```
% assignment 2

% elec4700

% Huanyu Liu

% 100986552

function []=assignment2()

clear;

clc;

% all parameters defined in assignment 2

L=60;

W=40;

Vo=5;



% part 1 (a)

% G*V=z

% G is the operation

% V is the distribution

% z is the requirement of the result

k=W*L;

G=sparse(k,k);

z=zeros(k,1);

% define G

% take the idea of
```

```
for i=1:L

    for j=1:W

        n = j + (i-1)*W;

        nxm = j + (i-2)*W;

        nxp = j + i*W;

        nym = j-1+(i-1)*W;

        nyp = j+1+(i-1)*W;

        if i == 1

            G(n,n)=1;

            z(n)=Vo; % G*V=z, G is only operation, not related to values

        elseif i == L

            G(n,n) = 1;

        elseif j == 1

            G(n,n)=-3;

            G(n,nxm)=1;

            G(n,nxp)=1;

            G(n,nyp)=1;

        elseif j == W

            G(n,n)=-3;

            G(n,nxm)=1;

            G(n,nxp)=1;

            G(n,nym)=1;
```

```
        else
            G(n,n)=-4;
            G(n,nxm)=1;
            G(n,nxp)=1;
            G(n,nym)=1;
            G(n,nyp)=1;
        end
    end
end
% G*V=z
V1=G\z;
% reshape V
V1=reshape(V1,L,W);
[X,Y]=meshgrid(1:1:W,1:1:L);
figure(1)
title('part1(a)');
surf(X,Y,V1);
hold on


% part 1 (b)
V2=zeros(L,W);
V2(L,:)=Vo;
```

```
for x=(-L/2+1):L/2

    for y=1:W

        if x==(-L/2+1)||x==L/2

            V2(x+L/2,y)=Vo;

        elseif y==1||y==W

            V2(x+L/2,y)=0;

        else

        for n=1:2:87


V2(x+L/2,y)=V2(x+L/2,y)+4*Vo/pi*1/n*cosh(n*pi*x/W)/cosh(n*pi*0.5*L/W)*sin(n*pi*y/W);

        end

        end

    end

end

figure(2)

title('part1(b)');

surf(X,Y,V2);

xlabel('X');

ylabel('Y');




% part 2 (a)

    % G*V=Z
```

```
S = zeros(W, L);    % sigma

sigma1 = 1;

sigma2 = 1e-2;



% derive an expression for sigma (only used to plot sigma)

for x = 1 : L

    for y = 1 : W

        if x >= 0.4*L && x <= 0.6*L && (y <= 0.4*W || y >= 0.6*W)

            % area in the blocks

            S(y, x) = sigma2;

        else

            % area outside the blocks

            S(y, x) = sigma1;

        end

    end

end

% set up the G

[J2a, V4]=setupG( W,L,sigma1,sigma2 );



% plot for sigma

figure(3)

surf(S);
```

```
title('Surface plot of sigma')



% plot for voltage

figure(4)

surf(X',Y',V4)

title('Surface plot of voltage with bottle neck condition')


% Calculating the electric field from voltage

[Ex, Ey] = gradient(V4);


% Creating surface plots for x and y component for electric field

figure(5)

surf(-Ex)

title('Surface plot of x-component of electric field')


figure(6)

surf(-Ey)

title('Surface plot of y-component of electric field')


% Calculating the current density

Jx = S.*Ex;
```

```
Jy = S.*Ey;

J = sqrt(Jx.^2 + Jy.^2);


% Creating a surface plot for the current density

figure(7)

surf(J)

title('Surface plot of current density')


% part 2 (b)

% Creating plot for comparing current density with various mesh size

 for a=20:10:100

     [J2b,V2b]=setupG( a,3/2*a,sigma1,sigma2 );

    if a == 20

        Cy = sum(J2b, 1);  % sum of current density is current

        C = sum(Cy);  % need to get a scalar

        previousC = C;

        figure(8)

        plot([a, a], [previousC, C])  % start point

        hold on

    end

    if a > 20

        previousC = C;
```

```
        Cy = sum(J2b, 2);

        C = sum(Cy);

        figure(8)

        plot([a-10, a], [previousC, C])  % get a lime using two points, the step size
