Table of Contents

ELEC 4700 Assignment 3 Baldeep Kooner 101004107

Part 1

This section involves modifying the Monte-Carlo simulator from assignment 1, part 2, by adding a voltage across the x-axis of the semiconductor crystal. This voltage results in an electric field forming within the semiconductor. At first, the applied voltage was set to 0.1 V, but to clearly see the effects of the electric field, it was increased to 0.4 V.

```
clear all
close all

nElectrons = 1000;
nTime = 1000;
time = zeros(1, nTime);
m0 = 9.10938215e-31; %rest mass of an electron (kg)
mn = 0.26*m0; %effective mass of an electron (kg)
tau = 0.2e-12; %mean time between collisions (s)
Kb = 1.38064852e-23; %boltzmann constant (J/K)
T = 300; %temperature (K)
vth = sqrt(2*Kb*T/mn); %thermal velocity
dt = (100e-9)/vth/100;
l = vth*tau; %mean free path
temperature = zeros(1, nTime);
Pscat = 1 - exp(-dt/tau);
x = linspace(0, 200, 400)*10^(-9); %x axis
y = linspace(0, 100, 400)*10^(-9); %y axis

voltagex = 0.1;
voltagey = 0;
Ex = voltagex/(200e-9);
Ey = voltagey/(100e-9);
```

The electric field seen by the electrons is equal to:

```
E = sqrt(Ex^2 + Ey^2)
```

```
E =

   5.0000e+05
```

The force on each electron is equal to:

```
F = E * (1.60217653e-19)
```

```
F =

   8.0109e-14
```

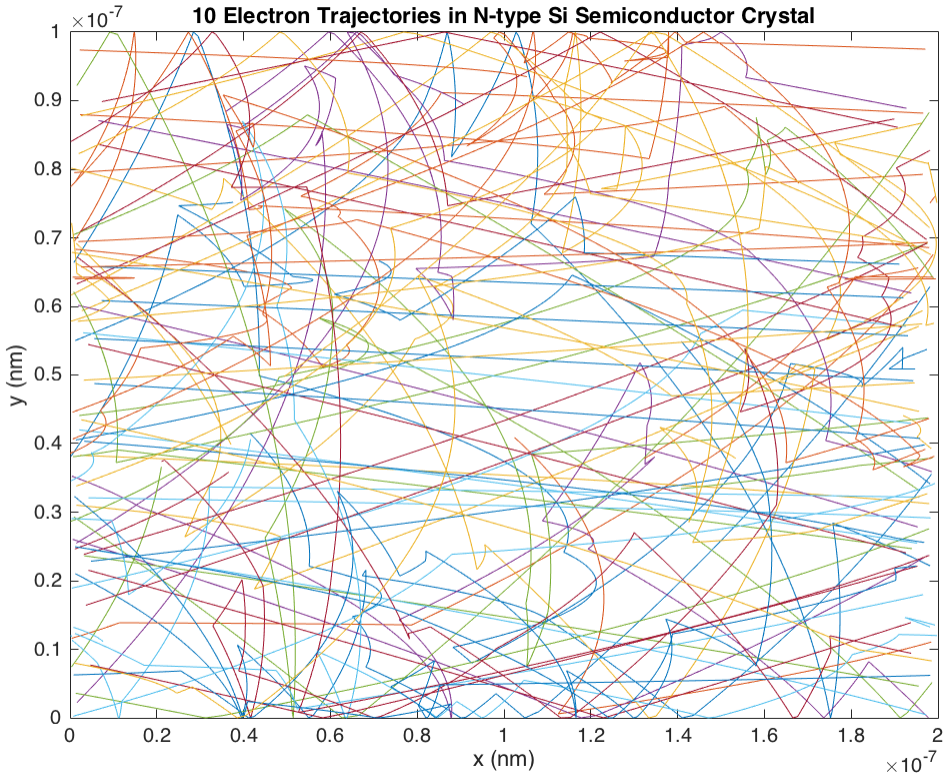
The acceleration on each electron is equal to:

```
a = F/mn


a =

   3.3823e+17
```

```
voltagex = 0.4;
voltagey = 0;
Ex = voltagex/(200e-9);
Ey = voltagey/(100e-9);

Px = zeros(nElectrons, nTime);
Py = zeros(nElectrons, nTime);

for n = 1 : nElectrons
    Px(n, 1) = x(randi(400));
    Py(n, 1) = y(randi(400));
end

Vx = zeros(nElectrons, nTime);
Vy = zeros(nElectrons, nTime);
accelx = zeros(nElectrons, nTime);
accely = zeros(nElectrons, nTime);

MaxwellBoltzmannVdist = makedist('Normal', 'mu', 0, 'sigma',
 sqrt(Kb*T/mn));

for k = 1 : nElectrons
    Vx(k, :) = random(MaxwellBoltzmannVdist);
    Vy(k, :) = random(MaxwellBoltzmannVdist);
    accelx(k, :) = Ex * (-1.60217653e-19/mn); %F = ma
    accely(k, :) = Ey * (-1.60217653e-19/mn);
end

avgV = sqrt(sum(Vx(:, 1).^2)/nElectrons + sum(Vy(:, 1).^2/
nElectrons));

for j = 1 : nElectrons
    for w = 2 : nTime
        Vx(j, w) = Vx(j, w-1) + accelx(j, w-1) * dt;
        Vy(j, w) = Vy(j, w-1) + accely(j, w-1) * dt;
        if isnan(Px(j, w-1))
            if left == 1
                Px(j, w) = 0 + Vx(j, w)*dt;
            end
            if right == 1
```

```
                Px(j, w) = 200e-9 + Vx(j, w)*dt;
            end
        else
            Px(j, w) = Px(j, w-1) + Vx(j, w)*dt;
        end

        if Px(j, w) > 200e-9
            left = 1;
            right = 0;
            Px(j, w) = NaN;
        end
        if Px(j, w) < 0
            left = 0;
            right = 1;
            Px(j, w) = NaN;
        end

        Py(j, w) = Py(j, w-1) + Vy(j, w)*dt;
        if Py(j, w) > 100e-9
            Py(j, w) = 100e-9;
            Vy(j, w:end) = -Vy(j, w);
        end
        if Py(j, w) < 0
            Py(j, w) = 0;
            Vy(j, w:end) = -Vy(j, w);
        end
        if Pscat > rand()
            Vx(j, w:end) = random(MaxwellBoltzmannVdist);
            Vy(j, w:end) = random(MaxwellBoltzmannVdist);
        end
    end
end

for i = 1:nTime
        temperature(i) = (sum(Vx(:, i).^2) + sum(Vy(:, i).^2))*mn/
Kb/2/nElectrons;
        if i > 1
            time(i) = time(i-1) + dt;
        end
end

n = 100e15;
for i = 1:nTime
        Jd(i) = sqrt(sum(Vx(:, i).^2)/nElectrons + sum(Vy(:, i).^2/
nElectrons))*(1.60217653e-19);
        if i > 1
            time(i) = time(i-1) + dt;
        end
end

for g = 1:10
    %subplot(3, 1, 1)
    figure(1)
    plot(Px(g, :), Py(g, :))
```

