Elec 4700 Assignment 2 - Kyle Poulin: 100939284

Table of Contents

Due February 25th, 2018

Purpose

The purpose of this assignment is to explore finite difference problems with solutions in the form of matrix math.

Question 1

The Finite Difference Method to solve for the electrostatic potential in the rectangular region nx x ny using $\nabla^2 V = 0$ involves the following iterative action on a matrix nx x ny:

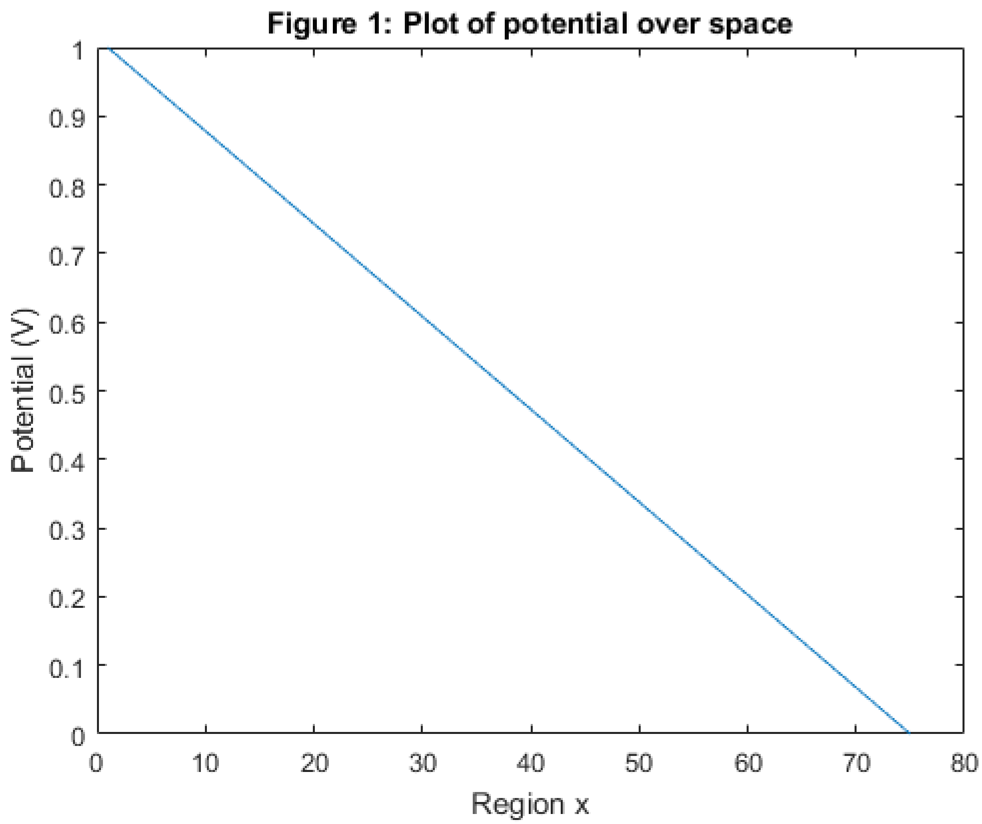
$$\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}$$

```
Q1a = true;
Q1b = true;
Q2a = true;
Q2b = true;
Q2c = true;
Q2d = true;

nx = 75;
ny = 50;

map = @(j,i) j + (i-1)*ny;

v0 = 1;
G = sparse(nx*ny);
B = zeros(1,nx*ny);
```

Question 1.a:

It can be seen from figures 2 and 3 that the finite difference and the analytical solutions are very similar, although the analytic solution is less defined near the edges of the defined space. This is the benefit of using the finite difference method. The mesh size affects the precision of the solution.

