BD SIMULATION

End Term Evaluation

ARMEET LUTHRA

Assignment 1

Problem statement :

In this assignment we were required to compare the analytical and numerically calculated velocity of a freely falling sphere as a function of time, in a medium of given density and viscosity. Plots by both the methods were compared.

Physics and equations :

- A freely falling spherical ball experiences three forces:
- Gravitational force downwards,
- Buoyancy upwards,
- Drag force, which acts opposite to the direction of velocity of the particle

Assignment 1

- The equation of motion can be written as: $m\,du/dt = mg - \rho V g - 6\pi r \eta u$.

  - m=mass of sphere
  - u=velocity of sphere
  - g=gravitational acceleration
  - ρ=density of fluid
  - V=volume of sphere
  - r=radius of sphere
  - η=viscosity of the fluid

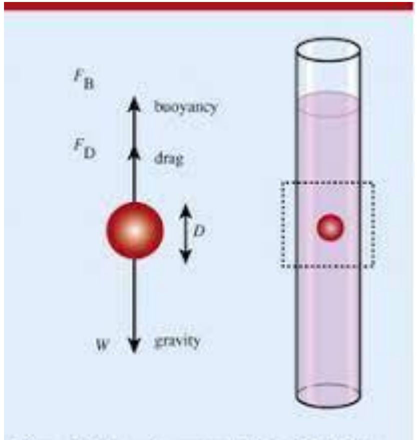


Figure 1. Schematic representation of the forces applied to the falling ball in the liquid.

Assignment 1

- The equation of motion can be written as: **$mdu/dt=mg-\rho Vg-6\pi r\eta u$.**

- Solving the differential equation would give the analytical expression and to get the analytical answer, boundary condition $u(t=0)=0$ can be used. Analytical expression:

$$u(t)=mg/(6\pi r\eta)*(1-\rho V/m)*[1-e^\wedge(6\pi r\eta t/m)]$$

- Numerical Method involves linear approximation: **$du/dt \simeq [u(t+\Delta t)-u(t)]/\Delta t$**

  $u(t+\Delta t)=g\Delta t(1-\rho V/m) +u (t)[1+6\pi r\eta\Delta t/m]$

- This equation is then solved iteratively using a loop and the velocity at different time stamps is stored in an array.
- This data is plotted.

Matlab Code

```
a=input("Enter upto what time do you want the analysis ")
n=input("How many intervals do you want to divide the time into? ")
d= a/(n+1);
i=1;
v1(1)=0;
vel(1)=0;

for t=d:d:a
    v1(i+1)=v1(i)+ d*( (( 7050/8050)*9.8)-((9*v1(i))/(2*10^(-7)*8050)) );
    vel(i+1)=analytical(t);
    i=i+1;

end
t= 0:d:a;
plot(t,v1,'r--',t,vel,'g-');
xlabel("time");
ylabel("velocity");
legend('numerical','Analytical');

function vel = analytical(t)
vel= 7050*(2/9)*10^(-7)*9.8*(1-exp(  (-9/2)*(10^7)*t/8050 ));
end
```

