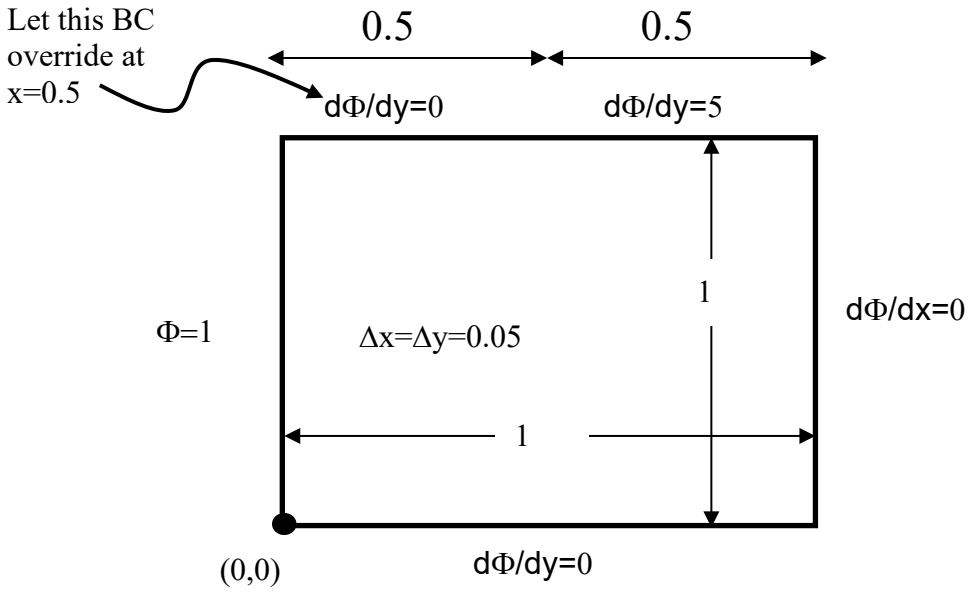
BME 7310 Computational Laboratory #4 Solution

Due: October 3, 2023

Problem #1.

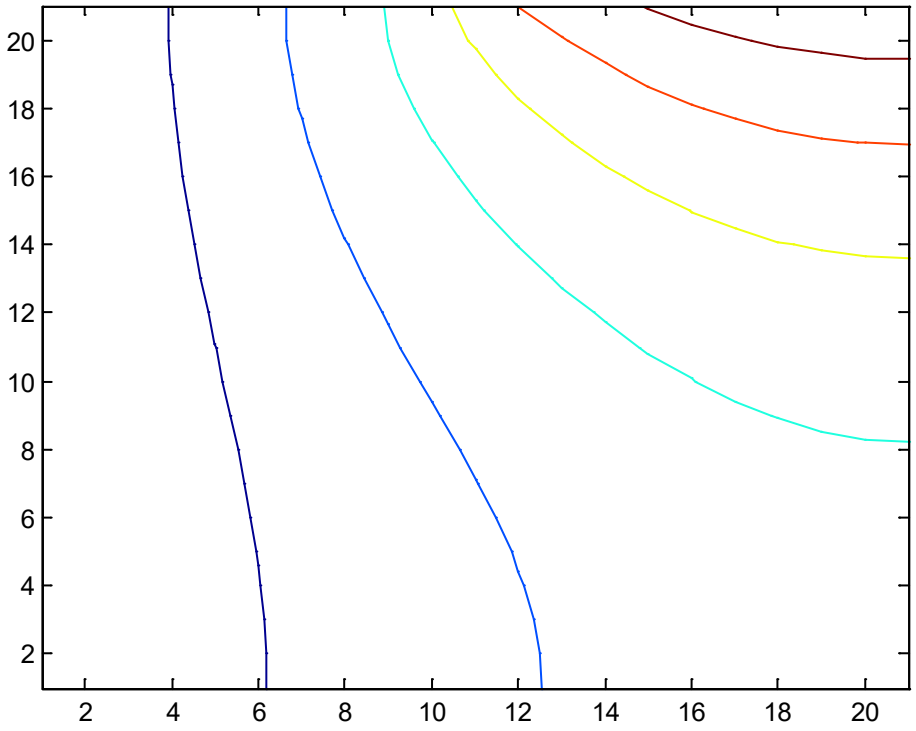
(a) Now let's look at a more interesting problem. Using **the point iterative method of your choice**. Calculate the solution to the following geometry and: plot using the contour command, report the number of iterations, and report the value for x=0.7, y=0.7.



