Table of Contents

Question 1

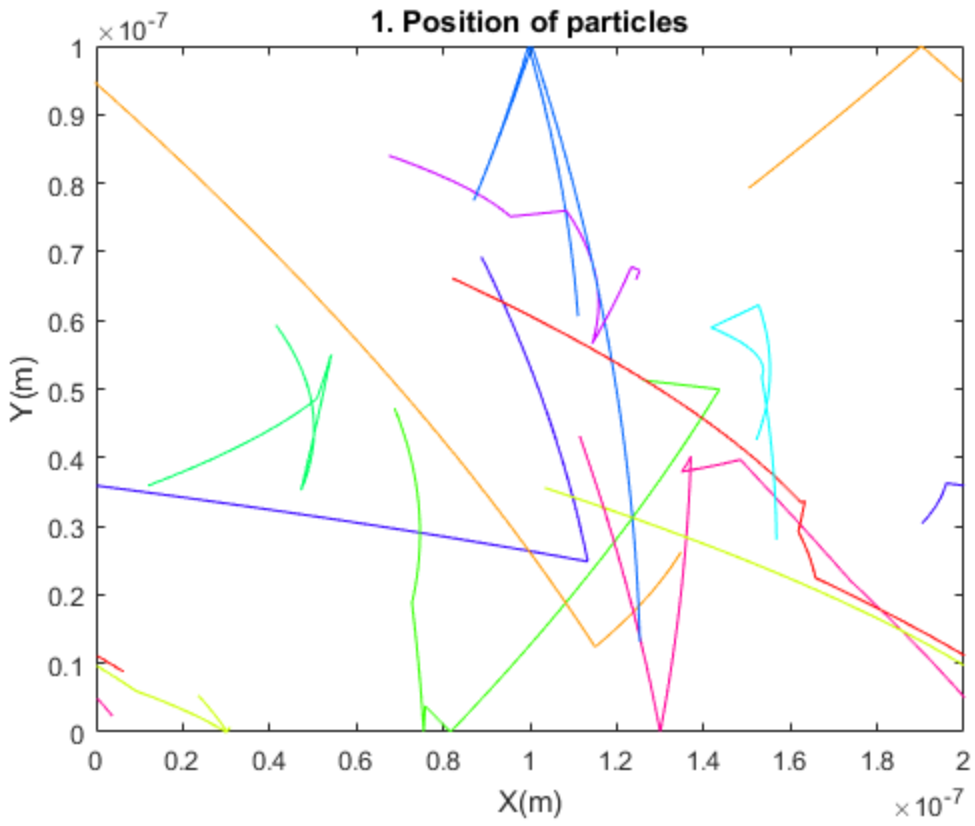
```
% a constant .1 V is applied accross the 200nm by 100nm space.

% the current density formula is:
% Current Density = average drift velocity*(charge density*charge of
 e-);
% Current = density * width
% the plot shows a current density that grows as the system starts up,
 and levels off
% this is expected since it takes time for electron to accelerate so
 the
% mean free path grows.  As it starts to level off due to the
% collisions, the drift velocity will also level off, causing the
 current
% density to cap at a value.

%variables you can edit
num_e = 1000;
x_dim = 200*10^-9;
y_dim = 100e-9;
voltage = 0.1;
eperp=1e19/(num_e/(x_dim*y_dim));

%electric field = Voltage over distance
efield = voltage/x_dim
%force = electric field * charge of electron
force = efield*1.60217662*10^-19
%accel = force/mass
accel=force/(0.26*9.1093*10^-31)

%colours for plot
col=hsv(10);

Temp_arr=[300];
tau = zeros(1,num_e);
Tau=0;
mfp=0;
count=0;

close all

%get initial positions and velocities
[x_arr,y_arr, vx_arr,vy_arr] = gen_e(num_e,x_dim,y_dim,2);
```