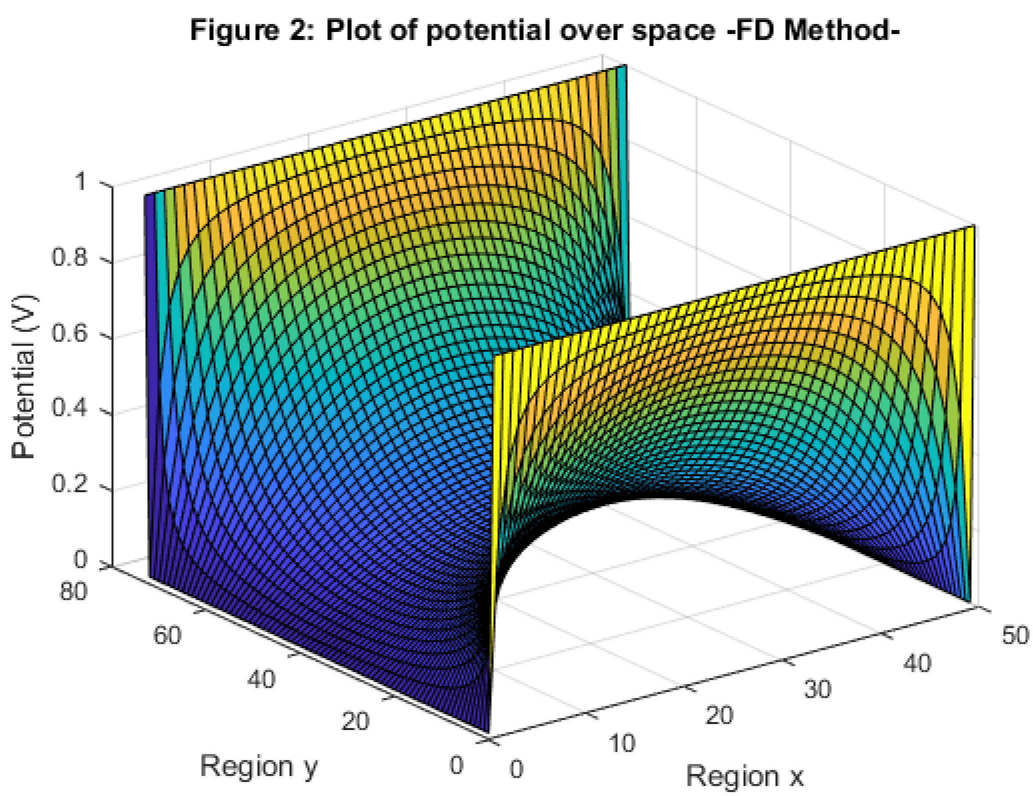
```
if(Q1a == true)
    for i = 1:nx
        for j = 1:ny
            n = map(j,i);
            if i==1
                B(n) = v0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif i == nx
                B(n) = 0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif j ==1
                G(n,n) = -3;
                G(n,map(j,i-1)) = 1;
                G(n,map(j,i+1)) = 1;
                G(n,map(j+1,i)) = 1;
            elseif j == ny
                G(n,n) = -3;
                G(n,map(j,i-1)) = 1;
                G(n,map(j,i+1)) = 1;
                G(n,map(j-1,i)) = 1;
            else
                G(n,n) = -4;
                G(n,map(j,i-1)) = 1;
                G(n,map(j,i+1)) = 1;
                G(n,map(j+1,i)) = 1;
                G(n,map(j-1,i)) = 1;
            end
        end
    end
    E = G\B';
    V = zeros(ny,nx);
    for i = 1:nx
        for j = 1:ny
            n = map(j,i);
            V(j,i) = E(n);
        end
    end
    figure(1);
    plot(V');
    title('Figure 1: Plot of potential over space');
    xlabel('Region x');
    ylabel('Potential (V)');
end
```



Question 1.b:

```
if(Q1b == true)

    for i = 1:nx
        for j = 1:ny
            n = map(j,i);
            if i==1
                B(n) = v0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif i == nx
                B(n) = v0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif j ==1
                B(n) = 0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif j == ny
                B(n) = 0;
                G(n,:) = 0;
                G(n,n) = 1;
            else
                G(n,n) = -4;
                G(n,map(j,i-1)) = 1;
```

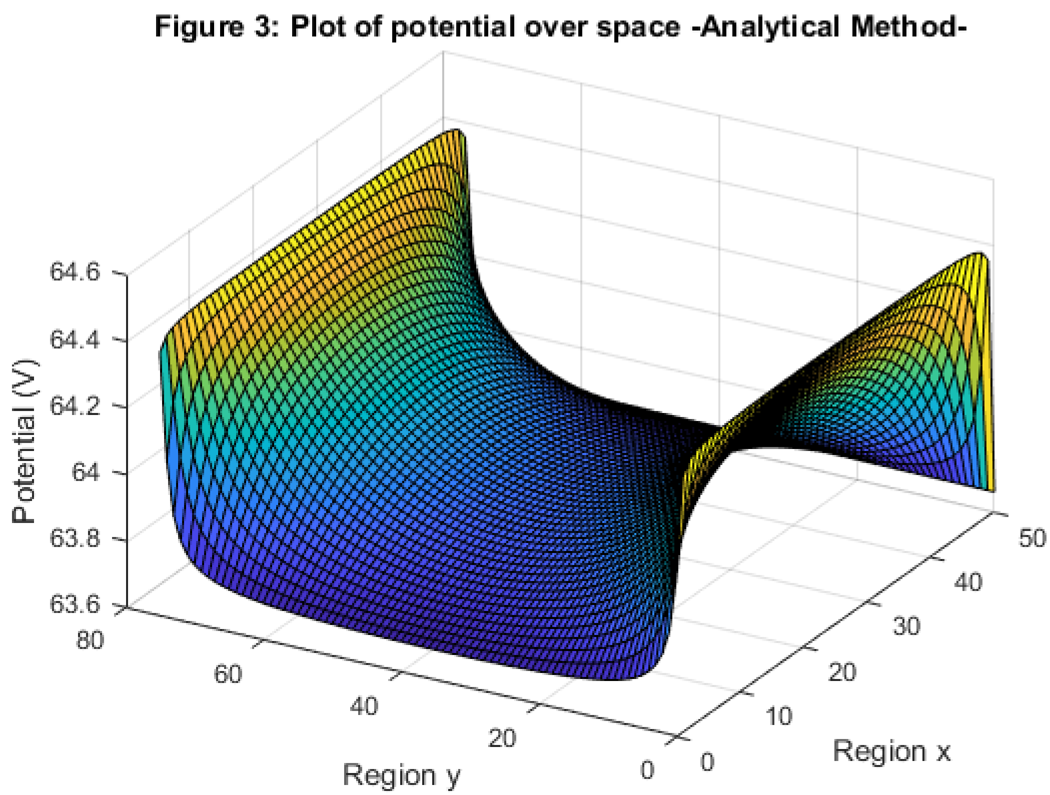
```
                G(n,map(j,i+1)) = 1;
                G(n,map(j+1,i)) = 1;
                G(n,map(j-1,i)) = 1;
            end
        end
    end
    E = G\B';
    V = zeros(ny,nx);
    for i = 1:nx
        for j = 1:ny
            n = map(j,i);
            V(j,i) = E(n);
        end
    end
    figure(2);
    surf(V');
    title('Figure 2: Plot of potential over space -FD Method-');
    xlabel('Region x');
    ylabel('Region y');
    zlabel('Potential (V)');
```



