```
%Author: Richard Finney 100967048

%%ELEC 4700 Monte-Carlo Modeling of Electron Transport
```

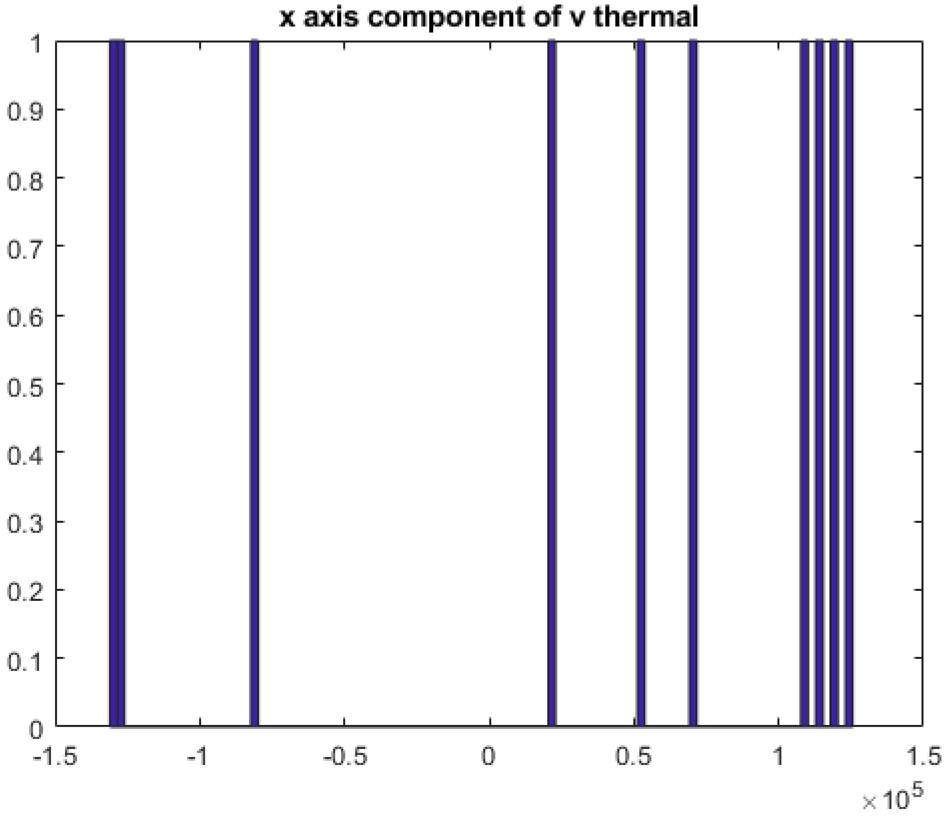
Part 3

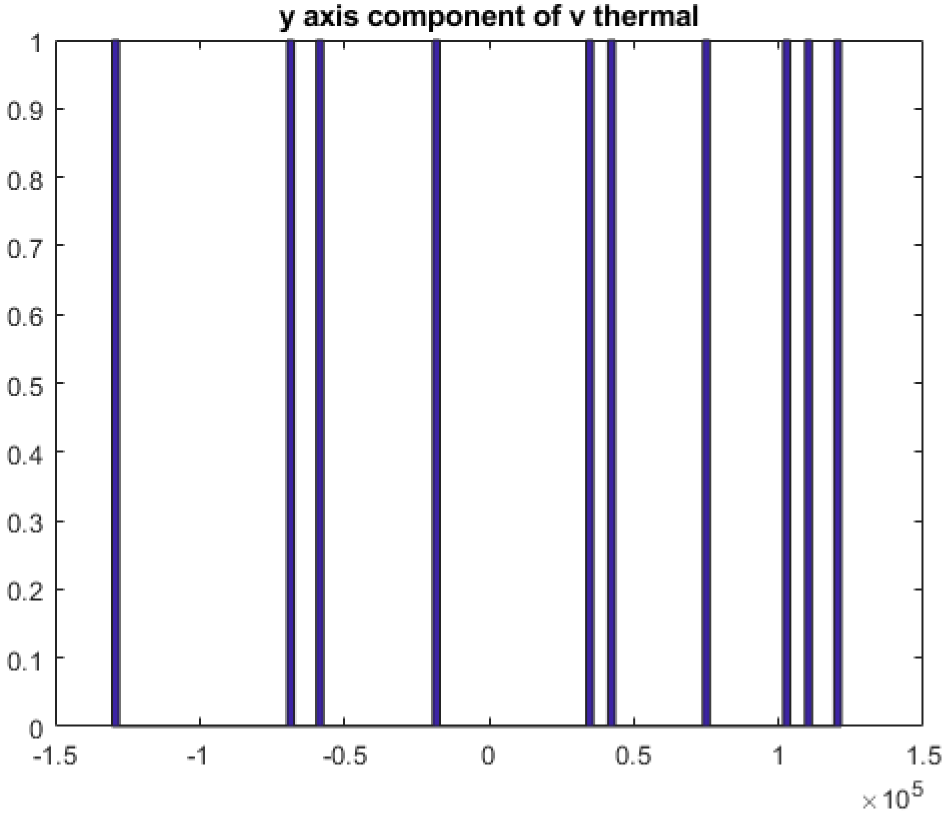
```
%  The puepose of this code is to model the electrons in the silicon
 as particles
%  with the effective mass above using a simplistic Monte-Carlo model.
%  now has scattering and rectangle boundaries. Forgot to answer this
 in
%part 2 but the MFP and temperature changes because of the scattering
%but settles. Also has density plot and temperature map

%Author: Richard Finney 100967048
clear all;
close all;

%Constants

q_0 = 1.60217653e-19;             % electron charge
m_0 = 9.10938215e-31;             % electron mass
kB = 1.3806504e-23;               % Boltzmann constant
deltat = 0.2e-12;                 % mean time between collisions
mn = 0.26*m_0;                    % effective mass of electrons



%variables

numofelec = 10;             %current numbers of electrons t be
 simulated
T = 300;                      %temperature in kelvin

dt = 1;
%Assign each particle with the fixed velocity given by vth but give
 each one a
%random direction.

vth = sqrt((kB*T)/mn);
%Spatial Boundaries

Length = 200;
Width = 100;


    %I am going to represent the location of each electron using
 vectors

x = randi([0 Length], 1, numofelec)*1e-9;       %initializing x
y = randi([0 Width], 1, numofelec)*1e-9;        %initializing y
```

