Table of Contents

Question 1: Part (a)

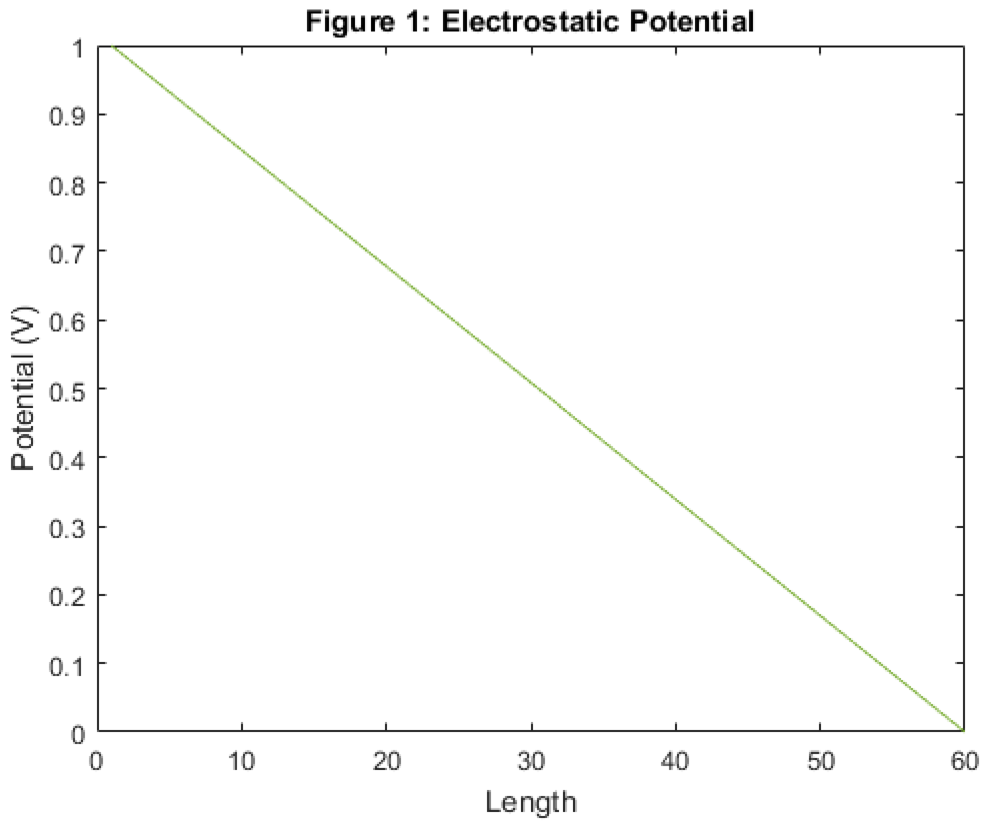
Using the finite difference method, this script solves for the electrostatic potential in a rectangular region using the equation

$$\nabla^2 V = 0$$

in this first case, the boundary conditions used were $V = V_0$ at $x = 0$ and $V = 0$ at $x = L$ with the y boundaries (top and bottom) not fixed. V_0 is set to 1, and the resulting potential is shown in figure 1.

```
clear
close all
L=60;
W=40;
G=sparse(L*W);
B=zeros(1,L*W);

for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        if i==1
            G(n,:)=0;
            G(n,n)=1;
            B(n)=1;
        elseif i==L
            G(n,:)=0;
            G(n,n)=1;
            B(n)=0;
        elseif j==1
            G(n,n)=-3;
            G(n,n+1)=1;
            G(n,n-W)=1;
            G(n,n+W)=1;
        elseif j==W
            G(n,n)=-3;
            G(n,n-1)=1;
            G(n,n+W)=1;
            G(n,n-W)=1;
        else
            G(n,n)=-4;
            G(n,n+1)=1;
            G(n,n-1)=1;
            G(n,n+W)=1;
            G(n,n-W)=1;
```

```
        end
    end
end
E=G\B';
Ematrix=zeros(L,W);
for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        Ematrix(i,j)=E(n);
    end
end
figure(1)
plot(Ematrix)
title('Figure 1: Electrostatic Potential')
xlabel('Length')
ylabel('Potential (V)')
```



