To: D. W. Mueller, Jr., Ph.D., P.E.

From: Mason Averill

Subject: Special Problem 1

Date: 29 August 2020

PURDUE UNIVERSITY FORT WAYNE

Purpose:

The purpose of this memo is to communicate the methodology and results of Special Problem 1 for ME-544: Modeling and Simulation of Mechanical Engineering Systems, completed August 29th 2020.

Purpose and Scope of Assignment:

The purpose of this assignment was to determine the minimum initial velocity required to hit a 600ft home run in baseball with drag forces considered. Other parameters considered included: mass of baseball, diameter of baseball, air density, gravity, angular velocity of the baseball (ball spin), wind speed, player height (initial height), initial hit angle with respect to the ground, and variable desired distance (600ft was the objective of the assignment, but the model allows for other choices of distance as well).

Mathematical Modeling of Problem:

First, the forces acting on the baseball during flight were determined, as shown in Figure 1.

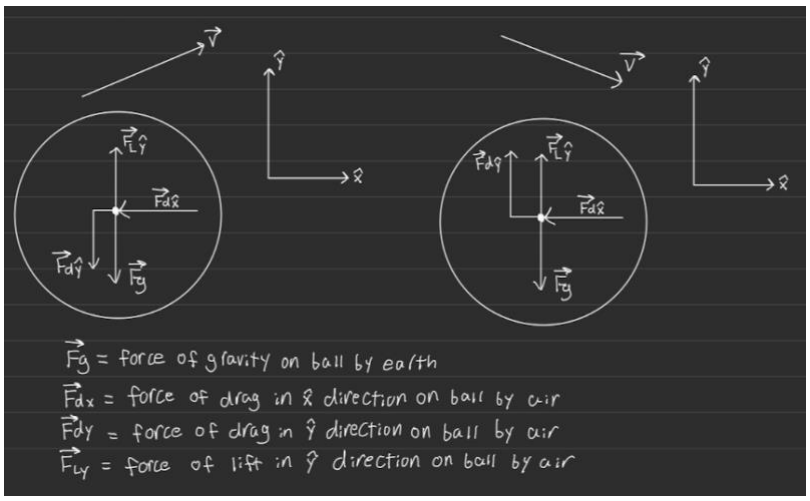


Figure 1: Forces Acting on Baseball During Flight

Of the four different forces identified to be acting on the ball during flight, only the force of gravity is easily known. To determine the other forces, Equation 1 was utilized.

$$C_D \equiv \frac{F_D}{\frac{1}{2}\rho V^2 A} => F_D \equiv \frac{1}{2}\rho V^2 A C_D$$

Equation 1

With:

- $F_D = drag\ force$
- $C_D = drag\ coefficient$
- $\rho = density\ of\ fluid$
- $V = velocity$
- $A = maximum\ cross - sectional\ area$

**Note that this expression is also valid for lift, the only difference being a coefficient of lift is utilized instead of a coefficient of drag

Of the variables identified in Equation 1, only the drag coefficient proves to be relatively difficult to determine. To determine the drag coefficient, the plot shown in Figure 2 was utilized. Note that the baseball was modeled as a smooth sphere in order to utilize the data in the plot.

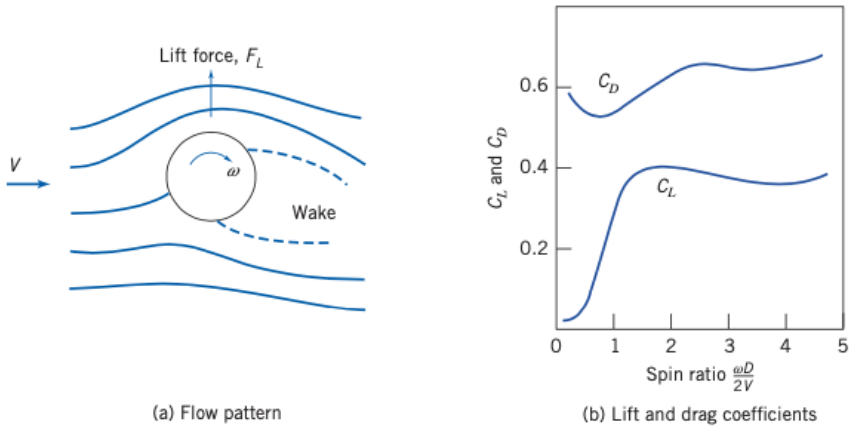


Fig. 9.27 Flow pattern, lift, and drag coefficients for a smooth spinning sphere in uniform flow. (Data from Reference [19].)

Figure 2: Lift and Drag Coefficients for A Smooth Spinning Sphere [3]

With:

- ω=angular velocity of the ball in rad/s
- D=diameter of the ball
- V=velocity of the ball relative to the uniform flow of the fluid

The plot in Figure 2 proves to be incredibly useful to determine the drag and lift coefficients for single values of spin ratio. However, since this project would involve using computational means to determine the drag and lift coefficient for varying values of velocity (all other parameters were kept constant during flight), a continuous function was needed to determine the value of the drag and lift coefficients as a function of spin ratio. To develop a continuous function, the following steps were performed:

Step 1: An image of the plot was imported into excel.

Step 2: A finely incremented grid was overlayed on the image with the same axis bounds, alignment, and size.

Step 3: Data points were gathered for the drag and lift coefficient at tenths increments in spin ratio

Step 4: Two trendlines, each consisting of a polynomial of order 10, were created to match the data gathered.

Step 5: The data gathered was plotted in orange circular dots and the trendline was plotted as a yellow line.

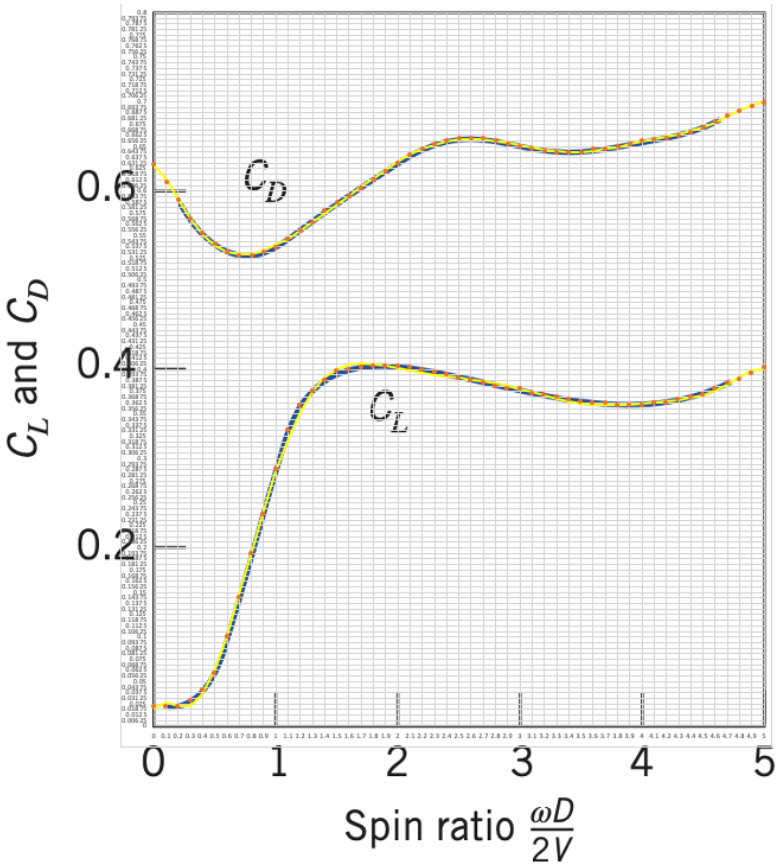


Figure 3: Trendline Fit For Drag and Lift Coefficient Data

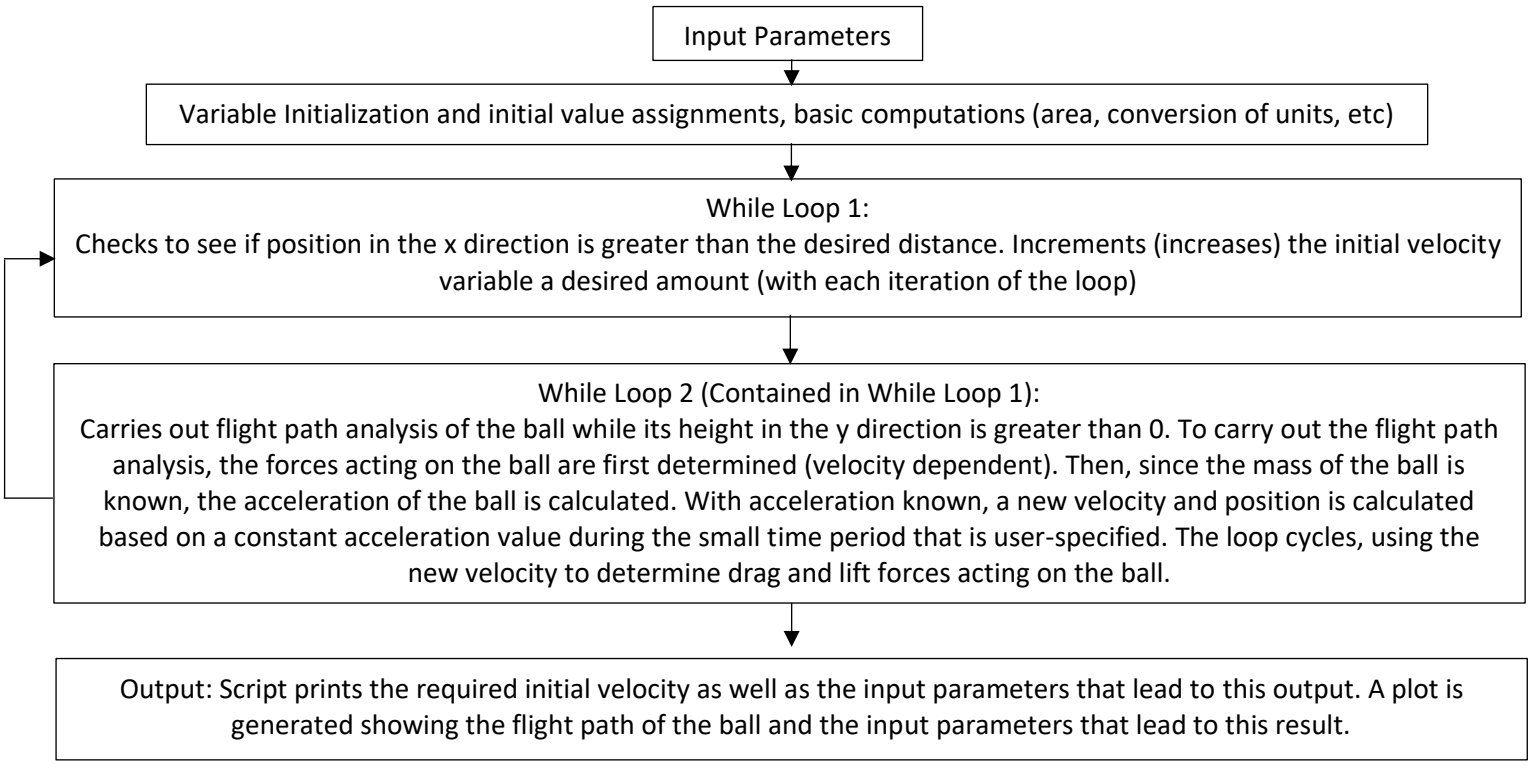
As can be seen in Figure 3, the trendline fits for the data match the original data very well.

