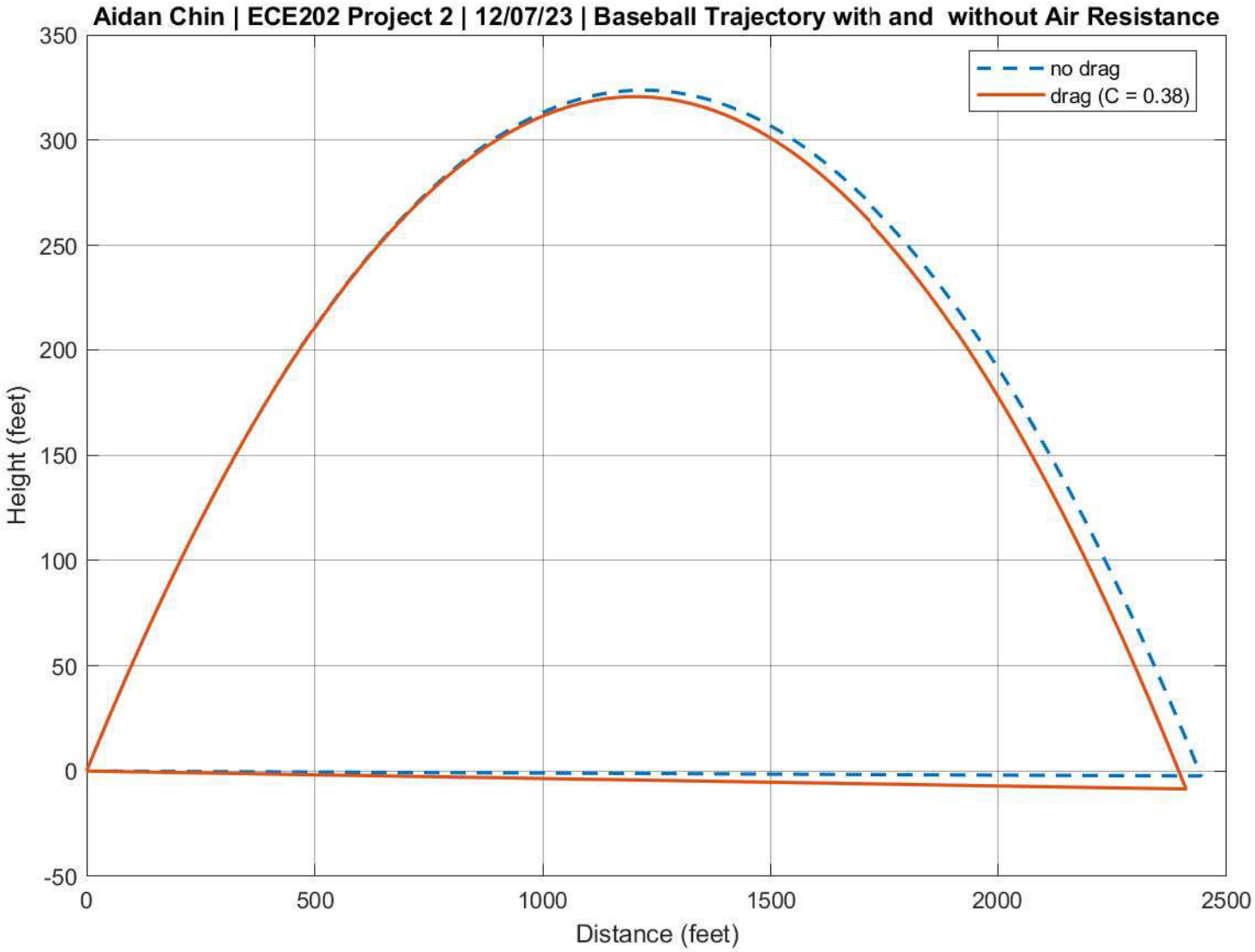
	A	B	C	D	E	F	G
1	Aidan Chin			tval	xdrag	ydrag	ydrag ratio
2	ECE202			0	0	0	0
3	10-Dec			0.01	4.92814	2.609774	0.501029
4				0.02	9.855981	5.208826	0.668042
5	min value of ydrag ratio	index	flight time value	0.03	14.78352	7.797155	0.75155
6	-2.70843297	494	4.92	0.04	19.71077	10.37476	0.801657
7				0.05	24.63771	12.94165	0.835063
8				0.06	29.56436	15.49782	0.858925
9				0.07	34.49071	18.04327	0.876823
10				0.08	39.41677	20.57801	0.890745
11				0.09	44.34252	23.10202	0.901883
12				0.1	49.26798	25.61532	0.910997
13				0.11	54.19315	28.1179	0.918592
14				0.12	59.11802	30.60977	0.92502
15				0.13	64.04259	33.09093	0.93053
16				0.14	68.96687	35.56137	0.935306
17				0.15	73.89085	38.0211	0.939486
18				0.16	78.81454	40.47012	0.943174
19				0.17	83.73793	42.90843	0.946453
20				0.18	88.66103	45.33602	0.949388

	A	B	C	D	E	F	G
1	Aidan Chin			tval	xdrag	ydrag	ydrag ratio
2	ECE202			0	0	0	=IF(F3<>0,F2/F3,"")
3	45270			0.01	4.92813983709927	2.60977412713235	=IF(F4<>0,F3/F4,"")
4				0.02	9.85598077305959	5.20882566165534	=IF(F5<>0,F4/F5,"")
5	min value of ydrag ratio	index	flight time value	0.03	14.7835231087026	7.79715540301312	=IF(F6<>0,F5/F6,"")
6	=MIN(G:G)	=MATCH(A6,G:G,)	=INDIRECT("D"&B6)	0.04	19.710767143923	10.374764148912	=IF(F7<>0,F6/F7,"")
7				0.05	24.6377131776872	12.9416526953252	=IF(F8<>0,F7/F8,"")
8				0.06	29.5643615080309	15.4978218364979	=IF(F9<>0,F8/F9,"")
9				0.07	34.4907124320573	18.0432723649518	=IF(F10<>0,F9/F10,"")
10				0.08	39.4167662459357	20.5780050714902	=IF(F11<>0,F10/F11,"")
11				0.09	44.3425232448988	23.102020745203	=IF(F12<>0,F11/F12,"")
12				0.1	49.2679837232418	25.6153201734713	=IF(F13<>0,F12/F13,"")
13				0.11	54.1931479743194	28.1179041419725	=IF(F14<>0,F13/F14,"")
14				0.12	59.1180162905451	30.6097734346853	=IF(F15<>0,F14/F15,"")
15				0.13	64.0425889633884	33.0909288338947	=IF(F16<>0,F15/F16,"")
16				0.14	68.9668662833732	35.5613711201966	=IF(F17<>0,F16/F17,"")
17				0.15	73.8908485400762	38.0211010725034	=IF(F18<>0,F17/F18,"")
18				0.16	78.8145360221248	40.4701194680484	=IF(F19<>0,F18/F19,"")
19				0.17	83.7379290171948	42.9084270823913	=IF(F20<>0,F19/F20,"")
20				0.18	88.6610278120094	45.3360246894229	=IF(F21<>0,F20/F21,"")

```
%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
```


Aidan Chin | ECE202 Project 2 | 12/07/23 | Baseball Trajectory with and without Air Resistance
no drag
drag (C = 0.38)
Height (feet)
Distance (feet)
350
300
250
200
150
100
50
0
-50
0
500
1000
1500
2000
2500