Analytic Solution

```
y = 1:ny;
x = -(nx+1)/2:(nx+1)/2;
[vx,vy] = meshgrid(x,y);
```

```
    steps = 100;

    f = 4*v0/pi + cosh(pi*vx/ny)/cosh(pi*((nx+1)/2)/ny).*sin(pi*vy/
ny);

    for n = 3:2:steps
        f = f+ 4*v0/pi + (1/n)*cosh(n*pi*vx/ny)/cosh(n*pi*((nx+1)/2)/
ny).*sin(n*pi*vy/ny);
    end
    figure(3);
    surf(vy,vx+38,f);
    title('Figure 3: Plot of potential over space -Analytical
 Method-');
    xlabel('Region x');
    ylabel('Region y');
    zlabel('Potential (V)');
    view(-60,38);
```




```
end
```

Question 2

By using the finite difference method with the inclusion of conductivity, we explored the affect of a resistive bottle-neck in a two-dimmensional region. From figure 4, we can see that the voltage drops mostly over the resistive boxes, which relates to our understanding of $V = I * R$. Figure 5 shows the conductivity in the region, while figures 6 and 7 show the electric field and current density in the region.

```
if(Q2a == true)

    cond1=1;
    cond2=0.01;
    boxXratio = (2/5);
    boxYratio = (2/5);
    box = [nx*boxXratio nx*(1-boxXratio) ny*boxYratio ny*(1-
boxYratio)];
    nx =75;
    ny=50;

    sigma = zeros(ny,nx);
    for i = 1:nx
        for j = 1:ny
            if ((j<box(3))&&(i>box(1))&&(i<box(2))) ||
 ((j>box(4))&&(i>box(1))&&(i<box(2)))
                sigma(j,i) = cond2;
            else
                sigma(j,i) = cond1;
            end
        end
    end

    for i = 1:nx
        for j = 1:ny
            n = map(j,i);
            if i==1
                B(n) = v0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif i == nx
                B(n) = 0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif j ==1
                up = (sigma(j,i) + sigma(j+1,i))/2;
                left = (sigma(j,i) + sigma(j,i-1))/2;
                right = (sigma(j,i) + sigma(j,i+1))/2;

                G(n,n) = -(up+left+right);
                G(n,map(j,i-1)) = left;
                G(n,map(j,i+1)) = right;
                G(n,map(j+1,i)) = up;
            elseif j == ny
                left = (sigma(j,i) + sigma(j,i-1))/2;
                right = (sigma(j,i) + sigma(j,i+1))/2;
                down = (sigma(j,i) + sigma(j-1,i))/2;

                G(n,n) = -(down+left+right);
                G(n,map(j,i-1)) = left;
                G(n,map(j,i+1)) = right;
                G(n,map(j-1,i)) = down;
            else
                up = (sigma(j,i) + sigma(j+1,i))/2;
```

```
            left = (sigma(j,i) + sigma(j,i-1))/2;
            right = (sigma(j,i) + sigma(j,i+1))/2;
            down = (sigma(j,i) + sigma(j-1,i))/2;

            G(n,n) = -(up+left+right+down);
            G(n,map(j,i-1)) = left;
            G(n,map(j,i+1)) = right;
            G(n,map(j+1,i)) = up;
            G(n,map(j-1,i)) = down;
        end
    end
end
E = G\B';
d = zeros(ny,nx);
for i = 1:nx
    for j = 1:ny
        n = map(j,i);
        d(j,i) = E(n);
    end
end

figure(4);
surf(d);
title('Figure 4: V(x,y) over a 1/5 width bottle-neck region');
xlabel('Region x');
ylabel('Region y');


figure(5)
surf(sigma);
title('Figure 5: Sigma over a 1/5 width bottle-neck region');
xlabel('Region x');
ylabel('Region y');

[Ex, Ey] = gradient(d);
figure(6)
quiver(Ex, Ey);
title('Figure 6: E(x,y) over a 1/5 width bottle-neck region');
xlabel('Region x');
ylabel('Region y');

Jx = Ex.*sigma;
Jy = Ey.*sigma;

figure(7)
quiver(Jx, Jy);
title('Figure 7: J(x,y) over a 1/5 width bottle-neck region');
xlabel('Region x');
ylabel('Region y');
```

