
Table of Contents

Ovwagbedia oghenero	1
101040228	1
1a	1
1b	1
1c	1
part 2	6
part 3	9

Ovwagbedia oghenero

101040228

1a

```
%The electric field equation is given by  $E = V/d$ 
%
% For a 0.1V voltage applied and a distance of 200nm, we obtain an
% electric field of
% magnitude 500
```

1b

```
%The force equation is given by  $F = E*q$ 
%
% with an electric field of 500 kN/C and the electron charge of
% 0.1602e-18 C, we obtain a force of 8.01e-13 N.
```

1c

The Acceleration equation is given by $a = F/m$

With an applied force of 8.01e-13 N and an electron mass of 9.109e-31 kg, we obtain an acceleration of 8.7935e17 m/s².

```
k = 1.28e-23; %J/K??
mo = 9.1e-31; %kg
mn = 0.26 * mo; %effective mass of electrons
T = 300; %K
v_th = sqrt((2*k*T)/mn); %solving for thermal velocity
tmn = 0.2e-12; %seconds(mean time between collisions)
nelectrons = 20;
q = 1.60217653e-19;

elecpop = 1000;
elecsiz = 4000;
row = zeros(elecsiz, 4);
traj = zeros(1000, nelectrons*2);
```

```

temp = zeros(1000,1);
width = 200e-9; %metres
height = 100e-9; %metres
deltaT= width/v_th/100;