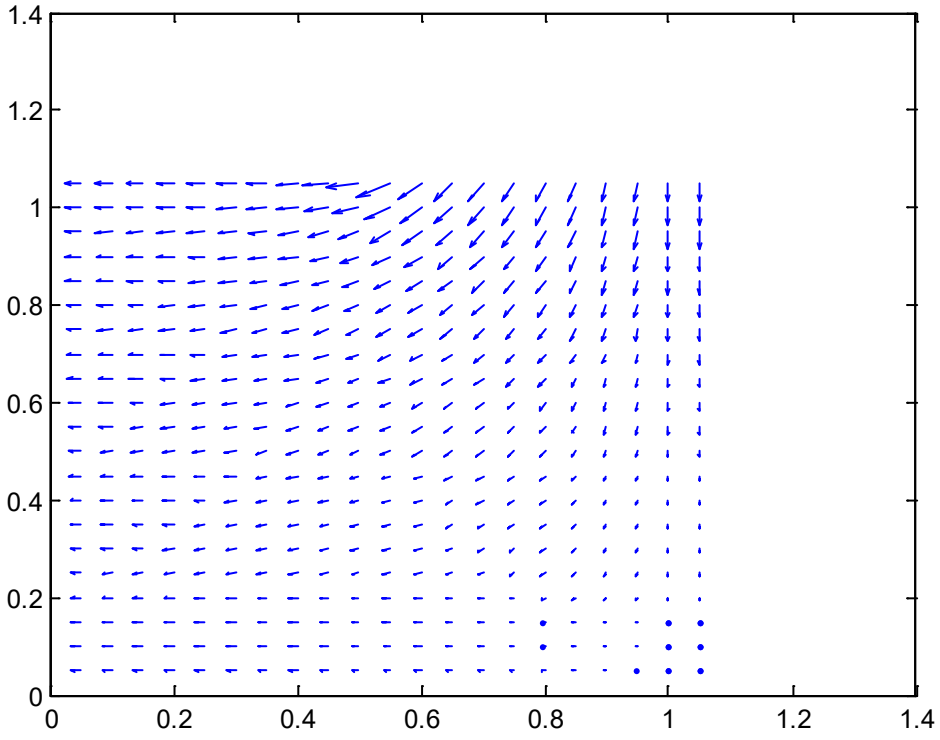
Using Jacobi's method it needed 3211 iterations to converge. The value at x=0.7, y=0.7 is Φ= 2.9234

```
clear;
h=0.05;
y=[0:.05:1]';

A=zeros(21,21);
for i=1:21
  A(i,1)=1;
end

error=1;
itr=0;
while (error > 1e-5 & itr < 10000)
   itr=itr+1;
   Aold=A;


A(2:20,2:20)=1/4*(Aold(1:19,2:20)+Aold(3:21,2:20)+Aold(2:20,1:19)+Aold(2:20,3
:21));

   i=1;
   for j=2:20
```

```
      A(i,j)=1/4*(2*Aold(i+1,j)+Aold(i,j+1)+Aold(i,j-1));
    end

    i=21;
    for j=2:20
     if j<=10
       A(i,j)=1/4*(2*Aold(i-1,j)+Aold(i,j+1)+Aold(i,j-1));
     end
     if j>10
       A(i,j)=1/4*(2*Aold(i-1,j)+10*h+Aold(i,j+1)+Aold(i,j-1));
     end
    end

    j=21;
    for i=2:20
      A(i,j)=1/4*(Aold(i-1,j)+Aold(i+1,j)+2*Aold(i,j-1));
    end

    A(1,21)=1/4*(2*Aold(1,20)+2*Aold(2,21));
    A(21,21)=1/4*(2*Aold(20,21)+2*Aold(21,20)+10*h);
    errorold=error;
    error=mean(mean(abs(A-Aold)))/mean(mean(abs(A)));
    errornew=error;
    spectral(i)=errornew/errorold;
end

 Vx=-(A(2:20,3:21)-A(2:20,1:19))/2/h;
 Vy=-(A(3:21,2:20)-A(1:19,2:20))/2/h;

 c1=0;
 for i=1:19
   for j=1:19
    c1=c1+1;
     xpos(c1)=j*.05;
     ypos(c1)=i*.05;
     vx(c1)=Vx(i,j);
     vy(c1)=Vy(i,j);
     PHI(c1)=A(i,j);
   end
 end

figure(1);
contour(A);
figure(2);
quiver(xpos,ypos,vx,vy);
```


20
18
16
14
12
10
8
6
4
2
2
4
6
8
10
12
14
16
18
20

A


1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4

B

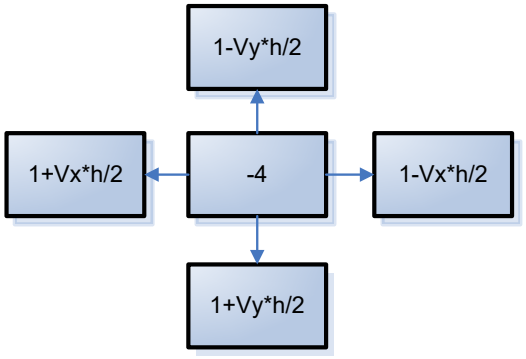
A contour in the **Figure A** above represents a line in which Φ does not change so if you were to take the differential along this contour you would get no change, i.e the gradient is zero. Similarly if you move perpendicular to the contour towards the next contour, the value of Φ would change and as a result the gradient operation would be non-zero; hence, what we observe in **Figure B** is that the arrows represent the gradient of Φ and are perpendicular to the isocontours.

With respect to boundary conditions, everywhere we specify that $d\Phi/dn=0$ where n is the respective unit vector normal to the surface, this means that gradient of Φ across this boundary must be zero, i.e. the arrow must not transect the boundary. This boundary condition is often referred to as a no flux condition meaning that flow of the potential (Φ in this case) does not cross this boundary. With respect to Φ=1, this just specifies that the potential is uniform across this region and in the top right where the flux is specified, this indicates the value of that specific change across the boundary. We should also note that this flux has been specified as a positive quantity which means that Φ must be increasing across the boundary. Given this fact, it makes sense that the values must increase from the top right boundary to the left in order to satisfy the boundary conditions.