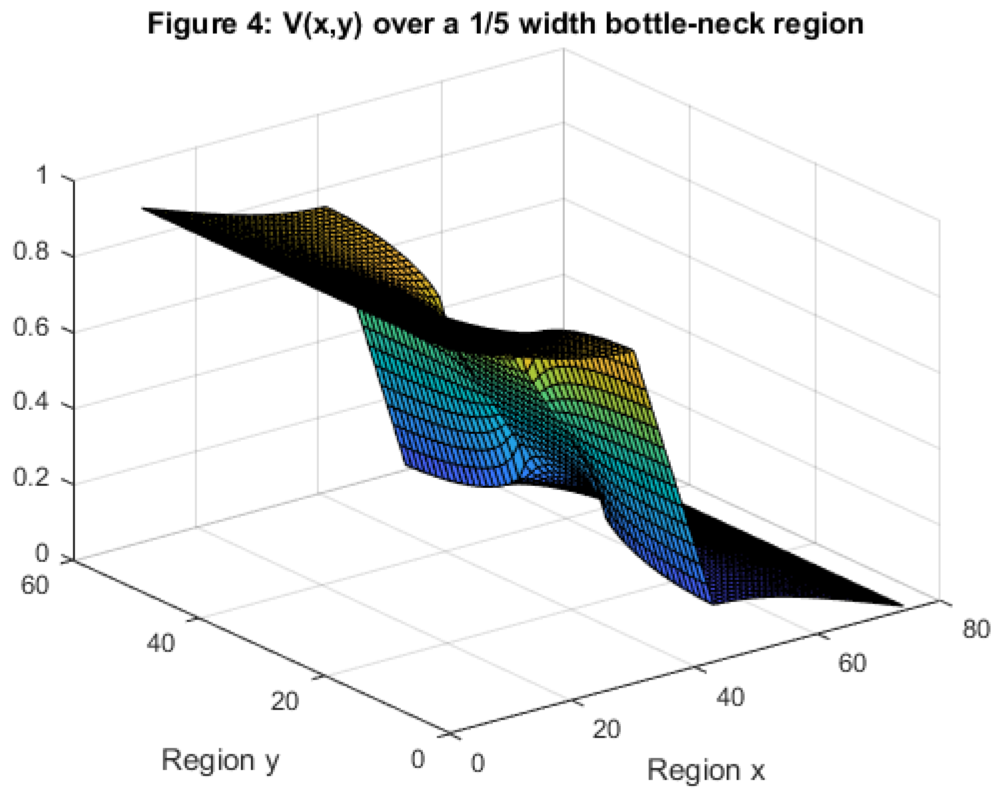


Figure 4: V(x,y) over a 1/5 width bottle-neck region
1
0.8
0.6
0.4
0.2
0
60
40
20
0
Region y
0
20
40
60
80
Region x