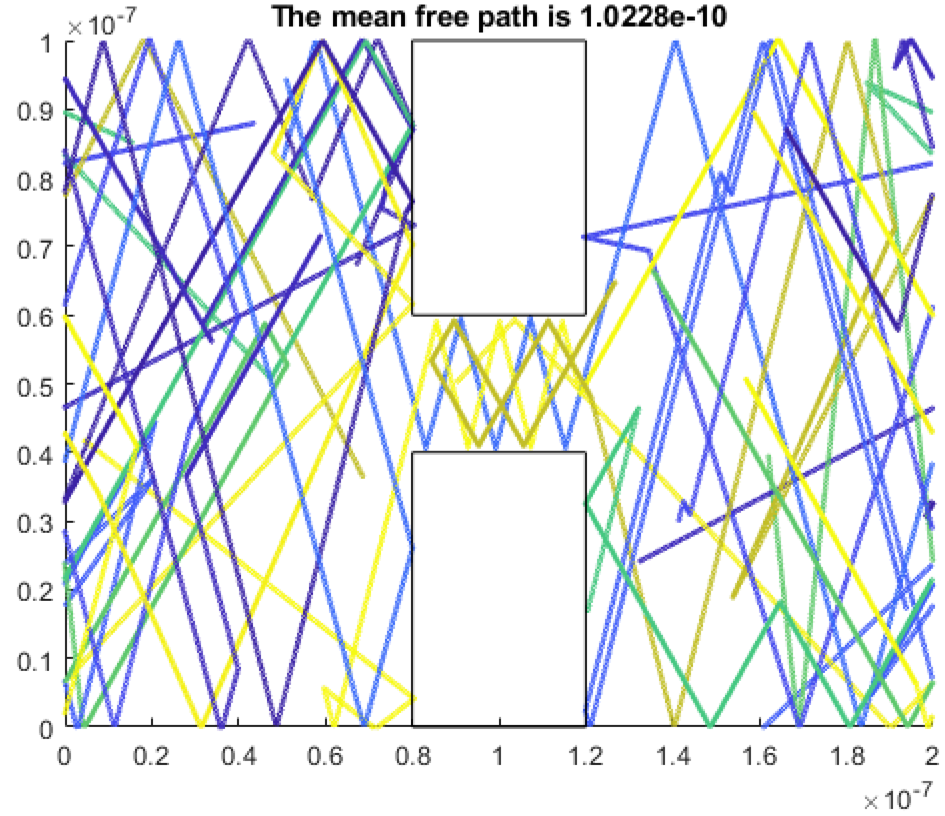
```
    %top side of lower rectangle
    for it=1:1:numofelec

        %moving spawned electrons outside of rectangles

        if x(1,it) >=(80e-9) && x(1,it) <= (120e-9) && y(1,it)<=
 (40e-9)
               x(1,it) = x(1,it) + randi([45 80], 1,1)*1e-9;
        end

        if x(1,it) >=(80e-9) && x(1,it) <= (120e-9) && y(1,it)>=
 (60e-9)
               x(1,it) = x(1,it) - randi([45 80], 1,1)*1e-9;
        end

    end

    %now we have position vectors for the x and y positions of each
    %electron. Need to create vectors for vy and vx. Remember that
 each
    %electron has a rand angle to start with, but same velocity vth.

angles = randi([0 360], 1, numofelec);
v_x = zeros(1, numofelec);
v_y = zeros(1, numofelec);

v_x = vth*cos(angles);
v_y = vth*sin(angles);

figure(1)
hist(v_x,100);
title('x axis component of v thermal');

figure(2)
hist(v_y,100);
title('y axis component of v thermal');

    %scatter
    pscat = 1 - exp(-1e-14/(1e-11*0.2));
    pscatvector = ones(1,numofelec)*pscat;

    colorarray= rand(1,numofelec);
for time= 1:dt:1000



    random = rand(1,numofelec);

    %all electrons with higher probabilities
    new = random < pscat;

    %all electrons with lower probabilities
    new2 = random >= pscat;
    rand_v_x = zeros(1,numofelec);
    rand_v_y = zeros(1,numofelec);

   for i = 1:1:numofelec
     r1 = randi([1 numofelec], 1,1);
     r2 = randi([1 numofelec], 1,1);
        rand_v_x(1,i) = v_x(1,r1);
        rand_v_y(1,i) = v_y(1,r2);
   end
        %all electrons with lower probabilities will stay the same
   v_x = v_x.*new2;
   v_y = v_y.*new2;

   rand_v_x=rand_v_x.*new;
   rand_v_y=rand_v_y.*new;

   v_x = v_x+rand_v_x;
   v_y = v_y+rand_v_y;

       rb1 = ( x > 80e-9 & x < 120e-9) & y < 40e-9;
    rb0 = rb1 == 0;
    rb1 = -1 * rb1;

    f = rb1 + rb0;

    v_x = v_x .* f;


    rb1 = ( x > 80e-9 & x < 120e-9) & (y < 41e-9 & y >= 40e-9);
    rb0 = rb1 == 0;
    rb1 = -1 * rb1;

    f = rb1 + rb0;

    v_y = v_y .* f;

    %tempFinalLower = x .* y

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%Dealing with the upper rectangle%%%%%%%
    rb1 = ( x > 80e-9 & x < 120e-9) & y > 60e-9;
    rb0 = rb1 == 0;
    rb1 = -1 * rb1;

    f = rb1 + rb0;

    v_x = v_x .* f;


    rb1 = ( x > 80e-9 & x < 120e-9) & (y >59e-9 & y < 60e-9);
    rb0 = rb1 == 0;
    rb1 = -1 * rb1;
```

