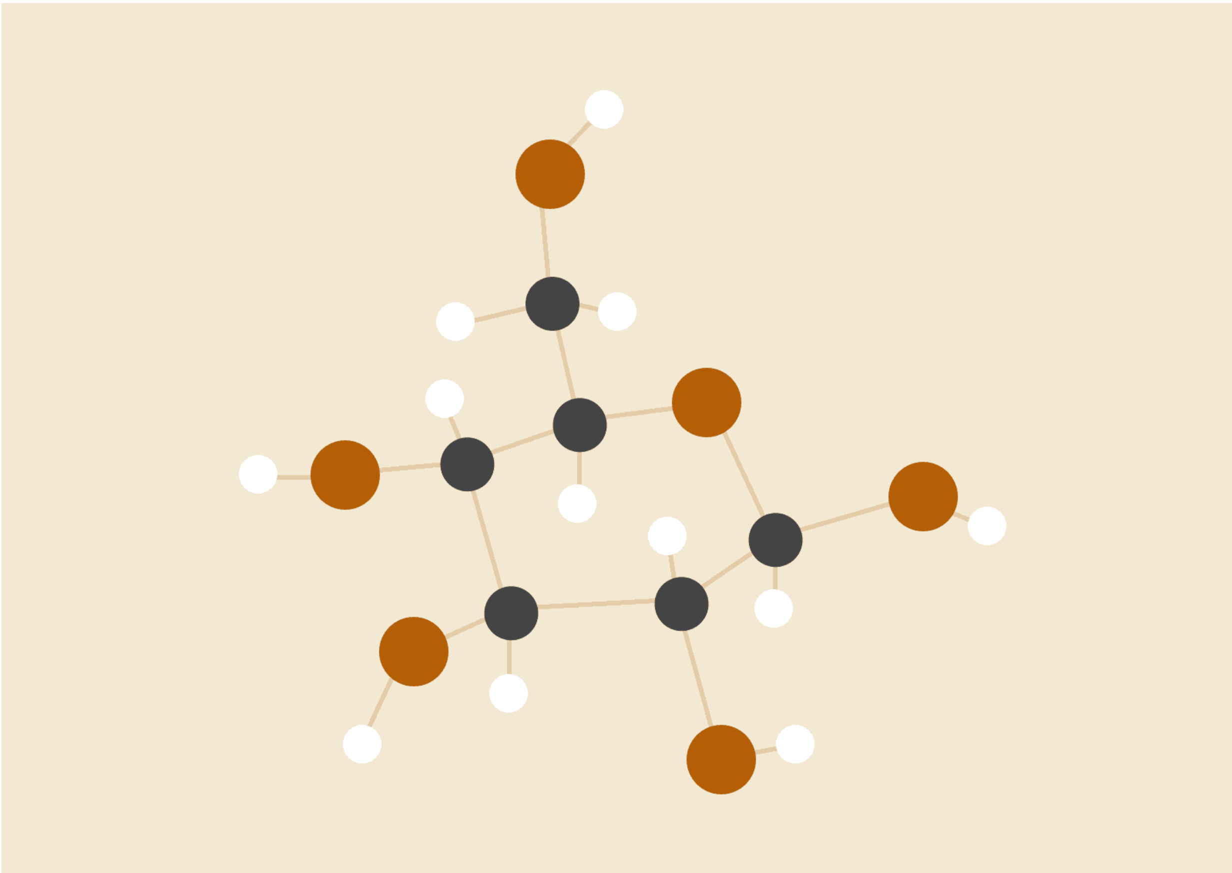
BD SIMULATION SIMUTECH PROJECT



Report by

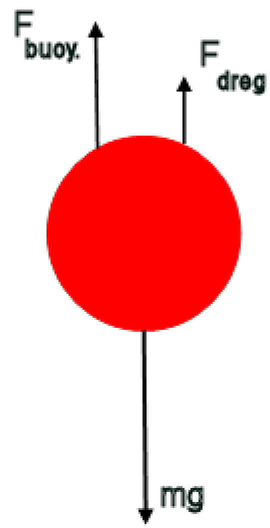
Devang Kumawat

Mentor: Diya Singhal

Assignment 1-Analyzing Force on a Particle

In assignment 1, we are given a problem with a sphere falling in a viscous fluid and we have to calculate the terminal velocity of the sphere & derive an analytical equation for the same. Also we have to plot the graph of termi