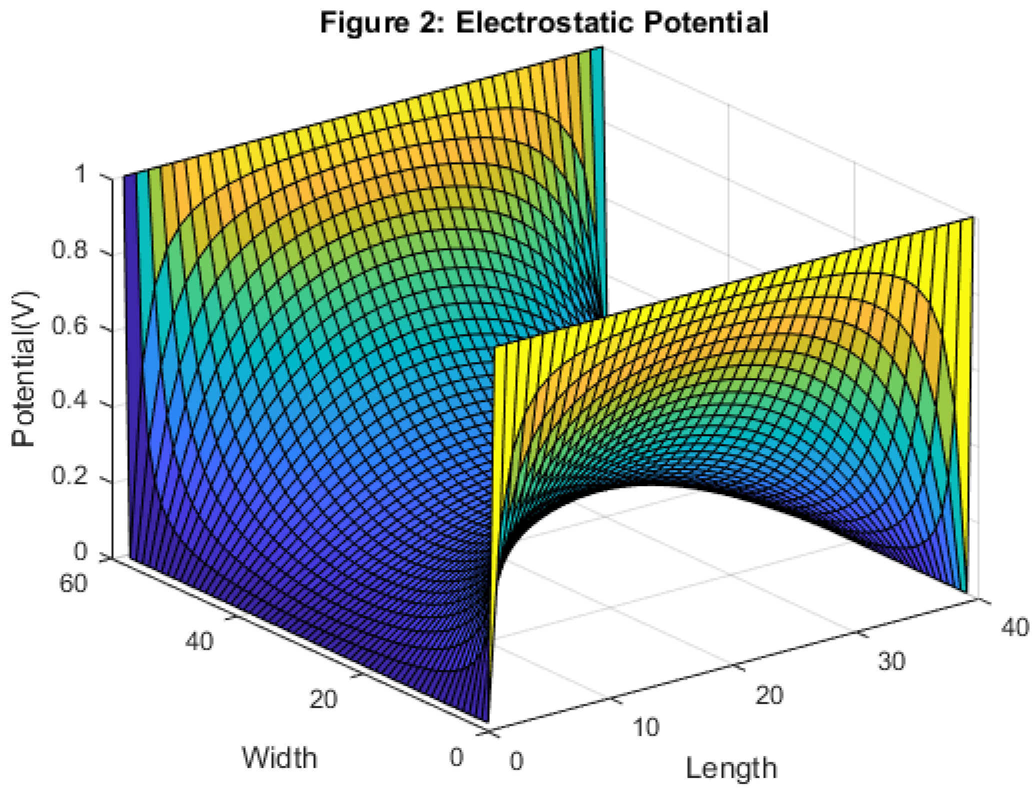
Question 1: Part (b)

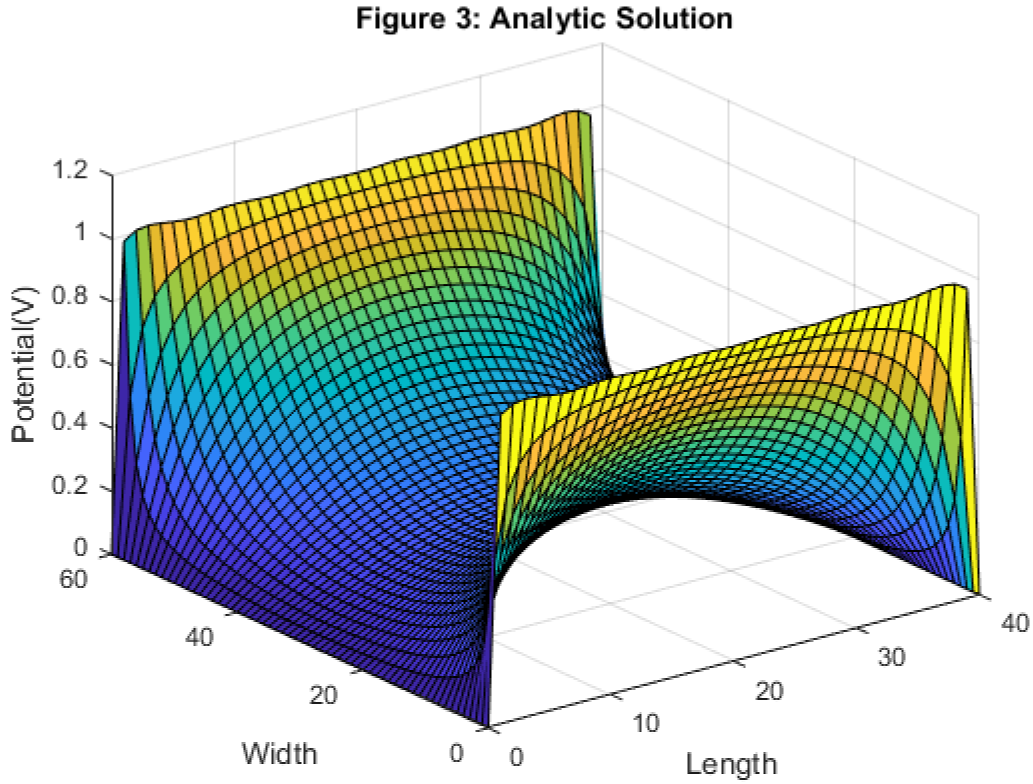
In part two, the same problem is solved with new boundary conditions. in this case $V = V_0$ on the left and right and $V = 0$ on the top and bottom. A few different mesh sizes were tried, and an example result is shown in figure 2. In addition, and analytical series solution was plotted and is shown in figure 3.

