Solution. GRADERS: Please record the scores in percentages, rounded up. I.e. 56 out of 60 points would be entered as a 94 (percent)

A reminder from the course syllabus: Mathematically rigorous solutions are expected; strive for clarity and elegance.

You may collaborate on the problems, but your write-up must be your own independent work. Transcribed solutions and copied MATLAB code are both unacceptable. You may not consult solutions from previous sections of this class. Unless it is specified that a particular calculation must be performed ‘by hand,’ you are encouraged to use MATLAB’s Symbolic Math Toolbox (or Mathematica/Wolfram Alpha/Maple) to compute and simplify tedious integrals and derivatives on the problem sets. As always, you must document your calculations clearly.

1. [12 pts (2 ea)]

For each of the following equations, specify whether each is (a) an ODE or a PDE; (b) determine its order; (c) specify whether it is linear or nonlinear. For those that are linear, specify whether they are (d) homogeneous or inhomogeneous, and (e) whether they have constant or variable coefficients.

(1.1) \( \frac{dv}{dx} + \frac{2}{x}v = 0 \)

(1.2) \( \frac{\partial v}{\partial t} - 3 \frac{\partial v}{\partial x} = x - t \)

(1.3) \( \frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left[ 2u \frac{\partial u}{\partial x} \right] = 0 \)

(1.4) \( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0 \)

(1.5) \( \frac{d^2 y}{dx^2} - \mu (1 - y^2) \frac{dy}{dx} + y = 0 \)

(1.6) \( \rho(x) \frac{d^2 u}{dx^2} = \sin(x) \)

Solution.

(1.1) ODE, first order, linear, homogeneous, variable coefficient
The \( 2/x \) factor in front of the \( v \) is the variable coefficient.

(1.2) PDE, first order, linear, inhomogeneous, constant coefficient
The \( x - t \) term on the right, which does not involve \( v \), makes the equation inhomogeneous.

(1.3) PDE, second order, nonlinear
Using the product rule for partial derivatives, we can write this equation in the equivalent form

\[ \frac{\partial u}{\partial t} - 2 \left( \frac{\partial u}{\partial x} \right)^2 - 2u \left( \frac{\partial^2 u}{\partial x^2} \right) = 0. \]

The second and third terms on the left hand side make this equation nonlinear.

(1.4) PDE, third order, nonlinear
The \( u(\partial u/\partial x) \) term makes this equation nonlinear. This a version of the famous Korteweg-de Vries (KdV) equation that describes shallow water waves.

(1.5) ODE, second order, nonlinear
The \( (1 - y^2)(dy/dt) \) term makes this ODE nonlinear.
(1.6) ODE, fourth order, linear, inhomogeneous, variable coefficient

Using the product rule for partial derivatives, we can write this equation in the equivalent form

\[
\frac{d^2 \rho}{dx^2} \frac{d^3 u}{dx^3} + 2 \frac{d \rho}{dx} \frac{d^3 u}{dx^3} + \rho(x) \frac{d^4 u}{dx^4} = \sin(x),
\]

hence we can see that it is fourth order. This equation, attributed to Euler, describes the deflection of a one-dimensional beam with a static load of \(\sin(x)\); \(\rho(x)\) describes the elasticity of the material that constitutes the beam.

2. [21 pts (7 ea)]

Consider the temperature function

\[
u(x, t) = e^{-\kappa \theta^2 t/(\rho c)} \sin(\theta x)\]

for constant \(\kappa\), \(\rho\), \(c\), and \(\theta\).

(a) Show that this function \(u(x, t)\) is a solution of the homogeneous heat equation

\[\rho c \frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2},\]

for \(0 < x < \ell\) and all \(t\).

(b) For which values of \(\theta\) will \(u\) satisfy homogeneous Dirichlet boundary conditions at \(x = 0\) and \(x = \ell\)?

(c) Suppose \(\kappa = 2.37\) W/(cm K), \(\rho = 2.70\) g/cm\(^3\), and \(c = 0.897\) J/(g K) (approximate values for aluminum found on Wikipedia), and that the bar has length \(\ell = 10\) cm. Let \(\theta\) be such that \(u(x, t)\) satisfies homogeneous Dirichlet boundary conditions as in part (b) and \(u(x, t) \geq 0\) for all \(x\) and \(t\).

Use MATLAB to plot the solution \(u(x, t)\) for \(0 \leq x \leq \ell\) and time \(0 \leq t \leq 20\) sec.

You may choose to do this in one of the following ways: (1) Plot the solution for \(0 \leq x \leq \ell\) at times \(t = 0, 4, 8, \ldots, 20\) sec., superimposing all six plots on the same axis (helpful commands: \texttt{linspace}, \texttt{plot}, \texttt{hold on}); (2) Create a three-dimensional plot of the data using \texttt{surf}, \texttt{mesh}, or \texttt{waterfall}. In either case, be sure to produce an attractive, well-labeled plot.

Solution.