Derived equation for Terminal Velocity



As we can see in the figure, in upward direction the buoyancy force due to the liquid in which sphere is submerged & drag force due to the viscosity of the fluid are acting and in downward direction the weight of the sphere is acting,

Therefore, the net force will be the difference of the force between Upward and downward direction.

$$\sigma V \left(\frac{du}{dt}\right) = W - \rho V g - 6\pi R \eta u$$

$$\sigma V \frac{du}{dt} = \sigma V g - \rho V g - 6\pi R \eta u$$

$$\sigma V \frac{du}{dt} = (\sigma - \rho) V g - 6\pi R \eta u$$

$$\frac{du}{(\sigma-\rho)Vg-6\pi\eta Ru}=\frac{dt}{\sigma V}$$

Integrating this expression, we get.

$$\int_0^u \frac{du}{(\sigma-\rho)Vg-6\pi\eta Ru}=\int_0^t \frac{dt}{\sigma V}.$$

$$\frac{\left[\ln\left[(\sigma-\rho)Vg-6\pi\eta Ru\right]\right]_0^u}{-6\pi\eta R}=\frac{t}{\sigma V}$$

$$\ln\left[\frac{(\sigma-\rho)Vg-6\pi\eta Ru}{(\sigma-\rho)Vg}\right]=-\frac{6\pi\eta Rt}{\sigma V}.$$

$$1-\frac{6\pi\eta Ru}{(\sigma-\rho)Vg}=e^{-\frac{6\pi\eta Rt}{\sigma V}}.$$

$$\frac{6\pi\eta Ru}{(\sigma-\rho)Vg}=1-e^{-\frac{6\pi\eta Rt}{\sigma V}}.$$

$$u=\frac{(\sigma-\rho)Vg}{6\pi\eta R}\left(1-e^{-\frac{6\pi\eta Rt}{\sigma V}}\right)$$

Problem Statement

EXERCISE:

- ❑ Fix the properties of the fluid and solid sphere.
- ❑ Do you get a terminal velocity? What will be the analytical expression for the same?
- ❑ Get the analytical answer for *u* vs *t* (assume *u* = 0 at *t* = 0)
- ❑ For the same initial velocity, numerically compute *u* vs *t*.
- ❑ Compare the analytical and numerical answers on a plot of *u* vs *t*.
  - ❑ Analytical – solid line; numerical – dashed line

$$\frac{u_{t+\Delta t} - u_t}{\Delta t} = g - \frac{V\rho g}{m} - \frac{6\pi R\eta u_t}{m}$$

