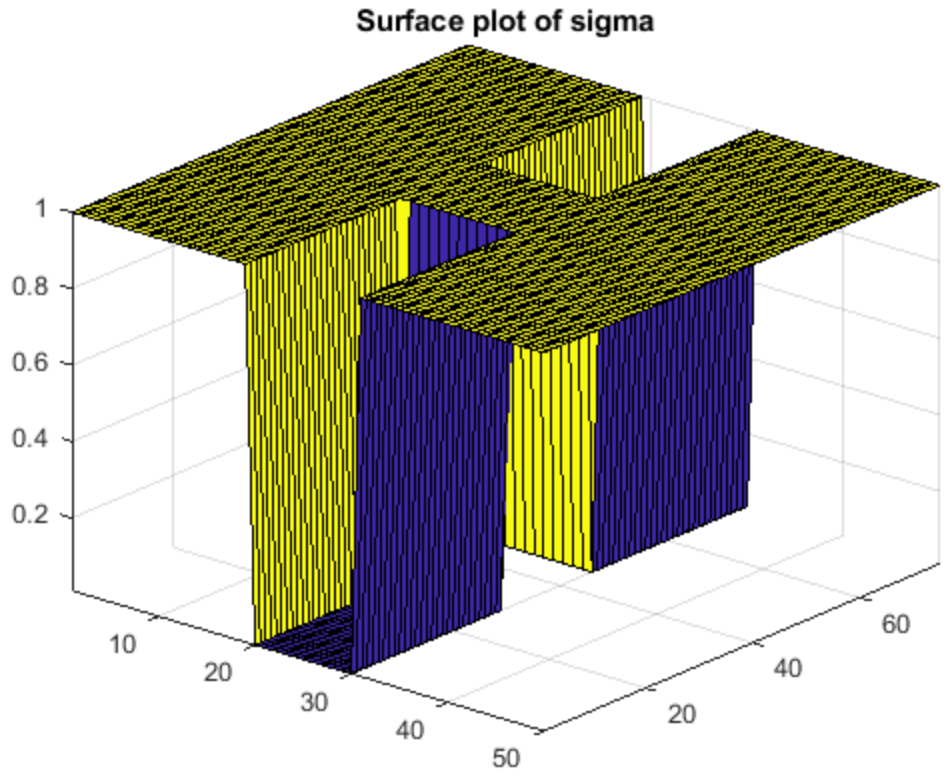
Part 2(a)

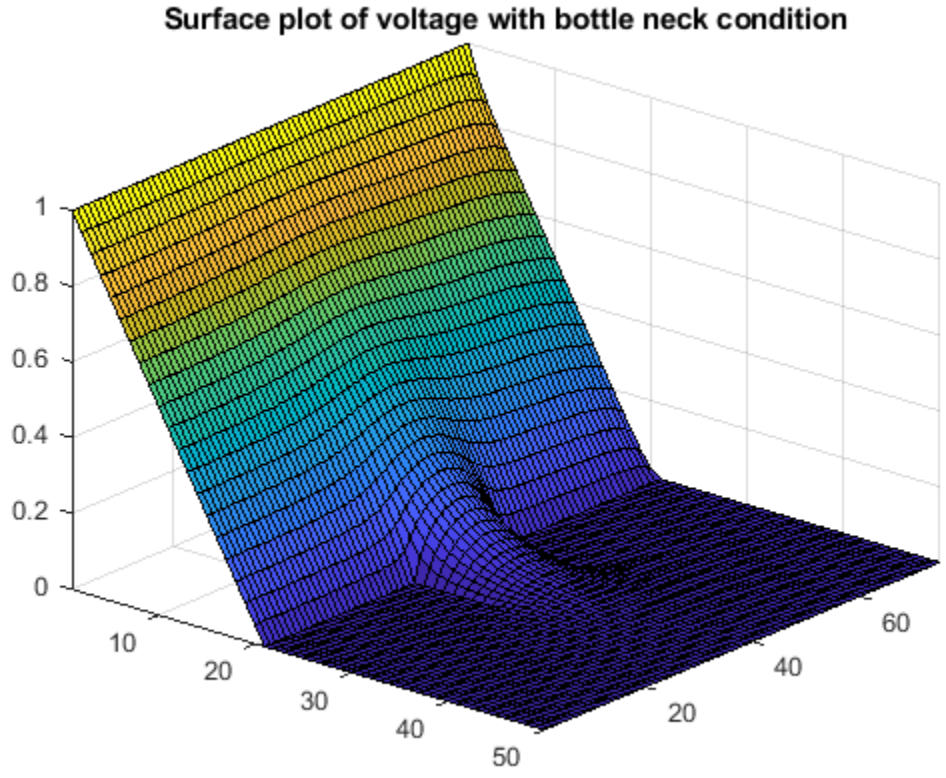
In part 2(a), we are now trying to model the current flow in the region with a bottle neck added.

