PHYS2300: Assignment 5 - Modeling motion with Newton's Laws

Objective

This purpose of this exercise is to use numerical integration to solve Newton's force law, thus giving you the basic tools for modeling physical systems.

Prerequisites: Assignments 1, 2, 3, and 4. This activity uses Visual Python.

Introduction

Newton's second law tells us that an object's acceleration is directly proportional to the force we apply to it. Measure the force and we can determine the subsequent motion. This means solving the equation as a second order differential equation:

$$F = ma$$
$$a = \frac{d^2x}{dt^2} = \frac{F}{m}$$

In principle, if we can write down a formula for the force we can solve the equation to get acceleration as a function of time. In practice, however, very few physical systems allow for this type of analytical solution.

Enter the computer. We can break this up into two first order differential equations like this

$$\frac{dx}{dt} = v$$
$$\frac{dv}{dt} = \frac{F}{m}$$

If we make *dt* sufficiently small, we can approximate the solution to these equations with the following formula.

$$v = v_0 + adt$$
$$x = x_0 + vdt$$

We can determine what will happen next provided we have some initial idea of what is happening now. This is a task particularly well suited to a computer.

The Task

Given this mathematical framework, do the following:

1. Design an algorithm to calculate the 3-D projectile motion using this method. This is not the same as using the analytical projectile motion formula you've seen in physics class, but rather describing the force on an object and determining time step by time step what is happening to the position and velocity. The user will provide initial position (x, y, z) and initial velocities (v_x, v_y, v_z), the time step, and how long the simulation will run. Your

program will compute positions and velocities as a function of time and animate the results, stopping the animation with the object hits the ground.

2. Write a code in Visual Python to perform this simulation, animating a ball being thrown. In order to see the motion, you'll need to provide some scenery (maybe a field or backdrop of some kind, or some clouds, be as creative as you like!). Determine when to start and stop the simulation so that you can visualize the parabolic motion of the object.
3. This should be identical to the projectile motion equation, but with this method we can add in any forces we like. Air resistance is typically parameterized as

$$\vec{F}_D = -\frac{1}{2}\rho v C_D A \vec{v}$$

   Where ρ is the air density, v is the velocity, C_D is the drag coefficient (0.5 would be a good estimate), and A is the cross sectional area of the object. Note that this vector force points opposite of the direction of motion. Add this air resistance into your force equation and rerun the simulation.
4. Finally, animate two objects, one with air resistance and one without, to show as a comparison.

To hand in

Your final animation, showing the comparison between projectile motion with and without air resistance.