ModSim Exercise 10

Yoonyoung Cho

November 17 2015

1 Key Frames

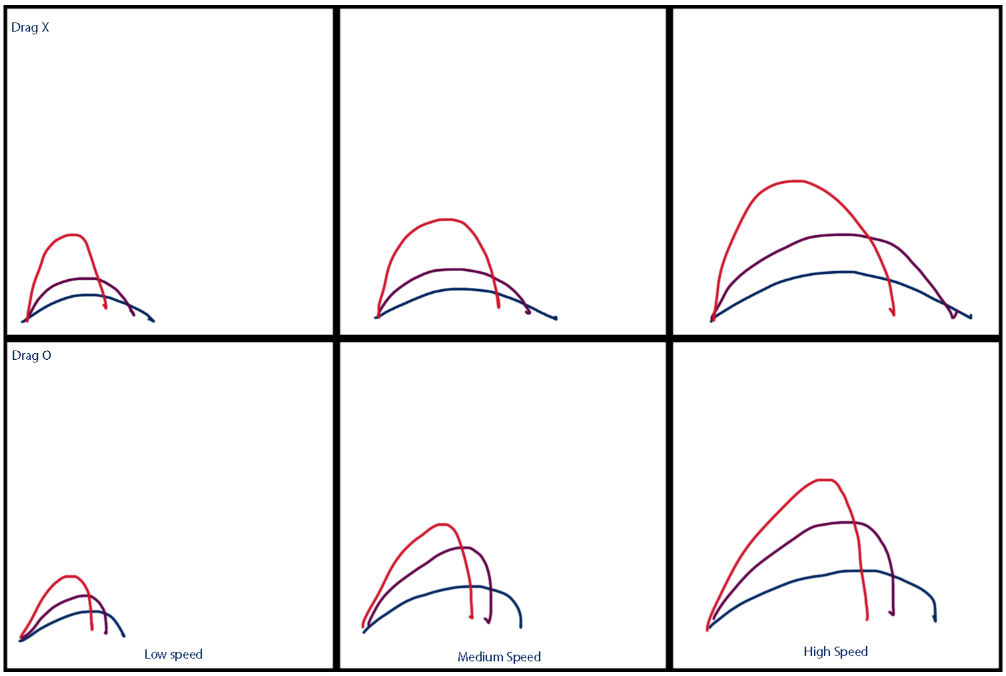


Figure 1: Key Frames of the mechanics involving the flight of the baseball; the graph depicts the trajectory of the baseball. The plot is color-coded based on the launch angle: red, purple, and blue corresponding to high, middle, and low angles, respectively.

In the key frames, I sought to capture the behavior that the distance covered was dependent on the magnitude of the initial velocity, as well as the launch angle; too high of a launch angle would expend much of the momentum in the upwards direction[1]. Moreover, I exaggerated the rightward skewing of the overall trajectory in the presence of drag, as a result of dissipation in the horizontal momentum due to drag.

2 Estimation

Based on the research parameters I had identified in the last exercise(replicated in Table 1), I calculated the initial drag and gravitational force for the starting velocity of 54 m/s

[1]The graph is, of course, faulty: later, I realized that the the angle couldn't be too low either, as the ball would fall to the ground prematurely.

Term	Description	Value	Unit
$\vec{r}$	Initial Position of baseball	$< 0, 1 >$	m
$\vec{v}$	Initial Velocity of baseball	$< 38, 38 >$	m/s
G	Gravitational Constant	$6.67408e - 11$	$m^3 kg^{-1} s^{-2}$
m_b	Mass of Baseball	.145	kg
m_e	Mass of Earth	$5.9742e24$	kg
A	Cross-sectional area of baseball	.004	m^2
$\vec{F_g}$	Gravity	$< 0, -1.42 >$	N
$\vec{F_d}$	Drag Force	$< -0.6, -0.09 >$	N
ρ	Density of Air	1.225	kg/m^3
C_d	Drag Coefficient of Baseball	0.3	-

Table 1: Constants.

