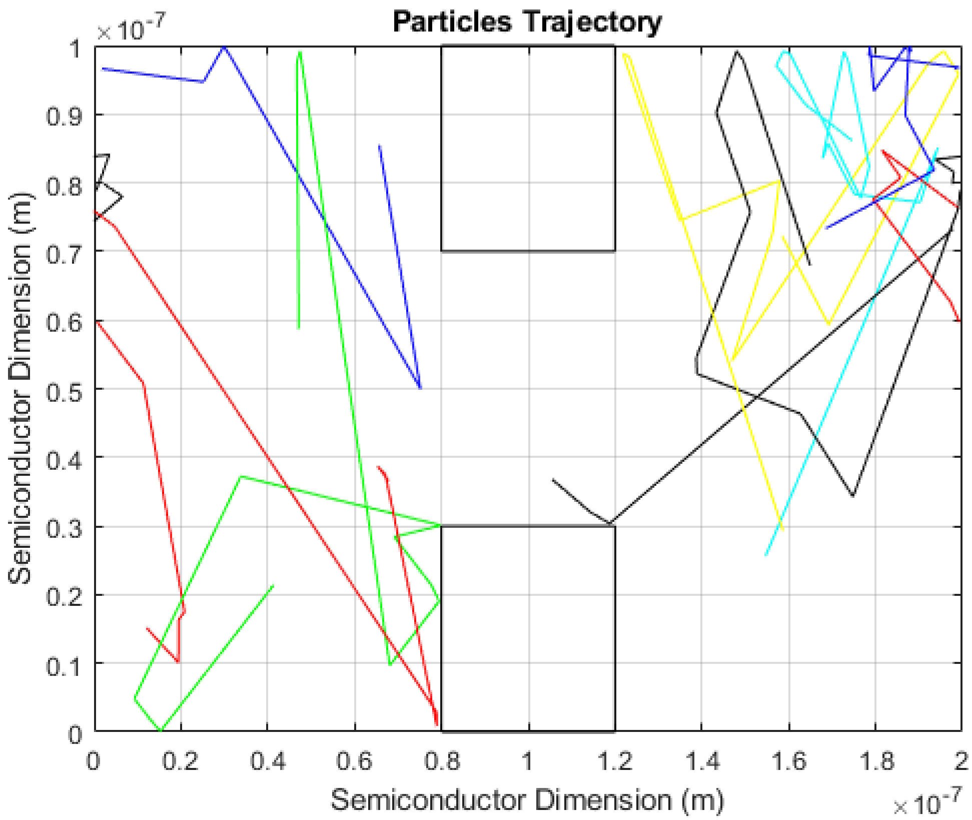
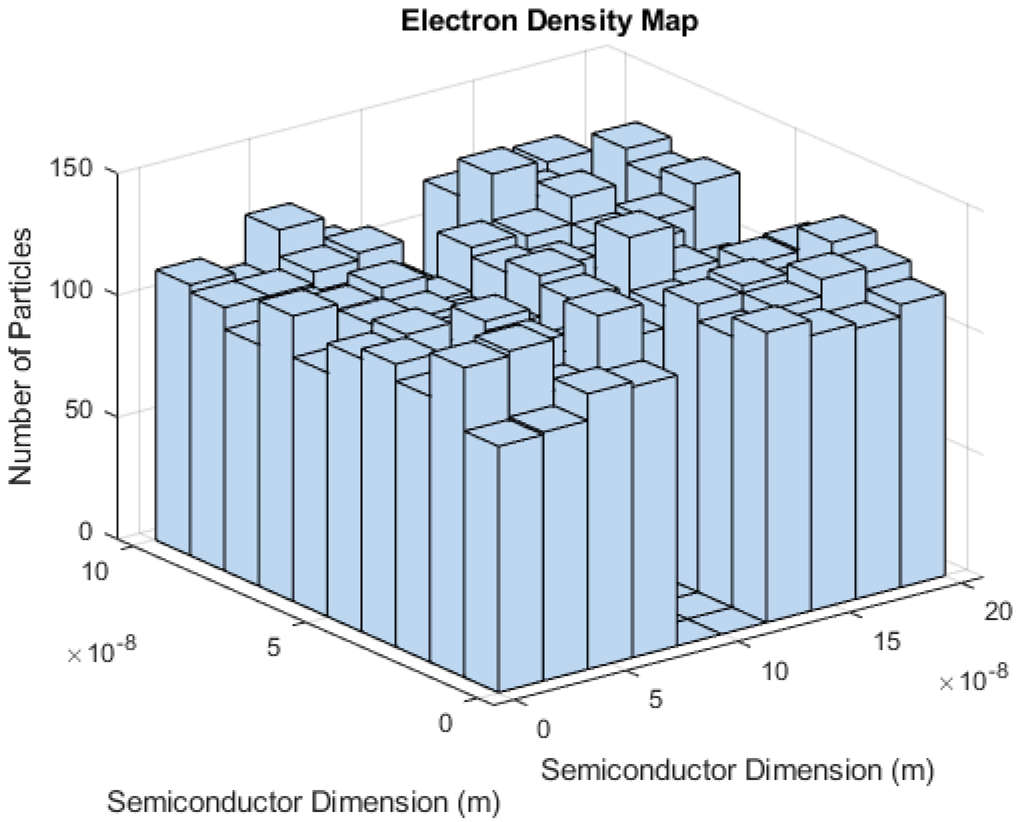
```
function [] = Q3(numOfAtom, numOfStep)
% ELEC4700 - Assignment 1
% Xiaochen Xin 100989338
C.q_0 = 1.60217653e-19;             % electron charge
C.hb = 1.054571596e-34;             % Dirac constant
C.h = C.hb * 2 * pi;                    % Planck constant
C.m_0 = 9.10938215e-31;             % electron mass(kg)
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 s²
C.am = 1.66053892e-27;

mn = 0.26*C.m_0; %Effective Mass
l = 200e-9; %Length of area (m)
w = 100e-9; %Width of area (m)
T = 300; %Kelvin

vth = sqrt(C.kb*T/mn);%thermal velocity(velocity at which the
 particles are travelling at)

tmn = 0.2e-12; %mean time between collision (s)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
rvx = randn(numOfAtom,1)*sqrt(C.kb*T/mn); %random vx
rvy = randn(numOfAtom,1)*sqrt(C.kb*T/mn); %random vY

v   = sqrt(rvx.^2+rvy.^2);
% figure (1)
% plot(hist(v,100))
% title ("Particle Velocity Distribution")

xr = 200e-9.*rand(numOfAtom,1); %x of 100 random locations
yr = 100e-9.*rand(numOfAtom,1); %y of 100 random locations

%Move the particles initially within the blocks outside of the blocks
xr(xr > 0.8e-7 & xr < 1e-7 & yr > 0.7e-7) = xr(xr > 0.8e-7 & xr < 1e-7
 & yr > 0.7e-7) - 0.4e-7;
xr(xr > 1e-7 & xr < 1.2e-7 & yr > 0.7e-7) = xr(xr > 1e-7 & xr < 1.2e-7
 & yr > 0.7e-7)  + 0.4e-7;
xr(xr > 0.8e-7 & xr < 1e-7 & yr < 0.3e-7) = xr(xr > 0.8e-7 & xr < 1e-7
 & yr < 0.3e-7) - 0.4e-7;
xr(xr > 1e-7 & xr < 1.2e-7 & yr < 0.3e-7) = xr(xr > 1e-7 & xr < 1.2e-7
 & yr < 0.3e-7)  + 0.4e-7;

%Define two arrays store the previous locations
xrp = xr;
yrp = yr;

MFPx = xr;
```

```
MFPy = yr;
MFP  = zeros(numOfAtom,1);
MTBC = zeros(numOfAtom,1);
scatter_number = zeros(numOfAtom,1);

t = 1.5e-14; %time interval that captures line
xd = rvx*t; %displacement in x during one time interval
yd = rvy*t; %displacement in y during one time interval

Pscat = 1-exp(-t/tmn); % Probability that a particle scatters

for p = 1:1:numOfStep
    scatter_prob = rand(numOfAtom,1);
%%%%%%%%%Calculate Mean Free Path%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    MFP(scatter_prob<Pscat) = MFP(scatter_prob<Pscat) +
