

Under the Hood: Aerodynamic Re-entry

Simulation Engine V2.0 Physics Architecture

Engineering Club

1 Introduction: The Intuition of Free-fall

When a spacecraft like the SpaceX Dragon capsule re-enters the Earth's atmosphere, it is locked in a battle between two primary forces. Gravity pulls the mass downward, attempting to accelerate the capsule indefinitely. However, as it falls, it collides with air molecules. This resistance pushes back—a force known as **aerodynamic drag**.

For high school students, we often learn Newton's Second Law as $F = ma$. But to build a real-time computer simulation, we must look deeper into the *causality* of the system: how forces at a specific instant dictate the velocity in the next instant.

2 Formulating the Equation

The net force acting on our descending capsule at any given moment t is the difference between the force of gravity (F_g) and the force of drag (F_d):

$$F_{\text{net}}(t) = F_g - F_d(t) \quad (1)$$

We know that gravity is constant near the surface ($F_g = mg$). But drag is dynamic; it scales with the *square* of the velocity. If you double your speed, the air resistance quadruples. The drag equation is given by:

$$F_d(t) = \frac{1}{2}\rho v(t)^2 C_d A \quad (2)$$

Where:

- ρ is the atmospheric density (approx. 1.225 kg/m^3 on Earth).
- $v(t)$ is the instantaneous velocity.
- C_d is the drag coefficient (shape dependence, ≈ 1.5 for capsules).
- A is the cross-sectional area (which drastically increases when the parachute deploys).

Substituting these into Newton's Second Law ($F_{\text{net}} = m \frac{dv}{dt}$), we obtain the governing **first-order, non-linear Ordinary Differential Equation (ODE)** for our system:

$$m \frac{dv}{dt} = mg - \frac{1}{2}\rho v^2 C_d A \quad (3)$$

Because area (A) changes dynamically as the user interacts with the simulation slider, an exact analytical solution (like you might solve on paper) is impossible. The system must be solved numerically.

3 Numerical Integration: The Euler Method

To simulate this reality in the browser, the Engine steps forward in tiny fractions of a second (our timestep, $\Delta t \approx 0.016\text{s}$ or 60 Frames Per Second).

We rearrange our differential equation to solve for instantaneous acceleration, a_n :

$$a_n = \frac{dv}{dt} = g - \frac{\rho C_d A}{2m} v_n^2 \quad (4)$$

Using the **Euler Method** of numerical integration, we can predict the state of the capsule in the very next frame based entirely on the conditions of the current frame:

$$v_{n+1} = v_n + a_n \Delta t \quad (5)$$

$$y_{n+1} = y_n - v_n \Delta t \quad (6)$$

By repeating this calculation 60 times a second, static algebraic constraints are transformed into a living, breathing physics simulation. The computer literally calculates the future by continually accumulating the present.