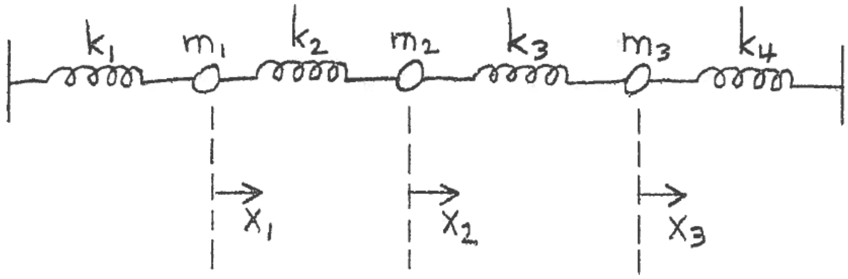
Practice Questions for Final on Wed May 6 12:30-2:30 PM on Moodle

1)



In the mass-spring system shown above, the masses m_1, m_2 and m_3 are .8, .6 and .5, the spring constants k_1, k_2, k_3 and k_4 are 4.3, 5.1, 4.6 and 5.4, and x_1, x_2 and x_3 are the displacements of m_1, m_2 and m_3 from their equilibrium positions.

Write a MATLAB program as follows:

1) t will go from 0 to 8 sec in steps of .001 sec.
2) Calculate the displacements and velocities of the masses for each value of t. Use 1e-7 as the accuracy factors, .7, .2 and .4 as the initial values of x_1, x_2 and x_3 , and 0 as the initial values of the velocities.
3) Plot x_1, x_2 and x_3 versus t using the colors blue, red and green and the t axis in black.
4) In a separate figure, plot the velocities v_1, v_2 and v_3 versus t using the colors blue, red and green and the t axis in black.

<u>Just write the figure statements and the plot statements. Do not write any other statements for the graphs.</u>

This program has a function defined in a separate MATLAB file. Name this function prog1f.

<u>Write both the main program and the function</u>.

<u>Equations</u>

$$m_1 \frac{d^2 x_1}{dt^2} = -k_1 x_1 + k_2(x_2 - x_1)$$

$$m_2 \frac{d^2 x_2}{dt^2} = -k_2(x_2 - x_1) + k_3(x_3 - x_2)$$

$$m_3 \frac{d^2 x_3}{dt^2} = -k_3(x_3 - x_2) - k_4 x_3$$

The answer is on the next page.

Problem 1 Answer

Before writing the program for Problem 1, convert each 2nd order differential equation into 2 1st order differential equations:

$$\frac{dx_1}{dt} = v_1$$

$$\frac{dv_1}{dt} = \frac{1}{m_1}\left(-k_1 x_1 \qquad + k_2(x_2 - x_1)\right)$$

$$\frac{dx_2}{dt} = v_2$$

$$\frac{dv_2}{dt} = \frac{1}{m_2}\left(-k_2(x_2 - x_1) + k_3(x_3 - x_2)\right)$$

$$\frac{dx_3}{dt} = v_3$$

$$\frac{dv_3}{dt} = \frac{1}{m_3}\left(-k_3(x_3 - x_2) - k_4 x_3 \qquad \right)$$

Then Write the Program for Problem 1:

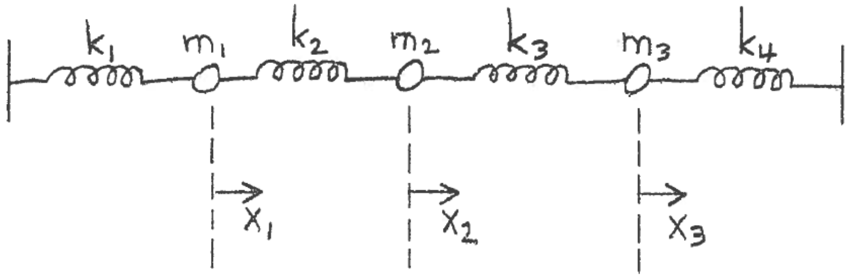
```
% main program
t=0 : .001 : 8;
u0 = [.7  0  .2  0  .4  0];
options = odeset('AbsTol', 1e-7, 'RelTol', 1e-7);
[t  u] = ode45('prog1f', t, u0, options);
line1x = [0  8];
line1y = [0  0];
figure(1);
plot(t,u(:,1),'b',t,u(:,3),'r',t,u(:,5),'g',line1x,line1y,'k');
figure(2);
plot(t,u(:,2),'b',t,u(:,4),'r',t,u(:,6),'g',line1x,line1y,'k');
```

The function for Problem 1 is on the next page.

Problem 1 Function

```
% function prog1f
function f = prog1f(t,uf)
m1 = .8;
m2 = .6;
m3 = .5;
k1 = 4.3;
k2 = 5.1;
k3 = 4.6;
k4 = 5.4;
x1 = uf(1);
v1 = uf(2);
x2 = uf(3);
v2 = uf(4);
x3 = uf(5);
v3 = uf(6);
f = zeros(6,1);
f(1) = v1;
f(2) = 1/m1*( -k1*x1      + k2*(x2-x1) );
f(3) = v2;
f(4) = 1/m2*( -k2*(x2-x1) + k3*(x3-x2) );
f(5) = v3;
f(6) = 1/m3*( -k3*(x3-x2) - k4*x3      );
```

2)



In the mass-spring system shown above, the masses m_1, m_2 and m_3 are .8, .6 and .5, the spring constants k_1, k_2, k_3 and k_4 are 4.3, 5.1, 4.6 and 5.4, and x_1, x_2 and x_3 are the displacements of m_1, m_2 and m_3 from their equilibrium positions.

Write a MATLAB program as follows:

1) t will go from 0 to 8 sec in steps of .001 sec.

2) For each of the 3 natural frequencies, plot x_1, x_2 and x_3 versus t using the colors blue, red and green and the t axis in black (there will be 3 figures). Only Figure 3 is shown below, but the program must plot all 3 figures (Figures 1-3) and the title of each figure must have a different figure number. The horizontal and vertical axes of all 3 figures should look like the ones on Figure 3.