By solving the spin ratio expression for velocity and utilizing expected values for angular velocity and diameter it was found that when the value of the spin ratio was 0.05 the associated velocity was 92.5m/s (206.9mph) and when the value of the spin ratio was 5 the associated velocity was less than 1 m/s (2.24mph). A quick analysis of this problem without drag indicates that the required minimum initial velocity is 42m/s (94mph). This demonstrates that the function used to determine the drag and lift coefficients will be valid over almost the entirety of the flight path of the ball (the exception being when the velocity in the Y direction falls to below 1m/s). This was planned to be taken into account in the model by only considering drag and lift forces when the spin ratio value was less than 5 and greater than 0.05.

Development of Simulation:

To computationally determine the required minimum initial velocity, MATLAB was used. A script was written which allows for easy modification of the input parameters mass of baseball, diameter of baseball, air density, acceleration due to gravity, angular velocity of the baseball, wind speed, player height, hit angle with respect to the ground, and desired hit distance.

The MATLAB script operates in the following manner:



Verification/Validation of Model

The mass and diameter of a baseball was found to be 145 grams and 74 mm, respectively [2]. The angular velocity of a baseball tends to be between 1500 and 4500 rpm, or 157 and 471 rad/s, respectively [4]

The model was verified by determining the initial velocity required to hit a 400 ft home run, an average home run distance, that has typical initial velocities of ~100mph [1]. The following Figures show the results with varying input parameters (baseball mass of 145g, baseball diameter of 74mm, air density of 1.225kg/m^3, and player height of 6' 3" with a 1 ft drop to bat contact height were kept constant).