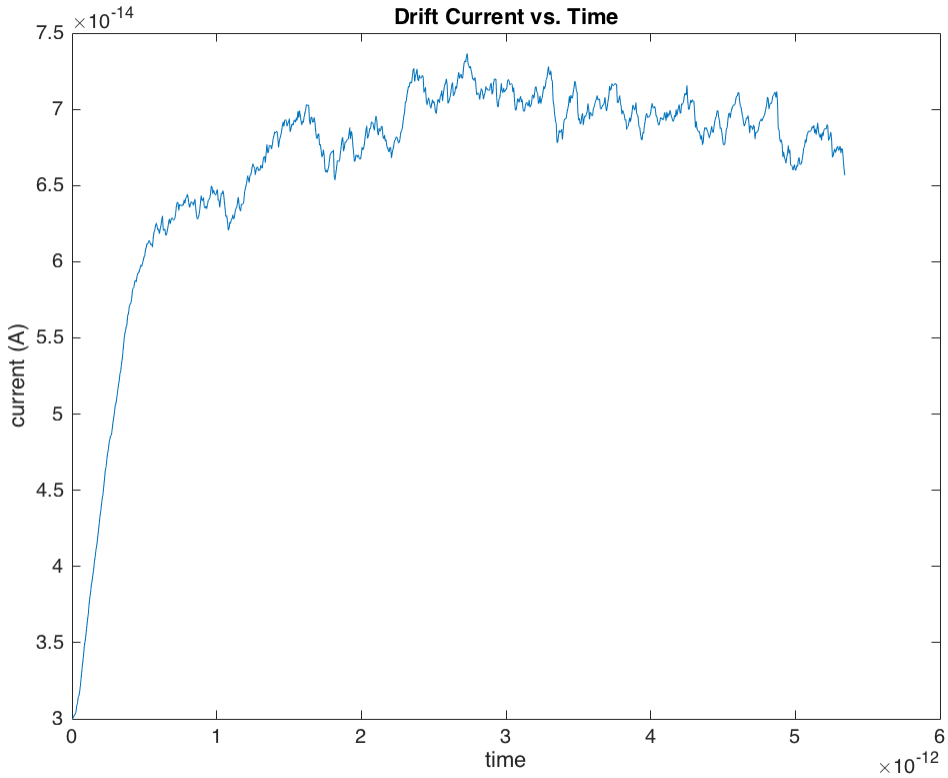
```
    xlabel('x (nm)')
    ylabel('y (nm)')
    title('10 Electron Trajectories in N-type Si Semiconductor
 Crystal')
    hold on
end
%{
subplot(3, 1, 2)
figure(2)
plot(time, temperature)
title('Temperature Variation Over 5ps Time Duration')
xlabel('time (s)')
ylabel('Temperature (K)')
xlim([0 5e-12])

%subplot(3, 1, 3)
figure(3)
vel = sqrt(Vx(:, 1).^2 + Vy(:, 1).^2);
histogram(vel)
title('Initial Electron Speeds')
xlabel('Speed (m/s)')
ylabel('# of electrons')
%}
```

