THE COLLEGE OF WOOSTER

How to Be a Good Shooter: Analysis of Friction-involved Projectile Motion

Math Modeling Final Project

Author:

James Bai

December 11, 2016

Abstract

This study investigates the projectile motion associated with friction due to air resistance and wind. The paper will firstly derive a 2D projectile motion model as system of second order differential equations, based on Newton's second law. Next, a 3D model consisting of three second order differential equations is introduced with similar ideas. First of all, according to the built 2D model, some experimental data of projectile motion is used to compute the friction coefficient of two different balls. Next, the found coefficient value is substituted into the 2D model again, to find the optimal initial velocity and the optimal launch angle in terms of certain distance and height. In the end, the found coefficient, initial velocity and launch angle are substituted into the 3D model, for people to visualize the trajectory of an object during the projectile motion in the real world.

Contents

List of Figures

Chapter 1

How to Be a Good Shooter: Analysis of Friction-involved Projectile Motion

1.1 Introduction

When people shoot an object to a target, the object is expected to hit the target straight forward for accuracy. However, the object will fall because of the effects of gravity and friction due to air resistance and wind during the motion. This movement is described as projectile motion, which is the motion compounded of one which is uniform and horizontal and of another which is vertical and naturally accelerated without taking friction as a part.[5] To hit the target, people actually expect the object to have a curve motion rather than going straight forward. Like shooting a basketball, the basket is higher than basketball players, so players need to make a beautiful curve to get the ball in. Moreover, during the projectile motion, the object is affected by friction including air resistance and wind in the real world. Therefore, the trajectory of an object during projectile motion is different between the ideal situation and the real-world situation.

To better understand the projectile motion in the real world, the general factors that affect the trajectory of an object are defined and declared in this paper as below:

• The launch angle, which is the angle between the direction the object is launched and the horizontal ground, and is denoted as θ in figure 1.1 shown below.

• The initial velocity of the object given by people, represented as v_0 in figure 1.1.

• The wind angle α, which indicates the angle between the direction of the wind and the horizontal ground, and wind velocity u.

• The friction f which always has the direction tangent to the trajectory of the object.

• The weight of the object itself w.

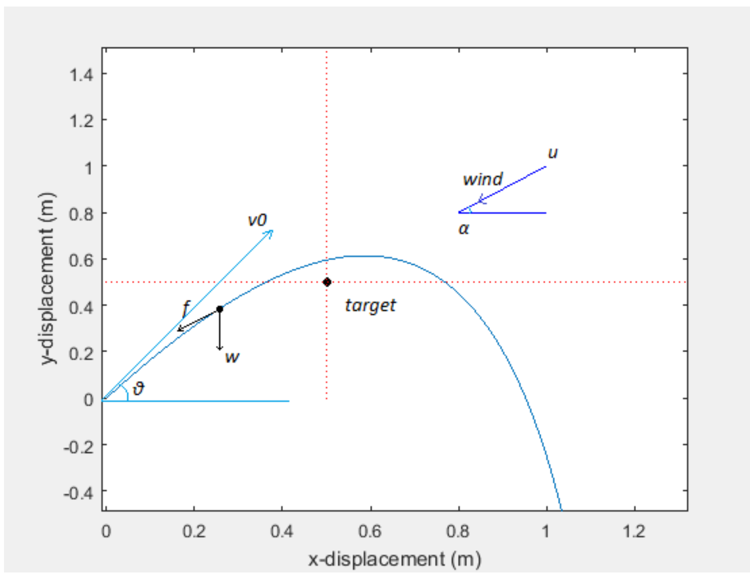

Figure 1.1: A Sample Graph of Projectile Motion Trajectory

Apparently, for a projectile motion of an object, the launch angle, initial velocity, wind angle, wind velocity and weight of the object could all be easily measured except the friction f. To investigate how friction is involved in the projectile motion, this paper will introduce a general 2D model and 3D model, then compute the friction coefficient through 2D model from experimental data, and find the optimal launch angle and initial velocity based on certain distance and height, finally substitute all the values into 3D model to visually analyze the projectile motion.

1.2 Assumptions

Before introducing the model, the immeasurable friction can be described as

$$f = k \cdot V^n,$$

where k is friction coefficient and V is compounded velocity of the object at certain time. Since it is unclear what the order n should be in the real world, our first assumption is that:

• The friction is linear to the compounded velocity of the object. In other words, $n = 1$.

