

---

# ELEC 4700 Assignment 3

## Table of Contents

Q1 - Modify Assignment 1 Monte-Carlo to include Electric feild .....	1
Addition of constant electric field of 0.1V .....	2
Q1 Main Loop .....	2
Function for Electric Field (No change from Assignment 2) .....	6
Q2 - Add electric field for Lab 2 bottle neck program .....	7
Q2 Main Loop .....	10
%% Q3 - Add electric field for Lab 2 bottle neck with density map .....	12
Q3 Main Loop .....	15

Monte\_Carlo + Finite Difference Method - Jinseng Vanderkloot 101031534 - Due: March 20, 2022

## Q1 - Modify Assignment 1 Monte-Carlo to include Electric feild

```
clc;
clear all;
close all;

m0 = 9.10938215e-31;           % electron mass
mn = 0.26*m0;                  % Effective mass
Temp = 300;                    % Inital Temp (K)
kb = 1.3806504e-23;            % Boltzmann constant
tmn = 0.2e-12;                 % Mean time between collision
e = 1.602e-19;                 % Charge of an Electron (Assignment 3 for
    force)

%Thermal Velocity (Question 1.A)
vt=sqrt((2*kb*Temp)/mn);       % Sim in 2D so (2*kb*Temp), 3D is (3*kb*Temp)

% Plotting Area
wArea = 200e-9;
lArea = 100e-9;

numElec = 30000;               %Number of simulated Electrons
numEPlot = 20;                 %Number of plotted Electrons
dt = (lArea*wArea)/2;          %Typically 1/100 of region size
stepsTot = 150;                %Total amount of steps (1000 was a long
    simulation)
tTot= stepsTot*dt;             %Total Simulation time
x = zeros(1,numElec);          %Inital X matrix
y = zeros(1,numElec);          %Inital y matrix
vx = zeros(1,numElec);         %Inital velocity x matrix
vy = zeros(1,numElec);         %Inital velocity y matrix
vtot = zeros(1,numElec);       %Inital velocity matrix
VEnew = zeros(1,numElec);      %add x vector with field force velocity
colors = rand(numElec,3);       %Color assignment for each electron
```

```
%Probability of Scatter
scatOn = 1; %Turn Scatter on (1) or off(0)
Pscatter = 1-exp(-dt/tmn); %Scatter Equation
tScatter = zeros(1,numElec); %track scatter for each particle

%S1 Electron Random Assignments
for cnt = 1:numElec
    x(cnt)=rand()*wArea;
    y(cnt)=rand()*lArea;
    angle = (2*pi*rand());
    vx(cnt)=vt*cos(angle); % velocity * random direction
    vy(cnt)=vt*sin(angle); % velocity * random direction
    % vtot(cnt)= sqrt (vx(cnt)^2)+(vy(cnt)^2);
end
```

## Addition of constant electric field of 0.1V

```
VAreaX = 0.1; %0.1V across x direction
VAreaY = 0; %added for possibility of Y field force
Ex = VAreaX/wArea; % electric field along the x-axis
Ey = VAreaY/lArea; % electric field along the y-axis
aX = (e * Ex)/mn; % Acceleration in x due to Field
aY = (e * Ey)/mn; % Acceleration in y due to Field
vxForce= (aX)*dt; % Velocity in x due to Field
vyForce= (aY)*dt; % Velocity in y due to Field

%The value of Ex = 50000(V/m) for 0.1V across the area, The force applied
%to each particle is 8e-14 N

%The acceleration due to the field in the x direction is a =
%3.381973859521560e+17 m/s^2
```

## Q1 Main Loop

```
t=0;
intCNT = 2;
eTemp(1) = Temp;
while t < tTot
    t = t + dt;

    %Store old position
    oldx = x;
    oldy = y;

    %Update to new position with velocity including Field Force
    vx(1:numElec) = vx + vxForce;
    vy(1:numElec) = vy + vyForce;
    x(1:numElec) = x(1:numElec) + (vx(1:numElec).*dt);
    y(1:numElec) = y(1:numElec) + (vy(1:numElec).*dt);

    vtot(1:numElec)= sqrt((vx(1:numElec).^2)+(vy(1:numElec).^2));
```

```

%Apply boundary conditions
for check = 1:numElec
    if scatOn==1
        if Pscatter > rand()
            vx(check)=(vt/sqrt(2))*randn();
            vy(check)=(vt/sqrt(2))*randn();
            tScatter(check)= 0; %If collision, time goes to 0
        else
            tScatter(check)= tScatter(check) + dt; %track time increaing
while no collision
        end
    end

    %If bottom contact, bounce off in opposite direction
    if (y(check)<=0)
        y(check) = 0;
        vy(check) = -vy(check);
    end
    %If top contact, bounce off in opposite directio
    if (y(check)>=lArea)
        y(check) = lArea;
        vy(check) = -vy(check);
    end
    %if left side of box, come out right side
    if(x(check)<=0)
        x(check) = x(check) + wArea;
    end
    %if right side of box, come out left side
    if(x(check)>=wArea)
        x(check) = x(check) - wArea;
    end
end

%Plot Boundary and map some electrons
for Eplot = 1:numEPlot
    subplot (3,1,1)
    %if the electron went out of sides and back on other side, do not
    %draw line
    if abs(oldx(Eplot)-x(Eplot))<(wArea/2)
        p = plot([oldx(Eplot),x(Eplot)],[oldy(Eplot),y(Eplot)]);
    end
    p.Color=colors(Eplot,:);
    axis([0,wArea,0,lArea]);
    title('Electron Model'), xlabel('Position (m)', 'FontSize', 10),
ylabel('Position (m)', 'FontSize', 10);

    hold on;
end
pause(0.01);

%Average Temperature
subplot (3,1,2)
Time(:,intCNT) = t;