Code for numerical & analytical solution

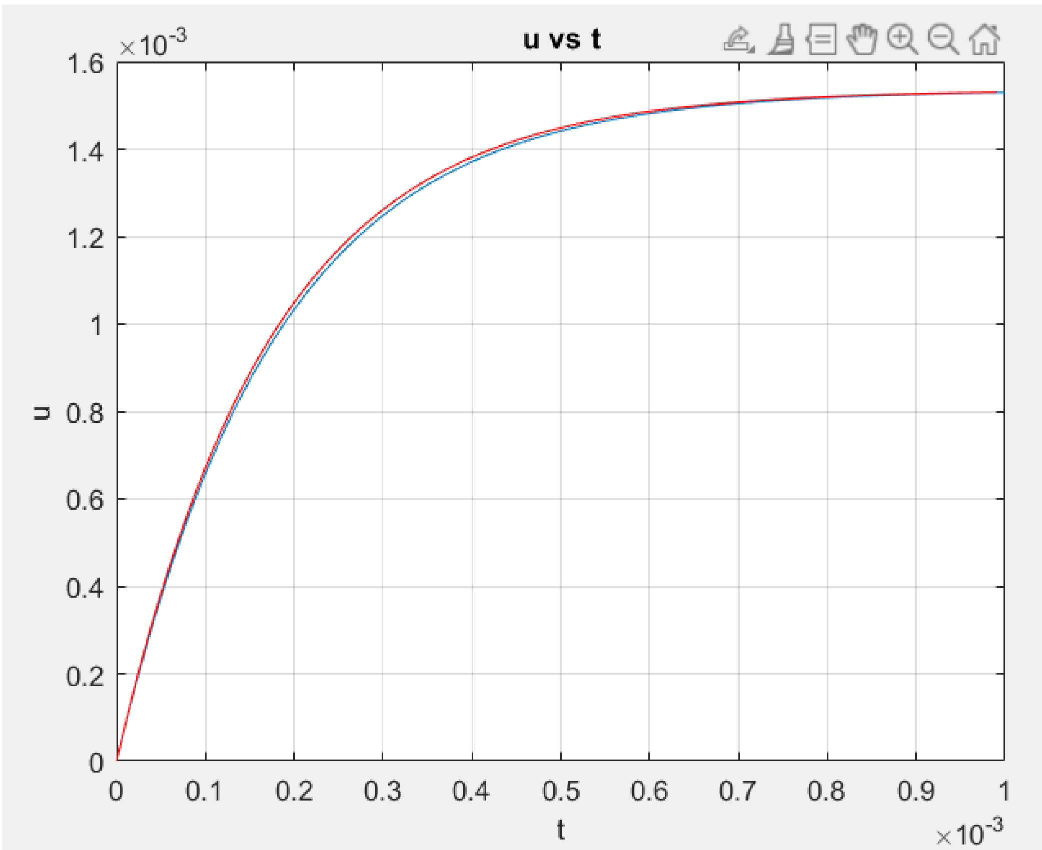
```
%  code for analatical
R = 10.^-5; % radius of sphere
p = 1000; %density of liquid
s = 8050; %density of sphere
n = 10.^-3; % cofficient of viscousity
g = 9.8; % accerelation due to gravity
t1 = [0:0.00001: 0.001];
u1 = ((2*(s-p)*R*R*g)/(9*n))*(1-exp((-9*n*t1)/(2*s*R*R))); % derived eq.
%code for numerical method:
dt = 10.^-4;
t2 = [0:dt:0.001];
u2 = [0];
ui=0;
uf=0;
j=1;
for i= 0: 0.0001 : 0.001
    uf = ui + (((s-p)*g/s) - (9*n*ui)/(2*s*R*R))*dt;
    u2(j) = uf;
    j=j+1;
    ui=uf;
end

u2(1)=0;

plot(t2 , u2 , t1 , u1)
title('U vs T for Analytical & Numerical Method')
xlabel('T')
ylabel('U')
```

GRAPH FOR (U VS T) FOR ANALYTICAL & NUMERICAL METHOD

U vs T GRAPH

Analytical Plot (RED LINE)

Numeric al Plot (BLUE LINE)



Concepts Learned

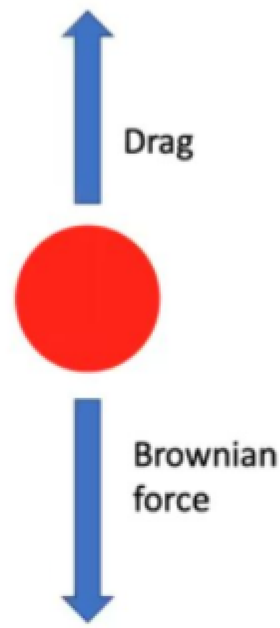
- Finding Drag force.
- Analytical and Numerical approach.

Assignment 2-Deriving Trajectory of one particle

Aim of the second assignment was:

- Plot a graph of the trajectory of the brownian motion using Matlab.
- Calculate Mean Square Displacement and plot a graph of MSD vs t*.
- Calculate the diffusivity.

