CAAM 336 · DIFFERENTIAL EQUATIONS

Homework 2 · Solutions

Posted Wednesday 21, January 2015. Due 5pm Wednesday 28, January 2015.

Please write your name and instructor on your homework.

1. [24 points: (a)-(b) 7 points, (c) 3 points, (d) 7 points]

   (a) In class we considered the 'forward difference' approximation

   $$u'(x) \approx \frac{u(x+h) - u(x)}{h}.$$

   Let $u(x) = \exp(2x)$. For each value $N = 2, 4, 8, 16, \ldots, 512$ (powers of 2), compute (in MATLAB) the error

   $$\left| u'(1/2) - \frac{u(1/2+h) - u(1/2)}{h} \right|,$$

   where $h = 1/(N+1)$. Print out these errors (in a table), and use MATLAB's `loglog` command to produce a plot of N versus the corresponding error. (In class, we showed that this error should be proportional to h as $h \to 0$.)

   (b) Consider the 'centered difference' approximation

   $$u'(x) \approx \frac{u(x+h) - u(x-h)}{2h}.$$

   Repeat part (a) with this approximation: That is, for $u(x) = \exp(2x)$, compute the error

   $$\left| u'(1/2) - \frac{u(1/2+h) - u(1/2-h)}{2h} \right|$$

   for $N = 2, 4, 8, 16, 32, \ldots, 512$ (powers of 2) with $h = 1/(N+1)$. Print out these errors (in a table), and use MATLAB's `loglog` command to produce a plot of N versus the corresponding error. (In class, we showed that this error should be proportional to h^2 as $h \to 0$.)

   Use the `hold on` *command to superimpose the plot for (b) on your plot for part (a): you should only turn in one plot for this problem.*

   (c) By inspecting the plot you have created, estimate the value of N that you need to approximate $u'(1/2)$ to an error of 10^{-2} using the methods in part (a) and part (b).

   (d) We also derived the second-order central approximation to the second derivative

   $$\frac{\partial^2 u(x)}{\partial x^2} \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

   Using the Taylor series expansion for $u(x+h)$

   $$u(x+h) = u(x) + u'(x)h + \frac{u''(x)}{2}h^2 + \frac{u'''(x)}{3!}h^3 + \ldots,$$

   and the equivalent Taylor expansion for $u(x-h)$, show that the second order finite difference approximation

   $$u''(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

   has accuracy $O(h^2)$. In other words, if $u''(x)$ is the exact second derivative, show that

   $$\left| u''(x) - \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \right| = O(h^2).$$

Solution.

(a) The error is shown in the table below. The plot is shown with the solution to part (c). The code that generated this plot follows at the end of the problem.

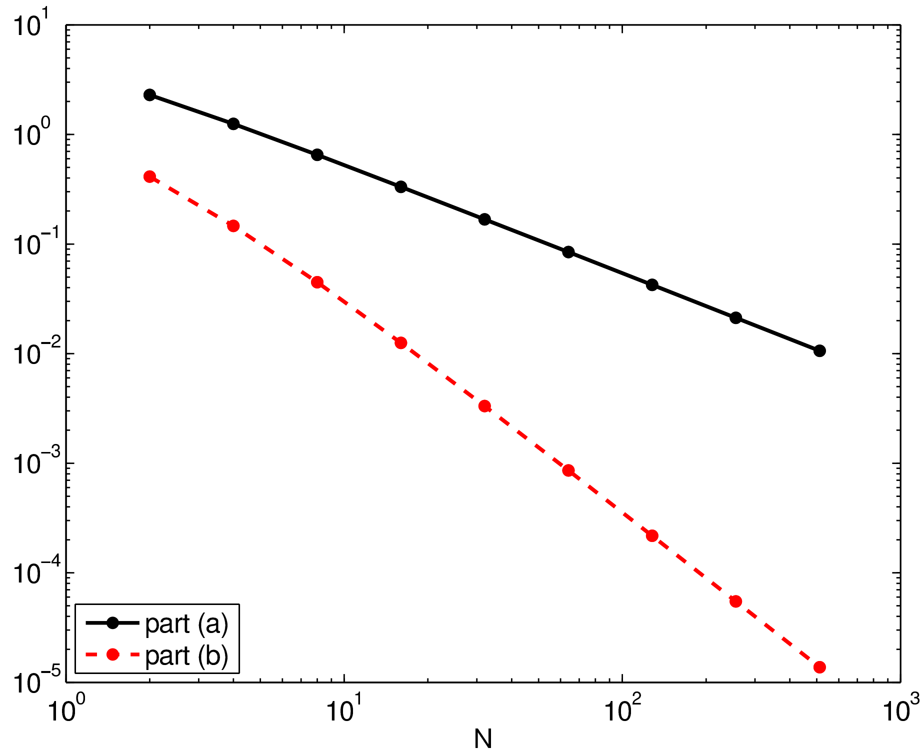
N	error
2	2.2920610
4	1.2480270
8	0.6514086
16	0.3327167
32	0.1681236
64	0.0845039
128	0.0423625
256	0.0212089
512	0.0106114