```

```
allT = ((vtot(:).^2).*mn)./(2*kb);
eTemp(:,intCNT) = mean(allT);

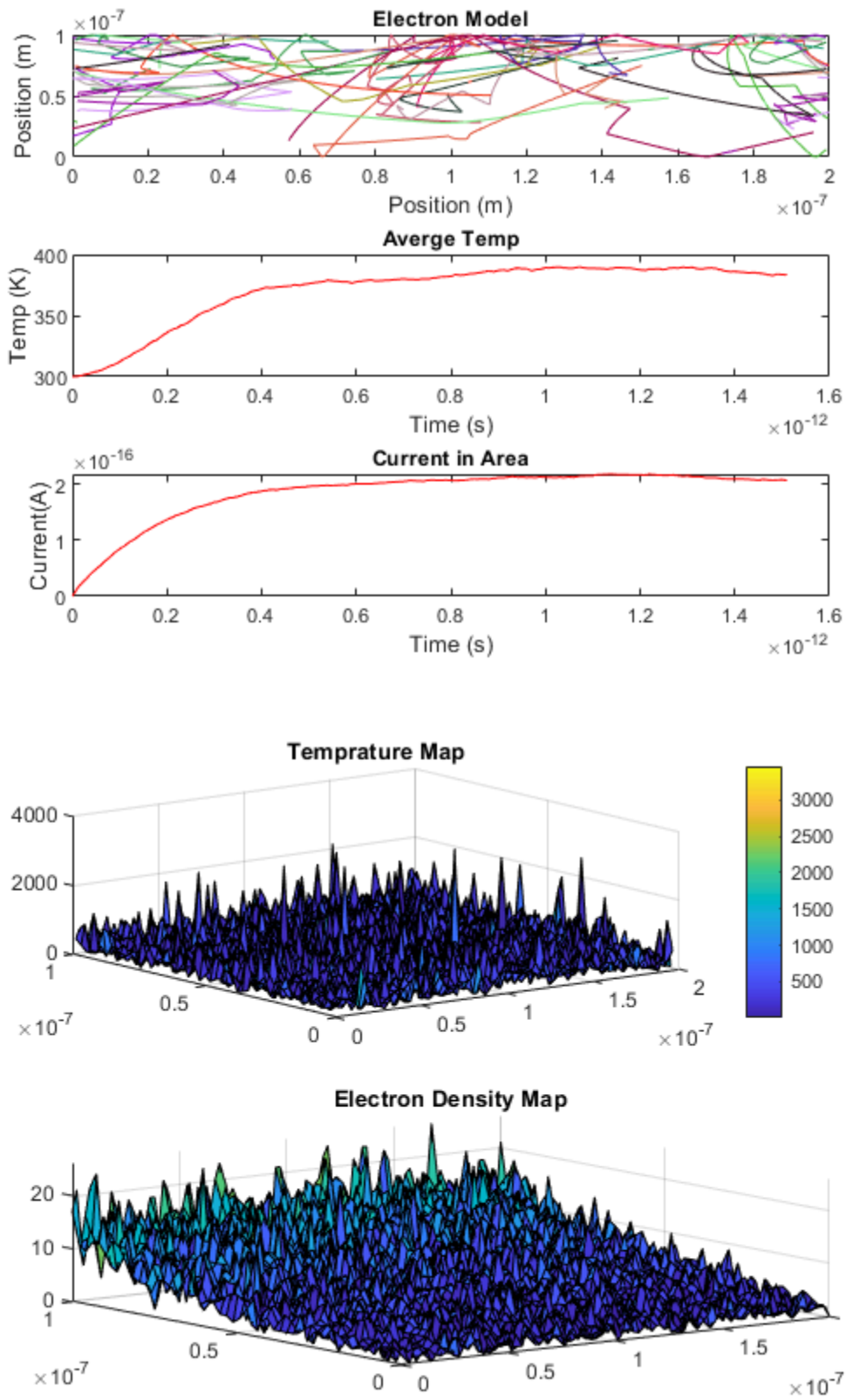
plot(Time,eTemp,"r");
title('Average Temp'),xlabel('Time (s)', 'FontSize', 10), ylabel('Temp
(K)', 'FontSize', 10);
hold on;

%Current over time with Field
Ix(:,intCNT)= e*10000000000000*mean(vx)* wArea * lArea; %Current = e * n *
vd * Area (Particle density is 10e15 cm^-2)
%not sure about value but the relationship of the current is shown in
%the plot
subplot (3,1,3)
plot(Time,Ix,"r");
title('Current in Area'),xlabel('Time (s)', 'FontSize', 10),
ylabel('Current(A)', 'FontSize', 10);
hold on;

intCNT = intCNT +1;
end
figure(2)
%Electron Temperature Map (Use Surf now to get 3D plot over the area)
allT = ((vtot(:).^2).*mn)./(2*kb);
xv = linspace(min(x), max(x),100);
yv = linspace(min(y), max(y),50);
[X,Y] = meshgrid(xv,yv);
ETM=griddata(x,y,allT,X,Y);
subplot(2,1,1);
surf(X,Y,ETM),colorbar,title("Temperature Map")
axis([0, wArea, 0 lArea]);

%Electron Density Map (Use Surf now to get 3D plot over the area)
eMapX=linspace(0, wArea, 101); %increase by 1 for surf
eMapY=linspace(0, lArea, 51); %increase by 1 for surf
EDM=histcounts2(y,x,eMapY,eMapX);
subplot(2,1,2)
surf(X,Y,EDM),title('Electron Density Map');

%The current increases since until equilibrium because there is a force
%not actin gon the particles so they move from left to reight creating
%current flow. The larger the current density, the larger the current>
%The larger the feidl strenth, the larger the current.
```



