CAAM 336 · DIFFERENTIAL EQUATIONS

Homework 9 · Solutions

Posted Wednesday 29 January 2014. Due 1pm Friday 7 February 2014.

9. [25 points]

Suppose $N \geq 1$ is an integer and define $h = 1/(N+1)$ and $x_j = jh$ for $j = 0, \ldots, N+1$. We can approximate the differential equation

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

with homogeneous Dirichlet boundary conditions $u(0) = u(1) = 0$ by the matrix equation

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{bmatrix},$$

where $u_j \approx u(x_j)$. (Entries of the matrix that are not specified are zero.)

(a) Suppose that $f(x) = 25\pi^2\cos(5\pi x)$. Compute and plot the approximate solutions obtained when $N = 8, 16, 32, 64, 128$. To solve the linear systems, you may use MATLAB's 'backslash' command: `u = A\f`.

(b) For each value of N used in part (a) compute the maximum error $|u_j - u(x_j)|$, given that the true solution is

$$u(x) = 1 - 2x - \cos(5\pi x).$$

Plot this error using a `loglog` plot with error on the vertical axis and N on the horizontal axis.

(c) Explain what adjustments to the right hand side of the matrix equation are necessary to accommodate the inhomogeneous Dirichlet boundary conditions

$$u(0) = 1, \quad u(1) = 2.$$

(d) Compute and plot the approximate solutions obtained when

$$u(0) = 1, \quad u(1) = 2$$

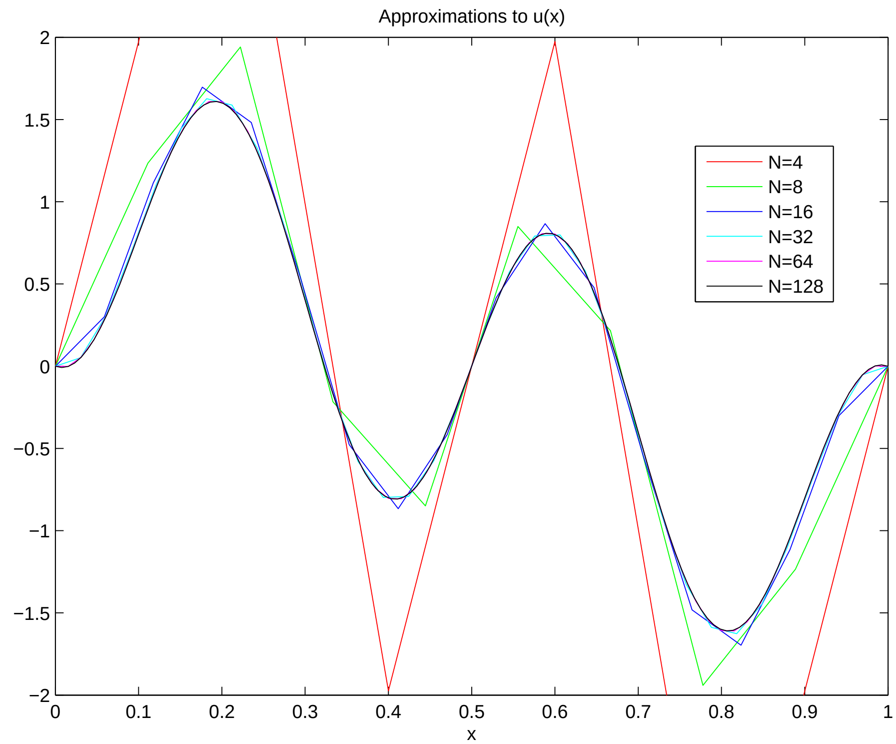
with $N = 8, 32, 128$.