(b) The $O(h^2)$ centered difference approximation gives

N	error
2	0.4117528
4	0.1461393
8	0.0448560
16	0.0125498
32	0.0033288
64	0.0008579
128	0.0002178
256	0.0000549
512	0.0000138

These errors decay much more rapidly than for the analogous expansion in part (a). This is made clear by the plot below, generated by the following code.

(c) Roughly speaking, the forward difference requires $N \approx 512$ before it is accurate to two digits, while the centered difference only requires $N \approx 16$. (When used in the context of solving differential equations, the improved accuracy of the centered difference formula allows one to work with smaller matrices than required for the forward difference formula, potentially delivering a great speed-up in run-time.)




```
u = inline('exp(2*x)');
uprime = inline('2*exp(2*x)');

Nvec = 2.^[1:9]';
err = zeros(size(Nvec));
x = 1/2;
fprintf('\n part (a)\n')
for k=1:length(Nvec)
   N = Nvec(k);
   h = 1/(N+1);
   deriv = (u(x+h)-u(x))/h;
   err(k) = abs(uprime(x)-deriv);
   fprintf(' %3d   %10.7f\n', N, err(k));
end
loglog(Nvec,err,'k.-','linewidth',2,'markersize',20)

fprintf('\n part (b)\n')
for k=1:length(Nvec)
   N = Nvec(k);
   h = 1/(N+1);
   deriv = (u(x+h)-u(x-h))/(2*h);
   err(k) = abs(uprime(x)-deriv);
   fprintf(' %3d   %10.7f\n', N, err(k));
end
hold on
loglog(Nvec,err,'r--','linewidth',2,'marker','.','markersize',20)
set(gca,'fontsize',14)
xlabel('N', 'fontsize',14)
legend('part (a)','part (b)',3)
print -depsc2 findiff.eps
```

(d) Adding together the Taylor expansions

$$u(x+h) = u(x) + u'(x)h + \frac{u''(x)}{2}h^2 + \frac{u'''(x)}{3!}h^3 + \dots$$
$$u(x-h) = u(x) - u'(x)h + \frac{u''(x)}{2}h^2 - \frac{u'''(x)}{3!}h^3 + \dots$$

gives

$$u(x+h) + u(x-h) = 2u(x) + u''(x)h^2 + 2\frac{u''''(x)}{4!}h^4 + \dots.$$

Subtracting $2u(x)$ from both sides and dividing by h^2 gives

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + 2\frac{u''''(x)}{4!}h^2 + \dots,$$

implying that the truncation error decreases at the same rate that h^2 decreases (if h is small enough). This implies the 2nd order central finite difference formula is $O(h^2)$ accurate.

---

2. [26 points: 9 each]

Suppose $N \geq 1$ is an integer and define $h = 1/(N+1)$ and $x_j = ih$ for $i = 0, \ldots, N+1$. 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 $u_i \approx u(x_i)$. (Entries of the matrix that are not specified are zero.)

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

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

(b) Suppose that we have

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

Since this differential equation is linear, we can split up the solution into

$$u(x) = u_1(x) + u_2(x),$$

where $u_1(x)$ satisfies

$$\begin{aligned} -u_1''(x) &= 0, \quad 0 < x < 1, \\ u_1(0) &= 1 \\ u_1(1) &= 2 \end{aligned}$$

and $u_2(x)$ satisfies the equation

$$\begin{aligned} -u_2''(x) &= (2\pi)^2 \sin(2\pi x), \quad 0 < x < 1, \\ u_2(0) &= 0 \\ u_2(1) &= 0. \end{aligned}$$

Show that $u(x) = u_1(x)+u_2(x)$ satisfies the original differential equation, and determine $u_1(x), u_2(x)$ and the exact solution $u(x)$.

