ID-2023

Assignment 2: Droplet simulation

Name – Gautam Kumar Mahar
Roll No. – 2103114
Branch – Computer Science Engineering

Question 1

Write the Fortran code for a droplet (R=100 micrometre) coming out of sneezing from a Covid-infected person having a height of 2 meters. Assume that only the drag force is acting on the droplet. Assume that initially, the droplet has the velocity of Vx(0) = 25 m/s, Vy(0) = 0, Vz(0) = 0. Compute the horizontal distance the droplet will travel and the time it stays in the air if it settles on the ground. Choose 1 Lakh particles. Use dt = 0.0001 sec.

The image below displays the time spent falling overall and the distance travelled in the X direction. The code used to calculate these data is also included.

ASSIGNMENT_1_A.f90 | ASSIGNMENT_1_B.f90 | ASSIGNMENT_2_1.f90 | ASSIGNMENT_1_A | graph.py

ASSIGNMENT_2_1.f90

```
1   program ASSIGNMENT_2_1
2       implicit none
3       double precision, dimension(100000) :: X, Y, Z, Vel_X, Vel_Y, Vel_Z, T
4       real, parameter :: H = 2.00, V = 25.00, dt = 0.0001, pi = 3.14159, M = 4.1887e-9, lambda =33.291e-9
5       integer :: i
6       do i = 1, 100000
7           X(i) = 0.0
8           Y(i) = 0.0
9           Z(i) = H
10          ! l(i) = 0.0
11          Vel_X(i) = V
12          Vel_Y(i) = 0.0
13          Vel_Z(i) = 0.0
14          T(i) = 0.0
15      end do
16
17      do i = 1, 100000
18          do while(Z(i)>0)
19              X(i) = X(i) + Vel_X(i)*dt
20              Y(i) = Y(i) + Vel_Y(i)*dt
21              Z(i) = Z(i) + Vel_Z(i)*dt
22
23              Vel_X(i) = Vel_X(i) - (lambda/M)*Vel_X(i)*dt
24              Vel_Y(i) = Vel_Y(i) - (lambda/M)*Vel_Y(i)*dt
25              Vel_Z(i) = Vel_Z(i) - (9.81)*dt - (lambda/M)*Vel_Z(i)*dt
26
27              T(i) = T(i) + dt
28          end do
29      end do
30      open(unit = 3, file = 'Velocities.dat', status = 'replace')
31      open(unit = 4, file = 'Positions.dat', status = 'replace')
32      do i = 1, 100000
33          write(3,*) T(i), ", ", Vel_X(i), ", ", -1*Vel_Z(i)
34          write(4,*) T(i), ",", X(i), ",", Z(i)
35      end do
36      print *, "THE HORIZONTAL DISTANCE CALCULATED IS :", X(59)
37      print *, "THE TOTAL TIME TAKEN IS : ",T(59)
38
39
40  end program ASSIGNMENT_2_1
```

Output -

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
cd "/home/gautamop/Desktop/Fortran/" && gfortran ASSIGNMENT_2_1.f90 -o ASSIGNMENT_2_1 && "/home/gautamop/Desktop/Fortran/"ASSIGNMENT_2_1
┌──(gautamop㉿gautamop)-[~/Desktop/Fortran]
└─$ cd "/home/gautamop/Desktop/Fortran/" && gfortran ASSIGNMENT_2_1.f90 -o ASSIGNMENT_2_1 && "/home/gautamop/Desktop/Fortran/"ASSIGNMENT_2_1
 THE HORIZONTAL DISTANCE CALCULATED IS :   3.1455170179730905
 THE TOTAL TIME TAKEN IS :    1.7461999558872776