As the mesh size is decresed(i.e. as the number of grid squares is increased) the solution smooths out and approaches the true solution. There is a trade off however, as a smaller mesh size means a slower simulation. In this case, the numerical solution apporaches the analytic solution very quickly requiring a mesh size on the order of ~50 squares across. the analytic solution has error in it as well, as the sum is technically infinite, although to plot here obviously the number of steps summed is finite. 100 steps is more than sufficient to produce a very good plot, and more steps help mitigate the ripple at the edges (though it will never fully remove it). I found there is also an upper limit on the number of steps, as the cosh function cannot handle the larger numbers as the number of steps increases, and values are thrown out due to this error.

```
close all
clear
L=60;
W=40;
G=sparse(L*W);
B=zeros(1,L*W);

for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        if i==1
            G(n,:)=0;
            G(n,n)=1;
            B(n)=1;
        elseif i==L
            G(n,:)=0;
            G(n,n)=1;
            B(n)=1;
        elseif j==1
            G(n,:)=0;
            G(n,n)=1;
            B(n)=0;
        elseif j==W
            G(n,:)=0;
            G(n,n)=1;
            B(n)=0;
        else
            G(n,n)=-4;
            G(n,n+1)=1;
            G(n,n-1)=1;
            G(n,n+W)=1;
            G(n,n-W)=1;
        end
    end
end
E=G\B';
Ematrix=zeros(L,W);
for i =1:L
    for j=1:W
        n=j+(i-1)*W;
        Ematrix(i,j)=E(n);
    end
end
figure(1)
surf(Ematrix)
title('Figure 2: Electrostatic Potential')
xlabel('Length')
```

```
ylabel('Width')
zlabel('Potential(V)')

x=-L/2:1:L/2;
y=0:1:W;
[X,Y]=meshgrid(x,y);

V=(cosh(pi*X/W)/(cosh(pi*(L/2)/W))).*sin(pi*Y/W);
for n=3:2:251

    V=V+(1/n)*(cosh(n*pi*X/W)/(cosh(n*pi*(L/2)/W))).*sin(n*pi*Y/W);
end
figure(2)
surf(Y,X+(L/2),4*V/pi)
title('Figure 3: Analytic Solution')
xlabel('Length')
ylabel('Width')
zlabel('Potential(V)')
```





Question 2: Part (a) Adding a Bottleneck

We now add resistive boxes to the rectangular region in part 1a to create a "bottleneck" for current to flow through. A mesh size of 50 grid units width wise and 75 grid units length wise, a conductivity of $\sigma = 1$ in the conducting region and $\sigma = 10^{-2}$ in the resistive boxes, and a bottleneck one third the total length in length, and one fifth the total width in width was used. Figure 4 shows the voltage distribution in the region. Figure 5 shows the Electric field in the region. Figure 6 shows a map of the conductivity in the region. Figure 7 shows the Current density distribution.

```
clear
close all
current=getcurrent(50,10,1,1e-2,1)
```

```
current =

    0.0045
```