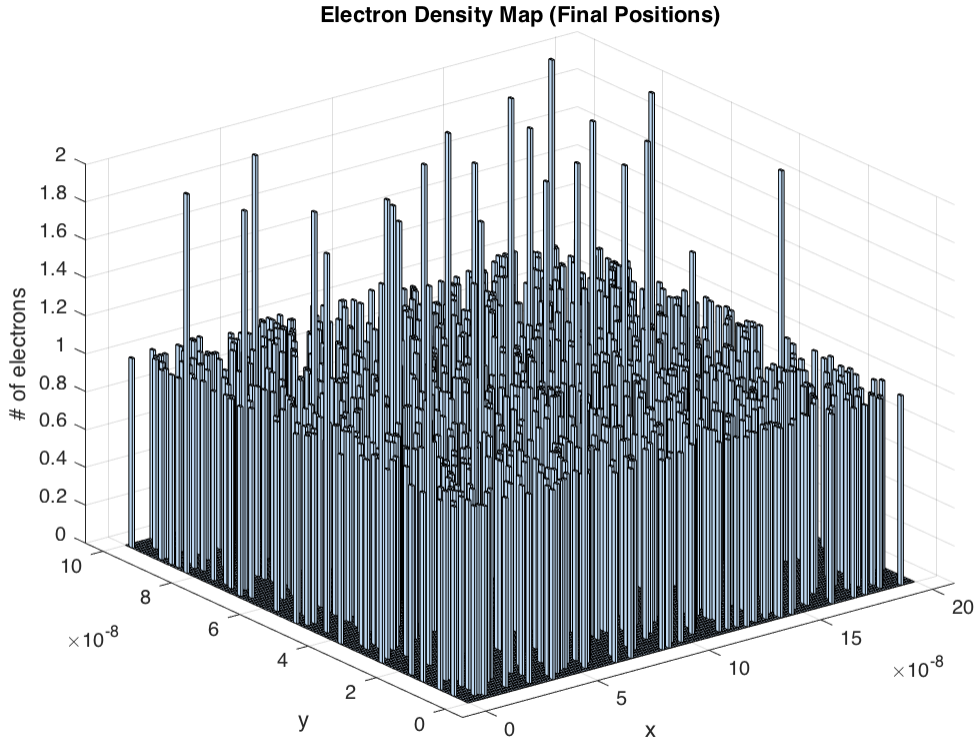


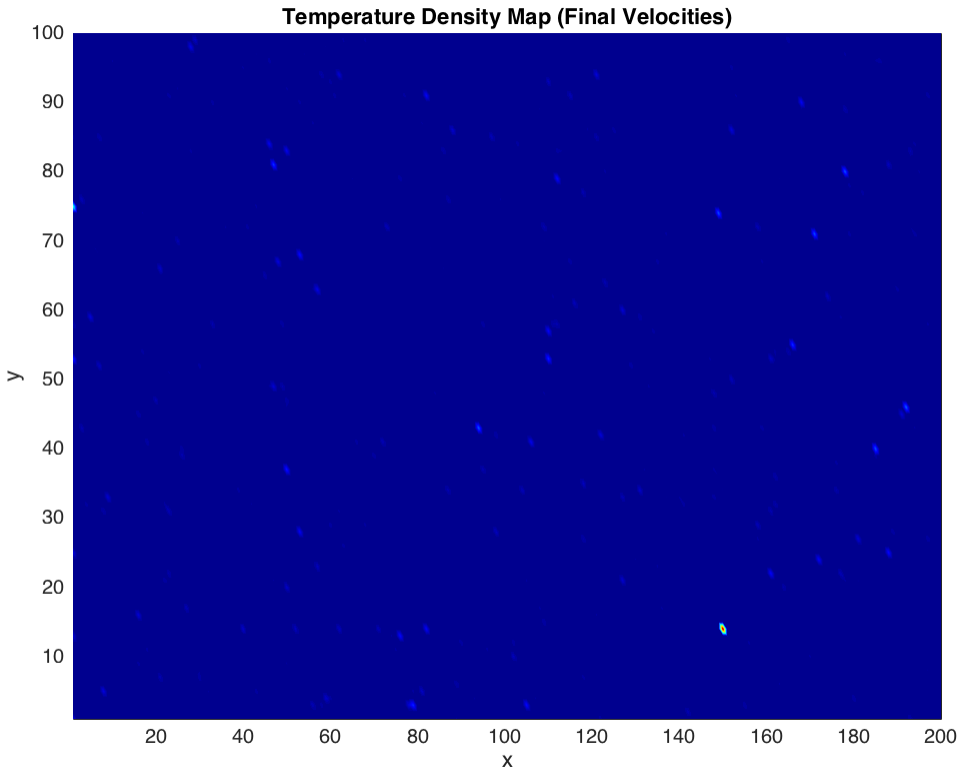
The electron drift current density is calculated by multiplying the charge of an electron, the number of electrons, the electron mobility in the material, and the electric field. The carrier velocity can be calculated

by multiplying the electron mobility and the electric field. Therefore, the electron drift current density can be simplified as the product of the carrier charge, number of carriers, and the average carrier velocity. A plot of the drift current over time is shown in the following figure and a noticeable property is that the current drastically increases at first and then levels out. This effect can be due to the initialization of the velocities. Considering the electric field begins to accelerate the electrons as time goes on, it is reasonable that the current follows an increasing trend.

```
figure(2)
plot(time, Jd)
title('Drift Current vs. Time')
xlabel('time')
ylabel('current (A)')

densityM = [Px(:, 1000), Py(:, 1000)];
figure(3)
hist3(densityM, [200 100])
title('Electron Density Map (Final Positions)')
xlabel('x')
ylabel('y')
zlabel('# of electrons')

tempx = zeros(ceil(200), ceil(100));
tempy = zeros(ceil(200), ceil(100));
tempn = zeros(ceil(200), ceil(100));

for z = 1:nElectrons
    x = floor(Px(z, 1000)/1e-9);
    y = floor(Py(z, 1000)/1e-9);
    if (x == 0 || isnan(x))
        x = 1;
    end
    if (y == 0 || isnan(y))
        y = 1;
    end
    tempy(x, y) = tempy(x, y) + Vy(z, 1000)^2;
    tempx(x, y) = tempx(x, y) + Vx(z, 1000)^2;
    tempn(x, y) = tempn(x, y) + 1;
end

temp2 = (tempx + tempy) .* mn ./ Kb ./ 2 ./ tempn;
temp2(isnan(temp2)) = 0;
temp2 = temp2';

figure(4)
xtemp = linspace(1, 200, 200);
ytemp = linspace(1, 100, 100);
pcolor(xtemp, ytemp, temp2)
shading interp
colormap(jet)
title('Temperature Density Map (Final Velocities)')
xlabel('x')
ylabel('y')
```


