VPython projectile motion simulation

M.H Jensen October 26, 2006, freshman at the University of Copenhagen (EU)

Survey comparing a VPython simulation with a moving particle in the gravitational field of the earth, special consideration will be given to comparing the simulation to the actual experiment, defining unstable parameters and determining the application range of the simulation.

Contents

1 The simulation

Our assignment this week was to fabricate a VPython simulation of a previously conducted experiment concerning a projectile motion in the gravitational field of the earth. We were to investigate and discuss the details of the simulation and compare them to the real experiment and find out how to use the collected data from the original experiment. It is important to have a basic understanding of the problem here. We only measured one varying parameter in the experiment, even though more exist and affects the outcome of the experiment. The simulation is therefore limited in its diversity to confirm the outcome of the first experiment. We therefore cant use the simulation to conclude anything contradictory from the original experiment. We can use the simulation as a rough estimation of the projectiles behaviour in the test environment, but as soon as one goes outside the intentional oprerating range for the model, it breaks down. If one were to compute a more complicated and reliable model, which has a broader operating range it would be obvious to include factors such as air resistance and amount of energy transfered from the spring to the ball. The simulation was made by inserting our statistical dispersion in the previously made projectile motion simulation. Since the ball was both deviating in the x and y axis a deviation in both directions were needed, and only one person from our workgroup had measured both deviation levels from the beginning of, so it was obvious to use that data set. The simulation code can be found at http://www.ylle.eu/simulation.py - the file also includes comments on the different sections of the code. We were forced to use a dataset that had measured both the x and the y axis during the last experiment, given that the simulation needs a deviation input in vector form, that accounts for both axis. We managed to identify the correct segment of the code that were responsible for gauss distribution. After inserting the values, not much else could be done since that was all the information we had from the last experiment. This leaves us with the above conclusion. Histograms of the simulation and the original experiment can be found at my science blog, http://www.new2n.blogspot.com http://www.fys.ku.dk/ mikkelr .