┌──(gautamop㉿gautamop)-[~/Desktop/Fortran]
└─$
```

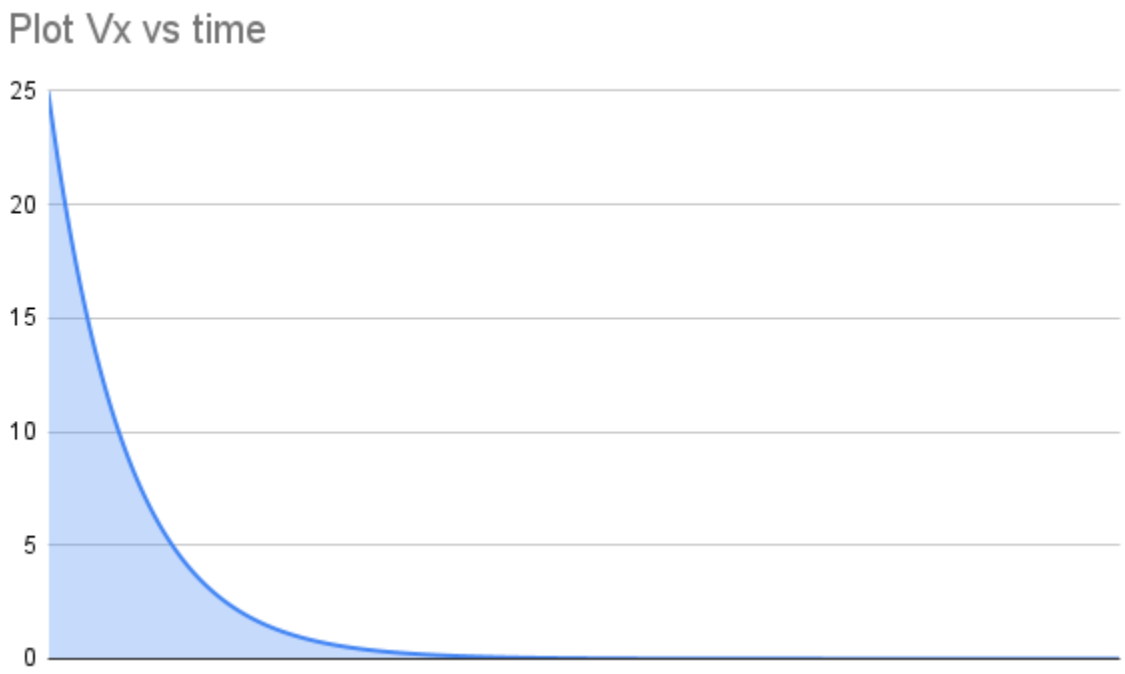
Here is the code for retrieving the data before we plot the graphs.

ASSIGNMENT_1_A.f90 | ASSIGNMENT_1_B.f90 | ASSIGNMENT_2_1.f90 | ASSIGNMENT_2_2.f90 | Velocities_part2.csv | graph.py

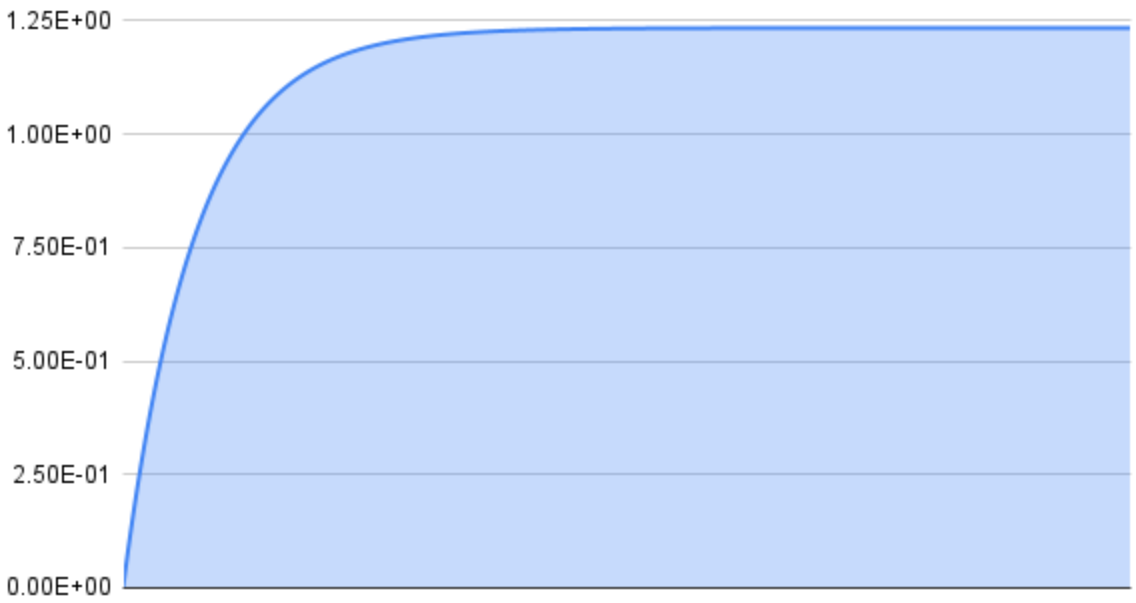
ASSIGNMENT_2_2.f90