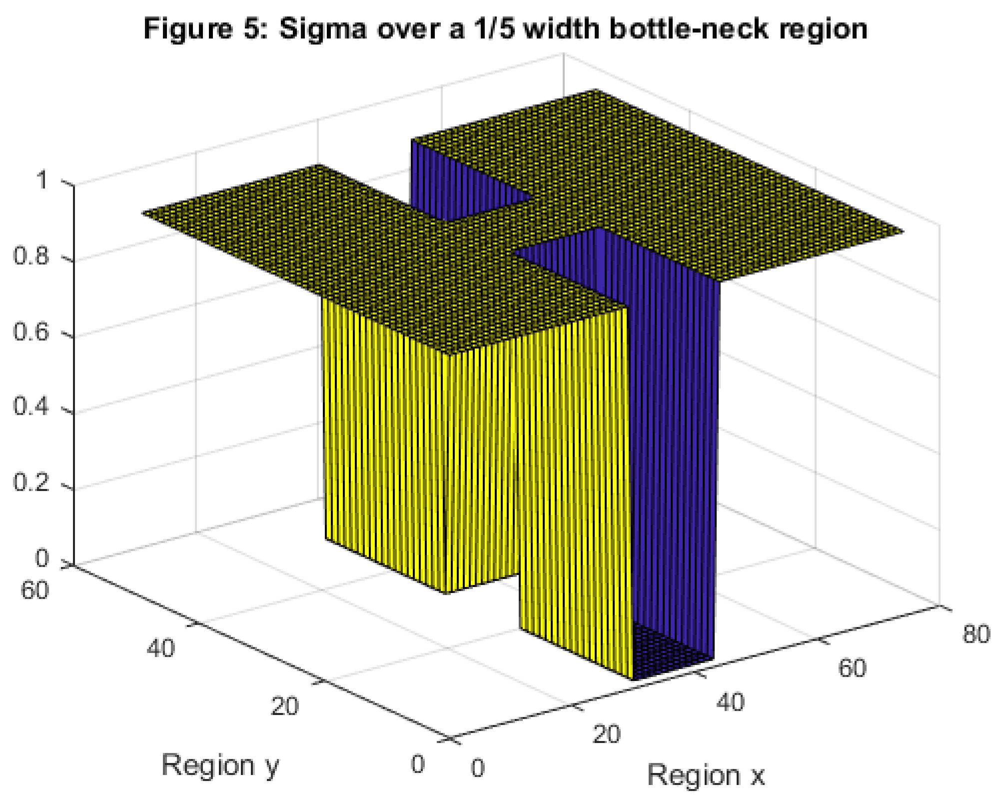


Figure 5: Sigma over a 1/5 width bottle-neck region
1
0.8
0.6
0.4
0.2
0
60
40
20
0
Region y
0
20
40
60
80
Region x

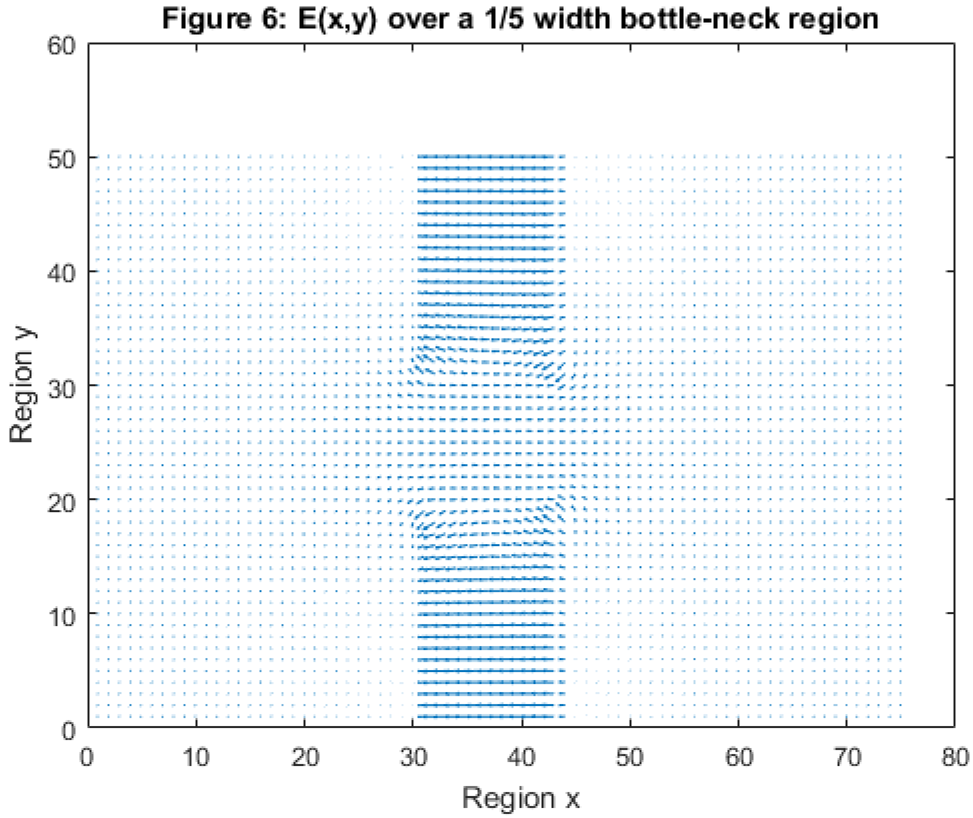


Figure 6: E(x,y) over a 1/5 width bottle-neck region
Region y
Region x
0
10
20
30
40
50
60
70
80

