Assignment 3

Part 1

In assignment 3, we are now implementing the result of assignment 2 into the electron simulation with bottle-neck of assignment, which is applying a voltage across the whole region with the bottle-neck of the simulation.

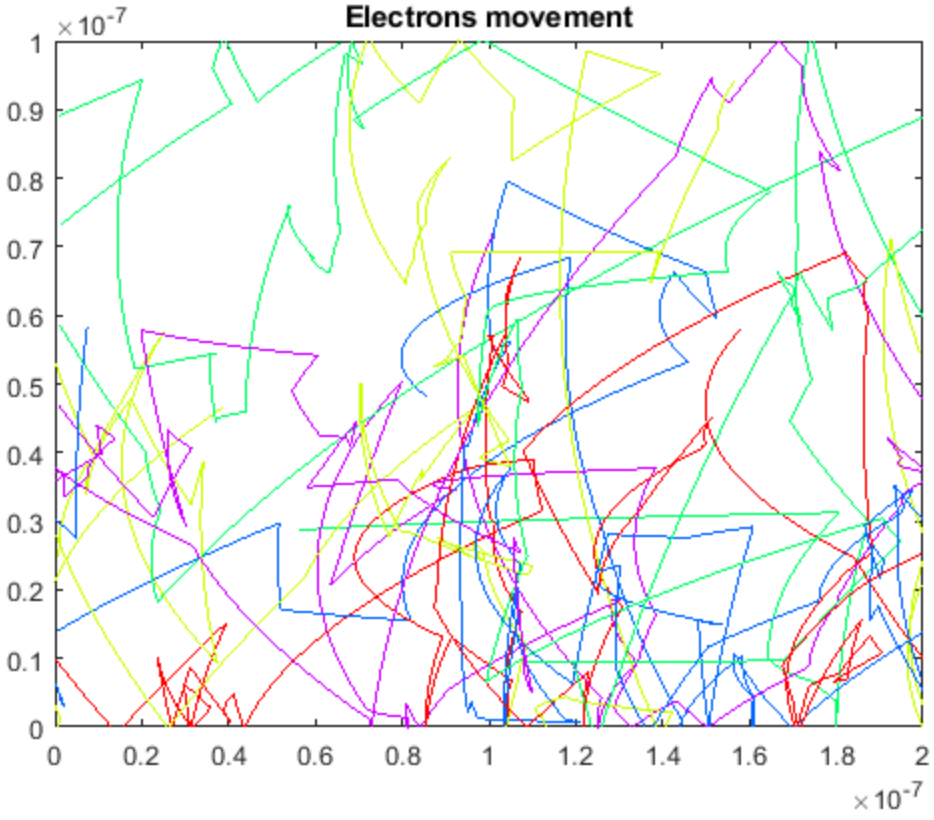
In the first part, we simply have to apply a constant voltage in the x-direction across the whole region to familarize ourselves for the later full implementation.