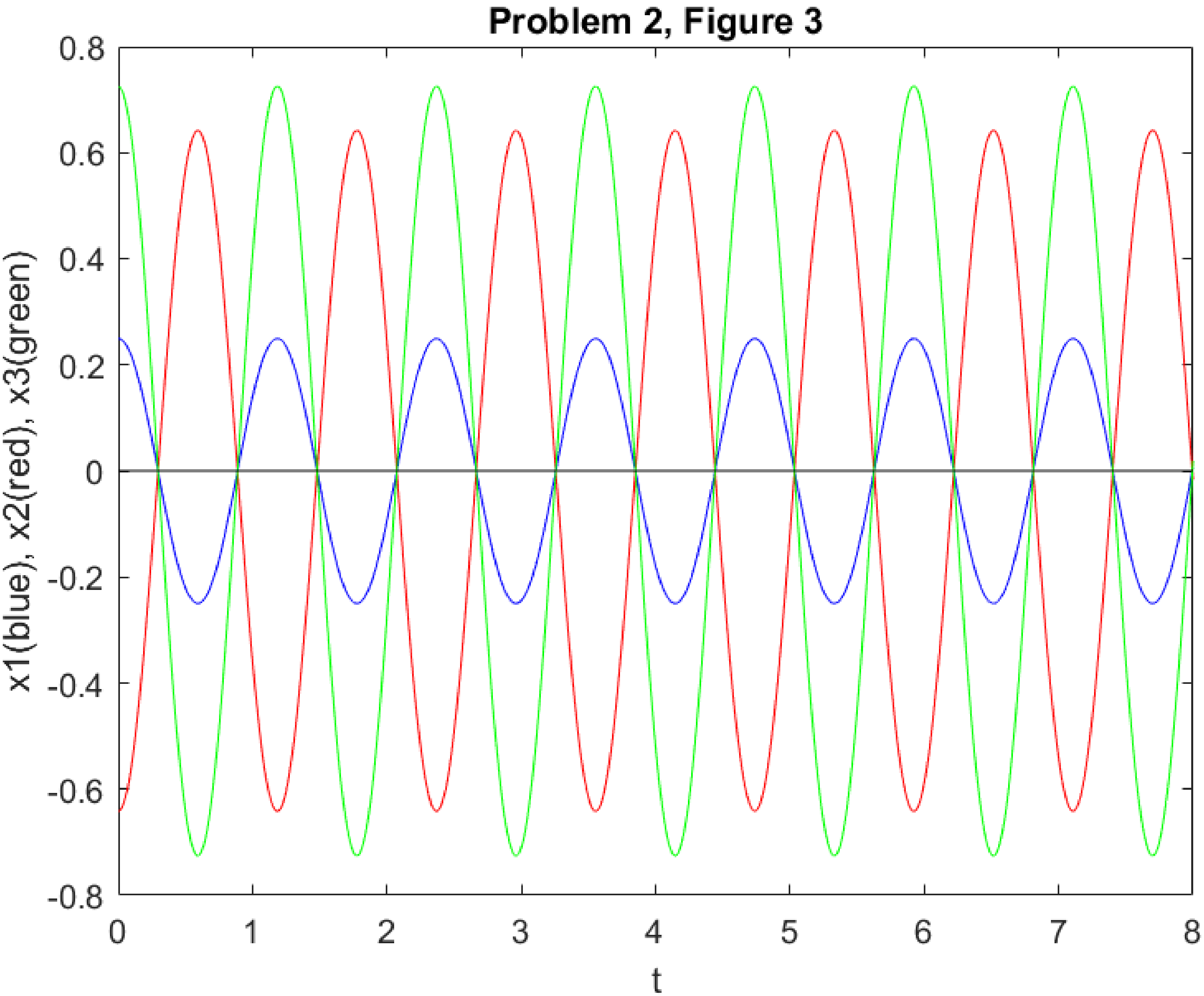
Equations

$$m_1 \frac{d^2 x_1}{dt^2} = -k_1 x_1 \qquad + k_2 (x_2 - x_1)$$

$$m_2 \frac{d^2 x_2}{dt^2} = -k_2 (x_2 - x_1) + k_3 (x_3 - x_2)$$

$$m_3 \frac{d^2 x_3}{dt^2} = -k_3 (x_3 - x_2) - k_4 x_3$$

Figure 3 is on the next page.



The answer is on the next page.

Problem 2 Answer

Do These Steps Before Writing the Program for Problem 2:

1) Rearrange the right sides of the equations by collecting the terms that multiply x_1, x_2 and x_3:

$$m_1 \frac{d^2 x_1}{dt^2} = -(k_1 + k_2) x_1 + k_2 x_2$$

$$m_2 \frac{d^2 x_2}{dt^2} = k_2 x_1 - (k_2 + k_3) x_2 + k_3 x_3$$

$$m_3 \frac{d^2 x_3}{dt^2} = k_3 x_2 - (k_3 + k_4) x_3$$

2) Divide both sides of the equations by the masses:

$$\frac{d^2 x_1}{dt^2} = \frac{1}{m_1} \left(-(k_1 + k_2) x_1 + k_2 x_2 \right)$$

$$\frac{d^2 x_2}{dt^2} = \frac{1}{m_2} \left(k_2 x_1 - (k_2 + k_3) x_2 + k_3 x_3 \right)$$

$$\frac{d^2 x_3}{dt^2} = \frac{1}{m_3} \left(k_3 x_2 - (k_3 + k_4) x_3 \right)$$

3) Define the matrix A that contains the coefficients of x_1, x_2 and x_3 in the equations. The first column of the matrix A contains the coefficients of x_1, the second column contains the coefficients of x_2, etc. The first equation gives the first row of the matrix A, the second equation gives the second row, etc. If x_1, x_2 or x_3 does not appear in an equation, its coefficient is 0 on the row of the matrix.

```
A = [ -(k1+k2)/m1        k2/m1            0
          k2/m2     -(k2+k3)/m2        k3/m2
              0          k3/m3   -(k3+k4)/m3   ]
```

Then Write the Program for Problem 2:

```
m1 = .8;
m2 = .6;
m3 = .5;
k1 = 4.3;
k2 = 5.1;
k3 = 4.6;
k4 = 5.4;
A = [ -(k1+k2)/m1          k2/m1              0
            k2/m2    -(k2+k3)/m2          k3/m2
                0          k3/m3   -(k3+k4)/m3  ];
A = -A;
[eigvec  eigval] = eig(A);
t = 0:.001:8;
line1x = [0  8];
line1y = [0  0];
titles(1,:) = 'Problem 2, Figure 1';
titles(2,:) = 'Problem 2, Figure 2';
titles(3,:) = 'Problem 2, Figure 3';
for(k=1:3)
    w = sqrt(eigval(k,k));
    c1 = eigvec(1,k);
    c2 = eigvec(2,k);
    c3 = eigvec(3,k);
    x1 = c1*cos(w*t);
    x2 = c2*cos(w*t);
    x3 = c3*cos(w*t);
    figure(k);
    plot(t,x1,'b',t,x2,'r',t,x3,'g',line1x,line1y,'k');
    axis([0  8  -.8  .8]);
    set(gca,'xtick',0:8);
    set(gca,'ytick',-.8:.2:.8);
    xlabel('t');
    ylabel('x1(blue), x2(red), x3(green)');
    title(titles(k,:));
end
```

3) Write a MATLAB program to calculate and print the following integral:

$$I = \int_4^8 \frac{\sqrt{2x^4+3x}\cdot\cos(5x)\cdot\ln\left(\frac{4x^3+7x}{x^2+1}\right)}{x^2\sin\left(\frac{3x+1}{2x}\right)\cdot e^{-\frac{5x^3+2x}{6x^2+2}}}\,dx$$