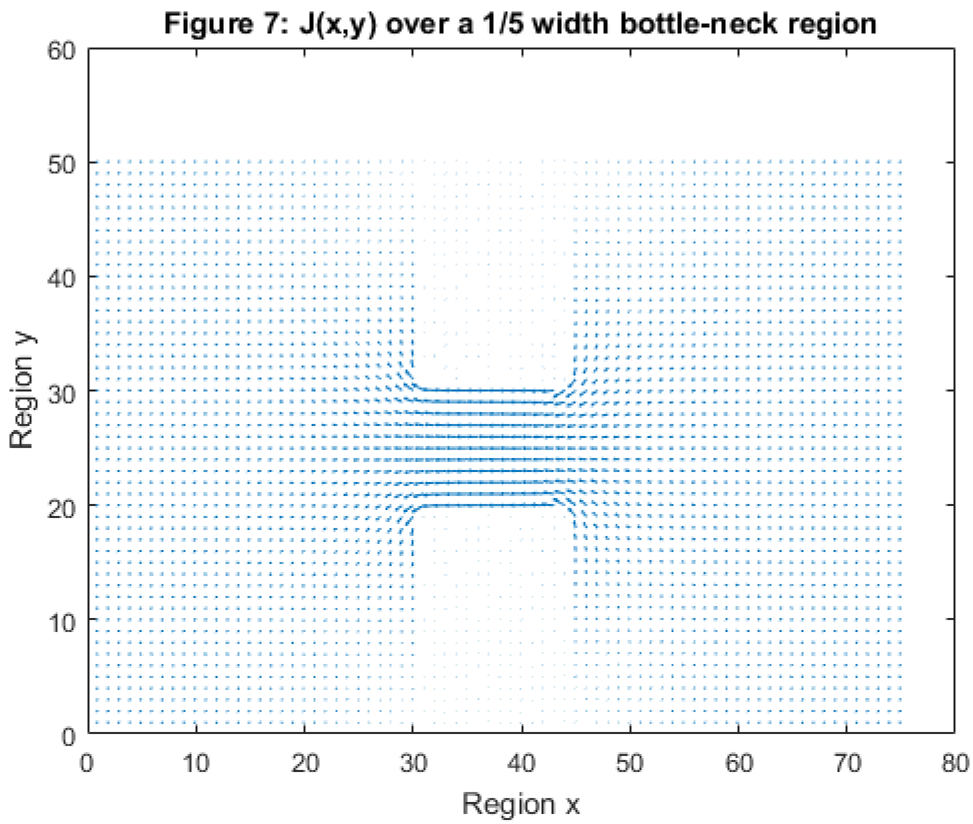
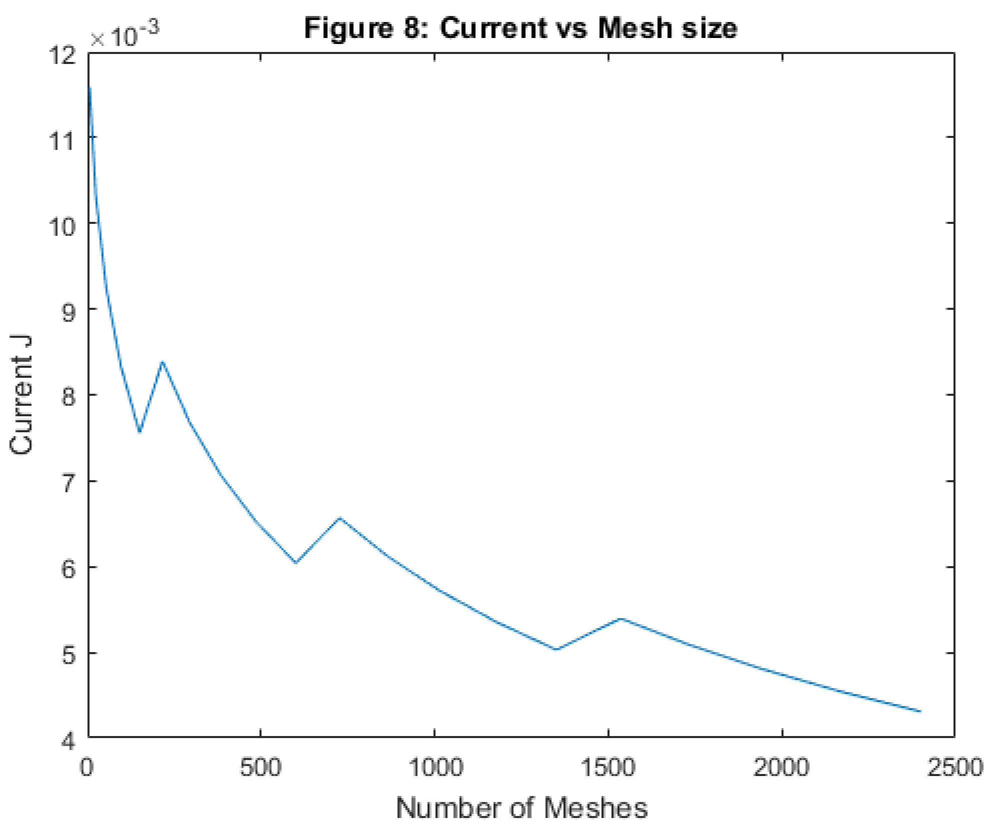


Figure 7: J(x,y) over a 1/5 width bottle-neck region
Region y
Region x
0
10
20
30
40
50
60
70
80

Q2.b

In this section, the affect of mesh density was explored by calculating the current through the region while decreasing the mesh size (increasing the number of vertices). We can see from figure 8 that the current value approaches the true value as the number of meshes increases.