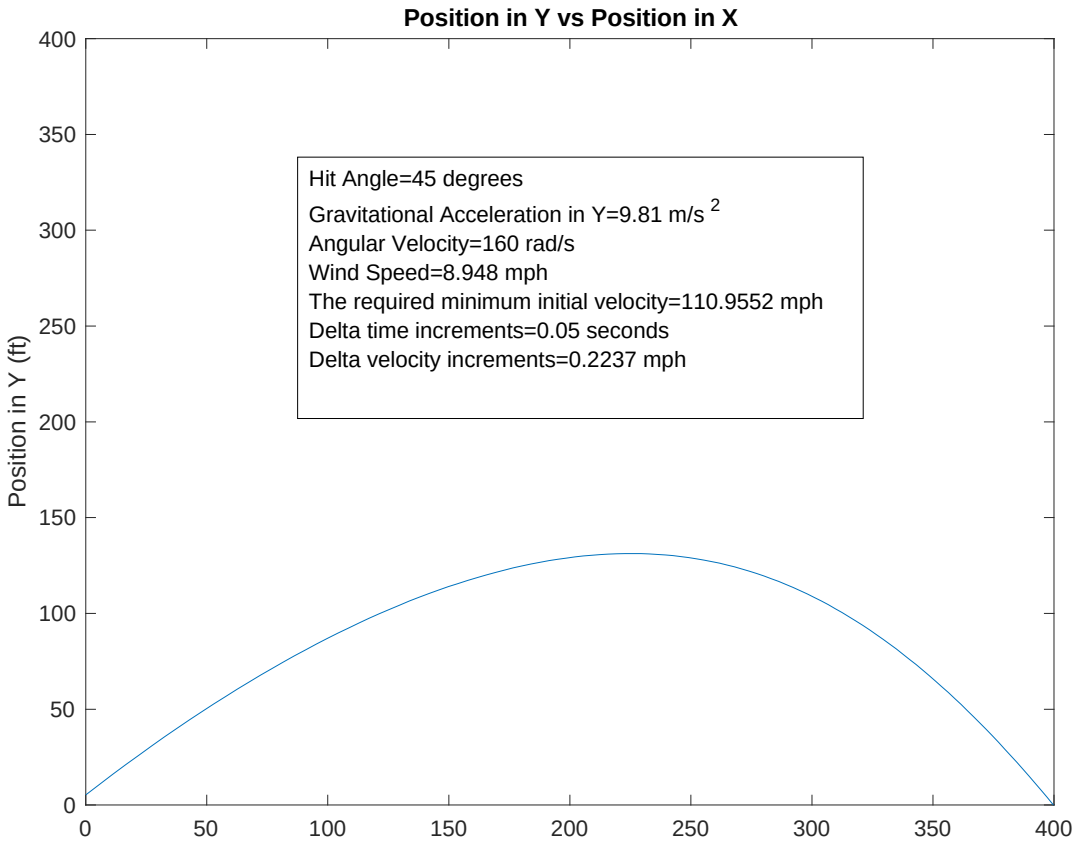


Figure 4: Typical Input Parameters, Relatively Low Angular Velocity

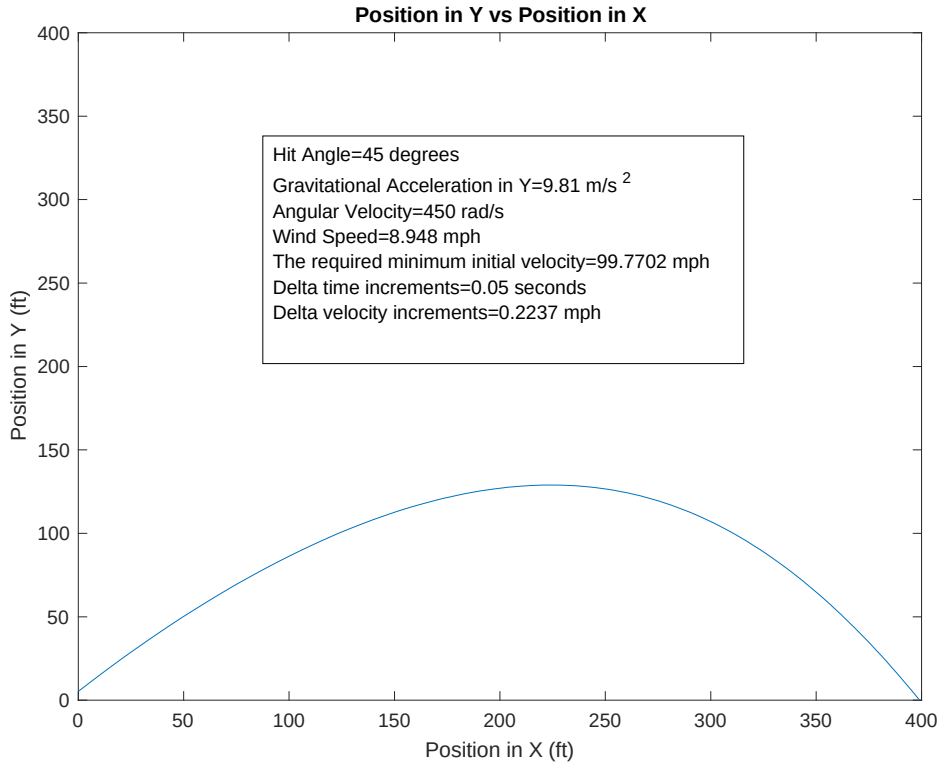


Figure 5: Typical Input Parameters, Relatively High Angular Velocity

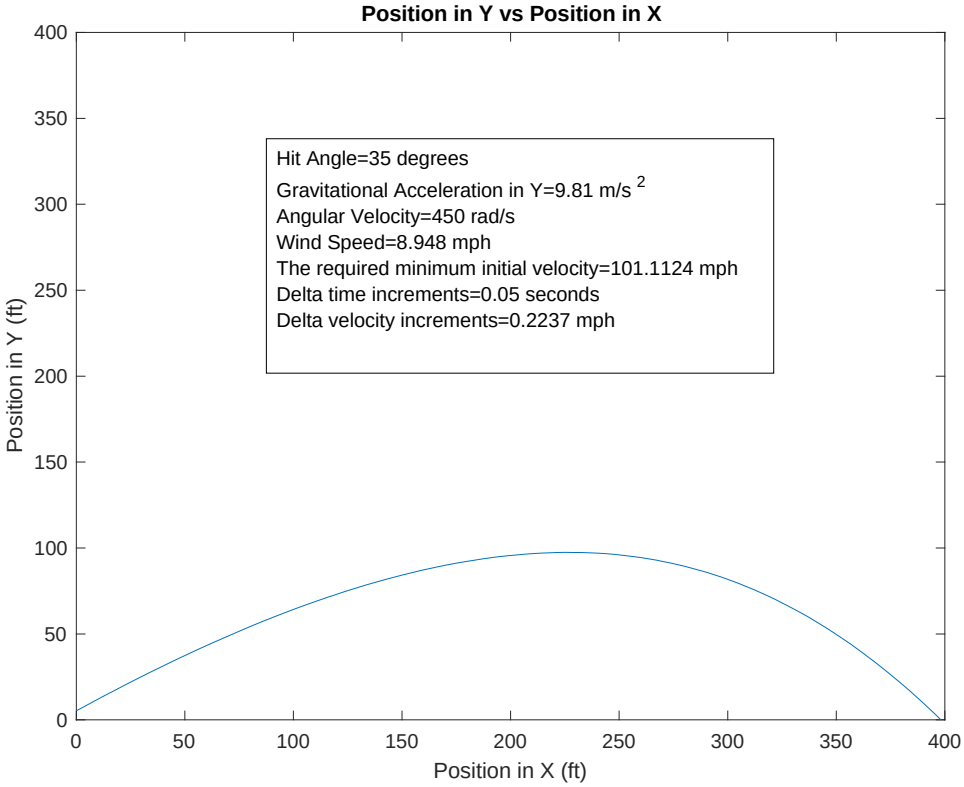


Figure 6: Typical Input Parameters, Relatively Low Hit Angle

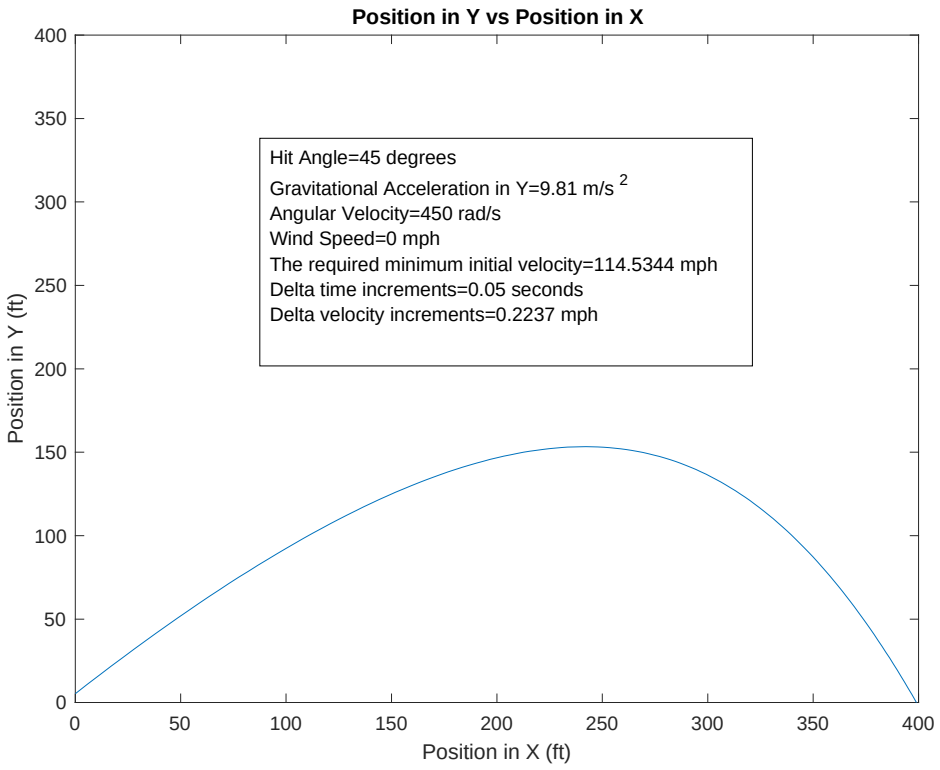


Figure 7: Typical Input Parameters, No Wind Speed

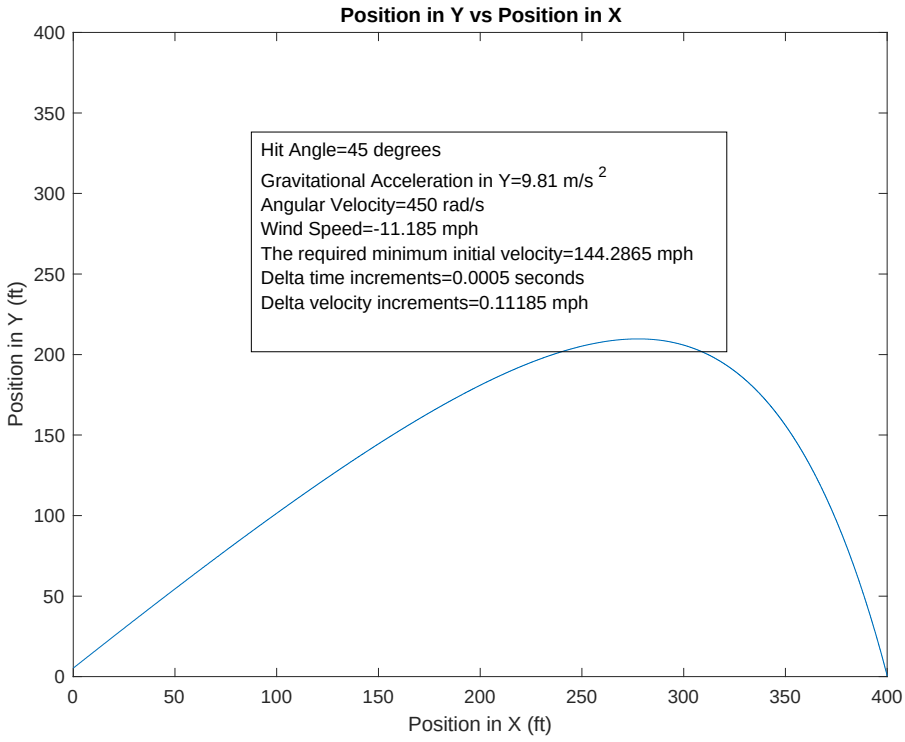


Figure 8: Typical Input Parameters, Strong Head-Wind

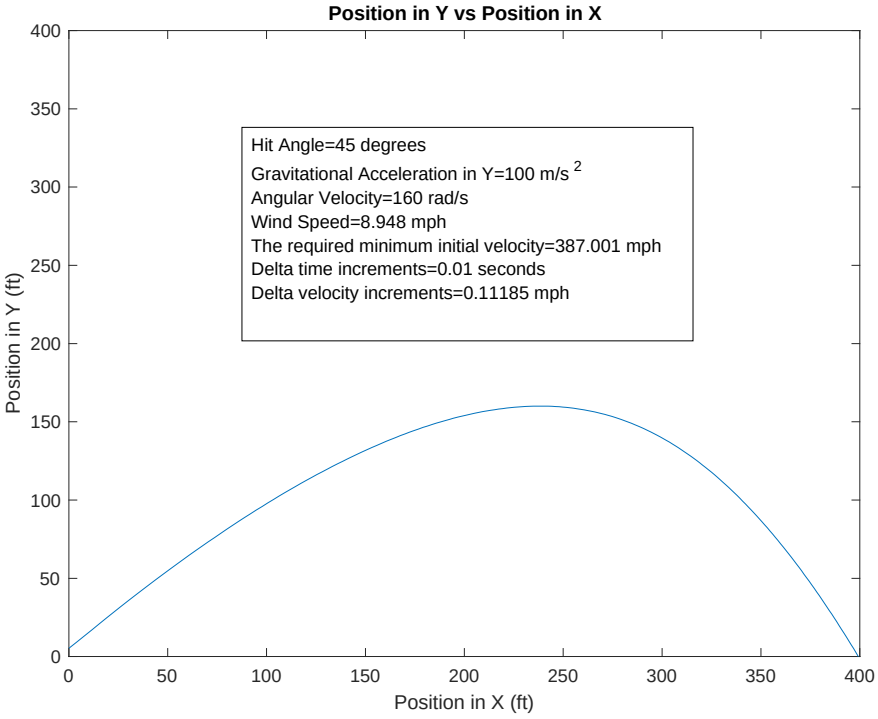


Figure 9: Very High Gravitational Acceleration

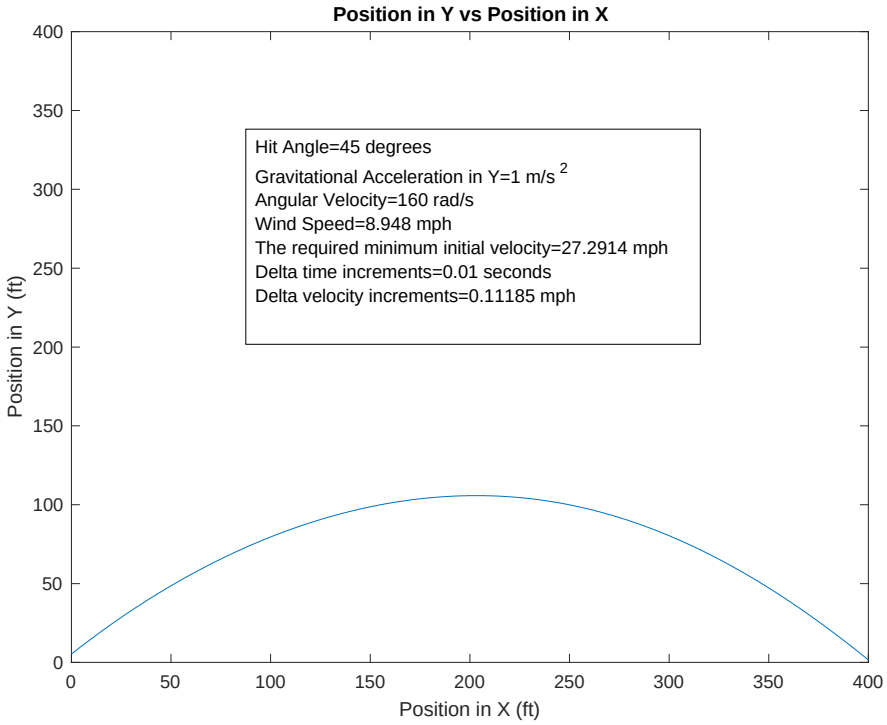


Figure 10: Very Low Gravitational Acceleration

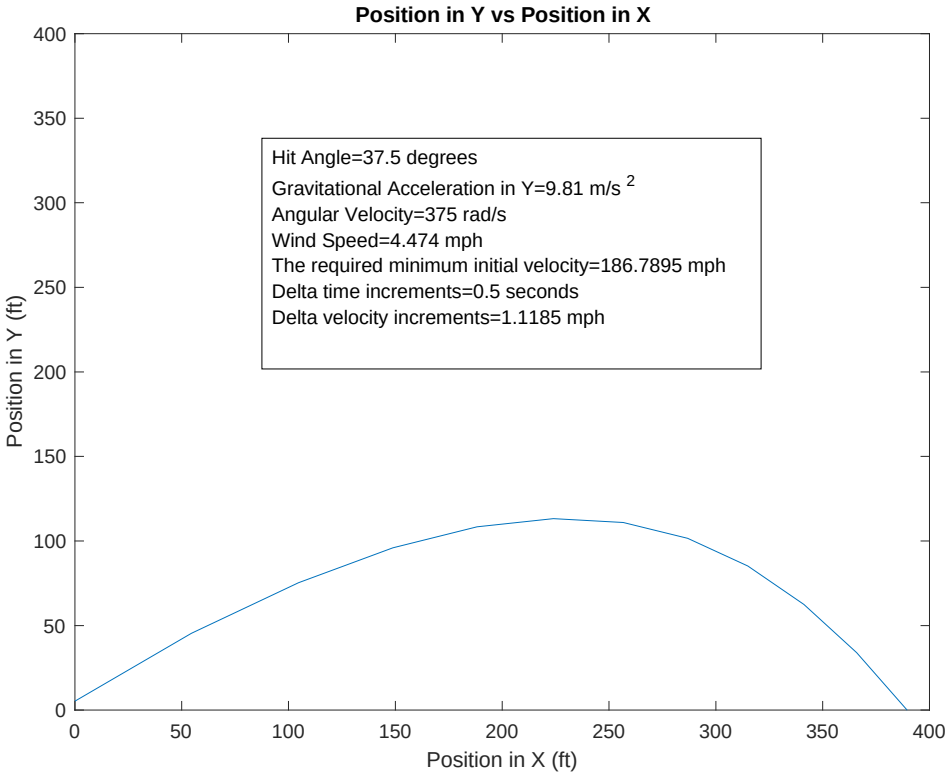


Figure 11: Convergence Demonstration-1