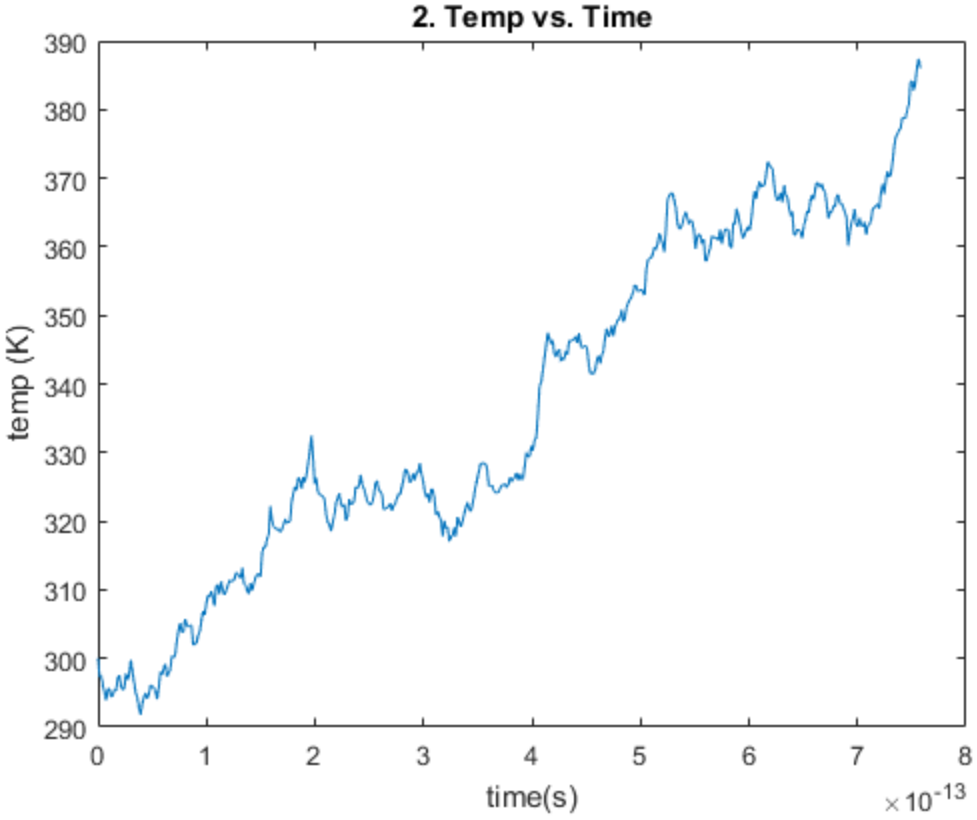
```
v = sqrt(vx_arr.*vx_arr + vy_arr.*vy_arr);

vth =132.2e3;

t=0 ;
curr_den=0;
t_step = max(x_dim,y_dim)/(1000*vth);
Tstop=500*t_step;
time_arr=zeros(1,10000);
for i=1:length(time_arr)
    time_arr(i)=(i-1)*t_step;
end

%scatter probablity
P_scat=1-exp(-t_step/(.2e-12));


while t< Tstop

    %calculate new velocity if scatter, update mfp, time between
 collisions
    for q= 1:length(x_arr)
        tau(q)=tau(q)+t_step;
        if rand()<P_scat
            Tau=[Tau,tau(q)];
            mfp=[mfp,tau(q)*sqrt(vx_arr(q)^2+vy_arr(q)^2)];
            tau(q)=0;
            vx_arr(q)=132.2e3*randn();
            vy_arr(q)=132.2e3*randn();

        end
    end

    %calculate temp
    Temp_arr = [Temp_arr,(1/2)/
(1.3806e-23)*9.109e-31*.26*(mean(vx_arr.^2)+mean(vy_arr.^2))];



    %add the time step to the position
    xp_arr=x_arr;
    xg_arr=x_arr;
    yp_arr=y_arr;
    yg_arr=y_arr;
    x_arr=x_arr+vx_arr*t_step;
    y_arr=y_arr+vy_arr*t_step;
    %add acceleration
    vx_arr=vx_arr-accel*t_step;



    %check to see if anything is out of bounds
    for q=1:num_e
```

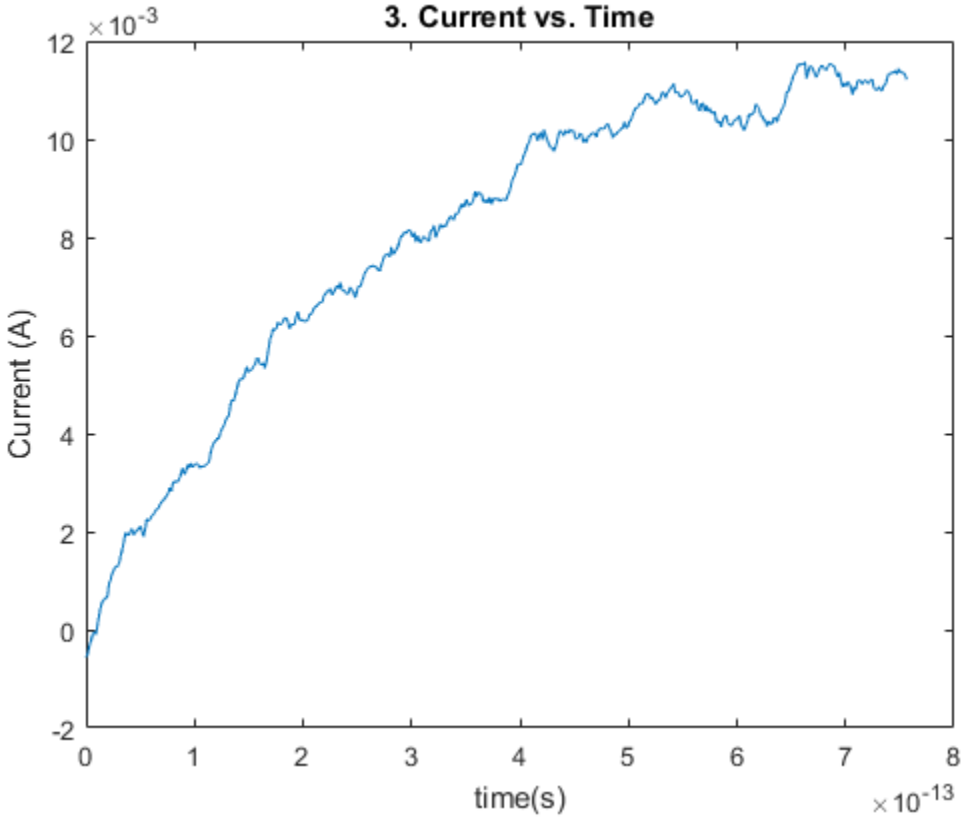
```
        if x_arr(q)<0
            x_arr(q)=x_arr(q)+x_dim;
            xg_arr(q)=x_dim;
        end
        if x_arr(q) > x_dim
            x_arr(q)=x_arr(q)-x_dim;
            xg_arr(q)=0;
        end
        if y_arr(q)>y_dim
            vy_arr(q)=-vy_arr(q);
            y_arr(q)=2*y_dim-y_arr(q);
        end
        if y_arr(q)<0
            vy_arr(q)=-vy_arr(q);
            y_arr(q)=abs(y_arr(q));
        end
    end

    %position plot
    figure(1)
    xlabel('X(m)')
    ylabel('Y(m)')
    title('1. Position of particles')
    xlim([0 x_dim])
    ylim([0 y_dim])
    pause(.01)
    for q=1:10
        plot([xg_arr(q);x_arr(q)],
[yg_arr(q);y_arr(q)],'color',col(q,:))
        hold on
    end



    %temperature plot

    figure(2)
    plot(time_arr(1:length(Temp_arr)),Temp_arr)
    xlabel('time(s)')
    ylabel('temp (K)')
    title('2. Temp vs. Time')

    count=count+1;
    t=t+t_step;
    %find current density

    driftv=mean(vx_arr);
    charge_den=1e19;
    charge=1.60217*10^-19;
    curr_den(count)=driftv*charge_den*charge;

    figure(3)
    plot(time_arr(1:length(curr_den)),-curr_den*y_dim)
```