## Function for Electric Field (No change from Assignment 2)

function [V] = A2\_Function(nx, ny, xBox, yBox, boxCond, x0, x1) %Inputs: %Area x dimension, Area y dimension, box x dimension in middle of area, %Box y dimension from bottom to high and from top down, box conductivity, %x0 = volatge at left side, x1 = volatge at right side.

global Carea %NEEDS TO BE GLOBAL

% Add bottleneck Carea = ones(nx,ny); %set conduction area to 1 % In area, place boxes with new conduction (faster than for loop) Carea(nx/2 - xBox/2:nx/2 + yBox/2,1:yBox) = boxCond; %Bottom Box Carea(nx/2 - xBox/2:nx/2 + yBox/2,ny-yBox:ny) = boxCond; %Top Box

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

for i = 1:nx for j = 1:ny n = j + (i-1) \* ny; % middle nxm = j + (i-2) \* ny; % right nxp = j + i \* ny; % left nym = j-1 + (i-1) \* ny; % top nyp = j+1 + (i-1) \* ny; % down if i == 1 %Left Boundary V=Vo G(n,n) = 1; F(n,1) = x0; elseif i == nx %Right Boundary V=Vo G(n,n) = 1; F(n,1) = x1; elseif j == 1 %Bottom Boundary (Free) bxm = (Carea(i,j) + Carea(i-1,j)) / 2; bxp = (Carea(i,j) + Carea(i+1,j)) / 2; byp = (Carea(i,j) + Carea(i,j+1)) / 2;

G(n,n) = -(bxm + bxp + byp);

G(n,nxm) = bxm;

G(n,nxp) = bxp;

G(n,nyp) = byp;

elseif j == ny %Top Boundary (Free)

bxm = (Carea(i,j) + Carea(i-1,j)) / 2;

bxp = (Carea(i,j) + Carea(i+1,j)) / 2;

bym = (Carea(i,j) + Carea(i,j-1)) / 2;

G(n,n) = -(bxm + bxp + bym);

G(n,nxm) = bxm;

G(n,nxp) = bxp;

G(n,nym) = bym;

else %Middle

bxm = (Carea(i,j) + Carea(i-1,j)) / 2;

bxp = (Carea(i,j) + Carea(i+1,j)) / 2;

byp = (Carea(i,j) + Carea(i,j+1)) / 2;

bym = (Carea(i,j) + Carea(i,j-1)) / 2;

G(n,n) = -(bxm + bxp + bym + byp);

G(n,nxm) = bxm;

G(n,nxp) = bxp;

G(n,nym) = bym;

G(n,nyp) = byp;

end

end

end

V = G\F;