Problem #2: Now let's study the advective diffusion equation of a chemical species released into the potential flow stream derived from part (b) of Problem 2. The advective-diffusive equation can be written as:

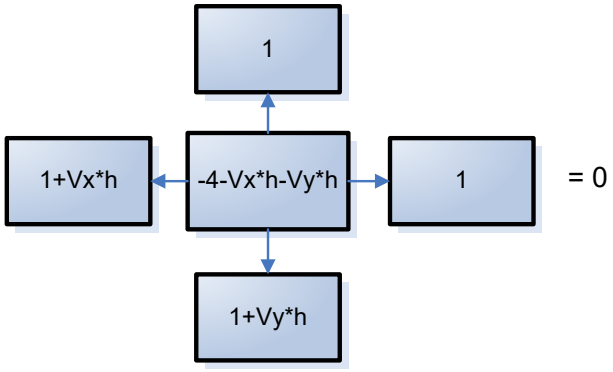
$$\nabla^2 C - \vec{v} \bullet \nabla C = 0 \qquad (5)$$

(a) Write out the finite difference molecule using three methods of handling the advective term (center difference, upstream weighting, and downstream weighting). Note, you will need to perform 2-D weighting so your result from Problem 2b will be important in this.
General computational molecule for Center Difference form:



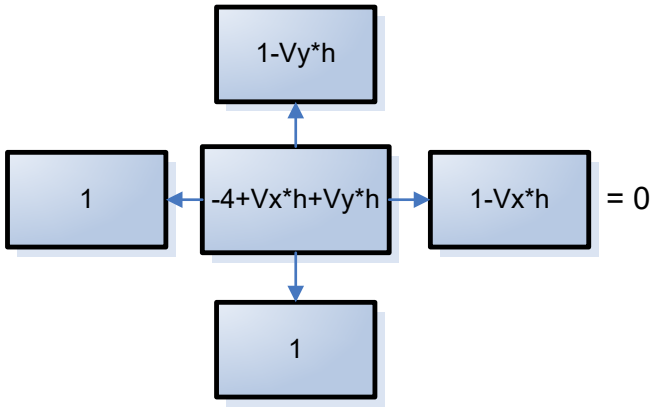
= 0

General computational molecule for Downstream Weighting form. Remember the idea of downstream and upstream is related to the direction of the flow. In this case downstream represents flow to the left and down:

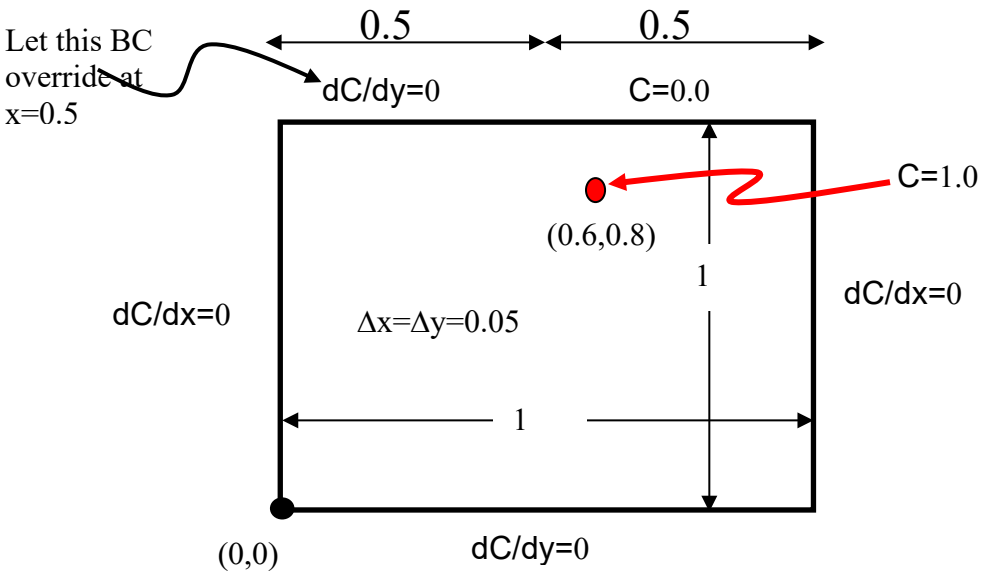


= 0

General computational molecule for Upstream Weighting form:



(b) Now for this problem, we are going to solve the problem below a total of 9 times. For the first 3 solutions, use each of the FD expressions found in (a) and use the flow field found in Part 2b. For the second 3 solutions, resolve the problem from problem 2 but increase the flux on the upper right boundary to $d\Phi/dy=25$, now use this resulting flow field to solve the below with each FD expression. For the last 3 solutions, resolve problem 2 but increase the flux to $d\Phi/dy=50$, now use this resulting flow field to solve the below with each FD expression. Looking at the results of all 9 graphs, discuss how this behavior is consistent with the discussion we had about the advective diffusive equation. HINT: Once you have established your method to solve the below. One strategy to handle the source in the middle of the domain is to set the value to C=1, solve the whole domain as normal for 1 iteration, then reset that value to 1 before starting the next iteration..



So the question with this problem is what to expect. According to the simple problem in the book:

$$\textbf{Upstream} \quad (Pe)_h = \frac{V\Delta x}{D} < \infty$$

$$\textbf{Centered} \quad (Pe)_h = \frac{V\Delta x}{D} < 2$$

$$\textbf{Downstream} \quad (Pe)_h = \frac{V\Delta x}{D} < 1$$

Although not necessarily the case for 2D work, we can use this as a guide. In our system, the maximum V is approximately 5, 25, and 50 as dictated by the flux designation in 2b. Δx is 0.05 and D =1. Evaluating this for our problem would yield:

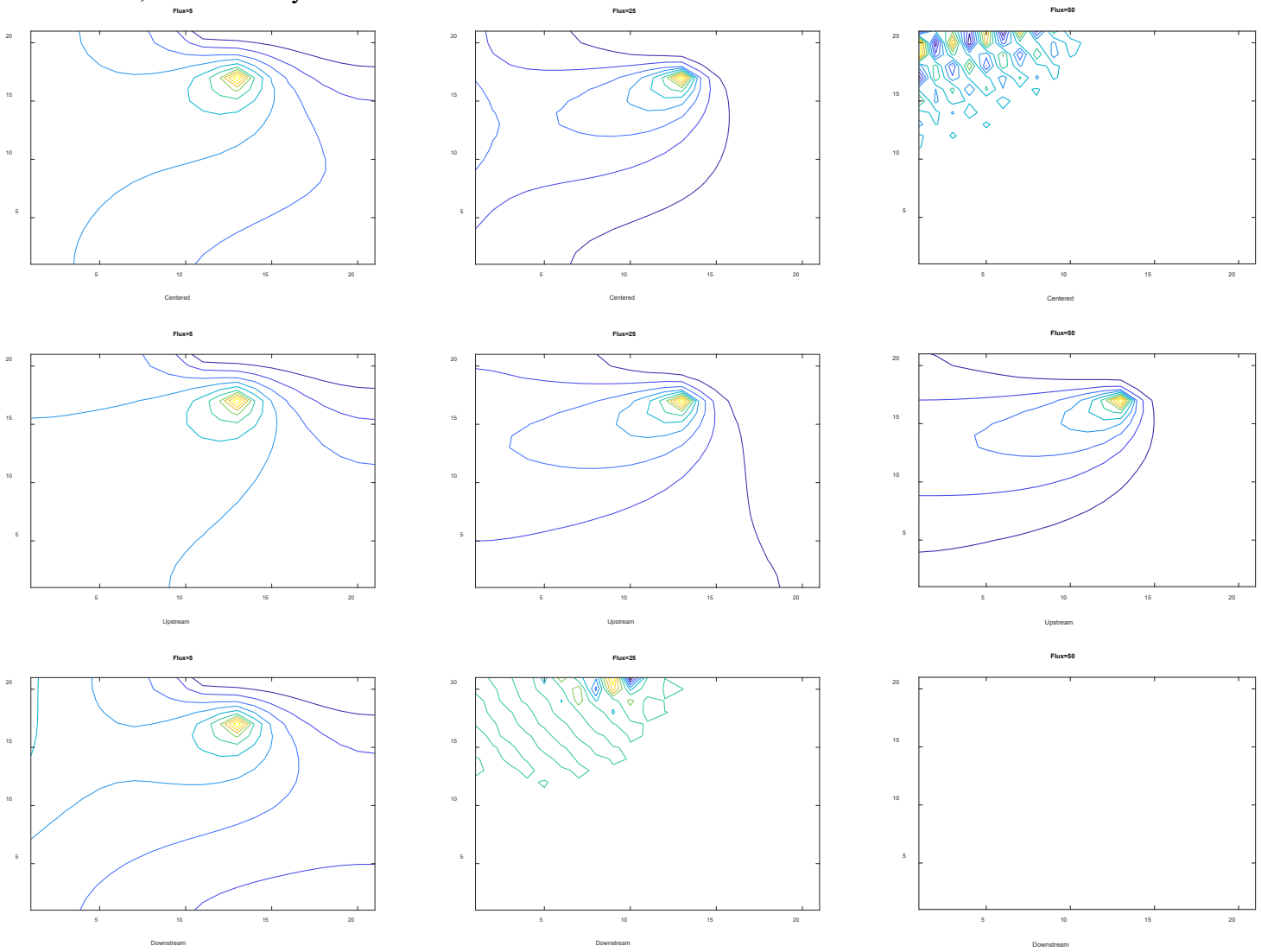
$$\textbf{Flux of 5} \quad (Pe)_h = \frac{V\Delta x}{D} = \frac{5(0.05)}{1} = 0.25$$

$$\textbf{Flux of 25} \quad (Pe)_h = \frac{V\Delta x}{D} = 1.25$$

$$\textbf{Flux of 50} \quad (Pe)_h = \frac{V\Delta x}{D} = 2.5$$

With those numbers, you might expect that all methods could handle of velocity of 5. Only Upstream and Centered could handle a velocity of 25. Only Upstream could handle the velocity of 50.

