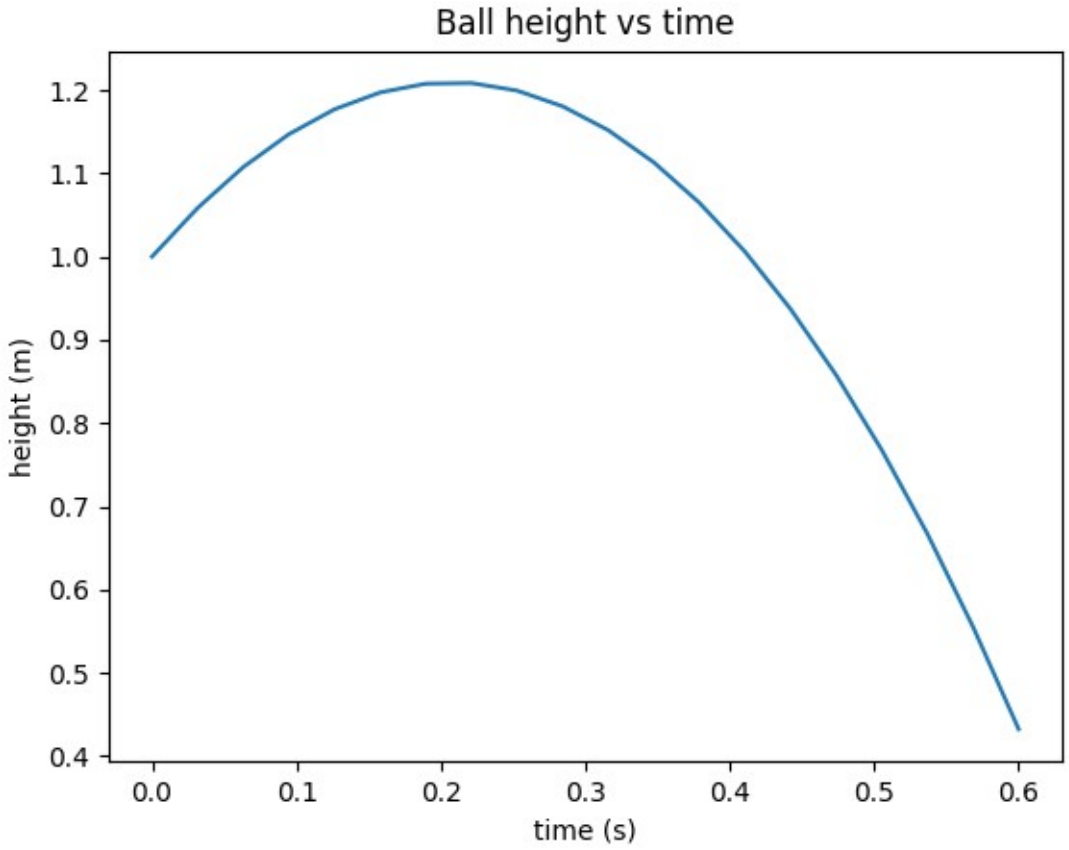
CS355
09-Mar-2022
Project 01
Noah Kaplan

Analysis

Part 01

Here is a graph.



I implemented a version of the code in C++ for comparison. The C++ code was compiled using g++ with the -O3 flag. I ran each implementation 4 times sequentially using the linux time command storing the results of the 2nd through 4th run. The user time results are shown below in seconds.

	Run2	Run3	Run4
C++	5.36	5.32	5.60
Python	7.36	7.49	7.31

The C++ version of the code consistently runs in shorter time than the python implementation, and the variation between runs is small enough that it is unlikely these differences are made up by system overhead.

The next rational question is whether this speed performance came at the cost of accuracy.

The final state of the python simulation was:

2.25684, -2.41745, 3.74205, -33.8215

And the final state of the C++ simulation was:

2.2575 -2.4188   3.7432 -33.8247

These numbers are comparable, and since we do not have an objective truth as to where the ball should have been at the final timestep, it is difficult to evaluate a runtime accuracy tradeoff, but it appears that the C++ code improves runtime performance without loss of accuracy.

Part 02

I used our vibrating string simulation to simulate the G string of a violin. I tuned the length of the string to be .3 meters, approximately the length of a violin string, and I tuned the frequency to be 196 hz, the frequency of a G.

The goal of the simulation is to predict the coupling constant c. I attempted to solve for this using Scipy optimization routines, but had little success. Most routines were prohibitively slow, or required derivative information about the period. The only routine that “worked” was a differential evolution, but this gave inconsistent results ranging between 1e9 and 1e11. Better results may be possible with a derivative function, or with a faster simulation.

The KTH Royal Institute of Technology collected some data on violin strings. They found that a G string has a linear density of 2.799g/m and a tension of 49.92 N. The mass of the string is then .3m*2.799e-3g/m = 8.397e-4kg. Our coupling constant, c, should be equal to the tension divided by the mass of the string = 49.92N/8.397e-4kg = 59449.8m/s^2. This is far from the results of the differential evolution. When I attempt to find the period using this predicted coupling constant, the simulation returns 1.19hz as the frequency, which is also a good deal away from the desired 192hz of a G. This may be accounted for by too coarse timesteps, too coarse field, or by our approximation of force as always being perpendicular downwards.

https://www.speech.kth.se/music/acviguit4/part4.pdf

Part 03

I implemented a damping factor into the vibrating string simulator.