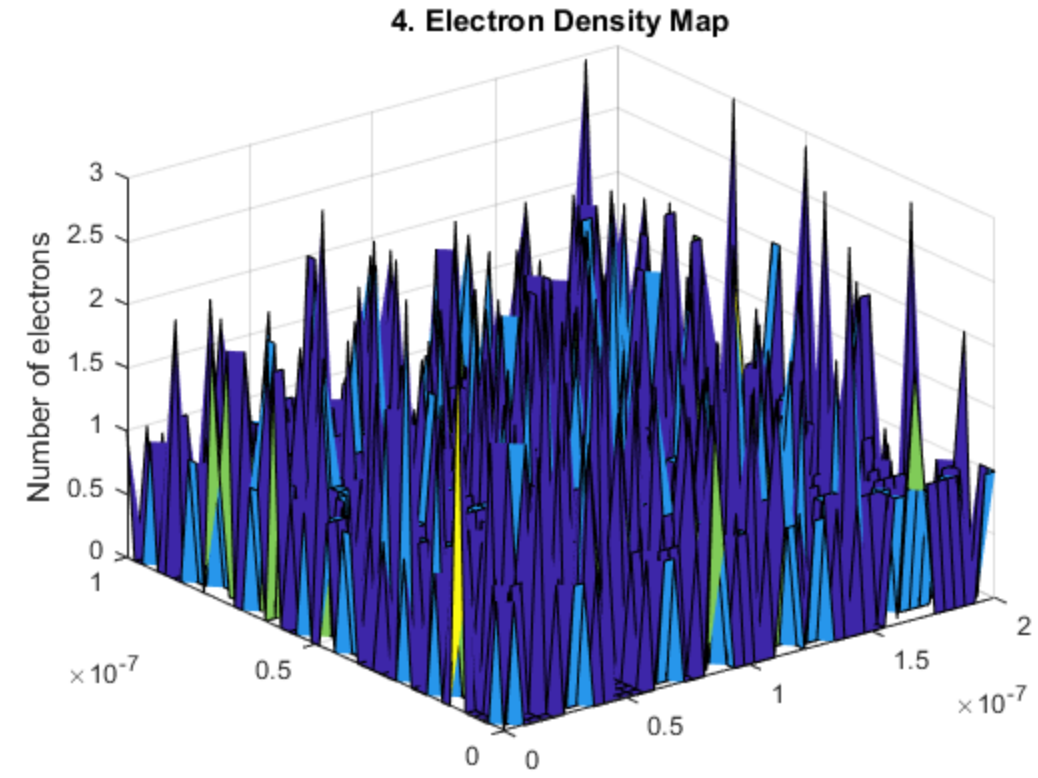
```
    xlabel('time(s)')
    ylabel('Current (A)')
    title('3. Current vs. Time')

end
%make the density maps
p=zeros(50);
vx=zeros(50);
vy=zeros(50);
temp=zeros(50);

for q=1:50
    for w=1:50
        for n=1:num_e
            if
 x_arr(n)>=(((q-1)*x_dim/50))&&(x_arr(n)<(q*x_dim/50))&&(y_arr(n)>=(w-1)*y_dim/50
                p(w,q)=p(w,q)+1;
                vx(w,q)=vx(w,q)+vx_arr(n)^2;
                vy(w,q)=vy(w,q)+vy_arr(n)^2;
            end
        end
        if p(w,q)==0
            temp(w,q)=0;

        else
            temp(w,q)=0.26*9.109e-31*(vx(w,q)+vy(w,q))/p(w,q)/
(1.3806e-23);
        end

    end
end

figure(4)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),p)
title('4. Electron Density Map')
zlabel('Number of electrons')

figure (5)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),temp)
title('5. Temperature Density Map')
zlabel('Temperature (K)')


efield =

      500000


force =

   8.0109e-14


accel =
```

