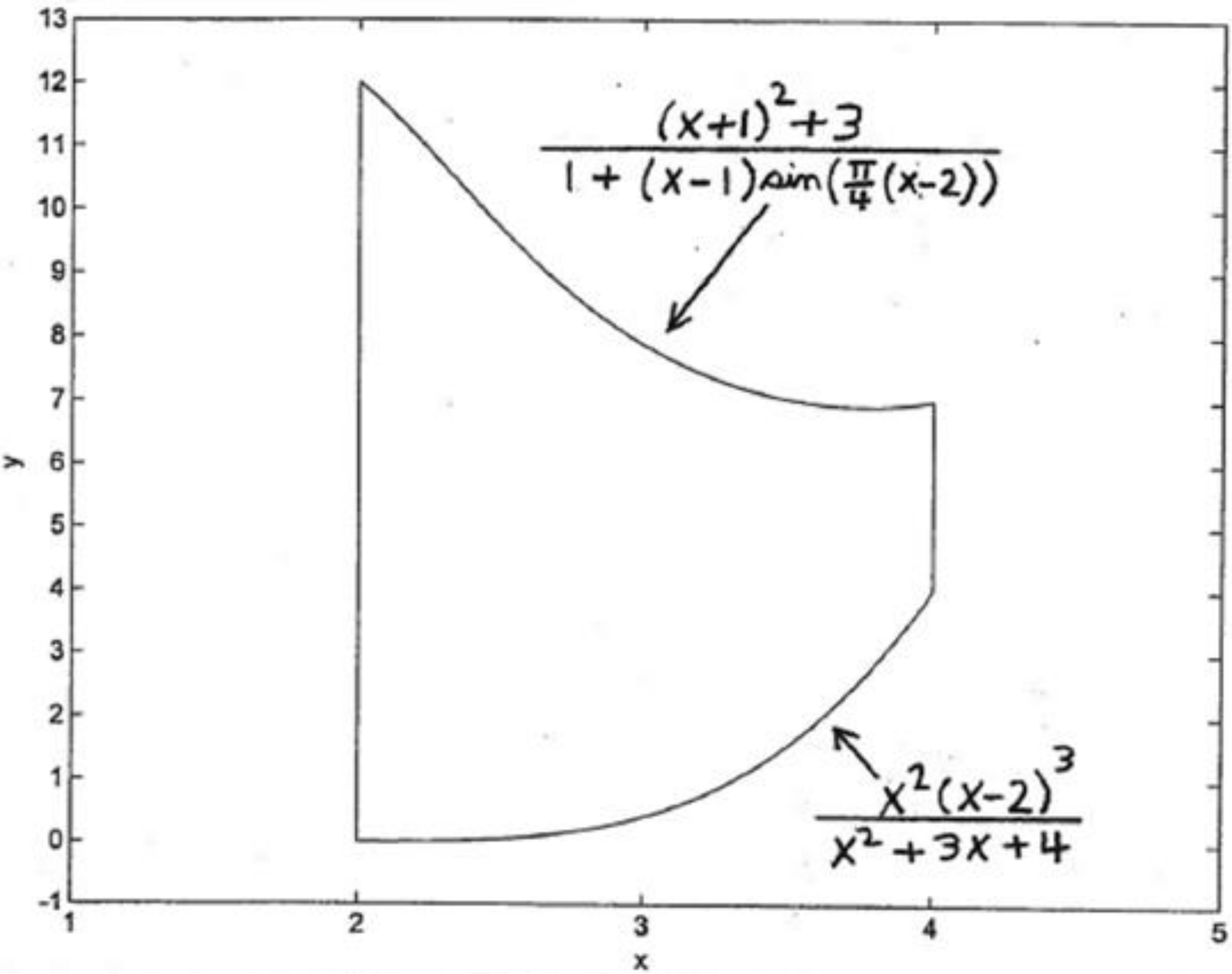
Question 1

Complete Points out of 1.00 Flag question

1)