area = width * height; %the area of the region
md = sqrt(k*T/mn).*randn(2,elecpop);
angle = rand(1,nelectrons).*2*3.14;

x_vel = v_th.*cos(angle);
Y_vel = v_th.*sin(angle);
for i = 1:elecsz
    angle = rand*2*3.14;
    row(i,:)= [height*rand width*rand v_th*cos(angle) v_th*sin(angle)];
end
for i = 1:1000
    %update old position with new
    row(:,1:2) = row(:,1:2) + deltaT.*row(:,3:4);

    % collisions with the boundaries
    n = row(:,1) < 0;
    row(n,1) = row(n,1) + 100e-9;

    n = row(:,1) > height;
    row(n,1) = row(n,1) - 100e-9;

    n = row(:,2) > width;
    row(n,2) = 2*width - row(n,2);
    row(n,4) = -row(n,4);

    n = row(:,2) < 0;
    row(n,2) = -row(n,2);
    row(n,4) = -row(n,4);

    s = sqrt(x_vel(1,:).^2 + Y_vel(1,:).^2);
    temp(n) = ((mean(s)^2)*mn)/(2*k);

    ts = linspace(0,deltaT*1000, 1000);

    % plotting the tranectories
    for n=1:nelectrons
        traj(i, (2*n):(2*n+1)) = row(n, 1:2);
    end
end

% Plot of the trajectories of the electrons

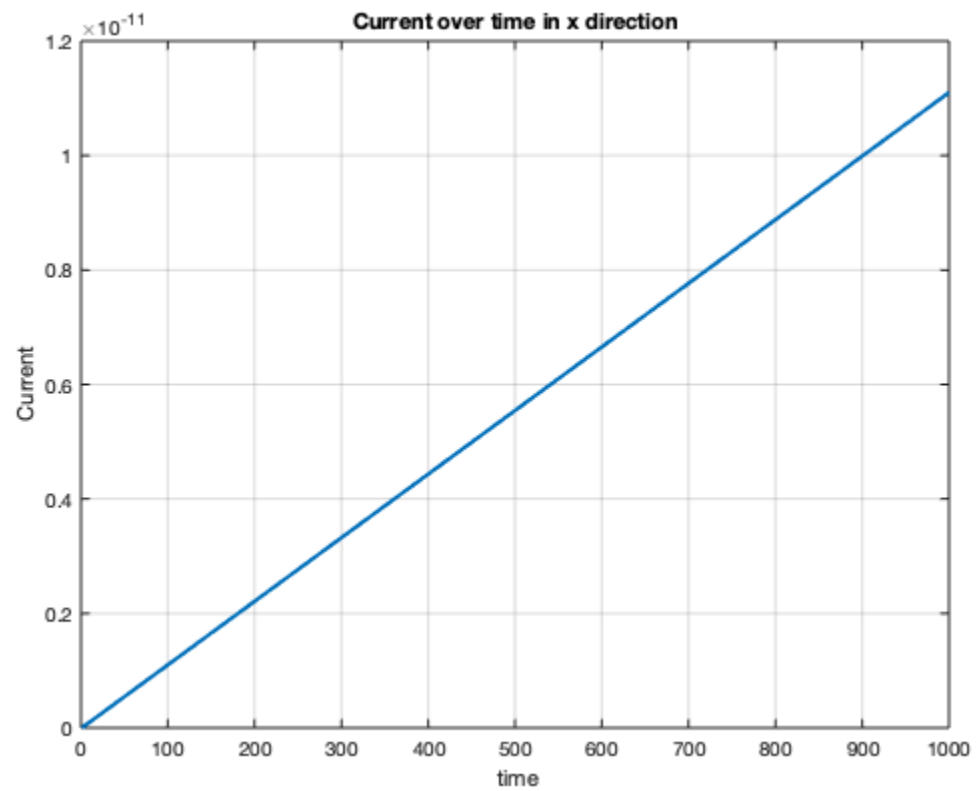
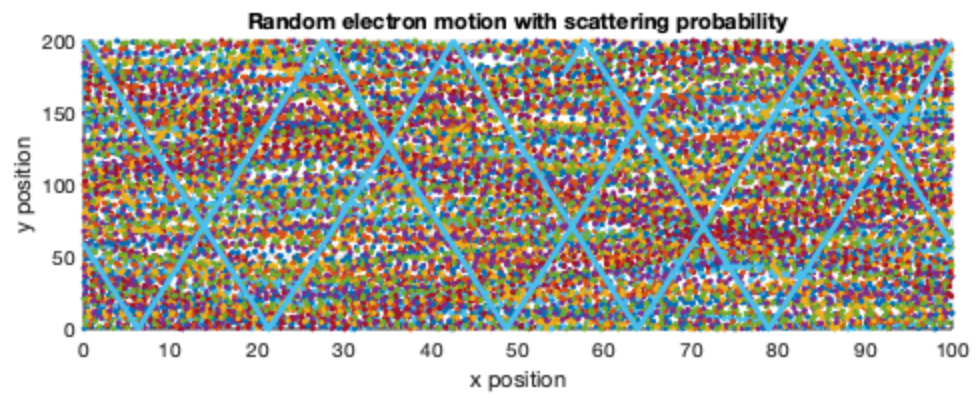
figure(1);
subplot(2,1,1);