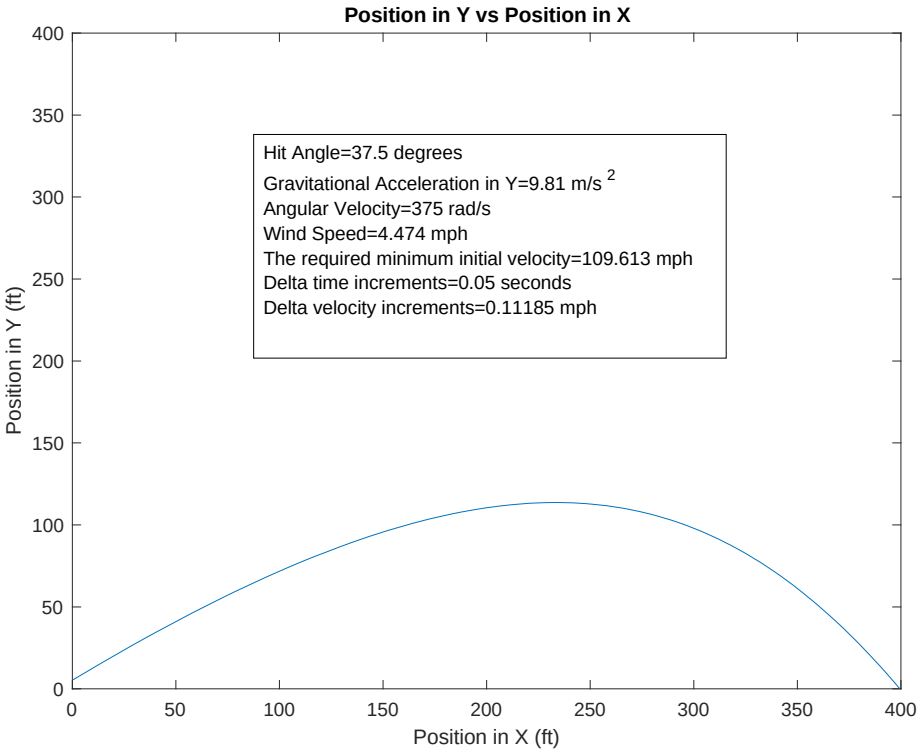


Figure 12: Convergence Demonstration-2

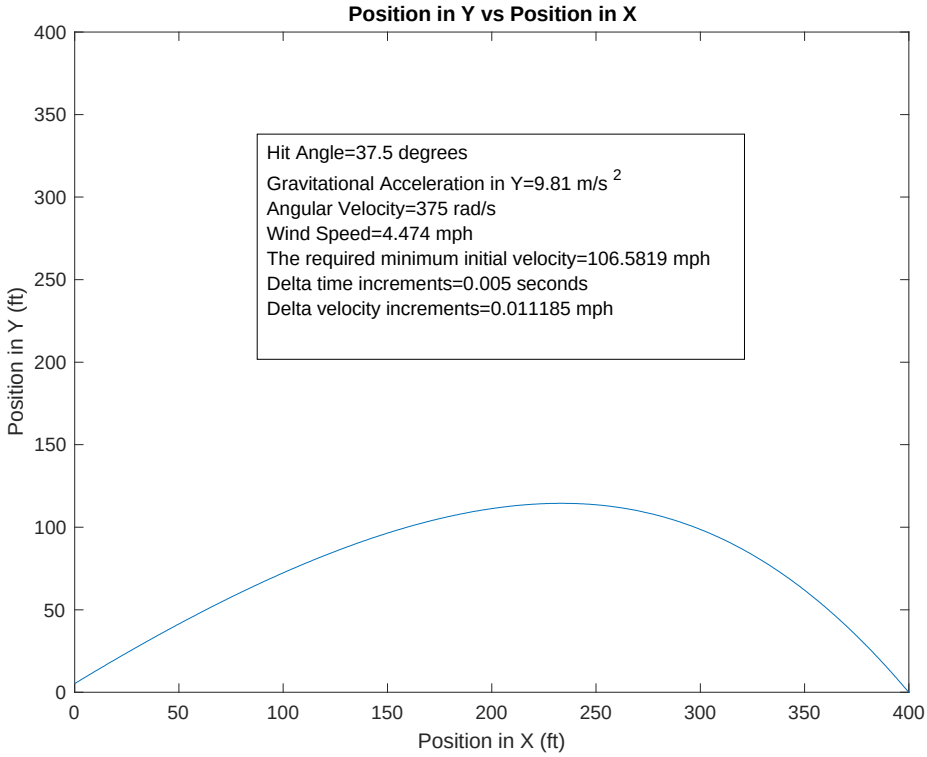


Figure 14: Convergence Demonstration-3

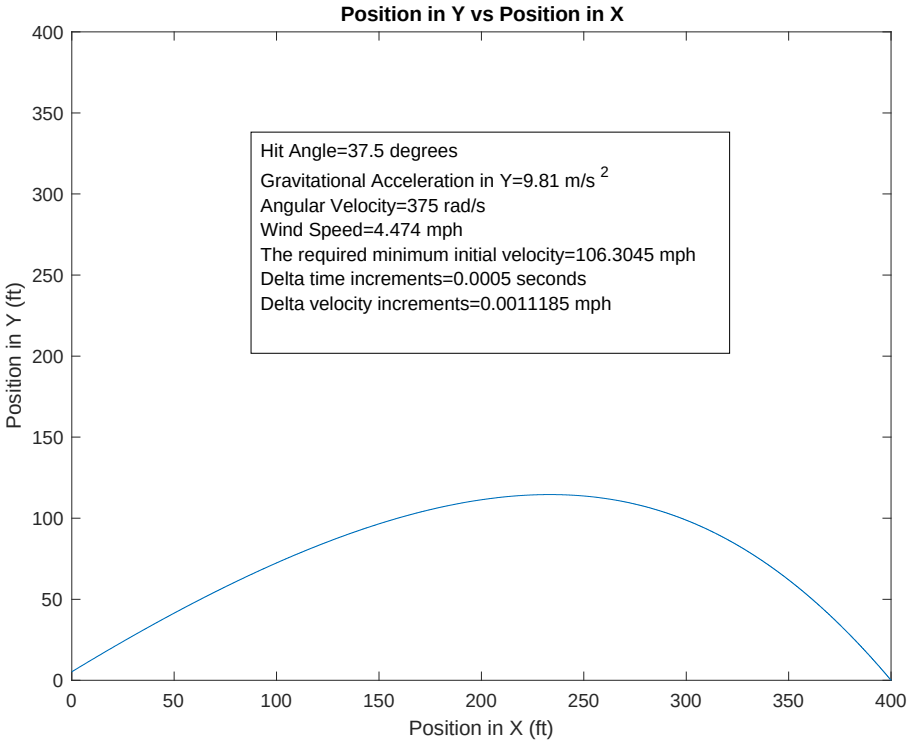


Figure 13: Convergence Demonstration-4

The above Figures demonstrate that not only does the model behave as expected, but it also converges as the resolution of the model increases.

Execution of Simulation to Determine Initial Velocity Required to Hit a 600ft Home Run:

Maintaining the same input parameters of baseball mass, baseball diameter, air density, and initial hit height, the following results were obtained for the minimum velocity required to hit a 600 ft home run for multiple configurations of other available input parameters.

