BME 7310 Computational Laboratory #2
Due: 9/15/2023, Midnight
SOLUTION

Problem 1 SOLUTION.
a.

```
Your integral value for Part I is -0.040104
Your integral value for Part II is -10.327510
```

```
load LINE_LIST_1.DAT
in=LINE_LIST_1(:,2:3);
load POINT_LIST_1.DAT;
x=POINT_LIST_1(:,2);
y=POINT_LIST_1(:,3);
Jx=POINT_LIST_1(:,4);
Jy=POINT_LIST_1(:,5);

nn=length(x);

elist=zeros(nn,2);
for i=1:nn
    idx1=find(i==in(:,2));
    idx3=find(i==in(:,1));
    elist(i,1)=in(idx1,1);
    elist(i,2)=in(idx3,2);
end

DX=x(elist(:,2))-x(elist(:,1));
DY=y(elist(:,2))-y(elist(:,1));
S=sqrt(DX.*DX+DY.*DY);
NX=[DY./S];
NY=[-DX./S];

Jn=NX.*Jx+NY.*Jy;

S_line=sqrt((x(in(:,1))-x(in(:,2))).^2+(y(in(:,1))-y(in(:,2))).^2);
Jn1_plus_Jn2=Jn(in(:,1))+Jn(in(:,2));
Line_Integral=sum(0.5*S_line.*(Jn1_plus_Jn2));
fprintf('Your integral value for Part I is %f\n',Line_Integral);

load LINE_LIST_2.DAT
in=LINE_LIST_2(:,2:3);
load POINT_LIST_2.DAT;
x=POINT_LIST_2(:,2);
y=POINT_LIST_2(:,3);
Jx=POINT_LIST_2(:,4);
Jy=POINT_LIST_2(:,5);

nn=length(x);

elist=zeros(nn,2);
for i=1:nn
    idx1=find(i==in(:,2));
    idx3=find(i==in(:,1));
    elist(i,1)=in(idx1,1);
    elist(i,2)=in(idx3,2);
end
```

```
DX=x(elist(:,2))-x(elist(:,1));
DY=y(elist(:,2))-y(elist(:,1));
S=sqrt(DX.*DX+DY.*DY);
NX=[DY./S];
NY=[-DX./S];

Jn=NX.*Jx+NY.*Jy;

S_line=sqrt((x(in(:,1))-x(in(:,2))).^2+(y(in(:,1))-y(in(:,2))).^2);
Jn1_plus_Jn2=Jn(in(:,1))+Jn(in(:,2));
Line_Integral=sum(0.5*S_line.*(Jn1_plus_Jn2));
fprintf('Your integral value for Part II is %f\n',Line_Integral);
```

b.

The following issues needed to be addressed in some form:

1) The divergence theorem is a mathematical statement that in the absence of physical losses or gains (matter or energy, depending on how you want to view it), the net effect of flux through a volume is equivalent to the summation over its boundary surface.
2) An important consequence of the divergence theorem is that a volume integral can be represented as a surface integral because internal fluxes cancel out. Thus, internal sources and/or sinks need only be considered for their effect at the outer surface.
3) In scenario 1, there was no influencing entity within the domain of SpongeBob, so the expected answer would be zero. In scenario 2, there was a source inside the contour, so the expected integral should evaluate to its magnitude (the ground truth was 10). The fact that the analytical values were not achieved can be easily explained as artifacts due to discretization of the contour and interpolation errors in determining the normal vectors to the contour.

sum over surface = net effect of flux through a volume

volume integral = surface integral (since internal fluxes cancels out)

Internal sources / sinks an only be considered for their effects at surface.

c.

Case1: no influence inside domain --> ~zero integration.
Case2: with source inside --> analytical value cant be achieved was due to 1) discretization of contour 2) error in computing normal vector