 sqrt((xd(scatter_prob<Pscat)-
MFPx(scatter_prob<Pscat)).^2+(yd(scatter_prob<Pscat)-
MFPy(scatter_prob<Pscat)).^2);
    MTBC(scatter_prob<Pscat) =  MTBC(scatter_prob<Pscat) +
 sqrt((xd(scatter_prob<Pscat)-
MFPx(scatter_prob<Pscat)).^2+(yd(scatter_prob<Pscat)-
MFPy(scatter_prob<Pscat)).^2)./v(scatter_prob<Pscat);
    scatter_number(scatter_prob<Pscat) =
 scatter_number(scatter_prob<Pscat) + 1;
    MFPx(scatter_prob<Pscat) = xr(scatter_prob<Pscat);
    MFPy(scatter_prob<Pscat) = yr(scatter_prob<Pscat);
%%%%%%%%%Calculate average temperature of all particles%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
    v = sqrt(rvx.^2+rvy.^2);
    TParticles = (0.5*mn*v.^2)/(C.kb); %Tempearture of individual
 particles
    Tave (p) = sum(TParticles)/numOfAtom; %Average temperature of all
 particles
%     figure (3)
%     plot (Tave)
%     xlabel("Number of steps (1.5e-14s/step)")
%     ylabel("Temperature (K)")
%     ylim ([0, 500])
%     xlim ([0, numOfStep])
%     title("average temperature over time")
%%%%%%%%%%%Calculate Mean Time Between Collision%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
    rvx_new = randn(numOfAtom,1)*sqrt(C.kb*T/mn); %new random vx
    rvy_new = randn(numOfAtom,1)*sqrt(C.kb*T/mn); %new random vY
    rvx(scatter_prob<Pscat) = rvx_new(scatter_prob<Pscat);
    rvy(scatter_prob<Pscat) = rvy_new(scatter_prob<Pscat);


    xd = rvx*t; %displacement in x during one time interval
    yd= rvy*t; %displacement in y during one time interval

    %Proceed to next step
    xr = xr+xd;
    yr = yr+yd;
```

```
   %Define the left&right wrap-around
   xrp(xr >2e-7) = - (2e-7 - xrp(xr >2e-7));% changing previous point
 to prevent line drawn across canvas
   xr(xr > 2e-7) = xr(xr > 2e-7)-(2e-7);
   xrp(xr <0)    = 2e-7 - xrp(xr <0);% changing previous point to
 prevent line drawn across canvas
   xr(xr < 0)    = xr(xr < 0 )+(2e-7);


   %Define the specular top&bottom
   rvy(yr > 1e-7) = - rvy(yr > 1e-7);
   yr(yr > 1e-7) = (1e-7)-(yr(yr > 1e-7)-(1e-7));
   rvy(yr < 0) = -rvy(yr < 0 );
   yr(yr < 0) = -yr(yr < 0 );

   %Define upper block
   %Define the left edge of both blocks
   rvx(xrp < 0.8e-7 & xr >= 0.8e-7 & (yr >= 0.7e-7 | yr < 0.3e-7)) =