of a is 10

        hold on

    end

 end

    title('Current vs. mesh size')



    % part2(c)

    for a=0.1:0.01:0.9

        % a new function is required to sweep the width of the bottle-neck

        J2c=setupGforpart2c(a);

     if a == 0.1

        Cy = sum(J2c, 2);

        C = sum(Cy);

        previousC = C;

        figure(9)

        plot([a, a], [previousC, C])

        hold on

    end

    if a > 0.1
```

```
    previousC = C;

    Cy = sum(J2c, 2);

    C = sum(Cy);

    figure(9)

    plot([a-1e-2, a], [previousC, C])

    hold on

end

end

title('Current vs. various bottle-necks')



%part2(d)

% Creating plot for comparing current density with various sigma

for a=0.01:0.01:0.9

    [J2d,V2d]=setupG(L,W,sigma1,a);

if a == 1e-2

    Cy = sum(J2d, 2);

    C = sum(Cy);

    previousC = C;

    figure(10)

    plot([a, a], [previousC, C])

    hold on

end
```

```
    if a > 1e-2

        previousC = C;

        Cy = sum(J2d, 2);

        C = sum(Cy);

        figure(10)

        plot([a-1e-2, a], [previousC, C])

        hold on

    end

    end

    title('Current vs. Sigmas')

end

function [ J,V4 ] = setupG( L,W,sigma1,sigma2 )

k=L*W;

G=sparse(k,k);

Z = zeros(k, 1);

Vo=5;

S = zeros(L, W);    % sigma

for x = 1 : L

    for y = 1 : W

        if x >= 0.4*L && x <= 0.6*L && (y <= 0.4*W || y >= 0.6*W)

            % area in the blocks

            S(x, y) = sigma2;
```

```
        else

            % area outside the blocks

            S(x, y) = sigma1;

        end

    end

end

for i = 1:L

    for j = 1:W

        n = j + (i-1)*W;

        nxm = j + (i-2)*W;

        nxp = j + i*W;

        nym = j-1+(i-1)*W;

        nyp = j+1+(i-1)*W;

        if i == 1

            G(n, n) = 1;

            % assume the current flows from left to right

            Z(n) = Vo;

        elseif i == L

            G(n, n) = 1;

            % by default Z(n)=0 here

        elseif j == 1  % lower bound

            if i > 0.4*L && i < 0.6*L % inside the blocks
```

```
                G(n, n) = -3;

                G(n, nyp) = sigma2;

                G(n, nxp) = sigma2;

                G(n, nxm) = sigma2;

            else

                G(n, n) = -3;

                G(n, nyp) = sigma1;

                G(n, nxp) = sigma1;

                G(n, nxm) = sigma1;

            end

        elseif j == W % upper bound

            if i > 0.4*L && i < 0.6*L % inside the block

                G(n, n) = -3;

                G(n, nym) = sigma2;

                G(n, nxp) = sigma2;

                G(n, nxm) = sigma2;

            else

                G(n, n) = -3;

                G(n, nym) = sigma1;

                G(n, nxp) = sigma1;

                G(n, nxm) = sigma1;

            end
```

```
        else

            if i > 0.4*L && i < 0.6*L && (j < 0.4*W||j > 0.6*W)

                % inside the blocks

                G(n, n) = -4;

                G(n, nyp) = sigma2;

                G(n, nym) = sigma2;

                G(n, nxp) = sigma2;

                G(n, nxm) = sigma2;

            else

                G(n, n) = -4;

                G(n, nyp) = sigma1;

                G(n, nym) = sigma1;

                G(n, nxp) = sigma1;

                G(n, nxm) = sigma1;

            end

        end

    end

end

% G*V=Z

V3 = G\Z;

V4=reshape(V3,L,W);

[Ex, Ey] = gradient(V4);
```

