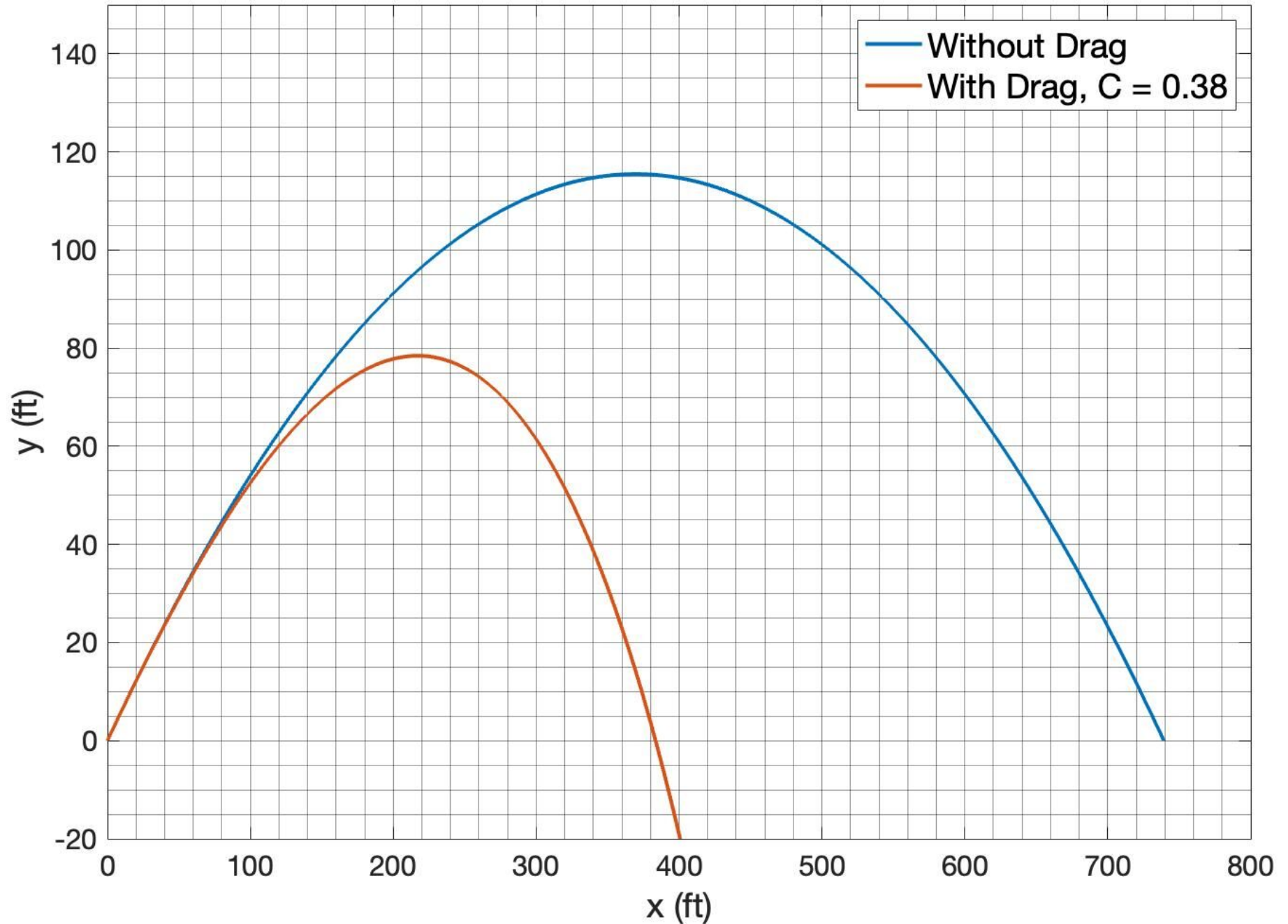


```
1 % Furquaan Syed
2 % 12/8/2021
3 % Professor Leonard
4 % ECE 202 Fall 2021 Project 2
5 % Using numerical techniques to calculate range, height and flight time
6 % of a baseball
7
8 % close collaborator with Ryan Palmer
9
10      %---Phase 2---%
11
12 % Adding air resistance (drag) and checking when drag is "turned off"
13
14
15 clear
16 clf
17 format shortg
18
19 %--- Parameters ---%
20
21 theta = 32; % launch angle in degrees
22 v0 = 112; % Exit velocity in miles per hour (mph)
23
24 x0 = 0; y0 = 0; % initial position of the ball
25
26 g = 10; % gravitational constant in N/kg (1 N/kg = 1 m/s^2)
27
28 m = 0.145; % mass of baseball in kilograms
29
30 p = 1.225; %density of air in kg/m^3
31 A = pi() * 0.0365^2; %cross sectional area of a baseball m^2
32
33 C = input('Enter C value for drag coefficient: ');
34
35 mph2mps = 5280 * 12 * 2.54 / 100 / 3600; % mph to m/s conversion
36 deg2rad = pi()/180; % degrees to radians
37 m2feet = 3.281; %meters to feet
38
39 % add a conversion from m to ft, e.g. m2ft, and use it several times
40
41
42 v0mps = v0 * mph2mps; % Exit velocity in meters per second
43 thetaRad = theta * deg2rad; % launch angle in radians
44
45 v0x = v0mps * cos(thetaRad); % x-component of v0
46 v0y = v0mps * sin(thetaRad); % y-component of v0
47
48
```

```
49 % ----- compute some useful characteristics of trajectory -----
50
51 tH = v0y/g;    % time to reach max. height
52 tLand = 2*tH;  % time to land (time of flight)
53
54 H = tH * v0y/2; % max. height
55 R_ft = v0x * tLand; % range
56
57 R_m = R_ft / m2feet; % ROUGH conversion from m to ft
58
59
60 %--- analytical x(t) and y(t)---%
61
62 tmin = 0; tmax = tLand;
63 N = 2000; % intervals
64
65 t = linspace(tmin, tmax, N+1); % time array, connects x(t) with y(t)
66
67 %--- calculations for no drag---%
68
69 xt = x0 + v0x*t; % x(t), ax = 0 (no drag)