Matlab Plot

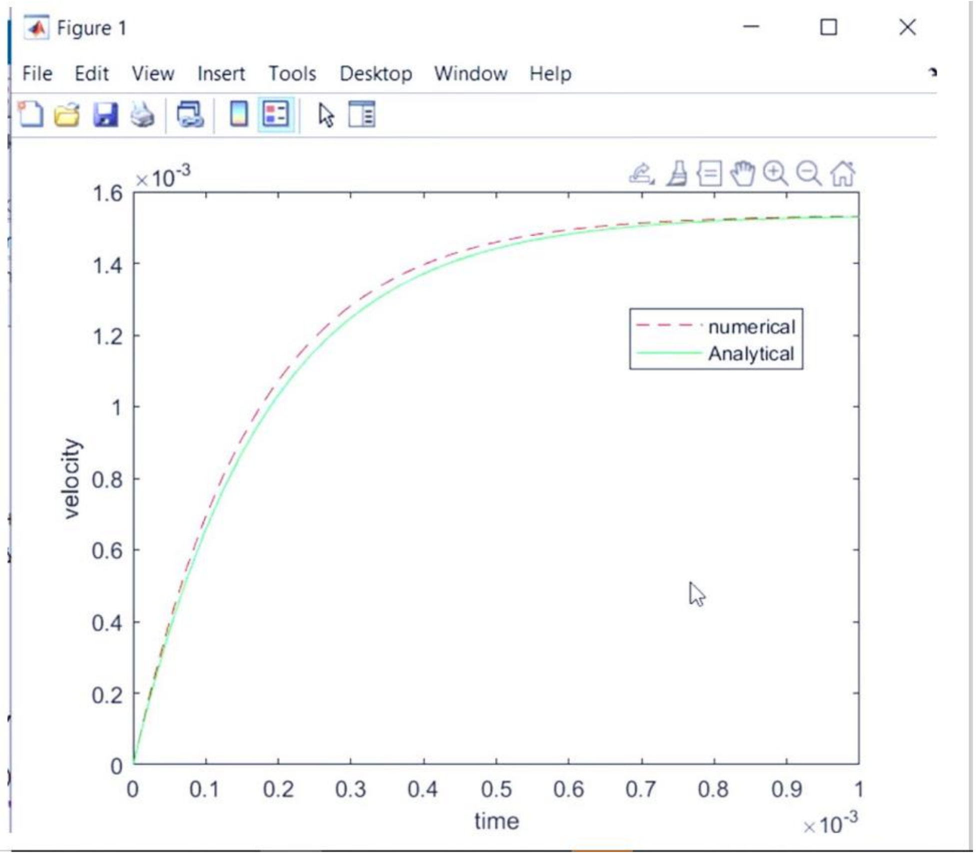


Figure 1
File Edit View Insert Tools Desktop Window Help
×10^-3
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
velocity
numerical
Analytical
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
time
×10^-3

Brownian Motion

- For the case of a free falling particle under gravity, the force driving the motion was in a fixed direction: downwards.

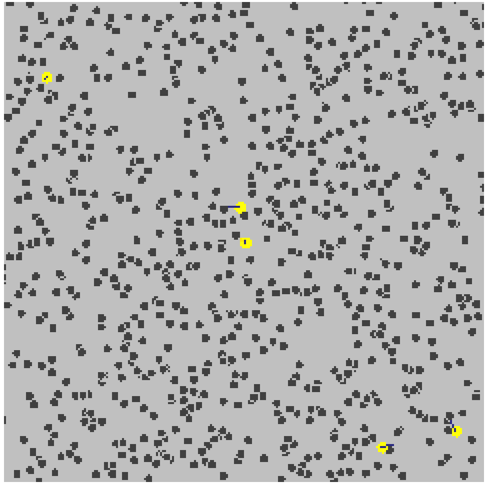
- But for mesoscale level particles in a colloidal solution, a random zig-zag motion is observed, known as brownian motion.

- Here the driving force is the brownian force, which originates due to random collisions of medium particles with the brownian particles.

- Pointing direction of force is random at each moment.

Brownian Motion

Source:https://en.wikipedia.org/wiki/Brownian_motion



Equation of Motion

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

Since particle is extremely small therefore it's inertia(mass)

Is neglected.

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

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

Non dimensionalising the equation we get:

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

Forces on a Brownian Particle

- **Drag Force:** Force exerted by the medium which opposes the relative motion of brownian particle.

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

- **Brownian Force:** Force originating due to random collision of medium molecules with the brownian particles

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

- **Gravitational Force:** Due to extremely small size of brownian particles, force due to gravity is negligible in comparison to the brownian force, hence neglected.

Assignment 2

<u>Problem Statement:</u> Assuming particle is at origin(0,0,0) at t*=0, it was asked to plot x* vs y*, and MSD vs τ*, and from this plot calculate diffusivity.