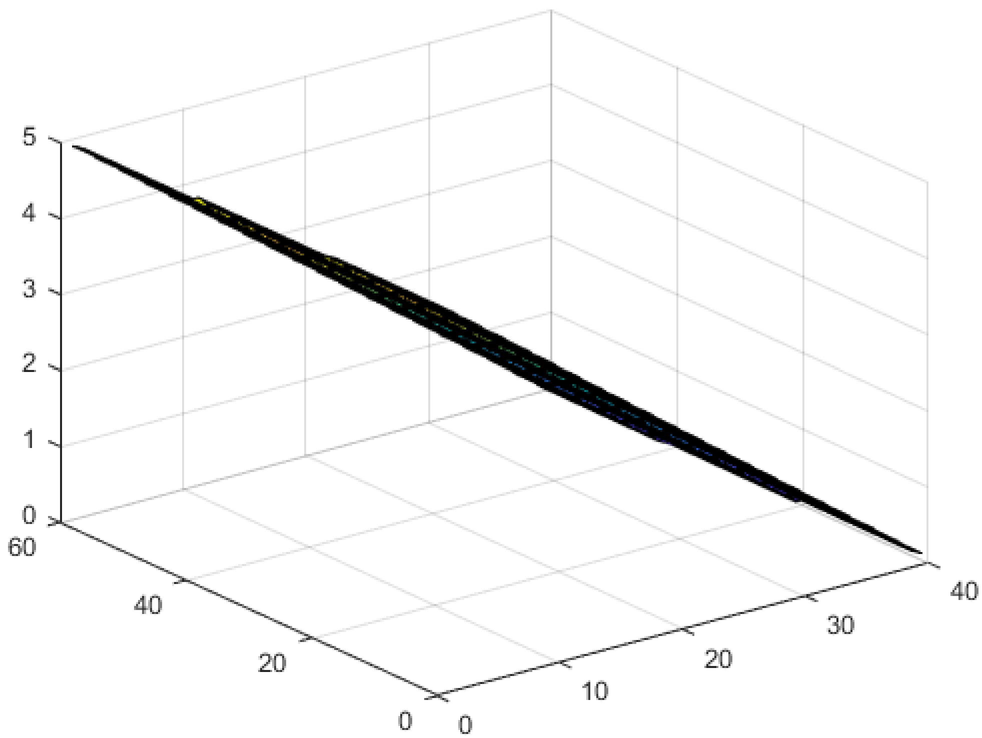
```
Jx = S.*Ex;

Jy = S.*Ey;

J = sqrt(Jx.^2 + Jy.^2);

end



function [ J ] = setupGforpart2c( a )

L=40;

W=60;

sigma1=1;

sigma2=0.01;

k=L*W;

G=sparse(k,k);

Z = zeros(k, 1);

Vo=5;

S = zeros(L, W);    % sigma

for x = 1 : L

    for y = 1 : W

        if x >= 0.4*L && x <= 0.6*L && (y <= a*W || y >= (1-a)*W)

            % area in the blocks

            S(x, y) = sigma2;

        else

            % area outside the blocks
```