The thin flat sheet shown above has density

$$\rho = \frac{y^4 \ln(x+y)}{x+3y+2} + y \ln\left(y^2 \sqrt{x+y^3}\right)$$

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:

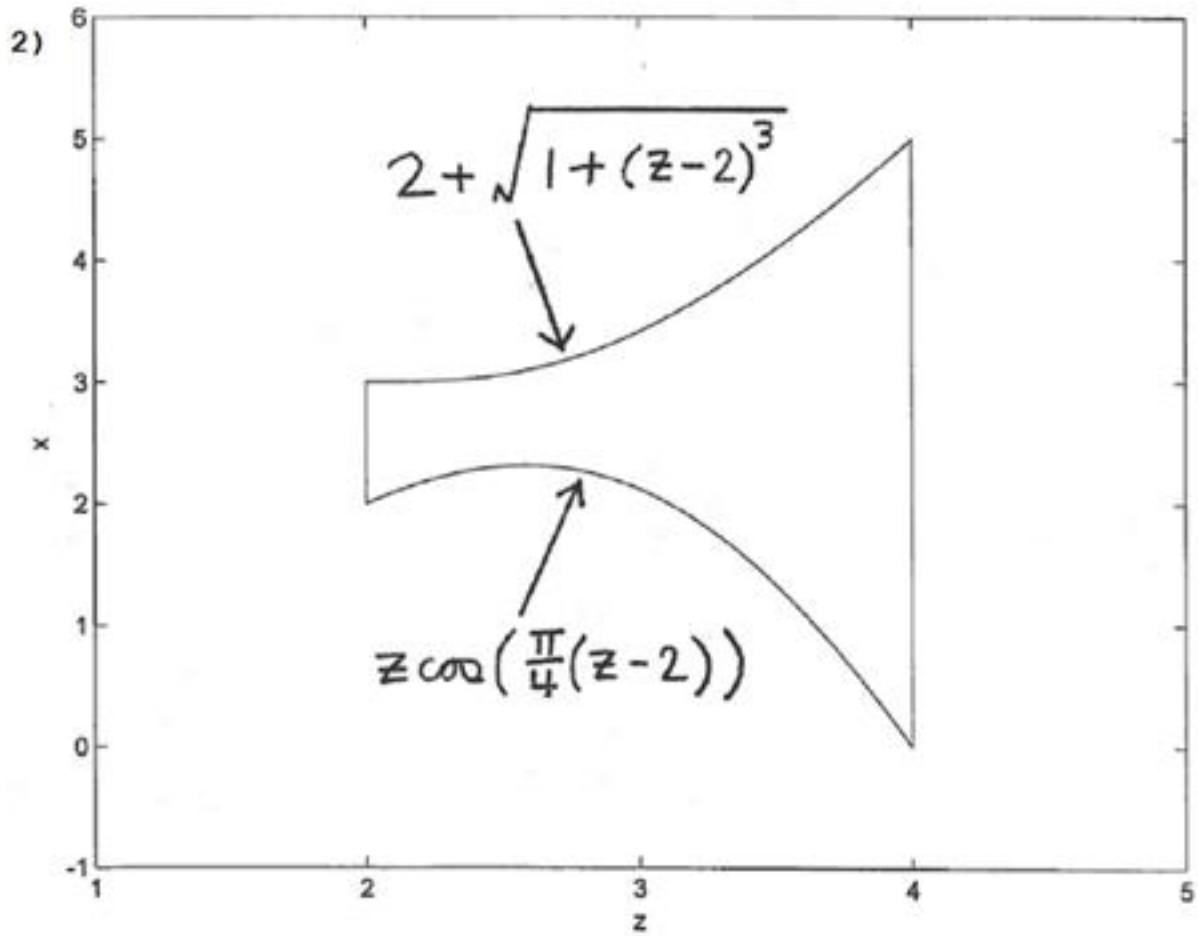
mass=2482.52707

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

Question 2

Complete Points out of 1.00 Flag question

2)