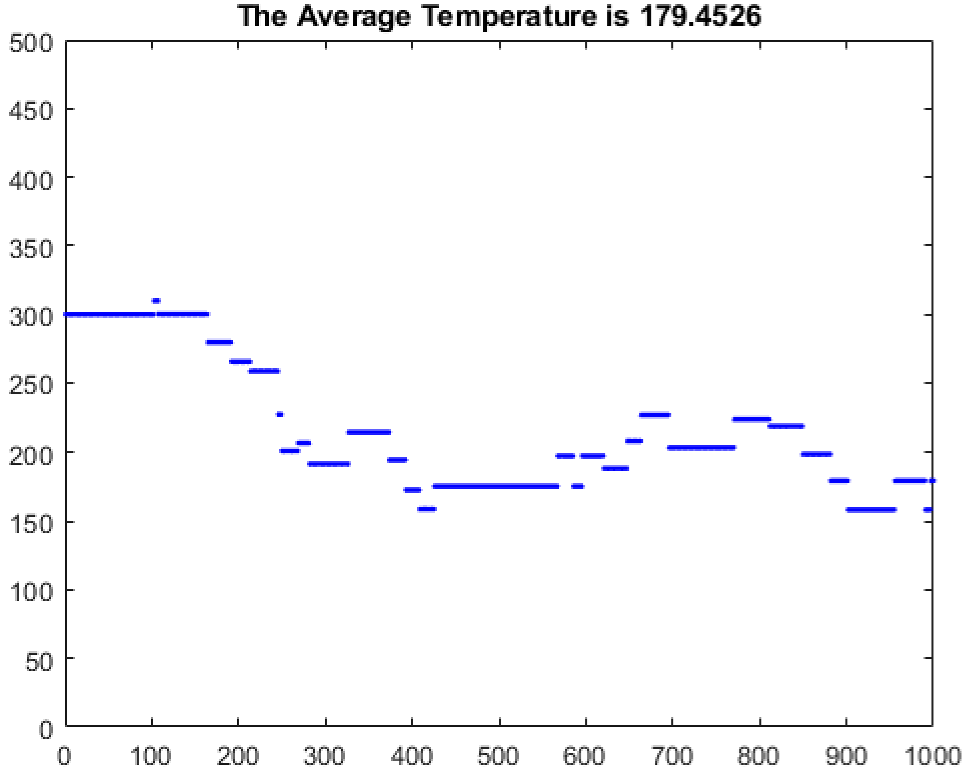
```
   f = rb1 + rb0;

   v_y = v_y .* f;
   %
   dx = v_x*dt*1e-15*5;
   dy = v_y*dt*1e-15*5;

   x = x + dx;
   y = y + dy;

   %if y is greater than 200
   temp = y>=Width*1e-9;
   temp1 = y<Width*1e-9;

   temp = temp*(-1);

   temphigher = temp + temp1;

   v_y = temphigher.*v_y;

    %if y is less than 100
   temp2 = y>=0;
   temp3 = y<0;

   temp3 = temp3*(-1);
   templower = temp3 + temp2;
   v_y = templower.*v_y;



%if x greater than 200
  temp5 = x<200*1e-9;

  x = x .* temp5;

  %if x is less than 0
  temp4 = x< 0;
  temp4 = temp4*200*1e-9;

  %temp4 = temp4*200*1e-9;
  x = x + temp4;

   %average thermal velocity
   v_avg = mean(sqrt((v_x.^2)+(v_y.^2)));
   v_matrix = sqrt((v_x.^2)+(v_y.^2));
   T_avg = (mn*(v_avg^2))/kB;

   T_matrix = (mn*(v_matrix.*v_matrix))/kB;

   %mean free path

   mfp = (10^-15)*(v_avg);
```

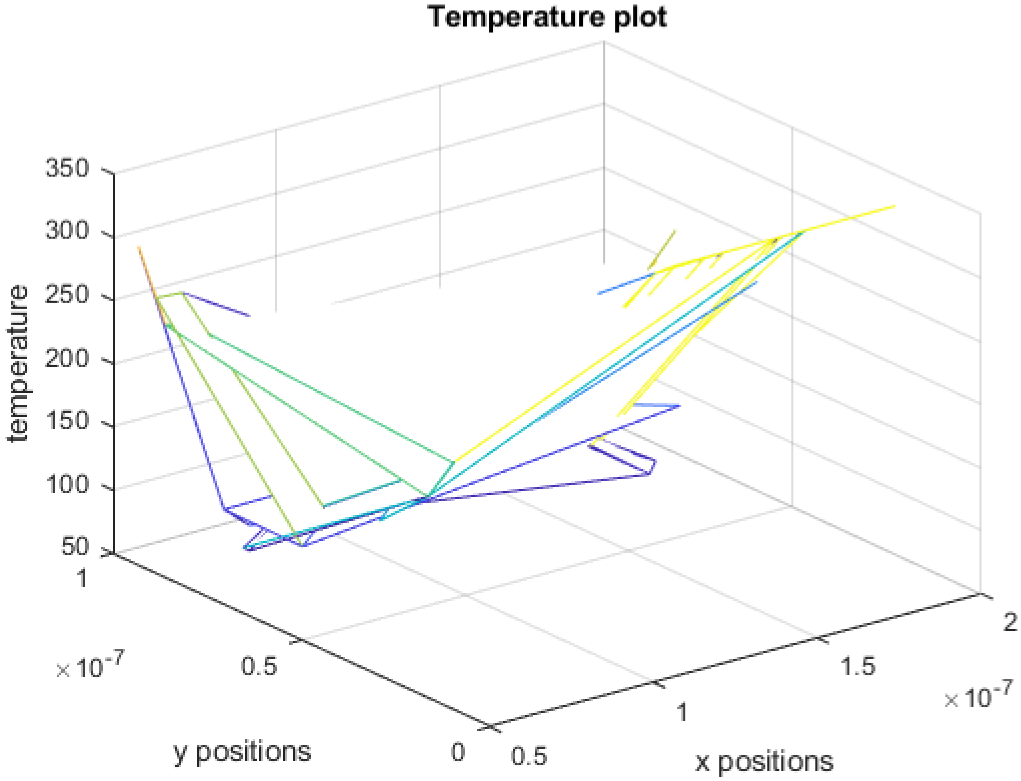
```
    figure(3)
    scatter(x,y,3,colorarray);
    axis([0 200*10^-9 0 100*10^-9])
    rectangle('Position' ,[0.8e-7 0 0.4e-7 0.4e-7]);
    rectangle('Position' ,[0.8e-7 0.6e-7 0.4e-7 0.4e-7]);
    title(['The mean free path is ', num2str(mfp)]);
   % pause(0.000000000000000000000000000000000000000001)
    hold on


    figure(4)
    plot(time,T_avg,'.b')
    title(['The Average Temperature is ', num2str(T_avg)]);
    axis([0 1000 0 500])
    hold on

    [X,Y] = meshgrid (x' , y');
    f1 = scatteredInterpolant(x',y',T_matrix');
    Z = f1(X,Y);
    figure (6);
    mesh(X,Y,Z);
    title('Temperature plot')
    xlabel('x positions')
    ylabel('y positions')
    zlabel('temperature')
    %axis tight;hold on
    %plot3(x,y,tMatrix,'.','MarkerSize',15)


end

%Density plot
elecpos = [x',y'];
Density = hist3(elecpos, 'Nbins',[20,10]);
figure(5)
xlabel('x positions')
ylabel('y positions')
title('Electron Density Plot')
surf(Density)
shading interp
```