- We took , $\Delta t^* = 0.001$ and used the finite difference method iteratively , as applied in previous assignment to get x*,y*,z* at time intervals multiple of $\Delta t^* = 0.001$ till t*=100sec.

  x*(t+Δt*)=sqrt(6Δt*)nx+x*(t)

- Then we calculated MSD(τ) for τ= 0, Δt*,2Δt*,3Δt*,4Δt*,5Δt*...... 10^4Δt*.

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

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

Assignment 2

<u>Stokes-Einstein diffusivity:</u>

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

Re-look at the time scale:

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

$$\mathrm{MSD} = 2nDt.$$

- We were then required to calculate Diffusivity value which was done by plotting MSD vs τ^*,and then fitting a line through 5 consecutive points from τ^* =0 using polyfit function.

  Diffusivity = (slope of MSD vs τ^*)/6=

```
diffusivity =

    0.9831
```

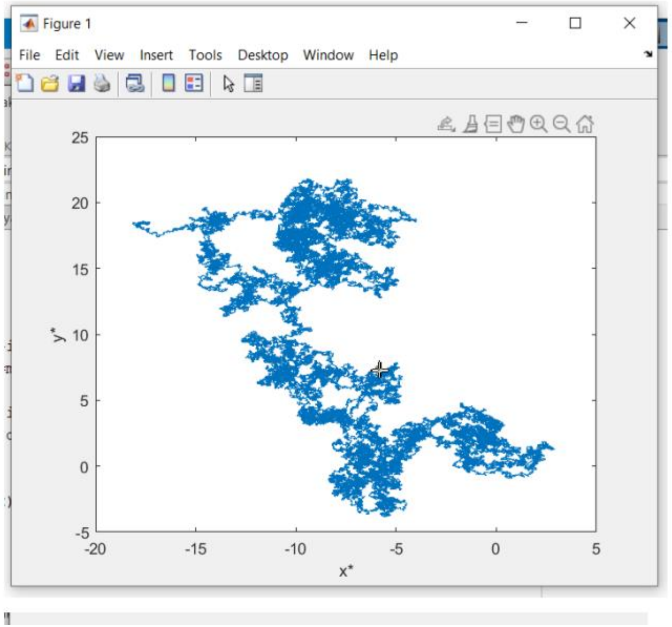
Matlab Code

```
clc
clear
delt=0.001;
n=10^5;
r(1,1)=0;
r(1,2)=0;
r(1,3)=0;
m=10^4;
msd=zeros(m,1);
fileid=fopen('diya.txt','w');
fprintf(fileid,"MSD        delt* \n");

for i= 1:n
    uvector=2.*rand(1,3)-[1,1,1];
    uvector=uvector./norm(uvector);
    r(i+1,:)=sqrt(6*delt).*uvector+r(i,:);

end

for i=1:m

    for j= 1:m+1-i
        msd(i,1)=msd(i,1)+sum((r(j+i,:)-r(j,:)).^2);
    end
    msd(i,1)=msd(i,1)./(m+1-i);
    xaxis(i,1)=i*delt;
end
```