(c) Compute and plot the approximate solutions for $N = 8, 16, 32, 64$, and compare it to the exact solution $u(x)$. On a separate plot, compute the maximum error e_h for a given h

$$e_h = \max_{0 \leq i \leq N+1} |u(x_i) - u_i|$$

and plot $\log(h)$ against $\log(e_h)$ (in class, we showed this line should have slope 2 - you may wish to check this is true by also plotting $\log(h)$ against $2\log(h)$ along with the error. Both the error and this line should have identical slopes).

Solution.

(a) Since boundary conditions are applied at $u(x_0) = u_0$ and $u(x_{N+1}) = u_{N+1}$, they only show up in the finite difference equations for x_1 and x_N. The finite difference equation at x_1 approximates $-u''(x_1) = f(x_1)$ via

$$-\frac{u_2 - 2u_1 + u_0}{h^2} = f_1.$$

Since $u_0 = u(x_0) = 1$ is known, we can modify the above equation to be

$$-\frac{u_2 - 2u_1}{h^2} = f_1 + \frac{1}{h^2}.$$

Similarly, at $u_N = u(x_N)$, we approximate $-u''(x_N) = f(x_N)$ via

$$-\frac{u_{N+1} - 2u_N + u_{N-1}}{h^2} = f_1.$$

Since $u_{N+1} = u(x_{N+1}) = 2$ is known, we can modify the above equation to be

$$-\frac{-2u_N + u_{N-1}}{h^2} = f_1 + \frac{2}{h^2}.$$

This leads to the system of equations

$$\frac{-1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 + \frac{1}{h^2} \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N + \frac{2}{h^2} \end{bmatrix},$$

(b) Since $-u_1''(x) = 0$, we know that u_1 should be a linear polynomial, or that

$$u_1(x) = ax + b.$$

Boundary conditions then give

$$u(0) = 1 = b, \qquad u(1) = 2 = a + b$$

or that $a = 1$, $b = 1$, and $u_1 = x + 1$.
To solve $-u_2''(x) = (2\pi)^2 \sin(2\pi x)$ with zero boundary conditions, we can note that $\sin(2\pi x)$ satisfies zero boundary conditions, and then observe that taking the negative of two derivatives of $\sin(2\pi x)$ gives back

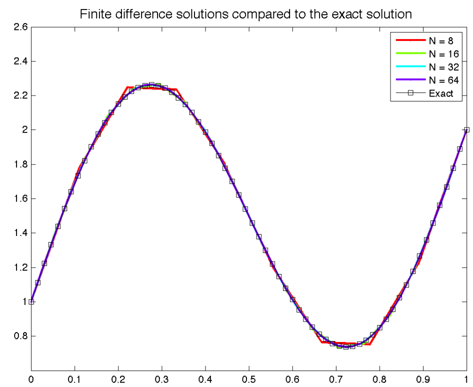
$$-\frac{\partial^2 \sin(2\pi x)}{\partial x^2} = (2\pi^2)\sin(2\pi x).$$

This implies that $u(x) = \sin(2\pi x)$ satisfies both the boundary conditions and the differential equation with inhomogenous source term.

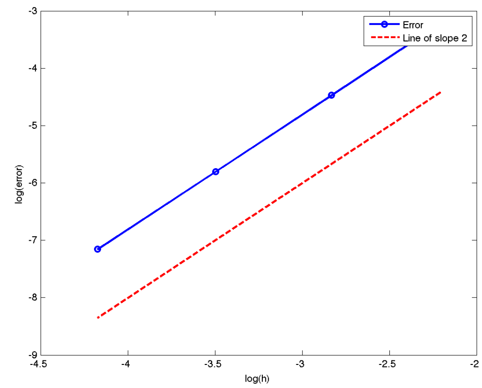
(c) Included is Matlab code that can be used to generate the finite difference solution, exact solution, and the error between it and the exact solution.

