Projectile Motion with Quadratic Air Drag

Realistic Ballistics Simulation

Joseph Barton

February 2026

Abstract

This document derives the equations of motion for a projectile under gravity and quadratic air resistance (drag force $\propto v^2$), a realistic model for high-speed objects like baseballs or cannon projectiles. The coupled nonlinear ODEs are solved numerically using `scipy.integrate.odeint`. The simulation compares trajectories, range, and maximum height with/without drag, demonstrating reduced range, asymmetric paths, and terminal velocity effects. Source code available in the repository at `projectile_drag.py`.

1 Introduction

Ideal projectile motion (constant gravity, no drag) yields parabolic trajectories and simple analytic formulas for range and height. Real projectiles experience air resistance, often modeled as quadratic drag for Reynolds numbers $\gtrsim 10^3$ (typical for sports balls, bullets, etc.). This force opposes motion and scales with velocity squared, leading to nonlinear ODEs without closed-form solutions.

This work accompanies a Python implementation that numerically integrates the system, visualizes trajectories for various launch angles/velocities, and quantifies drag effects (e.g., optimal angle shifts, energy dissipation).

2 Theoretical Background

The projectile (mass m) experiences two forces:

- Gravity: $\vec{F}_g = -mg\hat{y}$
- Quadratic drag: $\vec{F}_d = -c\,v^2\,\hat{v} = -c\,|\vec{v}|\,\vec{v}$,

where $c > 0$ is the drag coefficient (incorporating air density, cross-sectional area, and drag constant $C_D/2$), $v = |\vec{v}|$ is speed, and $\hat{v} = \vec{v}/v$.

Newton's second law in 2D Cartesian coordinates gives the accelerations:

$$\ddot{x} = -\frac{c}{m}\,v\,\dot{x}, \tag{1}$$

$$\ddot{y} = -g - \frac{c}{m}\,v\,\dot{y}, \tag{2}$$

with $v = \sqrt{\dot{x}^2 + \dot{y}^2}$.

To solve numerically, rewrite as four first-order ODEs by defining state vector $[x, y, v_x, v_y]$:

$$\begin{aligned}
\dot{x} &= v_x, \\
\dot{y} &= v_y, \\
\dot{v}_x &= -\frac{c}{m}\sqrt{v_x^2 + v_y^2}\, v_x, \\
\dot{v}_y &= -g - \frac{c}{m}\sqrt{v_x^2 + v_y^2}\, v_y. \qquad (3)
\end{aligned}$$

Initial conditions: launch from $(x_0, y_0) = (0, h)$ (often $h = 0$ for ground level), with $v_{x0} = v_0 \cos\theta_0$, $v_{y0} = v_0 \sin\theta_0$.

Often define drag parameter $\beta = c/m$ or use terminal velocity $v_t = \sqrt{mg/c}$ (downward fall balance: $mg = cv_t^2$), so $\beta = g/v_t^2$.

3 Numerical Solution

The system integrates with `scipy.integrate.odeint` (or `solve`$_{i}vpwithadaptivestepsforefficiency).Typicalpara$
= 9.81 m/s^2, $v_0 = 20$--$50\,\mathrm{m/s}$, $\theta_0 = 0°$--$90°$, drag coefficient tuned for realistic cases (e.g., baseball: $c \approx 0.0015$--$0.0035\,\mathrm{kg/m}$ depending on speed/spin, but simplified here).

Key results include: - Trajectories curve downward more sharply than parabolas, - Maximum range occurs at launch angle $< 45°$ (often $\sim 30°$--$40°$ depending on drag strength), - Ascent phase less affected than descent (asymmetric path), - Reduced maximum height and range compared to vacuum case, - Plots of range vs. angle, velocity components vs. time, and energy dissipation.

Visualizations use Matplotlib for trajectory overlays (with/without drag) and animations.

4 Conclusion

Projectile motion with quadratic drag illustrates the transition from idealized to realistic mechanics, the importance of nonlinear terms in ODEs, and numerical methods for non-analytic problems. This project demonstrates parameter sweeps, result comparison, and effective plotting for physical insight.

Repository: https://github.com/jbarton727/Portfolio