```
% Furquaan Syed
% 12/8/2021
% Professor Leonard
% ECE 202 Fall 2021 Project 2
% Using numerical techniques to calculate range, height and flight time
% of a baseball

% close collaborator with Ryan Palmer

        %---Phase 2---%

% Adding air resistance (drag) and checking when drag is “turned off”


clear
clf
format shortg

%--- Parameters ---%

theta = 32; % launch angle in degrees
v0 = 112;   % Exit velocity in miles per hour (mph)

x0 = 0; y0 = 0;  % initial position of the ball

g = 10;   % gravitational constant in N/kg (1 N/kg = 1 m/s^2)

m = 0.145;     % mass of baseball in kilograms

p = 1.225;   %density of air in kg/m^3
A = pi() * 0.0365^2; %cross sectional area of a baseball m^2

C = input('Enter C value for drag coefficient: ');

mph2mps = 5280 * 12 * 2.54 / 100 / 3600;   % mph to m/s conversion
deg2rad = pi()/180;   % degrees to radians
m2feet = 3.281;     %meters to feet

    % add a conversion from m to ft, e.g. m2ft, and use it several times


v0mps = v0 * mph2mps;       % Exit velocity in meters per second
thetaRad = theta * deg2rad;   % launch angle in radians

v0x = v0mps * cos(thetaRad);   % x-component of v0
v0y = v0mps * sin(thetaRad);   % y-component of v0
```

```
% ----- compute some useful characteristics of trajectory -----

tH = v0y/g;    % time to reach max. height
tLand = 2*tH;   % time to land (time of flight)

H = tH * v0y/2;   % max. height
R_ft = v0x * tLand;   % range

R_m = R_ft / m2feet;    % ROUGH conversion from m to ft


%--- analytical x(t) and y(t)---%

tmin = 0; tmax = tLand;
N = 2000;   % intervals

t = linspace(tmin, tmax, N+1);   % time array, connects x(t) with y(t)

        %--- calculations for no drag---%

xt = x0 + v0x*t;   % x(t), ax = 0 (no drag)
yt = y0 + v0y*t - (1/2)*g*t.^2;   % y(t), ay = -g (no drag)


% ----- calculations for drag -----%


dt = (tmax-tmin)/N;     %change in time interals

y = zeros(1, N+1);   % initialize y(t) and x(t)
x = zeros(1, N+1);

y(1) = y0;
vy = v0y;
x(1) = x0;
vx = v0x;

D = 0.5*C*p*A;  % Constant for drag

for n = 1:N

    v = sqrt(vx^2 + vy^2);
    fnetx = -D*v*vx;      % net forces acting on the baseball
    fnety = -g*m - D*v*vy;          % after initial launch

    ay = fnety/m;   % acceleration of y component of baseball
    ax = fnetx/m;

```

```
    y(n+1) = y(n) + vy*dt + (1/2)*ay*dt^2;   % vy = y', ay = y"
    x(n+1) = x(n) + vx*dt + (1/2)*ax*dt^2;
    vy = vy + ay*dt;   % vy(n+1) = vy(n) + ay*dt
    vx = vx + ax*dt;

end

xt = xt * m2feet;       % convert everything to feet
yt = yt * m2feet;
y = y * m2feet;
x = x * m2feet;

% check to see that y = yt and x = xt, , point by point

checkSumy = sum(abs(y-yt))

checkSumx = sum(abs(x-xt))



% ----- plot ---- %

plot(xt, yt, x, y, 'LineWidth', 2)



grid on
grid minor

ax = gca; ax.FontSize = 15; ax.GridAlpha = 0.5; ax.MinorGridAlpha = 0.45;
ax.MinorGridLineStyle = '-';

xlabel('x (ft)', 'FontSize', 18)

ylabel('y (ft)', 'FontSize', 18)

title({'ECE 202 Project 2, Phase 2: Trajectory of a baseball', ...
    'With vs. Without Drag'}, 'FontSize', 22)

legend('Without Drag', sprintf('With Drag, C = %g', C), 'FontSize', 18)

ylim([-20 150])
```