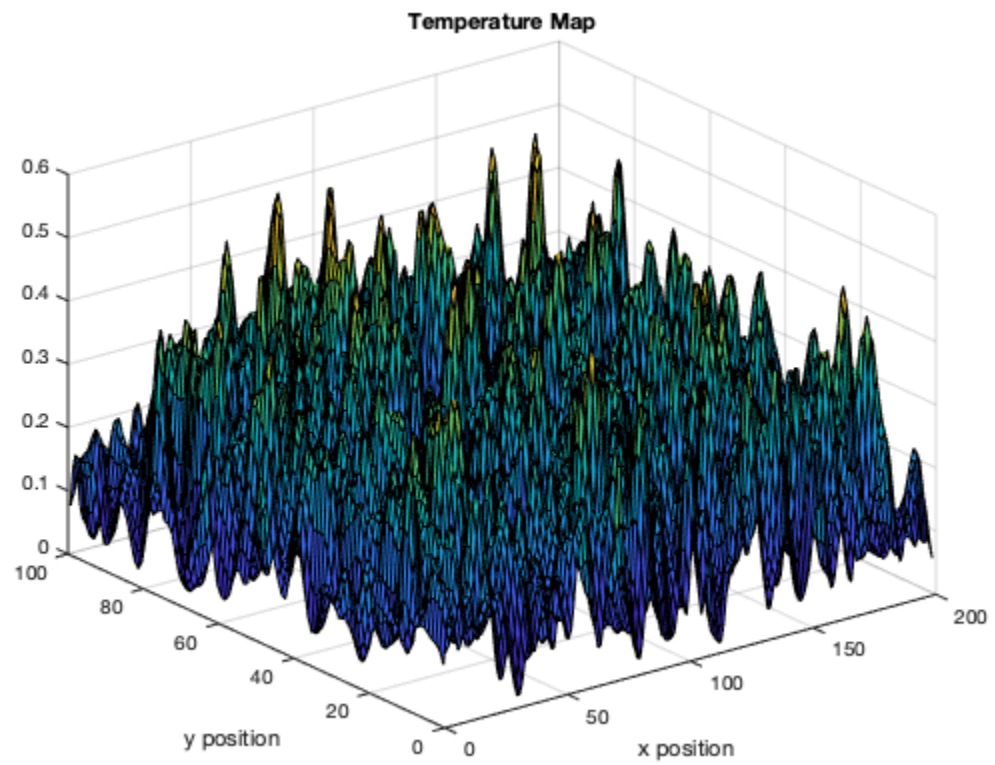
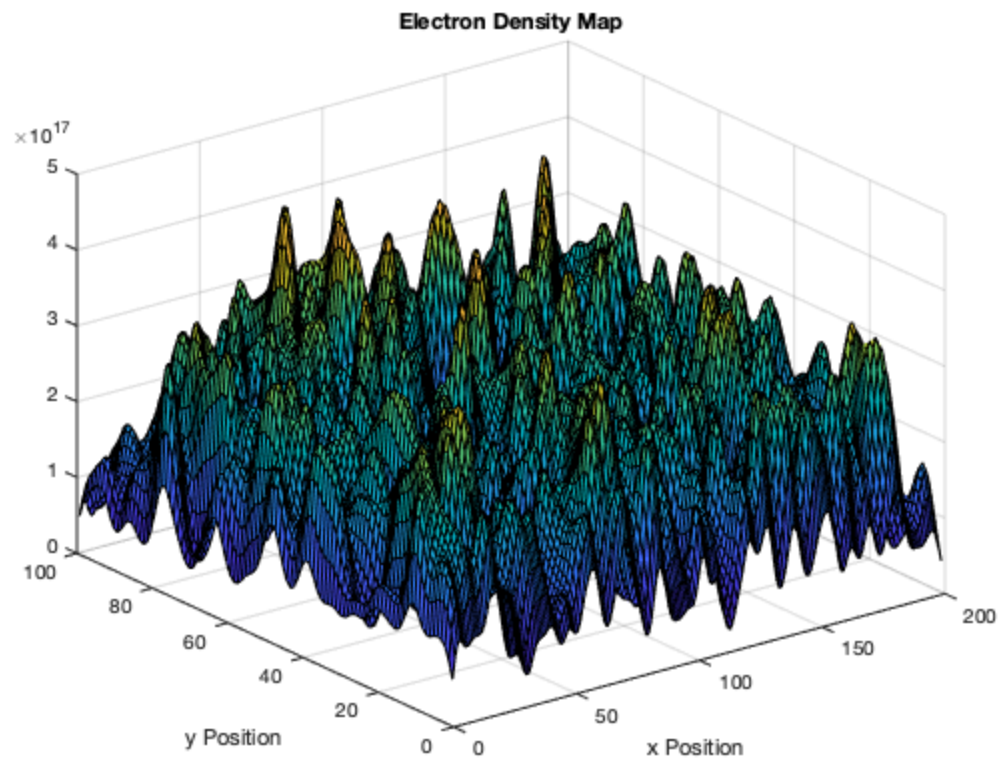
```

```

grid on
title('Random electron motion with scattering probability');
xlim([0 height/1e-9]);
ylim([0 width/1e-9]);
xlabel('x position');
ylabel('y position');
hold on;
for i=1:nelectrons
    plot(traj(:,i*2)./1e-9, traj(:,i*2+1)./1e-9, '.');
end
figure(2)
plot(ts,'linewidth',2)
title('Current over time in x direction')
xlabel(' time')
ylabel('Current')
grid on
edens = hist3(row(:,1:2),[200 100]);
N = 20;
st = 1.5;
[x y] = meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*st^2)-y.^2/(2*st^2));
f=f./sum(f(:));
figure(3);
edens = conv2(edens,f,'same');
edens = edens/(width./size(edens,1)*height./size(edens,2));
surf(conv2(edens,f,'same'));
title('Electron Density Map');
xlabel('x Position ');
ylabel('y Position ');
temp = hist3(row(:,1:2),[200 100]);
N = 20;
st = 1.5;
[x y] = meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*st^2)-y.^2/(2*st^2));
f=f./sum(f(:));
figure(4);
surf(conv2(temp,f,'same'));
title('Temperature Map');
xlabel('x position');
ylabel('y position');