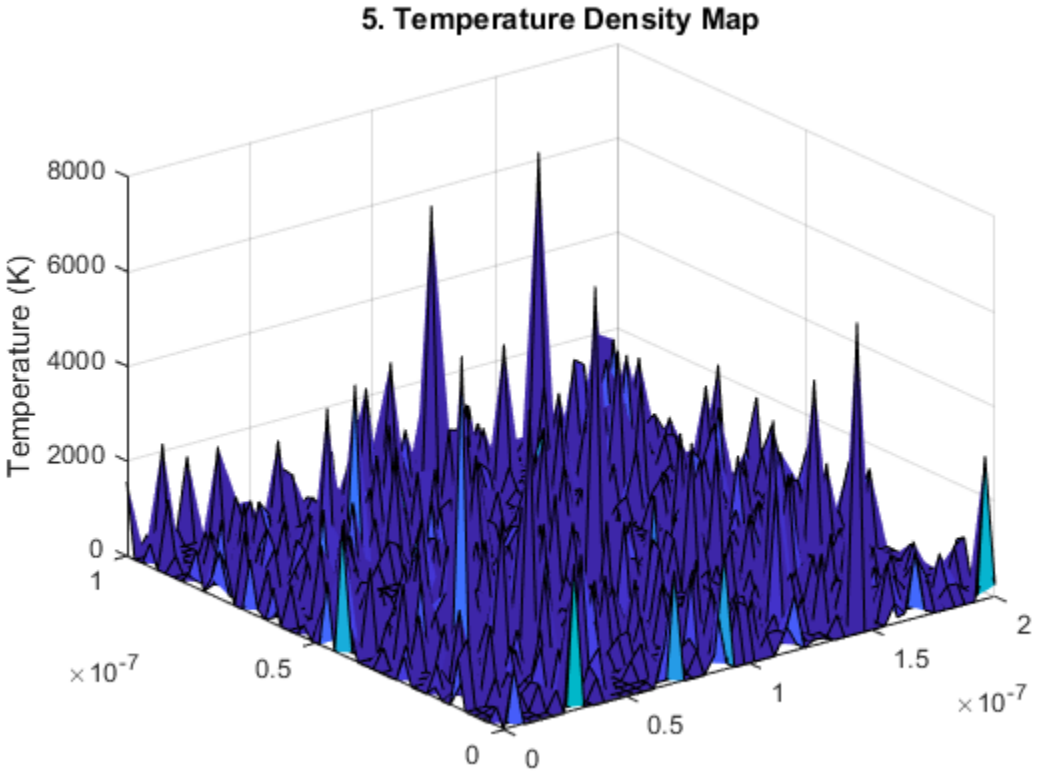
3.3824e+17


1. Position of particles
×10^-7
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Y(m)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
X(m)
×10^-7