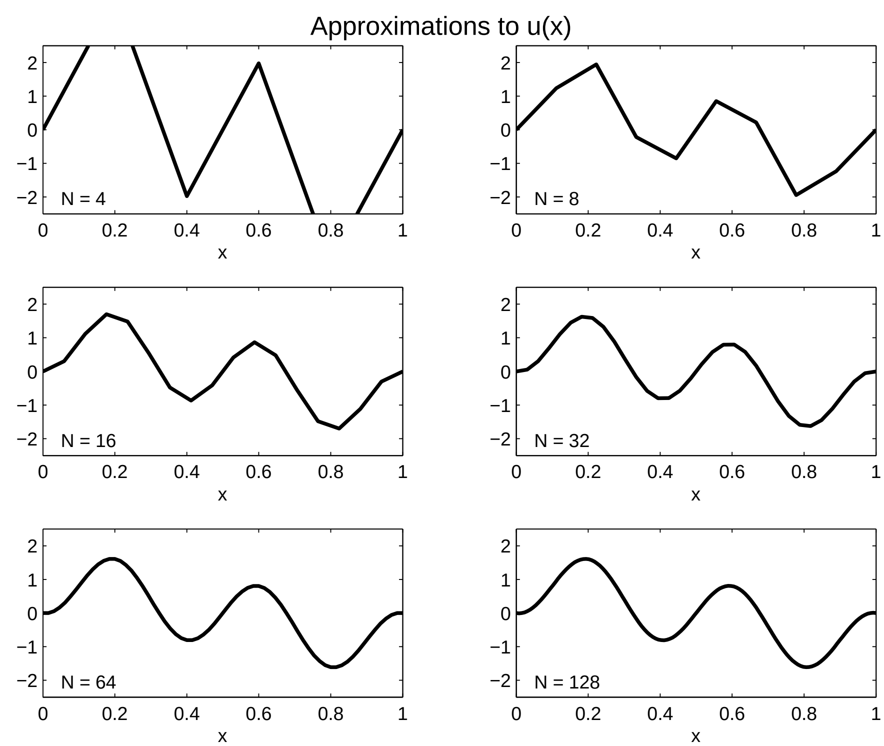
Solution.

(a) [7 points] The code used to produce the results shown in this part and part (b) is below.

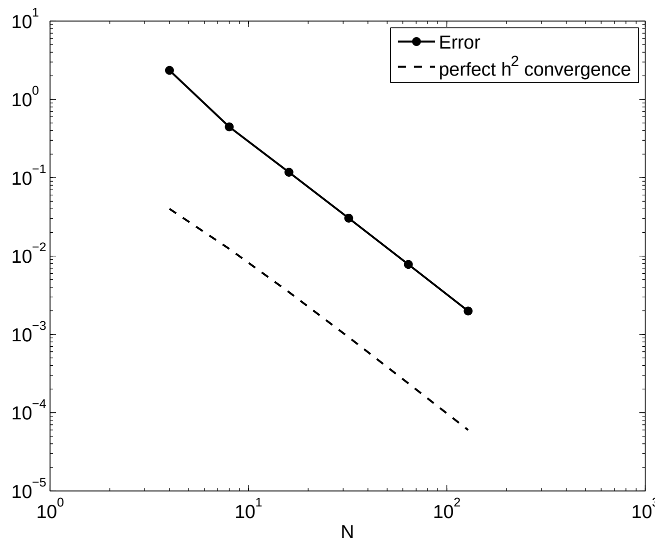
```
clear
clc
Nvec = [4 8 16 32 64 128];
err = zeros(size(Nvec));
colors='rgbcmk';
figure(1)
clf
suptitle('Approximations to u(x)')
figure(2)
clf
for j=1:length(Nvec)
    N = Nvec(j);
    h = 1/(N+1);
    x = h*(1:N).';
    A = (-2*eye(N)+diag(ones(N-1,1),1)+diag(ones(N-1,1),-1))/(h^2);
    f = 25*pi^2*cos(5*pi*x);
    u = A\f;
    % plot the function, adding in the homogeneous values at the boundary;
    % this tacks on extra entries for the x and u vectors:
    figure(1)
    subplot(3,2,j)
    plot([0;x;1],[0;u;0],'k-','linewidth',2)
    hold on
    axis([0 1 -2.5 2.5])
    xlabel('x')
    text(.05,-2.05,sprintf('N = %d',N))
    figure(2)
    plot([0;x;1],[0;u;0],colors(j))
    hold on
    legendStr{j}=['N=' num2str(N)];
    % compute error
    true_u = 1-2*x-cos(5*pi*x);
    err(j) = max(abs(true_u - u));
end
figure(1)
print -depsc2 separate_a.eps
figure(2)
axis([0 1 -2 2])
xlabel('x')
title('Approximations to u(x)')
legend(legendStr,'Location','best');
print -depsc2 together_a.eps

% error plot
figure(3)
clf
loglog(Nvec, err, 'k.-','linewidth',1.5,'markersize',20)
hold on
loglog(Nvec, (Nvec+1).^(-2), 'k--', 'linewidth',1.5)
legend('Error', 'perfect h^2 convergence')
set(gca,'fontsize',14)
xlabel('N')
print -depsc2 error_b.eps
```

Two acceptable styles of the desired plots are shown below. We also include $N = 4$ for the sake of comparison.