```





part 2

```
V0 = 1;

% Define Matrices

regx = 50;
regy = 75;
sp = sparse(regx*regy,regx*regy);
B = zeros(regx*regy,1);
Bot = 1e-10;
con=1.*ones(regx,regy);

for i = 1:regx
    for j = 1:regy
        if(((i>=20)&&(i<=180))||((j<=40)&&(j>=60)))
            con(i,j) = Bot;
        end
    end
end
for i = 1:regx
    for j = 1:regy
        n = j + (i-1)*regy;
        ax = (i-2)*regy + j;
        bx = i*regy + j;
        cx = (i-1)*regy + j-1;
        dx = (i-1)*regy + j+1;

        if i == regx
            sp(n,n) = 1;
            B(n,1) = 1;
        else if i == 1
            sp(n,1) = 1;
            sp(n,n) = 1;
            B(n) = V0;

            elseif j == regy
                B(n) = 0;
                conxm = (con(i,j) + con(i-1,j))/2;
                conxp = (con(i,j) + con(i-1,j))/2;;
                conym = (con(i,j) + con(i-1,j))/2;

                sp(n,n) = -(conxm+conxp+conyp);
                sp(n,ax) = conxm;
                sp(n,bx) = conxp;
                sp(n,cx) = conyp;

            elseif j == 1
                conxm = (con(i,j) + con(i-1,j))/2;
                conxp = (con(i,j) + con(i+1,j))/2;
                conyp = (con(i,j) + con(i,j+1))/2;
                sp(n,n) = -(conxm+conxp+conyp);
```

```

        sp(n,ax) = conxm;
        sp(n,bx) = conxp;
        sp(n,cx) = conyp;
        B(n) = 0;

    else
        conxm = (con(i,j) + con(i-1,j))/2;
        conxp = (con(i,j) + con(i+1,j))/2;
        conyp = (con(i,j) + con(i,j+1))/2;
        conym = (con(i,j) + con(i,j+1))/2;
        sp(n,n) = -(conxm+conxp+conyp);
        sp(n,ax) = conxm;
        sp(n,bx) = conxp;
        sp(n,cx) = conyp;
    end
end
end

v=sp\B;
dz = zeros(regx,regy);
dx = linspace(1,regx,regx);
dy = linspace(1,regy,regy);

for i = 1:regx
    for j = 1:regy
        n = j+(i-1)*regy;
        dz(i,j) = v(n,1);
    end
end

figure(5)
surf(con)
colorbar
grid on
view(-125,45)
title('Figure 5: Plot of V(x,y) for Bottle Neck')

[Ex,Ey] = gradient(dz',1,1);
figure(6)
quiver(Ex, Ey, 2, 'r')
title('Electric Field of Region')
xlabel('Region Width')
ylabel('Region Length')
grid on