As we see here, this is exactly the behavior seen.



```
%Advection-Diffusion
 clear;
%Following file has all the velocities
load vel_grid.mat;
Vx=Vx_5;
Vy=Vy_5;
h=0.05;
y=[0:.05:1]';
B=zeros(441,1);

% I chose a different strategy to do this problem.  Below you see a
% function call to designate a sparse matrix
Acd=spalloc(441,441,441*6);
Aus=spalloc(441,441,441*6);
Ads=spalloc(441,441,441*6);

c1=0;
for i=1:21
    for j=1:21
    c1=c1+1;
    A(i,j)=c1;
    end
end

% Rather than cycling through the grid with SOR, I chose to construct a
% matrix with the domain, and then constructed an SOR matrix solver.
c1=0;
for i=1:21
    for j=1:21
      c1=c1+1;
      Acd(c1,A(i,j))=4/h/h;
      Aus(c1,A(i,j))=4/h/h-(Vx(i,j)+Vy(i,j))/h;
      Ads(c1,A(i,j))=4/h/h+(Vx(i,j)+Vy(i,j))/h;
      if (j+1)<=21
        Acd(c1,A(i,j+1))=-1/h/h+(Vx(i,j))/2/h;
        Aus(c1,A(i,j+1))=-1/h/h+(Vx(i,j))/h;
        Ads(c1,A(i,j+1))=-1/h/h;
      end
      if (j-1)>=1
        Acd(c1,A(i,j-1))=-1/h/h-(Vx(i,j))/2/h;
        Aus(c1,A(i,j-1))=-1/h/h;
        Ads(c1,A(i,j-1))=-1/h/h-(Vx(i,j))/h;
      end
      if (i+1)<=21
        Acd(c1,A(i+1,j))=-1/h/h+(Vy(i,j))/2/h;
        Aus(c1,A(i+1,j))=-1/h/h+(Vy(i,j))/h;
        Ads(c1,A(i+1,j))=-1/h/h;
      end
      if (i-1)>=1
        Acd(c1,A(i-1,j))=-1/h/h-(Vy(i,j))/2/h;
        Aus(c1,A(i-1,j))=-1/h/h;
        Ads(c1,A(i-1,j))=-1/h/h-(Vy(i,j))/h;
      end
    end
end
```

```
i1=1;
i2=21;
for j=1:21
    Acd(A(i1,j),A(i1+1,j))=Acd(A(i1,j),A(i1+1,j))-1/h/h-(Vy(i,j))/2/h;
    if j<11
      Acd(A(i2,j),A(i2-1,j))=Acd(A(i2,j),A(i2-1,j))-1/h/h+(Vy(i,j))/2/h;
    end
    Aus(A(i1,j),A(i1+1,j))=Aus(A(i1,j),A(i1+1,j))-1/h/h;
    if j<11
      Aus(A(i2,j),A(i2-1,j))=Aus(A(i2,j),A(i2-1,j))-1/h/h+(Vy(i,j))/h;
    end
    Ads(A(i1,j),A(i1+1,j))=Ads(A(i1,j),A(i1+1,j))-1/h/h-(Vy(i,j))/h;
    if j<11
      Ads(A(i2,j),A(i2-1,j))=Ads(A(i2,j),A(i2-1,j))-1/h/h;
    end
end
j1=1;
j2=21;
for i=1:21
    Acd(A(i,j1),A(i,j1+1))=Acd(A(i,j1),A(i,j1+1))-1/h/h-(Vx(i,j))/2/h;
    if i<21
     Acd(A(i,j2),A(i,j2-1))=Acd(A(i,j2),A(i,j2-1))-1/h/h+(Vx(i,j))/2/h;
    end
    Aus(A(i,j1),A(i,j1+1))=Aus(A(i,j1),A(i,j1+1))-1/h/h;
    if i<21
     Aus(A(i,j2),A(i,j2-1))=Aus(A(i,j2),A(i,j2-1))-1/h/h+(Vx(i,j))/h;
    end
    Ads(A(i,j1),A(i,j1+1))=Ads(A(i,j1),A(i,j1+1))-1/h/h-(Vx(i,j))/h;
    if i<21
     Ads(A(i,j2),A(i,j2-1))=Ads(A(i,j2),A(i,j2-1))-1/h/h;
    end
end

Acd(A(17,13),:)=Acd(A(17,13),:)*0;
Acd(A(17,13),A(17,13))=1;
Aus(A(17,13),:)=Aus(A(17,13),:)*0;
Aus(A(17,13),A(17,13))=1;

B(A(17,13))=1;

i=21;
for j=11:21
    Acd(A(i,j),:)=Acd(A(i,j),:)*0;
    Acd(A(i,j),A(i,j))=1;
    Aus(A(i,j),:)=Aus(A(i,j),:)*0;
    Aus(A(i,j),A(i,j))=1;
    Ads(A(i,j),:)=Ads(A(i,j),:)*0;
    Ads(A(i,j),A(i,j))=1;
end

[xcd]=sor_solver(Acd,B,zeros(441,1),1.72,1000,1e-5);
[xus]=sor_solver(Aus,B,zeros(441,1),1.72,1000,1e-5);
[xds]=sor_solver(Ads,B,zeros(441,1),1.72,1000,1e-5);

BBcd=zeros(21,21);
BBus=zeros(21,21);
```

```
BBds=zeros(21,21);
c1=0;
for i=1:21
  for j=1:21
   c1=c1+1;
   BBcd(i,j)=xcd(c1);
   BBus(i,j)=xus(c1);
   BBds(i,j)=xds(c1);
  end
end

figure(1);
subplot(3,1,1)
contour(BBcd,10);
xlabel('Centered');
title('Flux=5');
subplot(3,1,2)
contour(BBus,10);
xlabel('Upstream');
title('Flux=5');
subplot(3,1,3)
contour(BBds,10);
title('Flux=5');
xlabel('Downstream');


function [x]=sor_solver(A,b,xo,omega,itrmax,TOL)

k=1;
n=length(b);
x=zeros(n,1);
while k <= itrmax
    for i=1:n
        S1=0;
        S2=0;
        if i>1
          S1=sum(A(i,1:i-1)'.*x(1:i-1));
        end
        if i<n
          S2=sum(A(i,i+1:n)'.*xo(i+1:n));
        end
        x(i)=(1.0-omega)*xo(i)+ (omega*(-S1-S2+b(i)))/A(i,i);
    end
    error=max(abs(x-xo));
    if error < TOL
       fprintf('Solution convergence in %d iterations\n',k);
       return;
    end
    k=k+1;
    xo=x;
end
fprintf('Maximum number of iterations exceeded');
```

Problem 3. The equation used to describe one-dimensional drug diffusion is:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right)$$

(a) With the above formulation, write out the finite difference expression including the leading truncation error terms. Assume a purely *explicit* formulation.