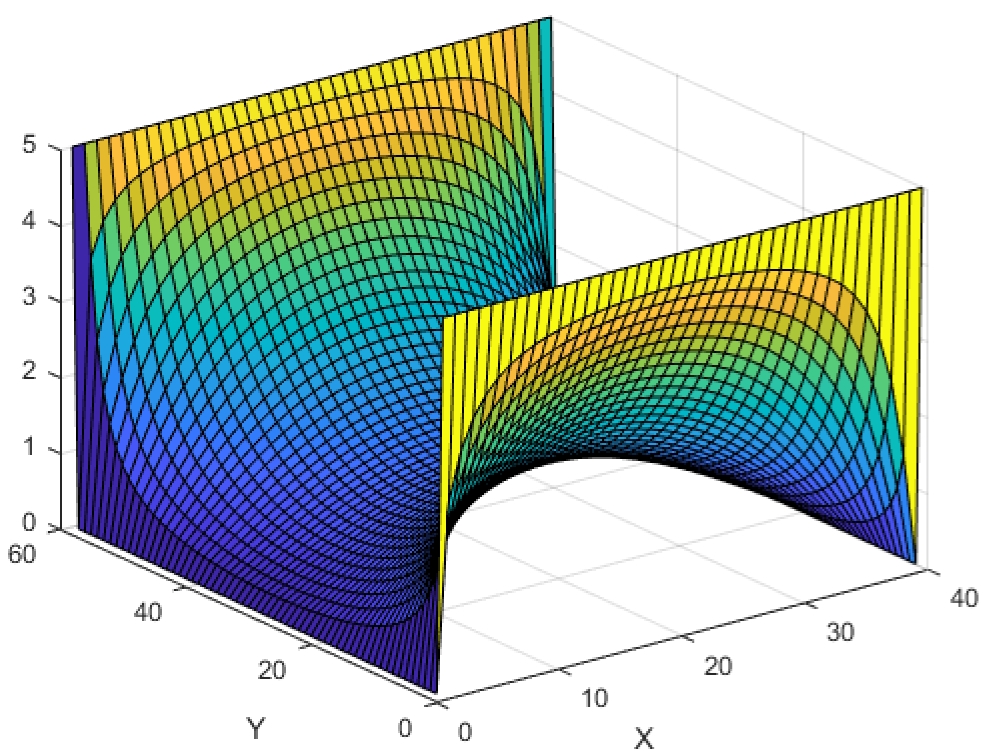
```
            S(x, y) = sigma1;

        end

    end

end

for i = 1:L

    for j = 1:W

        n = j + (i-1)*W;

        nxm = j + (i-2)*W;

        nxp = j + i*W;

        nym = j-1+(i-1)*W;

        nyp = j+1+(i-1)*W;

        if i == 1

            G(n, n) = 1;

            % assume the current flows from left to right

            Z(n) = Vo;

        elseif i == L

            G(n, n) = 1;

            % by default Z(n)=0 here

        elseif j == 1  % lower bound

            if i > 0.4*L && i < 0.6*L % inside the blocks

                G(n, n) = -3;

                G(n, nyp) = sigma2;
```

```
            G(n, nxp) = sigma2;

            G(n, nxm) = sigma2;

        else

            G(n, n) = -3;

            G(n, nyp) = sigma1;

            G(n, nxp) = sigma1;

            G(n, nxm) = sigma1;

        end

    elseif j == W % upper bound

        if i > 0.4*L && i < 0.6*L % inside the block

            G(n, n) = -3;

            G(n, nym) = sigma2;

            G(n, nxp) = sigma2;

            G(n, nxm) = sigma2;

        else

            G(n, n) = -3;

            G(n, nym) = sigma1;

            G(n, nxp) = sigma1;

            G(n, nxm) = sigma1;

        end

    else

        if i > 0.4*L && i < 0.6*L && (j < a*W||j > (1-a)*W)
```