```

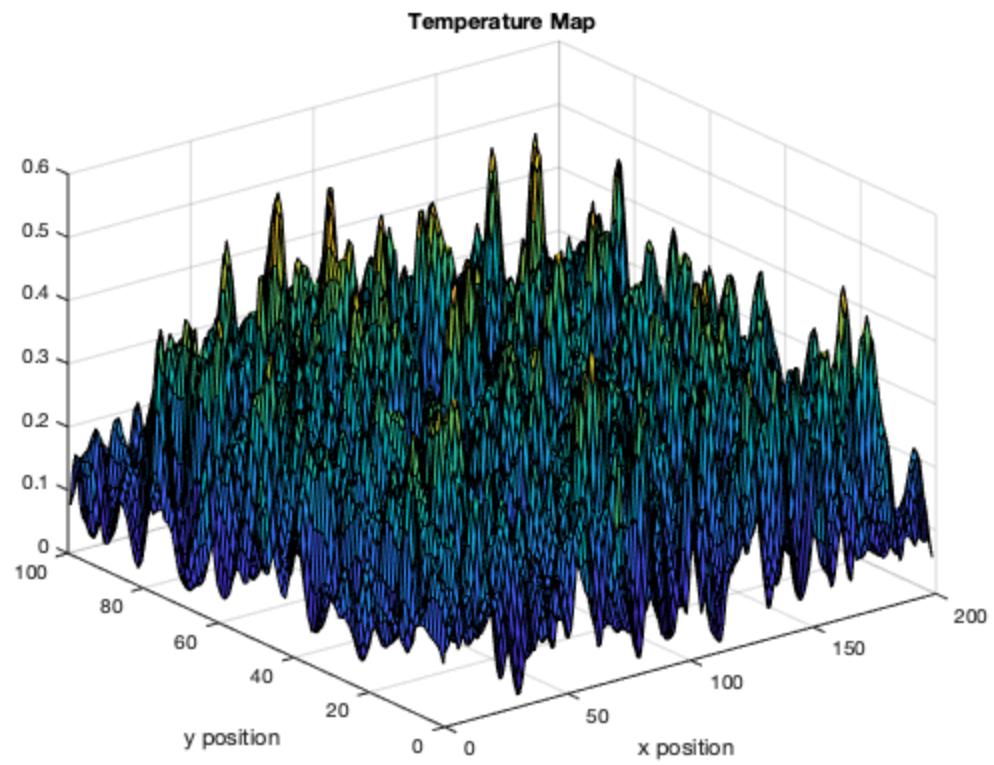