Deriving Trajectory of a Brownian Particle



Forces on a Brownian Particle

Drag Force

$$\vec{F}_{drag} = -\zeta\vec{u} = -\zeta\frac{d\vec{r}}{dt}$$

Brownian Force

$$\vec{F}_B = \sqrt{\frac{6k_BT\zeta}{\Delta t}}\vec{n}$$

Equation of Motion

$$m\frac{d\vec{u}}{dt} = \vec{F}_B + \vec{F}_{drag}$$

$$0 = \vec{F}_B + \vec{F}_{drag}$$

$$\boxed{\zeta\frac{d\vec{r}}{dt} = \sqrt{\frac{6k_BT\zeta}{\Delta t}}\vec{n}}$$

Now instead of putting so many constants in this expression we used the non-dimensional version of the equation of motion to make calculations easier.

$$\frac{d\vec{r^*}}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}\vec{n}$$

where r* and t* are:

$$r^* = \frac{r}{R} \qquad t^* = \frac{t}{\frac{\zeta R^2}{k_B T}} = t\frac{k_B T}{\zeta R^2}$$

According to the given data in the assignment we calculated the value of the expression in each of the directions that is x,y,z. And hence plotted xy projection of the trajectory of the BD particle.

Problem Statement

Exercise

- Consider a particle at origin at t* = 0.
- Consider Brownian dynamics simulation of this particle
  - Take $\Delta t^* = 0.001$
  - Perform simulation for a total time t* = 100 (10^5 time steps)
  - Write the x*, y*, z* vs t* on a file
  - Plot the trajectory on a graph. Just show the xy projection (only x* and y*). For this, join all (x*,y*) points collected by a line. X-axis of plot denotes x* and y-axis denoted y*.
  - Calculate MSD vs Δt^* for the trajectory. Plot MSD vs t* (only upto t* = 10)
  - Calculate diffusivity. What's the value you get?

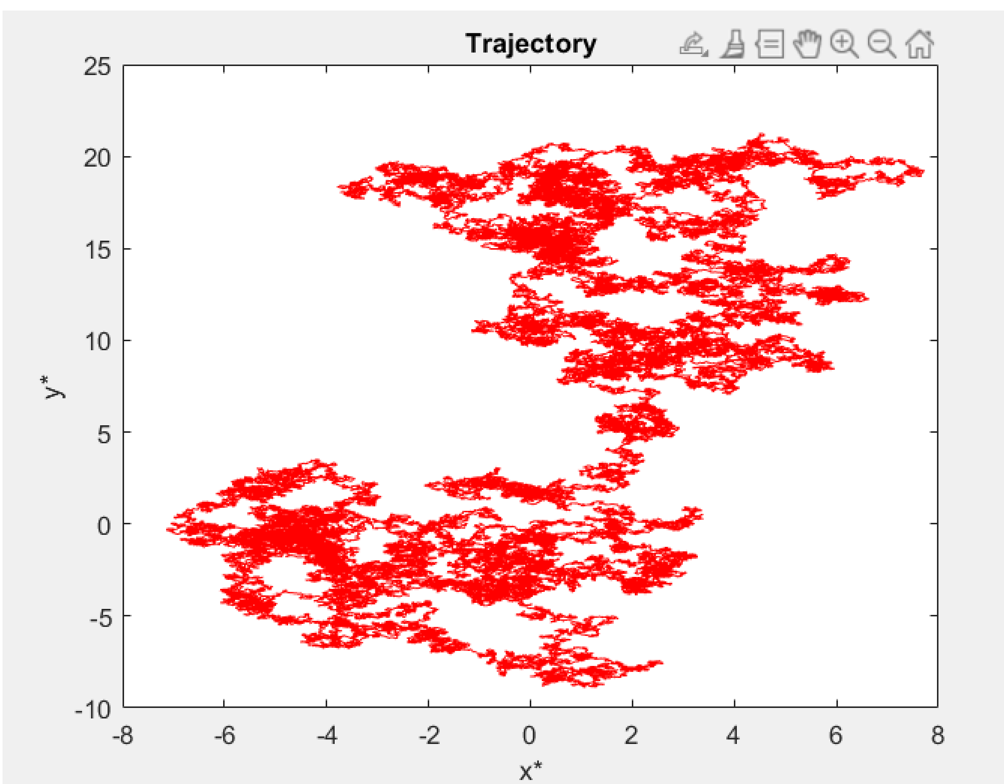
Matlab Code

```
function Assignment_2()
dt = 0.001;
data(1,1) = 0;
data(1,2) = 0;
data(1,3) = 0;
data(1,4) = 0;
for i = (2:100000)
    %t
    data(i,4) = data(i-1,4)+dt;

    n = 2*(rand(3,1))-1;
    n = n/norm(n);
    %x
    data(i,1) = ((6/0.001)^0.5)*n(1,1)*0.001+data(i-1,1);
    %y
    data(i,2) = ((6/0.001)^0.5)*n(2,1)*0.001+data(i-1,2);
    %z
    data(i,3) = ((6/0.001)^0.5)*n(2,1)*0.001+data(i-1,3);
end
plot(data(:,1),data(:,2),'r')
xlabel('x*'), ylabel('y*'), title('Trajectory')
hold off
```

