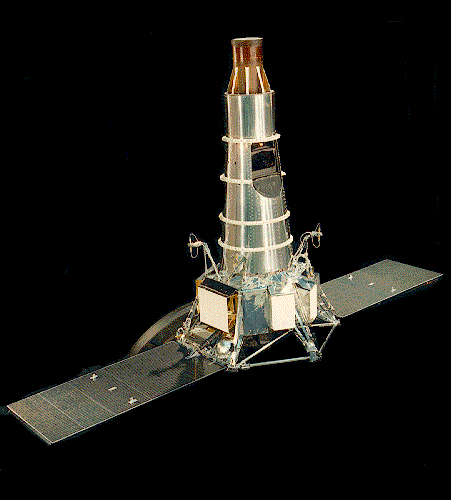
# [It took 7 tries for this mission to actually work.](http://nssdc.gsfc.nasa.gov/planetary/lunar/ranger.html)Lab 7: The Ranger 7 Mission

The first U.S. spacecraft to photograph the Moon close up was the unmanned “Ranger 7” mission, in 1964. The spacecraft, shown below, contained television cameras that transmitted close-up pictures of the Moon back to Earth as the craft approached the moon. The spacecraft had no way to slow down and eventually made a little crater of its own in the moon. The image at right is from the first successful mission, Ranger 7, about 17 minutes before impact.

[](http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1964-041A)To send a spacecraft to the Moon, we put it on top of a large rocket containing lots of rocket fuel and fire the rocket upward. Although the rocket moves quite slowly at first, it rapidly accelerates. As rocket fuel is used up, the corresponding sections of the rocket, called “stages”, fall away. As the mass which needs to be further accelerated decreases, the remaining propellant has an increasingly important effect. By the time the rocket reached the edge of the earth's atmosphere (about 50 km) the propellant is gone and the rocket coasts to the moon.

The Ranger series of missions is documented on NASA's website, [http://nssdc.gsfc.nasa.gov/](http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1964-041A). Top of Form

Bottom of Form

## Problem

I want you to model the mission from the starting point of about 50 km above the surface of the earth, where the ranger probe has an initial velocity of .

Compared to the gravitational force exerted by the earth and the moon, the other planets don't have a big effect on the probe's trajectory. Accordingly, to make the model simple at first, you should ignore the gravitational effects of the other planets. In addition, since the earth takes 365 days to go around the sun, you can assume that from the reference frame of the earth, the moon is basically stationary.

As a first step, write a program with python which models the trajectory of the ranger probe as it moves from the earth to the moon. Remember that the probe should stop moving when it reaches the surface of the moon. When you have your solution working, tackle the questions below.

## Tasks

1. By trial and error, or with physics (better), can you figure out what the *minimum* initial speed of the Ranger probe needs to be for it to reach the moon? What happens if the Ranger is launched at a speed slower than this? Related, where is the point of no return, after which the probe will “fall” towards the moon?
2. Make a graph of the ranger's velocity as a function of travel time. Does the velocity remain constant? How does the gravitational force from the earth and from the moon affect the probe's trajectory?
3. If you use an initial speed of 10% more than the minimum speed, how long does it take the Ranger probe to reach the moon? Given this time, is it reasonable to say that the moon and the earth are stationary with respect to each other?
4. Make plots of the probe's position, velocity, and net external force (from the moon and earth) as functions of time. Do these three quantities behave as you expect? Do the graphs show the moon “sucking in” the Ranger probe? From these graphs, describe where the moon's gravitational attraction becomes more important than the earth's. Top of Form

Bottom of Form

## Possibly useful information

The following data will probably be useful:

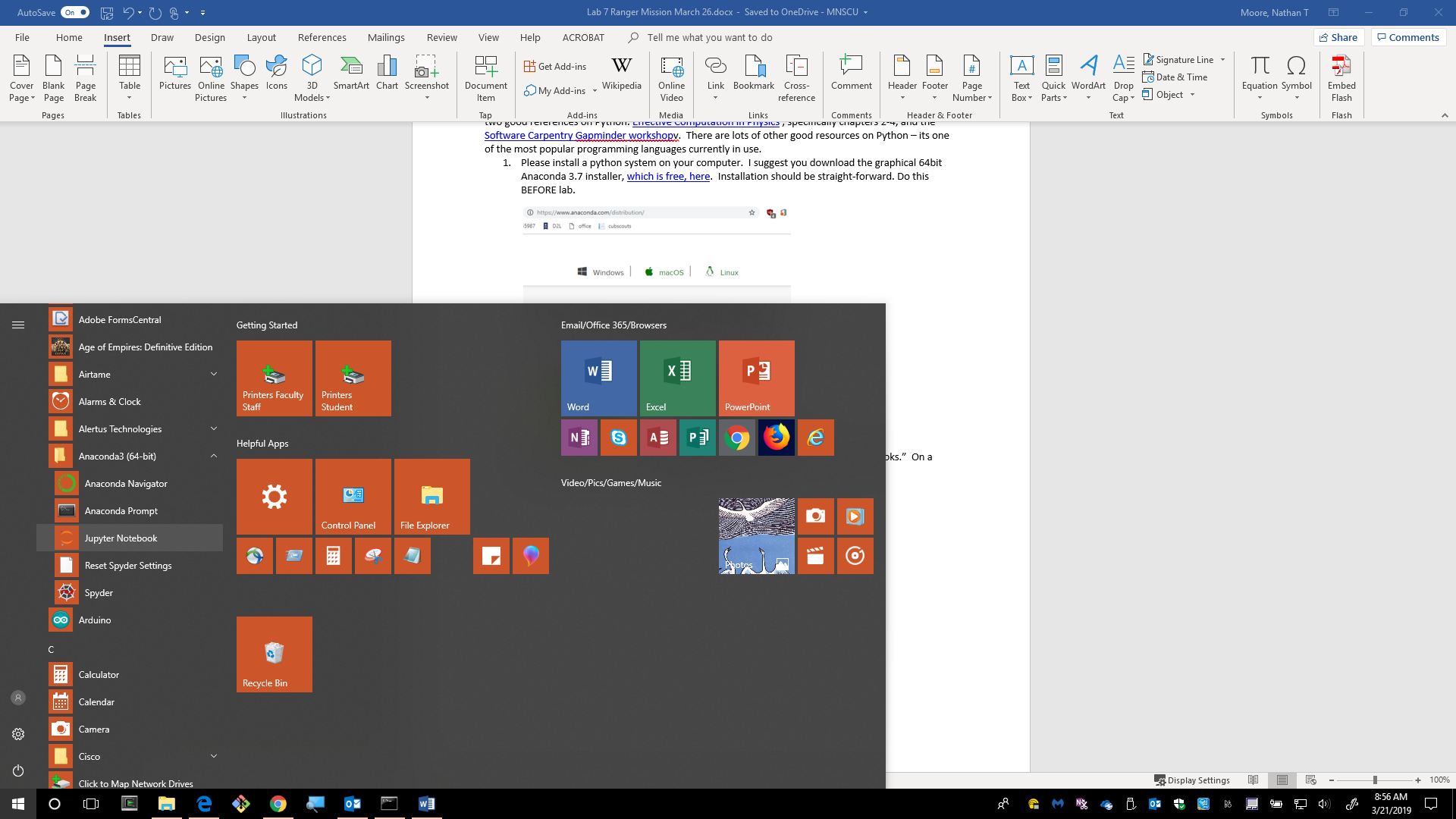
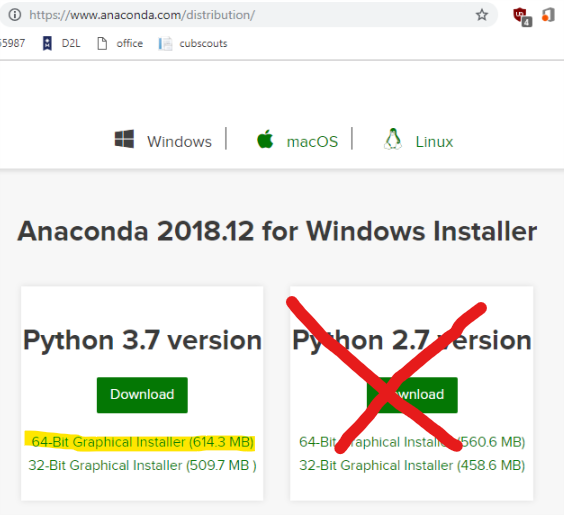
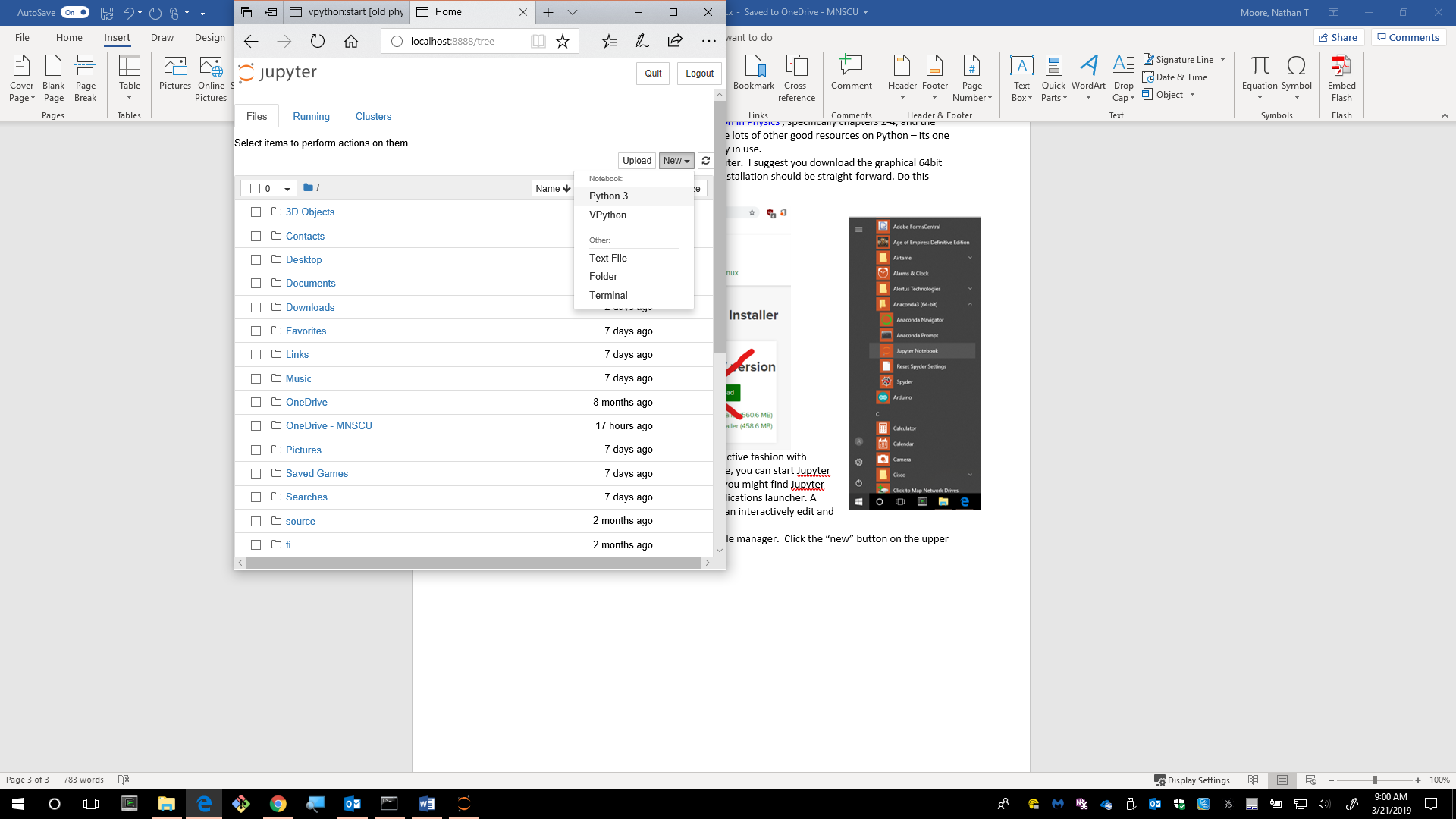
|  |  |
| --- | --- |
| mass of Ranger spacecraft |  |
| mass of Earth |  |
| mass of the moon |  |
| radius of the moon |  |
| radius of the earth |  |
| earth to moon distance[1)](http://localhost:8800/doku.php?id=phys221s09:lab:moon_shot&do=#fn__1) |  |
|  |