Since you have seen that $\int_{\ell}(\vec{J}\bullet\vec{n})d\ell = J_{n1}\int\phi_1 d\ell + J_{n2}\int\phi_2 d\ell$

$$\int_1^2(\vec{J}\bullet\vec{n})\phi_1 d\ell = J_{n1}\int\phi_1\phi_1 d\ell + J_{n2}\int\phi_2\phi_1 d\ell$$

$$\int\phi_n\phi_n = L/3 \qquad \text{and} \qquad \int\phi_m\phi_n = L/6$$

then for the first scenario we obtain

$J_{n1}(L/3)+J_{n2}(L/6)$

Likewise,

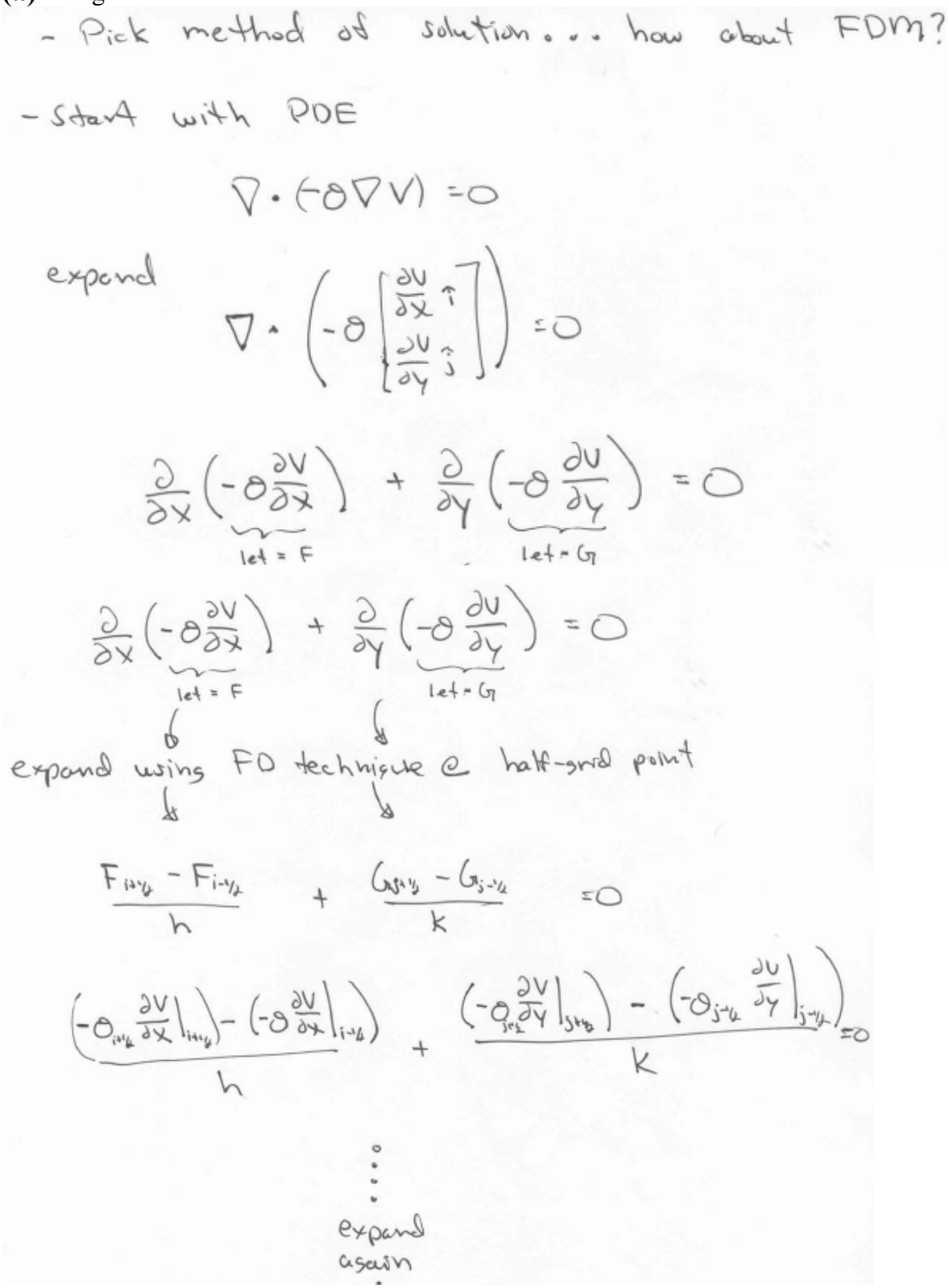
$\int_1^2(\vec{J}\bullet\vec{n})\phi_2 d\ell$ evaluates to

$J_{n1}(L/6)+J_{n2}(L/3)$

The point here was to realize that the interpolating functions apply a different weight to each node depending on its range of effect.

