Section 2 - Collisions with Mean Free Path (MFP)

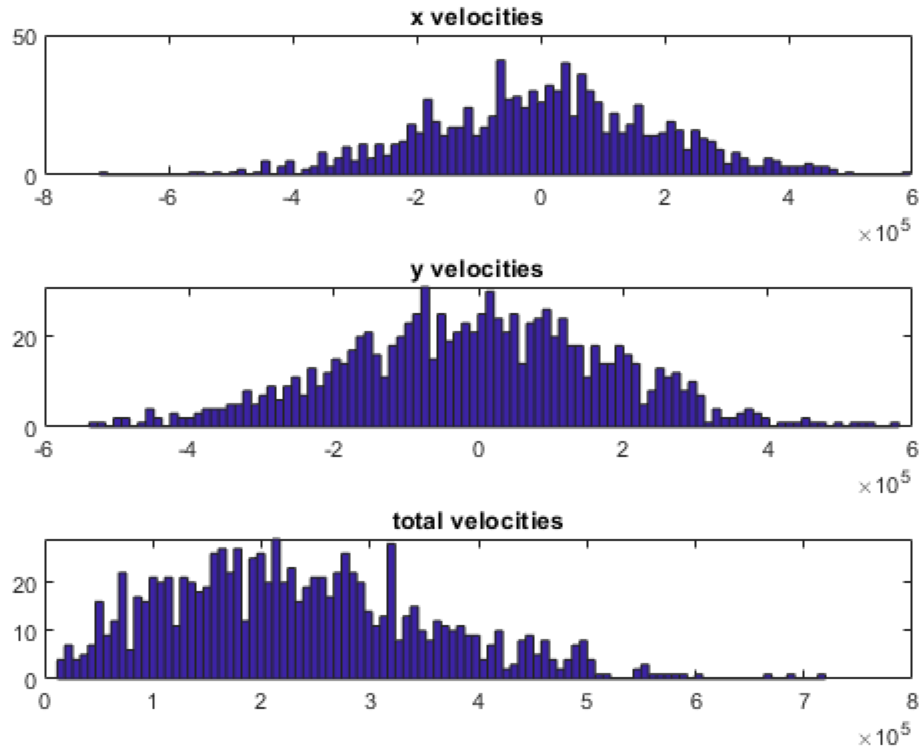
I found this section to be considerably easier than the previous, since most of the hard work was already done from the previous section. Adding the scattering was fairly easy, and I already had the temperature and trajectory plots working. Measuring the mean free path and mean time between collisions was also fairly straightforward. Adding the histograms for the velocity distributions wasn't hard, and I got the results I expected. The average speed was indeed the thermal voltage, and the x and y velocities were properly normally distributed.