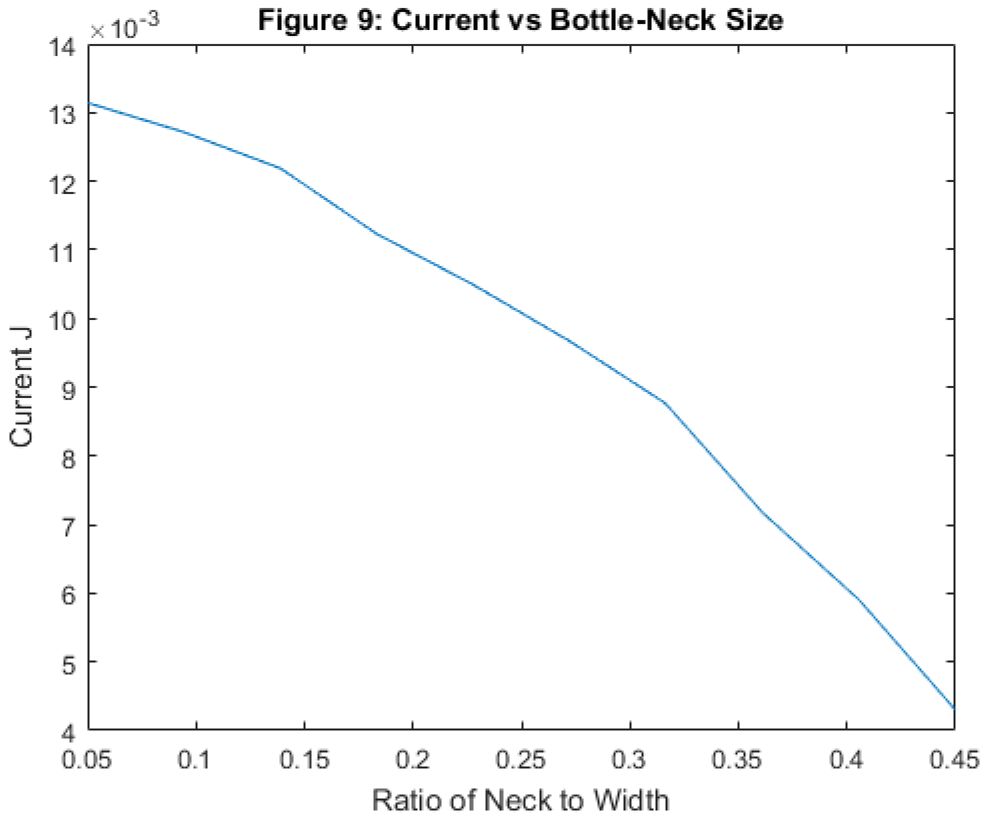
```
    if (Q2b==true)
        meshMax = 20;
        currentArray = zeros(1,meshMax);
        meshArray = zeros(1,meshMax);
        for i = 1:(meshMax)
            meshArray(i) = (i^2)*2*3;
        end

        for mesh = 15:meshMax+14
            currentArray(mesh-14) =
 getCurrent(1,0.01,2/5,2/5,round(mesh*3),round(mesh*2),1);
        end
        figure(8)
        plot(meshArray,currentArray);
        title('Figure 8: Current vs Mesh size');
        xlabel('Number of Meshes');
        ylabel('Current J');
    end
```



Q2.c

In this section, the affect of narrowing the bottle-neck wass explored. As expected, decreasing the width of the bottle-neck decreases the amount of current.

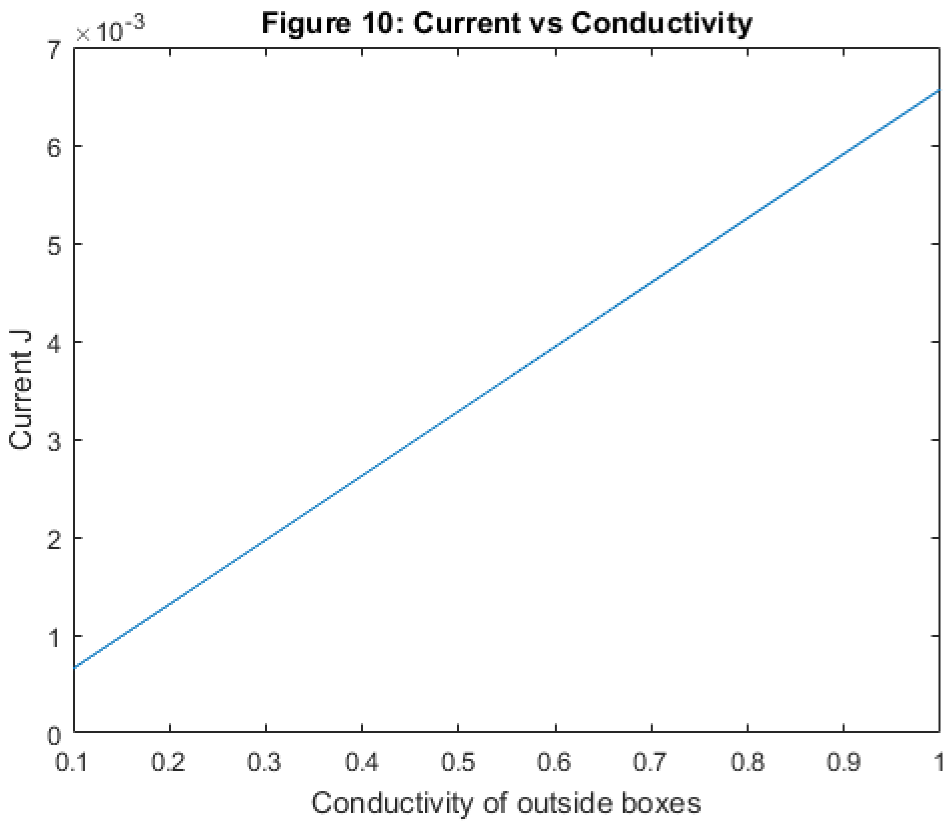
```
    if(Q2c==true)
        nx=75;
        ny=50;
        maxRatio = (9/20);
        minRatio = (1/20);
        density = 10;
        ratioArray = linspace(minRatio,maxRatio,density);
        currentArray2 = zeros(1,density);
        for i= 1:density
            currentArray2(i)  = getCurrent(1, 0.01, 2/5,
 ratioArray(i), nx, ny, 1);
        end
        figure(9)
        plot(ratioArray,currentArray2);
        title('Figure 9: Current vs Bottle-Neck Size');
        xlabel('Ratio of Neck to Width');
        ylabel('Current J');
    end
```



