Week 5 & Week 6: Projectile Motion

Accelerate

Alexander Van Doren [1]

1 Introduction

Projectile motion is a fundamental topic of classical mechanics. It describes the trajectory of an object moving under the influence of gravity alone. From a thrown ball to artillery shells to planetary orbits (in the limit of weak gravitational fields), projectile motion provides essential insights into two-dimensional kinematics. While the easy case (without air resistance) has simple answers, realistic projectile motion requires numerical methods to account for drag forces, making it an perfect introduction to computational physics.

2 Mathematical Formulation

Consider a projectile of mass m launched from position (x_0, y_0) with initial velocity components (v_{x0}, v_{y0}). Not considering air resistance, only gravity acts on the projectile.

2.1 Equations of Motion

The acceleration components are constant:

$$\ddot{x} = 0 \tag{1}$$

$$\ddot{y} = -g \tag{2}$$

where g is gravitational acceleration and y is measured vertically upward.

Affiliation [1] Hack Club
Correspondence alexvd@hackclub.com
Version November 26, 2025

2.2 Analytical Solution (No Air Resistance)

For the easy case, the equations of motion integrate directly to yield:

$$x(t) = x_0 + v_{x0}t \tag{3}$$

$$y(t) = y_0 + v_{y0}t - \frac{1}{2}gt^2 \tag{4}$$

$$v_x(t) = v_{x0} \tag{5}$$

$$v_y(t) = v_{y0} - gt \tag{6}$$

Eliminating time gives the trajectory equation:

$$y = y_0 + x\tan\theta_0 - \frac{g}{2v_0^2\cos^2\theta_0}x^2 \tag{7}$$

where $v_0 = \sqrt{v_{x0}^2 + v_{y0}^2}$ is the initial speed and $\theta_0 = \arctan(v_{y0}/v_{x0})$ is the launch angle.

2.3 Key Parameters

For a projectile launched from height y_0 with initial speed v_0 at angle θ_0:

Range: The horizontal distance traveled when the projectile returns to height y_0 (for $y_0 = 0$):

$$R = \frac{v_0^2\sin(2\theta_0)}{g} \tag{8}$$

Maximum range occurs at $\theta_0 = 45$.

Maximum Height: The apex of the trajectory:

$$h_{\max} = y_0 + \frac{v_{y0}^2}{2g} \tag{9}$$

Time of Flight: For launch and landing at the same height ($y_0 = 0$):

$$T = \frac{2v_0\sin\theta_0}{g} \tag{10}$$

2.4 Conservation of Energy

In the absence of retarding forces, mechanical energy is conserved:

$$E = \frac{1}{2}m(v_x^2 + v_y^2) + mgy = \text{constant} \tag{11}$$

This consists of kinetic energy (from horizontal and vertical motion) and gravitational potential energy.

2.5 Air Resistance

Real projectiles experience drag forces that oppose their motion. For moderate speeds, drag is approximately proportional to velocity squared:

$$\vec{F}_{\text{drag}} = -\frac{1}{2}C_D\rho A|\vec{v}|\vec{v} \tag{12}$$

where C_D is the drag coefficient, ρ is air density, A is the cross-sectional area, and $\vec{v}$ is the velocity vector. This can be simplified using a drag constant $k = \frac{1}{2}C_D\rho A/m$:

$$\ddot{x} = -kv|\vec{v}|_x = -kv_x\sqrt{v_x^2 + v_y^2} \tag{13}$$

$$\ddot{y} = -g - kv|\vec{v}|_y = -g - kv_y\sqrt{v_x^2 + v_y^2} \tag{14}$$

These nonlinear equations describing projectile motion with drag, have no closed-form solution and require numerical integration.

3 Computational Modeling

Numerical methods let us simulate realistic projectile motion including air resistance, wind, and other effects.

3.1 State Space Representation

We reformulate the second-order equations as a first-order system with state vector $\vec{s} = (x, y, v_x, v_y)^T$:

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ -kv_x\sqrt{v_x^2 + v_y^2} \\ -g - kv_y\sqrt{v_x^2 + v_y^2} \end{pmatrix} \tag{15}$$

For the simple case without drag, $k = 0$.

