Enhancements

In this part , two rectangular boundaries are added. These boundaires do not allow electrons to go through and they are reflected depending on where they are coming from.

```
clear

global C



C.q_0 = 1.60217653e-19;             % electron charge
C.hb = 1.054571596e-34;             % Dirac constant
C.h = C.hb * 2 * pi;                    % Planck constant
C.m_0 = 9.10938215e-31;             % electron mass
C.kb = 1.3806504e-23;               % Boltzmann constant
C.eps_0 = 8.854187817e-12;          % vacuum permittivity
C.mu_0 = 1.2566370614e-6;           % vacuum permeability
C.c = 299792458;                    % speed of light
C.g = 9.80665; %metres (32.1740 ft) per sÂ²


nSim = 300;
noe = 100;
r2 = randi(360,noe,1);




colourArray = rand(noe,1);


xbound = 200;
ybound = 100;
x =  randi(200,noe,1);
y =  randi(100,noe,1);
vth = sqrt((C.kb * 300)/(C.m_0 * 0.26));
vx = vth * cos(r2) ;
vy = vth * sin(r2);

%
% %these will have x > 1.2 and x < 0.8 as 1s
% tempsxupper = x >1.2
% tempxlower = x < 0.8
%
% tempxupper0
% tempxlower0 =


for pos = 1: noe
    xpos = x(pos);
    if (xpos < 120 && xpos > 80)
```

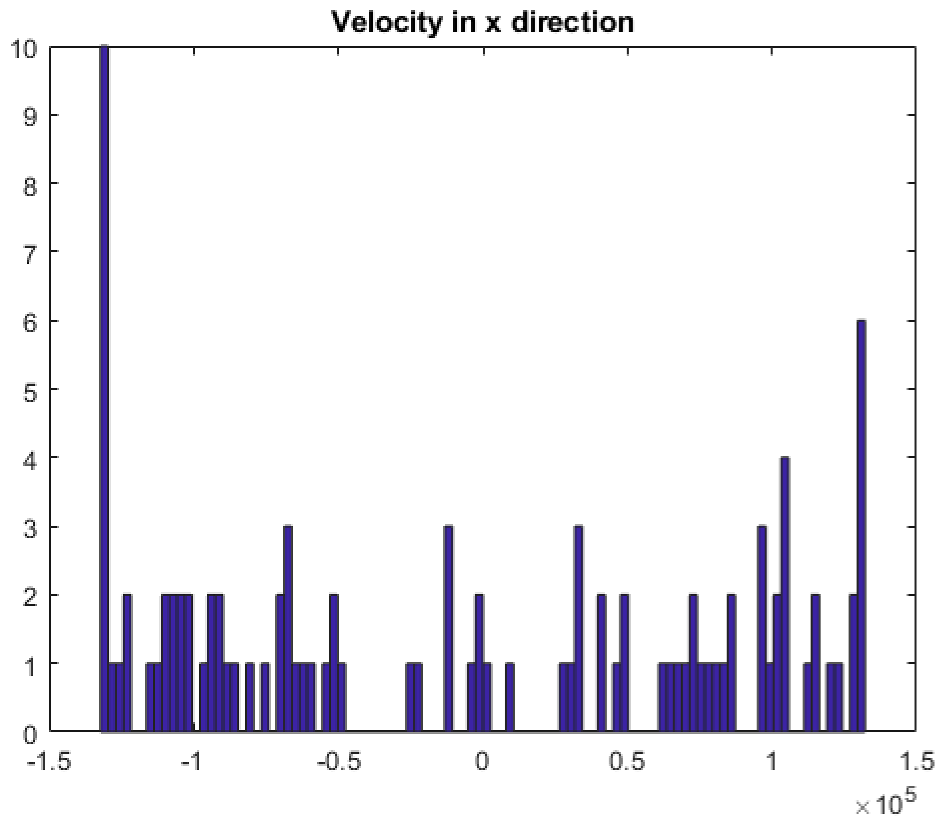
```
        if (y(pos) < 40)
            xpos = xpos + 50;
            x(pos) = xpos;


        elseif(y(pos) > 60)
            xpos = xpos - 50;
            x(pos) = xpos;

        else

        end


    end
end


MFP = vth * 0.2 * 10^-12;

figure(1);
hist(vx,100);
title("Velocity in x direction");

figure(2);
hist(vy,100);
title ("Velocity in y direction");

pScat = 1 - exp((-3 * 10^-16)/(0.2 * 10^-12));
%PscatArray = pScat * ones(noe,1)

tMatrix = zeros(noe);





for t = 1:nSim


    vxc = vx; % create copy of vx
    vyc = vy; % create copy of vy
    [n,m] = size(vx);
    [n1,m1] = size(vy);

    %%randomly permutation of positions in vx and vy%%%%%
    idx = randperm(n);
    randomvx = vx;
    randomvx(idx,1)= vx (:,1) ;

    idy = randperm(n1);
    randomvy = vy;
    randomvy(idy,1) = vy(:,1);
```