2. Temp vs. Time
temp (K)
time(s)
×10^-13
390
380
370
360
350
340
330
320
310
300
290
0
1
2
3
4
5
6
7
8


3. Current vs. Time
×10^-3
Current (A)
time(s)
×10^-13
12
10
8
6
4
2
0
-2
0
1
2
3
4
5
6
7
8


4. Electron Density Map
Number of electrons
3
2.5
2
1.5
1
0.5
0
1
×10^-7
0.5
0
0
0.5
1
1.5
2
×10^-7

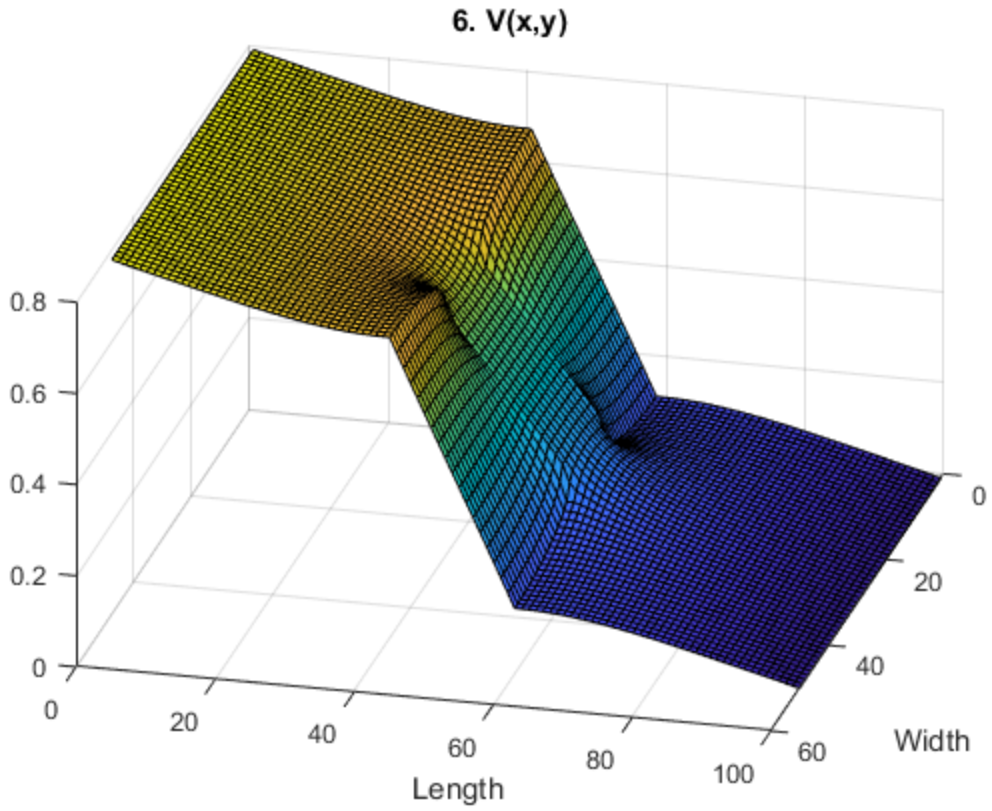

5. Temperature Density Map
Temperature (K)
8000
6000
4000
2000
0
1
×10^-7
0.5
0
0
0.5
1
1.5
2
×10^-7

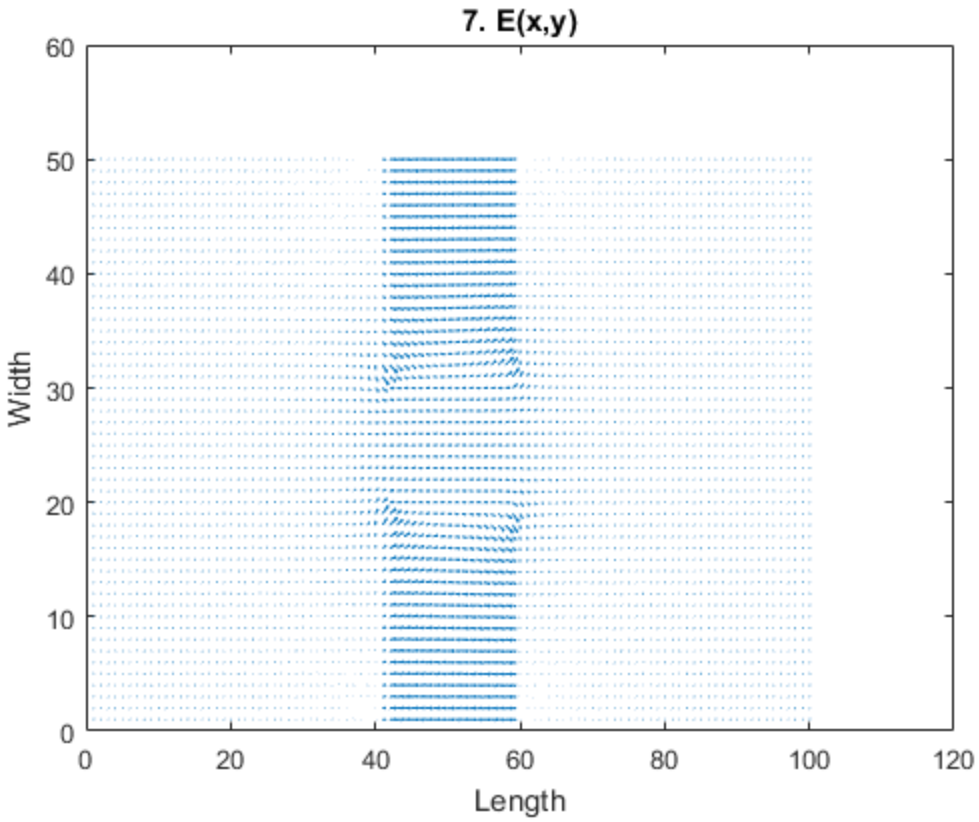
Question 2

```
close all


nx=100;
ny=50;

res1=1;
res2=1e-3;
left=round(2/5*nx);
right=round(3/5*nx);
bottom=round(0.4*ny);
top=round(0.6*ny);
plotx=1;



g=sparse(nx*ny);
b=zeros(1,nx*ny);

sig1=res1;
sig2=res2;

% box = [ left right bottom top] of center high conductive region
box=[left right bottom top];

sigma=zeros(nx,ny);
for i = 1:nx
    for j=1:ny

        if i > box(1) && i < box(2) &&j >box(4) %upper box - low cond
            sigma(i,j)=sig2;
        elseif i > box(1) && i < box(2) &&j <box(3) %lower box - low
 cond
            sigma(i,j)=sig2;
        else %high cond
            sigma(i,j)=sig1;
        end
    end
end

for i = 1:nx
    for j=1:ny
        n=j+(i-1)*ny;
        if i==1  %left
            g(n,:)=0;
            g(n,n)=1;
            b(n)=0.8;
        elseif i==nx %right
            g(n,:)=0;
            g(n,n)=1;
```

```
        elseif j==1 %bottom
            up=(sigma(i,j)+sigma(i,j+1))/2;
            left=(sigma(i,j)+sigma(i-1,j))/2;
            right=(sigma(i,j)+sigma(i+1,j))/2;

            g(n,n)=-(up+left+right);
            g(n,n+1)=up;
            g(n,n-ny)=left;
            g(n,n+ny)=right;

        elseif j==ny %top
            %low conductivity
            down=(sigma(i,j)+sigma(i,j-1))/2;
            left=(sigma(i,j)+sigma(i-1,j))/2;
            right=(sigma(i,j)+sigma(i+1,j))/2;

            g(n,n)=-(up+left+right);
            g(n,n+ny)=right;
            g(n,n-1)=down;
            g(n,n-ny)=left;


        else %bulk node

            down=(sigma(i,j)+sigma(i,j-1))/2;
            left=(sigma(i,j)+sigma(i-1,j))/2;
            right=(sigma(i,j)+sigma(i+1,j))/2;
            up=(sigma(i,j)+sigma(i,j+1))/2;

            g(n,n)=-(up+down+right+left);
            g(n,n+1)=up;
            g(n,n-1)=down;
            g(n,n+ny)=right;
            g(n,n-ny)=left;

        end
    end
end



E=g\b';

d=zeros(nx,ny);
for i = 1:nx
    for j=1:ny
       n=j+(i-1)*ny;
       d(i,j)=E(n);

    end
end
if plotx
    figure(1)
    surf(d) %V(x,y)
```

```
    xlabel('Width')
    ylabel('Length')
    title('6. V(x,y)')
    view(-256,36)
end

%make sigma(x,y) graph


[ex,ey]=gradient(d);
ex=-ex;
ey=-ey;
if plotx
    figure(3) %E(x,y)
    quiver(ey',ex')
    xlabel('Length')
    ylabel('Width')
    title('7. E(x,y)')
end
```