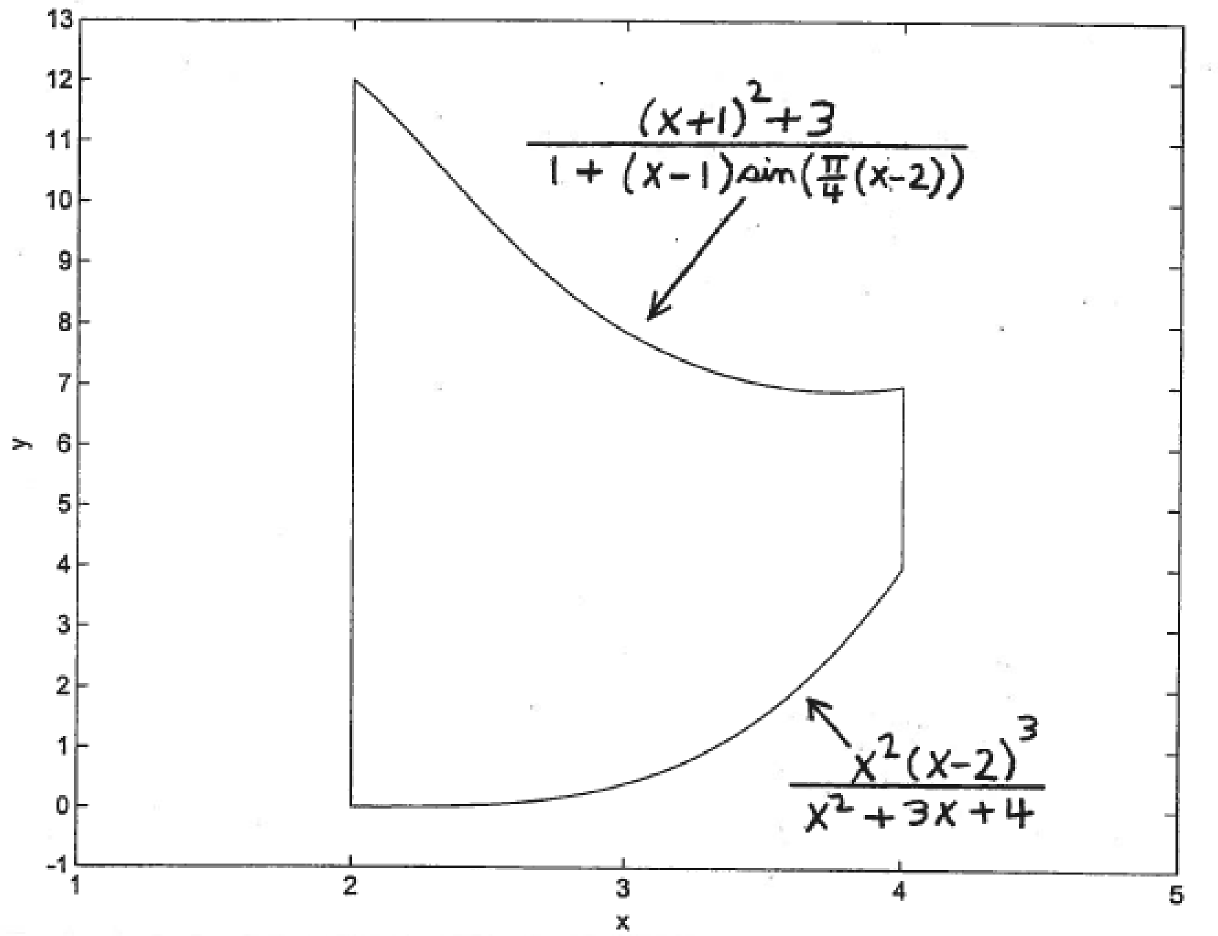
Use 1e-7 as the accuracy factor. The output of this program should look like this:

I=454.36167

Answer

```
a=4;
b=8;
accuracy=1e-7;
f=@(x) sqrt(2*x.^4+3*x).*cos(5*x).*log((4*x.^3+7*x)./(x.^2+1))./ ...
       (x.^2.*sin((3*x+1)./(2*x)).*exp(-(5*x.^3+2*x)./(6*x.^2+2)));
I=quad(f,a,b,accuracy);
fprintf('I=%.5f\n',I);
```

4)



The thin flat sheet shown above has density

$$\rho = \frac{y^2\cos(x+y)}{x+2y+1} + y^3 e^{\frac{2x+3y}{x+y+1}}$$

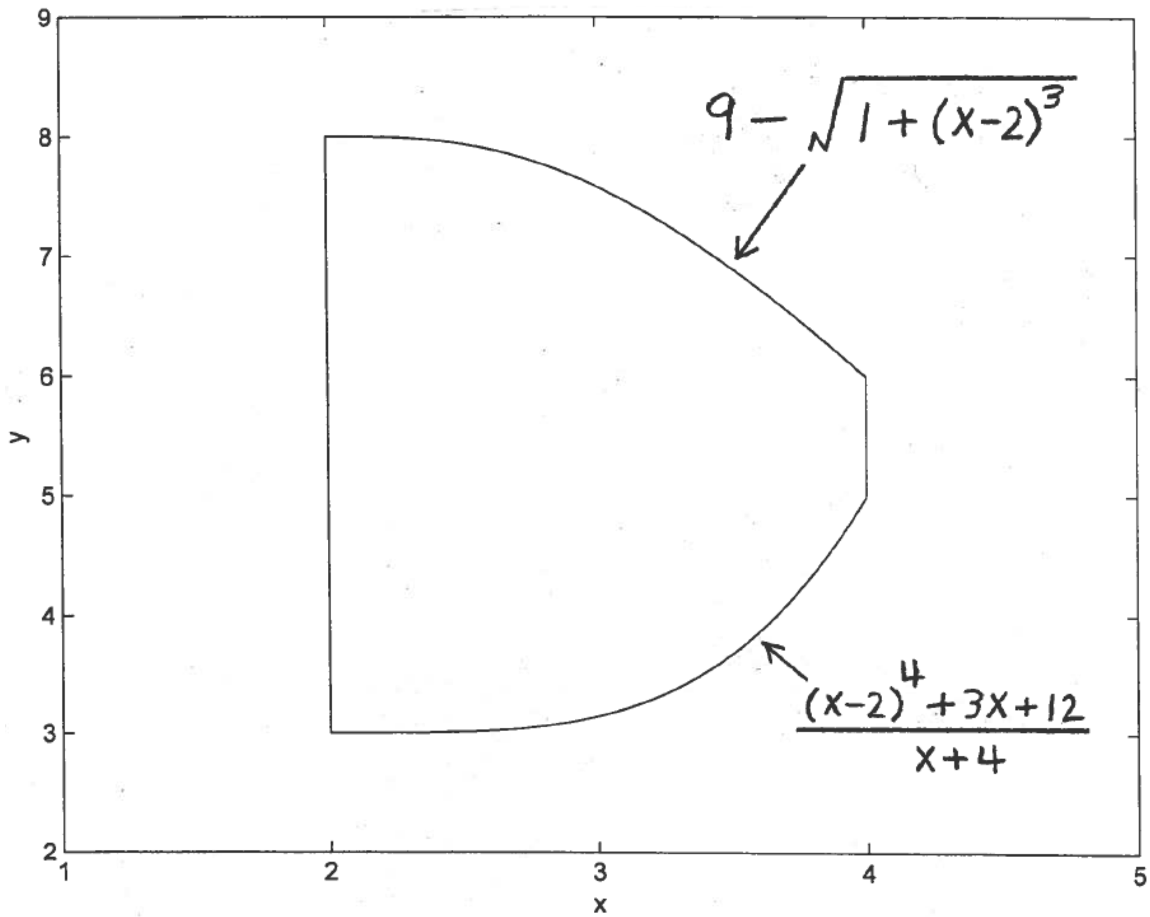
Write a MATLAB program to calculate and print the mass of the thin flat sheet. Use 1e-8 as the accuracy factors. The output of this program should look like this:

```
mass=38721.48074
```

<u>Answer</u>

```
a = 2;
b = 4;
g = @(x) x.^2.*(x-2).^3./(x.^2 + 3*x + 4);
h = @(x) ((x+1).^2 + 3)./(1 + (x-1).*sin(pi/4*(x-2)));
f=@(x,y) y.^2.*cos(x+y)./(x+2*y+1) + y.^3.*exp((2*x+3*y)./(x+y+1));
mass = quad2d(f,a,b,g,h,'RelTol',1e-8,'AbsTol',1e-8);
fprintf('mass=%.5f\n',mass);
```