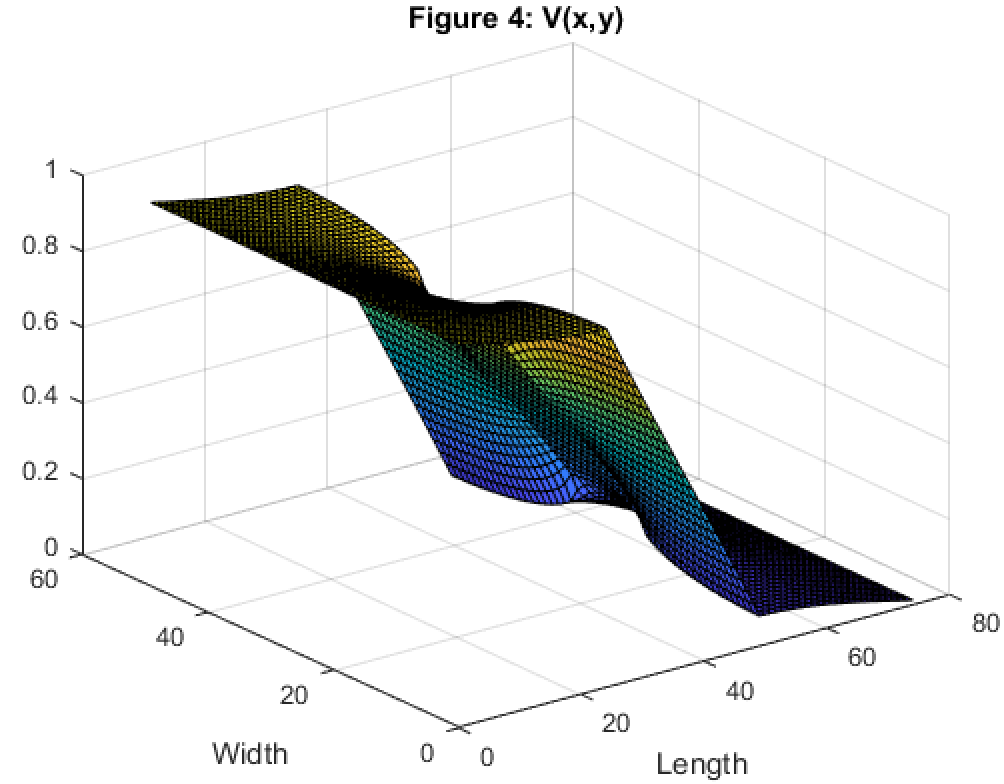


Figure 4: V(x,y)
1
0.8
0.6
0.4
0.2
0
60
40
20
0
Width
0
20
40
60
80
Length

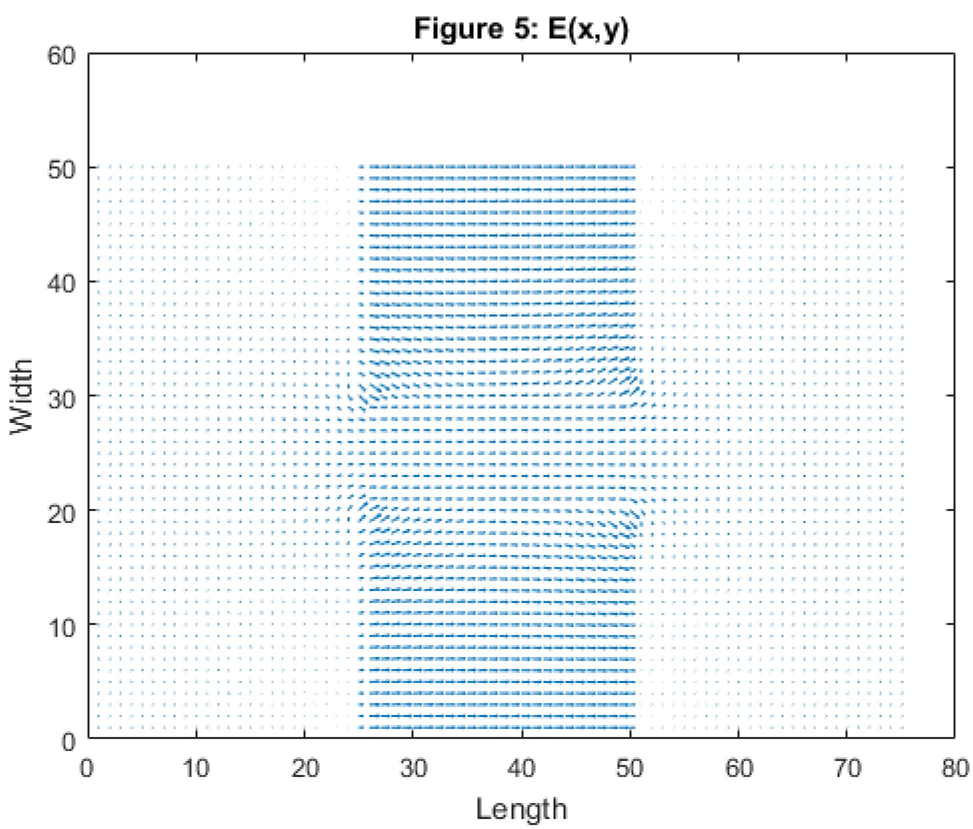


Figure 5: E(x,y)