Graders: please do not take off if the students did not plot the error — we only asked for a comparison of the exact solution to the computed solutions.

```matlab
% HW 2, Problem 2c. CAAM 336, Fall 2014
% solves the steady heat equation u''(x) = (2 pi)^2 sin(2 pi x)
% with u(0) = 1, u(1) = 2
clear

uexact = @(x) sin(2*pi*x) + x + 1;

i = 1;
C = hsv(4); % neat trick: makes a matrix whose values determine colors.
Nlist = [8 16 32 64]; % number of interior points
for N = Nlist
    K = N+1; % number of line segments
    h = 1/K; % spacing between points
    x = (0:N+1)/(N+1);
    x = x(:); % makes x a column vector.

    A = -2*diag(ones(N,1)) + diag(ones(N-1,1),1) + diag(ones(N-1,1),-1);
    A = -A/h^2;
    b = (2*pi)^2*sin(2*pi*x(2:end-1));
    b(1) = b(1) + 1/h^2; % modify b for inhomogeneous BCs
    b(N) = b(N) + 2/h^2; % modify b for inhomogeneous BCs
    u = A\b;

    figure(1)
    plot(x,[1;u;2],'.-','color',C(i,:),'linewidth',2);
    hold on

    err(i) = max(abs(uexact(x)-[1;u;2]));
    hvec(i) = h;
    i = i+1;
end
figure(1)
title('Finite difference solutions compared to the exact solution','fontsize',14)
plot(x,uexact(x),'ks-')
legend('N = 8','N = 16','N = 32','N = 64','Exact')
print(gcf,'-dpng','p2c_sol') % print out graphs to file

figure(2)
title('Error between finite difference and exact solutions','fontsize',14)
plot(log(hvec),log(err),'o-','linewidth',2);hold on
plot(log(hvec),2*log(hvec),'r--','linewidth',2);hold on
xlabel('log(h)')
ylabel('log(error)')
legend('Error','Line of slope 2')
print(gcf,'-dpng','p2c_error') % print out graphs to file
```



(a) Finite difference solutions for various N



(b) Error between the exact solution and finite difference solution at points x_i.

3. [26 points: 8 each (a),(b), 10 for (c)]

Recall the 1D steady-state heat equation with constant diffusivity over the interval $[0,1]$

$$-\frac{\partial^2 u}{\partial x^2} = f$$
$$u(0) = u(1) = 0.$$

Recall from class the finite difference approximation to this problem: given a set of points $x_0, \ldots, x_{N+1}$, solved for the solution $u(x_i)$ at each point by approximating $\frac{\partial^2 u}{\partial x^2}$ with

$$u''(x_i) \approx \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2}, \quad i = 1, \ldots, N$$

(where h is the spacing between points x_{i+1} and x_i) along with the conditions that

$$u(x_0) = u(x_{N+1}) = 0.$$

We will modify this finite difference approximation to accomodate instead the Neumann boundary condition of $u'(1) = 0$ at $x = 1$.

(a) We would like to enforce that $u'(x_{N+1}) = 0$, but if we approximate $u'(x_{N+1})$ with a central difference

$$u'(x_{N+1}) \approx \frac{u(x_{N+\frac{3}{2}}) - u(x_{N+\frac{1}{2}})}{h},$$

we end up with an equation involving $u(x_{N+\frac{3}{2}})$, which does not lie inside the interval $[0,1]$. Instead, we can define a *backward difference* approximation to the derivative

$$u'(x_{N+1}) \approx \frac{u(x_{N+1}) - u(x_N)}{h} = 0$$

and set this to zero instead. Write out the expression for $u''(x_N)$ in terms of $u(x_i)$ and use the backward difference approximation for $u'(x_{N+1})$ to eliminate $u(x_{N+1})$.

(b) Determine the exact solution to $-u''(x) = 1$ for $u(0) = 0$, $u'(1) = 0$ (hint: the solution is a quadratic function).

(c) Create a MATLAB script that constructs the matrix system $Au = f$ resulting from the finite difference equations when $f = 1$. Plot the computed solution values $u(x_i)$ for $i = 0, \ldots, N+1$ for $N = 16, 32, 64, 128$. On a separate plot, compute the maximum error e_h for a given h

$$e_h = \max_{0 \le i \le N+1} |u(x_i) - u_i|$$

and plot $\log(h)$ against $\log(e_h)$. Does the error decrease faster or slower compared to the error computed in Problem 2? Can you explain why?

Solution.