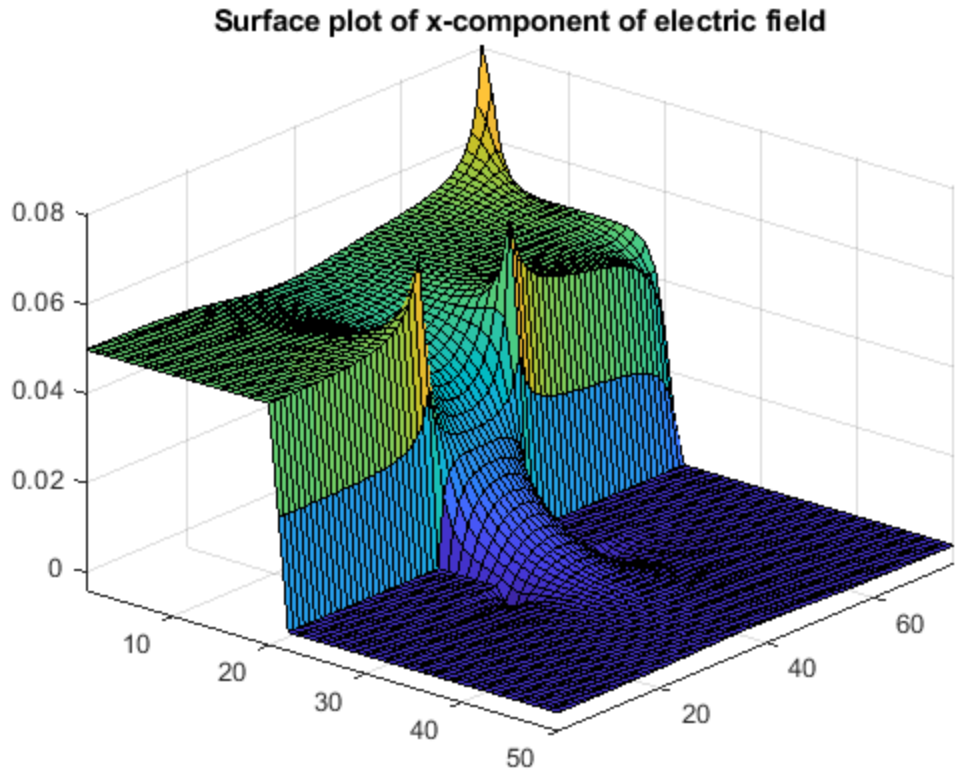
```
% Reset Everything
close all
clear

% Setting variables
nx = 50;                % Length of the region
ny = nx*3/2;            % Width of the region, 3/2 of length
G = sparse(nx*ny);      % Initialize a G matrix
D = zeros(1, nx*ny);    % Initialize a matrix for G matrix operation
S = zeros(ny, nx);      % Initialize a matrix for sigma
sigma1 = 1;             % Setting up parameter of sigma in different
 region
sigma2 = 1e-2;
box = [nx*2/5 nx*3/5 ny*2/5 ny*3/5]; % Setting up the bottle neck

% Implement the G matrix with the bottle neck condition in the region
for i = 1:nx
    for j = 1:ny
        n = j + (i-1)*ny;

        if i == 1
            G(n, :) = 0;
            G(n, n) = 1;
            D(n) = 1;
        elseif i == nx
            G(n, :) = 0;
            G(n, n) = 1;
            D(n) = 0;
        elseif j == 1
            if i > box(1) && i < box(2)
                G(n, n) = -3;
                G(n, n+1) = sigma2;
                G(n, n+ny) = sigma2;
                G(n, n-ny) = sigma2;
            else
                G(n, n) = -3;
                G(n, n+1) = sigma1;
                G(n, n+ny) = sigma1;
                G(n, n-ny) = sigma1;
            end
        elseif j == ny
            if i > box(1) && i < box(2)
                G(n, n) = -3;
                G(n, n+1) = sigma2;
                G(n, n+ny) = sigma2;
                G(n, n-ny) = sigma2;
            else
                G(n, n) = -3;
```