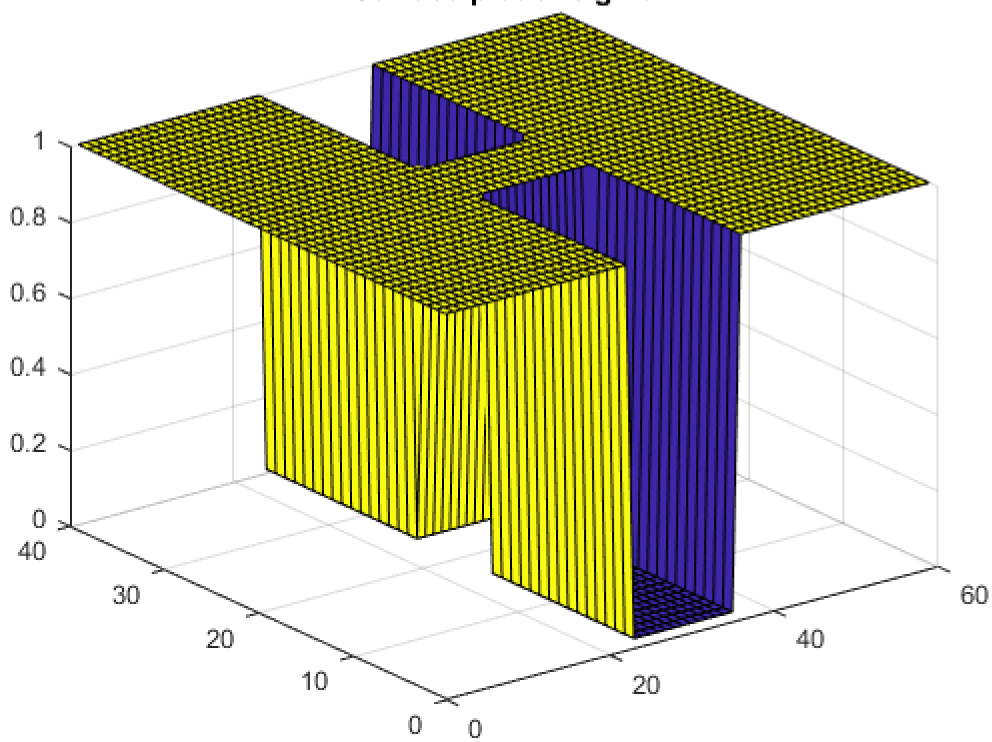
```
                % inside the blocks

                G(n, n) = -4;

                G(n, nyp) = sigma2;

                G(n, nym) = sigma2;

                G(n, nxp) = sigma2;

                G(n, nxm) = sigma2;

            else

                G(n, n) = -4;

                G(n, nyp) = sigma1;

                G(n, nym) = sigma1;

                G(n, nxp) = sigma1;

                G(n, nxm) = sigma1;

            end

        end

    end

end

% G*V=Z

V3 = G\Z;

V4=reshape(V3,L,W);

[Ex, Ey] = gradient(V4);

Jx = S.*Ex;

Jy = S.*Ey;
```

```
J = sqrt(Jx.^2 + Jy.^2);


end
```

