Lab 4 (Collision Detection)

Simulation and Modeling (CSCI 3010U)

Faisal Qureshi

Due back Oct. 20, 11:59 pm

Introduction

Earlier in the course we simulated a bouncing ball. The ball would fall under the effect of gravity, hit a floor and bounce back up. There we discovered that our ball was actually penetrating the floor. We can fix this problem by using ever smaller time steps. However, this approach doesn't work in practice. A better way is to *find the exact time of collision* and respond to the collision when it occurs.

Task

In this lab you are asked to develop (implement) binary search to find the exact collision time of a ball with a floor at height 0. We assume that our ball has mass 1 and that it is moving under the influence of gravity (no friction, please).

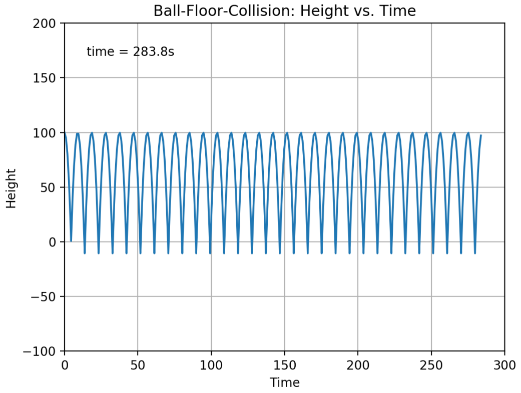
Distance tolerance tol_d

We know that the floor sits at height 0, and we are interested in finding the *exact* time at which the ball is located within distance tol_d from the floor.

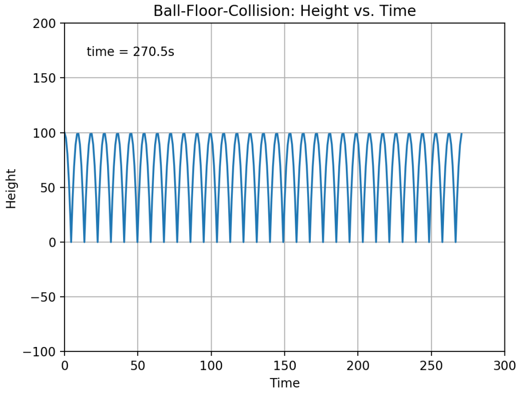
Why do we care about this tol_d

This tol_d enables us to control the accuracy vs. speed of our simulation. Use larger tolerance to speed up the simulation at the cost of accuracy, or lower the tolerance and get very accurate ball-floor-collisions (See figure below).

Using large tolerance



Using small tolerance



Coding

The primary function that you should focus on is

```
def respond_to_collision(self, state, t):
        return [0, -1*state[1]], t
```

This needs to be thoroughly revised.

Challenge

Extend this simulation to handle multiple balls falling from different heights with different initial velocities.

Submission

Via Blackboard.

- Python file
- 1 page report summarizing your experience with tolerances