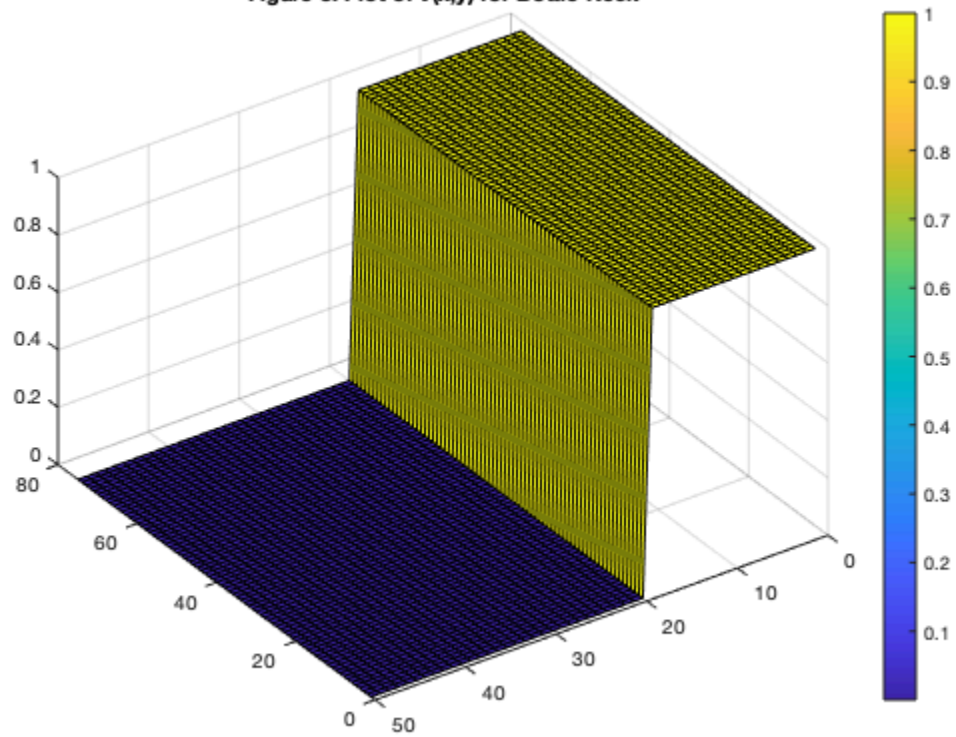
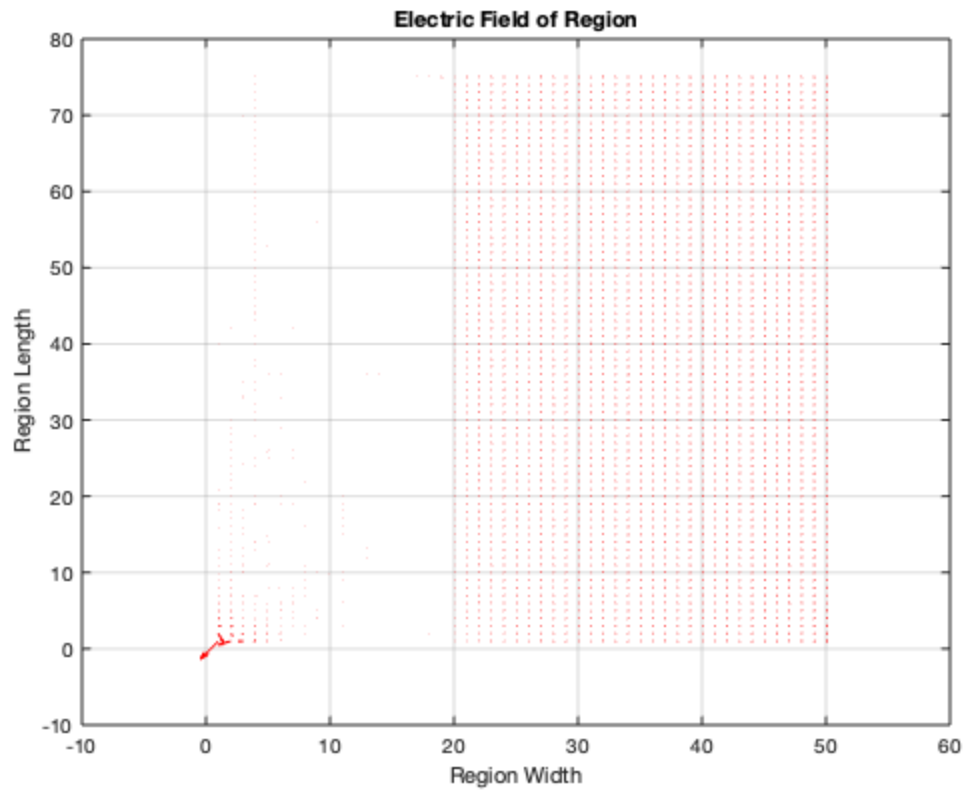


