Outline of ideas for presentation 1 coding options

Everyone who plans to code should code along with the following video BEFORE coming to class. It's only 20 minutes and teaches you how to do your first simple animations in Python.
https://www.youtube.com/watch?v=H0ZXCFYiIOo&list=PL4Sl1ZPMcTDX-cQkqRjsVD4HvaLVvJPE8&index=2

The method used in this animation is a variation of the Euler-Cromer Method. I found some pretty good videos explaining this method in greater detail. Each vid in the playlist is super short and the guy explaining it is also using glowscript just like us. Some examples in his vids are strikingly similar to stuff I expect you to understand.
https://www.youtube.com/playlist?list=PLdCdV2GBGyXOOutOEKggaZo1rCHtUYh-A

Code Option 1:
Simulate cotton ball falling under influence of gravity and drag. Use simulation to determine drag constant.
Tip: first make the code run properly *without* drag.
- Draw a super-thin box for the ground such that top of box is at $y = 0$.
- Specify initial height above ground, mass, and velocity vector of cotton ball.
- Inside a loop using a fairly large time step ($dt = 0.25$), code computes net force on ball.
- Use net force to complete Euler-Cromer method (update acceleration, velocity and position)
- Stop code when ball impacts ground at $y = 0$.
- Output data for $t, y, v_y, \& a_y$.
    - The following video may help getting data out of GlowScript and into Excel:
    -
- Confirm the code is working well by comparing the time for a 2.00 meter fall to what is predicted by constant acceleration kinematics. Note: at this point, you may have to verify this result with a smaller step size such as $dt = 0.01$.
- Once working, revert to $dt = 0.25$.
- Add in a drag force inside the loop.
    - Assume $F_{drag} = bv^2$ where b is a positive constant.
    - For now, assume $b = 9 \times 10^{-3}$…figure out the units I am assuming using the previous equation!
    - **Think:** your code must use a drag force *vector* directed opposite the direction of motion. What fragment of code outputs a direction (a 3D unit vector) opposite the direction of motion?

When you get to this point, you should be able to input initial height & velocity, mass, and the drag constant b.
You should be able to output data to Excel and make plots of $xt, v_y t, \& a_y t$.
When this is done, obtain experimental data for plots of $xt, v_y t, \& a_y t$.
You can probably get some from a different group to save time if someone already took that data.

Once you have both experimental & theoretical data, use trial and error to find the value for b which most accurately represents the experimental data.
- First verify you have a small step size such as $dt = 0.01$.
- Then, instead of comparing plots, compare experimental & simulation total times for a 2.00 meter fall. Run your sim several times with several different values of b until the total times match.
- Once times match well, output the data and compare your simulation data plots to the experimental plots.

Things you can discuss on your slides:
1. Your goal: to get the drag coefficient of falling cotton ball. The theoretical math is similar to workbook problem **2.58** with α changed to b. Ask me to briefly explain the physics & draw you an FBD to use in your talk and so you have something to say. Also briefly explain what terminal velocity is.
2. Explain how the Euler-Cromer method works.
3. The physics used in the code (i.e. compare how the physics looks on paper to how it looks in the code)
4. Include images of the code output to keep the slides from getting too boring.
5. If time permits, run the code with and without drag simultaneously so people can see if the effect is noticeable or not (might need to run sim at slow speed).
6. Show the plots of sim data versus exp data (ensure these adhere to formatting guidelines).
7. Compare several slopes on yt-plot to points on $v_y t$-plot.
8. Compare several slopes on $v_y t$-plot to points on $a_y t$-plot.
9. Compare an area under the $v_y t$-plot to a displacement on the yt-plot.
10. Finally, tell us the drag constant your sim predicts for the cotton ball.

Code Option 2:
Simulate cotton ball falling under influence of gravity and drag. Use simulation to determine drag constant.
Tip: first make the code run properly *without* drag.
- Draw a super-thin box for the ground such that top of box is at $y = 0$.
- Specify initial height above ground, mass, and velocity vector of cotton ball.
- Inside a loop using a fairly large time step ($dt = 0.25$), code computes net force on ball.
- Use net force to complete Euler-Cromer method (update acceleration, velocity and position)
- Stop code when ball impacts ground at $y = 0$.
- Output data for $t, y, v_y, \& a_y$.
    - The following video may help getting data out of GlowScript and into Excel:
    - https://www.youtube.com/watch?v=oAbYQWcBj1w&list=PL4Sl1ZPMcTDX-cQkqRjsVD4HvaLVvJPE8&index=19
- Confirm the code is working well by comparing the time for a 2.00 meter fall to what is predicted by constant acceleration kinematics. Note: at this point, you may have to verify this result with a smaller step size such as $dt = 0.01$.
- Once working, revert to $dt = 0.25$.
- Add in a drag force inside the loop.
    - Assume $F_{drag} = \alpha v$ where α is a positive constant.
    - For now, assume $\alpha = 1 \times 10^{-3}$...figure out the units I am assuming using the previous equation!
    - **Think:** your code must use a drag force *vector* directed opposite the direction of motion. What fragment of code outputs a direction (a 3D unit vector) opposite the direction of motion?

When you get to this point, you should be able to input initial height & velocity, mass, and the drag constant α.
You should be able to output data to Excel and make plots of $xt, v_y t, \& a_y t$.
When this is done, obtain experimental data for plots of $xt, v_y t, \& a_y t$.
You can probably get some from a different group to save time if someone already took that data.