3.2 Numerical Integration Methods

Several numerical methods can integrate this system:

Euler Method: The simplest approach updates position and velocity using:

$$x^{n+1} = x^n + \Delta t\, v_x^n \tag{16}$$

$$y^{n+1} = y^n + \Delta t\, v_y^n \tag{17}$$

$$v_x^{n+1} = v_x^n - \Delta t\, k v_x^n \sqrt{(v_x^n)^2 + (v_y^n)^2} \tag{18}$$

$$v_y^{n+1} = v_y^n - \Delta t\, (g + k v_y^n \sqrt{(v_x^n)^2 + (v_y^n)^2}) \tag{19}$$

While simple to implement, this method can accumulate significant errors as $n \to \infty$ and violates energy conservation for undamped motion.

Runge-Kutta Methods: The fourth-order Runge-Kutta method (RK4) provides excellent accuracy with moderate computational cost:

$$k_1 = f(t_n, \vec{s}_n) \tag{20}$$

$$k_2 = f(t_n + \frac{\Delta t}{2}, \vec{s}_n + \frac{\Delta t}{2} k_1) \tag{21}$$

$$k_3 = f(t_n + \frac{\Delta t}{2}, \vec{s}_n + \frac{\Delta t}{2} k_2) \tag{22}$$

$$k_4 = f(t_n + \Delta t, \vec{s}_n + \Delta t k_3) \tag{23}$$

$$\vec{s}_{n+1} = \vec{s}_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{24}$$

where f represents the right-hand side of equation (15).

3.3 Implementation Considerations

- **Time Step:** Choose Δt based on the characteristic time scale. I find, $\Delta t = 0.01$ s works well.
- **Boundary Conditions:** Stop integrating when $y < 0$ or when the projectile leaves the region of interest.
- **Energy Conservation:** For the no-drag case, monitor total energy to verify numerical accuracy.

- **Trajectory Plotting:** Parametric plots of y vs. x clearly show the parabolic (no drag) or shortened (with drag) trajectory.
- **Interpolation:** Use interpolation to find the exact landing position rather than stopping at the first step where $y < 0$.

3.4 Basic Computational Algorithm

A typical simulation follows this structure:

1. Initialize position (x_0, y_0) and velocity (v_{x0}, v_{y0})
2. Set physical parameters: g, k (drag coefficient), m (mass)
3. For each time step $n = 0, 1, 2, \ldots$:
   (a) Compute speed: $v = \sqrt{(v_x^n)^2 + (v_y^n)^2}$
   (b) Compute accelerations: $a_x = -kv_x^n v$, $a_y = -g - kv_y^n v$
   (c) Update velocity and position using chosen integrator
   (d) Check termination condition: $y < 0$ or out of bounds
   (e) Record state $(t_n, x^n, y^n, v_x^n, v_y^n)$ for visualization
4. Interpolate to find exact landing position and time
5. Output trajectory data and calculate range, flight time, maximum height
6. Create visualizations: trajectory plot, velocity components vs. time, height vs. time

3.5 Validation

Ensure mechanical energy is conserved to within numerical precision

4 Extensions and Variations

4.1 Linear Drag

For low speeds, drag is approximately proportional to velocity: $\vec{F}_{\text{drag}} = -b\vec{v}$. This leads to exponential approach to terminal velocity $v_{\text{term}} = mg/b$ in the vertical direction.

4.2 Wind Effects

A constant horizontal wind velocity v_{wind} modifies the drag force calculation by using relative velocity: $\vec{v}_{\text{rel}} = \vec{v} - \vec{v}_{\text{wind}}$.

You could extend this with variable wind velocity.

4.3 Spin and Magnus Force

Spinning projectiles (e.g., baseballs, soccer balls) experience the Magnus force perpendicular to both velocity and spin axis, causing curved trajectories.

You can take into account this force for different Moments of Inertia, does this have any affect?

4.4 Variable Gravity

For very high altitude trajectories, gravity decreases with altitude: $g(h) = g_0(R_E/(R_E + h))^2$, where R_E is Earth's radius.

4.5 Optimal Launch Angle

With air resistance, the optimal launch angle for maximum range is typically less than 45 and depends on initial speed and drag coefficient—a problem requiring numerical optimization.

5 Conclusion

Projectile motion serves as an ideal bridge between analytical and computational physics. The elegant analytical solutions for the easy case provide an intiutitve starting point, while the realistic case with air resistance demonstrates the necessity and power of numerical methods. From sports to aerospace engineering, understanding projectile dynamics through both analytical and computational approaches provides essential skills for tackling more complex problems in classical mechanics and beyond. The techniques developed here—state space formulation, numerical integration, and systematic validation—extend naturally to more sophisticated systems encountered in advanced physics courses.

References

[1] © Alex Van Doren (2025).