Scratch Rocket Simulator Manual with Gravity Turn and Altitude-Dependent Air Density

Kerbal Space Math Adventure

1 Introduction

This manual explains a Scratch-based rocket simulator that uses Euler's method to approximate the rocket's trajectory. The simulator accounts for:

- Engine thrust and mass loss (fuel burn)
- Gravitational acceleration
- Air resistance (drag) with an altitude-dependent air density model
- A gravity turn maneuver for an optimized launch trajectory

2 Key Variables and Their Descriptions

Below is a list of all variables used in the simulation, along with their meanings and units:

- **T (thrust):** Engine thrust, measured in newtons (N). In the simulator, this is represented by the variable `thrust`.
- **m (mass):** Current mass of the rocket (rocket + remaining fuel) in kilograms (kg). Represented by `mass`. This decreases as fuel is burned.
- **fuel:** Remaining fuel mass in kilograms (kg). Represented by `fuel`.
- $\dot{m}$ **(burnRate):** Fuel consumption rate in kg/s. Represented by `burnRate`.
- **g (gravity):** Gravitational acceleration, set to 9.81 m/s^2. Represented by `gravity`.
- θ **(angle):** The current pitch (launch) angle in degrees. Initially 90° for a vertical launch, then adjusted during the gravity turn. Represented by `angle`.
- **rad:** The angle in radians (converted from `angle` using

  $$\text{rad} = \theta \times \frac{\pi}{180}$$

).

- v_x, v_y **(vx, vy):** Horizontal and vertical components of velocity in m/s.
- x, y **(x, y):** Horizontal and vertical positions in meters (m).
- Δt **(dt):** Time step for Euler integration in seconds (s). Represented by `dt`.
- ρ **(air density):** Air density in kg/m^3. At sea level, $\rho_0 = 1.225$ kg/m^3. In our model, it decreases with altitude according to

  $$\rho(y) = \rho_0 \, e^{-y/h},$$

  where h is the scale height (taken as 5000 m). In the simulator, the current air density is stored in `currentAirDensity`.
- C_d **(dragCoefficient):** Drag coefficient (dimensionless). Represented by `dragCoefficient`.
- **A (area):** Frontal (cross-sectional) area of the rocket in m^2. Represented by `area`.
- $F_{\textbf{drag}}$**:** Drag force in newtons (N), calculated as:

  $$F_{\text{drag}} = \frac{1}{2} \rho(y) \, v^2 \, C_d \, A.$$

- $a_{\textbf{drag}}$**:** Drag acceleration in m/s^2, found by dividing the drag force by the mass:

  $$a_{\text{drag}} = \frac{F_{\text{drag}}}{m}.$$

  Its components (acting opposite to the direction of motion) are:

  $$a_{dx} = -a_{\text{drag}} \cdot \frac{v_x}{v}, \quad a_{dy} = -a_{\text{drag}} \cdot \frac{v_y}{v},$$

  where

  $$v = \sqrt{v_x^2 + v_y^2}.$$

- **Gravity Turn Parameters:**
  - `turnStart`: Altitude (in m) at which the gravity turn begins.
  - `turnEnd`: Altitude (in m) at which the gravity turn is completed.
  - `finalAngle`: The final pitch angle (in degrees) after the gravity turn, denoted as θ_{final}.

3 The Core Formulas

Thrust Acceleration

The acceleration produced by the engine's thrust is given by:

$$a_{x,\text{thrust}} = \frac{T \cos(\text{rad})}{m}, \quad a_{y,\text{thrust}} = \frac{T \sin(\text{rad})}{m} - g.$$

Here, T is thrust, θ is the pitch angle, and rad is the angle in radians.

Drag Force and Acceleration

The drag force is:

$$F_{\text{drag}} = \frac{1}{2}\,\rho(y)\,v^2\,C_d\,A,$$

and the corresponding acceleration is:

$$a_{\text{drag}} = \frac{F_{\text{drag}}}{m}.$$

Its components, acting opposite to the direction of velocity, are:

$$a_{dx} = -a_{\text{drag}}\frac{v_x}{v}, \quad a_{dy} = -a_{\text{drag}}\frac{v_y}{v}.$$

Total Acceleration

The net acceleration is the sum of thrust acceleration and drag acceleration:

$$a_x = a_{x,\text{thrust}} + a_{dx}, \quad a_y = a_{y,\text{thrust}} + a_{dy}.$$