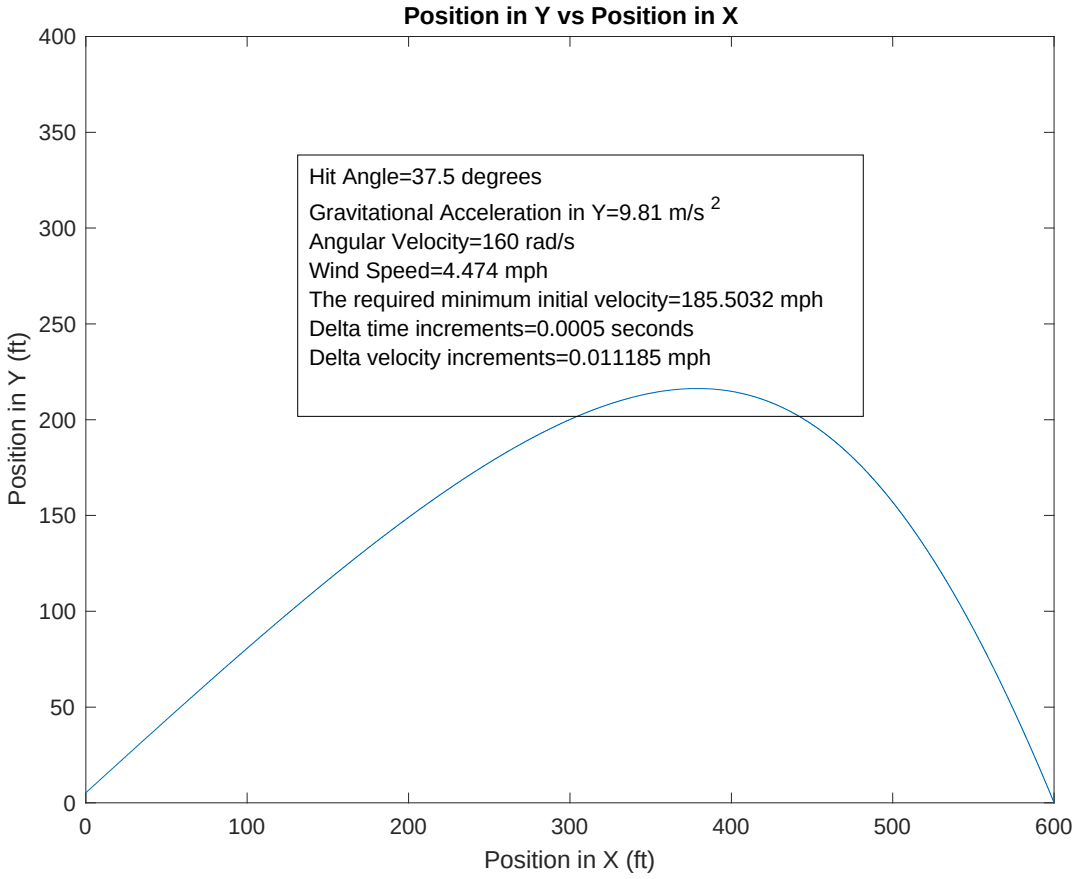


Figure 15: Typical Input Parameters, Relatively Low Rotational Velocity

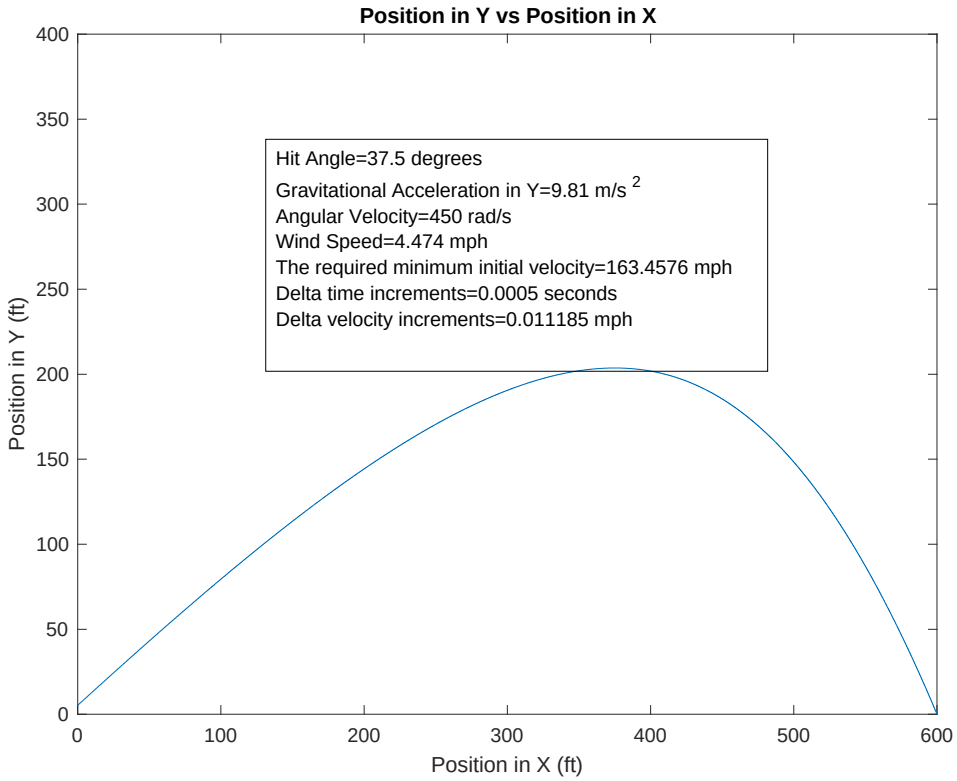


Figure 16: Typical Input Parameters, Relatively High Rotational Velocity

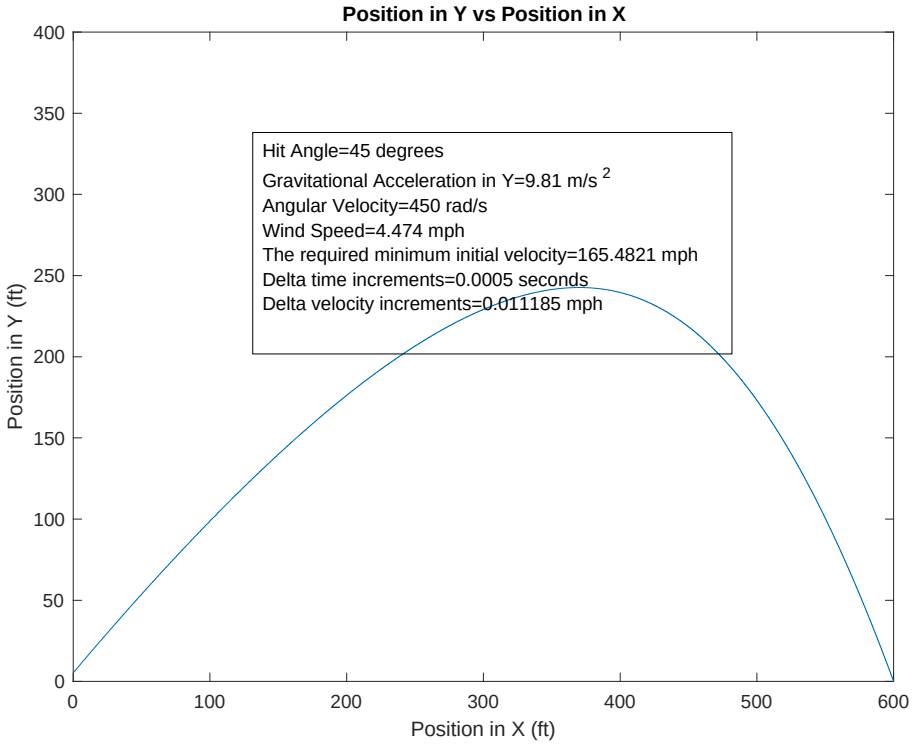


Figure 17: Typical Input Parameters, Relatively High Rotational Velocity, 45 Degree Hit Angle