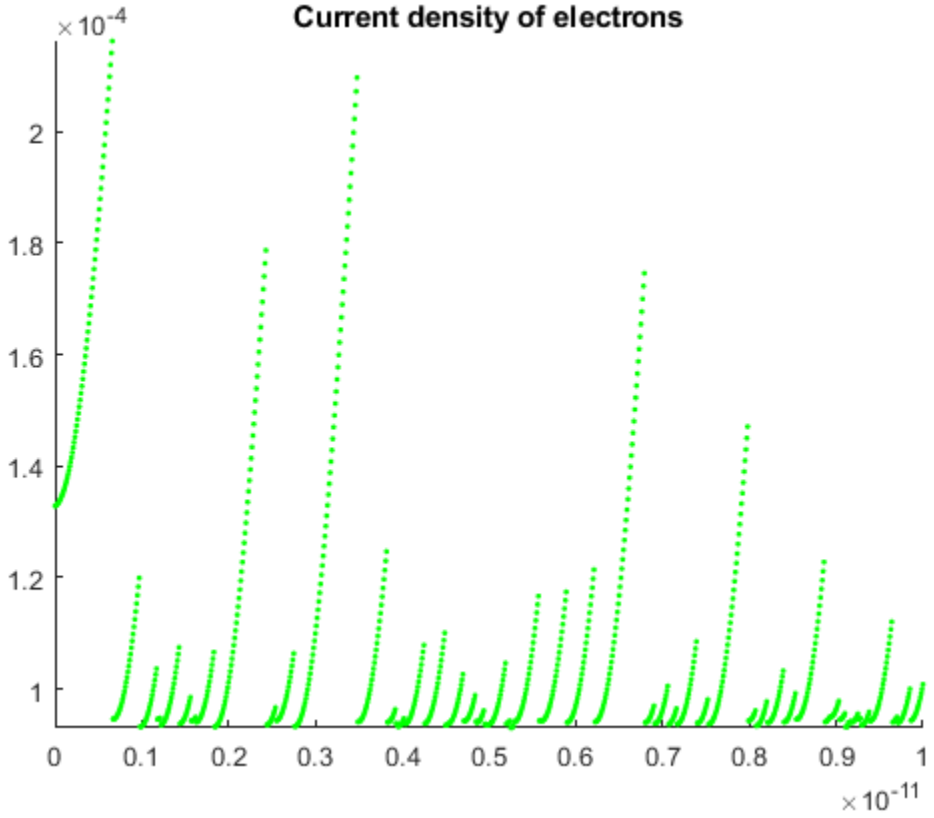
```
% Reset Everything
close all
clear

% Constant
q_0 = 1.60217653e-19;                   % electron charge
m_0 = 9.10938215e-31;                   % electron mass
meff = 0.26*m_0;                        % electron effective mass
kb = 1.3806504e-23;                     % Boltzmann constant
tmn = 0.2e-12;                          % mean time between collisions

% Region Defining
L = 200e-9;
W = 100e-9;

% Current Condition and variables
num = 1e4;                              % Number of electrons
T = 300;                                % Temperature (Kelvin)
V = 0.1;                                % Voltage applied
vth_e = sqrt((kb*T)/(meff));            % Thermal velocity of an
 electron
vth_ex = (vth_e)*randn(num, 1);         % X-component of thermal
 velocity
vth_ey = (vth_e)*randn(num, 1);         % Y-component of thermal
 velocity
vthdis = sqrt(vth_ex.^2+vth_ey.^2);     % Distribution of electrons
 thermal velocity
vthav = mean(sqrt(vth_ex.^2+vth_ey.^2));% Average of thermal velocity
MFP = vthav*tmn;                        % Mean free path of electrons
Ex = V/L;                               % Electric field on x-axis
F = Ex*q_0;                             % Force applied to electrons
accel = F/meff;                         % Acceleration of electrons

fprintf('Ex = %i\n', Ex);
fprintf('F = %i\n', F);
fprintf('Acceleration = %i\n', accel);

% Electrons Defining
Elec = zeros(num, 4);
Elec(:, 1) = L*rand(num, 1);
Elec(:, 2) = W*rand(num, 1);
```

```
Elec(:, 3) = vth_ex;
Elec(:, 4) = vth_ey;
previous = Elec;

% Electron simulation
t = 1e-11;                          % Total Time
dt = 1e-14;                         % Time Step
Psat = 1 - exp(-dt/tmn);            % Exponential Scattering
 Probability
numplot = 5;                        % Number of electron plotted
color = hsv(numplot);               % Colour Setup

% Creating figure for later assigning
f1 = figure;
f2 = figure;

for n = 0:dt:t

    % Applying acceleration into the velocity of electrons
    Elec(:, 3) = Elec(:, 3)+ accel*dt;

    % Electrons scattering
    if Psat > rand()
        vth_ex = (vth_e/sqrt(2))*randn(num, 1);
        vth_ey = (vth_e/sqrt(2))*randn(num, 1);
        Elec(:, 3) = vth_ex;
        Elec(:, 4) = vth_ey;
    end

    % Moving electrons
    for p = 1:1:num
        previous(p, 1) = Elec(p, 1);
        previous(p, 2) = Elec(p, 2);
        Elec(p, 1) = Elec(p, 1)+ Elec(p, 3)*dt;
        Elec(p, 2) = Elec(p, 2)+ Elec(p, 4)*dt;
    end

    % Plotting limited amount of electrons
    set(0, 'CurrentFigure', f1)
    for q = 1:1:numplot
        title('Electrons movement');
        plot([previous(q, 1), Elec(q, 1)], [previous(q, 2),
 Elec(q,2)],...
            'color', color(q, :))
        xlim([0 L])
        ylim([0 W])
        hold on
    end

    % Setting up the boundaries
    for o = 1:1:num
        % Looping on x-axis
        if Elec(o, 1) > L
            Elec(o, 1) = Elec(o, 1) - L;
```

```
        end
        if Elec(o, 1) < 0
            Elec(o, 1) = Elec(o, 1) + L;
        end
        % Reflecting on y-axis
        if Elec(o, 2) > W || Elec(o, 2) < 0
            Elec(o, 4) = -1*Elec(o, 4);
        end
    end

    % Plotting Current density
    set(0, 'CurrentFigure', f2)
    vaver = mean(sqrt(Elec(:, 3).^2 + Elec(:, 4).^2)); % Average
 thermal velocity
    I = vaver*num*Ex*q_0;                       % Drift current of
 electron
    scatter(n, I, 'g.')
    axis tight
    title('Current density of electrons');
    hold on

%     pause(0.01)
end
```