end

## Q2 - Add electric field for Lab 2 bottle neck program

```
clc
clear all
close all

%Initial
m0 = 9.10938215e-31;           % electron mass
mn = 0.26*m0;                 % Effective mass
Temp = 300;                   % Inital Temp (K)
kb = 1.3806504e-23;           % Boltzmann constant
tmn = 0.2e-12;                % Mean time between collision
q = 1.602e-19;                % Charge of an Electron (Assignment 3 for
    force)

% Region Area
wArea = 200e-9;
lArea = 100e-9;

%Thermal Velocity (Question 1.A)
vt=sqrt((2*kb*Temp)/mn);      % Sim in 2D so (2*kb*Temp), 3D is (3*kb*Temp)

%Electron motion
numElec = 1000;               %Number of simulated Electrons.
numEPlot = 30;                %Number of plotted Electrons
dt = (lArea*wArea)/2;         %Typically 1/100 of region size
stepsTot = 300;               %Total amount of steps (1000 took to long)
tTot= stepsTot*dt;            %Total Simulation time
x = zeros(1,numElec);         %Inital X matrix
y = zeros(1,numElec);         %Inital y matrix
vx = zeros(1,numElec);        %Inital velocity x matrix
vy = zeros(1,numElec);        %Inital velocity y matrix
vtot = zeros(1,numElec);      %Inital velocity matrix
xfield = zeros(1,numElec);     %Inital Position in E field x
yfield = zeros(1,numElec);     %Inital Position in E field y
vxForce = zeros(1,numElec);    %Inital Force due to position in E field x
vyForce = zeros(1,numElec);    %Inital Force due to position in E field y

%Probability of Scatter
scatOn = 1;                    %Turn Scatter on (1) or off(0)
Pscatter = 1-exp(-dt/tmn);     %Scatter Equation
tScatter = zeros(1,numElec);

%Bottle Neck Boundary reduced to prevent bleeding
% X limit (no particle between X1 and X2 - Cosider Y limits)
boxX1=80e-9;
boxX2=120e-9;
% Y Limit (no particles between 0 and Y1 and Y2 to Y limit)
boxY1 = 40e-9;
boxY2 = 60e-9;
```

```
%Electric Field added
x0 = 0.5; %voltage at right side of area increased to 0.5V to see huge effect
x1 = 0; %Voltage at left side of area
nx = wArea*1e9; % # of columns (changes to have same ratio as A1 area 2/1)
ny = lArea*1e9; % # of rows
xBox = 40; %Width of box (changes to same box ratio as A1 area l= 40 and w=40)
yBox = 40; %Hight of box
boxCond = 0.01;
V=A2_Function(nx, ny, xBox, yBox, boxCond, x0, x1);
Vmap = reshape(V, [ny, nx]);
[Ex,Ey] = gradient(-Vmap);

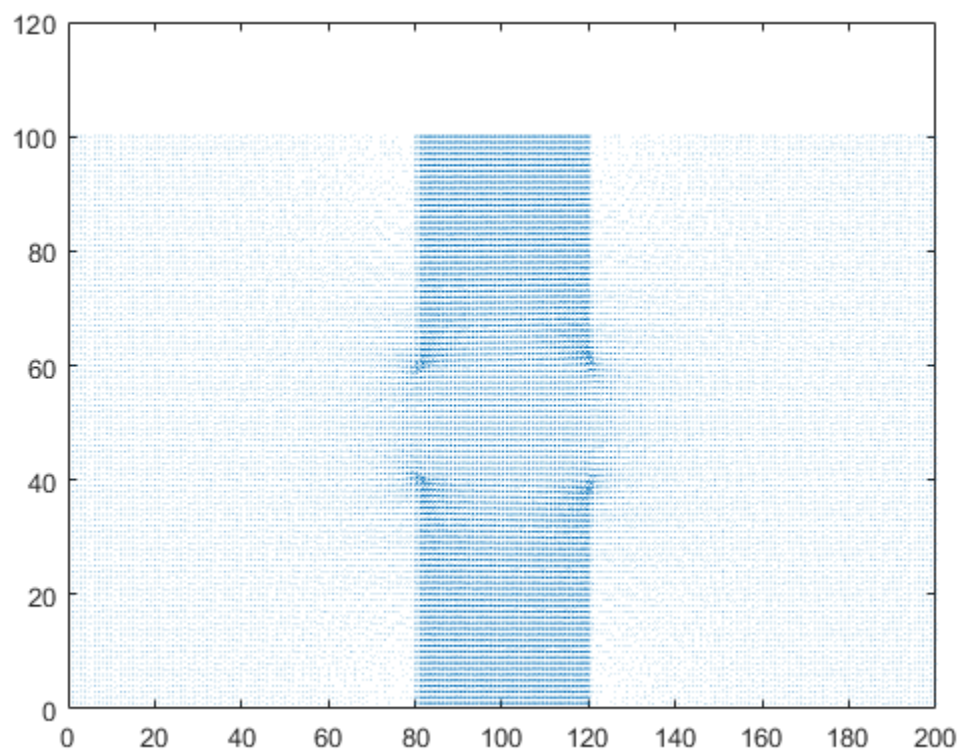
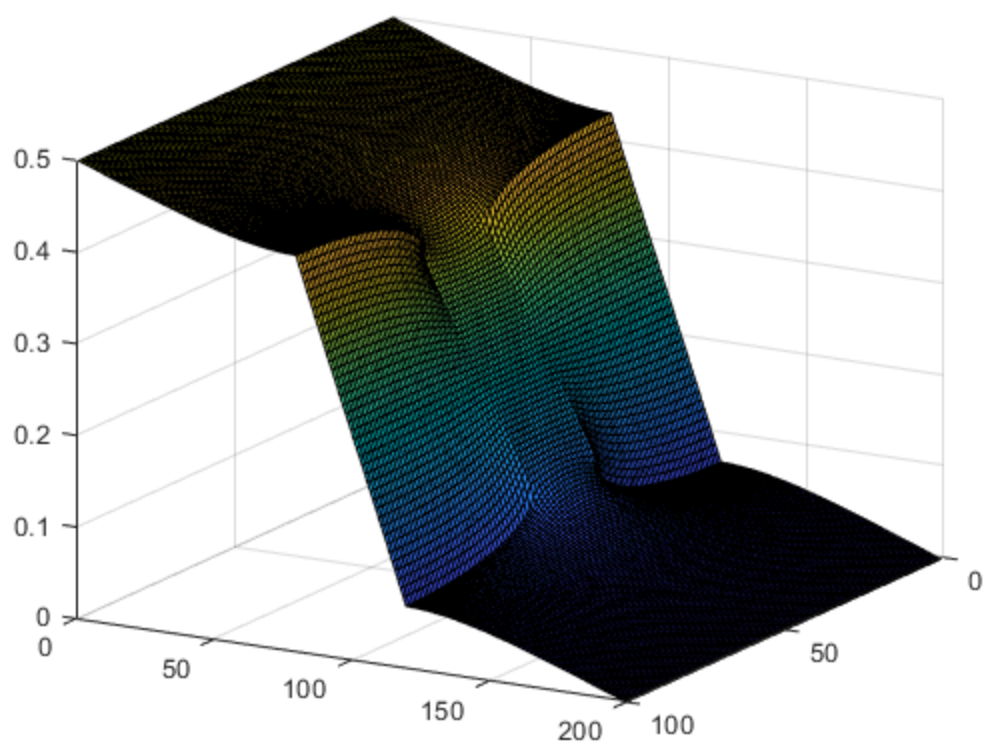
figure('name', 'Voltage Surface Plot')
surf(Vmap),view(120,20);

figure('name', 'Electric Field Vector Plot');
quiver(Ex,Ey);

%Electron Graph
for cnt = 1:numElec
    x(cnt)=rand()*wArea;
    y(cnt)=rand()*lArea;
    %If the electrons are place in the box, re-roll position
    while (x(cnt)>=boxX1 && x(cnt)<=boxX2 && (y(cnt)<=boxY1 ||
y(cnt)>=boxY2))) %Relocate them if in boundary
        x(cnt)=rand()*wArea;
        y(cnt)=rand()*lArea;
    end
    vx(cnt)=(vt/sqrt(2))*randn(); % velocity * Gaussian dist
    vy(cnt)=(vt/sqrt(2))*randn(); % velocity * Gaussian dist
    vtot(cnt)= sqrt (vx(cnt)^2)+(vy(cnt)^2);
    colors= rand(numElec,3);
end

%Boundary Energy/Velocity loss coefficient (reduction in velocity =
%reduction in temprature)
vloss = 0.95;
```