The forces that describe the system are:

$$\vec{F} = \vec{F_g}(\vec{r}) + \vec{F_d}(\vec{v})$$

$$\vec{F_g} = -\frac{G m_e m_b}{\vec{r}^2}\hat{\mathbf{r}}$$

$$\vec{F_d} = -1/2 \rho C_d A \vec{v}^2 \hat{\mathbf{v}}$$

The above interaction was implemented in MATLAB as follows:

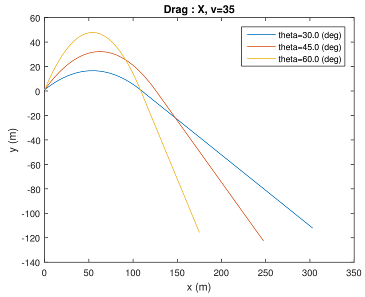
```
function res = Gravity(h_b)%gets height of ball
G = 6.67384e-11; %m^3 / kg*s^@ Gravity Constant
m_e = 5.9742e24; % kg, mass of earth
m_b = 145e-3; % kg, mass of baseball
r_e = 6378e3; %m, radius of earth
r = h_b + r_e; % h + r_e
res = [0 -G*m_e*m_b/r^2];
end

function res = Drag(v_x,v_y)
r_b = 37e-3; %m, radius of ball
rho = 1.225; %kg/m^3, density of air
C_d = 0.3; %drag coefficient of basesball
A = pi*r_b^2;
speed = sqrt(v_x^2+v_y^2);

nv_x = v_x / speed; %normalize
nv_y = v_y / speed;
res = -1/2*rho*C_d*A*speed^2*[nv_x nv_y];
end
```

which yielded Drag $= 2.304455N$ and Gravity $= 1.421198N$[2]. Based on this comparison, it was clear that Drag force could not be disregarded from the calculation.

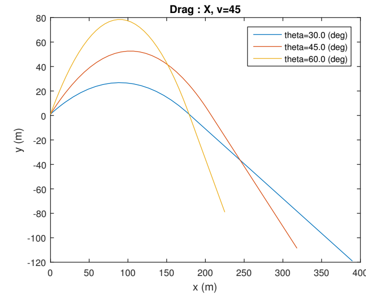
3 Basic Implementation

To avoid redundancies, the final code will be presented in the next section. Here, only the results of the basic implemention – without the later added constraints – will be shown:

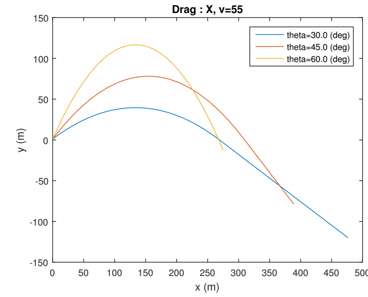
[2]since it is known that the height of the ball does not influence the magnitude of gravity very much, I simply chose 0 as the height in the above code.



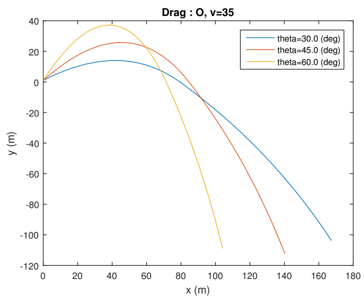
(a) 35m/s without Drag



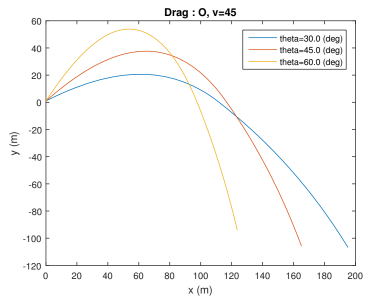
(b) 45m/s without Drag



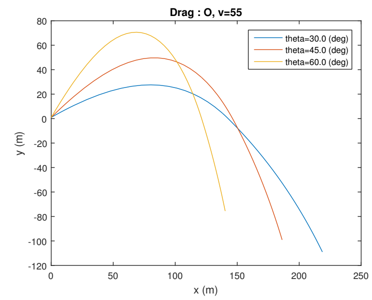
(c) 55m/s without Drag



(d) 35m/s with Drag



(e) 45m/s with Drag



(f) 55m/s with Drag

Figure 2: Matlab-generated keyframes

It can be seen that without the constraints that limit the ball above the ground (quite naturally) the ball goes farthest when shot in a lower angle, in a more horizontal orientation. As this does not accurately reflect what happens – and we are not currently equipped with the capacity to model the bouncing of the ball, a restriction will be enforced that limits the ball's location to above the ground, which leads to the next section:

4 Event Detection

The MATLAB implementation of the simulation is as follows:

