Homework: Arrays — Skydiving

CSS 161—Fundamentals of Computing

Summary

You will write a program with the classes: `Skydiver`, `Skydive`, and `DiveDriver` that calculates the velocity of a skydiver as a function of time. The main program will be in the class `DiveDriver` and will include a loop that allows you to enter in the specifications for more than one jump. Feel free to adapt your code from HW: Classes.

Work Items

1. **Write the program** described below. Put the **three Java files** (*DiveDriver.java*, *Skydive.java*, and *Skydiver.java*) that are your classes **in a directory, zip up the directory,** then submit the zip file to your instructor.
   (a) Please submit only a single zip file via Canvas. Do not email your instructor your work.
   (b) Name your directory *LastFirst_HW_Arrays* and your zip file *LastFirst_HW_Arrays.zip*, where you put in your own last name and first name for “Last” and “First”. You **must name your directory and file this way.**
   (c) If you’re not sure how to create a zip file of a directory, please do a Google search to find out. The process to do so will be different depending on what operating system you’re using.
2. In Excel, **make a plot of velocity vs. time** for one of your skydive cases. Submit an image file of this plot (you can print/save the plot to an image format or scan a paper printout of your image).
   (a) Your image file can be PDF, JPG, TIF, or PNG.
   (b) Please submit only one image file via Canvas. Do not email your instructor your work.
   (c) Name your file *LastFirst_HW_Arrays.pdf* (or whatever is the extension for the graphic format youre using), where you put in your own last name and first name for “Last” and “First”. **You must name your file this way.**

Introduction

If you had physics before, you will recognize the problem of finding the velocity of a skydiver as similar to that of the fall of an object under the influence of gravity. However, unlike the case studied in most introductory physics courses, where air resistance is neglected, in the case of a skydiver, you cannot neglect air resistance. Thus, Newton's Second Law has two forces you have to account for:

$$F_{\text{tot}} = ma = F_{\text{gravity}} - F_{\text{drag}}$$

where m is the mass (kg) and a is the acceleration (m/s^2) of the skydiver. Note, in that in this and subsequent equations, we will assume that the positive x-direction (and thus the positive velocity and acceleration) points down towards the ground.

(As an aside, in a vacuum where we can neglect air resistance and gravity is the only force acting on the skydiver, we can neglect F_{drag} and the above simplifies:

$$F_{\text{tot}} = ma = F_{\text{gravity}} = mg \Rightarrow ma = mg \Rightarrow a = g$$

Meaning, the acceleration of the skydiver (a) equals the acceleration of gravity at the surface of the Earth ($g = 9.81\text{m/s}^2$). Which makes sense ☺.)

In the case where we have air resistance, in general we cannot solve for acceleration (and thus velocity) by hand, using just algebra or calculus. The reason is because air resistance is generally a function of velocity:

$$F_{\text{drag}} = \frac{1}{2} C A \rho_a v^2$$

where C is the drag coefficient (unitless), A is the cross-sectional area (m^2), ρ_a is the air density (kg/m^3), and v is the velocity (m/s). So, we have to solve this using a computer. How do we do this? Going back to Newton's Second Law, we can plug in for F_{drag} to find:

$$F_{\text{tot}} = ma = F_{\text{gravity}} - F_{\text{drag}} = mg - \frac{1}{2} C A \rho_a v^2$$

$$\therefore a = g - \frac{C \rho_a A}{2m} v^2$$

Note this means that the acceleration at any given moment in time is determined by the velocity at any given moment in time. That is:

$$a(t) = g - \frac{C \rho_a A}{2m} v(t)^2$$

(assuming g, C, ρ_a, A, and m are constant with time). But the velocity at any given moment in time depends on the acceleration a small bit of time before the current time. And that past acceleration depends on the past velocity, which in turn depends on the acceleration before that. And so on.

Here's another way to think of the problem. The velocity at time t, in general, can be given as:

$$v(t) = v(t - \Delta t) + a(t - \Delta t)\Delta t$$

where Δt is a small bit of time. Applying this to the specific case of the skydiver, we can substitute in for $a(t - \Delta t)$ using the $a(t)$ equation before it, namely:

$$v(t) = v(t - \Delta t) + \left(g - \frac{C \rho_a A}{2m} v(t - \Delta t)^2 \right) \Delta t$$

But what this equation tells us is that the velocity at time t depends on the velocity at time $t - \Delta t$. And the velocity at time $t - \Delta t$ depends on the velocity at time $t - 2\Delta t$, and so on. This is why we need to use the computer, because in order to calculate the velocity at the current time t you have to first calculate the velocities at all intermediate bits of time prior to t.