## Q2 Main Loop

```
t=0;
while t < tTot
    t = t + dt;

    %Store old position
    oldx = x;
    oldy = y;

    %Update to new position
    x(1:numElec) = x(1:numElec) + (vx(1:numElec).*dt);
    y(1:numElec) = y(1:numElec) + (vy(1:numElec).*dt);

    vtot(1:numElec)= sqrt ((vx(1:numElec).^2)+(vy(1:numElec).^2));

    for check = 1:numElec
        %Scatter
        if scatOn==1
            if Pscatter > rand()
                vx(check)=(vt/sqrt(2))*randn();
                vy(check)=(vt/sqrt(2))*randn();
                tScatter(check)= 0; %If collision, time goes to 0
            else
                tScatter(check)= tScatter(check) + dt; %track time increaing
            end
        end
        while no collision
            end
        end

        % Bring X and Y of particle location to values able to be tracked
        % in electric field mesh to apply field forces
        xfield(check) = round(x(check)*1e9);
        yfield(check) = round(y(check)*1e9);

        %Make sure field effects still apply to particles which go across
        %area
        if xfield(check)>200
            xfield(check) = 200;
        end
        if xfield(check)<1
            xfield(check) = 1;
        end
        if yfield(check)>100
            yfield(check) = 100;
        end
        if yfield(check)<1
            yfield(check) = 1;
        end
        %Apply field force to particles
        vxForce(check)= ((q *Ex(yfield(check),xfield(check)))/(wArea/nx))/
mn)*dt; % V=(q*E)/mass F=q*E
        vyForce(check)= ((q *Ey(yfield(check),xfield(check)))/(lArea/ny))/
mn)*dt; % V=(q*E)/mass F=q*E
```

```
vx(check) = vx(check)+ vxForce(check);
vy(check) = vy(check)+ vyForce(check);

%Apply Boundary Conditions
%If bottom contact, bounce off in opposite direction
if (y(check)<=0)
    y(check) = 0;
    vy(check) = -vy(check);
end
%If top contact, bounce off in opposite direction
if (y(check)>=lArea)
    y(check) = lArea;
    vy(check) = -vy(check);
end
%If left side of box, come out right side
if(x(check)<=0)
    x(check) = x(check) + wArea;
end
%If right side of box, come out left side
if(x(check)>=wArea)
    x(check) = x(check) - wArea;
end

%Apply bottle neck conditions
%If contact on left walls of boundary (not in Gap)
if (oldx(check)<boxX1 && x(check)>=boxX1 && (y(check)<= boxY1 ||
y(check)>= boxY2) && oldx(check)>((1/5)*wArea) && oldx(check)<((4/5)*wArea))
    x(check)=boxX1;
    vx(check) = -(vx(check)*vloss);
end
%If contact on right walls of boundary (not in Gap)
if (oldx(check)>boxX2 && x(check)<=boxX2 && (y(check)<= boxY1 ||
y(check)>= boxY2) && oldx(check)>((1/5)*wArea)&& oldx(check)<((4/5)*wArea))
    x(check)=boxX2;
    vx(check) = -(vx(check)*vloss);
end
%If contact with bottom boundary in gap
if (x(check)>boxX1 && x(check)< boxX2 && oldy(check)>boxY1 &&
y(check)<= boxY1)
    y(check)= boxY1;
    vy(check) = -(vy(check)*vloss);
end
%If contact with top boundary in gap
if (x(check)>boxX1 && x(check)< boxX2 && oldy(check)<boxY2 &&
y(check)>=boxY2)
    y(check)=boxY2;
    vy(check) = -(vy(check)*vloss);
end
end

%Plot Boundary and map some electrons
for Eplot = 1:numEPlot
    figure(3)
    %if the electron went out of sides and back on other side, do not
```

