BD Simulation Simutech Project

Tushar Soni
201054
tushars20@iitk.ac.in

Brownian Dynamic simulation of polymer chains

Project Mentor:- **DIYA SINGHAL**
Software Required:- **MATLAB**

Assignment-1

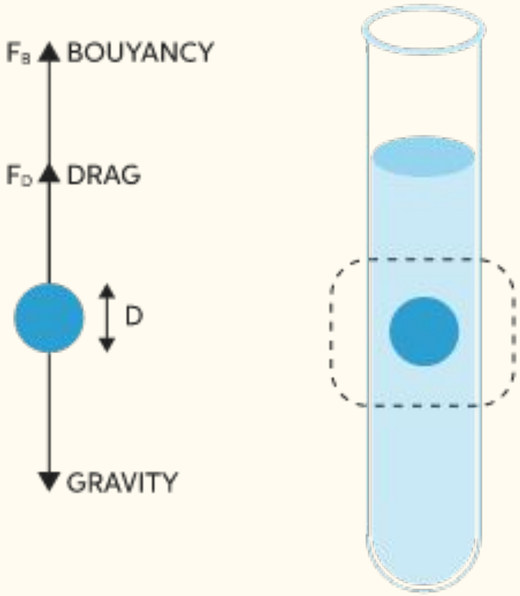
Sphere falling in an viscous fluid

Objective-

- Plotting the graph of velocity(u)vs Time(t)
- Numerically Compute u vs t
- Compare Analytical vs numerical (u vs t)

Theory:-

When the Sphere fall in a viscous fluid, it experiences three forces in which Fs(Buoyancy force) & Fd(drag force) acts on the vertical direction & Fg acts downward as shown in the following figure.



Therefore,

$m(du/dt) = Fg - Fs - Fd = \boldsymbol{\rho} Vg - \boldsymbol{\sigma} Vg - 6\boldsymbol{\pi}\eta ru$

Now, $du/dt = u1 - u0/\Delta t$;

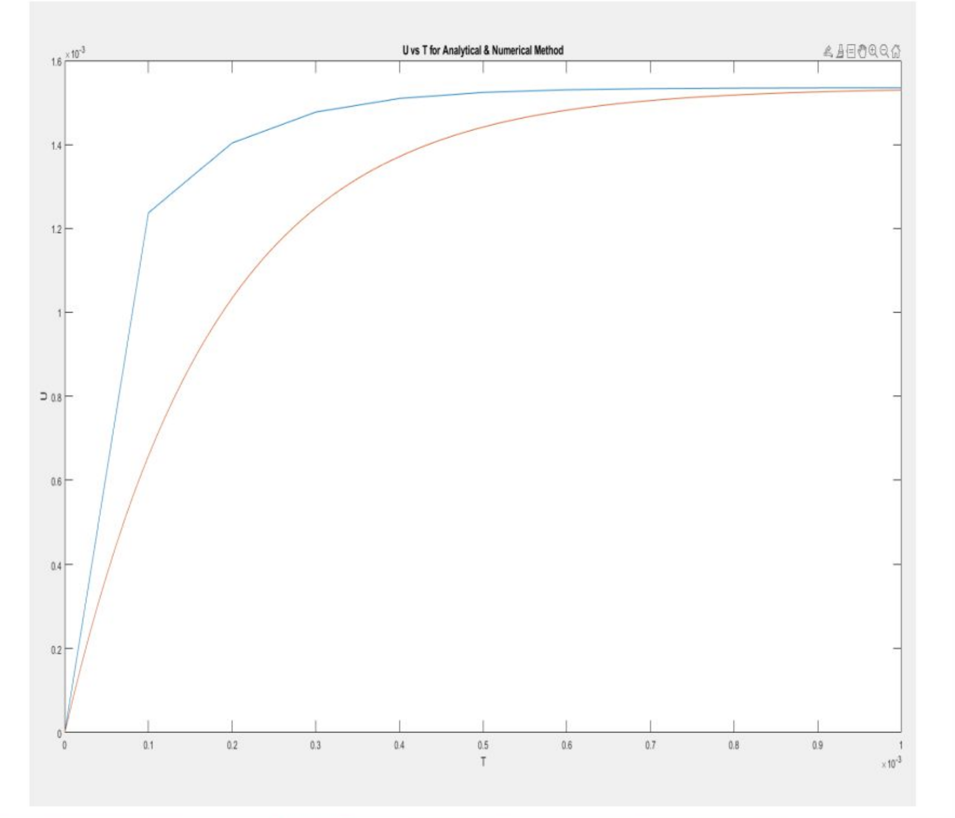
After Solving the equation & starting with a guess value of $u(0)=0$,
We iteratively solve the value of u at every Δt value & we will make a plot between the Velocity U and Time t.