Euler Integration Update Equations

The state of the rocket is updated every time step Δt as follows:

$$v_x(t+\Delta t) = v_x(t) + a_x\,\Delta t, \quad v_y(t+\Delta t) = v_y(t) + a_y\,\Delta t,$$

$$x(t+\Delta t) = x(t) + v_x(t)\,\Delta t, \quad y(t+\Delta t) = y(t) + v_y(t)\,\Delta t,$$
$$m(t+\Delta t) = m(t) - (\text{burnRate}\cdot\Delta t).$$

Gravity Turn (Dynamic Pitch Update)

When the rocket reaches an altitude above `turnStart`, the pitch angle is updated by linear interpolation:

$$\theta = 90^\circ - (90^\circ - \theta_{\text{final}})\cdot\frac{y - y_{\text{start}}}{y_{\text{end}} - y_{\text{start}}},$$

and clamped so that $\theta \geq \theta_{\text{final}}$.

4 Scratch Implementation Pseudocode

Below is the pseudocode for the complete Scratch simulator, now including altitude-dependent air density, drag, and a gravity turn:

```
when green flag clicked
    // Initialize variables:
    set [mass v] to 1000        // in kg (rocket + fuel)
    set [fuel v] to 400         // in kg
```

```
set [thrust v] to 15000     // in N (T)
set [burnRate v] to 10      // in kg/s (fuel consumption rate, m dot)
set [gravity v] to 9.81     // in m/s^2 (g)
set [angle v] to 90         // initial pitch angle in degrees (theta)
set [vx v] to 0             // horizontal velocity in m/s
set [vy v] to 0             // vertical velocity in m/s
set [x v] to 0              // horizontal position in m
set [y v] to 0              // vertical position in m
set [dt v] to 0.2           // time step in seconds (deltat)

// Drag and atmosphere variables:
set [dragCoefficient v] to 0.75  // (C_d), dimensionless
set [area v] to 1.0              // frontal area in m^2 (A)
// Base air density at sea level: 1.225 kg/m^3

// Gravity turn parameters:
set [turnStart v] to 500    // altitude in m at which to
  // start the turn (y_start)
set [turnEnd v] to 5000     // altitude in m at which the turn
    // is complete (y_end)
set [finalAngle v] to 45    // final pitch angle in degrees (theta_final)

repeat until <(fuel) <= 0 or (y) < 0>
    // Gravity turn: update angle based on altitude:
    if <(y) > (turnStart)> then
        set [angle v] to (90 - ((90 - finalAngle) * ((y - turnStart)
         / (turnEnd - turnStart))))
        if <(angle) < (finalAngle)> then
            set [angle v] to (finalAngle)
        end
    end
    // Convert angle to radians:
    set [rad v] to ((angle) * 3.14159 / 180)

    // Update current air density: rho(y) = 1.225 * e^( - y / 5000 )
    set [currentAirDensity v] to (1.225 * (e^( - (y / 5000) )))

    // Calculate thrust acceleration components:
    set [ax_thrust v] to ((thrust * cos(rad)) / mass)
    set [ay_thrust v] to (((thrust * sin(rad)) / mass) - gravity)

    // Compute current speed: v = sqrt((vx^2 + vy^2))
    set [v_total v] to (sqrt((vx)^2 + (vy)^2))

    // Calculate drag if v_total > 0:
```

```
    if <(v_total) > (0)> then
        set [F_drag v] to (0.5 * currentAirDensity * ((v_total)^2)
         * dragCoefficient * area)
        set [a_drag v] to (F_drag / mass)
        set [ax_drag v] to ((-1) * a_drag * (vx / v_total))
        set [ay_drag v] to ((-1) * a_drag * (vy / v_total))
    else
        set [ax_drag v] to 0
        set [ay_drag v] to 0
    end

    // Total acceleration:
    set [ax v] to (ax_thrust + ax_drag)
    set [ay v] to (ay_thrust + ay_drag)

    // Update velocities using Euler's method:
    set [vx v] to (vx + ax * dt)
    set [vy v] to (vy + ay * dt)

    // Update positions:
    set [x v] to (x + vx * dt)
    set [y v] to (y + vy * dt)

    // Update fuel and mass:
    change [fuel v] by ((-burnRate) * dt)
    change [mass v] by ((-burnRate) * dt)

    // Move rocket sprite (optional):
    go to x: (x) y: (y)
    wait (dt) seconds
end
```

5 Conclusion

This manual details a comprehensive Scratch-based rocket simulator that:

- Uses Euler's method for numerical integration.
- Accounts for thrust, fuel burn, gravity, drag with an altitude-dependent atmosphere, and a dynamic gravity turn.

Every variable and formula is clearly labeled and described, allowing you to experiment with and understand the physics behind rocket flight. Enjoy exploring and optimizing your simulated launch trajectories!