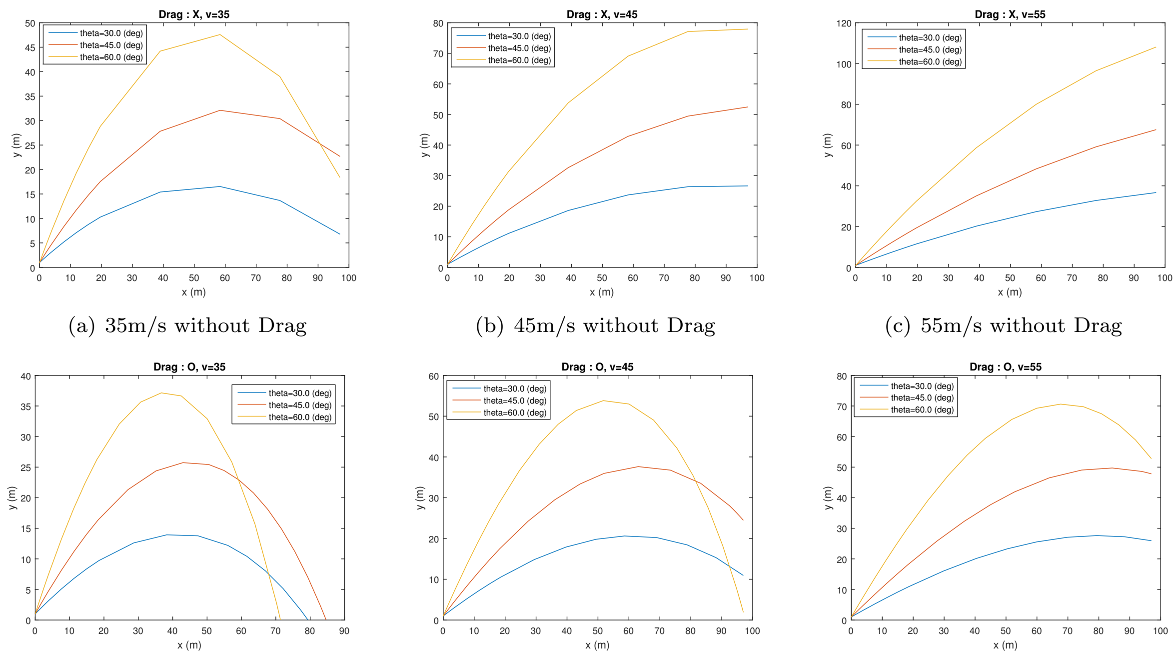
```
function res = simulateBaseball(speed,angle,useDrag,plotRes)
r_x =0;
r_y =1;%initial positions, m

v_x =speed*cos(angle);
v_y =speed*sin(angle);%initial velocity, m/s

y = [r_x r_y v_x v_y];
y = y';
opt = odeset('Events',@hitGround); %When it hits ground (r_y <=0)
netAccelDrag = @(t,y) netAccel(t,y,useDrag);
[t,Y] = ode45(netAccelDrag, [0 10],y, opt);

%PART 3
if(plotRes)
        plot(Y(:,1),Y(:,2));
        xlabel('x (m)');
        ylabel('y (m)');
end
```

```
res_raw = Y(:,2);
res = res_raw(end);
end
function res = netAccel(~,y,useDrag)
m_b = 145e-3; % kg, mass of baseball

dv = netForce(y,useDrag)/m_b;
res(1) = y(3); %dx
res(2) = y(4); %dy
res(3) = dv(1); %dvx
res(4) = dv(2); %dvy
res = res';
end

function res = netForce(y,useDrag) %returns [f_x f_y]
r_x = y(1);
r_y = y(2);
v_x = y(3);
v_y = y(4);
if(useDrag)
        res = Gravity(r_y) + Drag(v_x,v_y);
else
        res = Gravity(r_y);
end
        if(r_y <= 0)
                res(2) = 0; %downward force = 0
        end

end

function res = Gravity(h_b)%gets height of ball
G = 6.67384e-11; %m^3 / kg*s^@ Gravity Constant
m_e = 5.9742e24; % kg, mass of earth
m_b = 145e-3; % kg, mass of baseball
r_e = 6378e3; %m, radius of earth
r = h_b + r_e; % h + r_e
res = [0 -G*m_e*m_b/r^2];
end

function res = Drag(v_x,v_y)%force acting against velocity
r_b = 37e-3; %m, radius of ball
rho = 1.225; %kg/m^3, density of air
C_d = 0.3; %drag coefficient of basesball
A = pi*r_b^2;
speed = sqrt(v_x^2+v_y^2);

nv_x = v_x / speed; %normalize
nv_y = v_y / speed;
res = -1/2*rho*C_d*A*speed^2*[nv_x nv_y];
end

function [val,term,dir] = hitGround(t,y) %Event Condition to stop
%named hitground but also accomodates hitting the wall
val(1) = y(2); %y(2) = r_y = height
```

```
term(1) = 0;
dir(1) = 0;

val(2) = y(1) - 97; %second event : hit wall
term(2) = 0;
dir(2) = 0;
end
```