```
1   program ASSIGNMENT_2_2
2       implicit none
3       double precision, dimension(100000) :: X, Y, Z, Vel_X, Vel_Y, Vel_Z, T
4       real, parameter :: H = 2.00, V = 25.00, dt = 0.0001, pi = 3.14159, M = 4.1887e-9, lambda =33.291e-9
5       integer, parameter :: i = 1
6
7           X(i) = 0.0
8           Y(i) = 0.0
9           Z(i) = H
10          Vel_X(i) = V
11          Vel_Y(i) = 0.0
12          Vel_Z(i) = 0.0
13          T(i) = 0.0
14
15          open(unit = 3, file = 'Velocities_part2.csv', status = 'replace')
16          open(unit = 4, file = 'Positions_part2.csv', status = 'replace')
17
18          do while(Z(i)>0)
19              X(i) = X(i) + Vel_X(i)*dt
20              Y(i) = Y(i) + Vel_Y(i)*dt
21              Z(i) = Z(i) + Vel_Z(i)*dt
22
23              Vel_X(i) = Vel_X(i) - (lambda/M)*Vel_X(i)*dt
24              Vel_Y(i) = Vel_Y(i) - (lambda/M)*Vel_Y(i)*dt
25              Vel_Z(i) = Vel_Z(i) - (9.81)*dt - (lambda/M)*Vel_Z(i)*dt
26
27              T(i) = T(i) + dt
28
29          write(3,*) T(i), ", ", Vel_X(i), ", ", -1*Vel_Z(i)
30          write(4,*) T(i), ",", X(i), ",", Z(i)
31      end do
32
33  end program ASSIGNMENT_2_2
```

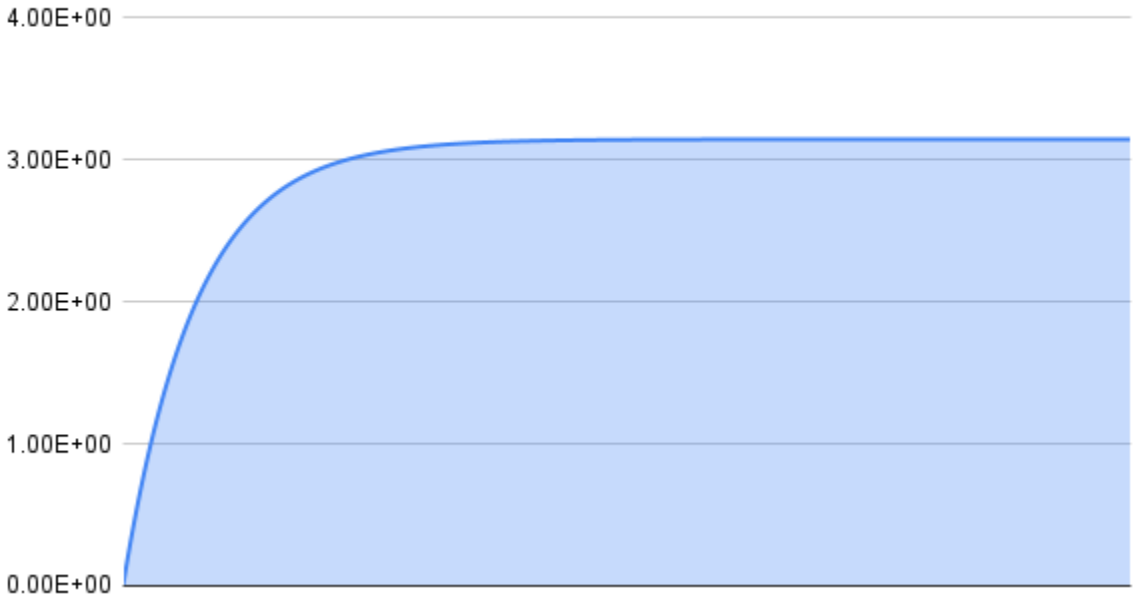
You may use Euler's method.
i) Plot Vx vs time
ii) Plot Vz vs time
iii) x vs time
iv) z vs time

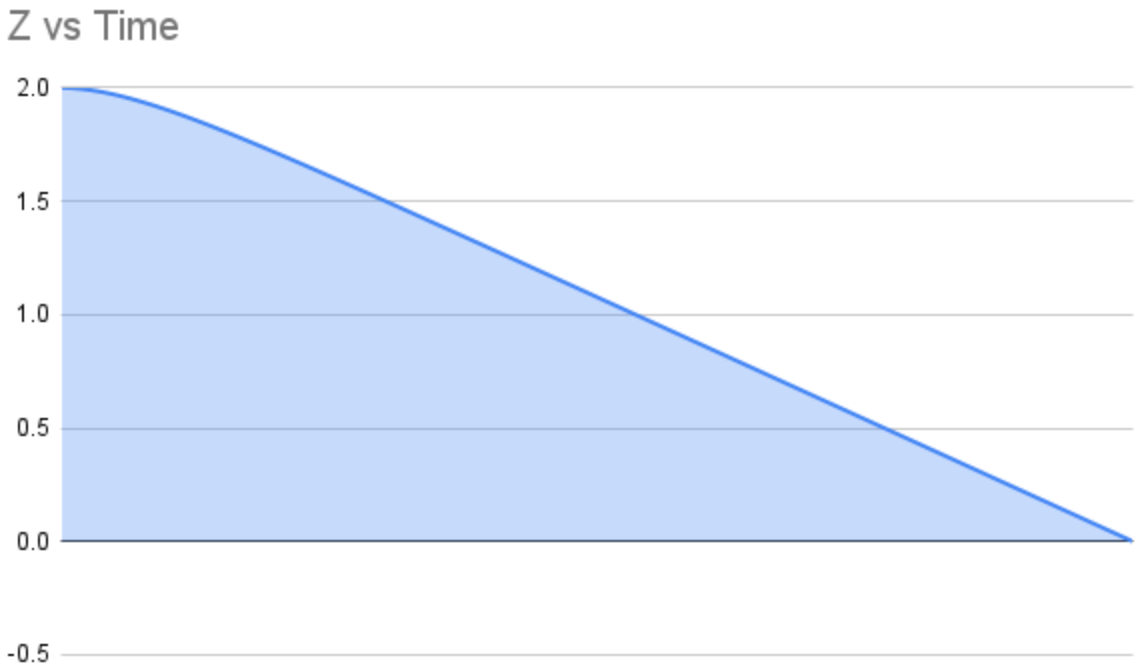


Plot Vz vs time



X vs Time





Thank You