Once you have both experimental & theoretical data, use trial and error to find the value for b which most accurately represents the experimental data.
- First verify you have a small step size such as $dt = 0.01$.
- Then, instead of comparing plots, compare experimental & simulation total times for a 2.00 meter fall. Run your sim several times with several different values of α until the total times match.
- Once times match well, output the data and compare your simulation data plots to the experimental plots.

Things you can discuss on your slides:
1. Your goal: to get the drag coefficient of falling cotton ball. The theoretical math is similar to workbook problem **2.57** with b changed to α. Ask me to briefly explain the physics & draw you an FBD to use in your talk and so you have something to say. Also briefly explain what terminal velocity is.
2. Explain how the Euler-Cromer method works.
3. The physics used in the code (i.e. compare how the physics looks on paper to how it looks in the code)
4. Include images of the code output to keep the slides from getting too boring.
5. If time permits, run the code with and without drag simultaneously so people can see if the effect is noticeable or not (might need to run sim at slow speed).
6. Show the plots of sim data versus exp data (ensure these adhere to formatting guidelines).
7. Compare several slopes on yt-plot to points on $v_y t$-plot.
8. Compare several slopes on $v_y t$-plot to points on $a_y t$-plot.
9. Compare an area under the $v_y t$-plot to a displacement on the yt-plot.
10. Finally, tell us the drag constant your sim predicts for the cotton ball.

Code Option 3:
Model the vertical oscillation of a ball attached to a spring released from rest at its unstretched length. Use this code to estimate the spring constant of a real spring.

Consider the figures at right. Input parameters include spring constant k, ball mass m, ball radius r, and unstretched length L_{un}. Initially assume $k = 5.00 \frac{N}{m}$, $m = 0.500$ kg, $r = 0.04$ m, and $L_{un} = 0.200$ m.

The key steps in this simulation is creating a loop which animates motion.
1. Start with a small time step such as $dt = 0.25$
2. Get current value of y.
3. Use this value to compute current value of $\vec{F}_{spring}$.
4. *Compute* NET force on ball
5. *Compute* acceleration of ball.
6. *Update* velocity of ball.
7. *Update* position of ball.
8. *Modify* the length of spring so it still appears attached.
9. *Modify* axis and position of the arrows representing
    a. $\vec{a} = \frac{\vec{F}_{NET}}{m}$
    b. $\vec{F}_{spring} = -ky\hat{\jmath}$
    c. $\vec{F}_{grav} = -mg\hat{\jmath}$
10. Increment time.
11. Stop the sim after three oscillations. The theoretical period of oscillations is

$$T = period = 2\pi\sqrt{\frac{m}{k}}$$

Once this is running, one must figure out how to output data for $t, y, v_y, \& a_y$. The following video may help getting data out of GlowScript and into Excel:
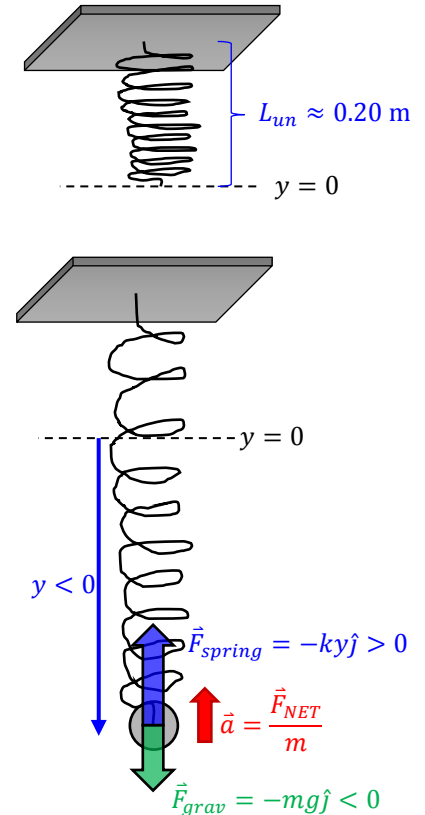
This is a good time to shrink your step size to $dt = 0.01$ so the data will be more precise.
If you've made it this far, you should be able to output data to Excel and make plots of $xt, v_yt, \& a_yt$.
When this is done, obtain experimental data for plots of $xt, v_yt, \& a_yt$.
You can probably get some from a different group to save time if someone already took that data.

Continues on the next page…

Once you have both experimental & theoretical data, determine the value for k which most accurately represents the experimental data.

- First verify you have a small step size such as $dt = 0.01$.
- Instead of comparing plots, use the experimental period & mass to determine the best value for k.
- Once times match well, output the data and compare your simulation data plots to the experimental plots.

Things you can discuss on your slides:
1. Your goal is to determine the spring constant. Ask me to briefly explain the physics & draw you an FBD to use in your talk and so you have something to say.
2. Explain how the Euler-Cromer method works.
3. The physics used in the code (i.e. compare how the physics looks on paper to how it looks in the code).
4. Include images of the code output to keep the slides from getting too boring.
5. If time permits, run the code with different values for k so people can see the effect on oscillations (might need to run sim at slow speed?).
6. Show the plots of sim data versus exp data (ensure these adhere to formatting guidelines).
7. Compare several slopes on yt-plot to points on $v_y t$-plot.
8. Compare several slopes on $v_y t$-plot to points on $a_y t$-plot.
9. Compare an area under the $v_y t$-plot to a displacement on the yt-plot.
10. Finally, tell us the spring constant your sim predicts.

This simulation is only valid for purely vertical motion.
When done, ask how one could generalize this code to accurately simulate physics in 3D.