Matlab Code:-

```
%  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')
```

Plot between U & T

Numerical(Blue) & Analytical(Red)



Assignment-2

Trajectory of Brownian Particle & Mean Square Displacement

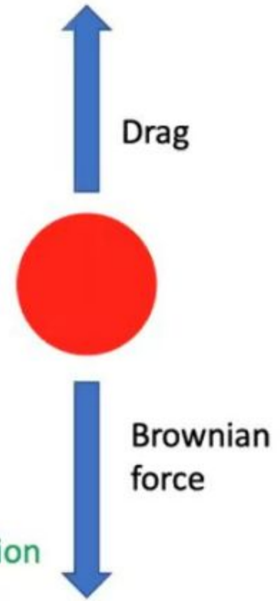
Objective-

- Plotting the graph of Position(x) vs Position(y)
- Calculating the mean square displacement

Theory:-

Drag force $\vec{F}_{drag} = -\zeta \vec{u} = -\zeta \dfrac{d\vec{r}}{dt}$

Brownian force $\vec{F}_B = \sqrt{\dfrac{6 k_B T \zeta}{\Delta t}}\,\vec{n}$



Equation of Motion

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

$$0 = \vec{F}_B + \vec{F}_{drag}$$ Neglect inertia: Brownian dynamics assumption

$\vec{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$; Each component is a random number between -1 and 1

Theory:-

Equation of Motion

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

Use non-dimensionless version of the equation of motion:

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

Assume $\overrightarrow{r^*} = 0$ at $t^* = 0$ (particle starts at origin)

$$\frac{dx^*}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}n_x \qquad \frac{dy^*}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}n_y$$

$$\frac{dz^*}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}n_z$$

Select a length scale equal to radius of particle: R

$$r^* = \frac{r}{R}$$

A relevant time scale: $\frac{\zeta R^2}{k_B T}$

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

Stokes-Einstein diffusivity:

$$D_{SE} = \frac{k_B T}{\zeta}$$

Re-look at the time scale:

$$\frac{\zeta R^2}{k_B T} = \frac{R^2}{k_B T/\zeta} = \frac{R^2}{D_{SE}}$$

Theory:-

Calculation of Mean Square Displacement (MSD)

Example trajectory (1D)

t	x^*
0	0
1	1
2	2.5
3	4.5
4	6.5

MSD(1)=MSD(Δt = 1)=AVG[$(1\text{-}0)^2,(2.5\text{-}1)^2,(4.5\text{-}2.5)^2,(6.5\text{-}4.5)^2$]

MSD(1) = [$1^2 + 1.5^2 + 2^2 + 2^2$]/4 = 2.81

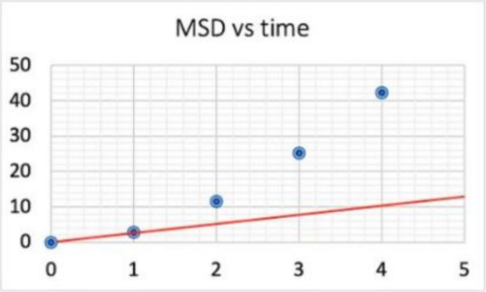
MSD(2) = [$(2.5\text{-}0)^2 + (4.5\text{-}1)^2 + (6.5\text{-}2.5)^2$]/3 = 11.5