Question 3

```
% This section of code models the flow of electrons in a 200nm by
 100nm box
% with two rectangle boundaries.  These boundaries can be specular or
% diffusive (currently set to diffusive).  Every time a particle
 strikes a
% boundary, it gains a new velocity.  This code also produces an
 electron
% density map, and a temperature density map


% the density plot has electrons pooling on the top and bottom of the
 right
% side of the boundary.  Since current travels to the right the
 electrons
% travel to the left. As they reach the boundary, electron in the
 vertical
% center will travel through the barrier and follow the boundary
 condition
% to appear on the right side again. electrons near the top and bottom
 will
% bounce off the boundary, and will be pushed back to the boundary by
 the
% electric field.  Electrons will be caught in this loop until the
```

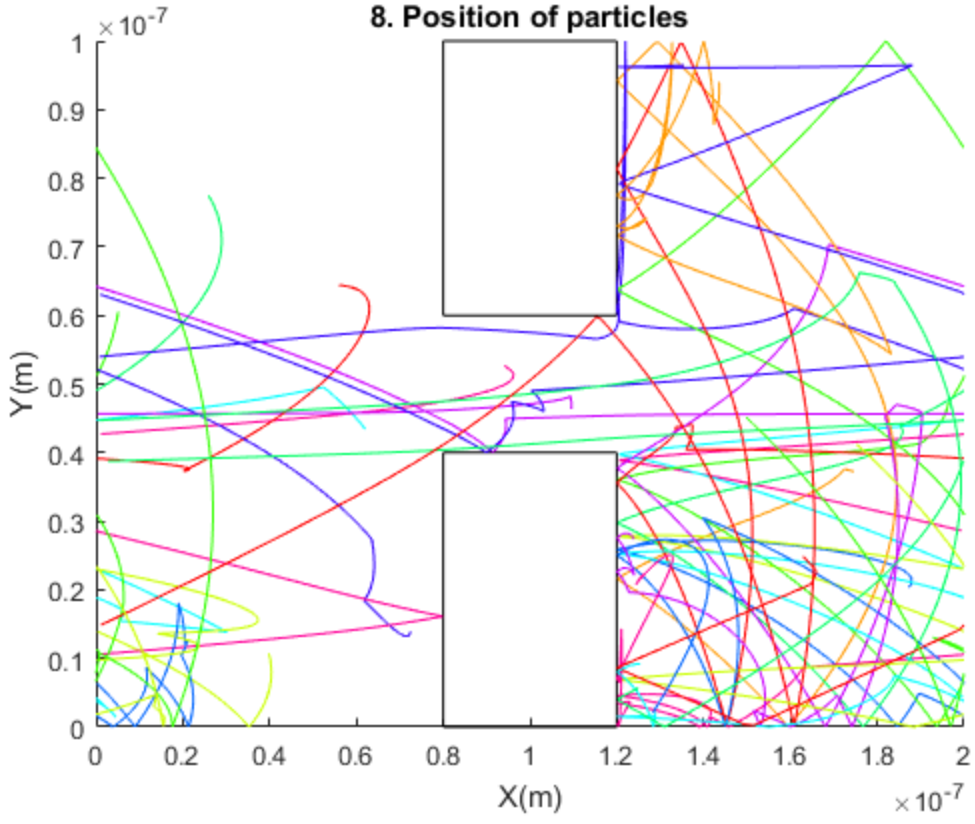
```
% rescatter and gain a large enough positive or negative velocity.
 This
% causes the majority of electron to be caught here, which is seen in
 the
% density graph


% The next step to make this simulation more accurate would be to
 remove
% the periodic boundary condition and instead have a constant flow of
 e-
% from the right and have electrons dissapear to the left.  This makes
 more
% sense because you would have a steady supply of electrons from a
 source
% into this device

close all


%variables you can edit
num_e = 1000;
x_dim = 200*10^-9;
y_dim = 100e-9;
retherm=0; %rethermalize variable.  1 to activate, 0 to deactivate

col=hsv(10);
Temp_arr=[300];
tau = zeros(1,num_e);
Tau=0;
mfp=0;
count=0;

hold off

[x_arr,y_arr, vx_arr,vy_arr] = gen_e(num_e,x_dim,y_dim,3);


vth =132.2e3;
%vth in m/s

t=0 ;
t_step = max(x_dim,y_dim)/(1000*vth);
Tstop=1000*t_step;
time_arr=zeros(1,10000);
for i=1:length(time_arr)
    time_arr(i)=(i-1)*t_step;
end

P_scat=1-exp(-t_step/(.2e-12));
hold off

%define boundary outline in figure
figure(1)
```

