Motion in a Uniform Gravitational Field

1. Objective

The goal of this module is to create a comprehensive, self-contained educational resource on kinematics in a uniform gravitational field. The resulting documentation should serve as a complete study guide, deriving physical principles from the ground up rather than presenting ready-made formulas.

2. Theoretical Scope

Your note must cover the motion of a particle in a two-dimensional space (x, y) under the influence of gravity.

2.1. Fundamental Derivations

You are required to derive the equations of motion from first principles (Newton's Laws). **Do not simply copy-paste final formulas for range or height.**

- **Starting Point**: Begin with the force definitions in a vacuum (gravity only).
- **Equations of Motion**: Write down the differential equations for acceleration in x and y directions.
- **Integration**: Show the mathematical process of integrating acceleration to obtain velocity functions $v_x(t), v_y(t)$ and position functions $x(t), y(t)$.
- **Initial Conditions**: Clearly define how initial velocity v_0, angle α, and initial height h_0 affect the integration constants.

2.2. Analytical Results

Once the equations of motion are established, derive analytical expressions for the following metrics:

- Maximum height (H).
- Time of flight (T).
- Horizontal range (R).
- Velocity vector at any given time t.

2.3. Scenarios to Analyze

You must apply your derivations to the following specific cases:

1. **Classic Projectile Motion**: Launch from the ground $(h_0 = 0)$ at an angle.
2. **Free Fall / Aerial Drop**: A scenario where an object is dropped from a moving vehicle (e.g., a package dropped from a plane flying horizontally). Analyze how the horizontal velocity of the plane affects the trajectory.

2.4. Extension: Air Resistance (Drag)

Analyze how the problem changes when air resistance is introduced.

- Discuss the modification of the equations of motion when a drag force $\vec{F}_d = -k\vec{v}$ or $\vec{F}_d = -cv^2\hat{v}$ is applied.
- Explain why analytical solutions might become complex or impossible and how this necessitates numerical methods.

3. Interactive Simulation Requirements

You must develop an interactive HTML/JavaScript application (to be placed in the `assets/` folder) that visualizes these concepts. The tool must go beyond simple animation and serve as a "physics inspection" instrument.

Functional Requirements:

- **Trajectory Visualization**: Plot the path of the projectile in 2D space (y vs x).
- **Vector Visualization (Crucial)**: The user must be able to toggle the visibility of specific vectors attached to the object in real-time. This includes:
  - Instantaneous Velocity vector $(\vec{v})$.
  - Acceleration vector ($\vec{g}$ or $\vec{a}_{total}$).

    - Drag Force vector ($\vec{F}_d$) (if air resistance is active).
    - Net Force vector ($\vec{F}_{net}$).
    - *Implementation Note*: Add checkboxes/toggles in the UI to show/hide these vectors independently. This allows observers to study how forces and velocities change direction and magnitude relative to each other during flight.
- **Interactivity**: Allow the user to adjust inputs such as:
    - Initial velocity (v_0).
    - Launch angle (α).
    - Initial height (h_0).
    - Gravitational acceleration (g).
- **Scenarios**: The app should support both the "Cannon" mode (ground launch) and the "Plane Drop" mode.
- **Real-time Feedback**: Display calculated values (Range, Time of Flight) dynamically as parameters change.

Technical Note: While analytical solutions work for a vacuum, consider implementing a simple numerical integrator (e.g., Euler or Verlet method) in your code to handle the optional air resistance scenario or simply to demonstrate numerical solving of ODEs.

4. Deliverables and Quality Standards

Your submission in `problem_solution.md` is expected to be a high-quality technical document, not a brief homework assignment.

- **Completeness**: The note must be understandable to someone who has not seen the lecture. Explain the "Why" and "How", not just the "What".
- **Mathematical Rigor**: Use LaTeX formatting for all equations. Ensure clear steps in derivations.
- **Visuals**: Include static plots or diagrams (can be generated via Python/Matplotlib or drawn) to explain coordinate systems and vector components.
- **Code Integration**: Your HTML simulation must be embedded or clearly linked, demonstrating that your theoretical derivations match the simulated reality.