MSD(3) = [$(4.5\text{-}0)^2 + (6.5\text{-}1)^2$]/2 = 25.25

MSD(4) = [$(6.5\text{-}0)^2$]/1 = 42.25

MSD(0) = 0, by definition itself

Slope at t=0 gives the diffusivity



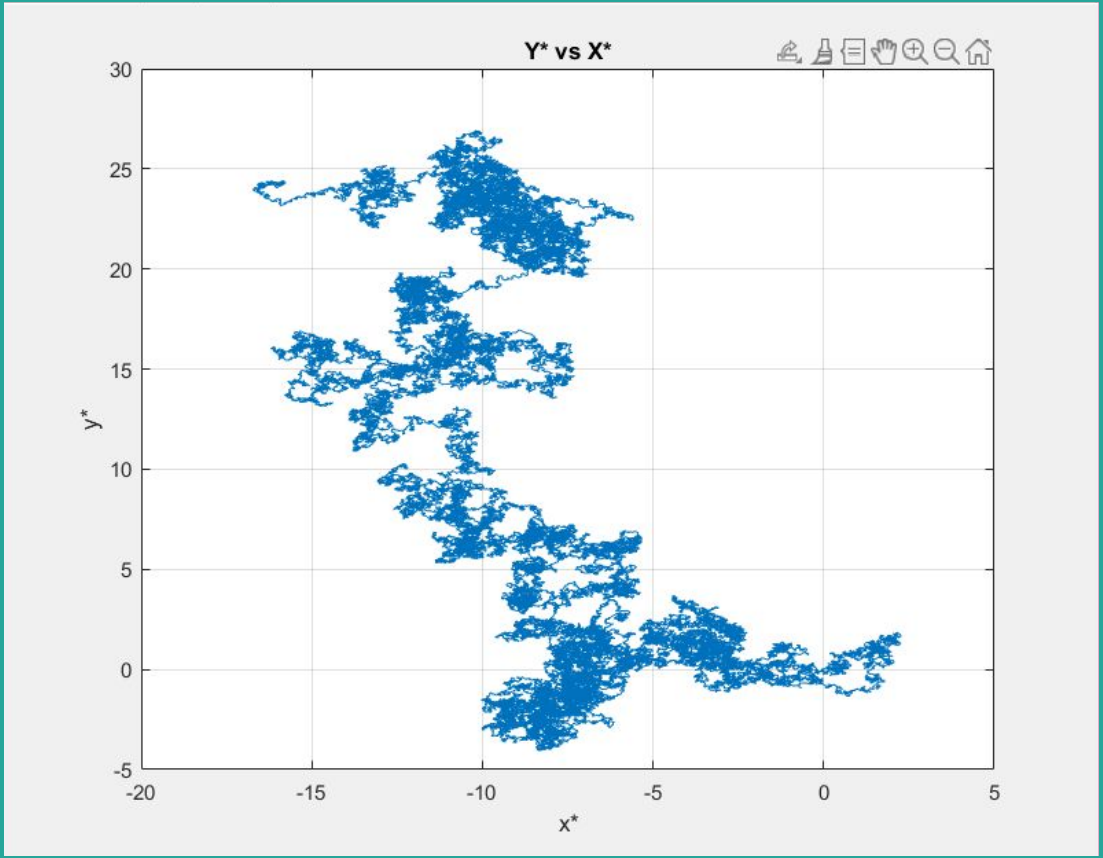
MSD vs time:

Δt	MSD
0	0
1	2.81
2	11.5
3	25.25
4	42.25

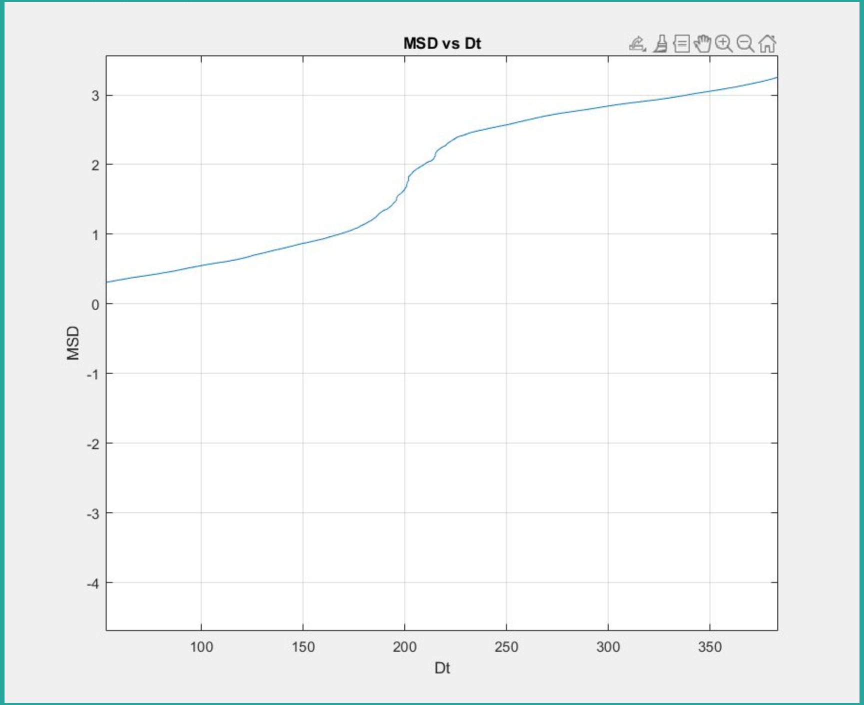
Matlab Code:-

```
dt = 0.001;
x0 = 0;
y0 = 0;
z0 = 0;
x = [0];
y = [0];
z = [0];
t = [0];
x1=0;
y1=0;
z1=0;
t0=0;
j=2;
for i=0.002:dt:100
    nx = 2*rand-1;
    ny = 2*rand-1;
    nz = 2*rand-1;
    x1 = x0 + sqrt(6*dt)*nx;
    y1 = y0 + sqrt(6*dt)*ny;
    z1 = z0 + sqrt(6*dt)*nz;
    x(j) = x1;
    y(j) = y1;
    z(j) = z1;
    t0 = t0 + dt;
    t(j)= t0;
    x0=x1;
    z0=z1;
    y0 =y1;
    j=j+1;
end
msd = [0];
k=1;
n = length(x);
delt = [0];
k=2;
for i=2:1:10000
    msd(i)=0;
    for j=i:1:10000
        msd(i) = msd(i) + (x(j)-x(j-k+1))^2;
    end
    msd(i) = msd(i)/((n-k)*0.001);
    delt(k)=(k-1)*0.001;
    k = k + 1;
end
diffusivity = msd(2)/0.001;
figure(1)
plot(x,y)
title('Y* vs X*')
xlabel('x*')
ylabel('y*')
grid
figure(2)
plot(msd,delt)
title('MSD vs Dt')
xlabel('Dt')
ylabel('MSD')
grid
```

Plot between X* & Y*



MSD VS TIME



Diffusivity:-199.56

Assignment-3

Dynamics of a single polymer chain

Objective-

- Plotting the graph between Rend (R2-R1) vs Time
- Calculating the Root mean square displacement

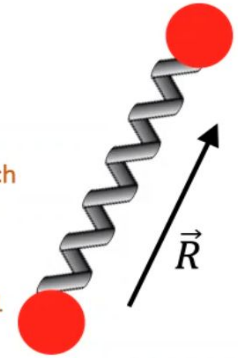
Theory:-

Dynamics of a single polymer chain

Drag force on each bead $\vec{F}_{drag,i} = -\zeta \vec{u_i} = -\zeta \dfrac{d\vec{r_i}}{dt}$

Brownian force on each bead $\vec{F}_{B,i} = \sqrt{\dfrac{6k_BT\zeta}{\Delta t}}\,\vec{n_i}$

$\vec{n_i} = \begin{bmatrix} n_{x,i} \\ n_{y,i} \\ n_{z,i} \end{bmatrix}$; Each component is a random number between -1 and 1



Spring force $\vec{F}_{sp,1} = \dfrac{k_BT}{\nu b_K^2}\dfrac{(3-\hat{r}^2)}{1-\hat{r}^2}\vec{R}$

$$\vec{F}_{sp,2} = -\frac{k_BT}{\nu b_K^2}\frac{(3-\hat{r}^2)}{1-\hat{r}^2}\vec{R} \qquad \vec{R} = \vec{r}_2 - \vec{r}_1 \qquad \hat{r} = \frac{|\vec{R}|}{\nu b_K}$$