Drift Current vs. Time
×10^-14
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
current (A)
0
1
2
3
4
5
6
time
×10^-12


Electron Density Map (Final Positions)
# of electrons
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
10
8
6
4
2
0
×10^-8
y
0
5
10
15
20
×10^-8
x



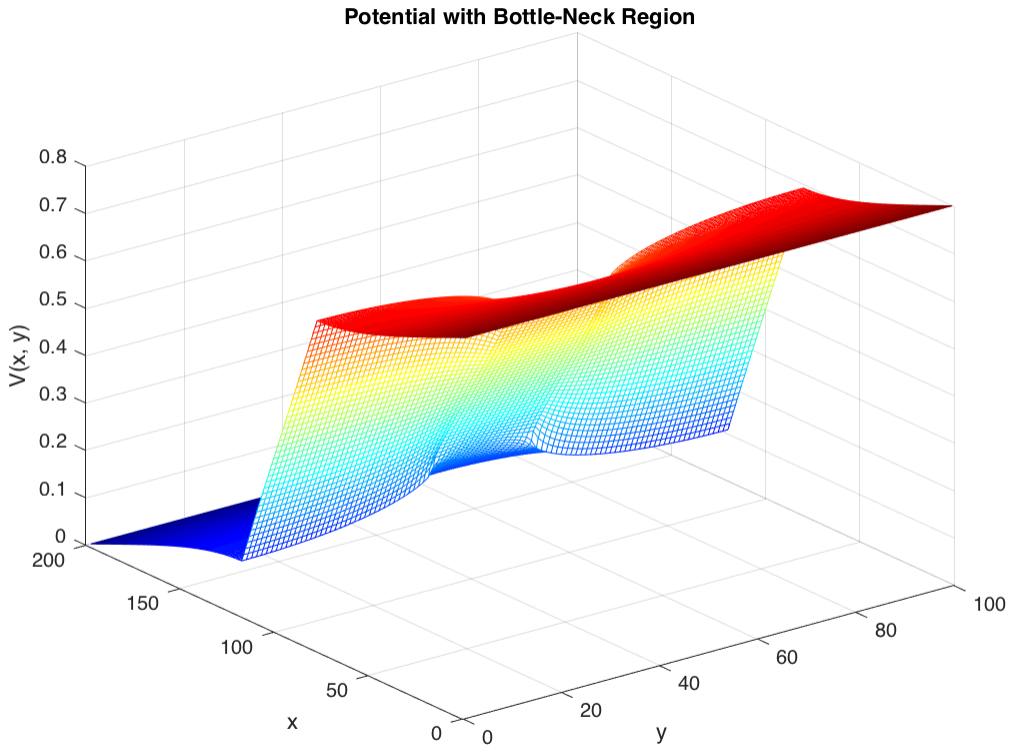
Part 2

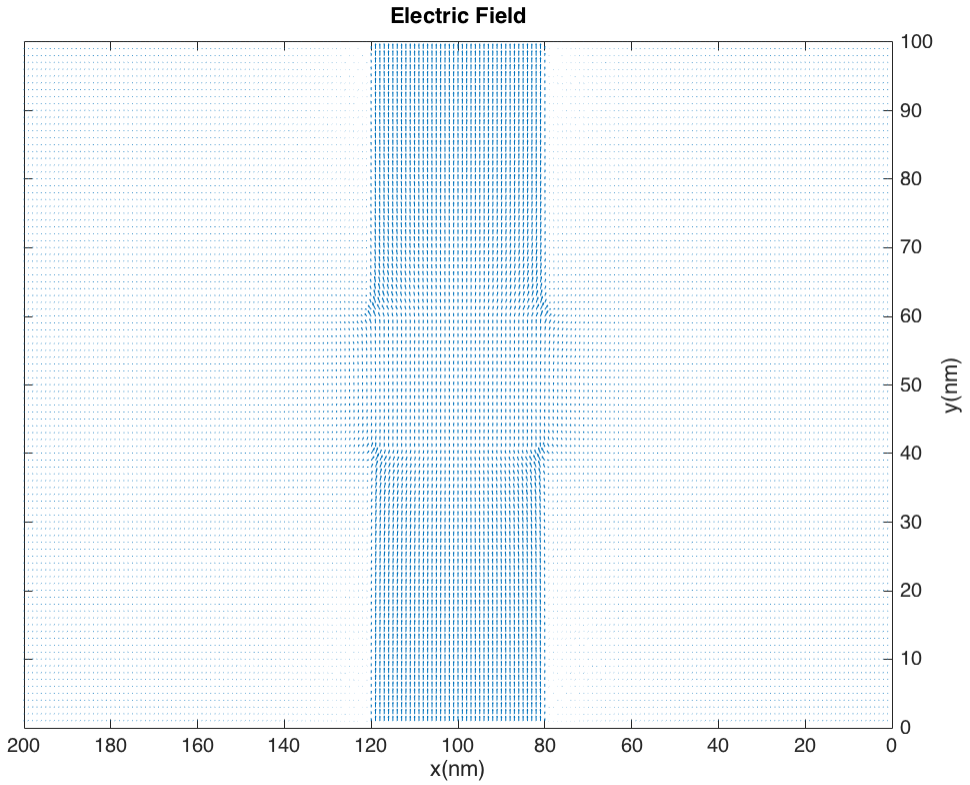
In this section, assigment 2 is modified to calculate and plot the potential (as defined in part 1) over the semiconductor crystal with a bottle neck barrier present. The potential across the x-axis was set to 0.8 V in this section to clearly demonstrate the effects. In addition, the electric field is calculated from the potential and plotted.

```
L = 200;
W = 100;
dx = 1;
dy = 1;
nx = L / dx;
ny = W / dy;

G = sparse(nx*ny, nx*ny);
F = zeros(1, nx*ny);

sigma = ones(nx, ny);
for i = 1:nx
    for j = 1:ny
        if j <= (40) || j >= (60)
            if i >= (80) && i <= (120)
                sigma(i, j) = 10^(-2);
            end
        end
    end
```

```
end

for i = 1:nx
    for j = 1:ny

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

        if i == 1
            G(n, :) = 0;
            G(n, n) = 1;
            F(n) = 0.8;
        elseif i == nx
            G(n, n) = 1;
            %F(n) = 1;
        elseif j == 1
            sigmaUPPER = (sigma(i, j) + sigma(i, j+1)) / 2.0;
            sigmaRIGHT = (sigma(i, j) + sigma(i+1, j)) / 2.0;
            sigmaLEFT = (sigma(i, j) + sigma(i-1, j)) / 2.0;
            %sigmaLOWER = (sigma(i, j) + sigma(i, j-1)) / 2.0;

            G(n, n) = -sigmaUPPER - sigmaRIGHT - sigmaLEFT;
            G(n, n + 1) = sigmaUPPER;
            G(n, n + ny) = sigmaRIGHT;
            G(n, n - ny) = sigmaLEFT;
