \cdots & a(\phi_1,\phi_n) \\ \vdots & \ddots & \vdots \\ a(\phi_n,\phi_1) & \cdots & a(\phi_n,\phi_n) \end{bmatrix}.$$

(b) Next we compute the entries of the load vector:

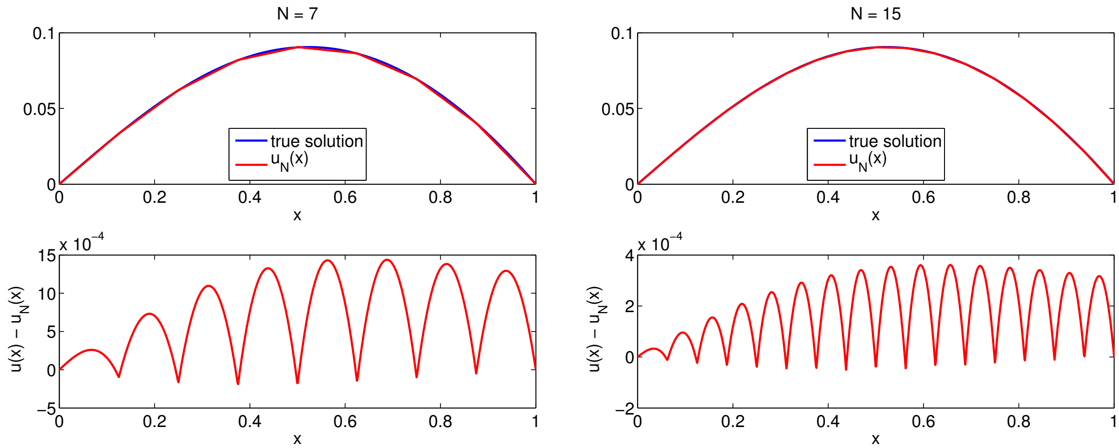
$$\begin{aligned} (f, \phi_j) &= \int_0^1 f(x)\phi_j(x)\,dx \\ &= \int_{x_{j-1}}^{x_j} (2x)\Big(\frac{x - x_{j-1}}{h}\Big)\,dx + \int_{x_j}^{x_{j+1}} (2x)\Big(\frac{x_{j+1} - x}{h}\Big)\,dx \\ &= \frac{1}{h}\Big[\frac{2x^3}{3} - x^2 x_{j-1}\Big]_{x_{j-1}}^{x_j} + \frac{1}{h}\Big[x^2 x_{j+1} - \frac{2x^3}{3}\Big]_{x_j}^{x_{j+1}} \\ &= 2h^2 j. \end{aligned}$$

The load vector is given by

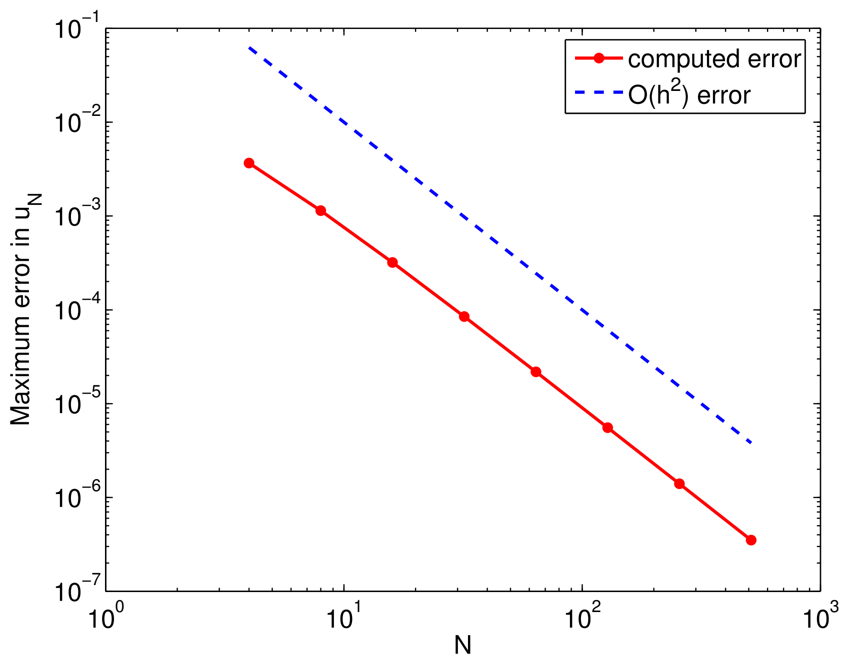
$$\mathbf{f} = \begin{pmatrix} (f,\phi_1) \\ \vdots \\ (f,\phi_n) \end{pmatrix}.$$

The MATLAB code at the end of this problem shows generates these matrices and produces plots similar to those shown in (b) and (c).