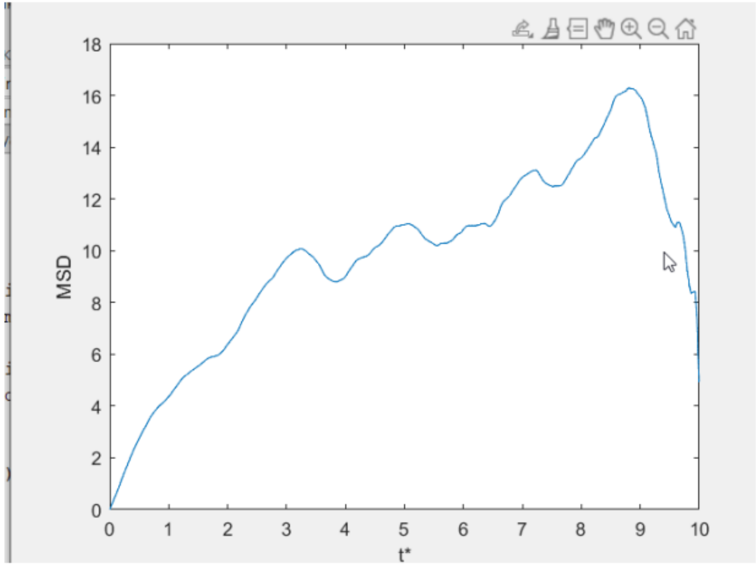
```
figure(1)
plot(r(:,1),r(:,2));
xlabel("x*");
ylabel("y*");


figure(2)

plot([0;xaxis],[0;msd]);
xlabel("t*");
ylabel("MSD");
P=polyfit([0;xaxis(1:5)],[0;msd(1:5)],1);
msd(:,2)=xaxis(:,1);
fprintf(fileid,"%f %f\n",msd');
fclose(fileid);
diffusivity=P(1,1)/6
```

Matlab Plots

x^* vs y^*



MSD vs τ^*



Assignment 3

Problem statement :

In this assignment we were required to plot the dimensionless distance between the 2 beads of the dumbbell model of a single polymer as a function of dimensionless time and also find the rms value of the separation between the beads.

Physics and equations :

- Polymer chains practically involve around a million or more particles
- It is computationally difficult to apply equations of motion individually to each particle and solve them
- We therefore model our polymer chain as a dumbbell with 2 beads

Assignment 3

- Three forces act on each bead: Drag force, brownian force and spring force.

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

v: number of Kuhn lengths mimicked by one spring
b_K: Kuhn length of the polymer chain

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

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

- Equation of motion

Equation of Motion (i = 1, 2)

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

Assignment 3

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

- Non-dimensionalising, we obtain EOS:

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

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

Select a length scale equal to Kuhn step: b_K

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

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

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

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

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

Assignment 3

- We solve the dimensionless equation of motions, using similar approach as done in previous 2 assignments.
- We then calculate the end to end distance between the 2 beads for different time stamps
- RMS value of this distance is calculated.

```
RMS VALUE IS :
    21.1698
```

- We plot the dimensionless end to end

  distance vs dimensionless time

Matlab Code

```
clc
clear
v=500;
delt=0.001;
n=10^7;
r1=zeros(3,1);
r2=zeros(3,1);
r2(1,1)=sqrt(500);
rms=0;
fileid=fopen('diya.txt','w');
fprintf(fileid,"Rend*         delt* \n");

for i= 1:n
    uvectorl=2*rand(3,1)-1;
    uvectorl=uvectorl/norm(uvectorl);
    uvector2=2*rand(3,1)-1;
    uvector2=uvector2/norm(uvector2);
    Rend=r2(:,i)-r1(:,i);
    normi(i)=norm(Rend);
    rms=rms+normi(i)^2;
    Rend2=(normi(i)/v)^2;
    S=delt*(3-Rend2)/((1-Rend2)*v);
    r1(:,i+1)=r1(:,i)+ sqrt(6*delt)*uvector1  +  S*Rend;
    r2(:,i+1)=r2(:,i)+ sqrt(6*delt)*uvector2  -  S*Rend;

end
Rend(:,n+1)=r2(:,n+1)-r1(:,n+1);
normi(n+1)=norm(Rend(:,n+1));
rms=sqrt((rms+normi(n+1)^2)/(n+1));
disp("RMS VALUE IS :  ")
disp(rms);
time=[0:0.001:n*delt];
rmsv=repelem(rms,n+1);

figure(1)
plot(  time,normi,'b-',time,rmsv,'k-');
legend("Rend*","RMS value");
xlabel("t*");
ylabel("Rend*");


fprintf(fileid,"%f    %f\n",[normi;time]);
fclose(fileid);
```

Matlab Plots

Plot