Figure 5: Plot of $V(x,y)$ for Bottle Neck





part 3

```
edens = hist3(row(:,1:2),[200 100])';
N = 20;
st = 1.5;
[x y] = meshgrid(round(-N/2):round(N/2), round(-N/2):round(N/2));
f=exp(-x.^2/(2*st^2)-y.^2/(2*st^2));
f=f./sum(f(:));
figure(7);
edens = conv2(edens,f,'same');
edens = edens/(width./size(edens,1)*height./size(edens,2));
surf(conv2(edens,f,'same'));
title('Electron Density Map');
xlabel('x Position ');
ylabel('y Position ');
A = zeros(1,50);
for cs = 1:50
    sol = ones(regx,regy).*(1/(6.2e2));
    for i = 1:regx
        for j = 1:regy
            if((i>=20)&&(i<=180)||((j<=40))&&(j>=60)))
                con(i,j) = Bot;
            end
        end
    end
end
end
```

```

for i = 1:regx
    for j = 1:regy
        n = j + (i-1)*regy;
        ax = (i-2)*regy + j;
        bx = i*regy + j;
        cx = (i-1)*regy + j-1;
        dx = (i-1)*regy + j+1;

        if i == regx
            sp(n,n) = 1;
            B(n,1) = 1;
        else if i == 1
            sp(n,1) = 1;
            sp(n,n) = 1;
            B(n) = V0;

            elseif j == regy
                B(n) = 0;
                conxm = (con(i,j) + con(i-1,j))/2;
                conxp = (con(i,j) + con(i-1,j))/2;
                conym = (con(i,j) + con(i-1,j))/2;