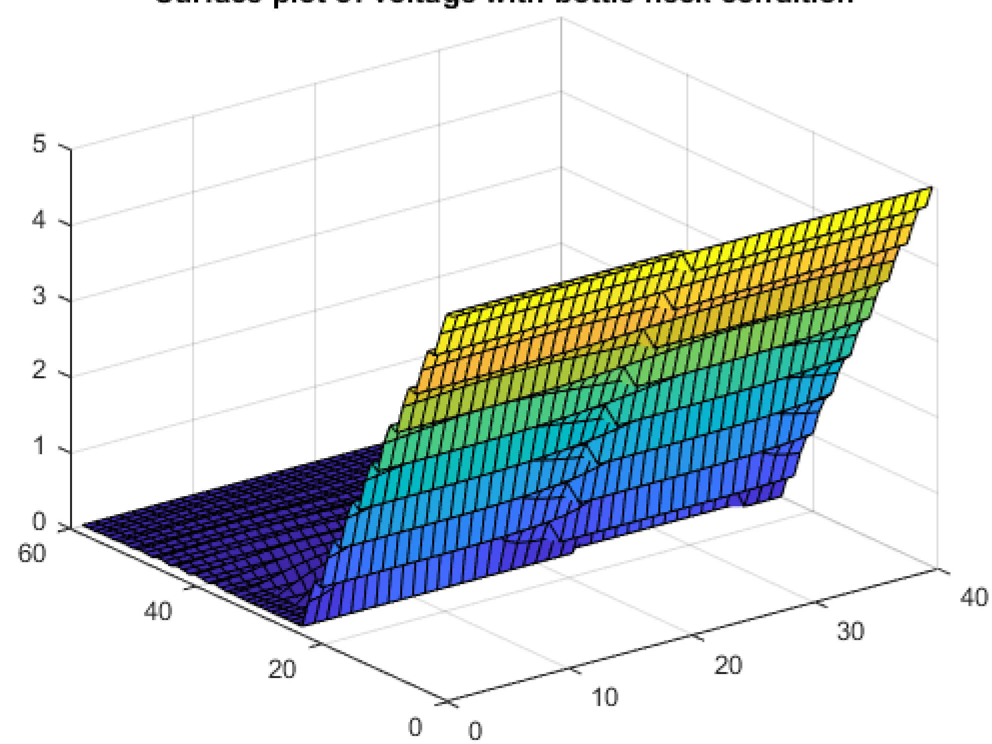

5
4
3
2
1
0
60
40
20
0
0
10
20
30
40


5
4
3
2
1
0
60
40
20
0
Y
0
10
20
30
40
X

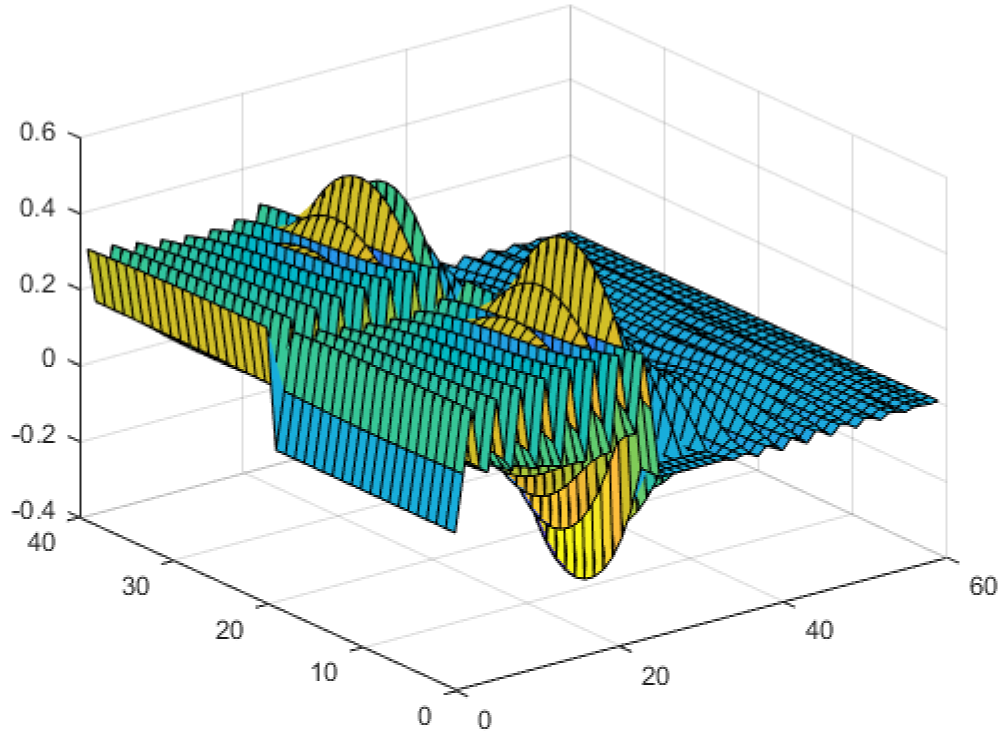
Surface plot of sigma



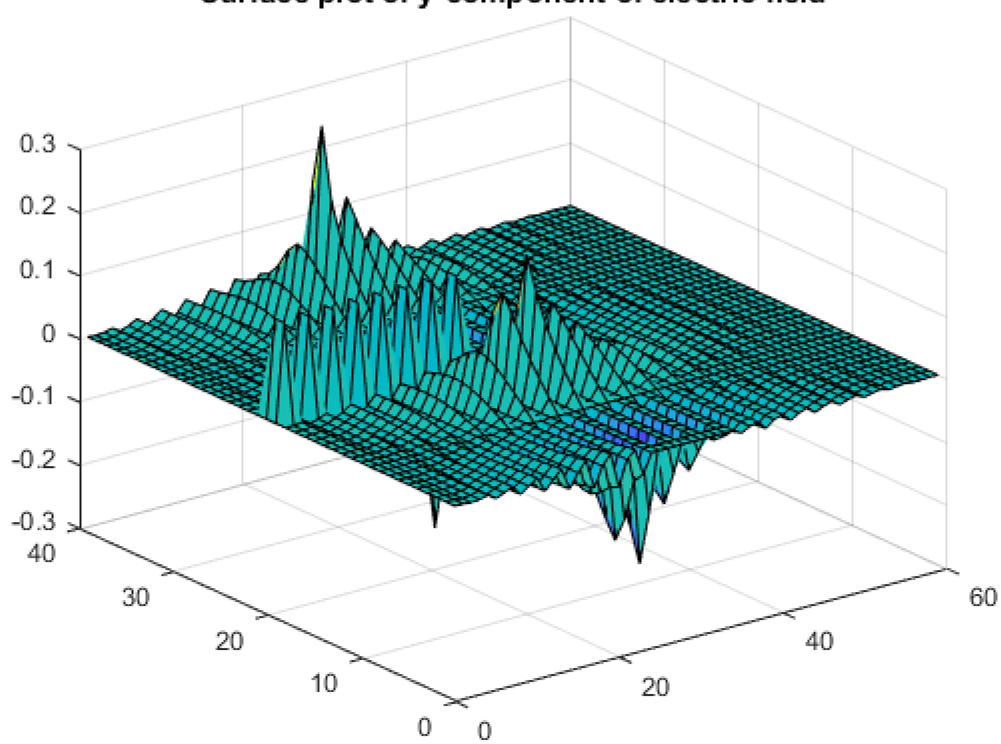
Surface plot of voltage with bottle neck condition