A solid is bounded in the z and x directions by the region shown above and is bounded below and above in the y direction by the planes y=1 and y=x+z+3. The density of the solid is

$$\rho = \frac{y^4\cos(x+y)}{2y^2+3z+1} + e^{\frac{4y+3z}{5x+2y+1}}$$

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:

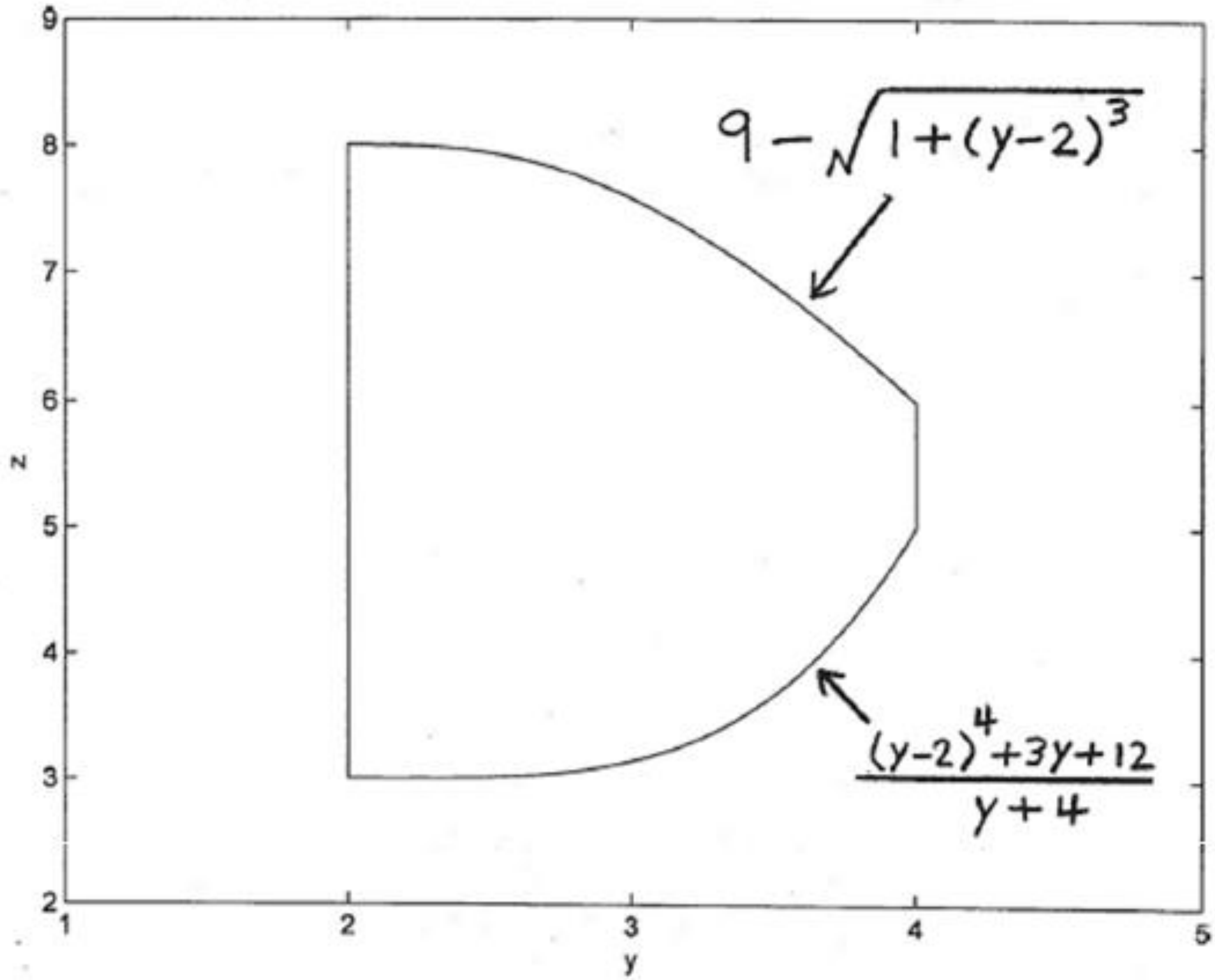
mass=90.816

```
global accuracy;
a = 2;
b = 4;
accuracy = 1e-4;
g = @(z) z*cos(pi/4*(z-2));
h = @(z) 2 + sqrt(1 + (z-2)^3);
v = @(x,z) 1;
w = @(x,z) x + z + 3;
f = @(y,z,x) y.^4.*cos(x+y)./(2*y.^2+3*z+1)+exp((4*y+3*z)./(5*x+2*y+1));
mass = quad('middle',a,b,accuracy,[],'inner',g,h,f,v,w);
fprintf('mass=%.3f \n', mass);
```