Secondly, one can image that the larger surface one object has, the larger the friction will be during the motion. Therefore,

• The friction coefficient k is proportional to the surface of the object.

This also indicates that k is different corresponding to different objects based on their shapes.

1.3 Model

1.3.1 Newton's Second Law

From figure 1.1 in the first section, one can see that the object is affected by two forces, weight and friction. To relate the behavior of an object with force, Newton's Second Law is utilized. Newton's Second Law states that the net force on an object is the product of the object's mass and acceleration,

$$F = m \cdot a,$$

and we will build our model based on this formula.

1.3.2 2D Model

Based on figure 1.2, a net force analysis from figure 1.1, the friction can be disintegrated along both x direction and y direction as f_x and f_y. By standardizing leftwards and downwards as negative, the horizontal net force $F_x = -f_x$ and the vertical net force $F_y = -f_y - w$. Moreover, according to Newton's Second Law, $F_x = m \cdot a_x$ and $F_y = m \cdot a_y$, and the general projectile 2D model can be constructed as:

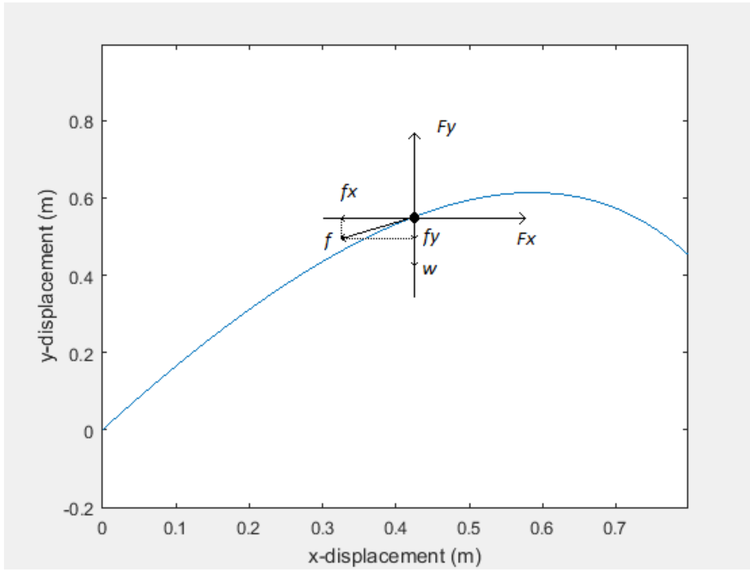

Figure 1.2: Net Force Analysis Graph of Projectile Motion Trajectory

$$\begin{cases} x : F_x = -f_x = m \cdot a_x = -k \cdot (V_x)^n, \\ y : F_y = -f_y - w = m \cdot a_y = -k \cdot (V_y)^n - mg, \end{cases}$$

where $w = mg$ by Newton's Second Law, m is the mass of the object and g is the natural acceleration, gravity.

What is more, the velocity is known as the derivative of displacement over time t, and the acceleration is the derivative of the velocity. Thus, acceleration is the second derivative of displacement. Then, the model becomes:

$$\begin{cases} x : F_x = m \cdot a_x = m \cdot x'' = -k \cdot (V_x)^n, \\ y : F_y = m \cdot a_y = m \cdot y'' = -k \cdot (V_y)^n - mg. \end{cases}$$

As V_x is the compounded horizontal velocity, V_x should be the velocity that abstracts the horizontal wind velocity from the object's horizontal velocity, which should be $V_x = v_x - u \cdot cos(\alpha)$. Similarly, $V_y = v_y - u \cdot sin(\alpha)$, where v_x and v_y represent for the horizontal and vertical velocity just of the object itself. As $n = 1$, the model becomes

$$\begin{cases} x : x'' = -\frac{k}{m} \cdot (v_x - u \cdot cos(\alpha)), \\ y : y'' = -\frac{k}{m} \cdot (v_y - u \cdot sin(\alpha)) - g. \end{cases}$$

Since *MatLab*, the mathematical programing application that I am using, cannot solve second differential equations, both second order differential equations from the model are converted into pairs of first order differential equations as:

$$\begin{cases} x : x' = v_x, v_x' = -\frac{k}{m} \cdot (v_x - u \cdot cos(\alpha)), \\ y : y' = v_y, v_y' = -\frac{k}{m} \cdot (v_y - u \cdot sin(\alpha)) - g. \end{cases}$$

The initial conditions will be $v_x(0) = v_0 \cdot cos(\theta)$ and $v_y(0) = v_0 \cdot sin(\theta)$.