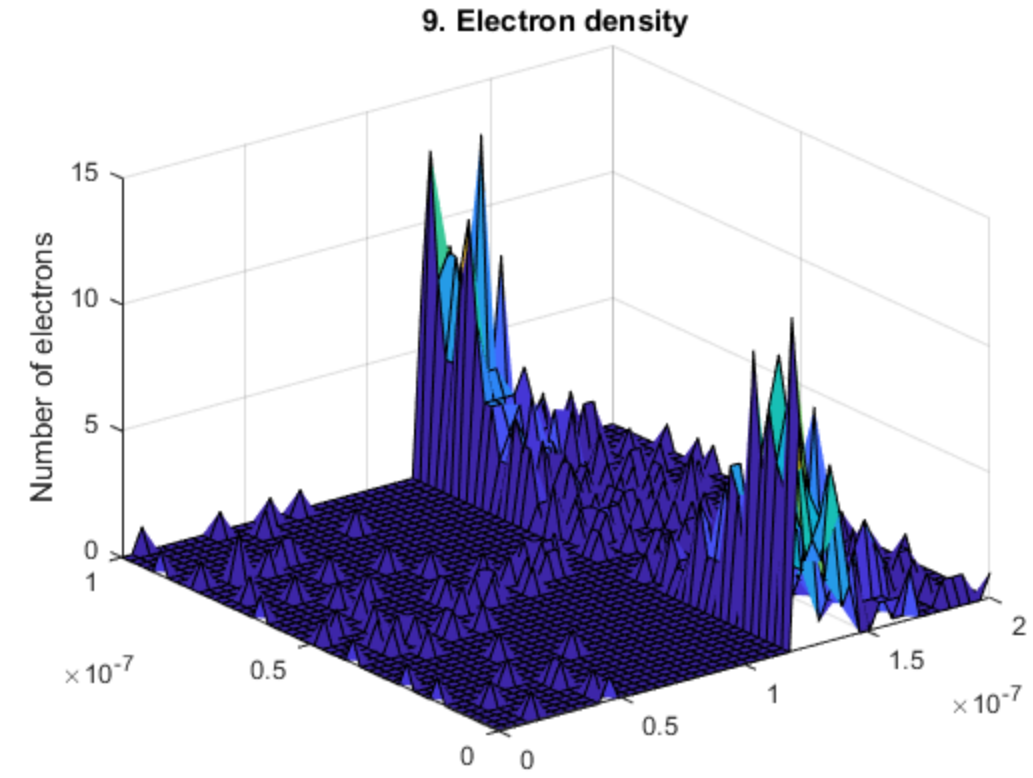
```
hold on
rectangle('position',[0.4*x_dim,0,0.2*x_dim,0.4*y_dim])
rectangle('position',[0.4*x_dim,0.6*y_dim,0.2*x_dim,0.4*y_dim])

while t< Tstop

    %calculate new velocity if scatter
    for q= 1:length(x_arr)
        tau(q)=tau(q)+t_step;
        if rand()<P_scat
            Tau=[Tau,tau(q)];
            mfp=[mfp,tau(q)*sqrt(vx_arr(q)^2+vy_arr(q)^2)];
            tau(q)=0;
            vx_arr(q)=132.2e3*randn();
            vy_arr(q)=132.2e3*randn();

        end
    end

    %find acceleration due to electric field
     for q=1:100
         for w=1:50
             for n=1:num_e
                 if
 x_arr(n)>=(((q-1)*x_dim/100))&&(x_arr(n)<(q*x_dim/100))&&(y_arr(n)>=(w-1)*y_dim/5
                     vx_arr(n)=vx_arr(n)-
(ey(q,w)/10^-9*1.60217662*10^-19/(0.26*9.1093*10^-31)*t_step);

 vy_arr(n)=vy_arr(n)+(ex(q,w)/10^-9*1.60217662*10^-19/
(0.26*9.1093*10^-31)*t_step);
                 end
             end
         end
     end



    %calculate temp

    Temp_arr = [Temp_arr,(1/2)/
(1.3806e-23)*9.109e-31*.26*(mean(vx_arr.^2)+mean(vy_arr.^2))];

    %add the time step to the position
    xp_arr=x_arr;
    xg_arr=x_arr;
    yp_arr=y_arr;
    yg_arr=y_arr;
    x_arr=x_arr+vx_arr*t_step;
    y_arr=y_arr+vy_arr*t_step;


    %check to see if anything is out of bounds
    for q=1:num_e
       if x_arr(q)<0
```