---

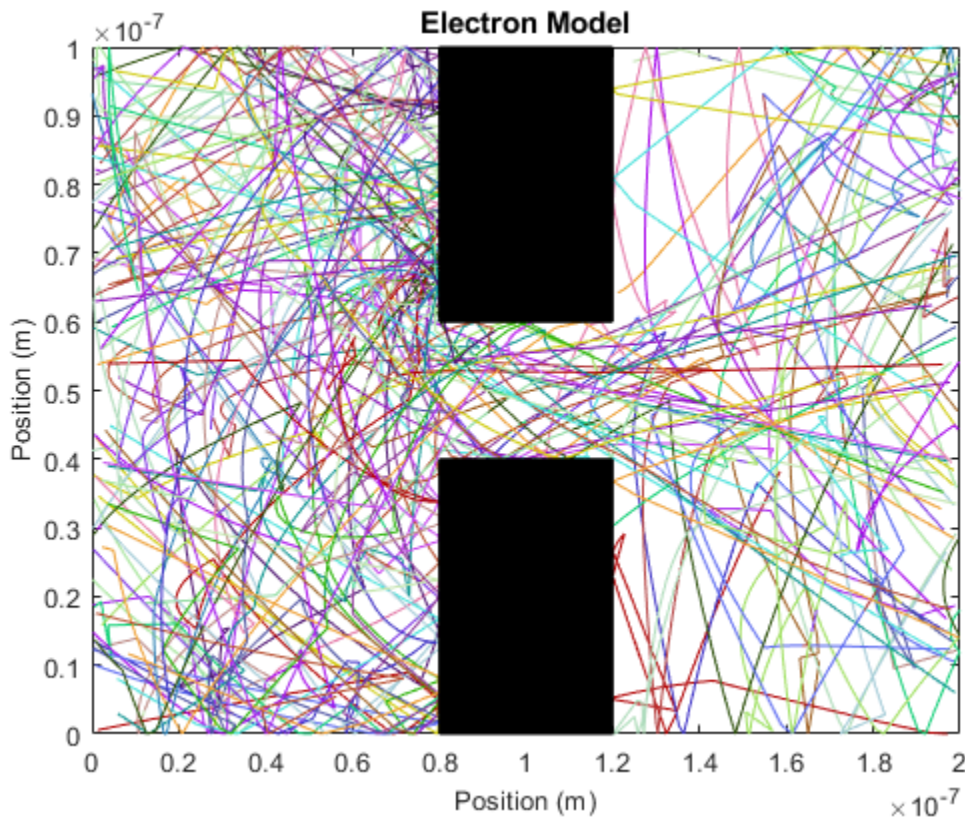
```

%draw line
if abs(olddx(Eplot)-x(Eplot))<(wArea/2)
    p = plot([olddx(Eplot),x(Eplot)],[oldy(Eplot),y(Eplot)]);
end

rectangle('Position',[boxX1 0 (boxX2-boxX1) boxY1],'FaceColor',[0 0
0])

rectangle('Position',[boxX1 boxY2 (boxX2-boxX1) (lArea-
boxY2)],'FaceColor',[0 0 0])
p.Color=colors(Eplot,:);
axis([0,wArea,0,lArea]);
title('Electron Model'), xlabel('Position (m)', 'FontSize', 10),
ylabel('Position (m)', 'FontSize', 10);
pause(0.1);
hold on;
end
end