        elseif j == ny
            %sigmaUPPER = (sigma(i, j) + sigma(i, j+1)) / 2.0;
            sigmaRIGHT = (sigma(i, j) + sigma(i+1, j)) / 2.0;
            sigmaLEFT = (sigma(i, j) + sigma(i-1, j)) / 2.0;
            sigmaLOWER = (sigma(i, j) + sigma(i, j-1)) / 2.0;

            G(n, n) = -sigmaUPPER - sigmaRIGHT - sigmaLEFT;
            G(n, n - 1) = sigmaLOWER;
            G(n, n + ny)= sigmaRIGHT;
            G(n, n - ny)= sigmaLEFT;
        else
            sigmaUPPER = (sigma(i, j) + sigma(i, j+1)) / 2.0;
            sigmaRIGHT = (sigma(i, j) + sigma(i+1, j)) / 2.0;
            sigmaLEFT = (sigma(i, j) + sigma(i-1, j)) / 2.0;
            sigmaLOWER = (sigma(i, j) + sigma(i, j-1)) / 2.0;

            G(n, n)= -sigmaUPPER - sigmaLOWER - sigmaRIGHT -
 sigmaLEFT;
            G(n, n + 1) = sigmaUPPER;
            G(n, n - 1) = sigmaLOWER;
            G(n, n + ny) = sigmaRIGHT;
            G(n, n - ny) = sigmaLEFT;
        end
    end
end

V2 = zeros(nx, ny);
V = G\F';

for i = 1:nx
```

```
    for j = 1:ny
        n = j + (i - 1)*ny;
        V2(i, j) = V(n);
    end
end

for i = 1:nx
    for j = 1:ny
        if sigma(i, j) == power(10, -12)
            V2(i, j) = sigma(i, j);
        end
    end
end

figure(5)
mesh(V2)
colormap(jet)
title('Potential with Bottle-Neck Region')
xlabel('y')
ylabel('x')
zlabel('V(x, y)')

[Ey, Ex] = gradient(V2*1e9);
Ex = -Ex;
Ey = -Ey;

figure(6)
quiver(Ex, Ey)
camroll(90)
xlim([0 100])
ylim([0 200])
title('Electric Field')
xlabel('y(nm)')
ylabel('x(nm)')
```


Potential with Bottle-Neck Region
V(x, y)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
200
150
100
50
0
x
0
20
40
60
80
100
y


Electric Field
100
90
80
70
60
50
40
30
20
10
0
y(nm)
200
180
160
140
120
100
80
60
40
20
0
x(nm)