5)



A solid is bounded in the x and y directions by the region shown above and is bounded below and above in the z direction by the planes z=8 and z=x+y+2. The density of the solid is given by

$$\rho = \frac{z^3\cos(x+y+z)}{2x+3z+2} + z^2\ln\left(y\sqrt{z+y^3}\right)$$

Write a MATLAB program to calculate and print the mass of the solid. Use 1e-4 as the accuracy factor. The output of this program should look like this:

mass=7523.429

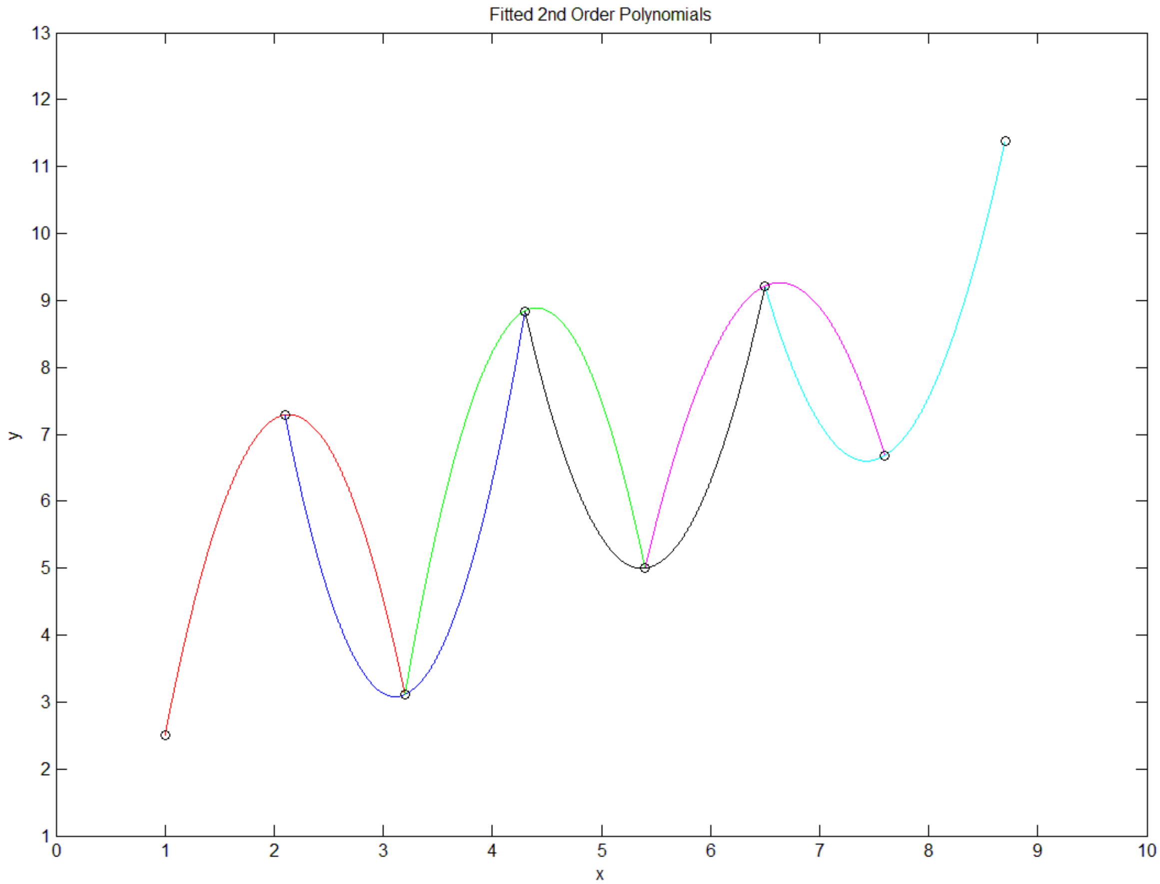
__Answer__

```
global accuracy;
a = 2;
b = 4;
accuracy = 1e-4;
g = @(x) ((x-2)^4 + 3*x + 12)/(x+4);
h = @(x) 9 - sqrt(1 + (x-2)^3);
v = @(x,y) 8;
w = @(x,y) x + y + 2;
f=@(z,x,y) z.^3.*cos(x+y+z)./(2*x+3*z+2) + z.^2.*log(y*sqrt(z+y^3));
mass = quad('middle',a,b,accuracy,[],'inner',g,h,f,v,w);
fprintf('mass=%.3f \n',mass);
```

6) Write a MATLAB program as follows:

a) Read a data file (prog6.dat) that has 8 lines, where each line contains a value of x and a value of y (data point).
b) For each data point from the second one to the second to last one, fit a second order polynomial to that data point and the data point on either side of it (fit the second order polynomial to three data points). Plot these fitted second order polynomials using the colors red, blue, green, black, magenta and cyan and the data points as black circles, all in the same graph.

The graph should look like this:



Answer

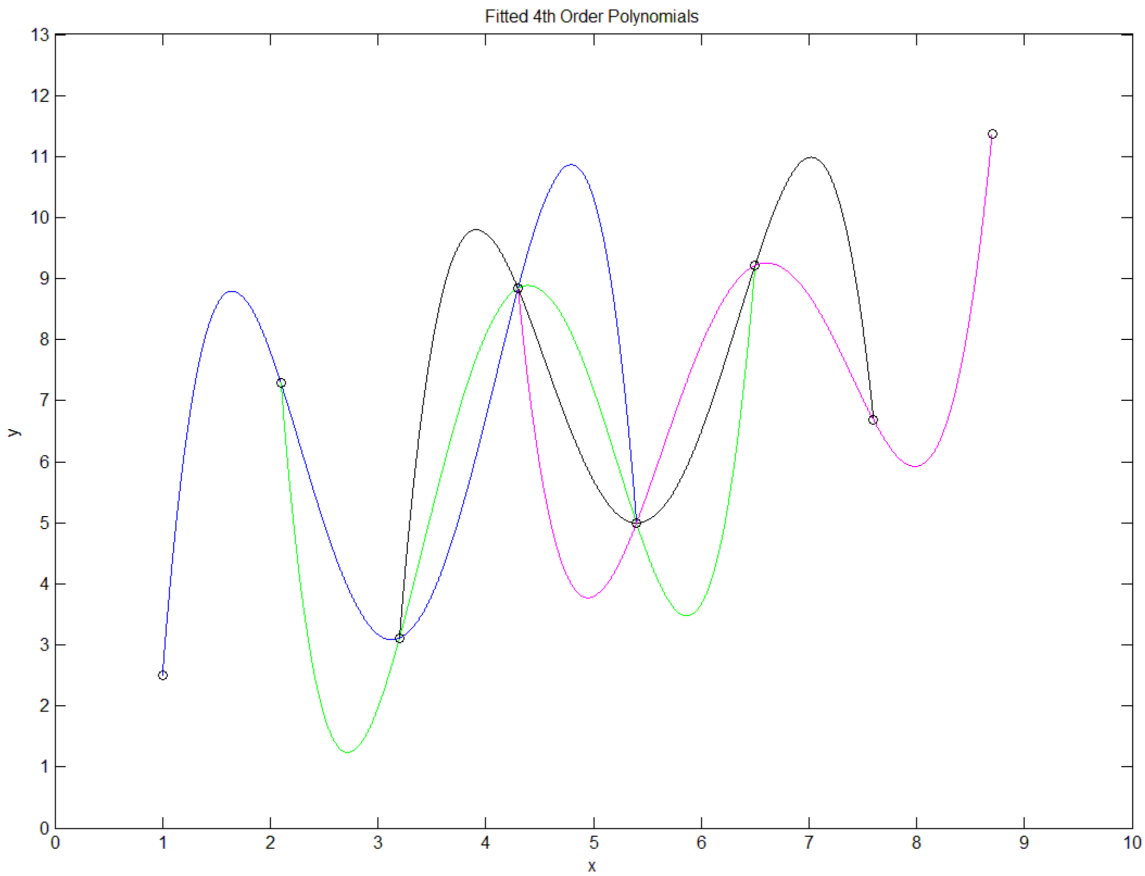
```
[xd yd] = textread('prog6.dat');
n = length(xd);
color = ['k','r','b','g','k','m','c','k'];
hold on;
box on;
for(k = 2:n-1)
    xd3 = [xd(k-1)  xd(k)  xd(k+1)];
    yd3 = [yd(k-1)  yd(k)  yd(k+1)];
    c2 = polyfit(xd3,yd3,2);
    x = xd(k-1):.001:xd(k+1);
    y2 = polyval(c2,x);
    plot(x,y2,color(k),xd,yd,'ko');
    axis([0 10 1 13]);
    set(gca,'xtick',0:10);
    set(gca,'ytick',1:13);
    xlabel('x');
    ylabel('y');
    title('Fitted 2nd Order Polynomials');
end
```