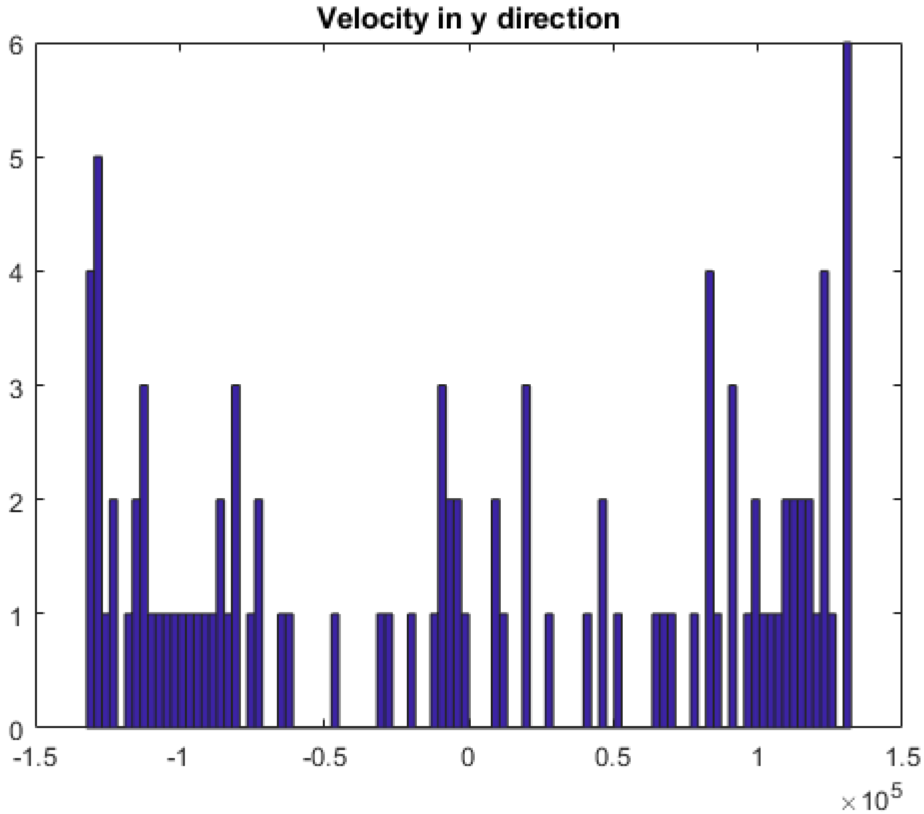
```
%Modelling scattering%%%%%%
rScatter= rand(noe,1);

%this gives 1s and 0s. 1 means it scatters
tempScatter = rScatter < pScat;
randomvx = tempScatter .* randomvx; % not scattered are 0s
randomvy = tempScatter .* randomvy;  % not scattered are 0s

%not scattered
notScatter = rScatter >= pScat;
%%%%%%%%%%%%%%%%%%%%%%%%%

vx = vx .* notScatter; % the scattered vx are now 0
vy = vy .* notScatter; % scattered vy = 0

vx = vx + randomvx;
vy = vy + randomvy;



%%%%%%%%%%%%%%
xc = x; % x copy
yc = y ;% y copy


%Reflecting for y bounds%
temp = y >= ybound;
temp1 = y < ybound;


temp = temp * -1;

tempHigher = temp + temp1;


temp2 = y <= 0;
temp3 = y > 0;

temp2 = temp2 * -1;
tempLower = temp2 + temp3;

vy = vy .* tempHigher;
vy = vy .* tempLower;

%%%%%%%%%%%%%%%%%%%

% when x > 200%%%%%
tempx1 = x <= 200;

x = x .* tempx1;
%%%%%%%%%%%%%%%%%%

%%When x goes less than zero , come from 200 %%%%%
```

