```
%Aidan Chin
%Project 2
%11/14/23


% initialize

clear
clc

% Constants

g = 32.2; % acceleration of gravity in ft/s^2
theta = deg2rad(28); % launch angle in radians
vE = 116 * 5280 / 3600; % exit velocity in ft/s (converted from mph)
aToF = 5.3; % time of flight in seconds
baseballmass = 0.145; % mass of a baseball in kg
rho_air = .00238; % air density in slugs/ft^3
C = input("Enter drag coefficient (e.g. .38): ");

baseballr = .06035; %radius of baseball in ft
baseballa = pi * baseballr^2; %cross sectional area of baseball

% Initial conditions
x0 = 0; y0 = 0; % initial position
vx0 = vE * cos(theta); % initial x-component of velocity
vy0 = vE * sin(theta); % initial y-component of velocity

% Time settings
dt = 0.01; % time step
tmax = aToF; % maximum time
tval = 0:dt:tmax; % array of time values

% Initialize arrays to store results
%no drag
x = zeros(size(tval));
y = zeros(size(tval));
%drag
xdrag = zeros(size(tval));
ydrag = zeros(size(tval));

% Initial conditions
%no drag
x(1) = x0;
y(1) = y0;
vx = vx0;
vy = vy0;
%drag
xdrag(1) = x0;
ydrag(1) = y0;
vxdrag = vx0;
```

```
vydrag = vy0;

% Numerical computation using Euler's method with and without drag
for i = 2:length(tval)
    % Acceleration components
    % no drag
    ax = 0; % no acceleration in x-direction
    ay = -g; % acceleration due to gravity in y-direction
    % drag
    vdrag = sqrt(vxdrag^2 + vydrag^2);
    axdrag = -.5 * C * rho_air * baseballa * vdrag * vxdrag / baseballmass;
    aydrag = -g - .5 * C * rho_air * baseballa * vdrag * vydrag / baseballmass;

    % Update velocities and positions using Euler's method
    %no drag
    vx = vx + ax * dt;
    vy = vy + ay * dt;
    x(i) = x(i - 1) + vx * dt;
    y(i) = y(i - 1) + vy * dt;
    %drag
    vxdrag = vxdrag + axdrag * dt;
    vydrag = vydrag + aydrag * dt;
    xdrag(i) = xdrag(i - 1) + vxdrag * dt;
    ydrag(i) = ydrag(i - 1) + vydrag * dt;

    % Check for the end of the trajectory
    if y(i) < 0
        break;
    end
end

% convert units to feet
x = x * 3.28084;
y = y * 3.28084;
xdrag = xdrag * 3.28084;
ydrag = ydrag * 3.28084;

%check function

check_x = abs(x(end) - (vx0 * tval(end)));
check_y = abs(y(end) - (y0 + vy0 * tval(end) - 0.5 * g * tval(end)^2));

disp(['max difference in x (with drag): ', num2str(check_x)])
disp(['max difference in y (with drag): ', num2str(check_y)])

% Plot trajectories
figure;
plot(x, y, '--','LineWidth', 1.5, 'DisplayName', 'no drag');
hold on;
plot(xdrag, ydrag, '-', 'LineWidth', 1.5, 'DisplayName', 'drag');
title(['Aidan Chin | ECE202 Project 2 | 12/07/23 |' ...
```

```
103     ' Baseball Trajectory with and  without Air Resistance']);
104 xlabel('Distance (feet)');
105 ylabel('Height (feet)');
106 legend('no drag', ['drag (C = ' num2str(C) ')'], 'Location', 'Best');
107 grid on;
108 ax = gca;
109 ax.GridAlpha = .4;
110 ax.MinorGridAlpha = .5;
111
112 % save to excel
113
114 % create a matrix of data
115 dataMatrix = [tval',xdrag', ydrag'];
116
117 % export the data to a CSV
118 filename = 'baseball_trajectory_data.csv';
119 writematrix(dataMatrix, filename);
120
121 % display filename
122 disp(['Data exported to ' filename]);
123
124
```