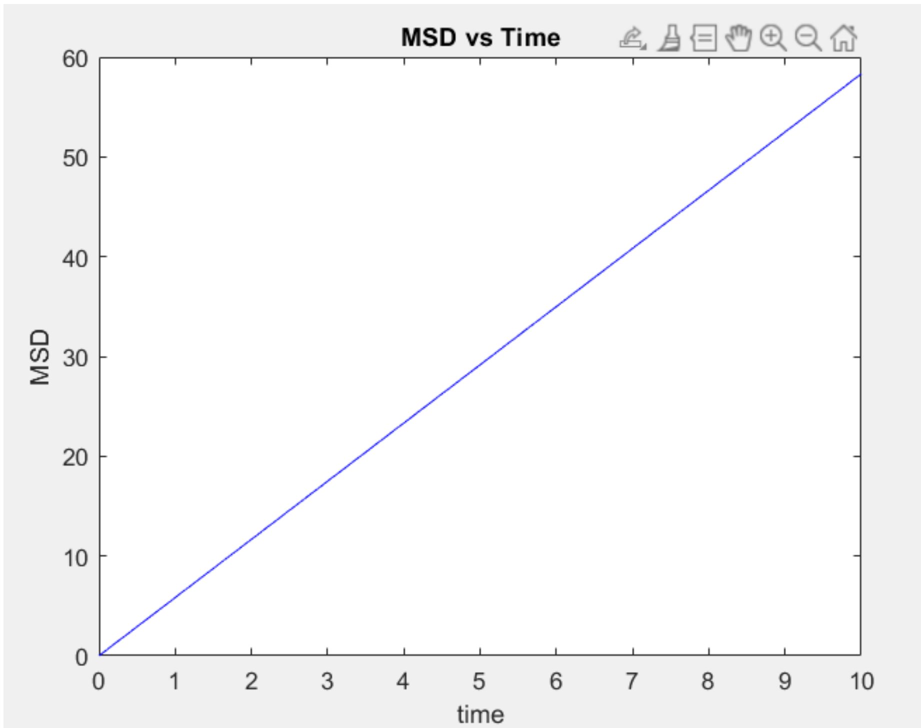
Calculation of MSD & Diffusivity

```
msd(1,1) = 0;
for i = (2:10000)
    sum = 0;
    for j = (i:10000)
        sum = sum + ((data(j,1)-data(j-i+1,1))^2+(data(j,2)-data(j-i+1,2))^2+(data(j,3)-data(j-i+1,3))^2) ;
    end
    msd(i,1) = sum/(10001-i);
end
figure
x_ = (0:dt:10);
X = x_(1:100);
y_ = (msd(1:100))';
f = polyfit(X,y_,1);
Y = polyval(f,x_);
plot(x_,Y,'b')
ylabel('MSD'),xlabel('time'),title('MSD vs Time')
diffusivity = (msd(2,1))/(0.001)/6
```

Trajectory of BD Particle



MSD vs t*



Diffusivity

```
>> Assignment_2

diffusivity =

    1.0006
```

Concepts Learned

- Simplifying expression by non-dimensionalising it.
- Calculating Mean Square Displacement
- Calculating Diffusivity

Assignment 3- Deriving Brownian Dynamics

AIM

This assignment is to understand the bead-spring model of single polymer chain and plot the graph of non dimensionalized R(distance between two beads) vs t(time).