7) Write a MATLAB program as follows:

a) Read a data file (prog7.dat) that has 8 lines, where each line contains a value of x and a value of y (data point).

b) For each data point from the third one to the third to last one, fit a fourth order polynomial to that data point and the two data points on either side of it (fit the fourth order polynomial to five data points). Plot these fitted fourth order polynomials using the colors blue, green, black and magenta and the data points as black circles, all in the same graph.

The graph should look like this:



<u>**Answer**</u>

```
[xd  yd] = textread('prog7.dat');
n = length(xd);
color = ['k','k','b','g','k','m','k','k'];
hold on;
box on;
for(k = 3:n-2)
    xd5 = [xd(k-2)  xd(k-1)  xd(k)  xd(k+1)  xd(k+2)];
    yd5 = [yd(k-2)  yd(k-1)  yd(k)  yd(k+1)  yd(k+2)];
    c4 = polyfit(xd5,yd5,4);
    x = xd(k-2):.001:xd(k+2);
    y4 = polyval(c4,x);
    plot(x,y4,color(k),xd,yd,'ko');
    axis([0 10 0 13]);
    set(gca,'xtick',0:10);
    set(gca,'ytick',0:13);
    xlabel('x');
    ylabel('y');
    title('Fitted 4th Order Polynomials');
end
```

8) Write a MATLAB program as follows:

1) Read a data file (prog8.dat) that has values of x and y (data points).

2) For each data point from the third one to the third to last one, do the following:

   a) Fit a second order polynomial to that data point and the data point on either side of it (fit the second order polynomial to three data points).

   b) Fit a fourth order polynomial to that data point and the two data points on either side of it (fit the fourth order polynomial to five data points).

   c) Use the fitted second order and fourth order polynomials to calculate numerical values for the first and second derivatives at that data point. Use the variables der2 and der4 for the first derivative obtained from the fitted second and fourth order polynomials, respectively, and use the variables secder2 and secder4 for the second derivative obtained from the fitted second and fourth order polynomials, respectively.

   d) Print the x coordinate of the data point and the numerical derivatives obtained from the fitted polynomials.

The output of this program should look like this:

```
x=-2.0  der2=0.518  der4= 0.538  secder2= 7.786  secder4= 10.348
x=-0.9  der2=0.567  der4= 0.612  secder2=-7.697  secder4=-10.237
x= 0.2  der2=0.345  der4= 0.319  secder2= 7.293  secder4=  9.768
x= 1.3  der2=0.279  der4= 0.197  secder2=-7.413  secder4= -9.939
x= 2.4  der2=0.705  der4= 0.751  secder2= 8.188  secder4= 10.830
x= 3.5  der2=0.858  der4= 0.998  secder2=-7.911  secder4=-10.466
x= 4.6  der2=0.169  der4=-0.046  secder2= 6.658  secder4=  8.891
x= 5.7  der2=0.766  der4= 0.829  secder2=-5.572  secder4= -7.553
```

Answer

```
[xd  yd] = textread('prog8.dat');
n = length(xd);
for(k = 3:n-2)
   xd3 = [xd(k-1)  xd(k)  xd(k+1)];
   yd3 = [yd(k-1)  yd(k)  yd(k+1)];
   xd5 = [xd(k-2)  xd(k-1)  xd(k)  xd(k+1)  xd(k+2)];
   yd5 = [yd(k-2)  yd(k-1)  yd(k)  yd(k+1)  yd(k+2)];
   c2 = polyfit(xd3,yd3,2);
   c4 = polyfit(xd5,yd5,4);
   cder2 = polyder(c2);
   der2 = polyval(cder2,xd(k));
   csecder2 = polyder(cder2);
   secder2 = polyval(csecder2,xd(k));
   cder4 = polyder(c4);
   der4 = polyval(cder4,xd(k));
   csecder4 = polyder(cder4);
   secder4 = polyval(csecder4,xd(k));
   fprintf('x=%4.1f der2=%5.3f der4=%6.3f secder2=%6.3f secder4=%7.3f\n',...
           xd(k),der2,der4,secder2,secder4);
end
```

9) Consider the following Poisson Equation for u(x,y) for $0 \le x \le 1$ and $0 \le y \le 1$:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$$

$$f(x,y) = -2(x^2+y^2)$$

with the following Dirichlet boundary conditions:

u(x,0) = gbottom(x) = $1-x^2$,   u(x,1) = gtop(x) = $4(1-x^2)$

u(0,y) = gleft(y) = $1+y^2$,   u(1,y) = gright(y) = 0

NOTE: There will be 4 functions, named gbottom(x), gtop(x), gleft(y) and gright(y), that give the boundary conditions on the bottom (y=0), top (y=1), left (x=0) and right (x=1) sides of the square.

Write a MATLAB program as follows:

1) Use the 5-point scheme to calculate numerical values for the unknown u for $0 < x < 1$ and $0 < y < 1$. Divide both the x interval [0, 1] and the y interval [0, 1] into 25 equal subdivisions (there will be 26 equally spaced grid points in both the x and y intervals). Use 1e-8 as the accuracy factor. The main program will call a function named poisson that solves the Poisson Equation for the unknown u and returns it to the main program. The first line of poisson is:

   function u=poisson(f,gbottom,gtop,gleft,gright,n,L,accuracy)

   where n is the number of grid points in both the x and y intervals, and L is the length of both the x and y intervals.

2) Plot u versus x and y for $0 \le x \le 1$ and $0 \le y \le 1$. u will be a surface in 3-dimensional space. Use the MATLAB function surf to plot u. <u>Do not write any other statements for the graph except the surf statement.</u>

There are two parts to this problem:

a) Write the main program.
b) Write the function poisson.