(c) The following plots show the solution (and error) at $N = 7$ (left) and $N = 15$ (right).



(d) The following plot shows the decay of the error as a function of N. Notice that the error decays like $1/N^2$.




```
% demo of the finite element method for the problem
%   -d/dx((1+x^2) du/dx)  = 2x, 0 < x < 1, u(0) = u(1) = 0.
% which has exact solution u(x) = (4/pi)*atan(x) - x.

Nvec = [4 8 16 32 64 128 256 512];    % 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)
   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 = 2*h^2*[1:N]';

% 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.
   u = (4/pi)*atan(xx)-xx;      % true solution

   figure(1), clf
   subplot(2,1,1)
   plot(xx, u, 'b-','linewidth',2)
   hold on

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

% plot the error in the solution for this N
   subplot(2,1,2)
   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(2), 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')
  print -depsc2 femb.eps
```

---

3. [20 points: 5 points each] A classical problem in quantum mechanics models a particle moving in an infinite square well, subject to an infinite potential at a point. The result is a Schrödinger operator posed on $C_D^2[0,1]$ of the form
$$Lu = -u'' + \delta_{1/2}u,$$
where $\delta_{1/2}$ is a "delta function" centered at the location of the infinite potential, $x = 1/2$. A beautiful theory supports these exotic functions (more properly called *distributions*). For this problem, you need only know the following fact: for any function $g \in C[0,1]$,
$$\int_0^1 \delta_{1/2}(x)g(x)\ dx = g(1/2).$$

The equation $Lu = f$ has the equivalent weak form

$$a(u,w) = (f,w) \qquad \text{for all } w \in V = C_D^2[0,1],$$

where

$$a(u,w) = \int_0^1 \Big(u'(x)w'(x) + \delta_{1/2}(x)u(x)w(x)\Big)\,dx.$$

We wish to use the Galerkin method to approximate solutions to $Lu = f$ from the finite dimensional subspace $V_N = \text{span}\{\phi_1, \ldots, \phi_N\}$. Use as basis vectors the eigenfunctions from the problem without the potential at $x = 1/2$:

$$\phi_k(x) = \sqrt{2}\sin(k\pi x).$$

(a) Compute a general formula for $a(\phi_j, \phi_k)$.

(b) Write out (by hand) the stiffness matrix for $N = 5$.

(c) Write down a general formula for the entries in the load vector, (f, ϕ_k), when $f(x) = 1$. (You may use formulas from prior homework.)

(d) Plot your approximate solutions to $-u''(x) + \delta_{1/2}(x)u(x) = 1$ for $N = 5$ and $N = 35$.

Solution.

(a) Compute

$$\begin{aligned} a(\phi_j, \phi_k) &= \int_0^1 \big(\phi_j'(x)\phi_k'(x) + \delta_{1/2}(x)\phi_j(x)\phi_k(x)\big)\,dx \\ &= 2kj\pi^2 \int_0^1 \cos(j\pi x)\cos(k\phi x)\,dx + 2\int_0^1 \delta_{1/2}(x)\sin(j\pi x)\sin(k\pi x) \\ &= 2kj\pi^2 \int_0^1 \cos(j\pi x)\cos(k\phi x)\,dx + 2\sin(j\pi/2)\sin(k\pi/2). \end{aligned}$$

The integral in this last expression is $1/2$ when $j = k$, and zero otherwise. The second term will be zero if either j or k is even (since in that case one of the sine terms must be zero). If both j and k are odd, this term will be nonzero, ± 2. In general, we can write

$$a(\phi_j, \phi_k) = \begin{cases} j^2\pi^2 + 2\sin^2(j\pi/2), & \text{if } j = k; \\ 2\sin(j\pi/2)\sin(k\pi/2), & \text{otherwise.} \end{cases}$$

[**GRADERS**: the amount that students simplify $a(\phi_j, \phi_k)$ will vary. The ultimate solution need not take the precise form that we have given above, but it should be simplified beyond just writing down the definition of $a(\phi_j, \phi_k)$.]