The function takes initial speed and angle as a parameter, as well as flags such as a whether or not to incorportate drag force into the system, and whether or not to plot the result.

and the resultant plot:



(a) 35m/s without Drag (b) 45m/s without Drag (c) 55m/s without Drag

(d) 35m/s with Drag (e) 45m/s with Drag (f) 55m/s with Drag

Figure 3: Matlab-generated keyframes, with events.

It is seen that, with enforced constraints, an angle neither too high nor too low performs best, as it has enough time in air to travel, as well enough horizontal velocity to travel far; there is evidence of the rightward skewing – as well as greatly reduced distance – in the presence of the drag force, though a bit more subtle than that hand-drawn in the keyframes.

v (m/s) \ θ(deg)	30	45	60
35	0	0	0
45	10.93	24.45	1.89
55	25.94	47.80	52.76

Table 2: Final Height of the ball based on the initial launch state.

5 Range-based Analysis

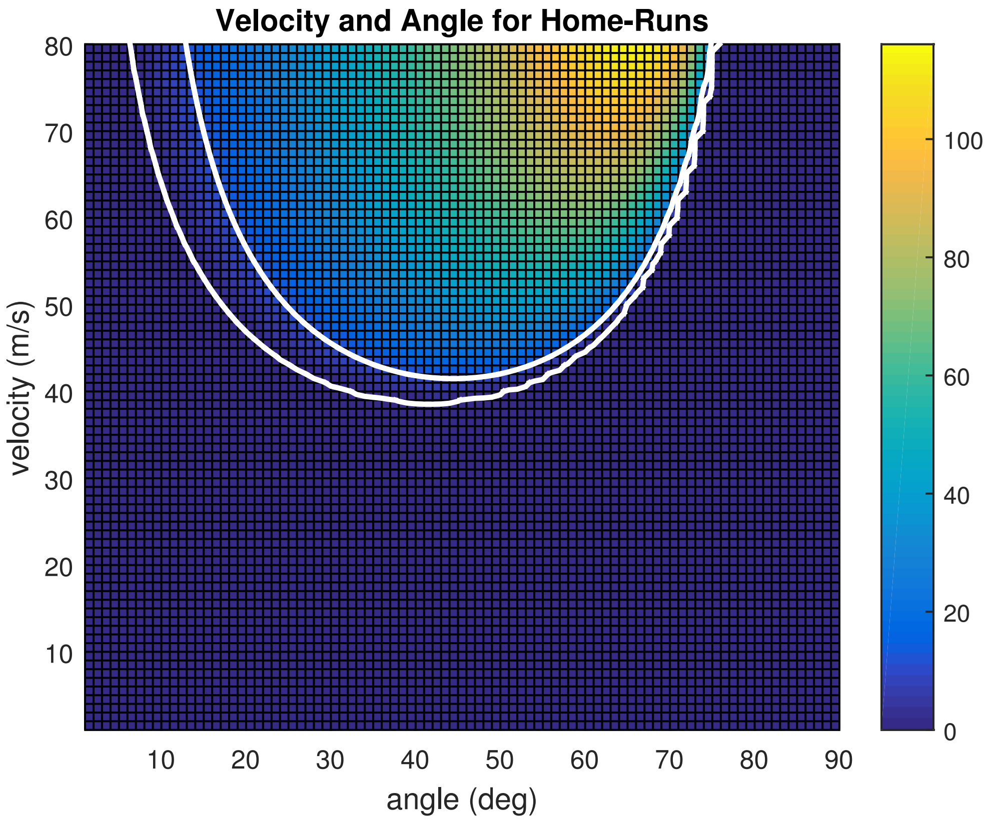


Figure 4: The final height of the ball, given velocity and angle as initial states.

From the graph, it is apparent that the minimum velocity to clear the green monster is around $40m/s$, at approximately 40 degrees above the horizontal.