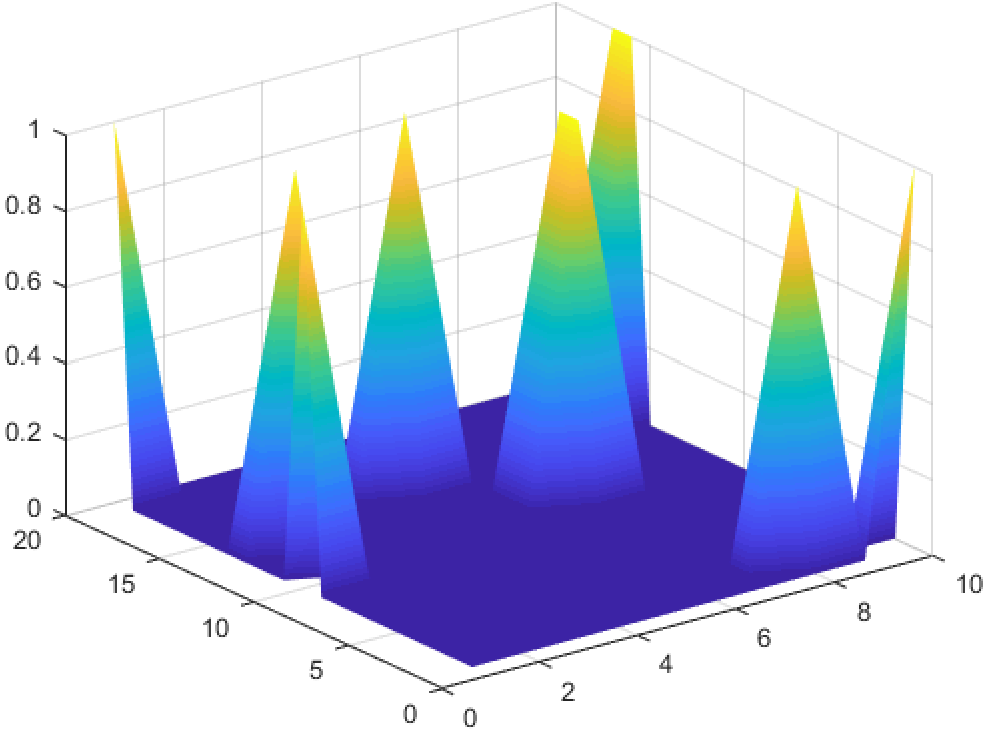

x axis component of v thermal
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^5


y axis component of v thermal
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^5


The mean free path is 1.0228e-10
×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


The Average Temperature is 179.4526
500
450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1000


Temperature plot
temperature
350
300
250
200
150
100
50
y positions
1
0.5
0
×10^-7
x positions
0.5
1
1.5
2
×10^-7


1
0.8
0.6
0.4
0.2
0
20
15
10
5
0
0
2
4
6
8
10

Published with MATLAB® R2018a