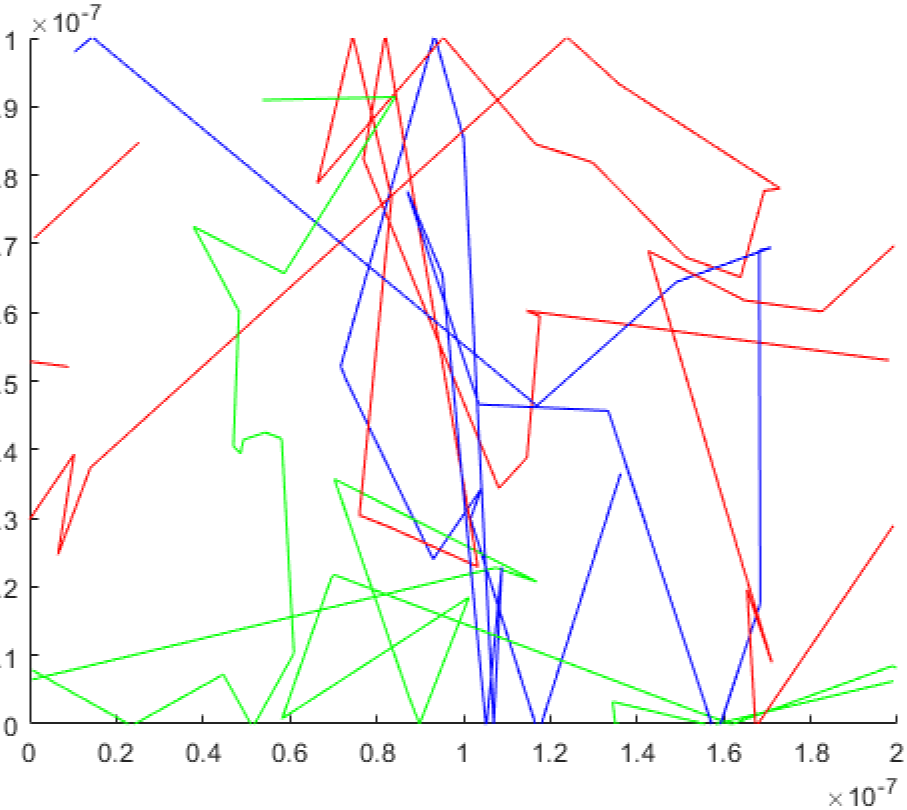
```
%constants
clear
C.q_0 = 1.60217653e-19;
C.m_0 = 9.10938215e-31;
C.kb = 1.3806504e-23;
C.T = 300;
frameWidth = 200e-9;
frameHeight = 100e-9;
nAtoms = 1000;
bins = nAtoms / 10;
Vth = sqrt(2*C.kb*C.T /(0.26*C.m_0));
dt = frameHeight/Vth/100;
Tstop = 750*dt;
t = 0;
freepath = 0.2e-12;
Pscatter = 1 - exp(-dt/freepath);

%initializing vectors
Xnext = zeros(1,nAtoms);
Ynext = zeros(1,nAtoms);
VX = Vth * randn(1,nAtoms);
VY = Vth * randn(1,nAtoms);
V = sqrt(VY.*VY+VX.*VX);
X = frameWidth * rand(1, nAtoms);
Y = frameHeight * rand(1, nAtoms);
R = zeros(1, nAtoms);
Temperature = zeros(1, 100);
meanpaths = zeros(1,nAtoms);
iteration = 1;
%histograms of X,Y, and overall velocities
figure(3)
subplot(3,1,1);
hist(VX,bins)
title('x velocities')
subplot(3,1,2);
hist(VY,bins)
title('y velocities')
subplot(3,1,3);
hist(V,bins);
title('total velocities')
```

```
while t < Tstop
    %determines which particles scatter and performs calculations on
 them
    %to determine mean free path and time between collisions
    R = rand(1,nAtoms);
    VX(R<Pscatter) = Vth*randn(1);
    VY(R<Pscatter) = Vth*randn(1);
    V = sqrt(VY.*VY+VX.*VX);
    meanpaths(R<Pscatter) = 0;
    unscattered = ismissing(R<Pscatter,0);
    meanpaths(unscattered) = meanpaths(unscattered) +
 V(unscattered)*dt;
    MFP = sum(meanpaths)/nAtoms;
    MTBC = sum(meanpaths)/sum(V);


    Xnext = X + VX*dt;
    Ynext = Y + VY*dt;
    %X boundary conditions set
    right = Xnext>frameWidth;
    left = Xnext<0;
    Xnext(right) = Xnext(right)-frameWidth;
    Xnext(left) = Xnext(left) + frameWidth;
    %Y boundary conditions set
    top = Ynext > frameHeight;
    bottom = Ynext < 0;
    VY(top | bottom) = VY(top | bottom) * -1;
    %calculations for temperature
    Temperature(iteration) = 0.26*C.m_0*mean(V.^2)/4/C.kb;
    figure(4)
    xlim([0 frameWidth])
    ylim([0 frameHeight])
    hold on
    %plotting, but avoid plotting the full horizontal jump
    if abs(Xnext(1) - X(1)) < 2*abs(VX(1))*dt
        figure(4)
        plot([Xnext(1) X(1)], [Ynext(1) Y(1)], 'blue')
    end
    if abs(Xnext(2) - X(2)) < 2*abs(VX(2))*dt
        figure(4)
        plot([Xnext(2) X(2)], [Ynext(2) Y(2)], 'red')
    end
    if abs(Xnext(3) - X(3)) < 2*abs(VX(3))*dt
        figure(4)
        plot([Xnext(3) X(3)], [Ynext(3) Y(3)], 'green')
    end

    %updating positions, and advancing time a step forward so the
 while
    %loop works
    X = Xnext;
    Y = Ynext;
    t = t+dt;
    iteration = iteration + 1;
```

