```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>

#include "Math.hpp"
#include "RocketState.hpp"
#include "Simulation.hpp"
#include "MonteCarlo.hpp"

int main()
{
    RocketState x0;

    x0.r_I     = Vector3(0.0, 0.0, 0.0);
    x0.v_I     = Vector3(0.0, 0.0, 0.0);

    // Launch angle:
    double launch_angle_deg = 45.0;
    double launch_angle_rad = launch_angle_deg * M_PI / 180.0;

    x0.euler   = Vector3(0.0, launch_angle_rad, 0.0); // roll, pitch, yaw
    x0.omega_B = Vector3(0.0, 0.0, 5.0* M_PI/180.0); // initial spin [rad/s]

    // V2 Rocket parameters
    x0.mass = 12500.0;  // kg (fully fueled)
    x0.I_B  = InertiaDiagonal(500.0, 50000.0, 50000.0);

    double dt = 0.01;
    int    numSteps = 100000; // 100 seconds total

    // V2 thrust and fuel (~7800 kg of propellant)
    Vector3 F_B(0.0, 0.0, 245000.0); // N (thrust along body +z axis)
    double t_burn = 500.0; // seconds
    double mass_flow_rate = 130.0; // kg/s
    double mass_empty = 4000.0; // kg

    // V2 drag parameters
    double Cd = 0.125;
    double A_ref = 2.14;

    Vector3 M_B(0.0, 0.0, 0.0);
    Vector3 g_I(0.0, 0.0, -9.81); // gravity

    std::vector<RocketState> states;

    simulateEuler(x0, dt, numSteps, F_B, M_B, g_I, t_burn, mass_flow_rate,
                  Cd, A_ref, mass_empty, states);

    std::ofstream trajFile("trajectory3d.csv");
    if (!trajFile)
    {
        std::cerr << "Error opening trajectory3d.csv for writing\n";
        return 1;
    }

    // Excel header
    trajFile << "t,"
            << "x,y,z,"
            << "vx,vy,vz,"
            << "roll,pitch,yaw,"
            << "mass\n";

    // Column
    for (int k = 0; k < states.size(); ++k)
    {
        double t = k * dt;
        const RocketState &s = states[k];

        trajFile << t << ","
                << s.r_I.x << ","
                << s.r_I.y << ","
                << s.r_I.z << ","
                << s.v_I.x << ","
                << s.v_I.y << ","
                << s.v_I.z << ","
                << s.euler.x << ","
                << s.euler.y << ","
                << s.euler.z << ","
                << s.mass
                << "\n";
    }


    trajFile.close();

    double launch_angle_nom_deg = launch_angle_deg;
    double mass_nom = x0.mass;
    RocketState x0_nom = x0;

    std::mt19937 rng(12345);

    const int Ntrials = 500;
    std::ofstream mc_out("monte_carlo_summary.csv");
    mc_out << "trial,angle_deg,mass_scale,impact_time,range,max_alt\n";

    for (int i = 0; i < Ntrials; ++i)
    {
        MonteCarloResult r = run_one_trial(
            rng,
            x0_nom,
            launch_angle_nom_deg,
            dt,
            numSteps,
            t_burn,
            mass_flow_rate,
            Cd,
            A_ref,
            mass_empty,
            F_B,
            M_B,
            g_I
        );

        mc_out << i << ","
            << r.launch_angle_deg << ","
            << r.mass_scale << ","
            << r.impact_time << ","
            << r.ground_range << ","
            << r.max_altitude << "\n";
    }

    return 0;
}
```