ν: number of Kuhn lengths mimicked by one spring

b_K: Kuhn length of the polymer chain

Equation of Motion (i = 1, 2)

$$0 = \vec{F}_{B,i} + \vec{F}_{drag,i} + \vec{F}_{sp,i} \qquad \zeta\frac{d\vec{r_i}}{dt} = \sqrt{\frac{6k_BT\zeta}{\Delta t}}\,\vec{n_i} + \vec{F}_{sp,i}$$

Theory:-

Solving equations of motion of beads

Equation of Motion

$$\zeta \frac{d\vec{r_i}}{dt} = \sqrt{\frac{6k_BT\zeta}{\Delta t}}\vec{n_i} + \vec{F}_{sp,i}$$

Use non-dimensionless version:

$$\frac{d\vec{r_1^*}}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}\vec{n_1} + \frac{3-\hat{r}^2}{\nu(1-\hat{r}^2)}\vec{R}^*$$

$$\frac{d\vec{r_2^*}}{dt^*} = \sqrt{\frac{6}{\Delta t^*}}\vec{n_2} - \frac{3-\hat{r}^2}{\nu(1-\hat{r}^2)}\vec{R}^*$$

Initial condition:

Assume $\vec{r_1^*} = \begin{bmatrix}0\\0\\0\end{bmatrix}$ and $\vec{r_2^*} = \begin{bmatrix}\sqrt{\nu}\\0\\0\end{bmatrix}$ at $t^* = 0$

Select a length scale equal to Kuhn step: b_K

$$r^* = \frac{r}{b_K}$$

A relevant time scale: $\dfrac{\zeta b_K^2}{k_BT}$

$$t^* = \frac{t}{\frac{\zeta b_K^2}{k_BT}} = t\frac{k_BT}{\zeta b_K^2}$$

Force scale $\dfrac{k_BT}{b_K}$

$$F^* = \frac{F}{k_BT/b_K} = \frac{Fb_K}{k_BT}$$

Theory:-

Calculation of Root Mean Square (RMS) value

Example:

t	r
0	1
1	0.7
2	1.2
3	1.1
4	0.8
5	0.6
6	0.9
7	1.2
8	1.5

Mean Square of r:

$= [1^2 + 0.7^2 + 1.2^2 + 1.1^2 + 0.8^2 + 0.6^2 + 0.9^2 + 1.2^2 + 1.5^2]/9$

$= 1.0711$

Root Mean Square (RMS) of r:

$= \text{sqrt}\{[1^2 + 0.7^2 + 1.2^2 + 1.1^2 + 0.8^2 + 0.6^2 + 0.9^2 + 1.2^2 + 1.5^2]/9\}$