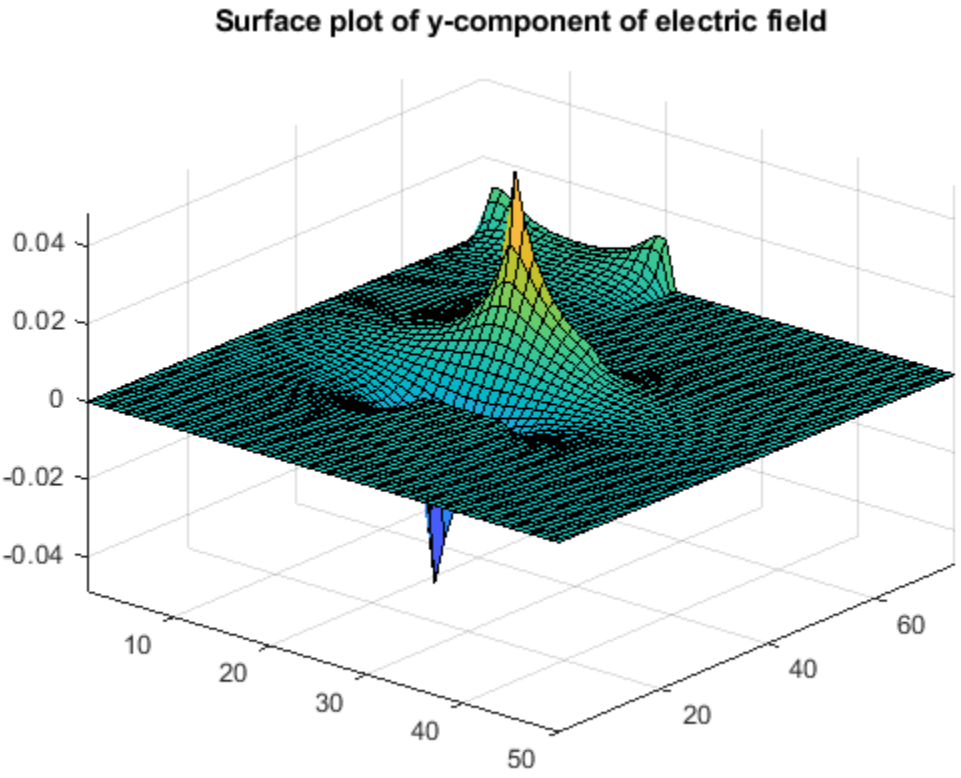
```
                G(n, n+1) = sigma1;
                G(n, n+ny) = sigma1;
                G(n, n-ny) = sigma1;
            end
        else
            if i > box(1) && i < box(2) && (j < box(3)||j > box(4))
                G(n, n) = -4;
                G(n, n+1) = sigma2;
                G(n, n-1) = sigma2;
                G(n, n+ny) = sigma2;
                G(n, n-ny) = sigma2;
            else
                G(n, n) = -4;
                G(n, n+1) = sigma1;
                G(n, n-1) = sigma1;
                G(n, n+ny) = sigma1;
                G(n, n-ny) = sigma1;
            end
        end
    end
end

% Implement a matrix for sigma
for L = 1 : nx
    for W = 1 : ny
        if L >= box(1) && L <= box(2)
            S(W, L) = sigma2;
        else
            S(W, L) = sigma1;
        end
        if L >= box(1) && L <= box(2) && W >= box(3) && W <= box(4)
            S(W, L) = sigma1;
        end
    end
end

% Creating a surface plot for sigma
figure(1)
surf(S);
axis tight
view([40 30]);
title("Surface plot of sigma")

% Calculating the voltage
V = G\D';

% Inverting the G matrix
X = zeros(ny, nx, 1);
for i = 1:nx
    for j = 1:ny
        n = j + (i-1)*ny;
        X(j,i) = V(n);
    end
end
```