(a) We can rewrite the finite difference approximation to $-u''(x) = f(x)$ at point x_N as

$$-\frac{u(x_{N-1}) - 2u(x_N) + u(x_{N+1})}{h^2} = f(x_N).$$

Note that

$$\frac{u(x_{N-1}) - 2u(x_N) + u(x_{N+1})}{h^2} = \frac{u(x_{N-1}) - u(x_N)}{h^2} + \frac{u(x_{N+1}) - u(x_N)}{h^2}$$

Using the backwards difference approximation to $u'(x_{N+1})$, we have

$$\frac{u(x_{N+1}) - u(x_N)}{h^2} = 0$$

which simplifies our finite difference equation at x_N to

$$-\frac{u(x_{N-1}) - u(x_N)}{h^2} = f(x_N).$$

(Note that the boundary condition also implies $x_N = x_{N+1}$).

(b) We can integrate the differential equation twice to get the boundary conditions.

$$\int_0^x -u''(s)ds = \int_0^x 1ds$$

where s is a dummy variable for integration. By the fundamental theorem of calculus, this gives

$$-u'(x) + u'(0) = x.$$

Since we don't know the value of $u'(0)$, we consider it an unknown constant C_1 that we have to determine using our boundary conditions. Repeating the process again gives

$$\int_0^x (-u'(s) + C_1)ds = \int_0^x xds$$

which results in

$$-u(x) + C_1 x + u(0) = \frac{x^2}{2}.$$

We could set $u(0)$ to be a constant C_2 to be determined by the boundary conditions as well; however, since we know $u(0) = 0$ from the boundary conditions, we can go ahead and zero it out. The end result gives

$$u(x) = -\frac{x^2}{2} + C_1 x$$

The above form of the equation and the boundary condition $u'(1) = 0$ give the condition that

$$u'(1) = -1 + C_1 = 0$$

implying $C_1 = 1$, and

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

Alternatively, since the problem specifies the solution is a quadratic, it is possible to simply specify

$$u(x) = ax^2 + bx + c$$

and use the differential equation and boundary conditions to determine the constants.

(c) Since the finite difference equations must be satisfied at each point x_i, they lead to a series of N equations with N unknowns (the values of $u(x_i)$ for $i = 1, \ldots, N$). The matrix system resulting from these equations for homogeneous boundary conditions

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

is

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

where $u_i \approx u(x_i)$. Since we have the boundary condition $u'(1) = 0$ instead, this changes our finite difference equation at point x_N, which corresponds to the final row of our matrix. Thus, our new matrix system for a no-flux boundary condition at $x = 1$ will be

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

We observe in the solution that the error converges more slowly, at a rate of $O(h)$ instead of the $O(h^2)$ observed before. This is because we've mixed in an $O(h)$ backwards difference approximation into our finite difference equations — the error is largest at the endpoint, where we applied the backwards difference approximation as a boundary condition.

Included is Matlab code that can be used to generate the finite difference solution and the error between it and the exact solution:

```
% HW 2, Problem 3c. CAAM 336, Fall 2014
% solves the steady heat equation u''(x) = 1 with u(0) = 0, u'(1) = 0
clear

uexact = @(x) .5*x.*(2-x);

i = 1;
C = hsv(4); % neat trick: makes a matrix whose values determine colors.
Nlist = [16 32 64 128]; % number of interior points
for N = Nlist
    K = N+1; % number of line segments
    h = 1/K; % spacing between points
    x = (1:N)*h; x = x(:);

    A = -2*diag(ones(N,1)) + diag(ones(N-1,1),1) + diag(ones(N-1,1),-1);
    A(N,N-1:N) = [1 -1]; % modify last row of matrix for no-flux boundary condition
    A = -A/h^2;
    b = ones(N,1); % f(x) = 1
    u = A\b;

    figure(1)
    x = [0;x;1];
    plot(x,[0;u;u(N)],'.-','color',C(i,:),'linewidth',3);
    hold on % append value at x(N+1) = x(N)

    err(i) = max(abs(uexact(x)-[0;u;u(N)]));
```