```



**%% Q3 - Add electric field for Lab 2 bottle neck with density map**

```

clc
clear all
close all
warning('off')

```

```
%Initial
m0 = 9.10938215e-31;           % electron mass
mn = 0.26*m0;                  % Effective mass
Temp = 300;                    % Inital Temp (K)
kb = 1.3806504e-23;           % Boltzmann constant
tmn = 0.2e-12;                 % Mean time between collision
q = 1.602e-19;                 % Charge of an Electron (Assignment 3 for
    force)

% Region Area
wArea = 200e-9;
lArea = 100e-9;

%Thermal Velocity (Question 1.A)
vt=sqrt((2*kb*Temp)/mn);       % Sim in 2D so (2*kb*Temp), 3D is (3*kb*Temp)

%Electron motion
numElec =30000;                %Number of simulated Electrons
numEPlot = 20;                 %Number of plotted Electrons
dt = (lArea*wArea)/2;          %Typically 1/100 of region size
stepsTot = 400;                %Total amount of steps (1000 was a long
    simulation)
tTot= stepsTot*dt;             %Total Simulation time
x = zeros(1,numElec);          %Inital X matrix
y = zeros(1,numElec);          %Inital y matrix
vx = zeros(1,numElec);         %Inital velocity x matrix
vy = zeros(1,numElec);         %Inital velocity y matrix
vtot = zeros(1,numElec);       %Inital velocity matrix
xfield = zeros(1,numElec);     %Inital Position in E field x
yfield = zeros(1,numElec);     %Inital Position in E field y
vxForce = zeros(1,numElec);    %Inital Force due to position in E field x
vyForce = zeros(1,numElec);    %Inital Force due to position in E field y

%Probability of Scatter
scatOn = 1;                    %Turn Scatter on (1) or off(0)
Pscatter = 1-exp(-dt/tmn);     %Scatter Equation
tScatter = zeros(1,numElec);

%Bottle Neck Boundary reduced to prevent bleeding
% X limit (no particle between X1 and X2 - Cosider Y limits)
boxX1=80e-9;
boxX2=120e-9;
% Y Limit (no particles between 0 and Y1 and Y2 to Y limit)
boxY1 = 40e-9;
boxY2 = 60e-9;

%Electric Field
x0 = 0.8; %voltage at right side of area increased to 0.8V to see huge effect
x1 = 0; %Voltage at left side of area
nx = wArea*1e9; % # of colums (changes to have same ratio as A1 area 2/1)
ny = lArea*1e9; % # of rows
xBox = 40; %Width of box (changes to same box ratio as A1 area l= 40 and w=40)
yBox = 1:1:50; %Hight of box for bottleneck
```

```
boxCond = 0.01;
V=A2_Function(nx, ny, xBox, yBox, boxCond, x0, x1);
Vmap = reshape(V, [ny, nx]);
[Ex,Ey] = gradient(-Vmap);

% figure('name', 'Voltage Surface Plot')
% surf(Vmap),view(120,20);
%
% figure('name', 'Electric Field Vector Plot');
% quiver(Ex,Ey);

cur = zeros(50,1);
global Carea %Must declare global for both in and out of function

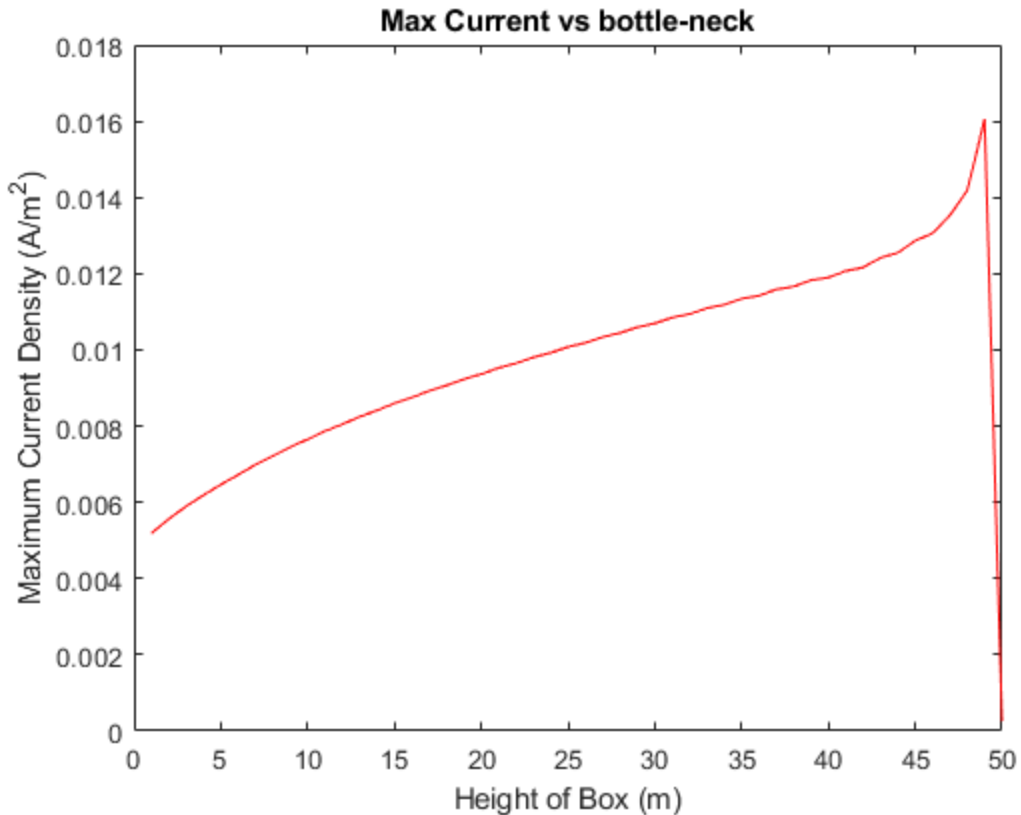
for a = 1:50
    V=A2_Function(nx, ny, xBox, a, boxCond, x0, x1);
    Vmap = reshape(V, [ny, nx]);
    J = Carea'.*gradient(-Vmap);
    cur(a,1) = max(J,[],"all");
end

figure('name', 'Max Current vs bottle-neck');
plot(yBox,cur, 'r');
xlabel('Height of Box (m)');
ylabel('Maximum Current Density (A/m^2)');
title('Max Current vs bottle-neck');

%Reset yBox value after bottle-neck plot
yBox = 40;

%Electron Graph
for cnt = 1:numElec
    x(cnt)=rand()*wArea;
    y(cnt)=rand()*lArea;
    %If the electrons are place in the box, re-roll position
    while (x(cnt)>=boxX1 && x(cnt)<=boxX2 && (y(cnt)<=boxY1 ||
y(cnt)>=boxY2))) %Relocate them if in boundary
        x(cnt)=rand()*wArea;
        y(cnt)=rand()*lArea;
    end
    vx(cnt)=(vt/sqrt(2))*randn(); % velocity * Gaussian dist
    vy(cnt)=(vt/sqrt(2))*randn(); % velocity * Gaussian dist
    vtot(cnt)= sqrt (vx(cnt)^2)+(vy(cnt)^2);
    colors= rand(numElec,3);
end

%Boundary Energy/Velocity loss coefficient (reduction in velocity =
%reduction in temprature)
vloss = 0.95;
```



## Q3 Main Loop

```

t=0;
while t < tTot
    t = t + dt;

    %Store old position
    oldx = x;
    oldy = y;

    %Update to new position
    x(1:numElec) = x(1:numElec) + (vx(1:numElec).*dt);
    y(1:numElec) = y(1:numElec) + (vy(1:numElec).*dt);

    vtot(1:numElec)= sqrt ((vx(1:numElec).^2)+(vy(1:numElec).^2));

    for check = 1:numElec
        %Scatter
        if scatOn==1
            if Pscatter > rand()
                vx(check)=(vt/sqrt(2))*randn();
                vy(check)=(vt/sqrt(2))*randn();
                tScatter(check)= 0; %If collision, time goes to 0
            else

