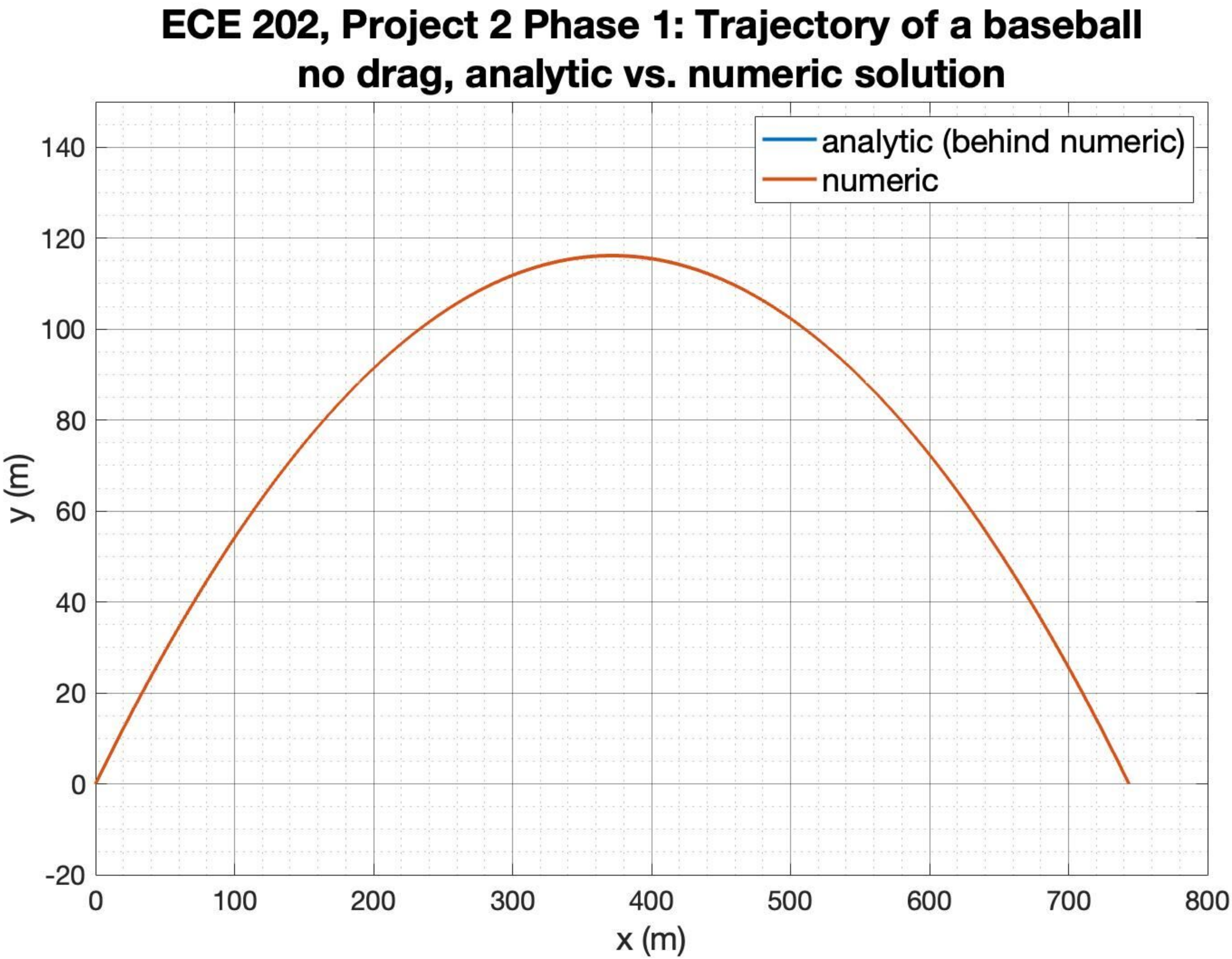
```
% Furquaan Syed
% 12/8/2021
% Professor Leonard
% Close collaborator with Ryan Palmer (and Professor Leonard)
% ECE 202 Fall 2021 Project 2
% Using numerical techniques to calculate range, height and flight time
% of a baseball


        %---Phase 1---%

% Comparing the analytic solutions to the numeric, without drag


clear
clf
format shortg

%--- Parameters ---%

phi0_deg = 32; % launch angle in degrees
v0mph = 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 (kg)


mph2mps = 5280 * 12 * 2.54 / 100 / 3600;   % mph to m/s conversion
deg2rad = pi()/180;   % degrees to radians
m2feet = 3.3;   %Convert from meters to feet



v0 = v0mph * mph2mps;       % Exit velocity in meters per second
phi0 = phi0_deg * deg2rad;   % launch angle in radians

v0x = v0 * cos(phi0);   % x-component of v0
v0y = v0 * sin(phi0);   % 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), y(t)  -----

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;     %change in time interals

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

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

for n = 1:N   % stop at N
    F_x = 0;             % net forces acting on the baseball
    F_y = -g*m;          % after initial launch

    ay = F_y/m;   % acceleration of y component of baseball
    ax = F_x/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;
yt = yt * m2feet;
y = y * m2feet;
x = x * m2feet;
```

```matlab

% 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;

xlabel('x (m)', 'FontSize', 18)
ylabel('y (m)', 'FontSize', 18)

title({'ECE 202, Project 2 Phase 1: Trajectory of a baseball', ...
    'no drag, analytic vs. numeric solution'}, 'FontSize', 22)

legend({'analytic (behind numeric)', 'numeric'}, ...
    'FontSize', 18)

ylim([-20 150])
```

```
>> Project2Phase1

checkSumy =

   1.0482e-08


checkSumx =

   2.1225e-08

>>
```


ECE 202, Project 2 Phase 1: Trajectory of a baseball
no drag, analytic vs. numeric solution
analytic (behind numeric)
numeric
y (m)
x (m)
140
120
100
80
60
40
20
0
-20
0
100
200
300
400
500
600
700
800