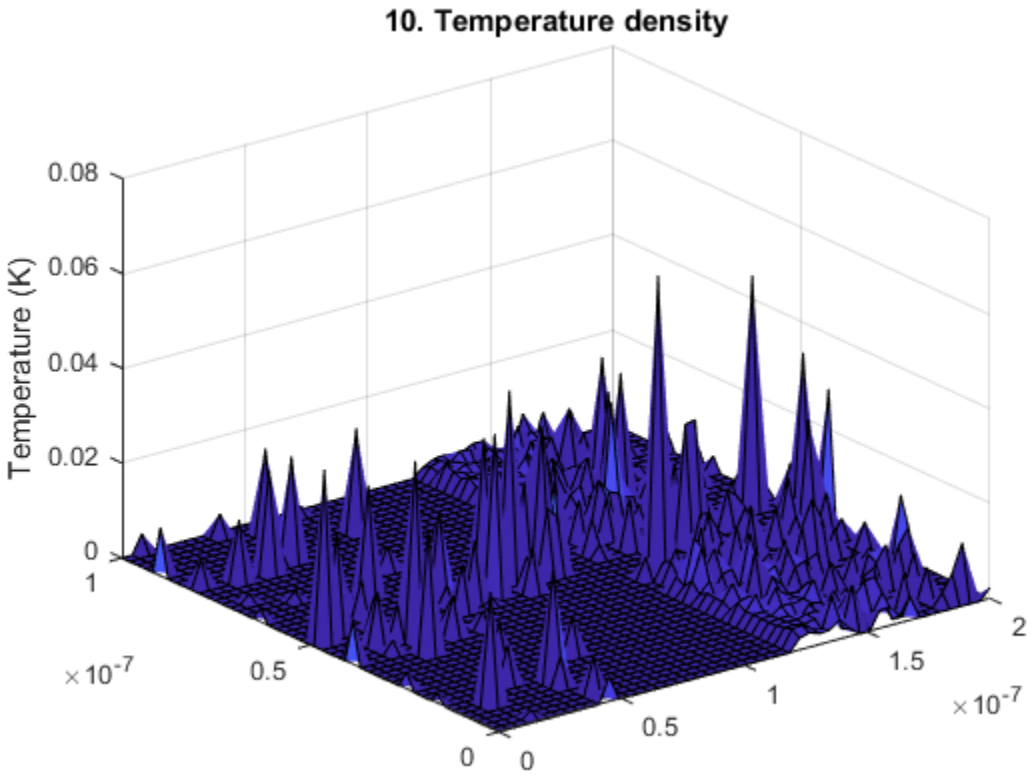
```
            x_arr(q)=x_arr(q)+x_dim;
            xg_arr(q)=x_dim;
        end
        if x_arr(q) > x_dim
            x_arr(q)=x_arr(q)-x_dim;
            xg_arr(q)=0;
        end
        if y_arr(q)>y_dim
            vy_arr(q)=-vy_arr(q);
            y_arr(q)=2*y_dim-y_arr(q);
        end
        if y_arr(q)<0
            vy_arr(q)=-vy_arr(q);
            y_arr(q)=abs(y_arr(q));
        end

        %bot box boundary
        if y_arr(q)<0.4*y_dim && x_arr(q)>0.4*x_dim &&
x_arr(q)<0.6*x_dim
            if y_arr(q)<0.4*y_dim && yp_arr(q)>0.4*y_dim
                y_arr(q)=abs(y_arr(q)-0.4*y_dim)+0.4*y_dim;
                if retherm
                    vy_arr(q)=(132.2e3)*abs(randn(1));
                    vx_arr(q)=132.2e3*randn(1);
                else
                    vy_arr(q)= -vy_arr(q);
                end
            end
            if x_arr(q)>0.4*x_dim && xp_arr(q)<0.4*x_dim
                x_arr(q)=0.4*x_dim-abs(x_arr(q)-0.4*x_dim);
                if retherm
                    vx_arr(q)= -(132.2e3)*abs(randn(1));
                    vy_arr(q)=132.2e3*randn(1);

                else
                    vx_arr(q)= -vx_arr(q);
                end

            end
            if x_arr(q)<0.6*x_dim && xp_arr(q)>0.6*x_dim
                x_arr(q)=abs(x_arr(q)-0.6*x_dim)+0.6*x_dim;
                if retherm
                    vx_arr(q)=(132.2e3)*abs(randn(1));
                    vy_arr(q)=132.2e3*randn(1);
                else
                    vx_arr(q)= -vx_arr(q);
                end
            end
        end

        %top box boundary
        if y_arr(q)>0.6*y_dim && x_arr(q)>0.4*x_dim &&
x_arr(q)<0.6*x_dim
            if y_arr(q)>0.6*y_dim && yp_arr(q)<0.6*y_dim
```

```
                if retherm
                    vy_arr(q)=(132.2e3)*(-abs(randn(1)));
                    vx_arr(q)=132.2e3*randn(1);
                else
                    vy_arr(q)= -vy_arr(q);
                end
                y_arr(q)=1.2*(y_dim)-y_arr(q);
            end
            if x_arr(q)>0.4*x_dim && xp_arr(q)<0.4*x_dim
                if retherm
                     vx_arr(q)= -(132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);
                else
                     vx_arr(q)=-vx_arr(q);
                end
                x_arr(q)=0.4*x_dim-abs(x_arr(q)-0.4*x_dim);
            end
            if x_arr(q)<0.6*x_dim && xp_arr(q)>0.6*x_dim
                if retherm
                     vx_arr(q)= (132.2e3)*abs(randn(1));
                     vy_arr(q)=132.2e3*randn(1);
                else
                     vx_arr(q)=-vx_arr(q);
                end
                x_arr(q)=abs(x_arr(q)-0.6*x_dim)+0.6*x_dim;
            end
        end


    end
    %plot positions
    xlabel('X(m)')
    ylabel('Y(m)')
    title('8. Position of particles')
    xlim([0 x_dim])
    ylim([0 y_dim])
    pause(.01)
    for q=1:10
        plot([xg_arr(q);x_arr(q)],
[yg_arr(q);y_arr(q)],'color',col(q,:))
        hold on
    end


    t=t+t_step;
    count=count+1;
end
p=zeros(50);
v=zeros(50);
temp=zeros(50);

%make the density maps
for q=1:50
    for w=1:50
```

```
        for n=1:num_e
            if
 x_arr(n)>=(((q-1)*x_dim/50))&&(x_arr(n)<(q*x_dim/50))&&(y_arr(n)>=(w-1)*y_dim/50
                p(w,q)=p(w,q)+1;
                v(w,q)=v(w,q)+sqrt(vx_arr(n)^2+vy_arr(n)^2);
            end
        end
        if p(w,q)==0
            temp(w,q)=0;

        else
            temp(w,q)=0.26*9.109e-31*v(w,q)/p(w,q)/(1.3806e-23);
        end

    end
end

figure(2)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),p)
title('9. Electron density')
zlabel('Number of electrons')

figure (3)
surf(linspace(0,x_dim,50),linspace(0,y_dim,50),temp)
title('10. Temperature density')
zlabel('Temperature (K)')
```




9. Electron density
Number of electrons
15
10
5
0
1
$\times 10^{-7}$
0.5
0
0
0.5
1
1.5
2
$\times 10^{-7}$


10. Temperature density
Temperature (K)
0.08
0.06
0.04
0.02
0
1
$\times 10^{-7}$
0.5
0
0
0.5
1
1.5
2
$\times 10^{-7}$

Published with MATLAB® R2017a