                sp(n,n) = -(conxm+conxp+conyp);
                sp(n,ax) = conxm;
                sp(n,bx) = conxp;
                sp(n,cx) = conyp;

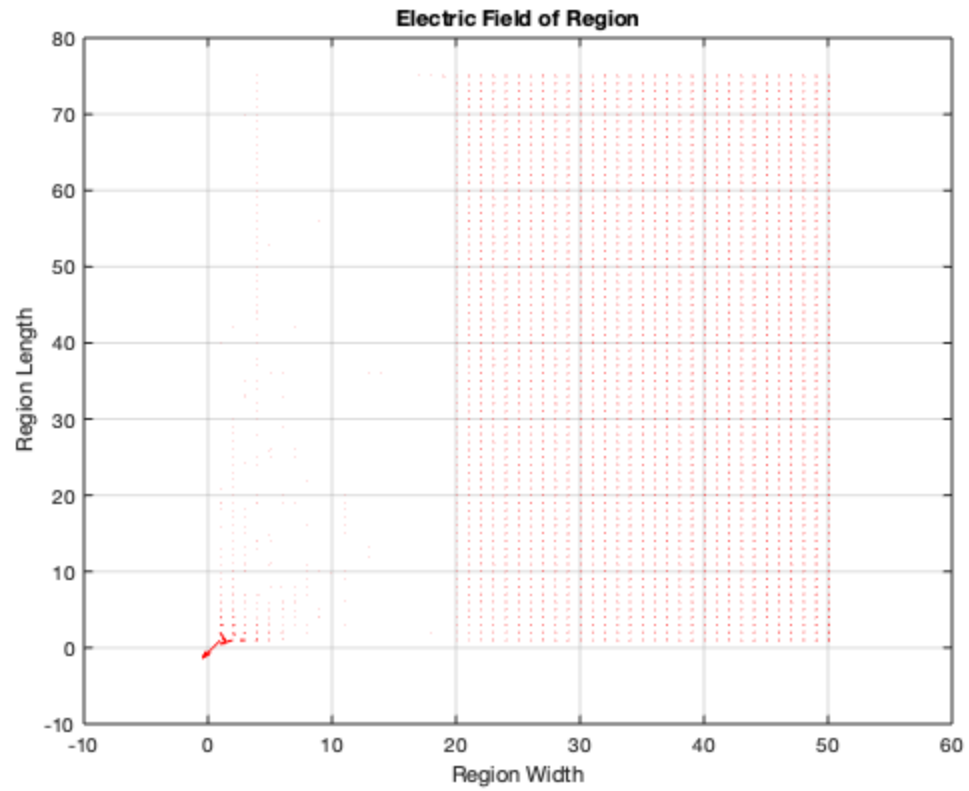
            elseif j == 1
                conxm = (con(i,j) + con(i-1,j))/2;
                conxp = (con(i,j) + con(i+1,j))/2;
                conyp = (con(i,j) + con(i,j+1))/2;
                sp(n,n) = -(conxm+conxp+conyp);
                sp(n,ax) = conxm;
                sp(n,bx) = conxp;
                sp(n,cx) = conyp;
                B(n) = 0;

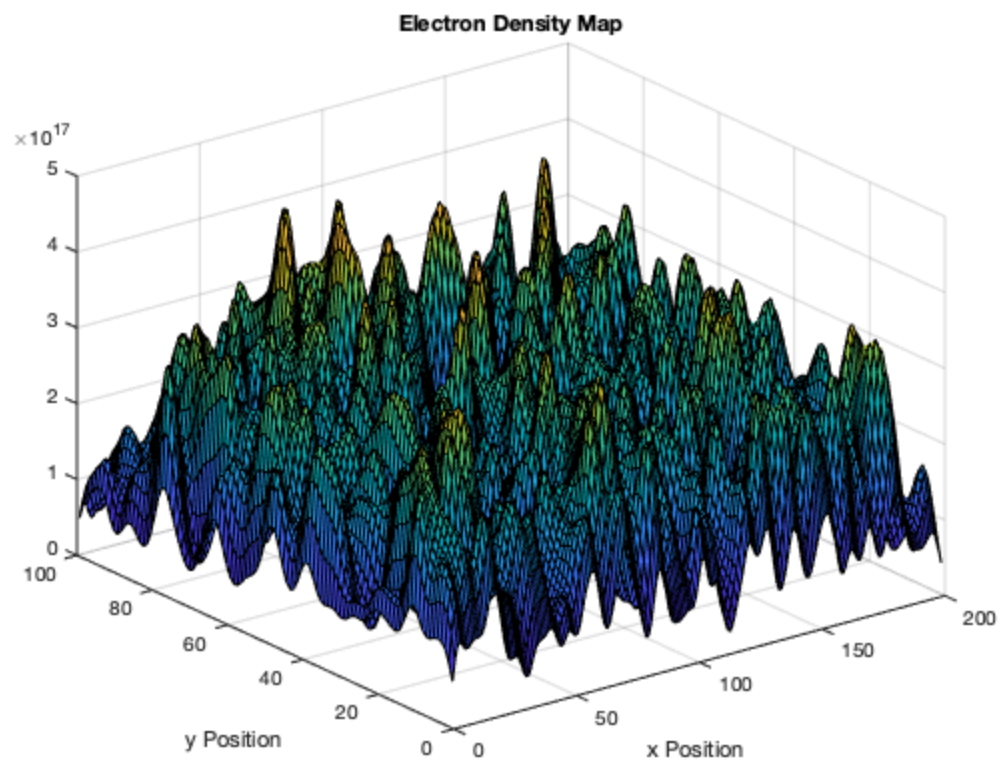
            else
                conxm = (con(i,j) + con(i-1,j))/2;
                conxp = (con(i,j) + con(i+1,j))/2;
                conyp = (con(i,j) + con(i,j+1))/2;
                conym = (con(i,j) + con(i,j+1))/2;
                sp(n,n) = -(conxm+conxp+conyp);
                sp(n,ax) = conxm;
                sp(n,bx) = conxp;
                sp(n,cx) = conyp;
            end
        end
    end
end
v=sp\B;
dz = zeros(regx,regy);
dx = linspace(1,regx,regx);

```

```
dy = linspace(1,regy,regy);

for i = 1:regx
    for j = 1:regy
        n = j+(i-1)*regy;
        dz(i,j) = v(n,1);
    end
end
end
end
```





Published with MATLAB® R2019b