```

```
        tScatter(check)= tScatter(check) + dt; %track time increaing
while no collision
    end
end

% Bring X and Y of particle location to values able to be tracked
% in electric field mesh to apply field forces
xfield(check) = round(x(check)*1e9);
yfield(check) = round(y(check)*1e9);

%Make sure field effects still apply to particles which go across
%area
if xfield(check)>200
    xfield(check) = 200;
end
if xfield(check)<1
    xfield(check) = 1;
end
if yfield(check)>100
    yfield(check) = 100;
end
if yfield(check)<1
    yfield(check) = 1;
end
%Apply field force to particles
vxForce(check)= ((q *Ex(yfield(check),xfield(check)))/(wArea/nx))/
mn)*dt; % V=(q*E)/mass F=q*E
vyForce(check)= ((q *Ey(yfield(check),xfield(check)))/(lArea/ny))/
mn)*dt; % V=(q*E)/mass F=q*E
vx(check) = vx(check)+ vxForce(check);
vy(check) = vy(check)+ vyForce(check);

%Apply Boundary Conditions
%If bottom contact, bounce off in opposite direction
if (y(check)<=0)
    y(check) = 0;
    vy(check) = -vy(check);
end
%If top contact, bounce off in opposite direction
if (y(check)>=lArea)
    y(check) = lArea;
    vy(check) = -vy(check);
end
%If left side of box, come out right side
if(x(check)<=0)
    x(check) = x(check) + wArea;
end
%If right side of box, come out left side
if(x(check)>=wArea)
    x(check) = x(check) - wArea;
end

%Apply bottle neck conditions
%If contact on left walls of boundary (not in Gap)
```

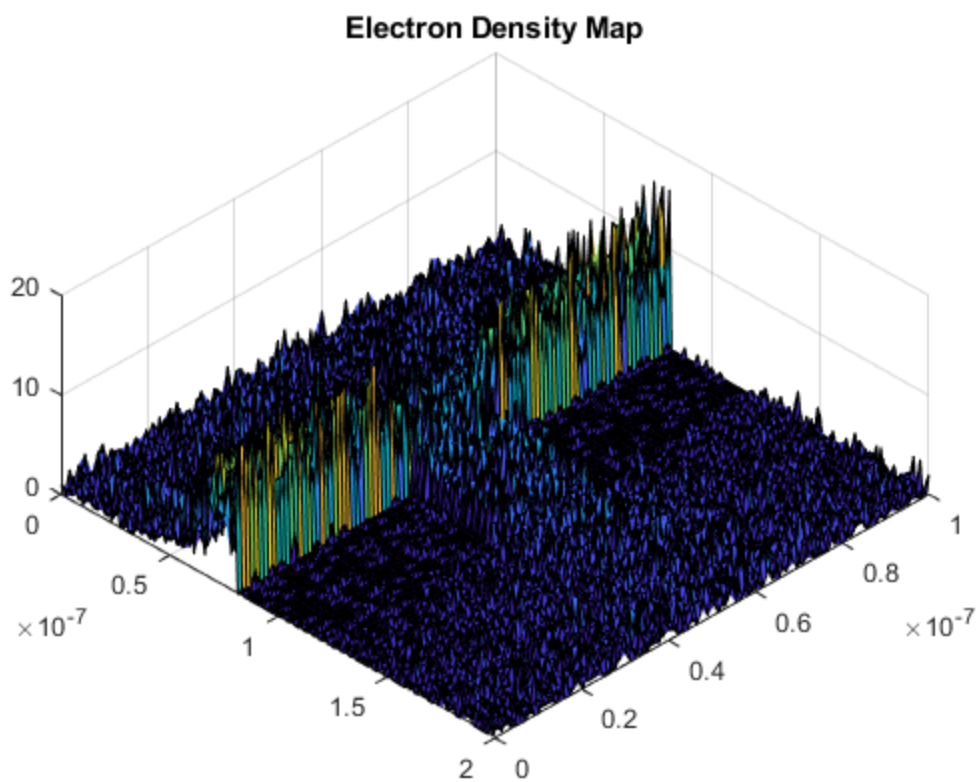


```
    if (oldx(check)<boxX1 && x(check)>=boxX1 && (y(check)<= boxY1 ||
y(check)>= boxY2) && oldx(check)>((1/5)*wArea) && oldx(check)<((4/5)*wArea))
        x(check)=boxX1;
        vx(check) = -(vx(check)*vloss);
    end
    %If contact on right walls of boundary (not in Gap)
    if (oldx(check)>boxX2 && x(check)<=boxX2 && (y(check)<= boxY1 ||
y(check)>= boxY2) && oldx(check)>((1/5)*wArea)&& oldx(check)<((4/5)*wArea))
        x(check)=boxX2;
        vx(check) = -(vx(check)*vloss);
    end
    %If contact with bottom boundary in gap
    if (x(check)>boxX1 && x(check)< boxX2 && oldy(check)>boxY1 &&
y(check)<= boxY1)
        y(check)= boxY1;
        vy(check) = -(vy(check)*vloss);
    end
    %If contact with top boundary in gap
    if (x(check)>boxX1 && x(check)< boxX2 && oldy(check)<boxY2 &&
y(check)>=boxY2)
        y(check)=boxY2;
        vy(check) = -(vy(check)*vloss);
    end
end
end
```

```
figure(2)
xv = linspace(min(x), max(x),200);
yv = linspace(min(y), max(y),100);
[X,Y] = meshgrid(xv,yv);
eMapX=linspace(0, wArea, 201); %increase by 1 for surf
eMapY=linspace(0, lArea, 101); %increase by 1 for surf
EDM=histcounts2(y,x,eMapY,eMapX);
surf(X,Y,EDM),view(45,60),title('Electron Density Map');
```

%Improving the simulation accuracy would be including a more realistic  
%bottleneck shape and structure. A perfect rectangle is not a realistic doping  
shape.

% Including Quantum tunneling was introduced for devices. Better model  
%for electron hitting eachother than a random scatter constant, two  
%electron hitting eachother moving forward would still both move forward.



*Published with MATLAB® R2021b*