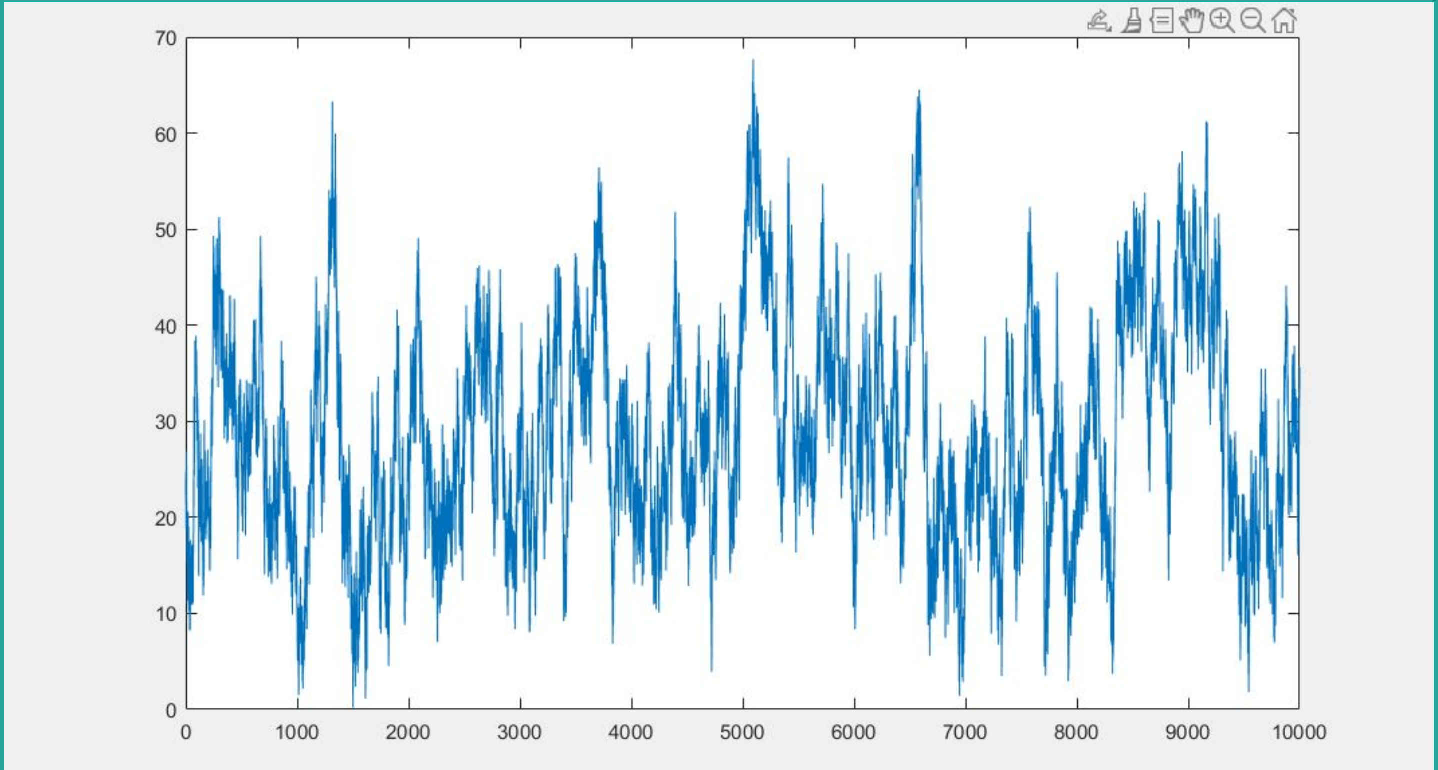
$= \text{sqrt}(1.0711)$

$= 1.035$

Matlab Code:-

```
dt=0.001;
v=500;
r1x=0;
r1y=0;
r1z=0;
r2x=sqrt(v);
r2y=0;
r2z=0;
time = [];
t=0;
Rend = [];
rms=0;
for i=1:1:10000000
    nx=2*rand-1;
    ny=2*rand-1;
    nz=2*rand-1;

    n_x=2*rand-1;
    n_y=2*rand-1;
    n_z=2*rand-1;

    R = sqrt((r2x-r1x)*(r2x-r1x) + (r2y-r1y)*(r2y-r1y) + (r2z-r1z)*(r2z-r1z));
    rcap = norm(R)/v;

    a = abs(r2x-r1x)/v;
    b = abs(r2y-r1y)/v;
    c = abs(r2z-r1z)/v;
    time(i)=t;
    t=t+0.001;
    rt = (3-rcap*rcap)/(1-rcap*rcap);
    r1x = (sqrt(6/dt)*nx + rt*(r2x-r1x)/v)*dt + r1x;
    r1y = (sqrt(6/dt)*ny + rt*(r2y-r1y)/v)*dt + r1y;
    r1z = (sqrt(6/dt)*nz + rt*(r2z-r1z)/v)*dt + r1z;

    r2x = (sqrt(6/dt)*n_x - rt*(r2x-r1x)/v)*dt + r2x;
    r2y = (sqrt(6/dt)*n_y - rt*(r2x-r1x)/v)*dt + r2y;
    r2z = (sqrt(6/dt)*n_z - rt*(r2x-r1x)/v)*dt + r2z;
    Rend(i) = sqrt((r2x-r1x)*(r2x-r1x) + (r2y-r1y)*(r2y-r1y) + (r2z-r1z)*(r2z-r1z));
    rms= rms + Rend(i)*Rend(i);
end
rms=sqrt(rms/10000000);
plot(time,Rend,time,rms)
```

Rend(R2-R1) vs T



RMS Value :- 31.9504

Thank you