Software used: MATLAB

HYPOTHESIS

A polymer chain consists of thousands of beads which can be simplified as N+1 beads connected by N springs. Beads experience certain forces which can be balanced. The forces experienced on a particle are drag force, Brownian force and spring force. Other forces are neglected.

FORCE EXPRESSION

1. **Drag Force**

   $F_{drag,i}$= -ζ dr_i /dt

2. **Brownian Force**

   $F_{B,i}$ = sqrt(6k_B Tζ/Δt)n_i

   where
   n = n vector, random vector with values in range [-1,1]

3. **Spring Force**

   $F_{sp,1}$= (k_B T/v$b_K{}^2$)((3-ṙ^2)/(1-ṙ^2))R

   $F_{sp,2}$= -(k_B T/v$b_K{}^2$)((3-ṙ^2)/(1-ṙ^2))R

   where

v= number of Kuhn lengths
b_K=Kuhn length of the polymer chain
R=r2 -r1
ṙ=|R|/(vb_K)

Non dimensionalizing r, t & F
r*=r/b_2
t*=t (k_B T)/($\zeta b_K{}^2$)
F*=Fb_K /(k_B T)

New non dimensionless version of equation:
dr1*/dt*=sqrt(6/ Δt*)n1 + (3-ṙ^2)/(v(1-ṙ^2))R*
dr2*/dt*=sqrt(6/ Δt*)n2 + (3-ṙ^2)/(v(1-ṙ^2))R*