Expression:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial c}{\partial x}\right)$$

LHS:

$$c^{k+1} = c^k + \Delta t \frac{\partial c^k}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 c^k}{\partial t^2} + \dots$$

$$\frac{\partial c}{\partial t} = \frac{\partial c^k}{\partial t} = \frac{c^{k+1} - c^k}{\Delta t} - \frac{\Delta t}{2}\frac{\partial^2 c^k(\xi_1)}{\partial t^2}$$

where $\xi_1 \in [0, \Delta t]$

LTE for LHS $= -\frac{\Delta t}{2}\frac{\partial^2 c(\xi)}{\partial t^2}$ $\xi \in [0, \Delta t]$

RHS:

$$\frac{\partial}{\partial x}\left(D\frac{\partial c}{\partial x}\right)$$

Let this equal F and expand first derivative about 'i' at half grid size $\Delta x = h/2$

$$F_{i+1/2} - F_{i-1/2} = h\frac{\partial F_i}{\partial x} + \frac{h^3}{24}\frac{\partial^3 F_i(\xi_2)}{\partial x^3}$$

where $\xi_2 \in [-h/2, h/2]$

$$\frac{\partial F_i}{\partial x} = \frac{F_{i+1/2} - F_{i-1/2}}{h} - \frac{h^2}{24}\frac{\partial^3 F_i(\xi_2)}{\partial x^3}$$

Now expand each term:

$$\frac{D_{i+1/2}\left.\frac{\partial C}{\partial x}\right|_{i+1/2} - D_{i-1/2}\left.\frac{\partial C}{\partial x}\right|_{i-1/2}}{h} - \underbrace{\frac{h^2}{24}\frac{\partial^3}{\partial x^3}\left(D_i\frac{\partial C_i(\xi_2)}{\partial x}\right)}_{D_i\frac{h^2}{24}\frac{\partial^4 C_i(\xi_2)}{\partial x^4}}$$

Now expanding each term above with center difference about $i+1/2$, $i-1/2$ respectively using $\Delta x = h/2$

$\xi_2 \in [-h, 0]$

$$\frac{D_{i+1/2}\left(\frac{C_{i+1}-C_i}{h} - \frac{h^2}{24}\frac{\partial^3 C_{i+1/2}(\xi_3)}{\partial x^3}\right)}{h}$$

$\xi_4 \in [0, h]$

$$- \frac{D_{i-1/2}\left(\frac{C_i - C_{i-1}}{h} - \frac{h^2}{24}\frac{\partial^3 C_{i-1/2}(\xi_4)}{\partial x^3}\right)}{h}$$

$$- D_i\frac{h^2}{24}\frac{\partial^4 C_i(\xi_5)}{\partial x^4}$$

So the FDM expression is:

$$\frac{C^{k+1} - C^k}{\Delta t} = \frac{D_{i+1/2}(C_{i+1} - C_i) + D_{i-1/2}(C_{i-1} - C_i)}{\Delta x^2}$$

where $\Delta x = h$

The truncation error is:

- assume homogeneous, $D_{i+1/2} = D_{i-1/2} = D_i = D$
- assume $\xi_1 \approx \xi_2 \approx \xi_3 \approx \xi_4 \approx \xi$

$$D\left(-\frac{h}{24}\frac{\partial^3 C_{i+1/2}(\xi)}{\partial x^3} + \frac{h}{24}\frac{\partial^3 C_{i-1/2}(\xi)}{\partial x^3} - \frac{h^2}{24}\frac{\partial^4 C_i(\xi)}{\partial x^4}\right)$$

rewrite:

$$D\left(-\left[\frac{h}{h}\frac{h}{24}\frac{\partial^3 C_{i+1/2}}{\partial x^3} - \frac{h}{h}\frac{h}{24}\frac{\partial^3 C_{i-1/2}}{\partial x^3}\right] - \frac{h^2}{24}\frac{\partial^4 C_i}{\partial x^4}\right)$$

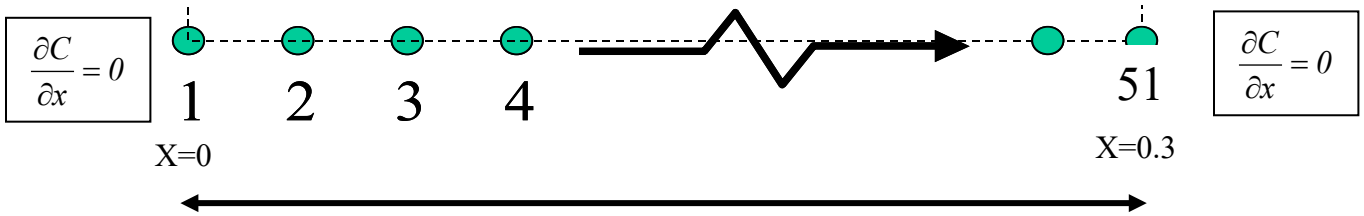
$$D\left(-\left[\frac{h^2}{24}\left(\frac{\frac{\partial^3 C_{i+1/2}}{\partial x^3} - \frac{\partial^3 C_{i-1/2}}{\partial x^3}}{h}\right)\right] - \frac{h^2}{24}\frac{\partial^4 C_i}{\partial x^4}\right)$$

$$= \frac{\partial}{\partial x}\left(\frac{\partial^3 C_i}{\partial x^3}\right)$$

$$D\left(-\frac{h^2}{24}\frac{\partial^4 C_i}{\partial x^4} - \frac{h^2}{24}\frac{\partial^4 C_i}{\partial x^4}\right)$$

SO

$$\text{LTE for RHS} = -D\frac{h^2}{12}\frac{\partial^4 C_i(\xi)}{\partial x^4} \quad \xi \in [-h, h]$$