The Gravitational force between mass 1 and mass 2 is attractive. Make sure your mathematical form for force reflects this! Reminder,

## Prelab

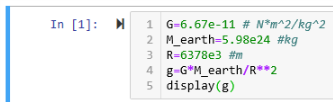
We’ll solve this problem by writing a computer program in the Python programming language. Here are two good references on Python: [Effective Computation in Physics](https://www.amazon.com/Effective-Computation-Physics-Research-Python/dp/1491901535) , specifically chapters 2-4, and the [Software Carpentry Gapminder workshop](https://swcarpentry.github.io/python-novice-gapminder/)v. There are lots of other good resources on Python – its one of the most popular programming languages currently in use.

### Installation

1. Please install a python system on your computer. I suggest you download the graphical 64bit Anaconda 3.7 installer, [which is free, here](https://www.anaconda.com/distribution/). Installation should be straight-forward. Do this BEFORE lab.
2. I normally write python programs in an interactive fashion with “Jupyter Notebooks.” On a Windows machine, you can start Jupyter from the start menu. On an Apple machine, you might find Jupyter inside of the Anaconda Navigator in your applications launcher. A Jupyter noebook is a webpage in which you can interactively edit and run python code.
3. When Jupyter is open, you’ll probably see a file manager. Click the “new” button on the upper right. I only work with Python 3 notebooks.

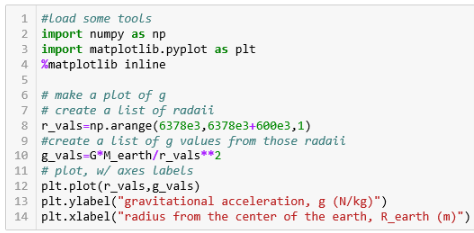
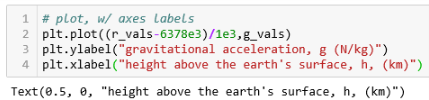
### Calculator Work

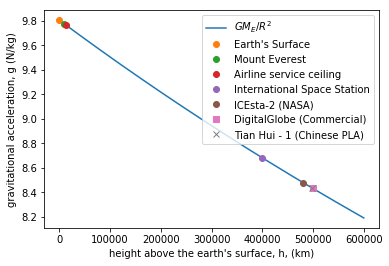
1. Once the notebook is open (in a new browser window), type out the following lines of code to compute the gravitational acceleration constant (from earth) on the surface of earth.



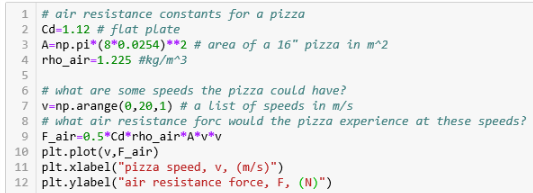
1. Once you’ve typed in these 5 lines of code you can run the commands by pressing shit+enter on your keyboard. You should get a result of .
2. So, you can treat python like a giant calculator. The altitude of the International Space station is 250 miles (400km) above earth’s surface. Write a few lines of python to compute g at the ISS.