Part 3

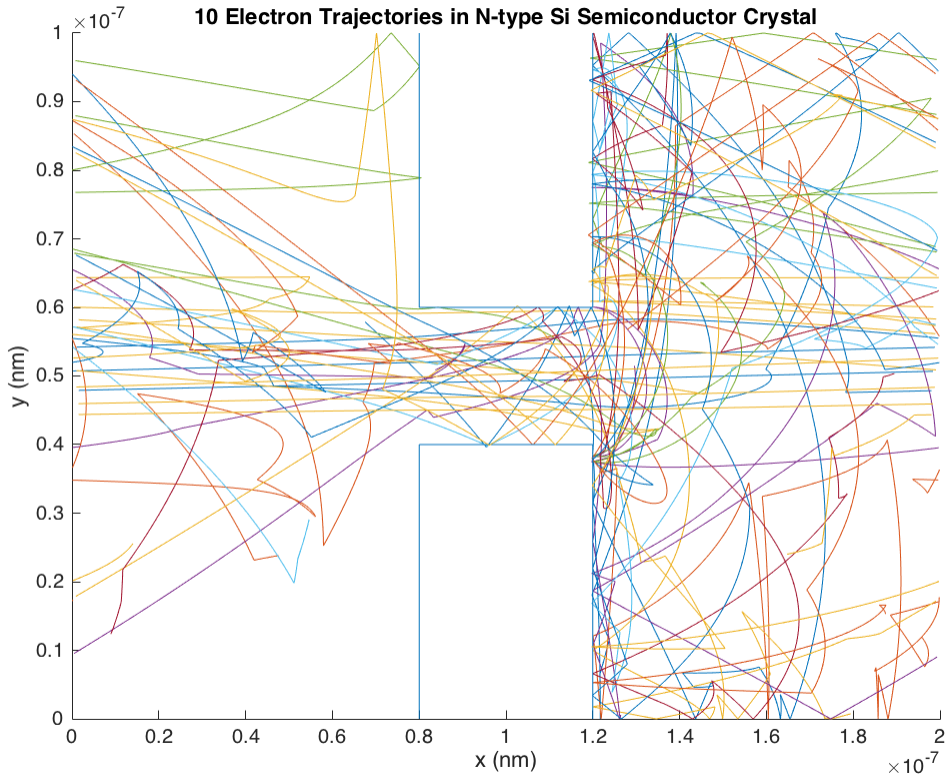
In this section, both parts 1 and 2 are coupled. The code from part 2 is directly used to establish a potential of 0.8 V across the x-axis of the semiconductor and also form the resulting electric field. The code from part 1 is modified to include the bottle neck barrier and to determine the electrons' acceleration due to the electric field at each time step.

```
nElectrons = 1000;
nTime = 3500;
time = zeros(1, nTime);
m0 = 9.10938215e-31; %rest mass of an electron (kg)
mn = 0.26*m0; %effective mass of an electron (kg)
tau = 0.2e-12; %mean time between collisions (s)
Kb = 1.38064852e-23; %boltzmann constant (J/K)
T = 300; %temperature (K)
vth = sqrt(2*Kb*T/mn); %thermal velocity
dt = (100e-9)/vth/500;
l = vth*tau; %mean free path
temperature = zeros(1, nTime);
Pscat = 1 - exp(-dt/tau);
x = linspace(0, 200, 400)*10^(-9); %x axis
y = linspace(0, 100, 400)*10^(-9); %y axis

Px = zeros(nElectrons, nTime);
Py = zeros(nElectrons, nTime);

for n = 1 : nElectrons
    Px(n, 1) = x(randi(400));
    Py(n, 1) = y(randi(400));
    while (Px(n, 1) >= 80e-9 && Px(n, 1) <= 120e-9 && Py(n, 1) >=
 60e-9) || (Px(n, 1) >= 80e-9 && Px(n, 1) <= 120e-9 && Py(n, 1) <=
 40e-9)
        Px(n, 1) = x(randi(400));
        Py(n, 1) = y(randi(400));
    end
end

Vx = zeros(nElectrons, nTime);
Vy = zeros(nElectrons, nTime);
accelx = zeros(nElectrons, nTime);
accely = zeros(nElectrons, nTime);

MaxwellBoltzmannVdist = makedist('Normal', 'mu', 0, 'sigma',
 sqrt(Kb*T/mn));

for k = 1 : nElectrons
    Vx(k, :) = random(MaxwellBoltzmannVdist);
    Vy(k, :) = random(MaxwellBoltzmannVdist);
    if round(Px(k, 1)*(10^9)) == 0 && round(Py(k, 1)*(10^9)) == 0
        accelx(k, 1) = Ex(1, 1) * (-1.60217653e-19/mn);
        accely(k, 1) = Ey(1, 1) * (-1.60217653e-19/mn);
    elseif round(Px(k, 1)*(10^9)) == 0
```

```
        accelx(k, 1) = Ex(1, round(Py(k, 1)*(10^9))) *
 (-1.60217653e-19/mn);
        accely(k, 1) = Ey(1, round(Py(k, 1)*(10^9))) *
 (-1.60217653e-19/mn);
    elseif round(Py(k, 1)*(10^9)) == 0
        accelx(k, 1) = Ex(round(Px(k, 1)*(10^9)), 1) *
 (-1.60217653e-19/mn);
        accely(k, 1) = Ey(round(Px(k, 1)*(10^9)), 1) *
 (-1.60217653e-19/mn);
    else
        accelx(k, 1) = Ex(round(Px(k, 1)*(10^9)), round(Py(k,
 1)*(10^9))) * (-1.60217653e-19/mn);
        accely(k, 1) = Ey(round(Px(k, 1)*(10^9)), round(Py(k,
 1)*(10^9))) * (-1.60217653e-19/mn);
    end
end

avgV = sqrt(sum(Vx(:, 1).^2)/nElectrons + sum(Vy(:, 1).^2/
nElectrons));

for j = 1 : nElectrons
    for w = 2 : nTime
        Vx(j, w) = Vx(j, w-1) + accelx(j, w-1) * dt;
        Vy(j, w) = Vy(j, w-1) + accely(j, w-1) * dt;
        if isnan(Px(j, w-1))
            if left == 1
                if Vx(j, w) < 0
                    Vx(j, w:end) = -Vx(j, w);
                end
                Px(j, w) = 0 + Vx(j, w)*dt;
            end
            if right == 1
                if Vx(j, w) > 0
                    Vx(j, w:end) = -Vx(j, w);
                end
                Px(j, w) = 200e-9 + Vx(j, w)*dt;
            end
        else
            Px(j, w) = Px(j, w-1) + Vx(j, w)*dt;
        end

        if Px(j, w) > 200e-9
            left = 1;
            right = 0;
            Px(j, w) = NaN;
        end
        if Px(j, w) < 0
            left = 0;
            right = 1;
            Px(j, w) = NaN;
        end

        Py(j, w) = Py(j, w-1) + Vy(j, w)*dt;
        if Py(j, w) > 100e-9
```