60
50
40
30
20
10
0
Width
0
10
20
30
40
50
60
70
80
Length

Figure 6: σ (x,y)

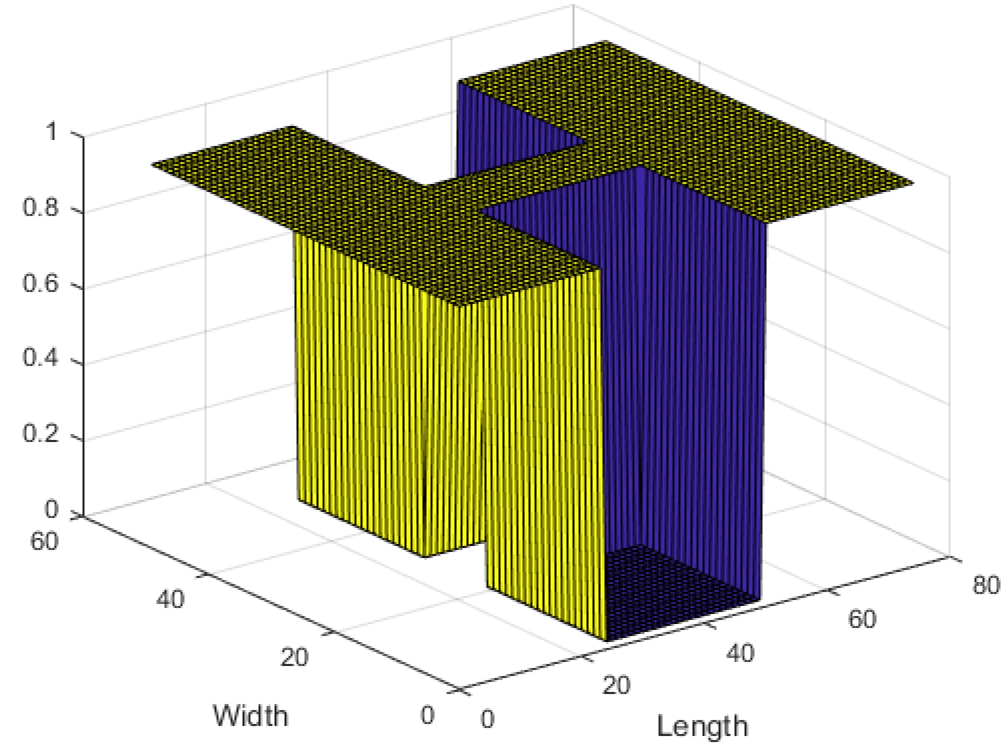
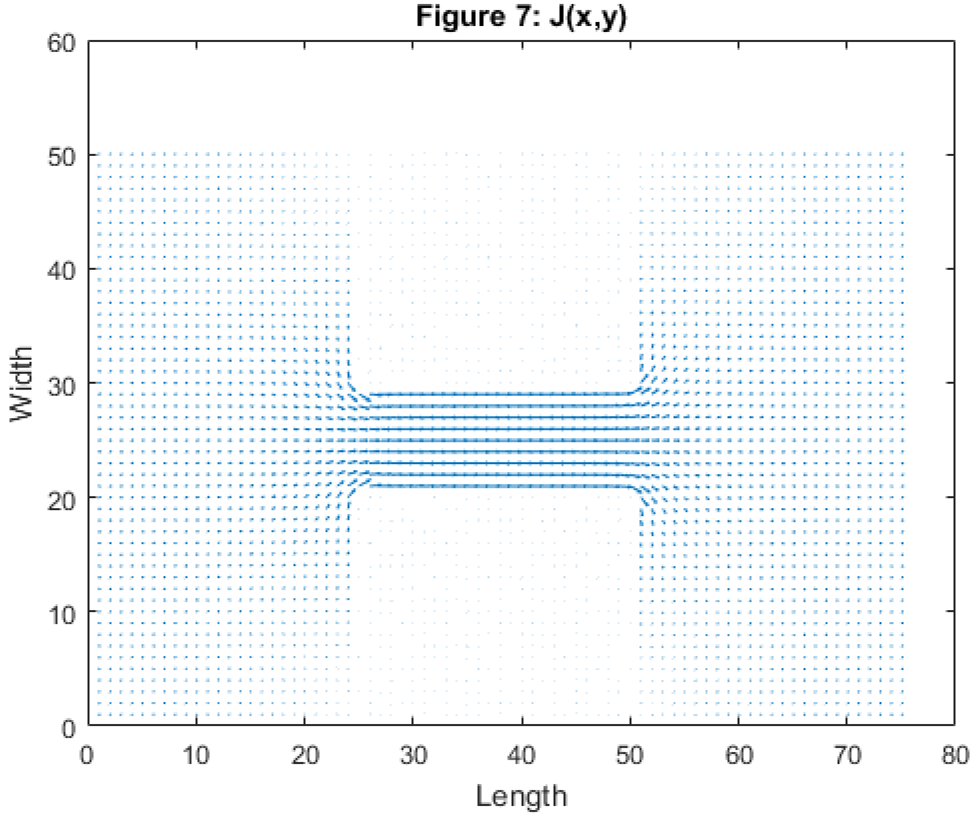