<u>Problem 9 Part a Answer</u>

```
% main program
L=1;
n=26;
accuracy=1e-8;
f=@(x,y) -2*(x^2+y^2);
gbottom=@(x) 1-x^2;
gtop=@(x) 4*(1-x^2);
gleft=@(y) 1+y^2;
gright=@(y) 0;
u=poisson(f,gbottom,gtop,gleft,gright,n,L,accuracy);
h=L/(n-1);
x=0:h:L;
y=0:h:L;
surf(x,y,u');
```

Problem 9 Part b Answer

Before writing the function poisson, obtain the equation to be iterated by doing the following:

1) Approximate the second order partial derivatives in the Poisson Equation by the 3-point second order central difference formula, using the point with indices i,j as the central point:

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} = f_{i,j}$$

where $u_{i,j} = u(i,j)$, $f_{i,j} = f(x_i,y_j)$, and h is the stepsize in both the x and y intervals.

2) Solve the equation for $u_{i,j}$:

$$u_{i,j} = \frac{1}{4}\left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - h^2 f_{i,j}\right)$$

Then write the function poisson:

```
% function poisson
function u=poisson(f,gbottom,gtop,gleft,gright,n,L,accuracy)
h=L/(n-1);
u=zeros(n,n);
for(i=1:n)
  u(i,1)=gbottom((i-1)*h);
  u(i,n)=gtop((i-1)*h);
end
for(j=1:n)
  u(1,j)=gleft((j-1)*h);
  u(n,j)=gright((j-1)*h);
end
max_diff = 1;
while(max_diff >= accuracy)
   max_diff = 0;
   for(i = 2:n-1)
      for(j = 2:n-1)
         uij_old = u(i,j);
         u(i,j) = (u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1) ...
                 -h^2*f((i-1)*h,(j-1)*h))/4;
         diff = abs(u(i,j)-uij_old);
         if(diff > max_diff)
             max_diff = diff;
         end
      end
   end
end
```

10) Consider the following Poisson equation for u(x,y) for $0 \le x \le 2$ and $0 \le y \le 1$:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$$

$$f(x,y) = -2(x^2+y^2)$$

with the following Dirichlet boundary conditions:

u(x,0) = gbottom(x) = $1-x^2$, u(x,1) = gtop(x) = $4(1-x^2)$

u(0,y) = gleft(y) = $1+y^2$, u(2,y) = gright(y) = 2

NOTE: The x and y intervals have different lengths in this problem (the length of the x interval is 2 and the length of the y interval is 1). Use the variables Lx and Ly for the lengths of the x and y intervals, nx and ny for the number of grid points in the x and y intervals (the number of grid points is different in the x and y intervals), and hx and hy for the stepsizes in the x and y intervals (the stepsize is different in the x and y intervals).

NOTE: There will be 4 functions, named gbottom(x), gtop(x), gleft(y) and gright(y), that give the boundary conditions on the bottom (y=0), top (y=1), left (x=0) and right (x=2) sides of the rectangle.

Write a MATLAB program as follows:

1) Use the 5-point scheme to calculate numerical values for the unknown u for $0 < x < 2$ and $0 < y < 1$. Divide the x interval [0, 2] into 52 equal subdivisions (there will be 53 equally spaced grid points in the x direction), and divide the y interval [0, 1] into 20 equal subdivisions (there will be 21 equally spaced grid points in the y direction). Use 1e-8 as the accuracy factor. The main program will call a function named poisson2 that solves the Poisson Equation for the unknown u and returns it to the main program. The first line of poisson2 is:

   ```
   function u = poisson2(f,gbottom,gtop,gleft,gright,nx,ny,Lx,Ly,accuracy)
   ```

   where f(x,y) is the function on the right of the equals sign in the Poisson Equation, accuracy is the accuracy factor, and the other parameters of poisson2 are defined above.

2) Plot u versus x and y for $0 \le x \le 2$ and $0 \le y \le 1$. u will be a surface in 3-dimensional space. Use the MATLAB function surf to plot u. Do not write any other statements for the graph except the surf statement.

There are two parts to this problem:

a) Write the main program.

b) Write the function poisson2.

Problem 10 Part a Answer

```
% main program
Lx = 2;
Ly = 1;
nx = 53;
ny = 21;
accuracy = 1e-8;
f=@(x,y) -2*(x^2+y^2);
gbottom=@(x) 1-x^2;
gtop=@(x) 4*(1-x^2);
gleft=@(y) 1+y^2;
gright=@(y) 2;
u = poisson2(f,gbottom,gtop,gleft,gright,nx,ny,Lx,Ly,accuracy);
hx = Lx/(nx-1);
hy = Ly/(ny-1);
x = 0:hx:Lx;
y = 0:hy:Ly;
surf(x,y,u');
```

Problem 10 Part b Answer

```
% function poisson2
function u = poisson2(f,gbottom,gtop,gleft,gright,nx,ny,Lx,Ly,accuracy)
hx = Lx/(nx-1);
hy = Ly/(ny-1);
u = zeros(nx,ny);
for(i = 1:nx)
  u(i,1) = gbottom((i-1)*hx);
  u(i,ny) = gtop((i-1)*hx);
end
for(j = 1:ny)
  u(1,j) = gleft((j-1)*hy);
  u(nx,j) = gright((j-1)*hy);
end
max_diff = 1;
while(max_diff >= accuracy)
   max_diff = 0;
   for(i = 2:nx-1)
      for(j = 2:ny-1)
         uij_old = u(i,j);
         u(i,j) = ( hy^2*(u(i-1,j)+u(i+1,j)) + hx^2*(u(i,j-1)+u(i,j+1)) ...
                    -hx^2*hy^2*f((i-1)*hx,(j-1)*hy) ) / (2*(hx^2+hy^2));
         diff = abs(u(i,j)-uij_old);
         if(diff > max_diff)
             max_diff = diff;
         end
      end
   end
end
```

11) Consider the following One-Dimensional Heat Equation for u(x,t) for $0 \le x \le 1$ and $0 \le t \le .2$:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + f(x,t)$$

$$f(x,t) = 0$$

$$a = 1$$

with the following initial conditions:

u(x,0) = u0(x) = sin(πx)

and the following boundary conditions:

u(0,t) = gleft(t) = 0

u(1,t) = gright(t) = 0

Write a MATLAB program as follows:

1) Use the explicit full discretization scheme to calculate numerical values for the unknown u(x,t) for $0 < x < 1$ and $0 < t \le .2$. Divide the x interval [0, 1] into 12 equal subdivisions and the t interval [0, .2] into 96 equal subdivisions (there will be 13 equally spaced grid points in the x interval and 97 equally spaced grid points in the t interval). Use the variables L for the length of the x interval, T for the length of the t interval, nx and nt for the number of grid points in the x and t intervals, and hx and ht for the stepsizes in the x and t intervals. The main program will call a function named heat1 that solves the One-Dimensional Heat Equation for the unknown u and returns it to the main program. The first line of heat1 is:

   ```
   function u = heat1(f, u0, gleft, gright, a, nx, nt, L, T)
   ```

2) Plot u versus x and t for $0 \le x \le 1$ and $0 \le t \le .2$. u will be a surface in 3-dimensional space. Use the MATLAB function surf to plot u. Do not write any other statements for the graph except the surf statement.

There are two parts to this problem:

a) Write the main program.

b) Write the function heat1.