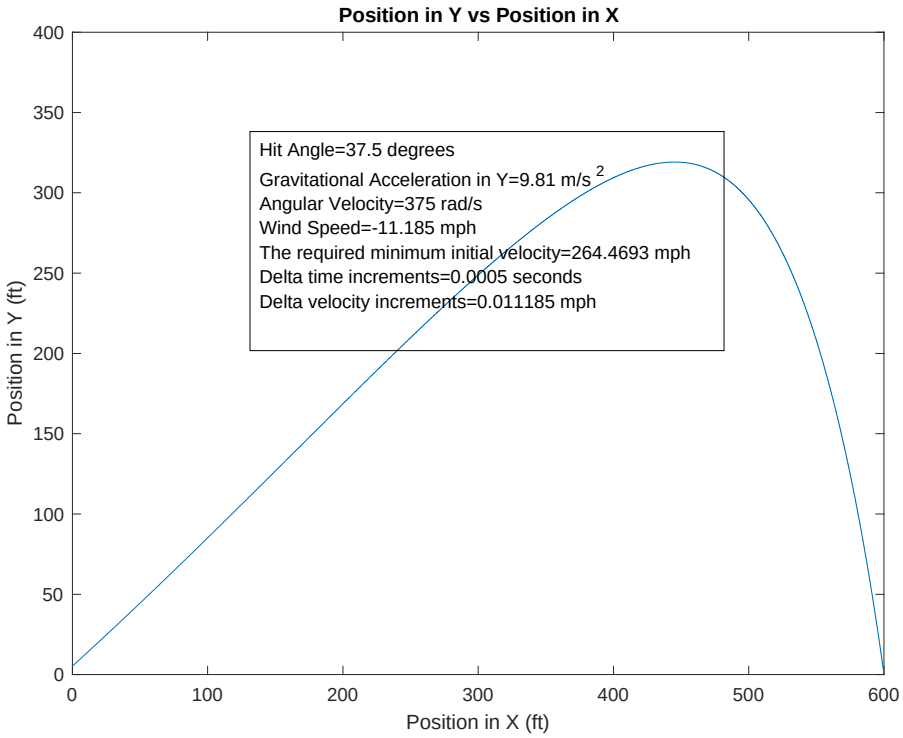


Figure 18: Strong Headwind

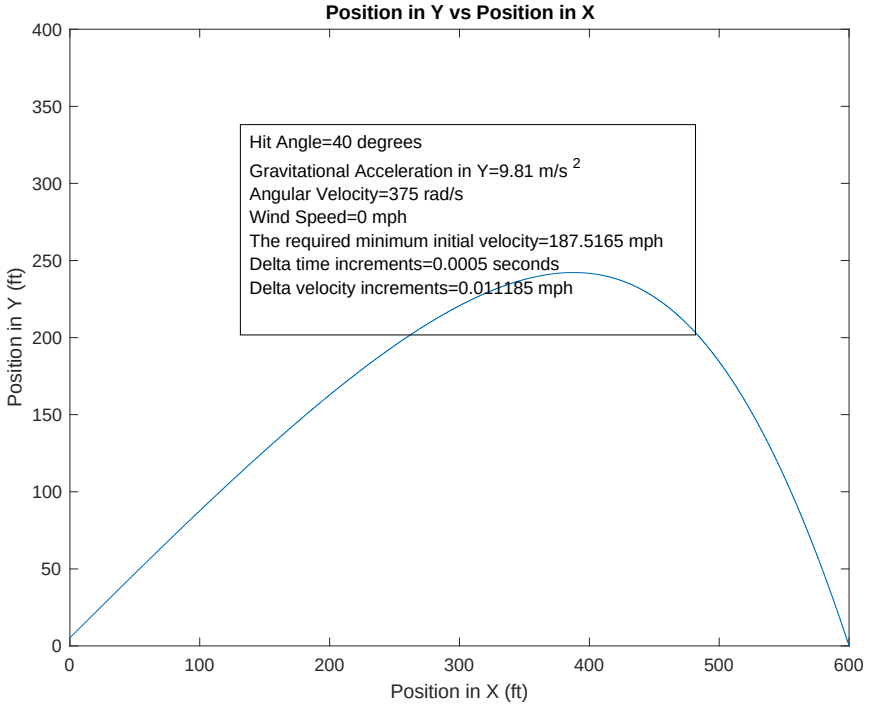


Figure 19: No Wind Speed

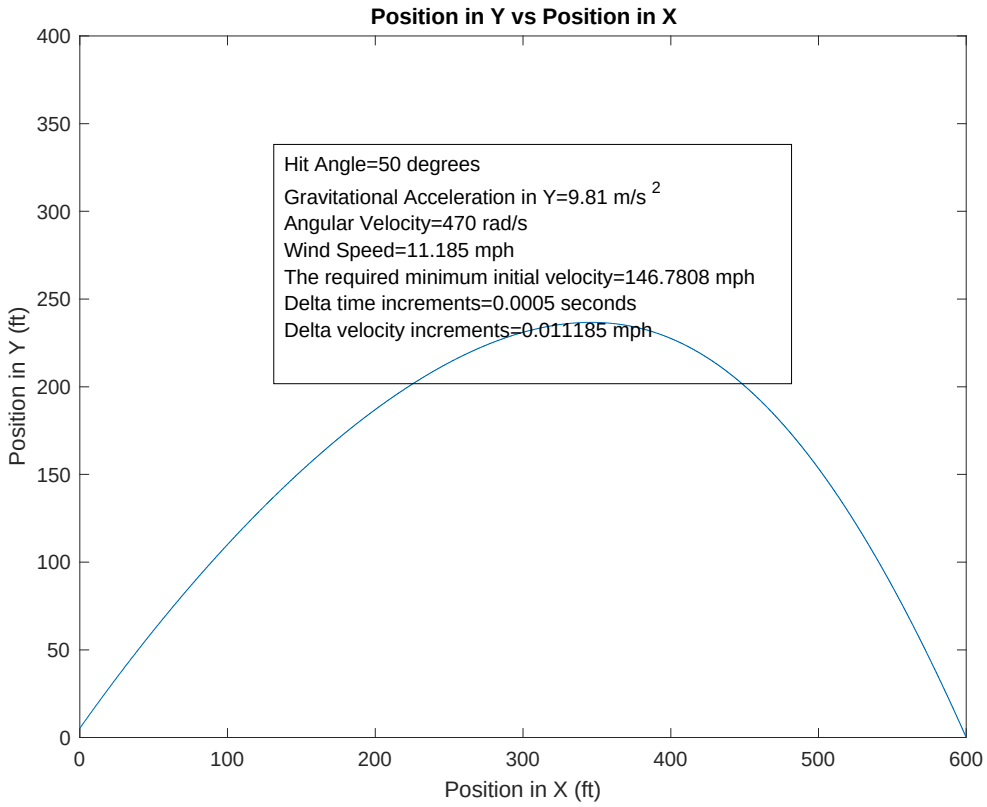


Figure 20: Ideal Conditions, Experimentally Optimizing Hit Angle-1

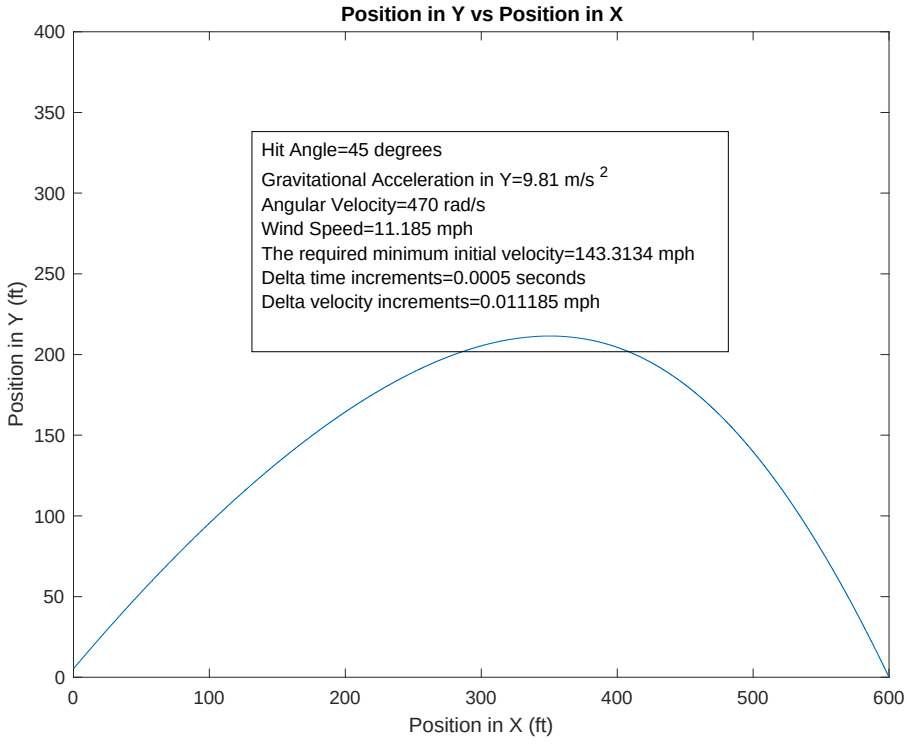


Figure 21: Ideal Conditions, Experimentally Optimizing Hit Angle-2

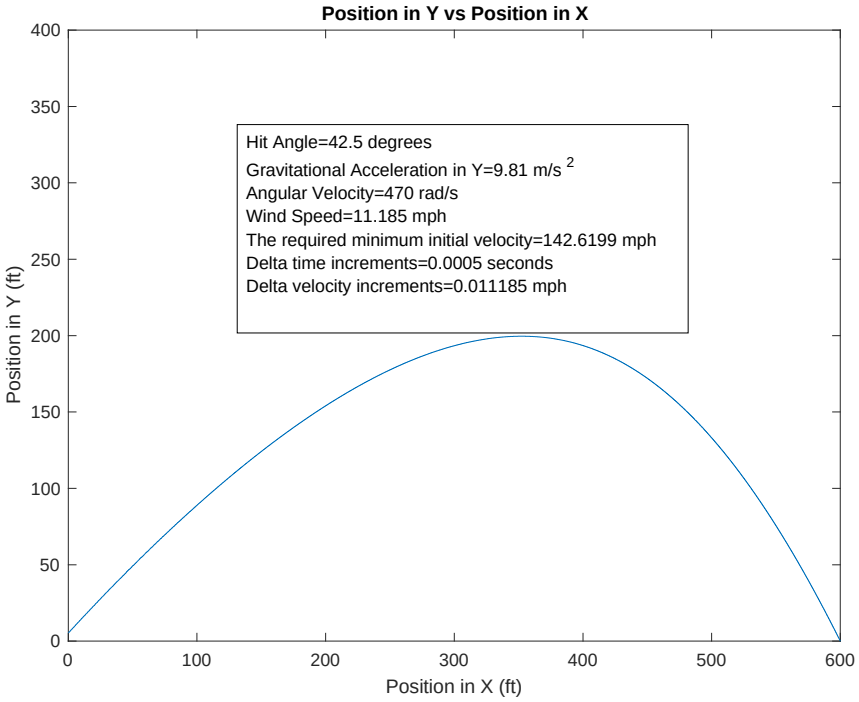


Figure 22: Ideal Conditions, Experimentally Optimizing Hit Angle-3

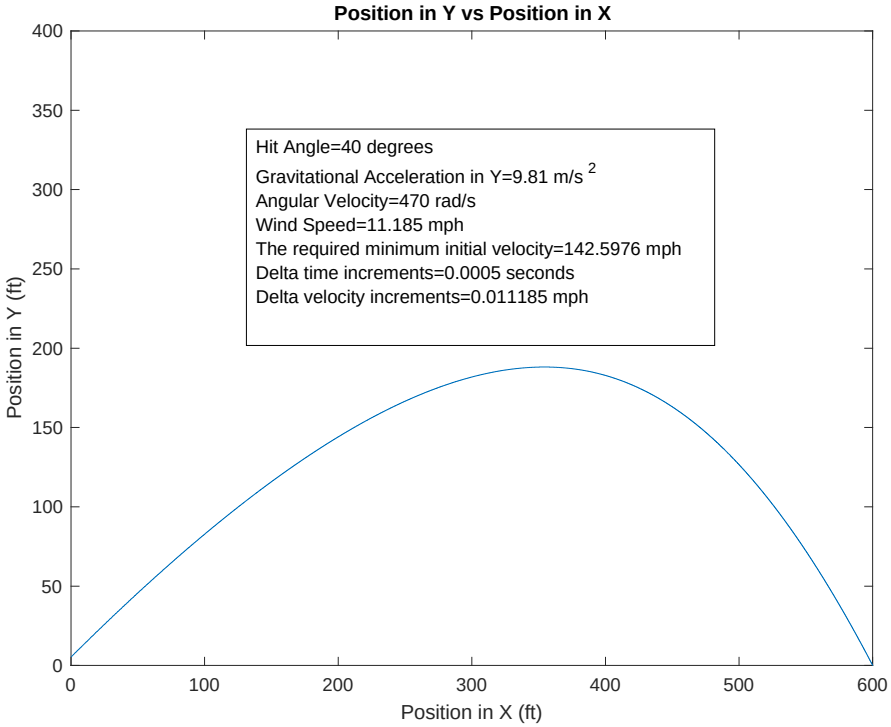


Figure 23: Ideal Conditions, Experimentally Optimizing Hit Angle-4

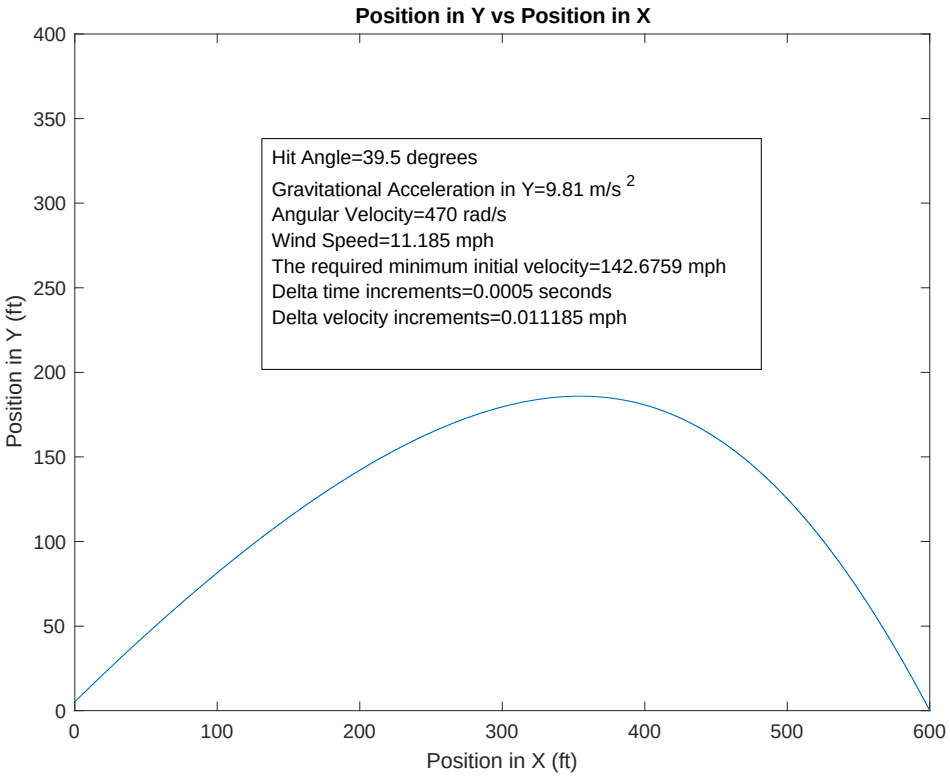


Figure 24: Ideal Conditions, Experimentally Optimizing Hit Angle-5