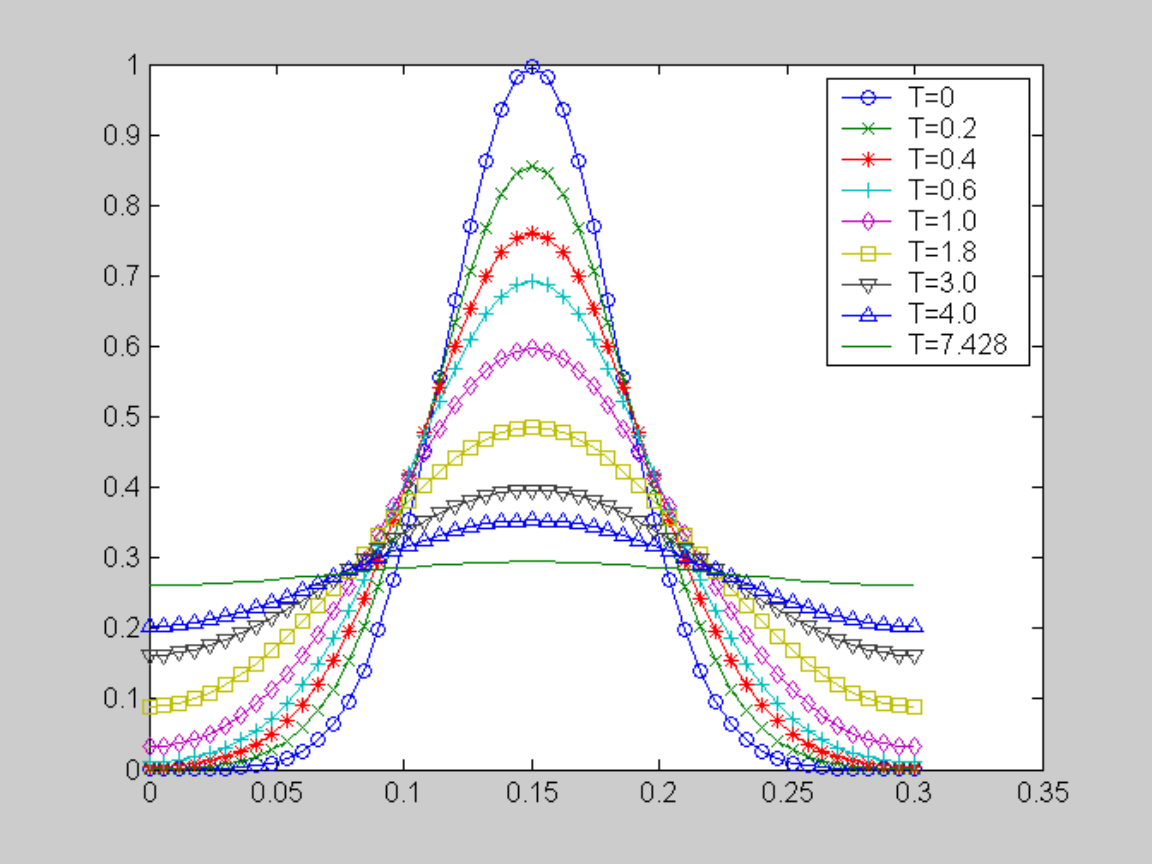


(b) Now let's assume a bolus of some chemoreactant solution has been placed within a column of tissue that represents the initial distribution in the tissue. The bolus is can be respresented as:

$$C(t=0) = \frac{2.5}{\sqrt{2\pi}} e^{-(30x-4.5)^2/2}$$

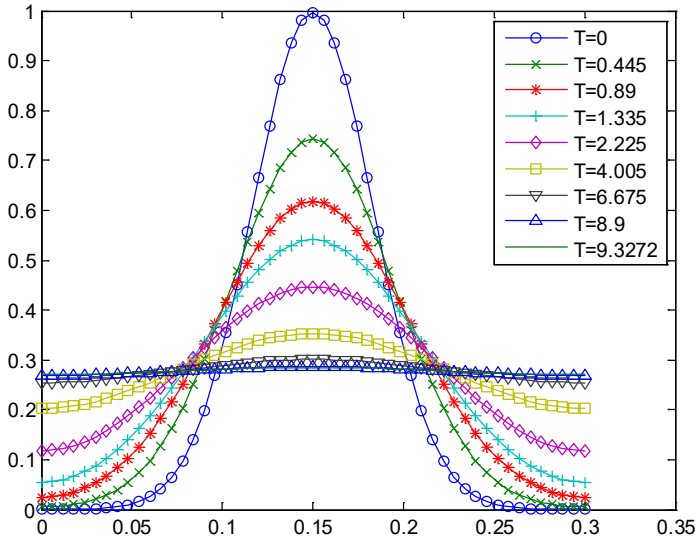
Using the scheme derived in part (a), the initial condition found in the above equation, and the problem description above, generate the time varying solution for this problem. For this problem assume, *dt=0.004, D=0.001*. Allow the process to come to steady state by monitoring a *relative* L_∞ *norm-based error* and stopping the process when this error goes below *1e-4*. Using an overlay function, plot the solutions at *(t=0, 0.2, 0.4, 0.6, 1.0, 1.8, 3,4 and final time at tolerance)*. Also report the number of iterations until convergence.

The system converged to the desired tolerance in 1857 iterations which corresponds to a time of t=7.428.