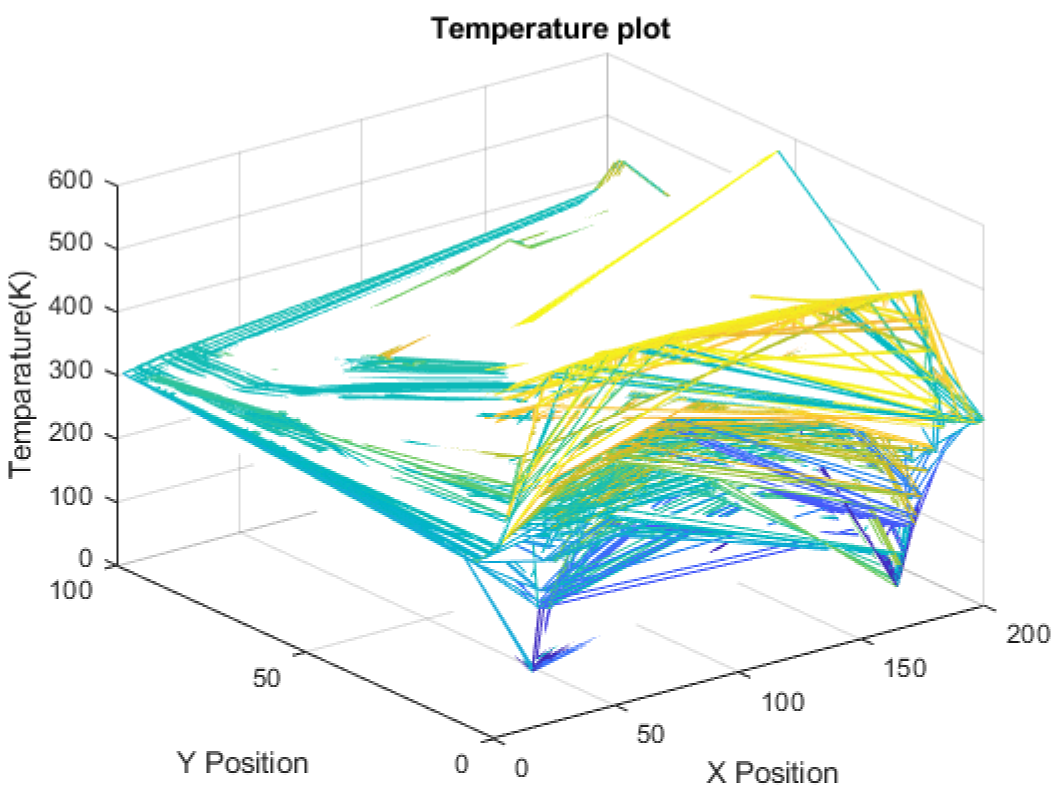
```
tempx2 = x < -0.1;


tempx2 = tempx2 * 200;
tempxFinal = x + tempx2;

x = tempxFinal;

%%%%Dealing with the lower rectangle%%%%%%
tLR1s = ( x > 80 & x < 120) & y < 40;
tLR0s = tLR1s == 0;
tLR1s = -1 * tLR1s;

f = tLR1s + tLR0s;

vx = vx .* f;

tLR1s = ( x > 80 & x < 120) & (y < 41 & y >= 40);
tLR0s = tLR1s == 0;
tLR1s = -1 * tLR1s;

f = tLR1s + tLR0s;

vy = vy .* f;

%tempFinalLower = x .* y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%Dealing with the upper rectangle%%%%%%%
tUR1s = ( x > 80 & x < 120) & y > 60;
tUR0s = tUR1s == 0;
tUR1s = -1 * tUR1s;

f = tUR1s + tUR0s;

vx = vx .* f;


tUR1s = ( x > 80 & x < 120) & (y >59 & y < 60);
tUR0s = tUR1s == 0;
tUR1s = -1 * tUR1s;

f = tUR1s + tUR0s;

vy = vy .* f;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%
dx = vx * (1/200000);
dy = vy * (1/200000);

x = x + dx;
```