Problem 11 Part a Answer

```
% main program
L = 1;
T = .2;
a = 1;
f = @(x,t) 0;
u0 = @(x) sin(pi*x);
gleft = @(t) 0;
gright = @(t) 0;
nx = 13;
nt = 97;
u = heat1(f, u0, gleft, gright, a, nx, nt, L, T);
hx = L/(nx-1);
ht = T/(nt-1);
x = 0 : hx : L;
t = 0 : ht : T;
surf(x, t, u');
```

The answer to Problem 11 Part b is on the next page.

Problem 11 Part b Answer

Before writing the function heat1, obtain the equation to be used in the explicit scheme by doing the following:

1) **In the One-Dimensional Heat Equation, approximate the first order partial derivative by the 2-point first order forward difference formula and approximate the second order partial derivative by the 3-point second order central difference formula, using the point with indices i,k-1 as the central point:**

$$\frac{u_{i,k} - u_{i,k-1}}{h_t} = a\frac{u_{i-1,k-1} - 2u_{i,k-1} + u_{i+1,k-1}}{h_x^2} + f_{i,k-1}$$

where $u_{i,k} = u(i,k)$, $f_{i,k-1} = f(x_i, t_{k-1})$, and h_x and h_t are the stepsizes in the x and t intervals.

2) **Solve the equation for $u_{i,k}$:**

$$u_{i,k} = \frac{a h_t}{h_x^2}\left(u_{i-1,k-1} - 2u_{i,k-1} + u_{i+1,k-1}\right) + h_t f_{i,k-1} + u_{i,k-1}$$

Then write the function heat1:

```
% function heat1
function u = heat1(f, u0, gleft, gright, a, nx, nt, L, T)
hx = L/(nx-1);
ht = T/(nt-1);
u = zeros(nx, nt);
for(i = 1 : nx)
  u(i,1) = u0( (i-1)*hx );
end
for(k = 2 : nt)
  u(1,k) = gleft( (k-1)*ht );
  u(nx,k) = gright( (k-1)*ht );
end
for(k = 2 : nt)
  for(i = 2 : nx-1)
    u(i,k) = ht*a/hx^2*( u(i-1,k-1) - 2*u(i,k-1) + u(i+1,k-1) ) ...
             + ht * f( (i-1)*hx, (k-2)*ht ) + u(i,k-1);
  end
end
```

12) Consider the following 2 Dimensional heat equation for u(x,y,t) for $0 \le x \le 2$, $0 \le y \le .6$, and $0 \le t \le .2$:

$$\frac{\partial u}{\partial t} = a\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f(x,y,t)$$

$$f(x,y,t) = -\frac{8}{\pi^2}\sin(\pi x)\cos(\pi y)e^{t}$$

$$a = \frac{4}{\pi^2}$$

With the following initial conditions:

$$u(x,y,0) = u0(x,y) = \sin\left(\frac{\pi}{2}x\right)\sin(\pi y) + xy + 2x(1-y)$$

and the following Dirichlet boundary conditions:

$$u(x,0,t) = \text{gbottom}(x,t) = 2x\cos(\pi t)$$
$$u(x,1,t) = \text{gtop}(x,t) = x\cos(\pi t)$$
$$u(0,y,t) = \text{gleft}(y,t) = 0$$
$$u(2,y,t) = \text{gright}(y,t) = (2y + 4(1-y))\cos(\pi t)$$

NOTE: The x and y intervals have different lengths in this problem (the length of the x interval is 2 and the length of the y interval is .6). Use the variables Lx and Ly for the lengths of the x and y intervals, nx and ny for the number of grid points in the x and y intervals (the number of grid points is different in the x and y intervals), and hx and hy for the stepsizes in the x and y intervals (the stepsize is different in the x and y intervals). Use the variable T for the size of the t interval and the variables nt and ht for the number of grid points and the stepsize in the t interval.

There will be 4 functions, named gbottom, gtop, gleft and gright, that give the boundary conditions on the bottom (y=0), top (y=.6), left (x=0) and right (x=2) sides of the rectangle, respectively.

Write a MATLAB program as follows:

1) Use the explicit full discretization scheme to calculate numerical values for the unknown u(x,y,t) for $0 < x < 2$, $0 < y < .6$ and $0 < t \leq .2$. Divide the x interval [0, 2] into 52 equal subdivisions (there will be 53 equally spaced grid points in the x direction), divide the y interval [0, .6] into 20 equal subdivisions (there will be 21 equally spaced grid points in the y direction), and divide the t interval [0, .2] into 192 equal subdivisions (there will be 193 equally spaced grid points in the t direction).
   The main program will call a function named heat2 that solves the Two-Dimensional Heat Equation for the unknown u and returns it to the main program. The first line of heat2 is:

   ```
   function u = heat2(f,u0,gbottom,gtop,gleft,gright,a,nx,ny,nt,Lx,Ly,T)
   ```

2) Plot u versus x and y for the t index k = 7 . u will be a surface in 3-dimensional space. Use the MATLAB function surf to plot u. Do not write any other statements for the graph except the surf statement.

There are two parts to this problem:

a) Write the main program.

b) Write the function heat2.

Problem 12 Part a Answer

```
% main program
Lx = 2;
Ly = .6;
T = .2;
a = 4/pi^2;
f = @(x,y,t) -8/pi^2*sin(pi*x)*cos(pi*y)*exp(t);
u0 = @(x,y) sin(pi/2*x)*sin(pi*y) + x*y + 2*x*(1-y);
gbottom = @(x,t) 2*x*cos(pi*t);
gtop = @(x,t) x*cos(pi*t);
gleft = @(y,t) 0;
gright = @(y,t) ( 2*y + 4*(1-y) )*cos(pi*t) ;
nx = 53;
ny = 21;
nt = 193;
u = heat2(f,u0,gbottom,gtop,gleft,gright,a,nx,ny,nt,Lx,Ly,T);
hx = Lx/(nx-1);
hy = Ly/(ny-1);
x = 0:hx:Lx;
y = 0:hy:Ly;
u2 = u(:,:,7);
surf(x,y,u2');
```

Problem 12 Part b Answer

```
% function heat2
function u = heat2(f,u0,gbottom,gtop,gleft,gright,a,nx,ny,nt,Lx,Ly,T)
hx=Lx/(nx-1);
hy=Ly/(ny-1);
ht=T/(nt-1);
u=zeros(nx,ny,nt);
for(i=1:nx)
    for(j=1:ny)
        u(i,j,1)=u0((i-1)*hx,(j-1)*hy);
    end
end
for(k=2:nt)
    for(i = 1:nx)
        u(i,1,k) = gbottom((i-1)*hx,(k-1)*ht);
        u(i,ny,k) = gtop((i-1)*hx,(k-1)*ht);
    end
    for(j = 1:ny)
        u(1,j,k) = gleft((j-1)*hy,(k-1)*ht);
        u(nx,j,k) = gright((j-1)*hy,(k-1)*ht);
    end
end
for(k=2:nt)
   for(i=2:nx-1)
      for(j=2:ny-1)
         u(i,j,k) =   ht*a/hx^2*(u(i-1,j,k-1)-2*u(i,j,k-1)+u(i+1,j,k-1)) ...
                    + ht*a/hy^2*(u(i,j-1,k-1)-2*u(i,j,k-1)+u(i,j+1,k-1)) ...
                    + ht*f((i-1)*hx,(j-1)*hy,(k-2)*ht) + u(i,j,k-1);
      end
   end
end
```