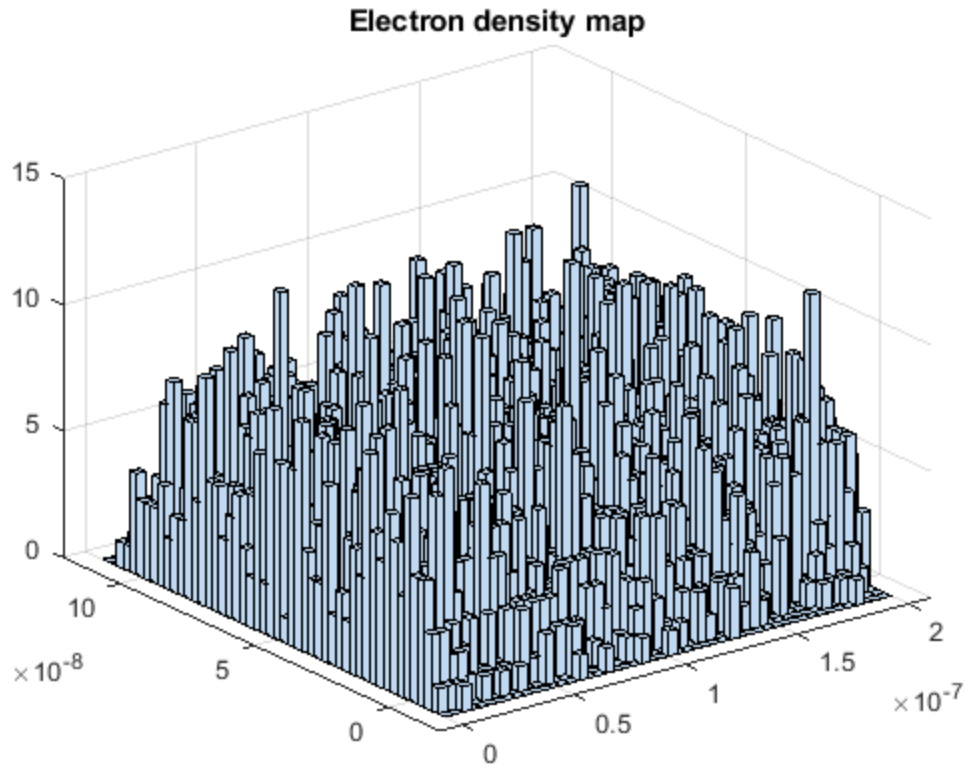
```
Ex = 5.000000e+05
F = 8.010883e-14
Acceleration = 338234653108549440
```