1.3.3 3D Model

To be more realistic, a 3D model is built as well for visualization later on. As one more dimension is included, some declarations have changed shown as figure 1.3:

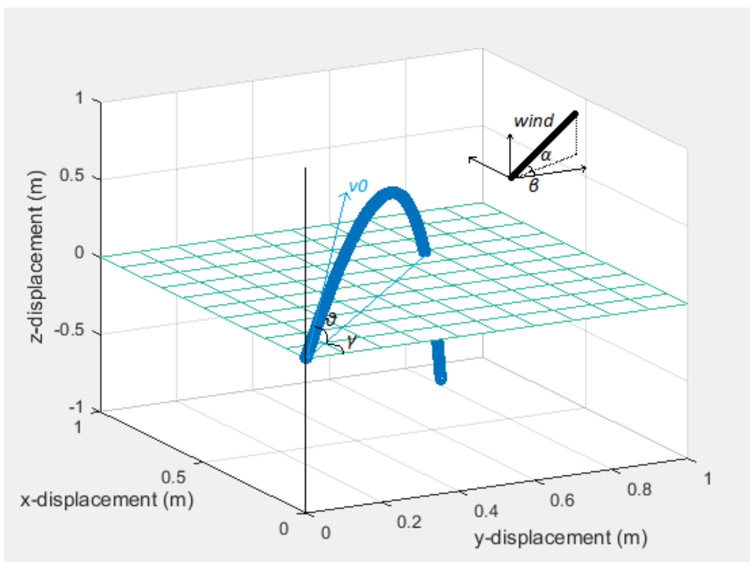

Figure 1.3: A 3D Sample Graph of Projectile Motion Trajectory

• launch angle is a pair of angles where one is the angle between the launch direction to the horizontal ground (xy plane), and the other one is the angle between the launch direction to the vertical side along $y - axis$. The pair is denoted as θ and γ.

• The wind angle is another pair of angles where one is the angle between the direction of the wind and the horizontal ground (yz plane), and the other one is the angle between the direction of the wind to the vertical side along the $y - axis$. The pair is denoted as α and β.

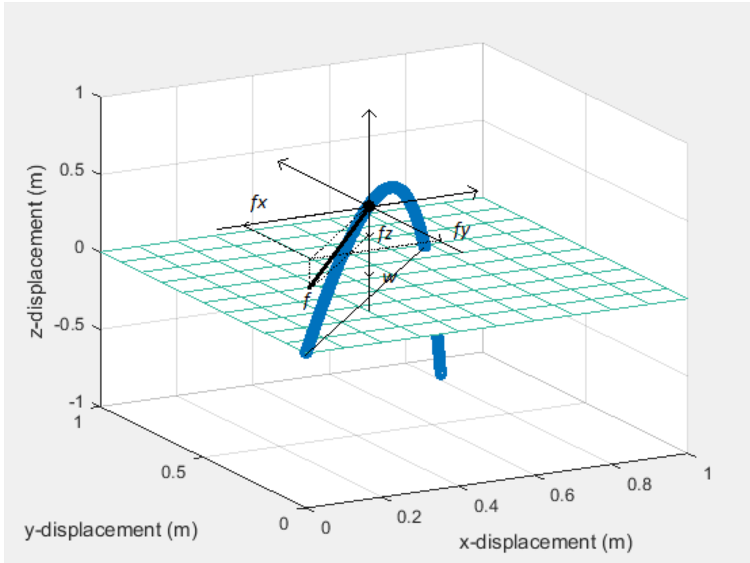

Figure 1.4: 3D Net Force Analysis of Projectile Motion Trajectory

In this case, the model is still firstly applied by Newton's second law. From figure 1.4, one can see that the object is only affected by frictions through the $x - axis$ and $y - axis$, and is affected by friction and weight through $z - axis$. Using the same idea mentioned in the 2D model, similarly, the 3D model is constructed as

$$\begin{cases} x : x' = v_x, v_x' = -\frac{k}{m} \cdot (v_x - u_x), \\ y : y' = v_y, v_y' = -\frac{k}{m} \cdot (v_y - u_y), \\ z : z' = v_z, v_z' = -\frac{k}{m} \cdot (v_z - u_z) - g. \end{cases}$$