 - rvx(xrp < 0.8e-7 & xr >= 0.8e-7 & (yr >= 0.7e-7 | yr < 0.3e-7));
   xr(xrp < 0.8e-7 & xr >= 0.8e-7 & (yr >= 0.7e-7 | yr < 0.3e-7))
 = 0.8e-7 - (xr(xrp < 0.8e-7 & xr >= 0.8e-7 & (yr >= 0.7e-7 | yr <
 0.3e-7)) - 0.8e-7);
   %Define the right edge of both blocks
   rvx(xrp > 1.2e-7 & xr <= 1.2e-7 & (yr >= 0.7e-7 | yr < 0.3e-7)) =
 - rvx(xrp > 1.2e-7 & xr <= 1.2e-7 & (yr >= 0.7e-7 | yr < 0.3e-7));
   xr(xrp > 1.2e-7 & xr <= 1.2e-7 & (yr >= 0.7e-7 | yr < 0.3e-7)) =
 1.2e-7 + (1.2e-7 -  xr(xrp > 1.2e-7 & xr <= 1.2e-7 & (yr >= 0.7e-7 |
 yr < 0.3e-7)));
   %Define the botton edge of top block
   rvy(yrp < 0.7e-7 & yr >= 0.7e-7 & xr >= 0.8e-7 & xr <= 1.2e-7) = -
rvy(yrp < 0.7e-7 & yr >= 0.7e-7 & xr >= 0.8e-7 & xr <= 1.2e-7);
   yr(yrp < 0.7e-7 & yr >= 0.7e-7 & xr >= 0.8e-7 & xr <= 1.2e-7)
 = 0.7e-7 - (yr(yrp < 0.7e-7 & yr >= 0.7e-7 & xr >= 0.8e-7 & xr <=
 1.2e-7) - 0.7e-7);
   %Define top edge of bottom block
   rvy(yrp > 0.3e-7 & yr <= 0.3e-7 & xr >= 0.8e-7 & xr <= 1.2e-7) = -
rvy(yrp > 0.3e-7 & yr <= 0.3e-7 & xr >= 0.8e-7 & xr <= 1.2e-7);
   yr(yrp > 0.3e-7 & yr <= 0.3e-7 & xr >= 0.8e-7 & xr <= 1.2e-7) =
 0.3e-7 + (0.3e-7 - yr(yrp > 0.3e-7 & yr <= 0.3e-7 & xr >= 0.8e-7 & xr
 <= 1.2e-7));

   figure (2)
   rectangle('Position', [0.8e-7 0 0.4e-7 0.3e-7])
   rectangle('Position', [0.8e-7 0.7e-7 0.4e-7 0.3e-7])
   plot([xrp(1), xr(1)], [yrp(1), yr(1)], 'r')
   plot([xrp(2), xr(2)], [yrp(2), yr(2)], 'b')
   plot([xrp(3), xr(3)], [yrp(3), yr(3)], 'k')
   plot([xrp(4), xr(4)], [yrp(4), yr(4)], 'g')
   plot([xrp(5), xr(5)], [yrp(5), yr(5)], 'y')
   plot([xrp(6), xr(6)], [yrp(6), yr(6)], 'c')
   xlabel("Semiconductor Dimension (m)")
   ylabel("Semiconductor Dimension (m)")
   title ("Particles Trajectory")
```

```
    xlim ([0, 2e-7])
    ylim([0,1e-7])
    grid on
    hold on
    pause(0.05)

    xrp = xr;
    yrp = yr;
end
%Display Overall MFP
overallMFP = sum(MFP./scatter_number)/numOfAtom;
overallMTBC = sum(MTBC./scatter_number)/numOfAtom;

%Calculate electron density map
xy = [xr,yr];
figure (4)
hist3(xy)
title ('Electron Density Map')
xlabel("Semiconductor Dimension (m)")
ylabel("Semiconductor Dimension (m)")
zlabel("Number of Particles")

%Calculate density map

end
```





Published with MATLAB® R2018b