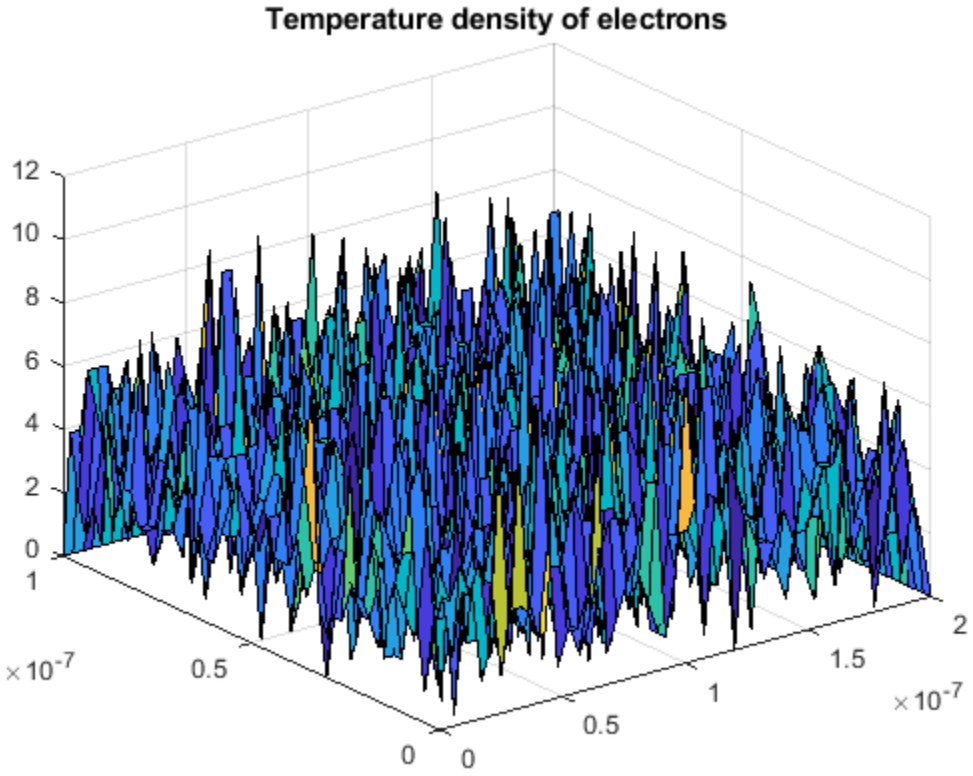

Electrons movement
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10^-7


Current density of electrons
×10^-4
2
1.8
1.6
1.4
1.2
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
×10^-11

The current density of the electrons in the simulation will slowly converge into a value which show that the current density will start to stablize after a period of time.

```
% Electron Density map
figure(4)
hist3(Elec(:, 1:2), [50 50]);
title("Electron density map")

% Temperature map
figure(5)
[binx, biny] = meshgrid(0:L/50:L, 0:W/50:W);% Setting the bins
zcheck = zeros(51, 51);                     % Initialize result matrix
tempcheck = zeros(51, 51);                  % Initialize temperature
 matrix
counter = 0;                                % Initialize counter
vtotal = 0;                                 % Initialize total
 velocity

% Mapping the temperature of electrons within each bin
for i = 1:50
    txmn = binx(1,i);
    txmx = binx(1, i+1);
    for r = 1:50
        tymn = biny(r, 1);
        tymx = biny(r+1, 1);
        for mm = 1:num
            if(Elec(mm,1)>txmn & Elec(mm,1)<txmx & Elec(mm,2)<tymx &
 Elec(mm,2)>tymn)
                counter = counter + 1;
                zcheck(i, r) = zcheck(i, r)+1;
                vtotal = vtotal + sqrt(Elec(mm, 3)^2+Elec(mm, 4)^2);
                if(counter ~= 0)
                    tempcheck(i,r) = meff*(vtotal^2)/(counter*kb);
                end
            end
        end
        vtotal = 0;
        counter = 0;
    end
end

% Surface plot of the temperature density map
surf(binx, biny,zcheck)
title("Temperature density of electrons")
```


Electron density map
15
10
5
0
10
5
0
×10^-8
0
0.5
1
1.5
2
×10^-7


Temperature density of electrons
12
10
8
6
4
2
0
1
0.5
0
×10^-7
0
0.5
1
1.5
2
×10^-7

Published with MATLAB® R2017b