(b) For $N = 5$ we have

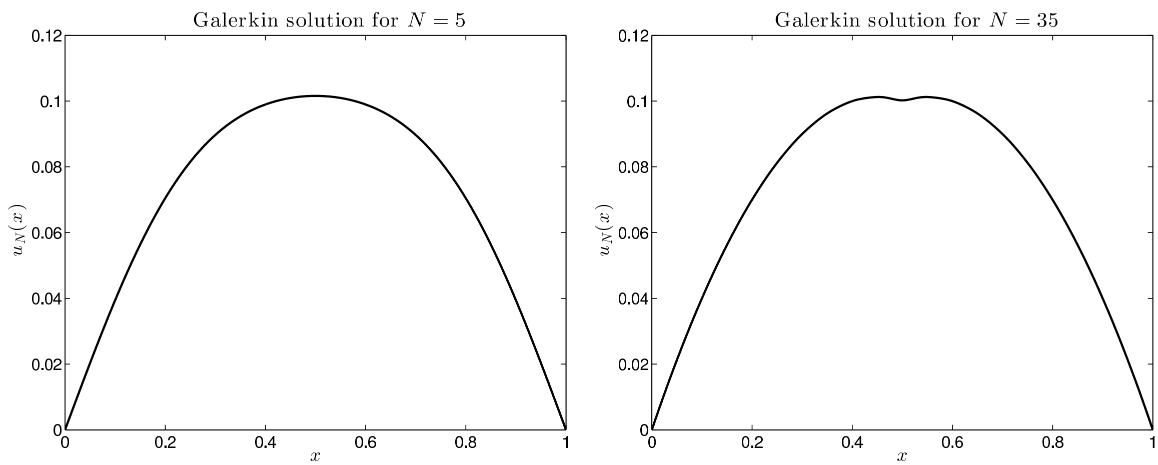
$$\begin{bmatrix} \pi^2+2 & 0 & -2 & 0 & 2 \\ 0 & 4\pi^2 & 0 & 0 & 0 \\ -2 & 0 & 9\pi^2+2 & 0 & -2 \\ 0 & 0 & 0 & 16\pi^2 & 0 \\ 2 & 0 & -2 & 0 & 25\pi^2+2 \end{bmatrix}$$

(c) The entries of the load vector are

$$(f, \phi_k) = \int_0^1 1 \cdot \sqrt{2} \sin(k\pi x)\, dx = \begin{cases} 2\sqrt{2}/(n\pi), & \text{if } k \text{ is odd;} \\ 0, & \text{if } k \text{ is even,} \end{cases}$$

as computed in previous examples earlier in the semester.
[**GRADERS**: students do not need to show work for this formula.]

(d) Approximate solutions for $N = 5$ and $N = 35$ are shown below, followed by the code that produced them.




```
for N = [5 35]

 K = zeros(N); f = zeros(N,1);

 for j=1:N, for k=1:N
    K(j,k) = 2*sin(j*pi/2)*sin(k*pi/2);
 end, end

 K = K + diag([1:N].^2*pi^2);

 for k=1:N
    f(k) = (sqrt(2)/pi)*(1+(-1).^(k+1))./k;
 end

 c = K\f;

 xx = linspace(0,1,1000);
 uN = zeros(size(xx));

 for k=1:N
    uN = uN + c(k)*sqrt(2)*sin(k*pi*xx);
 end

 figure(N), clf
 plot(xx,uN,'k-','linewidth',2)
 title(sprintf('Galerkin solution for $N=%d$',N),'interpreter','latex','fontsize',18)
 xlabel('$x$','interpreter','latex','fontsize',16)
 ylabel('$u_N(x)$','interpreter','latex','fontsize',16)
 set(gca,'fontsize',14)
 eval(sprintf('print -depsc2 delta_%d', N))

 if N==5, disp(K), end
end
```