### Plotting

1. Python is more than just a calculator though. It can also plot data. The following lines will plot a graph of g. Note, you have to first load a module (set of tools) to work with lists of data (numpy) and make plots (matplotlib). 
2. The graph that should result has axes in meters from the center of the earth. It would be nice to instead have the x-axis be kilometers from the surface of the earth. What do these lines produce? How does line 2 in this piece of code work? 
3. Finally, it would be nice to show the elevation of a few satellites on this plot. Its easy to get carried away at this stage. Here’s the code I used to make the attached plot. Satellite data from: https://www.wmo-sat.info/oscar/satellites : 



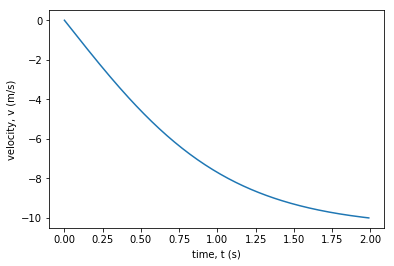
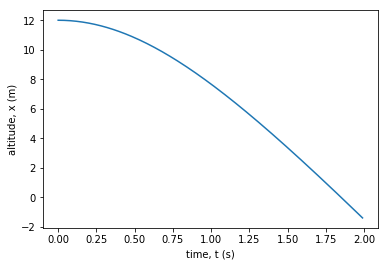
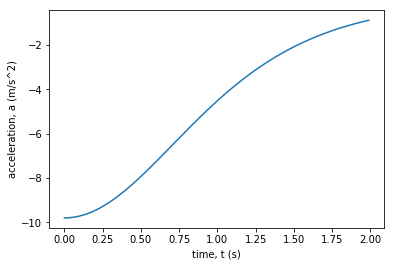
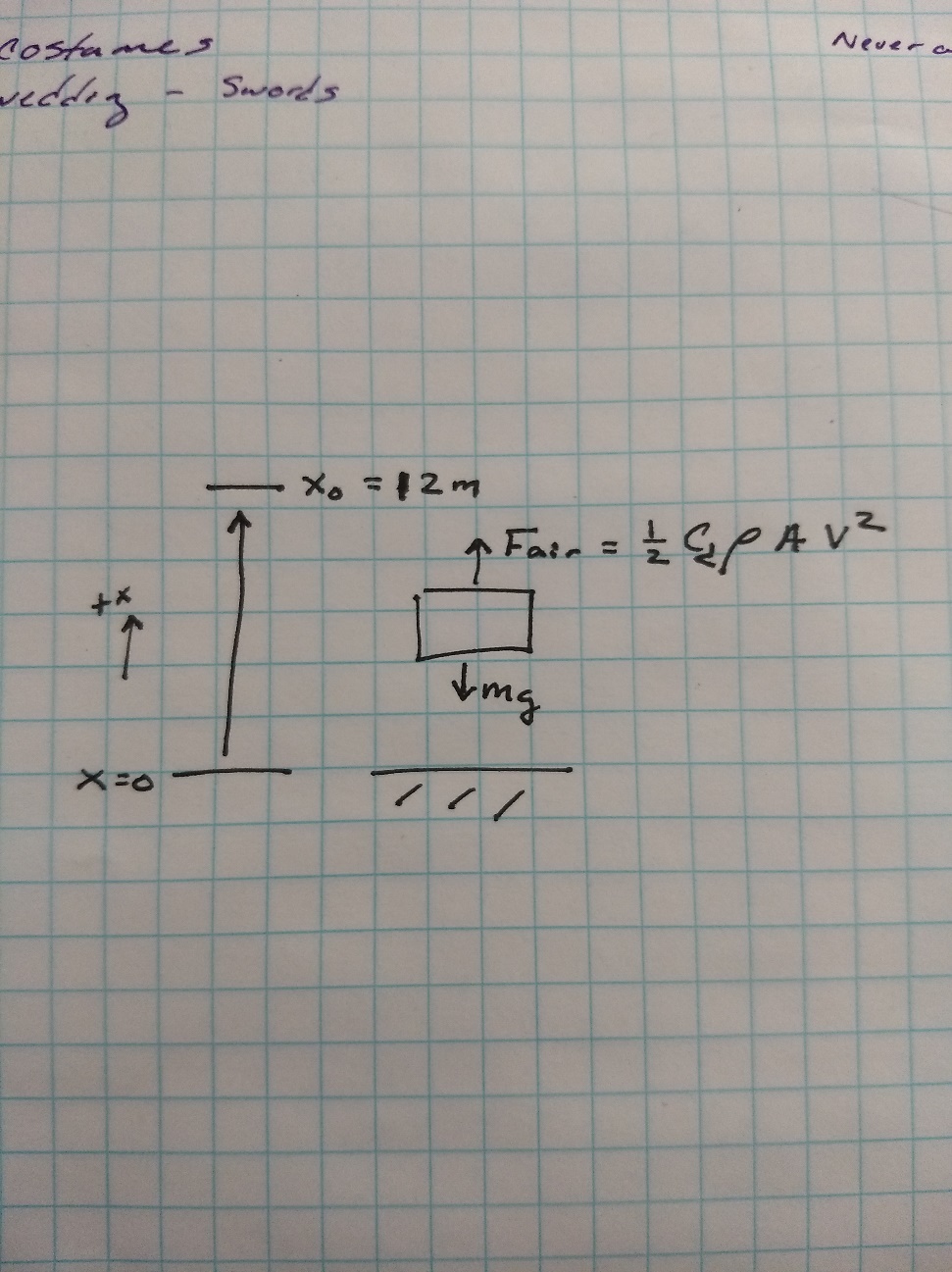
### Simulations