Problem #2 SOLUTION:

(a) Straight out of the notes:



$$\frac{\left(-\theta_{i+\frac{1}{2},j}\left(\frac{V_{i+1,j}-V_{ij}}{h}\right)\right)-\left(-\theta_{i-\frac{1}{2},j}\left(\frac{V_{ij}-V_{i-1,j}}{h}\right)\right)}{h}$$

$$+$$

$$\frac{\left(-\theta_{i,j+\frac{1}{2}}\left(\frac{V_{i,j+1}-V_{ij}}{k}\right)\right)-\left(-\theta_{i,j-\frac{1}{2}}\left(\frac{V_{ij}-V_{i,j-1}}{k}\right)\right)}{k}=0$$

simplify

$$\frac{-\theta_{i+\frac{1}{2},j}V_{i+1,j}+(\theta_{i+\frac{1}{2},j}+\theta_{i-\frac{1}{2},j})V_{ij}-\theta_{i-\frac{1}{2},j}V_{i-1,j}}{h^2}$$

$$+$$

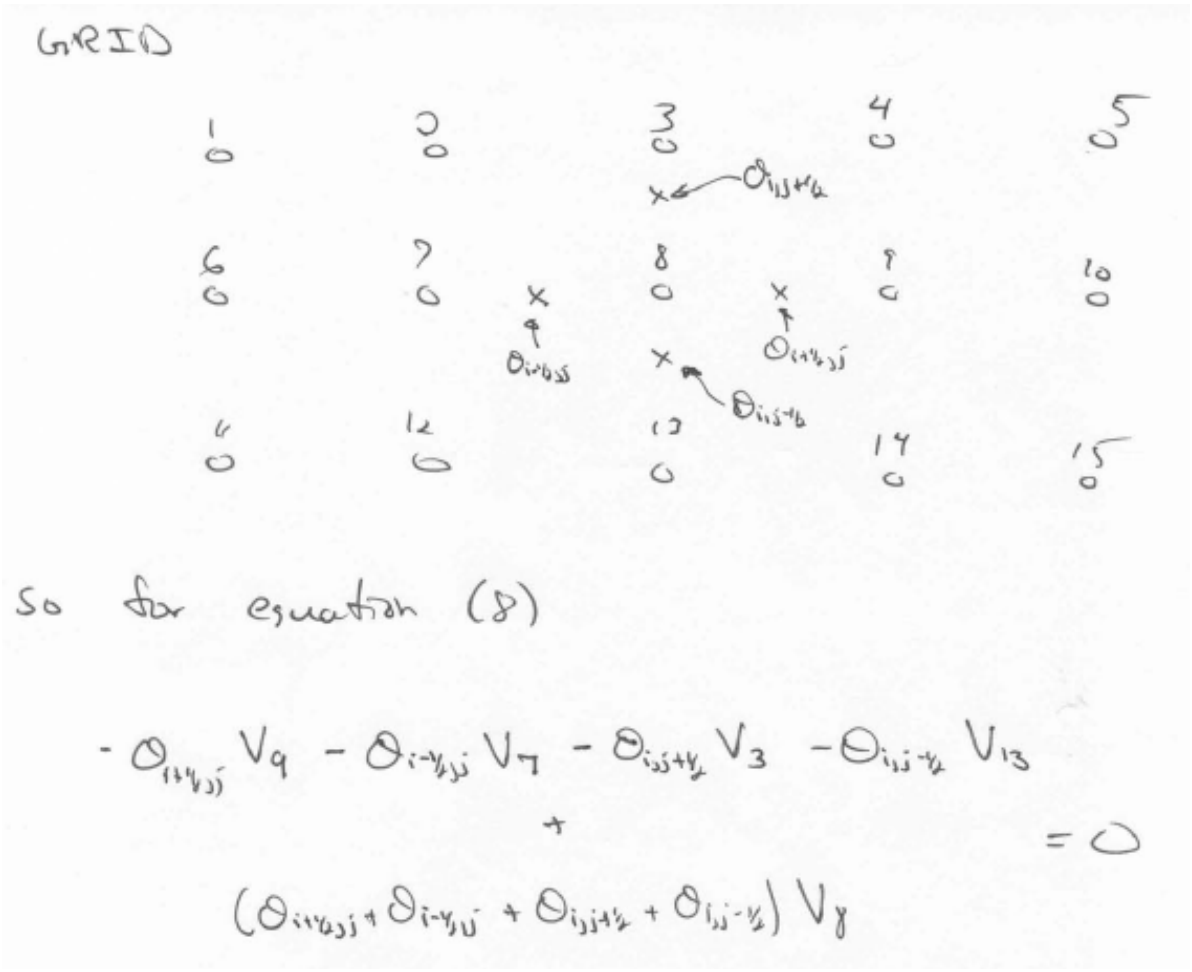
$$\frac{-\theta_{i,j+\frac{1}{2}}V_{i,j+1}+(\theta_{i,j+\frac{1}{2}}+\theta_{i,j-\frac{1}{2}})V_{ij}-\theta_{i,j-\frac{1}{2}}V_{i,j-1}}{k^2}=0$$