Q2.d

In this section, we explored the affect of changing the conductivity inside and outside of the resistive boxes. As expected, a less resistive region allows more current under a constant voltage.

```
if(Q2d==true)
    nx=75;
    ny=50;
    sigmaArray = linspace(0.1,1,10);
    currentArray3 = zeros(1,10);
    j=0;
    for i= 0.1:0.1:1
        j = j+1;
        currentArray3(j) = getCurrent(i,i/100,2/5,2/5,nx,ny,1);
    end
    figure(10)
    plot(sigmaArray,currentArray3);
    title('Figure 10: Current vs Conductivity');
    xlabel('Conductivity of outside boxes');
    ylabel('Current J');
end
```




```
end
```

Function to calculate the current:

```
function [current] = getCurrent(r1,r2,ratiox,ratioy,xmax,ymax, v0)
    cond1=r1;
    cond2=r2;
    nx = xmax;
    ny = ymax;
    boxXratio = ratiox;
    boxYratio = ratioy;
    box = [nx*boxXratio nx*(1-boxXratio) ny*boxYratio ny*(1-
boxYratio)];
    sigma = zeros(ny,nx);
    for i = 1:nx
        for j = 1:ny
            if ((j<box(3))&&(i>box(1))&&(i<box(2))) ||
 ((j>box(4))&&(i>box(1))&&(i<box(2)))
                sigma(j,i) = cond2;
            else
                sigma(j,i) = cond1;
            end
        end
    end

    for i = 1:nx
        for j = 1:ny
            n = j + (i-1)*ny;
            if i==1
                B(n) = v0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif i == nx
                B(n) = 0;
                G(n,:) = 0;
                G(n,n) = 1;
            elseif j ==1
                up = (sigma(j,i) + sigma(j+1,i))/2;
                left = (sigma(j,i) + sigma(j,i-1))/2;
                right = (sigma(j,i) + sigma(j,i+1))/2;

                G(n,n) = -(up+left+right);
                G(n,j + (i-2)*ny) = left;
                G(n,j + (i)*ny) = right;
                G(n,j+1 + (i-1)*ny) = up;
            elseif j == ny
                left = (sigma(j,i) + sigma(j,i-1))/2;
                right = (sigma(j,i) + sigma(j,i+1))/2;
                down = (sigma(j,i) + sigma(j-1,i))/2;

                G(n,n) = -(down+left+right);
                G(n,j + (i-2)*ny) = left;
                G(n,j + (i)*ny) = right;
                G(n,j-1 + (i-1)*ny) = down;
            else
```

```
                up = (sigma(j,i) + sigma(j+1,i))/2;
                left = (sigma(j,i) + sigma(j,i-1))/2;
                right = (sigma(j,i) + sigma(j,i+1))/2;
                down = (sigma(j,i) + sigma(j-1,i))/2;

                G(n,n) = -(up+left+right+down);
                G(n,j + (i-2)*ny) = left;
                G(n,j + (i)*ny) = right;
                G(n,j+1 + (i-1)*ny) = up;
                G(n,j-1 + (i-1)*ny) = down;
            end
        end
    end
    E = G\B';
    d = zeros(ny,nx);
    for i = 1:nx
        for j = 1:ny
            n = j + (i-1)*ny;
            d(j,i) = E(n);
        end
    end
    [Ex, Ey] = gradient(d);
    Ex = -Ex;
    Ey = -Ey;
    Jx = Ex.*sigma;
    Jy = Ey.*sigma;
    curr =0;

    for j= 1:ny
        curr = curr + Jx(j,15)/ny;
    end
    current = curr;
end
```

Published with MATLAB® R2017a