Question 3

Complete Points out of 1.00 Flag question

3)



A solid is bounded in the y and z directions by the region shown above and is bounded below and above in the x direction by the planes x=8 and x=y+z+4 . The density of the solid is given by

$$\rho = \frac{x^4 \sin(x+y+z)}{5x+4y+2} + x^3 \ln\left(x\sqrt{x+y^2+z^3}\right)$$

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=202529.969

```
global accuracy;
a = 2;
b = 4;
accuracy = 1e-4;
g = @(y) ((y-2)^4+3*y+12)/(y+4);
h = @(y) 9 - sqrt(1+(y-2)^3);
v = @(y,z) 8;
w = @(y,z) y+z+4;
f = @(x, y, z) x.^4.*sin(x+y+z)./(5*x+4*y+2) + x.^3.*log(x.*sqrt(x+y.^2+z.^3));
mass = quad('middle', a, b, accuracy, [], 'inner', g, h, f, v, w);
fprintf('mass=%.3f \n', mass);
```

Question 4 Complete Points out of 1.00 Flag question

4) Consider the following Two-Dimensional Heat Equation for u(x,y,t) for 0 = x = 3, 0 = y = 3, and 0 = t = .4:

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

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

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

with the following initial conditions at t=0:

u(x,y,0) = u0(x,y) = x^2 + y^2 - 1 + cos(π/2*(x+y))

and the following Dirichlet boundary conditions:

u(x,0,t) = gbottom(x,t) = 0 , u(x,3,t) = gtop(x,t) = 0

u(0,y,t) = gleft(y,t) = 0 , u(3,y,t) = gright(y,t) = 0

NOTE: In this problem, the x and y intervals have the same length, the same number of grid points, and the same stepsize. Use the variables L, n and h for the length, number of grid points and stepsize in the x and y intervals, and the variables T, nt and ht for the length, number of grid points and stepsize in the t interval.

NOTE: There will be a function named u0(x,y) that gives the initial conditions at t=0, and 4 functions, named gbottom(x,t), gtop(x,t), gleft(y,t) and gright(y,t), that give the boundary conditions on the bottom (y=0), top (y=L), left (x=0) and right (x=L) sides of the square.

Write a MATLAB program as follows:

Use the explicit full discretization scheme to calculate numerical values for the unknown u(x,y,t) for 0 < x < 3, 0 < y < 3 and 0 < t = .4 . The main program will call a function named heat3 that solves the Two-Dimensional Heat Equation for the unknown u and returns it to the main program.

The first line of heat3 is:

```
function u = heat3(f,u0,gbottom,gtop,gleft,gright,a,n,nt,L,T)
```

where a is the coefficient and f(x,y,t) is the function in the Two-Dimensional Heat Equation. The other parameters of heat3 are defined above.

Do not write the main program.