Problem Statement

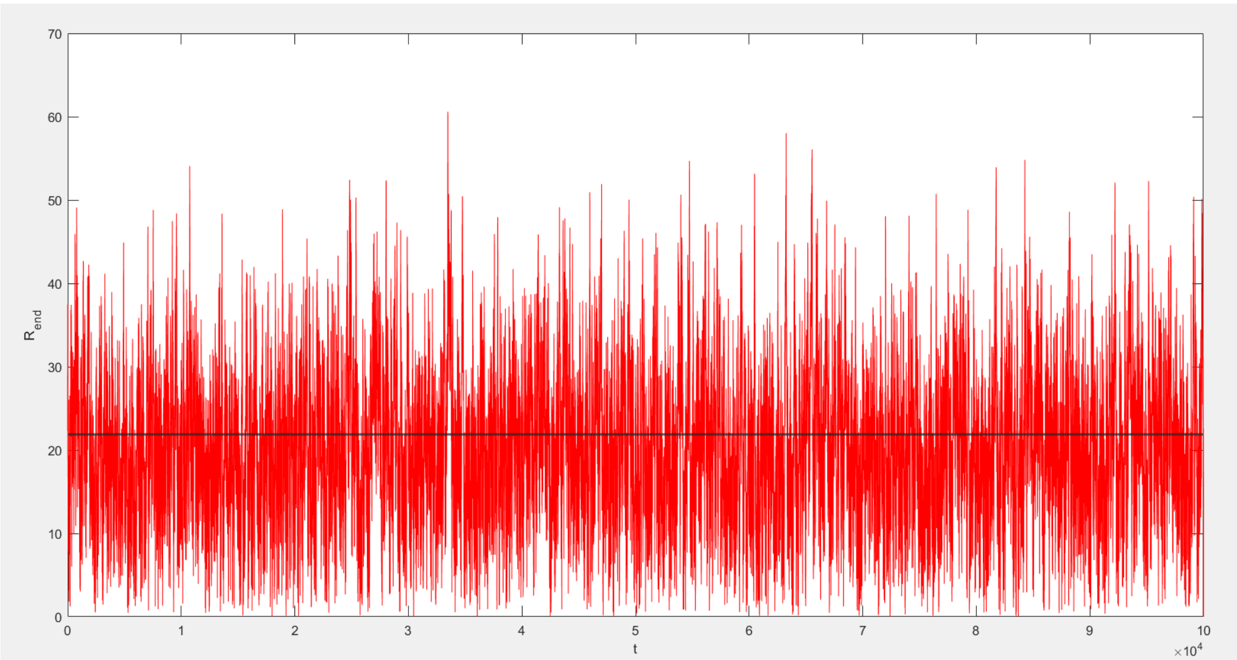
Exercise

- Take $\nu = 500$
- Consider an initial chain orientation at t* = 0 of $\vec{r_1^*} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ and $\vec{r_2^*} = \begin{bmatrix} \sqrt{\nu} \\ 0 \\ 0 \end{bmatrix}$ at $t^* = 0$
- Consider Brownian dynamics simulation of this chain
  - Take $\Delta t^* = 0.001$
  - Perform simulation for a total time t* = 10000 (10^7 time steps)
  - Write the R^*_{end} vs t* on a file (R^*_{end} denotes the magnitude of $\vec{R}^* = \vec{r}_2^* - \vec{r}_1^*$)
  - Plot R^*_{end} vs t* on a graph – blue thin line
  - Calculate RMS value of R^*_{end} from the values. Show it by a solid straight line (black thick line) on the graph. (You will need to write a separate code)

Matlab Code

```
function Assignment3()
v = 500;
dt = 0.001;
t = 100000;
N = t/dt;
data = zeros(N+1,1);
%initial condition
r1 = zeros(N+1,3);
r2 = zeros(N+1,3);
r2(1,1) = sqrt(v);
% R = zeros(3,1);
Rend=zeros(N+1,1);
sum = 0;
for i = (2:N+1)
    %t
    data(i,1) = data(i-1,1)+dt;

    n1 = 2*(rand(3,1))-1;
    n2 = 2*(rand(3,1))-1;
    R = r2(i-1,:)-r1(i-1,:);
    r_cap = norm(R)/v;

    %for r1_vec = (data(i,2),data(i,3),data(i,4));
    %x
    r1(i,1) = ((6/dt)^0.5)*n1(1,1)*dt+r1(i-1,1)+(3-r_cap^2)/(v*(1-r_cap^2))*R(1,1)*dt;
    %y
    r1(i,2) = ((6/dt)^0.5)*n1(2,1)*dt+r1(i-1,2)+(3-r_cap^2)/(v*(1-r_cap^2))*R(1,2)*dt;
    %z
    r1(i,3) = ((6/dt)^0.5)*n1(3,1)*dt+r1(i-1,3)+(3-r_cap^2)/(v*(1-r_cap^2))*R(1,3)*dt;
    %for particle 2
    %x
    r2(i,1) = ((6/dt)^0.5)*n2(1,1)*dt+r2(i-1,1)-(3-r_cap^2)/(v*(1-r_cap^2))*R(1,1)*dt;
    %y
    r2(i,2) = ((6/dt)^0.5)*n2(2,1)*dt+r2(i-1,2)-(3-r_cap^2)/(v*(1-r_cap^2))*R(1,2)*dt;
    %z
    r2(i,3) = ((6/dt)^0.5)*n2(3,1)*dt+r2(i-1,3)-(3-r_cap^2)/(v*(1-r_cap^2))*R(1,3)*dt;
    Rend(i-1)=norm(R);
    sum = sum+norm(R)*norm(R);
end
rms = sqrt(sum/N);
plot(data(:,1),Rend,'r')
xlabel('t'), ylabel('R_e_n_d')
hold on
yline(rms,'Linewidth',2)


hold off
```

PLOT



Concepts Learned

- Bead-Spring model of polymer chain
- Spring force on a molecular spring