Simulation Methods Project 1

Due: March 9, 2022

Part 1: Displaying simulation results

Take our thrown-ball simulation and use Matplotlib to visualize the results. The method of visualization is up to you, options include creating a 2-D plot of height against time, or a 3-D plot of position (x and y) over time, or an animation, or some other graphical representation of what happens to the ball.

For CS 355 students, you must make an effort to decrease the runtime of the simulation as much as possible while still maintaining sufficiently accurate results and obtaining the visualization results desired. It is up to you to define and justify what "sufficiently accurate" means for your simulation, explain the methods you used to increase the performance of the simulation, and report the performance improvements you were able to achieve.

$q = x,\ p = x'$

$w' = f(w) = (p, -q + \gamma p)$

Part 2: Justifying simulation parameters

$w'[0] = p$

$w'[1] = -q + \gamma p$

Choose any of the simulation examples we have worked on in class to date. Each of them has some simulation parameters: the strength of gravity, the strength of wind resistance, the arrival rate of customers to a checkout, etc. For the simulation that you have chosen, tune the simulation parameters to represent some real-world scenario, and justify the parameters chosen. A sufficient justification could involve a derivation from physical principles, or a citation of sources providing those constants, potentially with an explanation of why that source is a good fit for the real-world scenario that you've chosen.

For CS 355 students, you must also analyze the simulation results given those parameters, reflect on how closely they match observed reality, and discuss reasons that the simulation results might have deviated from observation.

Part 3: Adaptation

Choose any of the simulation examples we have worked on in class to date. You must add to or adapt that simulation to account for some new state of the system, and some new effect on the system. The new state could simply include adding another degree of freedom to the existing simulation (i.e. moving from a 2-D to a 3-D simulation) or it could involve tracking some property that was previously assumed constant or irrelevant (such as how much progress a person has made in checking out.) The new effect could include things like noting a ball should bounce when it reaches position 0, or that counters can be opened and closed in a checkout.