

```

1 %Aidan Carey
2 % project 2 phase 2
3 %12/07/23
4 % description- in this phase, we will use a majority of the script from
5 % phase 1, but will be adding air resistance. Again, we will plot the
6 % numeric versus analytic curves and can see the affect of drag on the
7 % trajectory based on the user input.
8
9 % Constants
10 g = 32.2; % acceleration due to gravity in ft/s^2
11 launch_angle = deg2rad(28); % launch angle in radians
12 exit_velocity = 116 * 5280 / 3600; % exit velocity in ft/s
13 % (converted from mph)
14 analytic_time_of_flight = 5.3; % time of flight in seconds
15
16 % Additional constants
17 mass_of_baseball = 0.145; % mass of a baseball in kg
18 rho_air = 0.00238; % air density in slugs/ft^3
19
20 % User input for drag coefficient
21 C = input('Enter the drag coefficient (e.g., 0.38): ');
22
23 % Cross-sectional area of a baseball (approximated as a sphere)
24 radius_of_baseball = 0.06035; % in feet
25 A = pi * radius_of_baseball^2; % cross-sectional area
26
27 % Initial conditions
28 x0 = 0; y0 = 0; % initial position
29 vx0 = exit_velocity * cos(launch_angle); % initial x-component of velocity
30 vy0 = exit_velocity * sin(launch_angle); % initial y-component of velocity
31
32 % Time settings
33 dt = 0.01; % time step
34 t_max = analytic_time_of_flight; % maximum time
35 t_values = 0:dt:t_max;
36
37 % Initialize arrays to store results
38 x_values_no_drag = zeros(size(t_values));
39 y_values_no_drag = zeros(size(t_values));
40 x_values_with_drag = zeros(size(t_values));
41 y_values_with_drag = zeros(size(t_values));
42
43 % Initial conditions
44 x_values_no_drag(1) = x0;
45 y_values_no_drag(1) = y0;
46 x_values_with_drag(1) = x0;
47 y_values_with_drag(1) = y0;
48 vx_no_drag = vx0;
49 vy_no_drag = vy0;
50 vx_with_drag = vx0;
51 vy_with_drag = vy0;
52
53 % Checking functions
54 check_x = @(x_values, t_values) sum(x_values - (vx0 * t_values));
55 check_y = @(y_values, t_values) sum(y_values - ...
56     (y0 + vy0 * t_values - 0.5 * g * t_values.^2));
57
58 % Numerical computation using Euler's method with and without drag
59 for i = 2:length(t_values)

```

```

60 % Without drag
61 ax_no_drag = 0; % no acceleration in x-direction
62 ay_no_drag = -g; % acceleration due to gravity in y-direction
63
64 % With drag
65 v_with_drag = sqrt(vx_with_drag^2 + vy_with_drag^2);
66 ax_with_drag = -0.5 * C * rho_air * A * v_with_drag * vx_with_drag ...
67 / mass_of_baseball;
68 ay_with_drag = -g - 0.5 * C * rho_air * A * v_with_drag ...
69 * vy_with_drag / mass_of_baseball;
70
71 % Update velocities and positions using Euler's method
72 vx_no_drag = vx_no_drag + ax_no_drag * dt;
73 vy_no_drag = vy_no_drag + ay_no_drag * dt;
74 x_values_no_drag(i) = x_values_no_drag(i - 1) + vx_no_drag * dt;
75 y_values_no_drag(i) = y_values_no_drag(i - 1) + vy_no_drag * dt;
76
77 vx_with_drag = vx_with_drag + ax_with_drag * dt;
78 vy_with_drag = vy_with_drag + ay_with_drag * dt;
79 x_values_with_drag(i) = x_values_with_drag(i - 1) + vx_with_drag * dt;
80 y_values_with_drag(i) = y_values_with_drag(i - 1) + vy_with_drag * dt;
81
82 % Check for the end of the trajectory
83 if y_values_no_drag(i) < 0
84     break;
85 end
86 end
87
88 % Convert positions to feet
89 x_values_no_drag = x_values_no_drag * 3.28084;
90 y_values_no_drag = y_values_no_drag * 3.28084;
91 x_values_with_drag = x_values_with_drag * 3.28084;
92 y_values_with_drag = y_values_with_drag * 3.28084;
93
94 % Checking functions
95 check_x = abs(x_values(end) - (vx0 * t_values(end)));
96 check_y = abs(y_values(end) - (y0 + vy0 * t_values(end) ...
97 - 0.5 * g * t_values(end)^2));
98
99 % Print the maximum differences (should be close to zero)
100 disp(['Max difference in x (with drag): ', num2str(check_x)]);
101 disp(['Max difference in y (with drag): ', num2str(check_y)]);
102
103 % Plot trajectories with and without drag
104 figure;
105 plot(x_values_no_drag, y_values_no_drag, '--', 'LineWidth', 1.5, ...
106 'DisplayName', 'No Drag');
107 hold on;
108 plot(x_values_with_drag, y_values_with_drag, '-', 'LineWidth', 1.5, ...
109 'DisplayName', 'With Drag');
110 title(['Aidan Carey | ECE202 Phase 2 | 12/08/23 | ' ...
111 'Baseball Trajectory with and without Air Resistance (Drag)']);
112 xlabel('Distance (feet)');
113 ylabel('Height (feet)');
114 legend('Location', 'Best');
115 grid on;
116 grid minor; % Add minor grid lines
117 ax = gca;
118 ax.GridAlpha = 0.4; % Adjust grid alpha for visibility

```

```
119 ax.MinorGridAlpha = 0.5; % Adjust minor grid alpha for visibility
```

```
>> project2phase2  
Enter the drag coefficient (e.g., 0.38): 0.38  
Max difference in x (with drag): 796.1597  
Max difference in y (with drag): 28.9234  
>>
```