70 yt = y0 + v0y*t - (1/2)*g*t.^2; % y(t), ay = -g (no drag)
71
72
73 % ----- calculations for drag -----%
74
75
76 dt = (tmax-tmin)/N; %change in time intervals
77
78 y = zeros(1, N+1); % initialize y(t) and x(t)
79 x = zeros(1, N+1);
80
81 y(1) = y0;
82 vy = v0y;
83 x(1) = x0;
84 vx = v0x;
85
86 D = 0.5*C*p*A; % Constant for drag
87
88 for n = 1:N
89
90     v = sqrt(vx^2 + vy^2);
91     fnetx = -D*v*vx; % net forces acting on the baseball
92     fnety = -g*m - D*v*vy; % after initial launch
93
94     ay = fnety/m; % acceleration of y component of baseball
95     ax = fnetx/m;
96
```

```
97     y(n+1) = y(n) + vy*dt + (1/2)*ay*dt^2;    % vy = y', ay = y''
98     x(n+1) = x(n) + vx*dt + (1/2)*ax*dt^2;
99     vy = vy + ay*dt;    % vy(n+1) = vy(n) + ay*dt
100    vx = vx + ax*dt;
101
102 end
103
104 xt = xt * m2feet;    % convert everything to feet
105 yt = yt * m2feet;
106 y = y * m2feet;
107 x = x * m2feet;
108
109 % check to see that y = yt and x = xt, , point by point
110
111 checkSumy = sum(abs(y-yt))
112
113 checkSumx = sum(abs(x-xt))
114
115
116
117 % ----- plot ---- %
118
119 plot(xt, yt, x, y, 'LineWidth', 2)
120
121
122
123 grid on
124 grid minor
125
126 ax = gca; ax.FontSize = 15; ax.GridAlpha = 0.5; ax.MinorGridAlpha = 0.45;
127 ax.MinorGridLineStyle = '-';
128
129 xlabel('x (ft)', 'FontSize', 18)
130
131 ylabel('y (ft)', 'FontSize', 18)
132
133 title({'ECE 202 Project 2, Phase 2: Trajectory of a baseball', ...
134       'With vs. Without Drag'}, 'FontSize', 22)
135
136 legend('Without Drag', sprintf('With Drag, C = %g', C), 'FontSize', 18)
137
138 ylim([-20 150])
139
140
```

ECE 202 Project 2, Phase 2: Trajectory of a baseball With vs. Without Drag



```
>> Project2Phase2  
Enter C value for drag coefficient: 0
```

```
checkSumy =  
    1.0422e-08
```

```
checkSumx =  
    2.1102e-08
```

```
>>
```