(b) [5 points] The desired plot is shown below.



On the above plot, we include a line showing the rate of convergence if the error was reduced exactly like h^2. We see that the rates (that is, the slope of the true error curve (solid) and this h^2 curve (dashed)) are quite close. The follows from the fact that, for small enough h, our approximation to the second derivative makes an error which is very nearly proportional to h^2.

(c) [7 points] Suppose we have $u(0) = \alpha$ and $u(1) = \beta$. The approximation to the second derivative

$$u''(x_j) \approx \frac{u(x_{j-1}) - 2u(x_j) + u(x_{j+1})}{h^2},$$

for $j = 1, \ldots, N$, yields

$$\frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} u(x_0) \\ u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \\ u(x_{N+1}) \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{bmatrix}.$$

Now,

$$\frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} u(x_0) \\ u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \\ u(x_{N+1}) \end{bmatrix}$$

$$= \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \\ 0 \end{bmatrix} + \begin{bmatrix} u(0) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ u(1) \end{bmatrix}\right)$$

$$= \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \\ 0 \end{bmatrix} + \begin{bmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \beta \end{bmatrix}\right)$$

$$= \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \\ 0 \end{bmatrix} + \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \beta \end{bmatrix}$$

$$= \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \end{bmatrix} + \begin{bmatrix} \alpha/h^2 \\ 0 \\ \vdots \\ 0 \\ \beta/h^2 \end{bmatrix}$$

and so

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \end{bmatrix} \approx \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{bmatrix} - \begin{bmatrix} \alpha/h^2 \\ 0 \\ \vdots \\ 0 \\ \beta/h^2 \end{bmatrix}.$$

Hence, we can approximate the differential equation

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

with inhomogeneous Dirichlet boundary conditions

$$u(0) = 1, \quad u(1) = 2$$

by the matrix equation

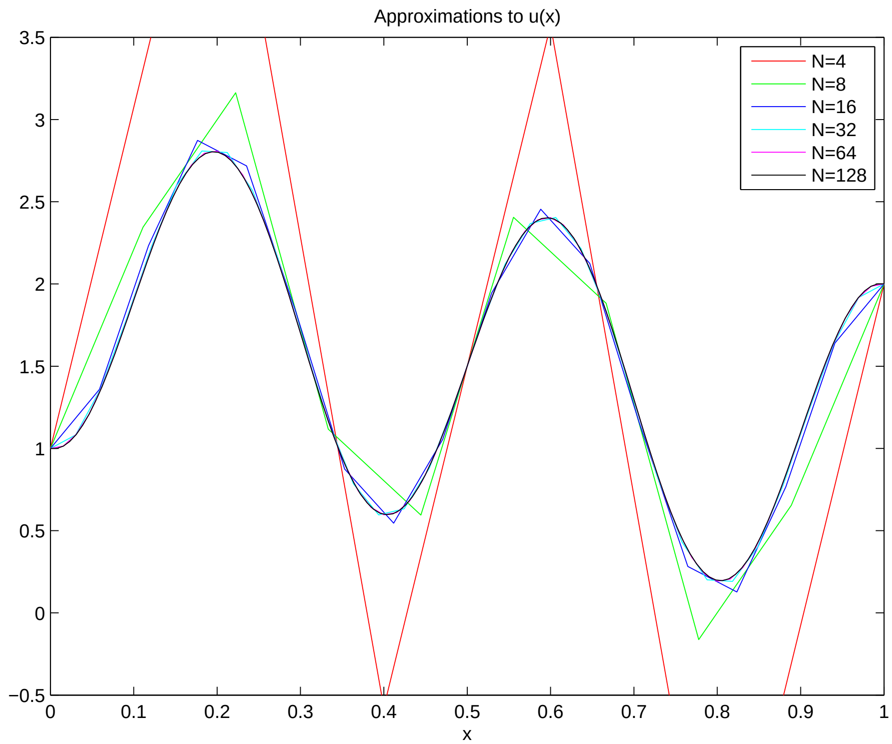
$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} f(x_1) - 1/h^2 \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) - 2/h^2 \end{bmatrix},$$