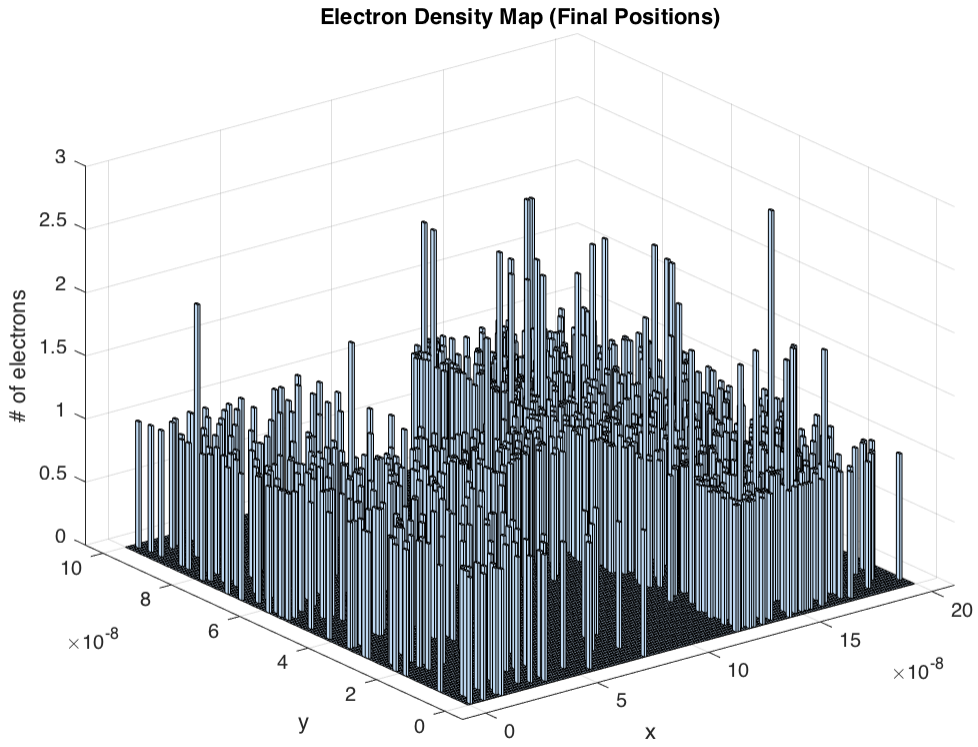
```
            Py(j, w) = 100e-9;
            Vy(j, w:end) = -Vy(j, w);
        end
        if Py(j, w) < 0
            Py(j, w) = 0;
            Vy(j, w:end) = -Vy(j, w);
        end

        if (Px(j, w) >= 80e-9 && Px(j, w) <= 120e-9 && Py(j, w) >=
60e-9) || (Px(j, w) >= 80e-9 && Px(j, w) <= 120e-9 && Py(j, w) <=
40e-9)
            if (Px(j, w-1) <= 80e-9 && Py(j, w-1) <= 40e-9) || (Px(j,
w-1) <= 80e-9 && Py(j, w-1) >= 60e-9) || (Px(j, w-1) >= 120e-9 &&
Py(j, w-1) <= 40e-9) || (Px(j, w-1) >= 120e-9 && Py(j, w-1) >= 40e-9)
                Vx(j, w:end) = -Vx(j, w-1);
            end
            if Px(j, w-1) >= 80e-9 && Px(j, w-1) <= 120e-9 && Py(j,
w-1) <= 60e-9 && Py(j, w-1) >= 40e-9
                Vy(j, w:end) = -Vy(j, w-1);
            end
        end

        if Pscat > rand()
            Vx(j, w:end) = random(MaxwellBoltzmannVdist);
            Vy(j, w:end) = random(MaxwellBoltzmannVdist);
        end

        if isnan(Px(j, w))
            if round(Px(j, w-1)*(10^9)) == 0 && round(Py(j,
w-1)*(10^9)) == 0
                accelx(j, w) = Ex(1, 1) * (-1.60217653e-19/mn);
                accely(j, w) = Ey(1, 1) * (-1.60217653e-19/mn);
            elseif round(Px(j, w-1)*(10^9)) == 0
                accelx(j, w) = Ex(1, round(Py(j, w-1)*(10^9))) *
(-1.60217653e-19/mn);
                accely(j, w) = Ey(1, round(Py(j, w-1)*(10^9))) *
(-1.60217653e-19/mn);
            elseif round(Py(j, w-1)*(10^9)) == 0
                accelx(j, w) = Ex(round(Px(j, w-1)*(10^9)), 1) *
(-1.60217653e-19/mn);
                accely(j, w) = Ey(round(Px(j, w-1)*(10^9)), 1) *
(-1.60217653e-19/mn);
            else
                accelx(j, w) = Ex(round(Px(j, w-1)*(10^9)),
round(Py(j, w-1)*(10^9))) * (-1.60217653e-19/mn);
                accely(j, w) = Ey(round(Px(j, w-1)*(10^9)),
round(Py(j, w-1)*(10^9))) * (-1.60217653e-19/mn);
            end
        else
            if round(Px(j, w)*(10^9)) == 0 && round(Py(j, w)*(10^9))
== 0
                accelx(j, w) = Ex(1, 1) * (-1.60217653e-19/mn);
                accely(j, w) = Ey(1, 1) * (-1.60217653e-19/mn);
            elseif round(Px(j, w)*(10^9)) == 0
```