13) Consider the following One-Dimensional Wave Equation for u(x,t) for $0 \le x \le 2\pi$ and $0 \le t \le 6$:

$$\frac{\partial^2 u}{\partial t^2} = a\frac{\partial^2 u}{\partial x^2} + f(x,t)$$

$$f(x,t) = 2e^{-\frac{t}{2}}\sin\left(\frac{x}{2}\right)$$

$$a = \frac{1}{4\pi^2}$$

with the following initial conditions at t=0:

u(x,0) = u0(x) = sin(x/2),

v(x,0) = v0(x) = -sin(x/2)

and the following boundary conditions:

u(0,t) = gleft(t) = sin(π/6*t)

u(2π,t) = gright(t) = sin(π/12*t)

Write a MATLAB program as follows:

1) Use the explicit full discretization scheme to calculate numerical values for the unknown u(x,t) for $0 < x < 2\pi$ and $0 < t \le 6$. Divide the x interval [0, 2π] into 20 equal subdivisions and the t interval [0, 6] into 30 equal subdivisions (there will be 21 equally spaced grid points in the x interval and 31 equally spaced grid points in the t interval). Use the variables L and T for the lengths of the x and t intervals, nx and nt for the number of grid points in the x and t intervals, and hx and ht for the stepsizes in the x and t intervals.
   The main program will call a function named wave1 that solves the One-Dimensional Wave Equation for the unknown u and returns it to the main program. The first line of wave1 is:

```
function u = wave1(f,u0,v0,gleft,gright,a,nx,nt,L,T)
```

2) Plot u versus x and t for $0 \le x \le 2\pi$ and $0 \le t \le 6$. u will be a surface in 3-dimensional space. Use the MATLAB function surf to plot u. Do not write any other statements for the graph except the surf statement.

There are two parts to this problem:

a) Write the main program.

b) Write the function wave1.

Problem 13 Part a Answer

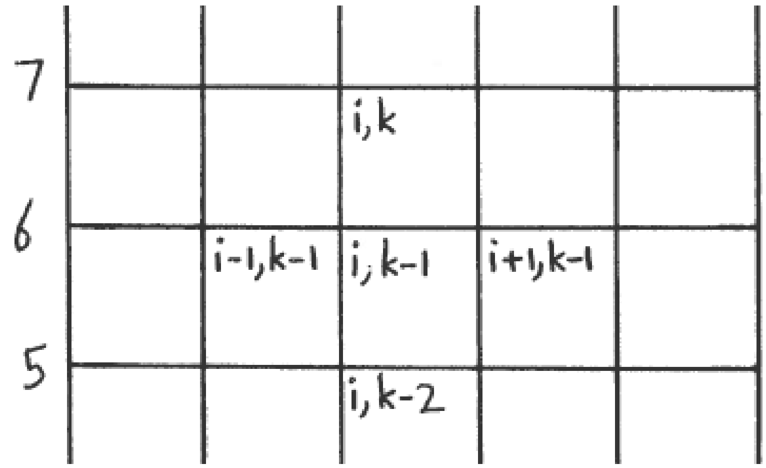
```
% main program
L = 2*pi;
T = 6;
a = 1/(4*pi^2);
f = @(x,t) 2*exp(-t/2)*sin(x/2);
u0 = @(x) sin(x/2);
v0 = @(x) -sin(x/2);
gleft = @(t) sin(pi/6*t);
gright = @(t) sin(pi/12*t);
nx = 21;
nt = 31;
u = wave1(f,u0,v0,gleft,gright,a,nx,nt,L,T);
hx = L/(nx-1);
ht = T/(nt-1);
x = 0:hx:L;
t = 0:ht:T;
surf(x,t,u');
```

The answer to Problem 13 Part b is on the next page.

Problem 13 Part b Answer

Before writing the function wave1, obtain the equation to be used in the explicit scheme by doing the following:

1) Approximate the second order partial derivatives in the One-Dimensional Wave Equation by the 3-point second order central difference formula, using the point with indices i,k-1 as the central point:



$$\frac{\partial^2 u}{\partial t^2} = a\frac{\partial^2 u}{\partial x^2} + f(x,t)$$

$$\frac{u_{i,k-2} - 2u_{i,k-1} + u_{i,k}}{h_t^2} = a\frac{u_{i-1,k-1} - 2u_{i,k-1} + u_{i+1,k-1}}{h_x^2} + f_{i,k-1}$$

where $u_{i,k} = u(i,k)$, $f_{i,k-1} = f(x_i, t_{k-1})$, and h_x and h_t are the stepsizes in the x and t intervals.

2) Solve the equation for $u_{i,k}$:

$$u_{i,k} = \frac{a h_t^2}{h_x^2}\left(u_{i-1,k-1} - 2u_{i,k-1} + u_{i+1,k-1}\right) + h_t^2 f_{i,k-1} + 2u_{i,k-1} - u_{i,k-2}$$

NOTE: This equation can be used only for $k \geq 3$. It cannot be used for $k = 2$ because when $k = 2$, the term $u_{i,k-2}$ becomes $u_{i,0}$ which is undefined because the second index cannot be less than 1 (1 corresponds to t=0; if the second index were less than 1, it would correspond to a negative time).

3) In order to obtain an equation that can be used when k = 2, do the following:

3a) The velocity v is the first derivative of u with respect to time:

$$V(x,t) = \frac{\partial u}{\partial t}$$

3b) Approximate the first order partial derivative in the above equation by the 2-point backward difference formula, using the point with indices i,k-1 as the central point:

$$v_{i,k-1} = \frac{u_{i,k-1} - u_{i,k-2}}{h_t}$$

where $v_{i,k-1} = v(x_i, t_{k-1})$.

3c) Solve this equation for $u_{i,k-2}$:

$$u_{i,k-2} = u_{i,k-1} - h_t v_{i,k-1}$$

3d) Substitute this equation for $u_{i,k-2}$ into the equation in step 2 above:

$$u_{i,k} = \frac{a h_t^2}{h_x^2}\left(u_{i-1,k-1} - 2u_{i,k-1} + u_{i+1,k-1}\right) + h_t^2 f_{i,k-1} + u_{i,k-1} + h_t v_{i,k-1}$$

Use the above equation for k = 2 in which case $v_{i,k-1}$ becomes $v_{i,1}$ which is given by the initial velocity v0 since $t_1 = 0$:

$v_{i,1} = v(x_i, t_1) = v(x_i, 0) = v0(x_i)$.

Then write the function wave1:

```
% function wave1
function u = wave1(f,u0,v0,gleft,gright,a,nx,nt,L,T)
hx = L/(nx-1);
ht = T/(nt-1);
u = zeros(nx,nt);
for(i = 1:nx)
    u(i,1) = u0((i-1)*hx);
end
for(k = 2:nt)
    u(1,k) = gleft((k-1)*ht);
    u(nx,k) = gright((k-1)*ht);
end
k = 2;
for(i = 2:nx-1)
    u(i,k) = ht^2*a/hx^2*(u(i-1,k-1) - 2*u(i,k-1) + u(i+1,k-1)) ...
             + ht^2*f((i-1)*hx,(k-2)*ht) + u(i,k-1) + ht*v0((i-1)*hx);
end
for(k = 3:nt)
    for(i = 2:nx-1)
        u(i,k) = ht^2*a/hx^2*(u(i-1,k-1) - 2*u(i,k-1) + u(i+1,k-1)) ...
                 + ht^2*f((i-1)*hx,(k-2)*ht) + 2*u(i,k-1) - u(i,k-2);
    end
end
```