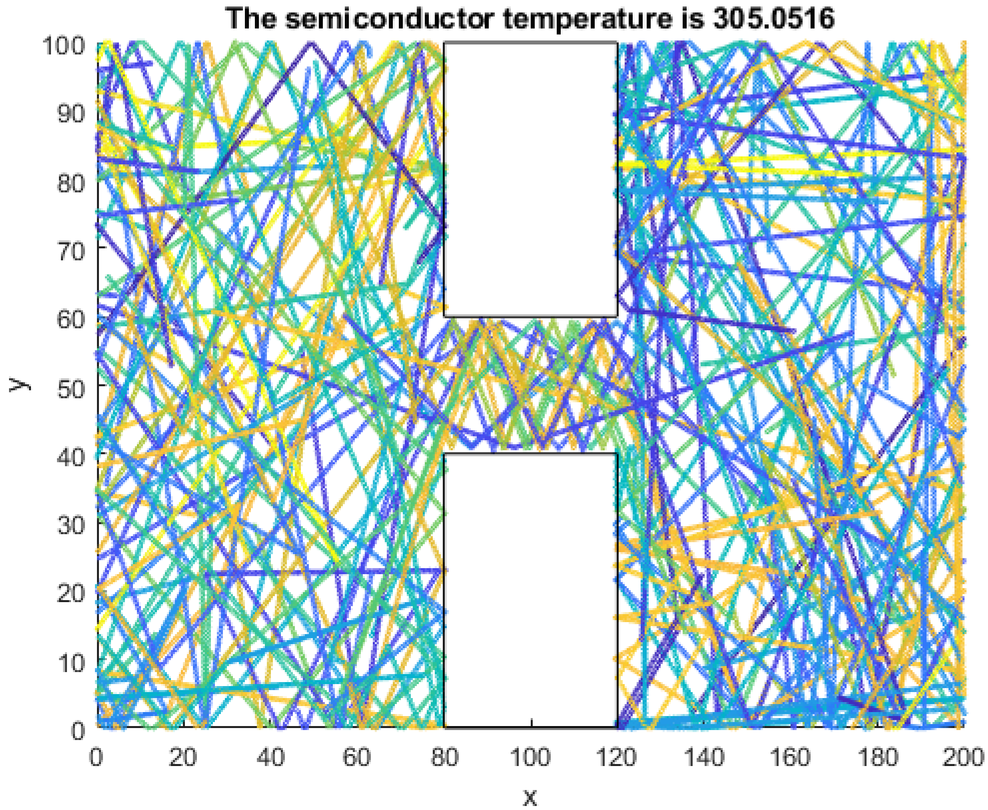
```
    y = y + dy;
    vsq = (vy).^2 + (vx).^2 ;
    average = mean(vsq);

    tMatrix = ((vsq * 0.26 *  C.m_0)/C.kb);

    semiCTemperature = (average *(0.26)* C.m_0)/(C.kb);

    [X,Y] = meshgrid (x , y);
    f1 = scatteredInterpolant(x,y,tMatrix);
    Z = f1(X,Y);
    figure (10);
    mesh(X,Y,Z);
    title('Temperature plot');
    xlabel('X Position');
    ylabel('Y Position');
    zlabel('Tempararature(K)');
    %axis tight;hold on
    %plot3(x,y,tMatrix,'.','MarkerSize',15)

    figure (3);
    semiCTemperature = (average *(0.26)* C.m_0)/(C.kb);
    plot(t , semiCTemperature,'.r');
    title('Temperature plot')
    xlabel('time');
    ylabel('temperature(K)');
    axis([0 300 200 400]);
    hold on;

    figure(4);

    scatter (x, y , 3 ,colourArray);
    axis([0 200 0 100]);
    rectangle('Position',[80 0 40 40]);
    rectangle('Position',[80 60 40 40]);
    xlabel("x");
    ylabel("y");
    hold on;

    title ("The semiconductor temperature is " + semiCTemperature);
    %pause(0.001)
    figure (3);
    hold on;
end

%define the value of r over a 2D grid:
scatter(x,y,'r.');
hold on;
%[n,c]= hist3([x,y])
%contour(c{1},c{2},n)

Elecpos = [x,y];
D = hist3(Elecpos(:,1:2),'Nbins',[20,10]);
figure (6);
```