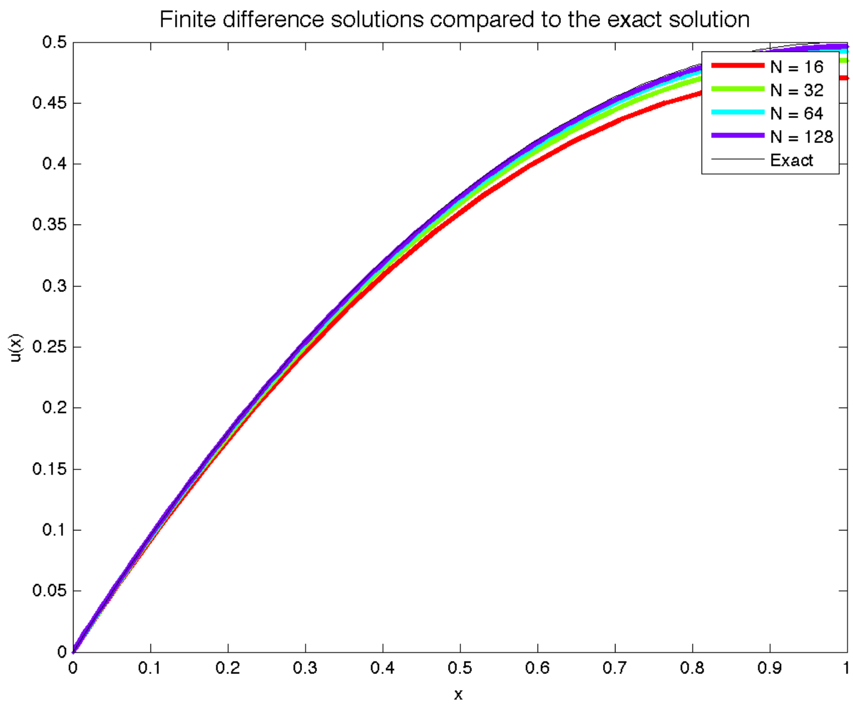


Figure 1: Finite difference solutions for various N

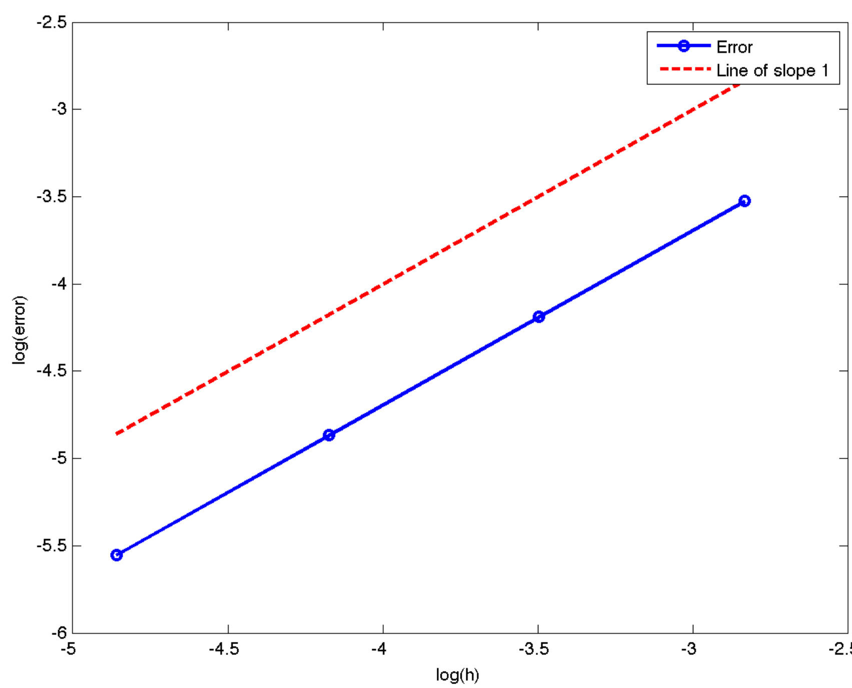


Figure 2: Error between the exact solution and finite difference solution at points x_i.

```
    i = i+1;
end
figure(1)
title('Finite difference solutions compared to the exact solution','fontsize',14)
plot(x,uexact(x),'k-')
xlabel('x');ylabel('u(x)')
legend('N = 16','N = 32','N = 64', 'N = 128','Exact')
print(gcf,'-dpng','p3c_sol') % print out graphs to file

figure(2)
title('Error between finite difference and exact solutions','fontsize',14)
plot(log(Nlist),log(err),'o-');hold on
xlabel('log(N)')
ylabel('log(error)')
print(gcf,'-dpng','p3c_error') % print out graphs to file
```

4. [24 points: 4 each]

(a) Demonstrate whether or not the set $S_1 = \left\{\mathbf{x} \in \mathbb{R}^2 : \ x_2 = x_1^3\right\}$ is a subspace of $\mathbb{R}^2$.