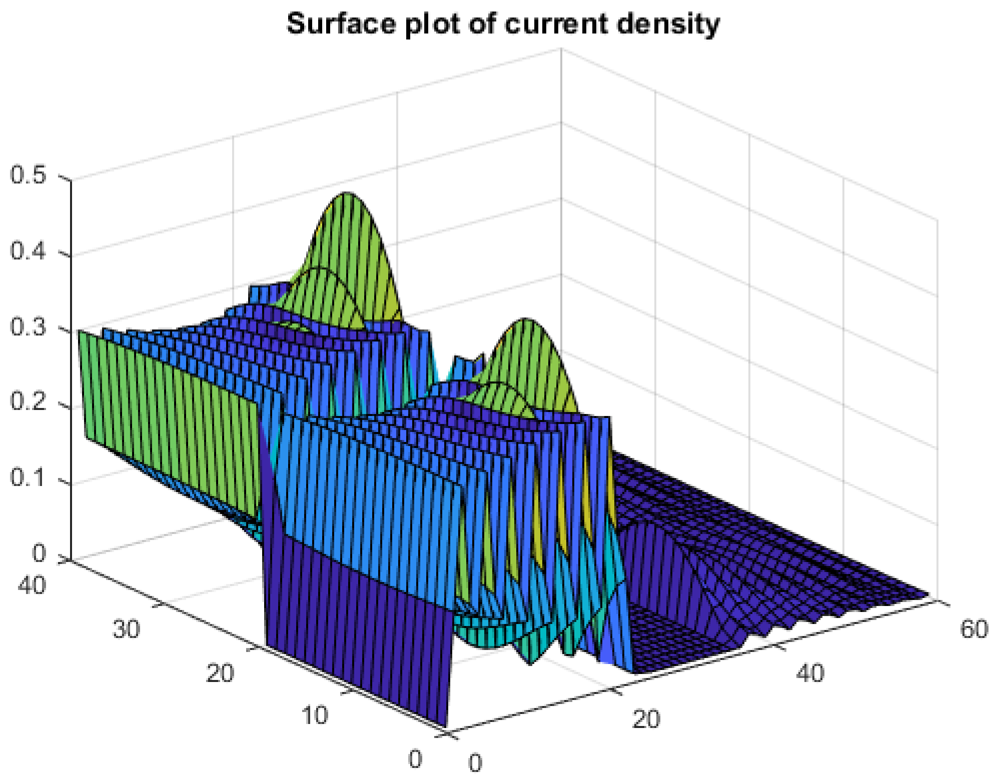


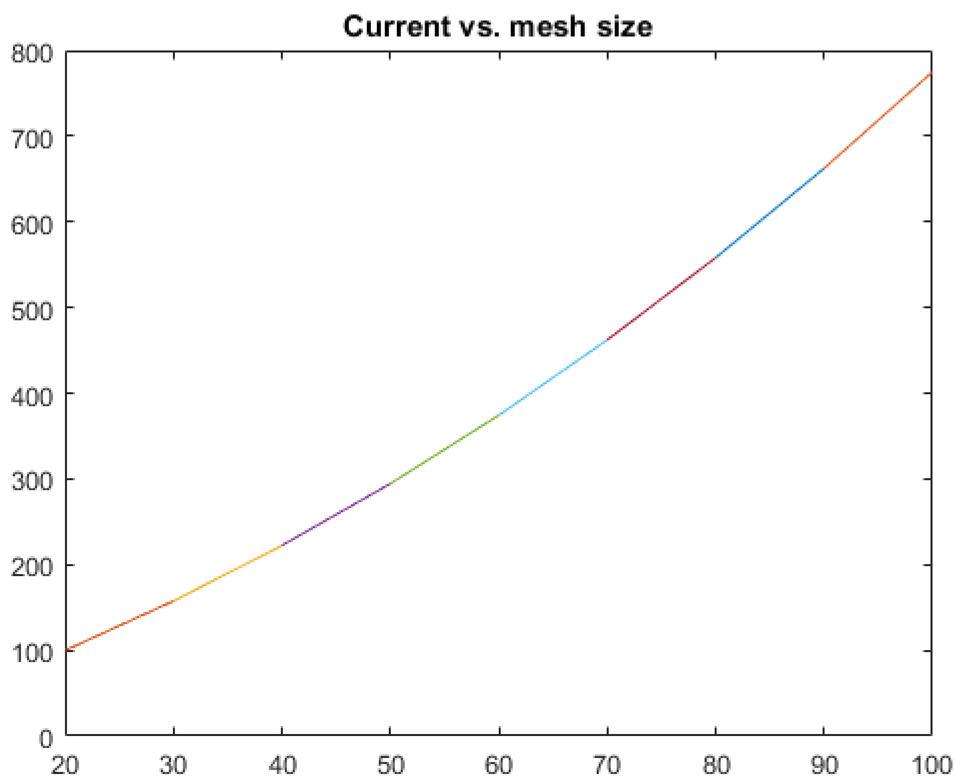
Surface plot of x-component of electric field