(a) We compute

\[
\frac{\partial u}{\partial t} = -\kappa \theta^2/(\rho c) e^{-\kappa \theta^2 t/(\rho c)} \sin(\theta x)
\]

\[
\frac{\partial u}{\partial x} = \theta e^{-\kappa \theta^2 t/(\rho c)} \cos(\theta x)
\]

\[
\frac{\partial^2 u}{\partial x^2} = -\theta^2 e^{-\kappa \theta^2 t/(\rho c)} \sin(\theta x).
\]

With these formulas in hand it is easy to verify that

\[\rho c \frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}.
\]

(b) We wish to find the values of \(\theta\) that give homogeneous Dirichlet boundary conditions, i.e., \(u(0, t) = u(\ell, t) = 0\) for all \(t\). Since \(e^{-\kappa \theta^2 t/(\rho c)}\) is positive for all \(t\), we can only get the homogeneous Dirichlet
conditions when \( \sin(\theta x) = 0 \). For any \( \theta \), \( \sin(\theta \cdot 0) = 0 \), so the condition at \( x = 0 \) is automatically satisfied. To get \( \sin(\theta \ell) = 0 \), we need \( \theta \ell \) to be an integer multiple of \( \pi \), that is,

\[
\theta \ell = \pi n, \quad n = 0, \pm 1, \pm 2, \ldots,
\]
or equivalently

\[
\theta = \frac{\pi n}{\ell}, \quad n = 0, \pm 1, \pm 2, \ldots
\]

(Notice that if \( n = 0 \) we have the trivial solution \( u(x, t) = 0 \) for all \( x, t \). If \( n = 1 \), we have a solution for which \( u(x, t) \geq 0 \) for all \( x, t \). For other values of \( n \) the solution will be negative for some \( x, t \). If our temperature is measured in Kelvin this could be a problem! However, this heat equation takes the same form if we shift to Celsius units, so we needn’t be so troubled by the negative values of temperature.)

Since \( n = 0 \) is trivial, we shall take \( n = 1 \) (\( \theta = \pi / \ell \)) to obtain

\[
u(x, t) = e^{-\kappa \pi^2 t / (\ell^2 \rho c)} \sin(\pi x / \ell)
\]

\[
= e^{-2.37 \pi^2 t / (100 \cdot 2.70 \cdot 0.897)} \sin(\pi x / 10).
\]

(c) Solutions are shown in the attached plots. Any of these style is acceptable. The MATLAB code that generated these plots follows.

[GRADERS: please make a note if students did not include their MATLAB code, but do not take off points for this first time. You should do so in the future, though!]
MATLAB code:

c = .897;
kappa = 2.37;
rho = 2.70;
l = 10;
theta = pi/l;

% first style: snapshots at t = 0, 4, 8, ..., 20

    t = 0:4:20;
x = linspace(0,l,100);

    figure(1), clf
    for j=1:length(t)
        u = exp(-kappa*theta^2*t(j)/(rho*c))*sin(theta*x); % compute u(:,t(j))
        plot(x,u,'k-','linewidth',2), hold on
        text(4.75, max(u)-.03, sprintf('t = %d', t(j))
    end
    axis([0 10 0 1.1])
    set(gca,'fontsize',14)
    xlabel('x')
    ylabel('u(x,t)')
    print -depsc2 checksol1

% generate data for 3-d plots

    x = linspace(0,l,100);
t = linspace(0,20,50);
    U = zeros(length(t), length(x));
    for j=1:length(t)
        U(j,:) = exp(-kappa*theta^2*t(j)/(rho*c))*sin(theta*x);
    end

% mesh plot
    figure(2), clf
    mesh(x,t,U,'edgecolor','k')
    view(-55,20)
    set(gca,'fontsize',14)
    xlabel('x'), ylabel('t'), zlabel('u(x,t)')
    print -depsc2 checksol2

% surf plot
    figure(3), clf
    surf(x,t,U)
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

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

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

$$-u''(x) = (2\pi)^2 \sin(2\pi x), \quad 0 < x < 1,$$

$$u(0) = 1$$

$$u(1) = 2.$$

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

$$-u''_1(x) = 0, \quad 0 < x < 1,$$

$$u_1(0) = 1$$

$$u_1(1) = 2.$$
and \( u_2(x) \) satisfies the equation
\[
-u_2''(x) = (2\pi)^2 \sin(2\pi x), \quad 0 < x < 1, \\
u_2(0) = 0 \\
u_2(1) = 0.
\]

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

(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
\[
u(0) = 1 = b, \quad u(1) = 2 = a + b
\]
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.

% HW 2, Problem 2c. CAAM 336, Fall 2016
% 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
(a) Finite difference solutions for various $N$

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

```matlab
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
```

4. [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 accommodate 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

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

we end up with an equation involving $u(x_{N+\frac{1}{2}})$, 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 \(h\)

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

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:

```matlab
% HW 2, Problem 3c. CAAM 336, Fall 2016
% 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
```
Figure 1: Finite difference solutions for various $N$

Figure 2: Error between the exact solution and finite difference solution at points $x_i$. 
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)]));
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-')
xlabel('log(N)')
ylabel('log(error)')
print(gcf,'-dpng','p3c_error') % print out graphs to file

5. [24 points: 4 each]

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

(b) Demonstrate whether or not the set $S_2 = \{ x \in \mathbb{R}^3 : 3x_1 + 2x_2 + x_3 = 0 \}$ 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 $S_5 = \{ f \in C^2[0,1] : f(1) = 1 \}$ is a subspace of $C^2[0,1]$.

(f) Demonstrate whether or not the set $S_6 = \{ f \in C^2[0,1] : f(1) = 0 \}$ 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 \( x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \) is in the set \( S_1 \), yet \( 2x = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \) 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 \( x \) and \( y \) 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 \( x + y \) 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 \( \alpha x \) 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 \( \alpha f \) is in the set \( S_6 \). Consequently, the set \( S_6 \) is a subspace of \( C^2[0,1] \).