

Computational Physics: Assignment 2

An object moving through a viscous medium experiences a force due to friction. In general, this force is quite complicated, but as a first approximation, we often say that the frictional force is

$$\vec{F}_f = -b v^n \hat{v}$$

Here the frictional force is in a direction that opposes the velocity. A simple case that we will consider is

$$\vec{F}_f = -b \vec{v}$$

The presence of this force makes a dramatic change in the manner that projectiles travel. In fact, being able to calculate where a projectile (like a cannon shell or nuclear warhead) has driven a wide range of studies. In the case of projectile motion, our vector equation is

$$m\vec{a} = -m\vec{g} - b\vec{v}$$

This problem separates into x and z motions:

$$ma_x = -bv_x$$

$$ma_z = -mg - bv_z$$

Since $a_x = \frac{dv_x}{dt}$ and $a_z = \frac{dv_z}{dt}$, both of these equations first order differential equations in the velocity components.

$$\frac{dv_x}{dt} = -\frac{b}{m}v_x$$

$$\frac{dv_z}{dt} = -g - \frac{b}{m}v_z$$

In this assignment, we wish to solve for and plot the velocity and position as a function of time. We wish to find the terminal velocity in each direction and how long it takes to get within 10% of that velocity.

Developing a simple algorithm

To develop a simple algorithm to solve for the velocities, we will begin with the definition of the

$$\frac{dv}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t}$$

If we consider a small enough Δt , we could replace the derivative with the delta expression.

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t}$$

Now we can use the $\frac{\Delta v}{\Delta t}$ to develop a simple algorithm to find the velocity as a function of time.

We'll begin by writing $\frac{\Delta v}{\Delta t}$ out in a more useful way. We will write out the expression in terms of the change of velocity over a change in time.

$$\frac{\Delta v}{\Delta t} = \frac{v(t + \Delta t) - v(t)}{\Delta t}$$

We can make this more useful if we write out time in terms of a number of Δt steps from the initial time.

$$t_i = t_0 + i \cdot \Delta t$$

$$t_{i+1} = t_0 + (i+1) \cdot \Delta t$$

$$\frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{\Delta t} = \frac{v(t_0 + (i+1) \cdot \Delta t) - v(t_0 + i \cdot \Delta t)}{\Delta t}$$

We can write out a very simple algorithm...

$$\frac{dv_x}{dt} = -\frac{b}{m}v_x$$

$$\frac{\Delta v_x}{\Delta t} = -\frac{b}{m}v_x$$

$$\frac{v_x(t_{i+1}) - v_x(t_i)}{\Delta t} = -\frac{b}{m}v_x(t_i)$$

$$v_x(t_{i+1}) = v_x(t_i) - \frac{b}{m}v_x(t_i) \cdot \Delta t$$

$$\frac{dv_z}{dt} = -g - \frac{b}{m}v_z$$

$$\frac{\Delta v_z}{\Delta t} = -g - \frac{b}{m}v_z$$

$$\frac{v_z(t_{i+1}) - v_z(t_i)}{\Delta t} = -g - \frac{b}{m}v_z(t_i)$$

$$v_z(t_{i+1}) = v_z(t_i) - g \cdot \Delta t - \frac{b}{m}v_z(t_i) \cdot \Delta t$$

Given that we know the velocity at some starting time t_0 (usually $t_0=0$), we can calculate the velocity at a time $t_1 = t_0 + \Delta t$ that is slightly later. We then use this result to calculate the next velocity and so on.

How to do it:

Read in the initial conditions and parameters from a file or from the terminal. Initial conditions include the magnitude of the velocity, the angle, the mass, the time step, the number of time steps, and the friction constant b .

Construct a function that will solve compute the frictional force. In this way, you will be able to compute with a variety of frictional forces.

Use a simple do loop to step through and compute velocity at each time.

Write the results to files with the form:

```
open(unit=20,file='vx.dat',status='unknown')
open(unit=21, file='vz.dat',status = 'unknown')
write(20,10)t, vx
write (21,10)t, vz
10 format(2x, e14.8, 10x, e14.8)
close(unit=20, status='keep')
close(unit=21,status='keep')
```

Use xmgr/grace or some other tool to plot the results. Consider the cases where b is 0, small, and large.

Programming Suggestions:

Echo print inputs with units so that the user knows what he/she is calculating. You should also print to the terminal while you are printing to a file.

Check:

Design and carry out tests to show the program is correct. Does the trajectory show the correct behavior in the cases tested? Do you get linearly increasing z velocity with no friction? Does the x velocity remain constant?

How many time points are necessary for the solution to converge? Try computing the same trajectory using different time step/Number of step cases.

Finally...

Try using different masses. How does the terminal velocity depend on the mass? How does the time depend on the mass.

Try using frictional forces that go as the second and third power. Plot in each case, assuming the same initial velocity. Compare how the terminal velocity changes