PHYS486-HW2: Realistic Projectile Motion

Nicholas Cemenenkoff

April 22, 2018

Abstract

I model the effects of quadratic drag on the motion of a horizontally shot projectile versus a falling projectile (with zero initial x-velocity) using the Runge-Kutta 2nd Order Method for solving ODEs. This simulation suggests that a launched projectile always has a longer time until collision than a dropped projectile. Specifically, this difference in collision times is dependent on the initial velocity of the launched object, characteristic parameters associated with the projectile's geometry, and the density of the fluid the projectile is traveling through. I show that error in the simulation is reduced by an appropriate choice of time step as well as the use of linear interpolation to approximate collision times.

1 Problem Statement

Consider two identical objects that start at the same height above the floor. One object is dropped, but the other is launched horizontally. Including air resistance, which object hits the ground first? Quantify how the time difference between the two objects hitting the ground (Δt) depends on launch speed (v_{x_0}), initial height (y_0), object mass (m), and drag coefficient (C_d). Reduce the dimensionality of the problem if possible, and explore approximately three orders of magnitude in each relevant dimension. Collapse any variables that have the same effect.

2 Numerical Solution

Quadratic drag (i.e. turbulent drag) is appropriate for this simulation because we are modeling trajectories at high velocity. Quadratic drag is given by $F_D(v) = \frac{1}{2}C_d\rho A v^2$, meaning the drag force is proportional to the square velocity of a projectile. The direct proportionality is determined by a unitless drag coefficient (which depends on the geometry of the object), the density of fluid in which the object is traveling, and the cross sectional area of the object. Essentially, $F_D(v) = Cv^2$. This form of the equation reduces all of the previous coefficients into one parameter (C) with units of $\frac{\text{kg}}{\text{m}}$. For motion involving both x- and y-components (given by the vector-valued position coordinate $\boldsymbol{r}$), Newton's second law gives us

$$\begin{aligned} m\ddot{\boldsymbol{r}} &= mg\hat{\boldsymbol{y}} - cv^2\hat{\boldsymbol{v}} \\ &= mg\hat{y} - cv\left(v_x\hat{\boldsymbol{x}} + v_y\hat{\boldsymbol{y}}\right), \end{aligned}$$

or rather

$$\frac{d\boldsymbol{v}}{dt} = \boldsymbol{g} - \left(\frac{C}{m}\right)v^2\left(\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right),$$

where $\boldsymbol{g} = -9.81\hat{\boldsymbol{y}}\,\frac{\text{m}}{\text{s}^2}$ is the acceleration due to gravity, $\frac{C}{m}$ is the drag parameter normalized by the mass of the projectile (which I call the object parameter) in units of m^{-1}, and $\boldsymbol{v}$ is the object velocity in $\frac{\text{m}}{\text{s}}$. This vector-valued differential equation is equivalent to a set of two coupled differential equations for the horizontal and vertical velocities of a projectile. The scalar x- and y-velocity equations then produce equations of motion for the range and height of a projectile, respectively, over time. Particularly,

$$\begin{aligned} \frac{dv_x}{dt} &= -\left(\frac{C}{m}\right)\sqrt{v_x^2 + v_y^2}\,v_x, \\ \text{and} \quad \frac{dv_y}{dt} &= -g - \left(\frac{C}{m}\right)\sqrt{v_x^2 + v_y^2}\,v_y. \end{aligned}$$

To solve this system, I employed the Runge-Kutta 2nd Order Method for solving ODEs. In general, this method is defined by the following iteration:

$$\begin{aligned} \text{Given} \quad & \frac{dy}{dt} = f\left(y, t\right), \\ \text{do} \quad & y_0 = y\left(t_0\right), \\ & y_{n+1} = y_n + f\left(y_n + f\left(y_n, t_n\right)\frac{\Delta t}{2}, t_n + \frac{\Delta t}{2}\right)\Delta t, \\ & t_{n+1} = t_n + \Delta t, \end{aligned}$$

where y is the modeled function and Δt is a discrete numerical time step (not to be confused with the time difference between the two objects hitting the ground, also labeled Δt).

3 Physical Analysis

In the following figures, the given $C/m = 2.88 \times 10^{-3}$ is characteristic of a spherical projectile with a cross-sectional area of $1.0\,\mathrm{m}^2$ moving through air (though there are conceivably differently shaped objects which have the same C/m ratio). Figure 1 shows height over time for the shot and falling projectiles. Notice that the time difference between collisions approaches a constant, representing the regime where both objects have reached terminal velocity. Next, note Figure 2 shows that as initial x-velocity increases, the time difference between collisions also increases, but does so at a decreasing rate. At extremely large initial velocities, the launched object will immediately begin to decelerate to terminal velocity, but this happens so quickly that Δt remains relatively stable over a wide range of v_{x0} values. Figure 3 shows that in the regime of large C/m, the launched object is so light (i.e. non-dense) that the majority of the projectile's initial x-velocity is immediately counteracted by drag. As such, the launched object behaves nearly identically to the falling projectile, hence the time difference between collisions for the shot and falling projectiles tends toward zero at a decreasing rate.

Summarizing, not only do we see that Δt increases as projectile density decreases, but Δt increases as initial launch velocity increases. This implies that given a tall enough initial height, the difference in collision times between the shot and falling projectiles will be nearly constant for high-velocity and high-C/m regimes, as confirmed by Figure 2 and Figure 3 respectively.