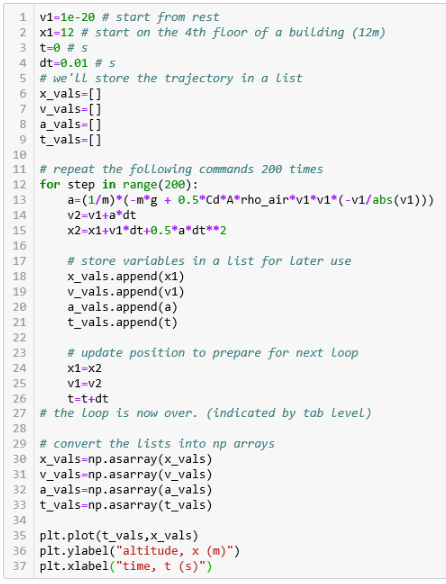
1. You can also use Python to simulate the motion of a moving object. Here’s an example from a few weeks ago. A cold pizza is dropped out of a window. The pizza is subject to air resistance force. . The values from class a few weeks ago are: 
2. Then, you can simulate the motion of the pizza (as it falls) with repeated use of kinematic equations over small time intervals:

,

, and,

.

The pizza’s acceleration isn’t constant over time, but if dt is small enough, repeated used of this approximation models the motion the pizza would exhibit. To make this work, we’ll use a “loop” that repeats a procedure. The code is on the next page. Here are the plots:



1. This force-velocity-position update routine is what we’ll use to simulate the Ranger-7’s trip to the moon. **The code above is available in the onedrive.** Download it from there and figure out how to make the following modifications:
   1. The pizza is dropped from the 10th floor, 30m above the ground. How long will it take to fall?
   2. The pizza is given an initial upward throw at How long will it take to hit the ground?
   3. How long does it take the pizza to reach 99% of its terminal velocity?
   4. The pizza is Chicago style, 3kg.
   5. The pizza takes 0.1s to come to a stop when it hits the ground. How big is the force it exerts on the ground?

## Lab Work

To solve the Ranger problem, I suggest you do the following things as a group on your whiteboard.

1. Draw a picture that includes the earth, moon and Ranger. Label distances. Select an origin and positive direction.
2. If the Ranger is at position x, what formula gives the distance from Moon to Ranger and from Earth to Ranger?
3. Draw an interaction picture and a force diagram for the Ranger probe. Express in detail.
4. Look back at the Pizza-drop from the prelab. What parts of that solution can you adapt to the Ranger problem?
5. Each person needs to create their own solution. Collaborative discussion is good but everyone needs to code this.

## Conclusion

Use your working simulation to answer the four questions (Tasks) at the start of the lab.