Just write the first part of the function heat3 that defines u(i,j,k) at the initial time t=0 and along the bottom, top, left and right boundaries of the rectangle. The first part of the function goes from the beginning of the function until right before the following for loop: for(k = 2:nt).

```
function u = heat3(f,u0,gbottom,gtop,gleft,gright,a,n,nt,L,T)
h = L/(n-1);
ht = T/(nt-1);
u = zeros(n, n, nt);
for (i = 1 : n)
for (j = 1 : n)
u (i, j, 1) = u0((i-1)*h, (j-1)*h);
end
end
for (k = 2 : nt)
for(i = 1 : n)
u(i, 1, k) = gbottom((i-1)*h, (k-1)*ht);
u(i, n, k) = gtop((i-1)*h, (k-1)*ht);
u(1, i, k) = gleft((i-1)*h, (k-1)*ht);
u(n, i, k) = gright((i-1)*h, (k-1)*ht);
end
end
```

Question 5

Complete Points out of 1.00 Flag question

5) Consider the following Two-Dimensional Heat Equation for u(x,y,t) for 0 = x = 3, 0 = y = 3, and 0 = t = .4:

4b

4b

4d 4e

Just write the last part of the function heat3 that calculates u(i,j,k) at the interior grid points. The last part of the function starts at the following for loop: for(k = 2:nt) and goes until the end of the function.

```
for (k = 2 : nt)
for (i = 2 : n-1)
for (j = 2 : n-1)
u(i, j, k) = ht*a/h^2*(u(i-1, j, k-1)+u(i+1, j, k-1)+u(i, j-1, k-1)+u(i, j+1, k-1)-4*u(i, j, k-1))+ht*f((i-1)*h, (j-1)*h, (k-2)*ht) + u(i, j, k-1);
end
end
end
```

Question 6

Complete Points out of 1.00 Flag question

6) Consider the following Two-Dimensional Wave Equation for u(x,y,t)

for $0 \le x \le 2$, $0 \le y \le 1$, and $0 \le t \le .1$:

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

with the following initial conditions at t=0:

u(x,y,0) = u0(x,y) = sin(pi/2*x)*sin(pi*y)+x^2*y^2+2*x*(1-y)

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

and the following Dirichlet boundary conditions:

u(x,0,t) = gbottom(x,t) = 2*x*(1-t^2)

u(x,1,t) = gtop(x,t) = x^2*(1+sin(pi*t))

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

u(2,y,t) = gright(y,t) = (4*y^2+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 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). Use the variable T for the length of the t interval, the variable nt for the number of grid points in the t interval, and the variable ht for the stepsize in the t interval.

NOTE: There will be two functions, named u0(x,y) and v0(x,y), that give the
initial conditions at t=0, and 4 functions, named gbottom(x,t), gtop(x,t), gleft(y,t) and gright(y,t), that give the boundary conditions on the bottom (y=0), top (y=1), left (x=0) and right (x=2) sides of the rectangle, respectively.
Write a MATLAB program as follows:

Use the explicit full discretization scheme to calculate numerical values for
the unknown u(x,y,t) for $0 < x < 2$, $0 < y < 1$ and $0 < t \le .1$. The main program will call a function named wave2 that solves the Two-Dimensional Wave Equation for the unknown u and returns it to the main program.
The first line of wave2 is:

```
function u = wave2(f,u0,v0,gbottom,gtop,gleft,gright,a,nx,ny,nt,Lx,Ly,T)
```

where a is the coefficient and f(x,y,t) is the function in the Two-Dimensional Wave Equation, and the other parameters of wave2 are defined above.

Do not write the main program.

Just write the first part of the function wave2 that defines u(i,j,k) at the initial time t=0 and along the bottom, top, left and right boundaries of the rectangle. The first part of the function goes from the beginning of the function until right before the following statement: k=2.

```
function u = wave2(f, u0, v0, 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
```

Question 7

Complete Points out of 1.00 Flag question

The same as question 6 except:

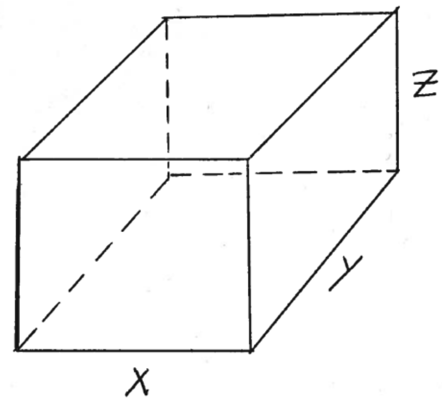
Just write the last part of the function wave2 that calculates u(i,j,k) at the interior grid points. The last part of the function starts at the following statement: k=2 and goes until the end of the function.

```
k=2;
for (i = 2: nx-1)
for (j = 2: ny-1)
u(i,j,k) = ht^2*a/hx^2*(u(i-1,j,k-1)-2*u(i,j,k-1)+u(i+1,j,k-1))+ht^2*a/hy^2(u(i,j-1,k-1)-2*u(i,j,k-1)+u(i,j+1,k-1))+ht^2*f((i-1)*hx, (j-1)*hy, (k-2)*ht) + u(i,j, k-1)+ht*v0((i-1)*hx, (j-1)*hy);
end
end
for (k = 3:nt)
for (i = 2: nx-1)
for (j = 2: ny-1)
u(i,j,k) = ht^2*a/hx^2*(u(i-1,j,k-1)-2*u(i,j,k-1)+u(i+1,j,k-1))+ht^2*a/hy^2(u(i,j-1,k-1)-2*u(i,j,k-1)+u(i,j+1,k-1))+ht^2*f((i-1)*hx, (j-1)*hy, (k-2)*ht) + 2*u(i,j, k-1) - u(i,j, k-2);
end
end
end
```

8) Consider the following 3-dimensional Poisson equation for u(x,y,z) for $0 \le x \le 1$, $0 \le y \le 1.5$, $0 \le z \le .75$:

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

$$f(x,y,z) = (x^2+y^2+z^2)e^{xyz}$$



with the following Dirichlet boundary conditions:

gbottom(x,y) = 1 , gleft(y,z) = e^{yz} , gfront(x,z) = x+z

gtop(x,y) = 1 , gright(y,z) = yz , gback(x,z) = 0

NOTE: The x, y and z intervals have different lengths in this problem (the length of the x interval is 1, the length of the y interval is 1.5, and the length of the z interval is .75). Use the variables Lx, Ly and Lz for the lengths of the x, y and z intervals, nx, ny and nz for the number of grid points in the x, y and z intzervals (the number of grid points is different in the x, y and z intervals), and hx, hy and hz for the stepsizes in the x, y and z intervals (the stepsize is different in the x, y and z intervals).

NOTE: There will be 6 functions, named gbottom(x,y), gtop(x,y), gleft(y,z), gright(y,z), gfront(x,z) and gback(x,z), that give the boundary conditions on the bottom (z=0), top (z=Lz), left (x=0), right (x=Lx), front (y=0) and back (y=Ly) sides of the rectangular solid.

Write a MATLAB program as follows:

Use the 7-point scheme to calculate numerical values for the unknown u(x,y,z) for 0 < x < 1, 0 < y < 1.5 and 0 < z < .75 . Use 1e-8 as the accuracy factor. The main program will call a function named poisson3 that solves the Poisson Equation in this problem for the unknown u and returns it to the main program. The first line of poisson3 is:

```
function u = poisson3(f,gbottom,gtop,gleft,gright,gfront,gback,nx,ny,nz,
                      Lx,Ly,Lz,accuracy)
```

where f(x,y,z) is the function in the Poisson Equation, accuracy is the accuracy factor, and the other parameters of poisson3 are defined above.

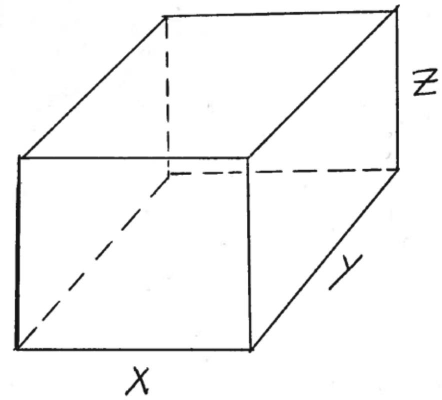
Do not write the main program.

Just write the first part of the function poisson3 that defines u(i,j,k) along the bottom, top, left, right, front and back boundaries of the rectangular solid. The first part of the function goes from the beginning of the function until right before the following statement: max_diff=1.