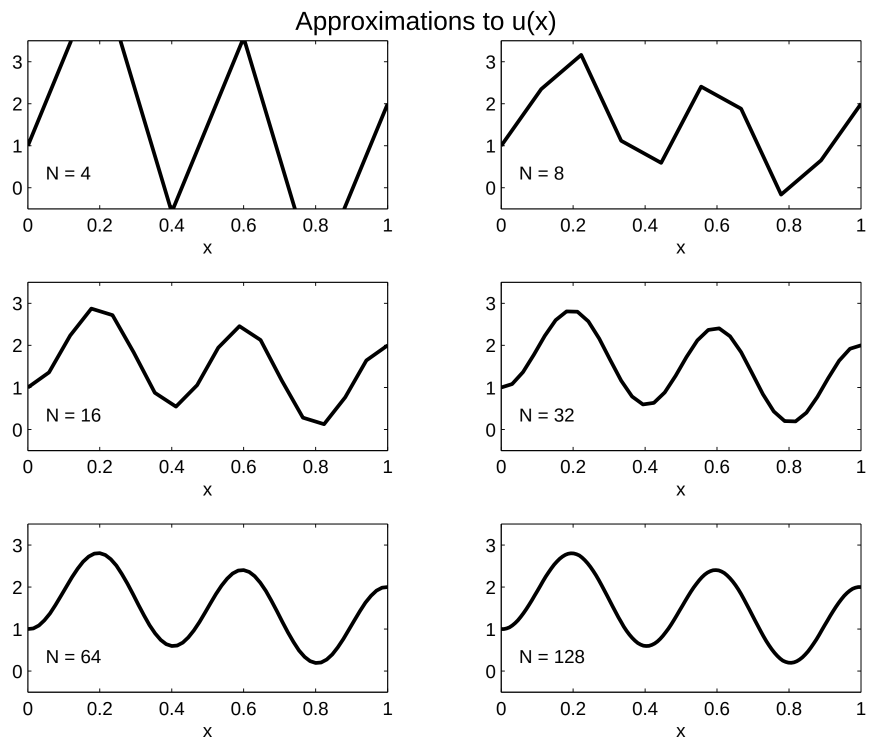
where $u_j \approx u(x_j)$. (Entries of the matrix that are not specified are zero.)

(d) [6 points] The code below produces two acceptable styles of the requested plots, plus a few extra.

```
clear
clc
Nvec = [4 8 16 32 64 128];
colors='rgbcmk';
u0 = 1;
u1 = 2;
figure(1)
clf
suptitle('Approximations to u(x)')
figure(2)
clf
for j=1:length(Nvec)
    N = Nvec(j);
    h = 1/(N+1);
    x = h*(1:N).';
    A = (-2*eye(N)+diag(ones(N-1,1),1)+diag(ones(N-1,1),-1))/(h^2);
    f = 25*pi^2*cos(5*pi*x);
    f(1) = f(1)-u0/(h^2);
    f(N) = f(N)-u1/(h^2);
    u = A\f;
    % plot the function, adding in the inhomogeneous values at the boundary;
    % this tacks on extra entries for the x and u vectors:
    figure(1)
    subplot(3,2,j)
    plot([0;x;1],[u0;u;u1],'k-','linewidth',2)
    hold on
    axis([0 1 -0.5 3.5])
    xlabel('x')
    set(gca,'ytick',0:1:3)
    text(.05,.35,sprintf('N = %d',N))
    figure(2)
    plot([0;x;1],[u0;u;u1],colors(j))
    hold on
    legendStr{j}=['N=' num2str(N)];
end
figure(1)
print -depsc2 separate_d.eps
figure(2)
axis([0 1 -0.5 3.5])
xlabel('x')
title('Approximations to u(x)')
legend(legendStr,'Location','northeast');
print -depsc2 together_d.eps
```

Two acceptable styles of the requested plots, plus a few extra, are below.


Approximations to u(x)
N=4
N=8
N=16
N=32
N=64
N=128
x


Approximations to u(x)
N = 4
N = 8
N = 16
N = 32
N = 64
N = 128
x