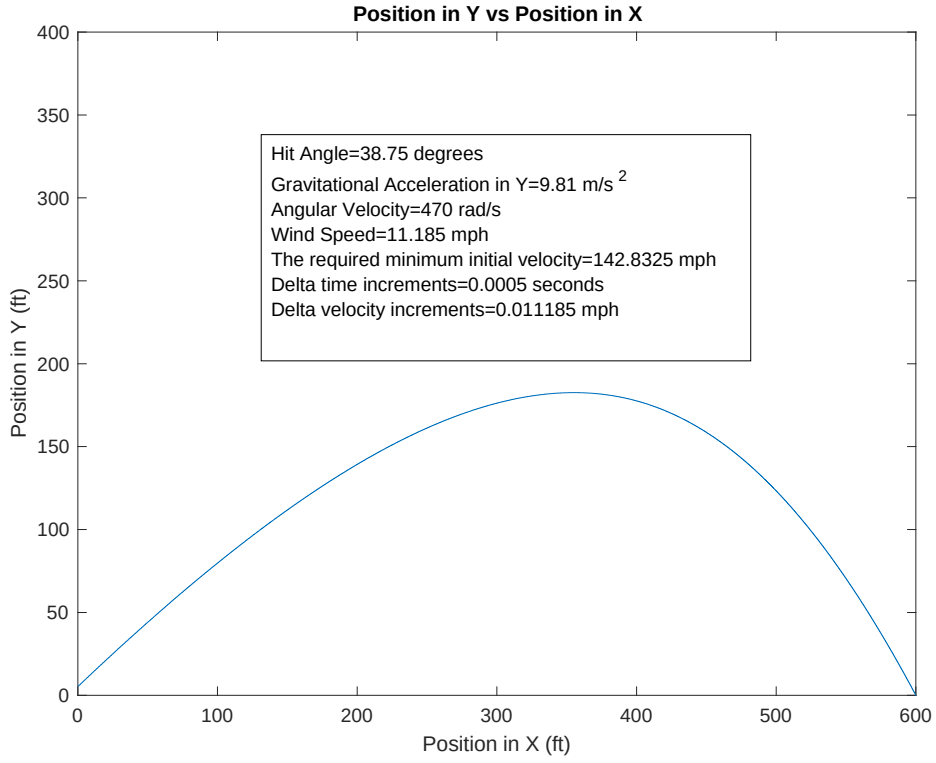


Figure 25: Ideal Conditions, Experimentally Optimizing Hit Angle-6

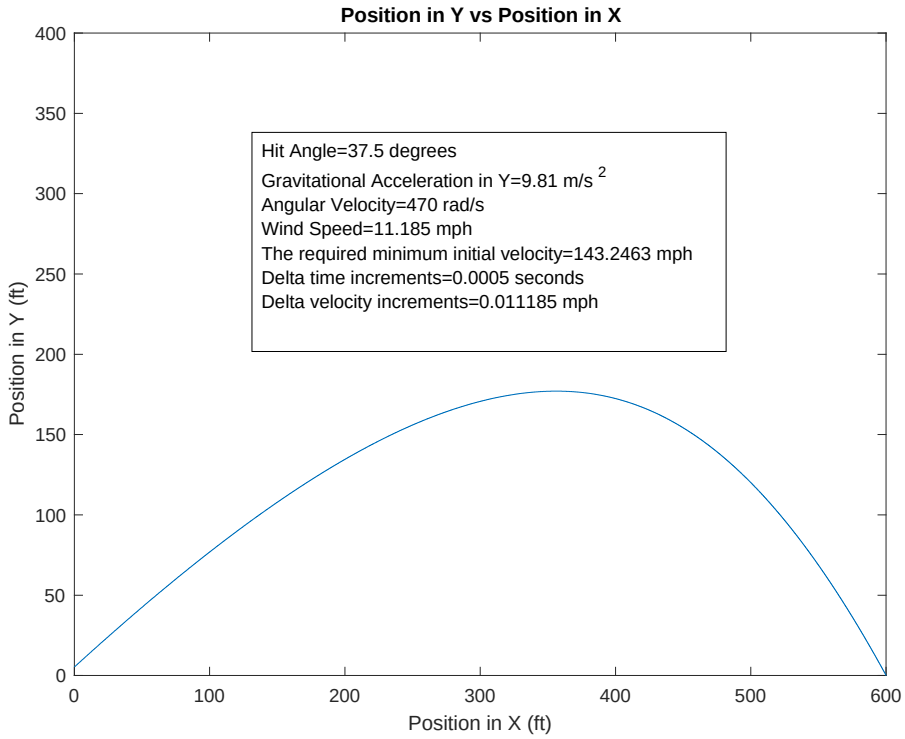


Figure 26: Ideal Conditions, Experimentally Optimizing Hit Angle-7

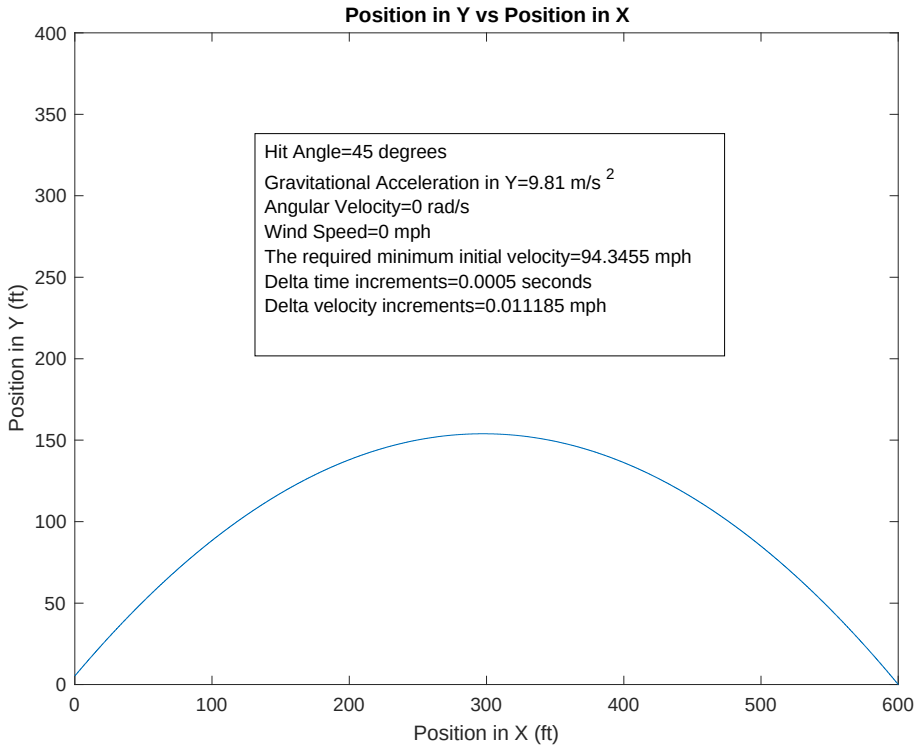


Figure 27: No Drag Case, Optimum Hit Angle

Conclusion:

After analyzing the above Figures, it becomes apparent that the answer to the question “Determine the minimum initial velocity required to hit a 600ft home run” can vary relatively drastically depending upon environmental conditions and the spin of the ball. Under “ideal” environmental conditions, and with the hit angle experimentally optimized, the answer to the question is 142.6 mph.