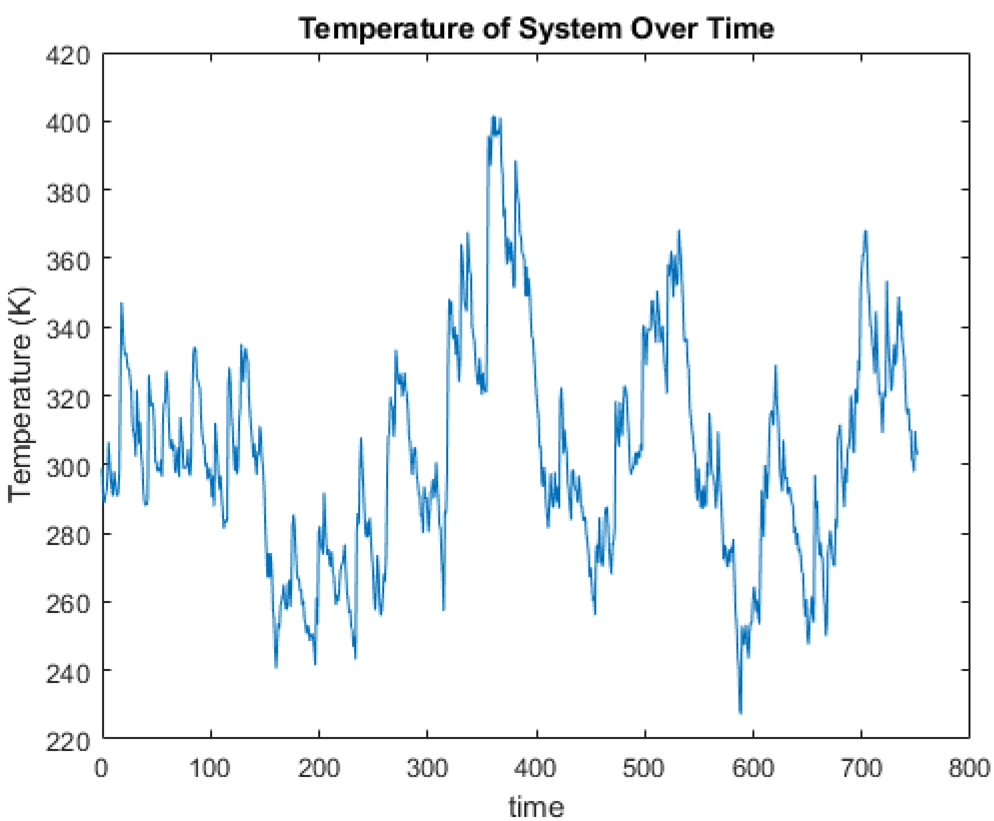
```
    pause(0.0001);
end
%Outputs, temperature, mean free path, and mean time between
 collisions
figure(5)
dummy = linspace(0,iteration, length(Temperature));
plot(dummy, Temperature)
title('Temperature of System Over Time')
xlabel('time')
ylabel('Temperature (K)')

fprintf('The mean free path is %d m.\n',MFP);
fprintf('The Mean Time Between Collisions is %d s.\n',MTBC);
```

The mean free path is 4.691454e-08 m.
The Mean Time Between Collisions is 1.965724e-13 s.





×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



Temperature of System Over Time
Temperature (K)
420
400
380
360
340
320
300
280
260
240
220
0
100
200
300
400
500
600
700
800
time

Published with MATLAB® R2018a