on equal grid, $h=k$, multiply through by spatial step size..

$$-\theta_{i+\frac{1}{2},j}V_{i+1,j}-\theta_{i-\frac{1}{2},j}V_{i-1,j}-\theta_{i,j+\frac{1}{2}}V_{i,j+1}-\theta_{i,j-\frac{1}{2}}V_{i,j-1}$$

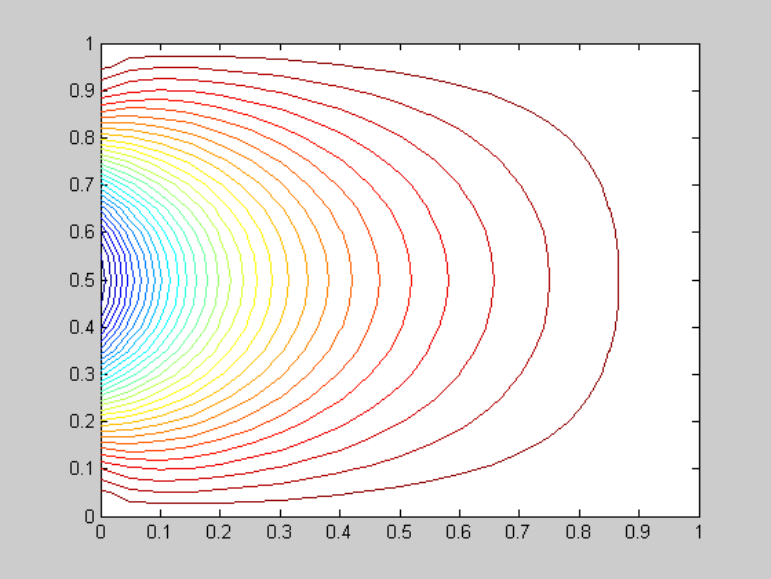
$$+$$

$$(\theta_{i+\frac{1}{2},j}+\theta_{i-\frac{1}{2},j}+\theta_{i,j+\frac{1}{2}}+\theta_{i,j-\frac{1}{2}})V_{ij}=0$$



(b) Given the problem domain and boundary conditions below, compute the solution of Φ using *Jacobi* iteration with an initial solution vector of Φ=0 everywhere. Iterate until reaching an *absolute L_∞ norm-based error* of successive iterates of less than $1\text{x}10^{-5}$ and report the number of iterations needed to reach this convergence criterion. Estimate the spectral radius of the Jacobi iteration matrix during the course of the iteration and compare with the theoretically expected value. Plot contours of your solution over the computational domain and report the actual numerical values of Φ for at the point x=0.7 and y=0.7

