Physics 331 - Advanced Mechanics
Computational Exercise – Ch 2, Moving in a linear medium

We'll start in class; due with HW on Tuesday.

Use VPython to do the following for a 1-kg object:

(a) Use the Euler-Cromer method to find the solutions to the equations of motion for a projectile in a linear medium. Plot the trajectory.

- Do not use analytical solutions for this part! The point is to get practice using the Euler-Cromer method.
- Assume the projectile starts at the origin. Make it so that you can enter any initial speed and angle above. To start, use 8 m/s at 50° above horizontal (note: VPython expects angles to be in radians.)
- You can copy the file "Euler-Cromer.py" and modify it.
- Define *b* just above the while loop (so it'll be easier to later also loop over different *b* values.) To start, use the value $b = 0.1$ kg/s
- Make sure that the time steps are small enough.
- Change the condition in the while statement to be "not ball.y < 0:".
- When you've got that working, add an outer loop in which you step through values of b, from 0.1 kg/s to 2 kg/s.
    a. For this, you can insert a new while statement (while not b>2:). You'll need to indent the lines below it by hand. You'll also want to add, a line at the bottom of this loop to reset the ball's position and velocity and to increase b.

(b) To check your results for part (a), also calculate the trajectory using the analytical result (2.37) and compare with your previous results.

- To do this, create a second trail, (maybe call it trailanal since it represents the analytical trajectory). It should be given the same initial position as the other trail, but inside the loop, it's position should be vector(ball.x, y, ball.z) where, on the previous line you define y according to the analytical trajectory expression. With both traces plotted at the same time, you can see how well they compare.
- This calculation should use the same initial conditions and *b* as part (a).

(c) Calculate the three approximations for the range $R_0 = R_{\text{vac}}$, R_1, and R_2 for the trajectory. Once again, these results should change when you change the initial conditions or *b*.

- Now, add code after the nested loop to simply print the final x-component of the ball at the end of its flight as well as the theoretical R_0, R_1, and R_2.
- For what values of the parameter *b* do the different approximations work reasonably well?