**2/20 In Class Chapter 2**

Edited after class to reflect that we got through only the first 8 in class. We will pick up the rest starting Thursday.

Today we have enough MATLAB under our belts to do some real physics! Yay! For the parts that are pure Physics, please do them in groups of two or three at the white boards. Even the Structure Plans of the code could be good to put on the board. Then each person goes back to their computer and writes code.

1. The first example in section 2.4 is vertical motion under the influence of gravity and no air resistance. (vertical = 1D)

   Do you remember the kinematic equations? If so, just use them to check your answer(s). I want you to practice some Newton's Second Law and the Second Law as a DE!

   (a) Start with Newton's Second Law for an object starting from height zero (to match the book) and initial velocity, $v_0 = 60$m/s (to match the book.) Please start with using symbols, so, $v_0$. Choose positive to be up, negative down.

   (b) Integrate to get velocity as a function of time. Use either definite integrals, or use the given initial conditions (symbols!) to solve for $C$, the constant of integration.

   (c) Now that you have $v(t)$, integrate that to get $y(t)$. (I chose $y$ since it's vertical.

   (d) Check it with the instructor before you go on.

2. (Still at the board.) Write a Structure Plan (pseudo code) for how you would plot the motion ($y$ vs $t$) of a particle falling near the surface of the earth.

3. Now go back to your computers and see if you can write a script to plot the motion (position vs time) of a particle falling near the surface of the earth.

   Feel free to keep talking to each other. If you get stuck, or just want to check your answer, read Section 2.4. And you can ask me.

   It's entirely possible to do something that is correct and is not much like the book's version. If you do something different from the text, please show me!

4. Write the script the book uses. I think they call it `throw.m`

   Please get me to check it off before you go on.

5. Modify the code so that it accepts the initial velocity as input. Enter an initial velocity of 60 m/s, and your graph should look just like the book's. Try values of $v_0 = 0$ and $-60$ m/s. What changes? Is it what you expect each time? What does the $(-)$ in $-60$m/s mean?

6. Modify the code so that it accepts an initial height as input. Enter an initial height of 0m, initial velocity of 60m/s, and your graph should look just like the book's. Play around with several different values of initial height and velocity.

7. Add an input for the duration of the motion. Play with different values of all three: initial height, initial velocity, and duration. Save this script. Call it throwMod.m or something you can remember.

8. **Projectile Motion**: You can either start over, or copy that script and rename the copy something like projectile.m (Save the old, we will come back to it.) See if you can figure out how to make it a graph of 2D motion. I break it down to the following steps:

   (a) Your variable from the old code will be the vertical part. (The book used $s$, you may want to use $y$ now.) You will now need to add the horizontal component. Find or derive an expression for $x$ as a function of time for a projectile near the surface of the earth. For this problem, assume there is no air resistance.

   (b) Back to the code: You will also probably want to enter the initial velocity as a magnitude and direction. Do that.

   (c) How could you add the $x$ to your code? Try it. For now, plot $x$ vs $t$. Is the plot what you expect? If not, see if you can find your bug. If so, please let me check.

   (d) Plot $y$ vs $x$.

   (e) Check it with me.

   (f) Play with the initial conditions. This is one of the amazing things about the power of computers. Once you have this code, it is easy to visualize many different cases. One of the ways I will evaluate your code is to use test cases.

I cut this off–this is as far as we agreed to go. If you didn't get this far in class, please do it as homework!