Some interesting things to note from the above Figures is how drastic of an influence wind speed has. In addition, the angular velocity of the ball is a significant factor. It is also interesting, though somewhat expected, that the ideal hit angle is not 45 degrees, as is the case when no drag is considered. Rather it is approximately 40 degrees. It is also quite astounding how drastic of an affect drag has on a baseball’s flight. This can be seen by viewing Figure 27, which is a no-drag case. The minimum required initial velocity for this scenario is just 94.3 mph. This is approximately 48.3 mph less than the ideal drag-considered case, or just 66.1% of the speed of the ideal drag-considered case.

References:

1.) (n.d.). Retrieved from http://baseball.physics.illinois.edu/PointC.html

2.) Baseball (ball). (2020, July 16). Retrieved from https://en.wikipedia.org/wiki/Baseball_(ball)

3.) Fox, R. W., McDonald, A. T., Pritchard, P. J., & Mitchell, J. W. (2015). *Fox and McDonalds introduction to fluid mechanics*. Hoboken, NJ: Wiley.

4.) Nathan, A. M. (2007, October 13). *The effect of spin on the flight of a baseball*[Scholarly project]. Retrieved August 30, 2020.

MATLAB Script:

```
%Mason Averill
%ME-544 Modeling and Simulation of ME Systems: Fall 2020
%Special Project 1: Initial Velocity Required for 600ft Home Run W/
%Drag/Air Resistance
%08_29_2020


%Environmental and Physical Parameters

mass=0.145; %mass of baseball in kg
diameter=0.074;%diameter of baseball in m
air_density=1.225; %density of ambient air in kg/m^3
gravity=9.81; %acceleration due to gravity in m/s^2

%Other Input Parameters
angular_velocity=160;%angular velocity of baseball in rad/s (at 0 no drag
will be considered, as function for determining drag coefficients is no
longer valid)
wind_speed=2; %wind speed in m/s, + for same direction as baseball travel, -
for opposite baseball travel
player_height=1.9; %height of player in m
hit_angle=40; %initial launch angle w.r.t the ground in degrees
hit_distance=600;%desired hit distance in feet

%convert hit_angle to radians for future use

hit_angle=hit_angle*pi()/180;

area=pi()*diameter^2/4;%effective area of interest for drag force
calculations



%Initialize lower bound for initial velocity

initial_velocity_ground=30; %initial velocity (relative to the ground) guess
in m/s (your CPU will appreciate a close guess at high resolutions)


%%LOOP

sx=0; %Initial position in x
delta_velocity=0.005;

while(sx<hit_distance/3.281)


    initial_velocity_ground=initial_velocity_ground+delta_velocity;

%Initialize initial conditions
```

```
sx=0; %Initial position in x
sy=player_height-.3; %initial position in y

vx=initial_velocity_ground*cos(hit_angle);%Initial velocity in x
vy=initial_velocity_ground*sin(hit_angle);%Initial velocity in y

ax=0;%initial acceleration in x
ay=-1*gravity;%initial acceleration in y


%Set time

t=0;

%Set increments in time desired

delta_t=.005; %in seconds

%Calculate trajectory based on given inputs and conditions

weight_force=-mass*gravity;

spin_ratio_x=0;%create variable spin_ratio_x and assign a value of 0
spin_ratio_y=0;%create variable spin_ratio_y and assign a value of 0
fd_x=0;%create variable Fd_x and assign a value of 0 (drag force)
fd_y=0;%create variable fd_y and assign a value of 0 (drag force)
fl_y=0;%create variable fl_y and assign a value of 0 (lift force)
cd_x=0;%create variable cd_x and assign a value of 0 (drag coefficient)
cd_y=0;%create variable cd_y and assign a value of 0 (drag coefficient)
cl_y=0;%create variable cl_y and assign a value of 0 (drag coefficient)

sx_store=[];
sy_store=[];

vx_store=[];
vy_store=[];

ax_store=[];
ay_store=[];



sx_store=[sx];
sy_store=[sy];

vx_store=[vx];
vy_store=[vy];

ax_store=[ax];
ay_store=[ay];

i=2;
```

```
while(sy>0)


%Determine Forces acting on projectile in x and y directions

%Drag Forces in x

    %Calculate Spin Ratio

    spin_ratio_x=(angular_velocity*diameter)/(2*(vx-wind_speed));

            %Determine Coefficient of Drag

            if(spin_ratio_x<5 && spin_ratio_x>0.05)

            cd_x=-
0.0001474916*(spin_ratio_x)^10+0.0037232759*(spin_ratio_x)^9+-
0.0400229087*(spin_ratio_x)^8+0.2391758192*(spin_ratio_x)^7+-
0.8712539421*(spin_ratio_x)^6+1.9971200562*(spin_ratio_x)^5+-
2.8586264480*(spin_ratio_x)^4+2.3770168518*(spin_ratio_x)^3+-
0.8364561899*(spin_ratio_x)^2+-0.0998227742*spin_ratio_x+0.6286980000;

                    %Determine Drag force in x
            end
                    fd_x=-0.5*air_density*(vx-wind_speed)^2*area*cd_x;
%Drag Forces in y

    %Calculate Spin Ratio in y
    spin_ratio_y=abs((angular_velocity*diameter)/(2*vy));

            %Determine Coefficient of Drag
            if(spin_ratio_y<5 && spin_ratio_y>0.05)

            cd_y=-
0.0001474916*(spin_ratio_y)^10+0.0037232759*(spin_ratio_y)^9+-
0.0400229087*(spin_ratio_y)^8+0.2391758192*(spin_ratio_y)^7+-
0.8712539421*(spin_ratio_y)^6+1.9971200562*(spin_ratio_y)^5+-
2.8586264480*(spin_ratio_y)^4+2.3770168518*(spin_ratio_y)^3+-
0.8364561899*(spin_ratio_y)^2+-0.0998227742*spin_ratio_y+0.6286980000;
            end
                %Determine Drag force in y



                fd_y=-0.5*air_density*(vy)^2*area*cd_y;

                if(vy<0)

                    fd_y=-1*fd_y;

                end

%Lift Forces in y
```

```
  %Calculate lift coefficient in y
  if(spin_ratio_x<5 &&spin_ratio_x>0.05)
  cl_y=-0.000319990769*(spin_ratio_x)^10+0.008584023046*(spin_ratio_x)^9+-
0.098681399148*(spin_ratio_x)^8+0.633281021323*(spin_ratio_x)^7+-
2.472378037439*(spin_ratio_x)^6+5.979932314805*(spin_ratio_x)^5+-
8.665063641190*(spin_ratio_x)^4+6.735554629484*(spin_ratio_x)^3+-
2.071120367*(spin_ratio_x)^2+0.215694805*spin_ratio_x+0.020034532;

  end


  %Calculate lift force in y

  fl_y=0.5*air_density*(vx-wind_speed)^2*area*cl_y;

  %All forces on ball are now known for its current conditions

  %Now determine position, velocity, and acceleration in x and y for this
  %increment in time

  ax=fd_x/mass;
  ay=(fl_y+fd_y+weight_force)/mass;

  %ay=(fd_y+weight_force)/mass;

  %ay=(weight_force)/mass;
  vx=vx+ax*delta_t;
  vy=vy+ay*delta_t;

  sx=sx+vx*delta_t+0.5*ax*(delta_t)^2;
  sy=sy+vy*delta_t+0.5*ay*(delta_t)^2;

 %Position, Velocity, and Acceleration have now been updated for this
 %increment in time


 %Store the new position, velocity, and acceleration for future plotting

sx_store(i)=sx;
sy_store(i)=sy;

vx_store(i)=vx;
vy_store(i)=vy;

ax_store(i)=ax;
ay_store(i)=ay;

i=i+1;

end
```

```
end

hit_angle=hit_angle*180/pi();
hit_angle=num2str(hit_angle);
hit_angle_string=strcat('Hit Angle=',hit_angle,' degrees');

gravity=num2str(gravity);
gravity_string=strcat('Gravitational Acceleration in Y=', gravity,' m/s^2');

angular_velocity=num2str(angular_velocity);
angular_velocity_string=strcat('Angular Velocity=',angular_velocity, '
rad/s');

wind_speed=wind_speed*2.237;
wind_speed=num2str(wind_speed);
wind_speed_string=strcat('Wind Speed=',wind_speed, ' mph');

initial_velocity_ground=initial_velocity_ground*2.237;
initial_velocity_print=num2str(initial_velocity_ground);
initial_velocity_print=strcat('The required minimum initial velocity= ',
initial_velocity_print,' mph');

delta_t_for_print=num2str(delta_t);
delta_t_for_print=strcat('Delta time increments=',delta_t_for_print, '
seconds');

delta_velocity=delta_velocity*2.237;
delta_velocity=num2str(delta_velocity);
delta_velocity=strcat('Delta velocity increments= ',delta_velocity,' mph');

string_for_print=sprintf('%s\n%s\n%s\n%s\n%s\n%s\n%s\n',hit_angle_string,grav
ity_string,angular_velocity_string,wind_speed_string,initial_velocity_print,d
elta_t_for_print,delta_velocity);


sx_store=sx_store*3.281;
sy_store=sy_store*3.281;

figure
plot(sx_store, sy_store)
title('Position in Y vs Position in X')
xlabel('Position in X (ft)')
ylabel('Position in Y (ft)')
xlim([0 hit_distance])
ylim([0 400])
annotation('textbox',[0.3 .5 0.4
0.3],'String',string_for_print,'FitBoxToText','on');

fprintf('The minimum required initial velocity is %.2f mph for the following
input parameters:\n%s\n',initial_velocity_ground,string_for_print)
```