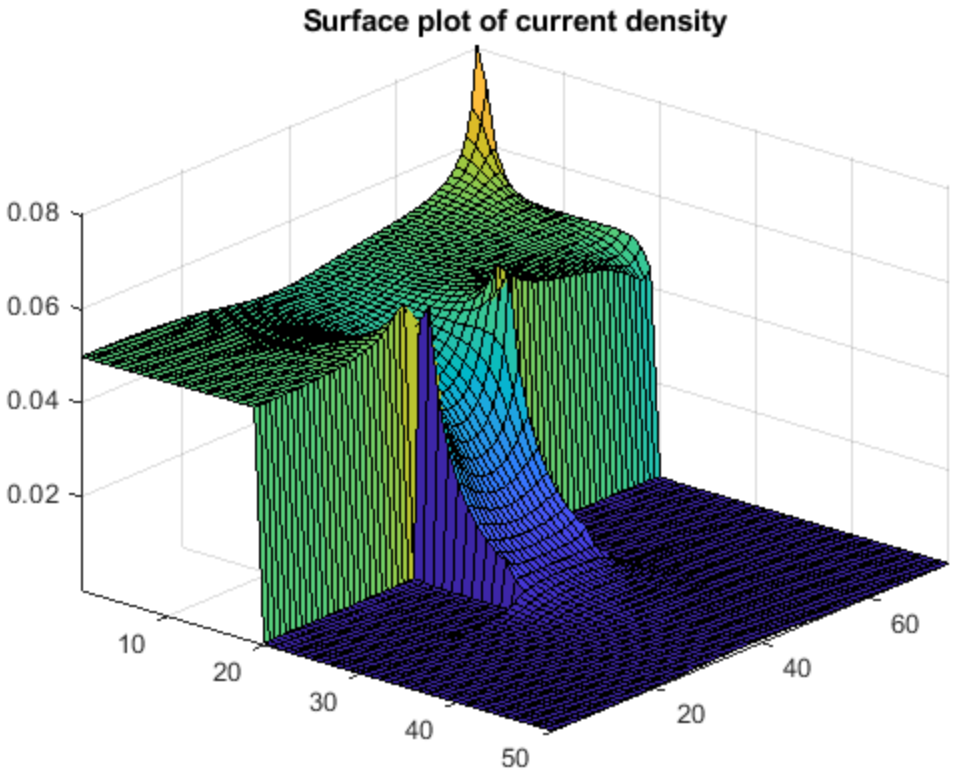
```
% Creating a surface plot for voltage
figure(2)
surf(X)
axis tight
view([40 30]);
title("Surface plot of voltage with bottle neck condition")

% Calculating the electric field from voltage
[Ex, Ey] = gradient(X);

% Creating surface plots for x and y component for electric field
figure(3)
surf(-Ex)
axis tight
view([40 30]);
title("Surface plot of x-component of electric field")

figure(4)
surf(-Ey)
axis tight
view([40 30]);
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(5)
surf(J)
axis tight
view([40 30]);
title("Surface plot of current density")
```


Surface plot of sigma
1
0.8
0.6
0.4
0.2
10
20
30
40
50
20
40
60


Surface plot of voltage with bottle neck condition
1
0.8
0.6
0.4
0.2
0
10
20
30
40
50
20
40
60


Surface plot of x-component of electric field
0.08
0.06
0.04
0.02
0
10
20
30
40
50
20
40
60


Surface plot of y-component of electric field
0.04
0.02
0
-0.02
-0.04
10
20
30
40
50
20
40
60



Published with MATLAB® R2017b