To derive the initial conditions, one can regard the initial velocity as a vector shown in figure 1.4, and track its projection onto xy plane. Since the launch angle between the direction and the yz plane is γ, the initial conditions are

$$v_x = v_0 \cdot cos(\theta) \cdot cos(\gamma), v_y = v_0 \cdot cos(\theta) \cdot sin(\gamma), v_z = v_0 \cdot sin(\theta).$$

Similarly, regarding the wind velocity as another vector from figure 1.3, and as wind angles are α and β, the wind velocities along each axis are

$$u_x = u_0 \cdot cos(\alpha) \cdot cos(\beta), u_y = u_0 \cdot cos(\alpha) \cdot sin(\beta), v_z = v_0 \cdot sin(\alpha).$$

Therefore, the final 3D model is

$$
\begin{cases}
x : x' = v_x, v_x' = -\frac{k}{m} \cdot [v_x - u_0 \cdot cos(\alpha) \cdot cos(\beta)], \\
y : y' = v_y, v_y' = -\frac{k}{m} \cdot [v_y - u_0 \cdot cos(\alpha) \cdot sin(\beta)], \\
z : z' = v_z, v_z' = -\frac{k}{m} \cdot [v_z - v_0 \cdot sin(\alpha)] - g.
\end{cases}
$$

1.4 Computation, Methods and Analysis

Recall the main task is to find the constant friction coefficient k, and to find the optimal launch angle θ_0 and optimal initial velocity v_0.

The data is obviously needed for computation, and is collected from *Physics 111 Projectile Motion Lab*. In that lab, students are asked to launch two different balls by a launch machine on a table, and the ball will fall as a projectile motion when it goes to the edge of the table, described as figure 1.5.

		order	theta_0	mass	Radius	A			B			C		
						v0	x_real	y_real	v0	x_real	y_real	v0	x_real	y_real
ball 1	run1	1	0	0.004		2.95	0.906	0.809	2.85	1.151	1.2	2.87	1.065	0.92
						2.91	0.894		2.77	1.117		2.94	1.08	
						3.15	0.911		2.81	1.134		2.88	1.066	
						2.84	0.905		2.8	1.1		2.9	1.06	
						3.32	0.985		2.77	1.12		2.83	1.035	
	avg					3.034	0.9202		2.8	1.1244		2.884	1.0612	
	run2	1	0	0.004		3.87	1.07	0.809	2.72	1.061	1.2	3.28	0.588	0.46
						3.91	1.06		2.8	1.055		3.28	0.588	
						3.71	1.09		2.78	1.065		3.26	0.587	
						3.75	1.07		2.81	1.078		3.32	0.592	
						4.31	1.04		2.8	1.088		3.3	0.591	
	avg					3.91	1.066		2.782	1.0694		3.288	0.5892	
ball 2	run3	1	0	0.005		5.36	1.01	0.809	4.3	1.146	1.2	2.97	1.068	0.92
						4.69	1.02		4.29	1.138		2.99	1.077	
						5.77	1.02		4.25	1.161		2.93	1.048	
						4.87	1.01		4.3	1.15		2.97	1.092	
						5.07	1.02		4.31	1.149		2.92	1.057	
	avg					5.152	1.016		4.29	1.1488		2.956	1.0684	
	run4	1	0	0.005		4.26	1.01	0.809	4.25	1.134	1.2	3.24	0.581	0.46
						4.52	1.02		4.25	1.141		3.28	0.59	
						4.21	1.02		4.27	1.149		3.3	0.577	
						4.57	1.01		4.28	1.14		3.26	0.582	
						4.36	1.02		4.26	1.144		3.28	0.594	
	avg					4.384	1.016		4.262	1.1416		3.272	0.5848	

Figure 1.5: Graph of Projectile Motion Lab

The velocity of the ball at the edge of the table is recorded by a specific tool, the x and y displacements are measured. By collecting and adjusting the data from three students, the data management is firstly shown in the coming section. Then the methods to find k, θ_0 and v_0 are introduced.