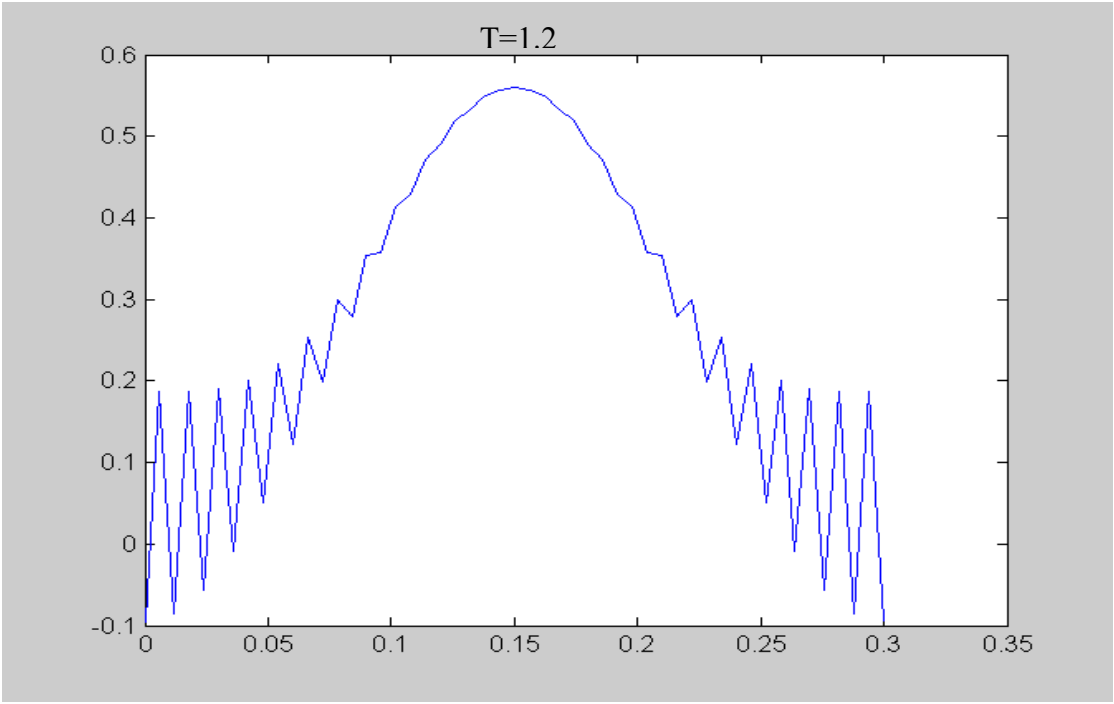


(c) Now, investigate your numerical algorithm from part **(b)**, specifically, does your model go unstable? if so, can you determine a threshold for when instability occurs empirically. Now go back the notes from class, write out the Von Neumann analysis yourself step by step for this model. How do your findings with the Von Neumann analysis compare to your empirical analysis?

According to the Von Neumann analysis in the notes, if the quantity $\frac{D\Delta t}{h^2} < \frac{1}{4}$ we have a stable system. At $\Delta t = 0.0089$, the quantity is 0.2472



According to the Von Neumann analysis in the notes, if the quantity $\frac{D\Delta t}{h^2} > \frac{1}{2}$, instability will occur. At $\Delta t = 0.02$, $\frac{D\Delta t}{h^2} = 0.5556$ and we should get instability.



(d) Looking at the time history of results from part **(b)**, discuss the implication of the boundary conditions. Are there any other interesting points about your solution regarding the nature of how this system works?

Obviously, we see the gradual diffusion associated with this high spike of concentration. The interesting point regarding the boundary conditions is that no-flux conditions have been

specified. This means that the concentration is required to stay within the domain and will ultimately come to a constant level over time. It makes sense when you think about it. The driving force for a diffusion problem is a gradient in concentration. If no gradient exists, the system is at rest. The only resting state is an equal concentration through out the domain. Also, if you were to estimate the area under the curve (AUC) at time *t=0* and compare that to the AUC at *t=steady-state*, you would see that the concentration is conserved within the domain. Neat !!

CODE:

```
x=[0:.006:.3];
h=0.006;
Co=2.5/sqrt(2*pi)*exp(- (30*x-4.5).*(30*x-4.5)/2);

D=0.001;
C=Co;
Cnew=zeros(1,51);
error=1;
tol=1e-4;
% SET DT HERE
dt=0.004;

r=D*dt/h/h;
itr=0;
cntr=0;
plot(x,C);axis([0 0.3 0 1]);
hold;
pcnt=[0;50;100;150;250;450;750;1000];
TC(:,1)=Co';
cntr=cntr+1;
figure(1);

while (error>tol)
   itr=itr+1;
   Cnew(2:50)= C(2:50)+r*(C(1:49)+C(3:51)-2*C(2:50));
   Cnew(1)=C(1)+r*(-2*C(1)+2*C(2));
   Cnew(51)=C(51)+r*(-2*C(51)+2*C(50));

   error=max(abs(Cnew-C))/max(abs(Cnew));
   C=Cnew;
   if (cntr+1)<=8
   if itr==pcnt(cntr+1)
     plot(x,C);axis([0 0.3 0 1]);
     pause(.2);
     cntr=cntr+1;
     TC(:,cntr)=C';
   end
 end
end
FinalTime=itr*dt;
cntr=cntr+1;
pcnt=[pcnt;itr];
plot(x,C);axis([0 0.3 0 1]);
TC(:,cntr)=C';
```

```
tx=x';

figure(2);
plot(tx,TC(:,1),'-o',tx,TC(:,2),'-x',tx,TC(:,3),'-*',tx,TC(:,4),'-
+',tx,TC(:,5),'-d',tx,TC(:,6),'-s',tx,TC(:,7),'-v',tx,TC(:,8),'-
^',tx,TC(:,9));
%Legend when dt=0.004
legend('T=0','T=0.2','T=0.4','T=0.6','T=1.0','T=1.8','T=3.0','T=4.0','T=7.428
');
```