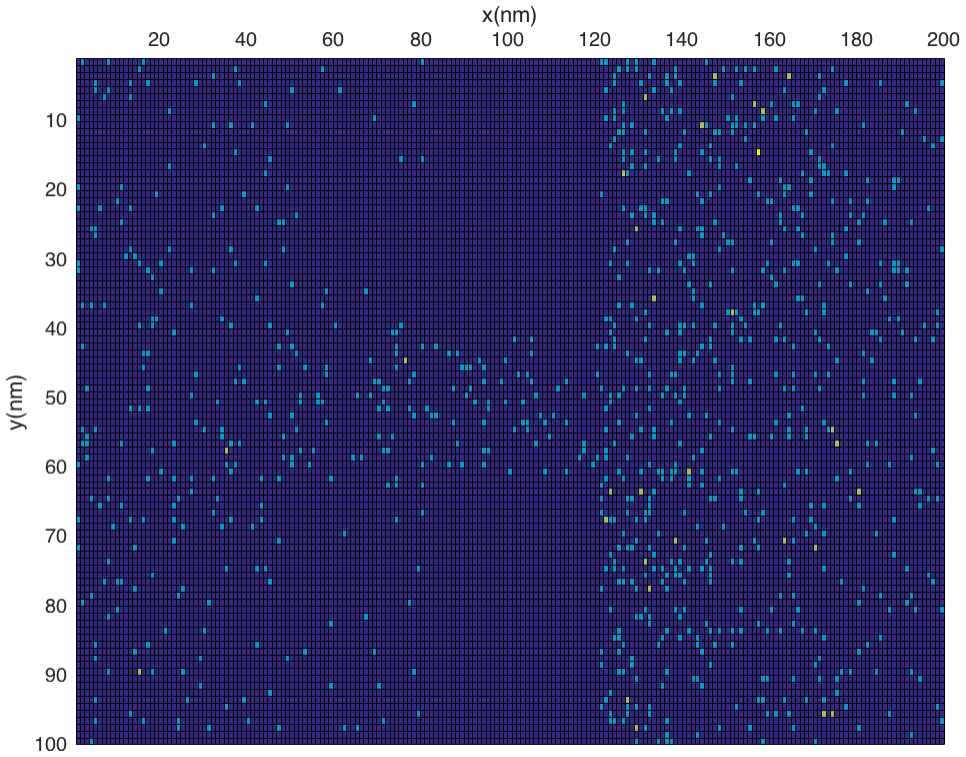
```
                accelx(j, w) = Ex(1, round(Py(j, w)*(10^9))) *
 (-1.60217653e-19/mn);
                accely(j, w) = Ey(1, round(Py(j, w)*(10^9))) *
 (-1.60217653e-19/mn);
            elseif round(Py(j, w)*(10^9)) == 0
                accelx(j, w) = Ex(round(Px(j, w)*(10^9)), 1) *
 (-1.60217653e-19/mn);
                accely(j, w) = Ey(round(Px(j, w)*(10^9)), 1) *
 (-1.60217653e-19/mn);
            else
                accelx(j, w) = Ex(round(Px(j, w)*(10^9)), round(Py(j,
 w)*(10^9))) * (-1.60217653e-19/mn);
                accely(j, w) = Ey(round(Px(j, w)*(10^9)), round(Py(j,
 w)*(10^9))) * (-1.60217653e-19/mn);
            end
        end
    end
end


for g = 1:10
    %subplot(3, 1, 1)
    figure(7)
    xLim = [80e-9 80e-9 120e-9 120e-9];
    yLim1 = [0 40e-9 40e-9 0];
    yLim2 = [100e-9 60e-9 60e-9 100e-9];
    line(xLim, yLim1)
    hold on
    line(xLim, yLim2)
    plot(Px(g, :), Py(g, :))
    xlabel('x (nm)')
    ylabel('y (nm)')
    title('10 Electron Trajectories in N-type Si Semiconductor
 Crystal')
    hold off
end
```



Considering there is an electric field present in the semiconductor, the expected behaviour of the electrons is that they will tend to travel towards the right side of the crystal. This effect can be seen in the following plots showing a 3D density plot and a 2D density map:

```
densityM = [Px(:, 1000), Py(:, 1000)];
figure(8)
hist3(densityM, [200 100])
title('Electron Density Map (Final Positions)')
xlabel('x')
ylabel('y')
zlabel('# of electrons')

figure(9)
pcolor(hist3(densityM, [200 100]))
camroll(-90)
ylabel('x(nm)')
xlabel('y(nm)')
```


Electron Density Map (Final Positions)
# of electrons
3
2.5
2
1.5
1
0.5
0
10
8
6
4
2
0
×10^-8
y
0
5
10
15
20
×10^-8
x


x(nm)
20
40
60
80
100
120
140
160
180
200
y(nm)
10
20
30
40
50
60
70
80
90
100

Published with MATLAB® R2015b