1.4.1 Data Management

As one can see in figure 1.6, there are three students' lab data managed. Each student operates four runs in the lab. They use one same ball for the first two runs, and another

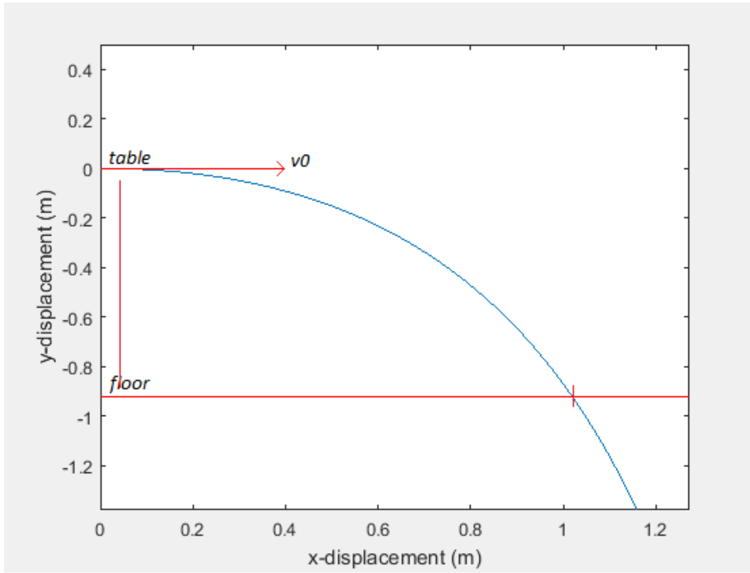

Figure 1.6: Lab Data Collected from Three Students

same ball for the last two runs. Based on the first assumption, order n is 1. Since the ball is launched on a horizontal table, it indicates that the launch angle θ_0 is 0, and the radius and mass of both balls are recorded as well. In each student's each run, I am also able to get a set of values of averaged initial velocity of the ball, the averaged x - displacement the ball reaches when it hits on the ground and actual y - displacement the ball falls. Since the lab is indoors, the wind velocity u is 0 and g is $9.8m/s$ from common sense.

1.4.2 Finding Friction Coefficient k

To find k, the *fminsearch* function from *MatLab* is utilized. By substituting all the variables mentioned above into the 2D model, the program will generate a set of simulated x coordinates and y coordinates. In terms of the simulated x coordinates and y coordinates, a simulated trajectory could be plotted in *MatLab*.

Assuming $k = 0.01$ initially, calling *fminseach* function then helps to keep tracking k till the simulated trajectory gets closest to the real x - displacement and y - displacement by using distance formula

$$\text{distance} = \sqrt{(\delta x)^2 + (\delta y)^2}.$$

The inner idea of *fminsearch* function is shown as figure 1.7 below.

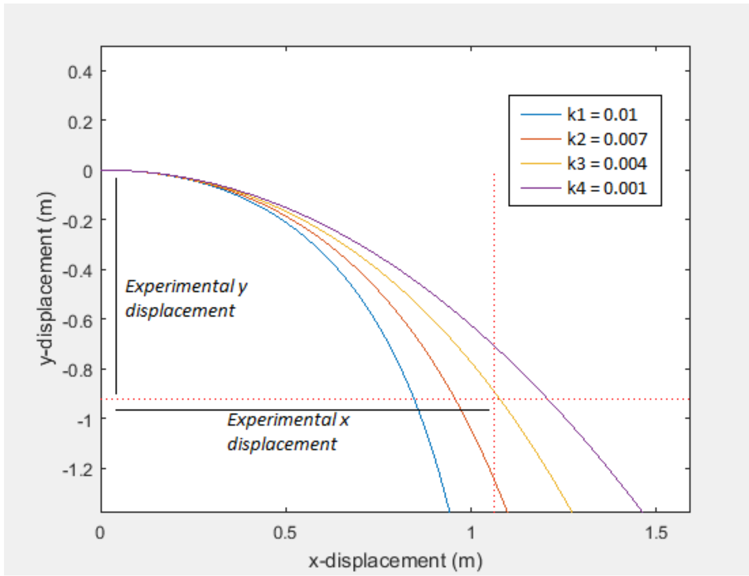

Figure 1.7: A Graph Showing the Inner Idea of *fminsearch* Function

By manually changing the values in the program, one optimized k is given in each run. By computing the average, the final drag friction coefficient is given in figure 1.8. For ball with radius of !!!, k is !!!. For the ball with radius of !!!, k is !!!. As the second k is larger than the first k, it also proves the second assumption. By looking at the standard deviation of the optimal $k's$ for both balls, a measure that is used to quantify the amount of variation or dispersion of a set of data values.[] The standard deviation is !!! for the first ball, and !!! for the second ball. As both values are really closest to 0, one can say that both sets of optimal $k's$ are not too spread, so the calculated averaged optimal $k's$ are pretty realistic.