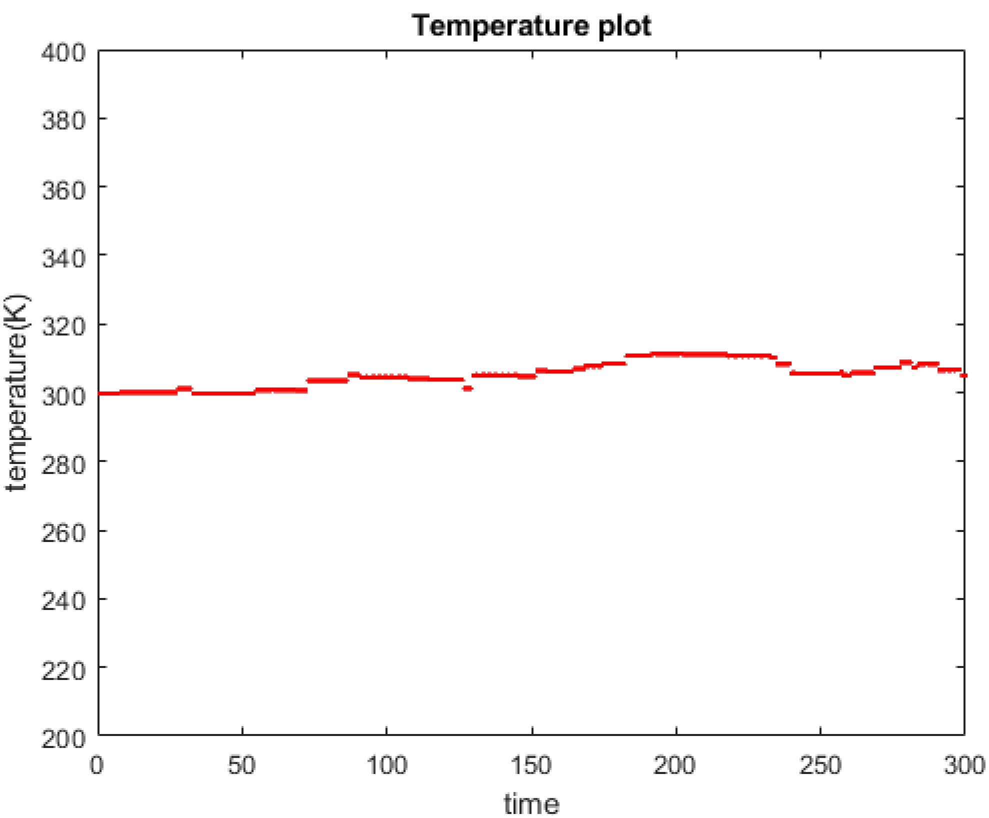
```
surf(D);
title('Electron Density plot');
shading interp;
```

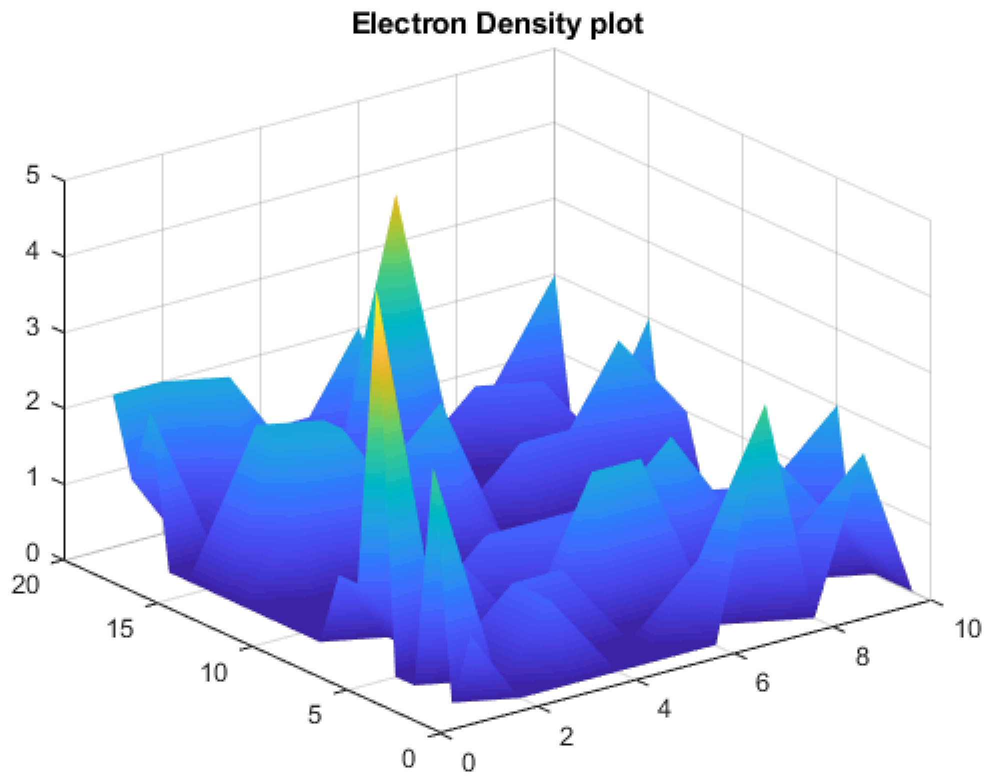



Velocity in y direction
6
5
4
3
2
1
0
-1.5
-1
-0.5
0
0.5
1
1.5
$\times 10^5$


Temperature plot
600
500
400
300
200
100
0
Temparature(K)
100
50
0
Y Position
0
50
100
150
200
X Position


The semiconductor temperature is 305.0516
y
x
0
20
40
60
80
100
120
140
160
180
200
10
30
50
70
90


Temperature plot
temperature(K)
time
200
220
240
260
280
300
320
340
360
380
400
0
50
100
150
250



Published with MATLAB® R2018a