(b) Demonstrate whether or not the set $S_2 = \left\{\mathbf{x} \in \mathbb{R}^3 : \ 3x_1 + 2x_2 + x_3 = 0\right\}$ is a subspace of $\mathbb{R}^3$.

(c) Demonstrate whether or not the set $S_3 = \{f \in C[0,1] : \ f(x) \geq 0 \text{ for all } x \in [0,1]\}$ is a subspace of $C[0,1]$.

(d) Demonstrate whether or not the set $S_4 = \left\{f \in C[0,1] : \ \max_{x\in[0,1]} f(x) \leq 1\right\}$ is a subspace of $C[0,1]$.

(e) Demonstrate whether or not the set $S_5 = \left\{f \in C^2[0,1] : \ f(1) = 1\right\}$ is a subspace of $C^2[0,1]$.

(f) Demonstrate whether or not the set $S_6 = \left\{f \in C^2[0,1] : \ f(1) = 0\right\}$ is a subspace of $C^2[0,1]$.

Solution.

(a) [4 points] The set S_1 is not a subspace of $\mathbb{R}^2$.
The vector $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is in the set S_1, yet $2\mathbf{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ is not, since $2 \neq 2^3 = 8$. Consequently, the set S_1 is not a subspace of $\mathbb{R}^2$.

(b) [4 points] The set S_2 is a subspace of $\mathbb{R}^3$.
The set S_2 is a subset of $\mathbb{R}^3$ and $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ is a member of the set S_2. Now, suppose $\mathbf{x}$ and $\mathbf{y}$ are members of the set S_2. Then $3x_1 + 2x_2 + x_3 = 0$ and $3y_1 + 2y_2 + y_3 = 0$. Adding these two equations gives

$$3(x_1 + y_1) + 2(x_2 + y_2) + (x_3 + y_3) = 0,$$

and hence $\mathbf{x} + \mathbf{y}$ is also in the set S_2. Multiplying $3x_1 + 2x_2 + x_3 = 0$ by an arbitrary constant $\alpha \in \mathbb{R}$ gives

$$3(\alpha x_1) + 2(\alpha x_2) + \alpha x_3 = 0,$$

and hence $\alpha\mathbf{x}$ is also in the set S_2. Consequently, the set S_2 is a subspace of $\mathbb{R}^3$.

(c) [4 points] The set S_3 is not a subspace of $C[0,1]$.
Let $f(x) = 1$ for $x \in [0,1]$. Then f is in the set S_3, but a scalar multiple, $-1 \cdot f(x) = -1$ for $x \in [0,1]$, takes negative values and thus violates the requirement for membership in the set S_3. Consequently, the set S_3 is not a subspace of $C[0,1]$.

(d) [4 points] The set S_4 is not a subspace of $C[0,1]$.
Let $f(x) = 1$ for $x \in [0,1]$. Then f is in the set S_4, but a scalar multiple, $2 \cdot f(x) = 2$ for $x \in [0,1]$, takes values greater than one and thus violates the requirement for membership in the set S_4. Consequently, the set S_4 is not a subspace of $C[0,1]$.

(e) [4 points] The set S_5 *is not* a subspace of $C^2[0,1]$.
The function z defined by $z(x) = 0$ for $x \in [0,1]$ is not in the set S_5 since $z(1) = 0$ and thus violates the requirement for membership in the set S_5. Consequently, the set S_5 is not a subspace of $C^2[0,1]$.

(f) [5 points] The set S_6 subspace of $C^2[0,1]$.
The set S_6 is a subset of $C^2[0,1]$ and the function z defined by $z(x) = 0$ for $x \in [0,1]$ is in the set S_6. If f and g are in the set S_6, then $f(1) = g(1) = 0$, so

$$(f+g)(1) = f(1) + g(1) = 0 + 0 = 0$$

and hence $f+g$ is in the set S_6. Also, if f is in the set S_6 and $\alpha \in \mathbb{R}$, then

$$(\alpha f)(1) = \alpha f(1) = \alpha \cdot 0 = 0$$

and hence αf is in the set S_6. Consequently, the set S_6 is a subspace of $C^2[0,1]$.
