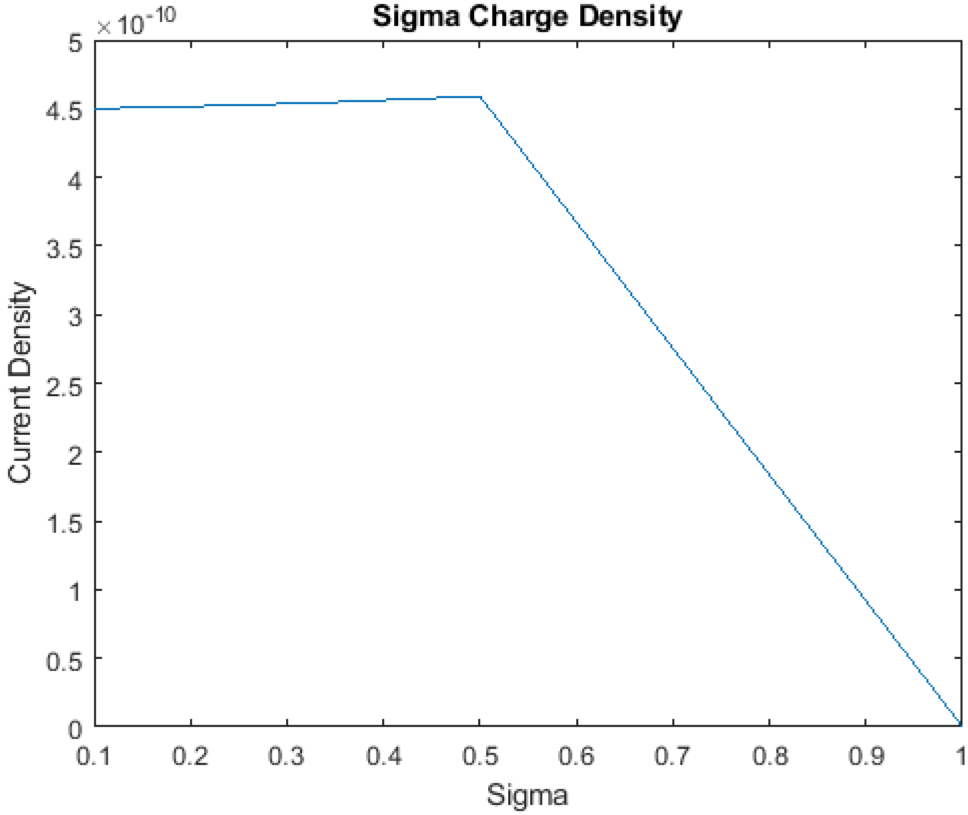
Question 2- Part D

```
%In this part, the value of sigma will be varied and observing the
 effect
%on current density

I = zeros(1,10);

for k =1:10
    sigma(k) = 1/(k);

    %Setting up Grid dimension
    Length=150;
    Width=100;

    G=sparse(Length*Width,Length*Width);
    V=zeros(Length*Width,1);

    sigOut=1;
    sigIn=sigma(k);

    %Setting up bottleneck parameters
    midX = Length/2;
    midY = Width/2;
    boxL = Length/4;
    boxW = Width*2/3;
    leftBC = midX - boxL/2;
    rightBC = midX + boxL/2;
    topBC = midY + boxW/2;
    bottomBC = midY - boxW/2;


    % Populating G matrix
    for i=1:Length
        for j=1:Width
            n=j+(i-1)*Width; %Current position

            nxm = j+(i-2)*Width;
            nxp = j+(i)*Width;
            nym = (j-1)+(i-1)*Width;
            nyp = (j+1)+(i-1)*Width;
              if i == 1
                G(n,n) = 1;
                V(n) = 1;
                sigmaMap(i,j) = sigOut;
            elseif i == Length
                G(n,n) = 1;
                V(n) = 0;
                sigmaMap(i,j) = sigOut;
            elseif (j == Width)
                G(n,n) = -3;
                if(i>leftBC && i<rightBC)
                G(n,nxm) = sigIn;
                G(n,nxp) = sigIn;
                G(n,nym) = sigIn;
                sigmaMap(i,j) = sigIn;
            else
                G(n,nxm) = sigOut;
                G(n,nxp) = sigOut;
                G(n,nym) = sigOut;
                sigmaMap(i,j) = sigOut;
            end
        elseif (j == 1)
            G(n,n) = -3;
            if(i>leftBC && i<rightBC)
                G(n,nxm) = sigIn;
                G(n,nxp) = sigIn;
                G(n,nyp) = sigIn;
                sigmaMap(i,j) = sigIn;
            else
                G(n,nxm) = sigOut;
                G(n,nxp) = sigOut;
                G(n,nyp) = sigOut;
                sigmaMap(i,j) = sigOut;
            end
        else
            G(n,n) = -4;
            if( (j>topBC || j<bottomBC) && i>leftBC && i<rightBC)
                G(n,nxp) = sigIn;
                G(n,nxm) = sigIn;
                G(n,nyp) = sigIn;
                G(n,nym) = sigIn;
                sigmaMap(i,j) = sigIn;
            else
                G(n,nxp) = sigOut;
                G(n,nxm) = sigOut;
                G(n,nyp) = sigOut;
                G(n,nym) = sigOut;
                sigmaMap(i,j) = sigOut;
            end
        end
    end
end

SolV = G\V;
SolVmatrix=zeros(Length,Width);

for i=1:Length
    for j=1:Width
        n=j+(i-1)*Width;
        SolVmatrix(i,j)= SolV(n);
    end
end

[Ey,Ex] = gradient(SolVmatrix);
E = gradient(SolVmatrix);
```

```
    J = -sigmaMap.* E;

    region = Length*Width;
    I(k)= (sum(sum(J))/(Length*Width))/region;
end


    figure (1)
    plot(sigma,I);
    title('Sigma Charge Density')
    xlabel('Sigma')
    ylabel('Current Density')
```



Published with MATLAB® R2019b