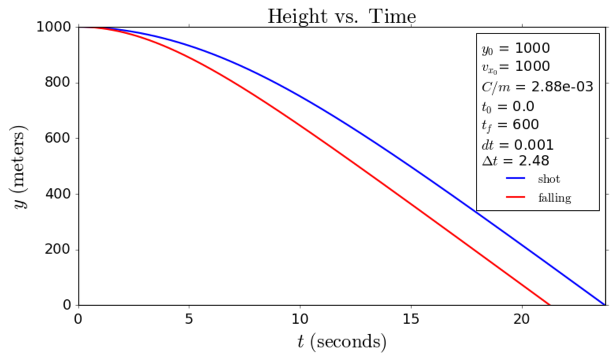

Figure 1: Note the falling object has a significantly reduced time until collision, its collision time differing from the collision time of the shot projectile by 0.49 s.

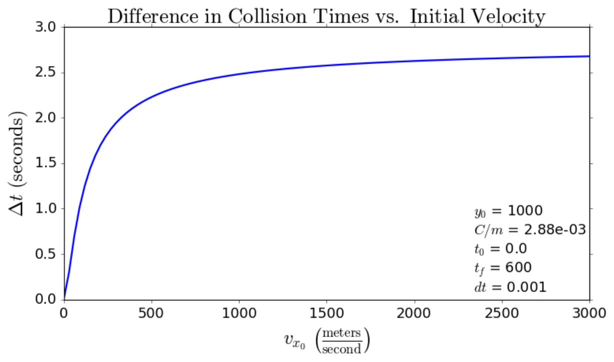

Figure 2: As the initial velocity of the launched projectile increases, the difference in time until collision between the shot and falling projectiles increases, but at a decreasing rate. Remarkably, the rate of increase in Δt is approximately linear for $v_{x0} \in [0, 100]\ \frac{\mathrm{m}}{\mathrm{s}}$.

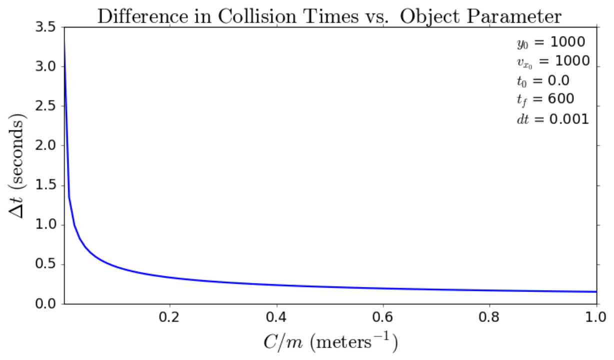

Figure 3: A smaller C/m value is indicative of a smaller cross-sectional area-to-mass ratio for the projectile, while a larger C/m value indicates a larger cross-sectional area-to-mass ratio. Note that as the cross-sectional area-to-mass ratio for the projectile increases to values well above 0.1, the difference in time until collision between the shot and falling projectiles decreases, but this decrease occurs at a rate smaller than the relative increase in C/m. To achieve the largest difference between collision times, the launched object should have a small cross-sectional area, and a large mass.

4 Error Analysis

To what degree may we trust the RK-2 method for approximating the solution to this set of differential equations? Does the error in Δt depend on dt?

Other than checking results against tables generated by more advanced simulations, no analytic solution exists for this problem, so analyzing numerical precision is a bit tricky. In the code for this simulation, a collision is registered by an if-conditional that checks whether the $i + 1$th y-value is less than or equal to zero. In practice, the $i + 1$th y-value is always negative, so it has some degree of "overshoot" past the true $y = 0\,\mathrm{m}$ ground level. Moreover, the larger in magnitude $\frac{dy}{dt}$ is at the $i+1$th y-value, the greater the precision in the t-value associated with a collision.

In an effort to reduce the numerical error associated with overshoot (especially if the projectile is approaching the ground at a shallow angle), I linearly interpolate between the $i+1$th and ith y-value to estimate the time of collision. Using point-slope form, this means we may write the time at collision as

$$t_{\text{hit}} = t_{i+1} - y_{i+1}\left(\frac{t_{i+1} - t_i}{y_{i+1} - y_i}\right).$$

In addition to linearly interpolating t_{hit} to reduce error, I also use a suitably small time step. Figure 4 shows that as the chosen time step tends to zero (zero representing a continuum of time values), the launched projectile's associated absolute overshoot error also tends to zero. As long as the chosen time step is on the order of 10^{-3} or less, the shot object's absolute overshoot error should be on the order of centimeters when the initial y-position is on the order of kilometers, thus strengthening the precision of the simulation.

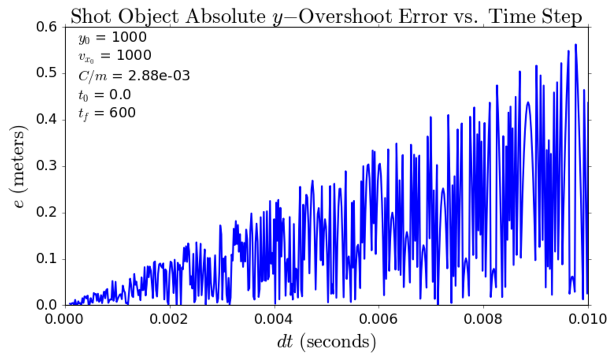

Figure 4: Note that as the time step of the iteration increases, there is a larger spread in the possible overshoots associated with collision times. Since overshoot is the main source of error in this simulation, a suitably small time step (approximately ≤ 0.001) ensures collision heights are within $5\,\mathrm{cm}$ of true ground level.