9) Consider the following 3-dimensional Poisson equation for u(x,y,z) for $0 \le x \le 1$, $0 \le y \le 1.5$, $0 \le z \le .75$:

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

$$f(x,y,z) = (x^2+y^2+z^2)e^{xyz}$$



with the following Dirichlet boundary conditions:

gbottom(x,y) = 1 , gleft(y,z) = e^{yz} , gfront(x,z) = x+z

gtop(x,y) = 1 , gright(y,z) = yz , gback(x,z) = 0

NOTE: The x, y and z intervals have different lengths in this problem (the length of the x interval is 1, the length of the y interval is 1.5, and the length of the z interval is .75). Use the variables Lx, Ly and Lz for the lengths of the x, y and z intervals, nx, ny and nz for the number of grid points in the x, y and z intzervals (the number of grid points is different in the x, y and z intervals), and hx, hy and hz for the stepsizes in the x, y and z intervals (the stepsize is different in the x, y and z intervals).

NOTE: There will be 6 functions, named gbottom(x,y), gtop(x,y), gleft(y,z), gright(y,z), gfront(x,z) and gback(x,z), that give the boundary conditions on the bottom (z=0), top (z=Lz), left (x=0), right (x=Lx), front (y=0) and back (y=Ly) sides of the rectangular solid.

Write a MATLAB program as follows:

Use the 7-point scheme to calculate numerical values for the unknown u(x,y,z) for $0 < x < 1$, $0 < y < 1.5$ and $0 < z < .75$. Use 1e-8 as the accuracy factor. The main program will call a function named poisson3 that solves the Poisson Equation in this problem for the unknown u and returns it to the main program. The first line of poisson3 is:

```
function u = poisson3(f,gbottom,gtop,gleft,gright,gfront,gback,nx,ny,nz,
                      Lx,Ly,Lz,accuracy)
```

where f(x,y,z) is the function in the Poisson Equation, accuracy is the accuracy factor, and the other parameters of poisson3 are defined above.

Do not write the main program.

Just write the last part of the function poisson3 that calculates u(i,j,k) at the interior grid points. The last part of the function starts at the following statement: max_diff=1 and goes until the end of the function.

8.

```
function u = poisson3(f, gbottom, gtop, gleft, gright, gfront, gback, nx, ny,
nz, Lx, Ly, Lz, accuracy)

hx = Lx/(nx-1);

hy = Ly/(ny-1);

hz = Lz/(nz-1);

u = zeros(nx, ny, nz);

for (i = 1 : nx)

for (j = 1 : ny)

end

u(i, j, 1) = gbottom((i-1)*hx, (j-1)*hy);

u(i, j, nz) = gtop((i-1)*hx, (j-1)*hy);

end

for (i = 1 : nx)

for (k = 1 : nz)

u(i, 1, k) = gfront((i-1)*hx, (k-1)*hz);

u(i, ny, k) = gback((i-1)*hx, (k-1)*hz);

end

end

for (j = 1 : ny)

for (k = 1 : nz)

u(1, j, k) = gleft((j-1)*hy, (k-1)*hz);

u(nx, j, k) = gright((j-1)*hy, (k-1)*hz);

end

end
```

9.

```
max_diff = 1;

while(max_diff >= accuracy)

max_diff = 0;

for(i = 2: nx - 1)

for(j = 2 : ny - 1)

for(k = 2: nz - 1)

uijk_old=u(i,j,k);

u(i,j,k) = (hx^2*hy^2*hz^2*f((i-1)*hx, (j-1)*hy, (k-1)*hz) - hy^2*hz^2*(u(i-
1,j,k) + u(i+1,j,k)) - hx^2*hz^2*(u(i,j-1,k) + u(i,j+1,k)) -
hx^2*hy^2*(u(i,j,k-1) + u(i+1,j,k+1)) )/(-2*(hy^2*hz^2+hx^2*hz^2+hx^2*hy^2));

diff = abs(u(i,j,k)-uijk_old);

if(diff > max_diff) max_diff = diff;

end

end

end

end
```