Figure 1.8: Computed Optimal k for each Run and Error

1.4.3 Finding Optimal Launch Angle and Initial Velocity

Similarly, after getting the value of k, optimal lunch angle and initial velocity can be found by substituting the k value into the 2D model, based on certain distance and height. Suppose people want to find the optimal θ_0 and v_0 for shooting the first ball to go through a hoop, where the hoop is $0.5m$ far away and $0.5m$ high away from the launch position, described as figure 1.9 above.

The launch angle θ_0 is assumed as $45°$ and initial velocity v_0 is assumed as $5m/s$ initially, which are pretty realistic assumptions in this case. By setting up θ_0 and v_0 as a vector in *MatLab*, the *fminsearch* function is called again to keep tracking the vector as a whole till the simulated trajectory gets closest to the position of the hoop in figure 1.9.

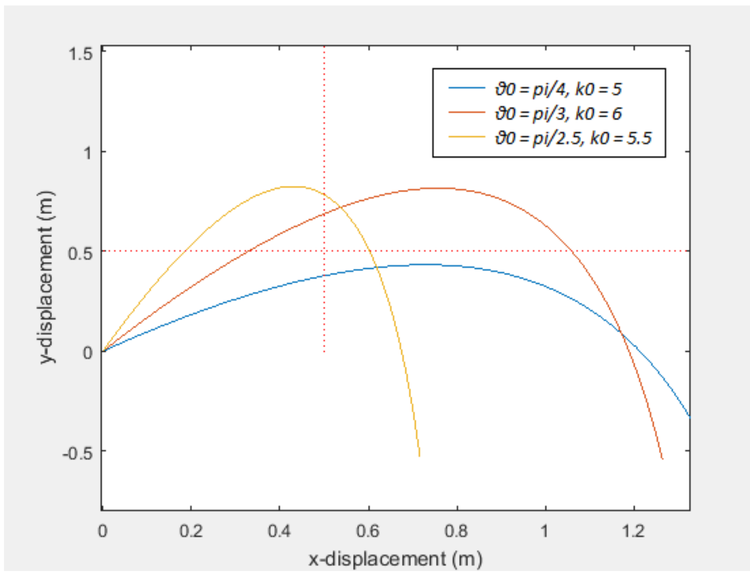

Figure 1.9: A Sample Graph of Shooting the First Ball from the Lab to Go Through A Hoop

The optimal launch angle is found as !!!, and the optimal initial velocity is found as !!!.

1.5 Results - Visualization in 3D Model

Apparently, it is hard for people to visualize how the object moves in a 2D model, thus the 3D model is utilized for plotting a more visualizing graph for people to understand the friction-involved projectile motion.

In figure 1.10 above, the exact same motion from figure 1.9 is converted from a 2D view into a 3D view, which makes the visualization more realistic. What is more, people are actually able to investigate how the trajectory will change by taking wind into account.

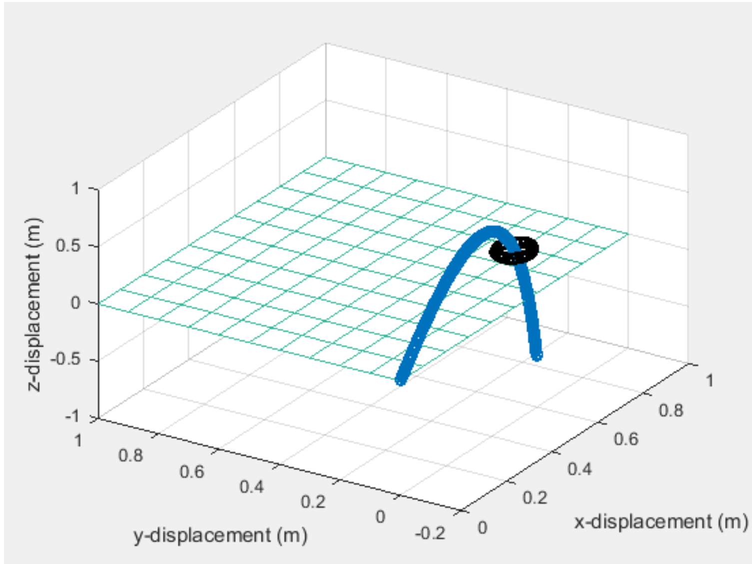

Figure 1.10: A 3D Sample Graph of Shooting the First Ball from the Lab to Go Through a Hoop

One special case would be that the wind is blowing parallel to the y - axis with the same velocity as the initial velocity, the actual trajectory in the real world is shown as figure 1.11 below.