Figure 7: J(x,y)



Question 2: Part (b) Experimentation

Finally, the effects of changing the mesh size, width of the bottleneck, and conductivity of the resistive regions was investigated. Increasing the number of grid squares in the mesh (i.e. making the mesh finer) caused the current value from the simulation to decrease. Note that the Mesh size is given by measuring the number of grid squares in the width of the region, and the ratio of length to width is always held at 3/2. As the mesh is made finer, the current appears to be approaching a value asymptotically, as seen in figure 8. It appears that more coarse meshes overestimate the current, however finer meshes cause the simulation to take dramatically longer to finish, so there is a trade off between efficiency and accuracy.

Increasing the width of the bottleneck causes the current to increase as expected. This is plotted in figure 9. Figure 10 shows the relation of current to the conductivity of the resistive regions.

```
clear
close all
%different mesh sizes
meshsize=linspace(10,100,19);
for i= 1:19
    current1(i)=getcurrent(meshsize(i),10,1,1e-2,0);
end
figure(8)
plot(meshsize,current1)
title('Figure 8: Current with Increasing Mesh Density')
xlabel('Number of Grid Squares in Width')
ylabel('Current(A)')



%different bottleneck widths
necksize=linspace(10,40,16);
for i= 1:16
    current2(i)=getcurrent(50,necksize(i),1,1e-2,0);
end
figure(9)
plot(necksize,current2)
title('Figure 9: Current with Increasing Bottlneck Width')
xlabel('Width of Bottlneck')
ylabel('Current(A)')

%different conductivity inside the boxes
sigma=linspace(0.01,1,100);
for i=1:100
    current3(i)=getcurrent(50,10,1,sigma(i),0);
end
figure(10)
plot(sigma,current3)
title('Figure 10')
xlabel('Conductivity of Resistive Region')
ylabel('Current(A)')
```