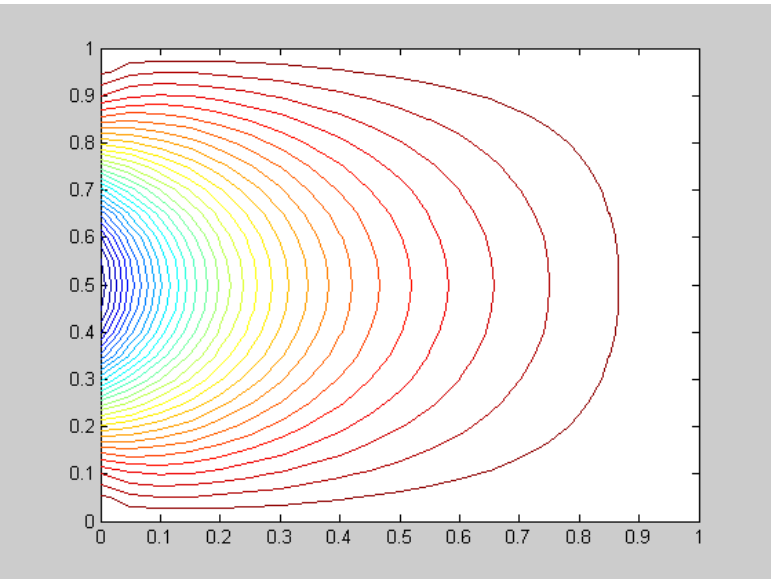
Jacobi # of iterations: 582
Spectral radius estimate: 0.9877
Analytic spectral radius: $1-\pi^2h^2/2=0.98766$
Φ(0.7,0.7)=0.86971



(c) part (a) using Gauss-Seidel. Be sure to report the tabular values of the solution requested in part (a) in order to verify that your solution is essentially unchanged. Is the speed-up in terms of convergence rate relative to Jacobi in agreement with theory? If the convergence critierion were extended to $1x10^{-6}$ how many more iterations would you ***predict*** would be needed with Gauss-Seidel? Is your prediction in reasonable agreement with practice?

Gauss-Seidel # of iterations: 322
Spectral radius estimate: 0.97553
Analytic spectral radius: $(\rho_{jacobi})^2$=0.97547
$\Phi(0.7,0.7)$=0.869996

Gauss-Seidel converges approximately twice as fast as Jacobi according to theory. Here we see that the reduction of iterations is approximately 2 with Jacobi taking 582 iterations verus 322 for Gauss-Seidel



In reducing our tolerance from 10^{-5} to 10^{-6} we expect the following number of additional iterations:

$N_{Jacobi\ @\ 10^{-6}} = N_{Jacobi\ @\ 10^{-5}} + \log(0.1)/\log(\rho_{Jacobi}) = 582+\log(0.1)/\log(0.9877)=768$

$N_{Gauss\text{-}Seidel\ @\ 10^{-6}} = N_{Gauss\text{-}Seidel\ @\ 10^{-5}} + \log(0.1)/\log(\rho_{Gauss\text{-}Seidel}) = 322+\log(0.1)/\log(0.97553)=415$

Rerunning the simulations at a tolerance of 10-6, produced the following:

$N_{Jacobi\ @\ 10^{-6}}=768$

$N_{Gauss\text{-}Seidel\ @\ 10^{-6}}=415$

The estimated increase in the number of iterations for the new tolerance is the same as the numerical output.

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

Aold=zeros(21,21);
for i=1:21
  Aold(i,21)=1;
  Aold(1,i)=1;
  Aold(21,i)=1;
  Aold(i,1)=cos(2*pi*(i*h-0.05));
end
A=zeros(21,21);
A=Aold;
error=1;
errorold=1;
itr=0;
while (error > 1e-5 & itr < 10000)
   itr=itr+1;

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));
   errorold=error;
   error=max(max(abs(A-Aold)));
   spectral(itr)=error/errorold;
   Aold=A;
end

fprintf('Jacobi Iterations %d\n',itr);
figure(1);
contour(A);
figure(2);
plot(spectral);


% Gauss-Seidel
clear;
h=0.05;
y=[0:.05:1]';
A=zeros(21,21);
for i=1:21
  A(i,21)=1;
  A(1,i)=1;
```

```
  A(21,i)=1;
  A(i,1)=cos(2*pi*(i*h-0.05));
end

error=1;
itr=0;
while (error > 1e-5 & itr < 10000)
   itr=itr+1;
   Aold=A;
   for i=2:20
        for j=2:20
         A(i,j)=1/4*(A(i-1,j)+Aold(i+1,j)+A(i,j-1)+Aold(i,j+1));
     end
    end
   errorold=error;
   error=max(max(abs(A-Aold)));
   errornew=error;
   spectral(itr)=errornew/errorold;
end


fprintf('Gauss Seidel Iterations %d\n',itr);
figure(3);
contour(A);
figure(4);
plot(spectral);
```