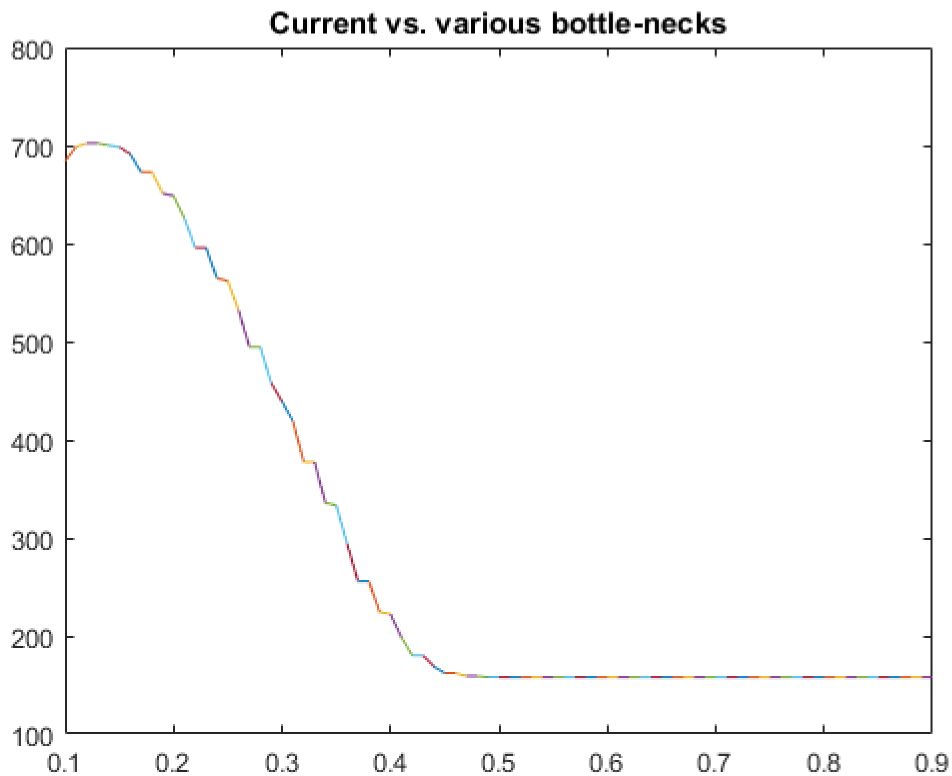


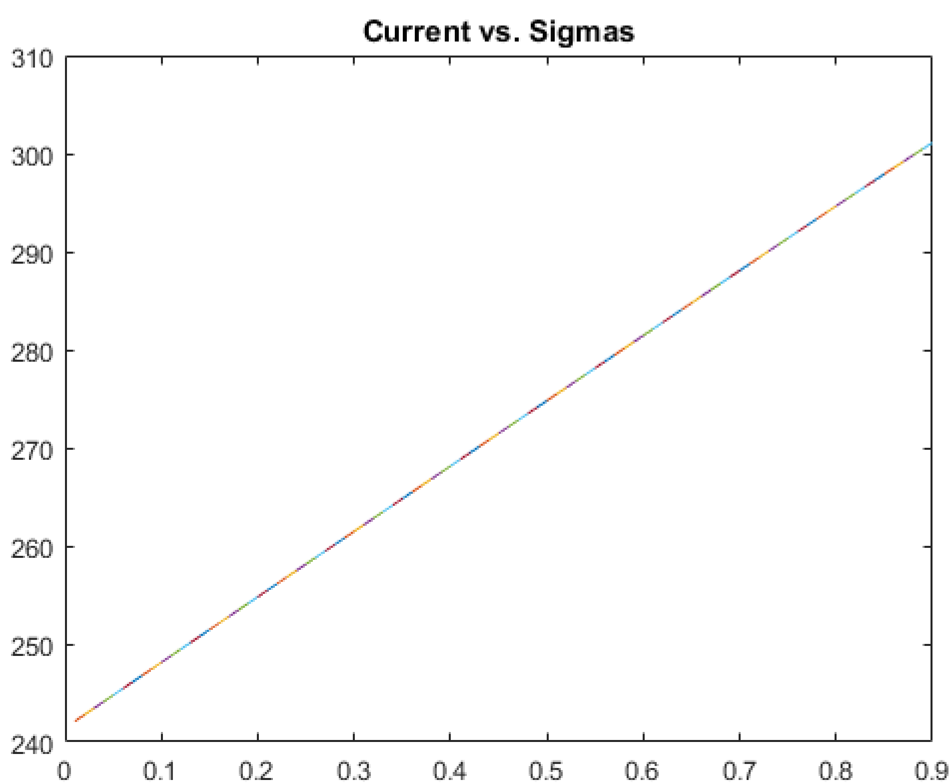
Surface plot of y-component of electric field




Surface plot of current density
0.5
0.4
0.3
0.2
0.1
0
40
30
20
10
0
0
20
40
60


Current vs. mesh size
800
700
600
500
400
300
200
100
0
20
30
40
50
60
70
80
90
100


Current vs. various bottle-necks
800
700
600
500
400
300
200
100
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Current vs. Sigmas
310
300
290
280
270
260
250
240
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9