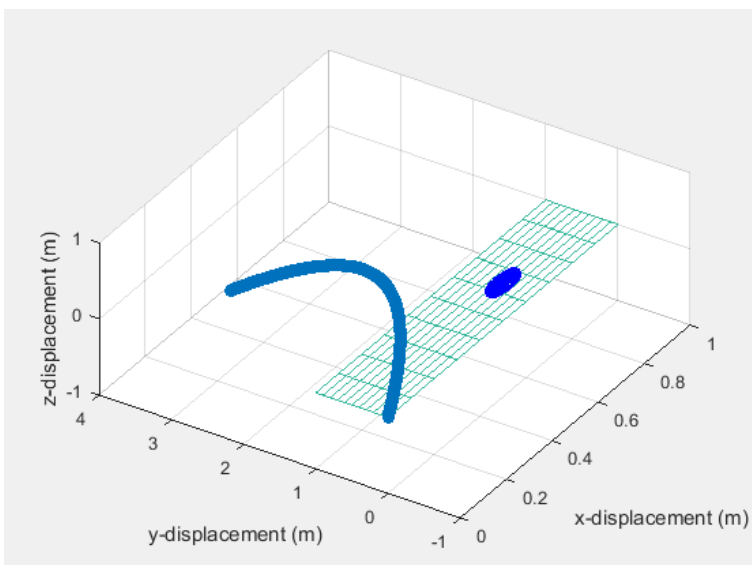

Figure 1.11: A 3D Graph of Projectile Motion Trajectory in a Special Case

In this process, one can see that the 3D model could be a strong tool to simulate an actual projectile motion of any object in the real world by setting up the initial values for certain variables.

1.6 Conclusion

In general, a comprehensive 2D and 3D projectile motion model with friction involved are successfully built up through *MatLab* in this paper, which is much more accurate than the original ideal model.

By assuming the friction is linear to the object's velocity and utilizing experimental data, the coefficient k is found as !!! for the first ball in the physics lab, and !!! for the second ball. In the scenario of shooting the ball going through the hoop at $(0.5, 0.5)$ to the launch position set as $(0, 0)$, the θ_0 is found as !!! and v_0 is found as !!! for the first ball, and the θ_0 is found as !!! and v_0 is found as !!! for the second ball.

Nevertheless, the methods described in this paper does have weakness. The first weakness is that the first assumption hasn't proved yet, it is still not clear if the friction is linear to the object's velocity or not, which will affect the accuracy of the value of k computed. The second weakness is that, to compute the friction coefficient k of a certain object, lots of data are required, and it is hard to keep all the data accurate, which affects the accuracy as well.

However, as long as the data is available, the model and methods stated in the paper are able to compute a relatively k. By substituting the found k value into the 2D model again, the model is able to find any θ_0 and v_0 by setting the hoop as any coordinate, for shooting any object corresponding to any certain order. People are also able to visualize how the object moves exactly in the real world in a 3D view.

Bibliography

[1] AHMAD, BASHIR, H. B. J. J. N. . O.-Z., AND SHAMMAKH, W. Projectile motion via riemann-liouville calculus. *Advances in Difference Equations 1* (2015), Web.

[2] ANDERSEN, P. W. Comment on wind-influenced projectile motion. *European Journal of Physics Eur. J. Phys 36 6* (2015), 068003.

[3] BERNARDO, REGINALD CHRISTIAN, J. P. E. J. D. V., AND CANDA, J. J. Wind-influenced projectile motion. *European Journal of Physics 36 2* (2015), 025016.

[4] KRON, G. A. *Advanced Dynamic-system Simulation: Model - replication and Monte Carlo Studies*. New Jersay, 2013.

[5] KUEHN, K. *A Students Guide through the Great Physics Texts*. Springer, New York, 2015.