```
% Ian Woodbury
% 12.13.2021
% ECE 202 Project 2: Hitting a home run, with air resistance, and
% calculating net force at each step
% Phase 3: Exporting data and analyzing it in Excel

clear; clf;

% ---- define given information -----

m = 0.145;  % mass of a baseball (kg)
v0mph = 112;   % exit velocity in mph
phi0deg = 32;    % launch angle in degrees

x0 = 0; y0 = 0;  % it doesn't really matter where the ball starts
% assume measurements in m to start

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

A = 0.00426;    % cross sectional area of a baseball, (m^2)
p = 1.225;      % density of air, in (kg/m^3)
C = input("Enter C value: \n");      % dimensional constant, C

% ----- set up more variables, and convertions -----

mph2mps = 5280 * 12 * 2.54 / 100 / 3600;   % mph to m/s conversion
deg2rad = pi()/180;   % conversion for degrees to radians
m2ft = 3.28;     % conversion for meters to feet

v0 = v0mph * mph2mps;       % converts v0 from mph to m/s
phi0 = phi0deg * deg2rad;   % converts launch angle from degrees to radians

v0x = v0*cos(phi0);   % x-component of v0 (m/s)
v0y = v0*sin(phi0);   % y-component of v0 (m/s)

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

D = (1/2)*C*A*p; % D for Drag, didn't want to compute this twice
% will multiply by speed and directional vector. (kg/m)

% ----- set up a time array, compute x(t), y(t) analytically -----

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

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

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)


% ----- add numeric solution -----

dt = (tmax-tmin)/N;

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

```matlab

y(1) = y0;
vy = v0y;   % vy(1) = v0y, i.e., no array is needed!

x(1) = x0;
vx = v0x;   % vy(1) = v0y, i.e., no array is needed!

for n = 1:N   % stop at N

    v = sqrt(vx^2 + vy^2);  % speed of the ball, given in m/s

    % net force of the ball
    Fnety = -m*g - D*v*vy; % net force on the y axis (N), -g with no drag
    Fnetx = 0 - D*v*vx;    % net force on the x axis (N), zero with no drag


    % updating position, velocity, and acceleration of
    % the ball on the y axis
    % acceleration (m/s^2)
    ay = Fnety/m;
    % position (m)
    y(n+1) = y(n) + vy*dt + (1/2)*ay*dt^2;   % vy = y', ay = y"
    % velocity (m/s)
    vy = vy + ay*dt;   % vy(n+1) = vy(n) + ay*dt

    % updating position, velocity, and acceleration of
    % the ball on the x axis
    % acceleration (m/s^2)
    ax = Fnetx/m;
    % position (m)
    x(n+1) = x(n) + vx*dt + (1/2)*ax*dt^2;   % vx = x', ax = x"
    % velocity (m/s)
    vx = vx + ax*dt;   % vx(n+1) = vx(n) + ax*dt


end

% ------ Checking ------\


% sum checks of anaylitic solution minus numeric solution
checky = sum(abs(yt-y))
checkx = sum(abs(xt-x))

% ------ Converting units for plotting ------

ytft = yt*m2ft; % all values converted from m to ft for plotting
xtft = xt*m2ft;
yft = y*m2ft;
xft = x*m2ft;

% ------ Exporting ------

export = [t; x; y].';
writematrix(export, 'P2_3.csv')

% ------ Plotting ------

```

```
plot(xtft, ytft, xft, yft, 'LineWidth', 2)
grid on
ax = gca; ax.FontSize = 15; ax.GridAlpha = 0.4;
grid minor
ax.MinorGridAlpha = 0.5;
xlabel('x (ft)', 'FontSize', 18)
ylabel('y (ft)', 'FontSize', 18)
title({'ECE 202, Project 2 Phase 3: Trajectory of a baseball', ...
    'with drag vs no drag'}, 'FontSize', 22)
legend({'no drag', sprintf('with drag, C = %g', C)}, ...
    'FontSize', 18)
ylim([-2 140])   % add a little space on the bottom, more on top for legend

```