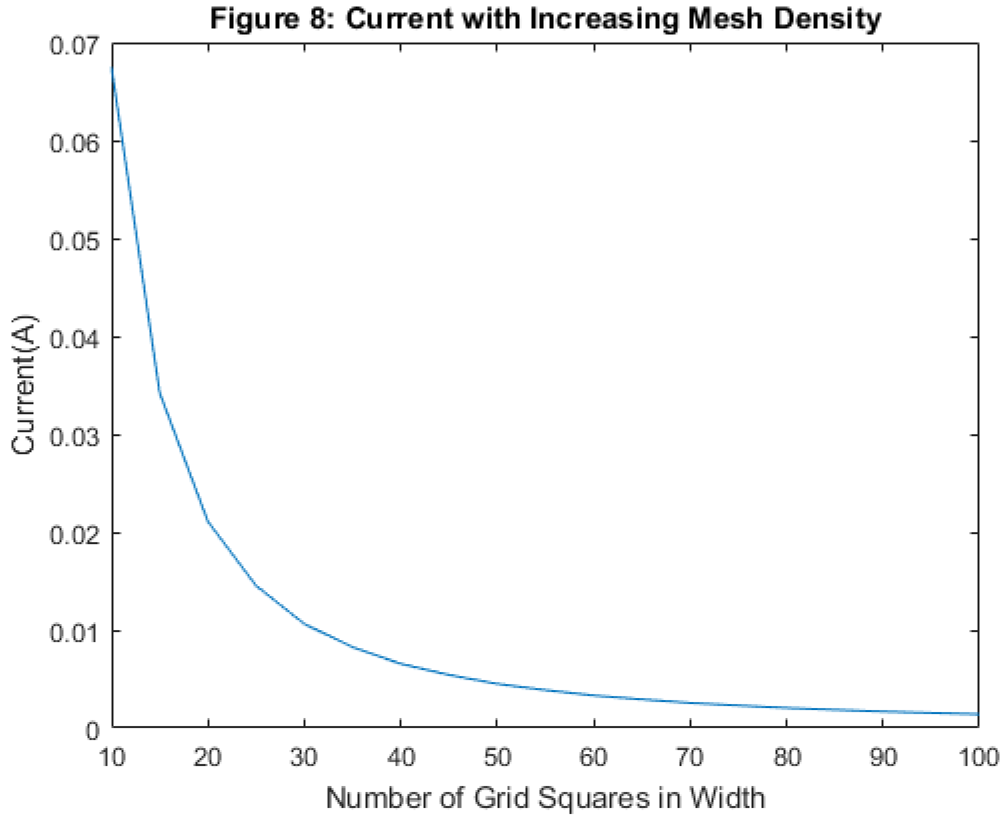


Figure 8: Current with Increasing Mesh Density
Current(A)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
10
20
30
40
50
60
70
80
90
100
Number of Grid Squares in Width