(As an aside, remember that notation like $v(t - \Delta t)^2$ does not mean "the velocity times the quantity t minus Δt, all squared" but rather "the square of the velocity evaluated at time $t - \Delta t$." That is to say, the notation $v(t)$ is not "v times t" but "the function v evaluated at t.")

Program Description

Your program should ask you to enter in the mass, cross-sectional area, and drag coefficient of the skydiver. Your program should also ask how long do you want to calculate the dive out to and what your timestep (Δt) will be. You can assume the density of air is 1.14 kg/m^3 and gravitational acceleration is 9.81 m/s^2.[1]

As your program does its calculations, it should save the time and velocities in an array (i.e., in two separate one-dimensional arrays, one for time and the other for velocities). One you have calculated all these values, write the output to a file. The file should have two columns, the first being the time (t) and the second being the velocity ($v(t)$) at that time. You can separate the two columns either by a space, tab, or comma; Excel can import files whose columns are separated by any of those characters. Your program, then, should also prompt you for the output filename.[2]

(If you're not sure how to use Excel to create a graph, do a Google search or see the "Miscellaneous Computing" page in the "Pages" section on the course web site. Importing a delimited text file into Excel usually means opening Excel and then choosing "Open" from the File menu. Excel should detect what kind of file it is and try to parse it into a spreadsheet grid format. Please see me if you have any questions.)

While the full output will go to a file, you should write out a few values to screen, so you can see how you're doing. Below is example console output:

```
Enter the mass of the skydiver (kg): 80
Enter the cross-sectional area of the skydiver (m^2): 1.035
Enter the drag coefficient of the skydiver: 0.581
Enter the ending time (sec): 16
Enter the time step (sec): 0.1
Enter the output filename: soln.csv
Writing out file.  Here are the first few lines:
0.100, 0.981
0.200, 1.9616
0.300, 2.9409
0.400, 3.9182
0.500, 4.8927
```

[1]Density of the atmosphere value from https://answers.yahoo.com/question/index?qid=20110201191027AADvsI0 (accessed September 6, 2014).

[2]You actually don't need arrays to solve this homework, but arrays are used in scientific computing all the time, and by asking you to write the code in this way, I'm hoping you'll get some practice manipulating arrays.

```
0.600, 5.8634
0.700, 6.8297
0.800, 7.7907
0.900, 8.7457
Enter another dive? (y/[n]): n
```

For the case whose plot you turn in, you'll want to calculate a substantial number of timesteps, out at least to $t = 16$ sec. Use a $\Delta t = 0.1$ sec, a mass of 80 kg, cross-sectional area of 1.035 m^2, and a drag coefficient of 0.581 (drag coefficient is unitless).[3] Your initial velocity, i.e., $v(t = 0 \text{ sec})$, should be 0 m/s. By the way, using these settings, your graph should show the skydiver achieves terminal velocity; at a certain time, the velocity shown on the graph should level off. If you aren't getting this behavior (or the sample output given above), something's wrong.

As in HW: Classes, I expect you to make appropriate choices about variable types and which variables are class variables and which are local to a method. The private data (attributes) of any class should only be the data directly associated with being that kind of object. Thus, the private attributes of `Skydiver` should only be the data associated with being a skydiver. Any other variables that help in performing the tasks (i.e., the methods) should be local (i.e., defined within the method). Also, don't forget to write accessor, mutator, and constructor methods, as appropriate.

Finally, there should be at least one method that you write that can be used to provide output for tracing variables: The method should be called `test`-something, e.g., `testStatistics`. Somewhere in your program, there should be a call to that method. In the code you submit, that call should be commented out, but I should be able to find it.

(If you're curious: This doesn't contribute to your grade, but try one case where Δt is quite a bit larger (and all the other parameters are the same), say $\Delta t = 5$ sec. What happens to your solution? Does this make physical sense? Any thoughts on why this occurs? We'll talk about this briefly in lecture. One hint: What you see is a classic example of what in scientific computing is called "numerical instability.")

About this Document

Course taught at the University of Washington Bothell. By Johnny Lin, Autumn 2014, with parts of the text from an assignment by Carol Zander, Spring 2013.

[3]Values for the skydiver from https://answers.yahoo.com/question/index?qid=20110201191027AADvsI0 (accessed September 6, 2014).