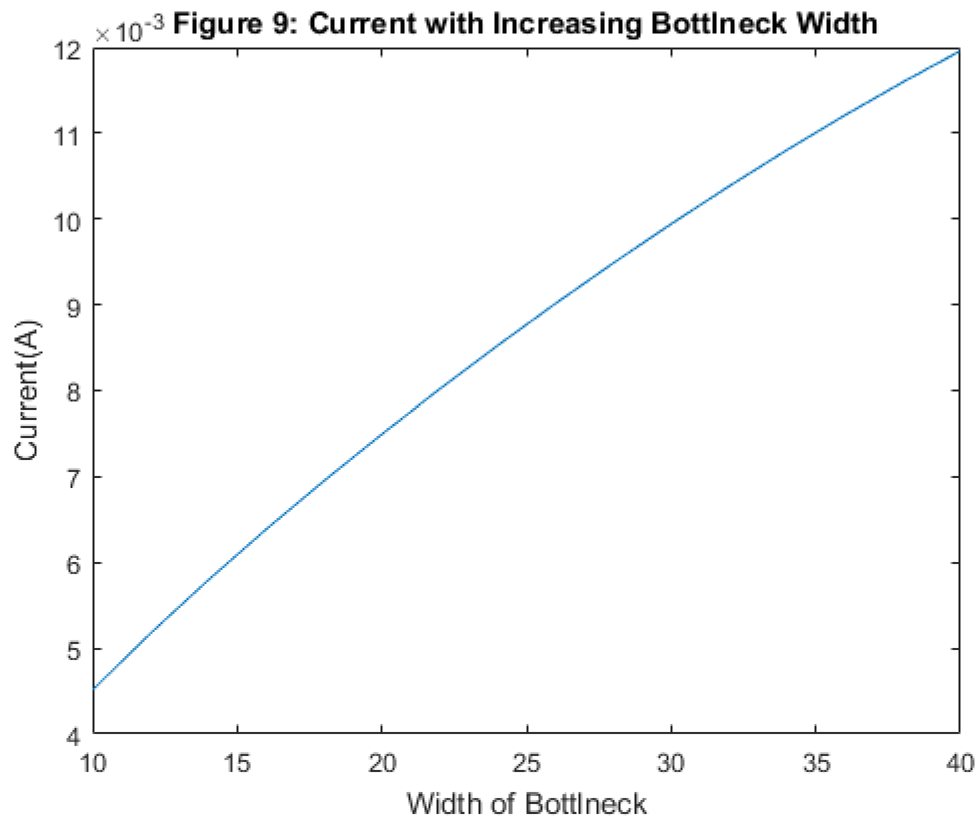
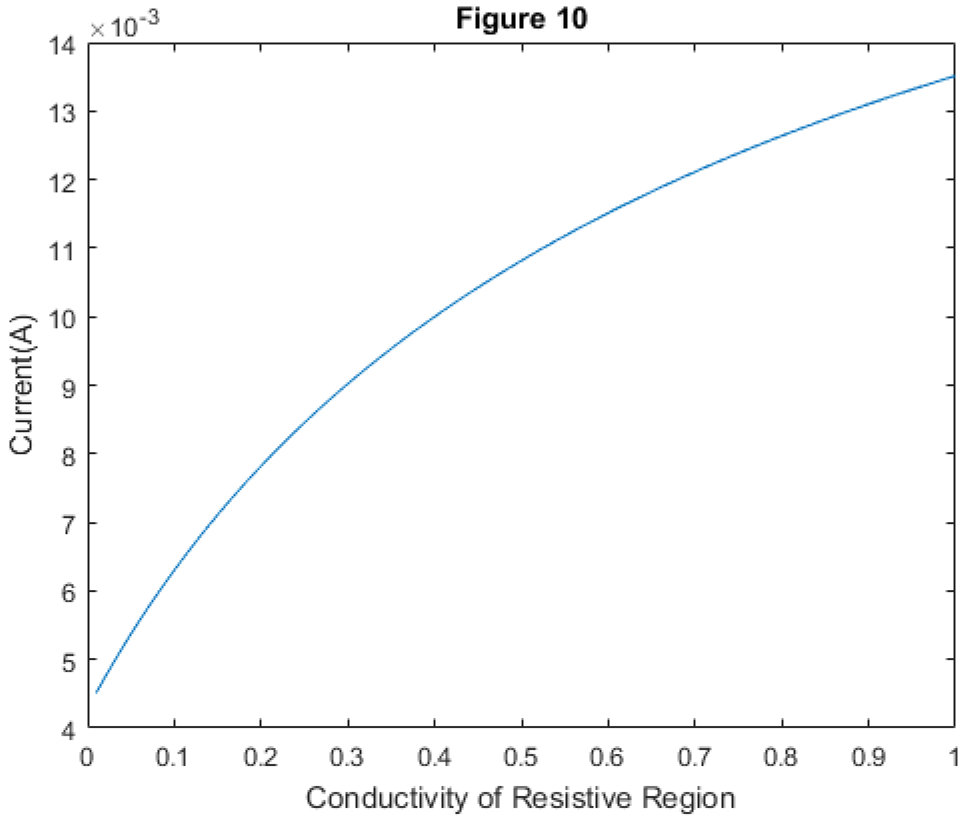


Figure 9: Current with Increasing Bottlneck Width
×10^-3
Current(A)
12
11
10
9
8
7
6
5
4
10
15
20
25
30
35
40
Width of Bottlneck



Published with MATLAB® R2017a