**Frisbees**

Frisbees are flying objects used to entertain people. As it does not require a lot of abilities to throw them, the concept behind them seemed very simple. In contrary to this assumption, they are complex since they operate under two physical concepts, aerodynamic lift and gyroscopic stability[[1]](#footnote-1). Looking at Bernoulli’s principle and the lift force which causes an object to stay in the air, the Frisbee could be compared to a wing. Gyroscopic stability is responsible to keep the Frisbee straight. Other forces implied with the stability such as the angular momentum are exerted, and they all play a role in preventing it from flipping over during its flight period. In this project, our focus was on what is the angle of attack to throw a Frisbee in order to reach a maximum flight distance. Using some different methods and equations, all the variables will be optimized to get the best possible answer. Our hypothesis was that the optimal angle of throw should be between 10 and 15 degrees since it would maximize the lift force which allows the frisbee to fly and minimize the drag force which slows it down.

**Model**

Since there are a lot of variables to consider (Figure 1), we will use many different equations. First, to be able to find the angle of attack which allows us to reach the greatest distance, we need to separate the problem into four phases.

Figure 1

First, we need to calculate the drag coefficient and force generated at a certain angle as well as the same variables for the lift. To find the drag force—a force parallel to the velocity, but that acts in opposite direction—we used the following formula:

(1)

where represents the density of the fluid, *A* represents the Frisbee’s area and *v* is the velocity of the frisbee at this point, but we will first need to calculate a drag coefficient given by the equation

. (2)

In this equation, Cd0 is the coefficient of drag at angle zero and this defined to 0.08, is 2.72 and is the drag coefficient depending on alpha, and is a constant defined to be -4 degrees and represents the angle of throw which minimize the lift force on the frisbee.

Same thing with the lift force which is perpendicular to the drag force. The lift coefficient *Cl* will need to be calculated with:

(3)

In this case, is the lift coefficient when the angle of attack is 0 degrees and was found to be 0.1 and is 1.4 and represents the lift coefficient dependent on alpha.

And the force generated by the lift corresponds to this equation:

(4)

The Frisbee’s area, the coefficient of lift, the density of the fluid which in this case will be the air, and the velocity of the frisbee at this moment are needed to calculate the lift force . The initial velocity of a Frisbee throw is approximatively 14 m/s. The standard air density we use in this case is 1.23 kg/m3, the Frisbee’s area is of 0.0531m2 and as we might need it, the viscosity of air used as found to be 1.73x10-5N s/m2. The values of the constants we use were found in the article we based our project on and is referred in the work cited. It is also important that the variables calculated in this step will be updated as the angle of the frisbee changes over time.

Then, we will calculate the acceleration of the object as it travels. To do so, we will decompose it into two components (x and y) because it is a two-dimensional problem which are calculated with

(5)

(6)

Where is the angle between the velocity vector and the x-axis, m is the mass of the frisbee, which we found to be 0.175 kg, and g being the gravity. We assumed the initial angle of the velocity vector to be the same as the angle of attack. However, as time goes on, it will be updated by using its x and y components. Since the acceleration varies depending on the point at which frisbee is in its trajectory, these formulas will only hold true for small period of time and will be updated with the new conditions to establish the acceleration throughout the displacement.

The third part of the problem is to make sure we get the most accurate angle of throw. To do so, the Golden Search method will be used to optimize the results to find the best angle of attack by taking in the distance reached with the angles it provided.

Lastly, to measure the distance reached by the frisbee at the angle found with the Golden Search method, we used Euler’s method to update the angle of the frisbee, its acceleration, velocity, as well as distance over time. By incrementing the conditions based on the previous ones, we were able to maximize the distance reached.

**Computational Methods**

The Frisbee’s components in our main method were used in some numerical methods described in the previous section. We either used some conventional constants such as the air density at sea level and the gravitational acceleration or we used constants like the frisbee’s area and mass, the minimum drag due to friction, the induced drag, the angle of attack that produces the less lift, and the initial velocity was that of an average throw. The values of the constants we used were found in the articles we based our project on.

The boundaries we used to calculate the optimal angle with the Golden Search method where based on the fact that we know the maximum (90o) and minimum (0o) angles at which we can throw a Frisbee, but we narrowed down the range of angle to between 0 and 30. We made this assumption based on the fact that from an angle greater that 30, the frisbee will just go higher up while reaching a smaller distance. Moreover, our program did not support angles greater than 30 so we simplified it. Since this method is usually used to find a minimum value, we had to use the inverse of our distance function because in this case, we want to find a maximum value. The Golden Search is also useful because it is a unimodal function, meaning there is a single optimal value.

In one of our methods, we have also calculated the distance travelled by the frisbee and we used data such as the velocity, the x and y initial positions, the angle and we have created arrays. In the same method, we have also used Euler’s method to find the x and y positions, the acceleration, the velocity and the angle at this specific point of the flight. Since all of these values are always changing, this method is also used to update them each time the program is run. To validate our program, we will compare our answers with graphs that have already been made. X and Y values will be printed after running our code and we will then plot a height (m) vs distance (m) graph with our optimal angle of attack.

**Results**

After running our program, we printed the x and y positions for the optimal angle and we transformed into an excel graph to illustrate the trajectory of the frisbee. Since the optimization part of our program was not working very well and simply went to infinity when the angle of attack was greater than 30, we used the angle found in the article which is 12 degrees to do the validation. Our results are therefore based on an initial vertical position of approximatively 1m and an initial horizontal position of 0m.

After looking at the data and the graph, a final horizontal distance of 29.985609 m was observed and a maximum height of 10.12517506 m (Figure 2) was also found. In *The Physics of Frisbees*, a vertical distance of 7.7 m was found while reaching 40 meters horizontally. This means that the percent error for our distance was of about 25 % and of 29% for the height relative to their findings.

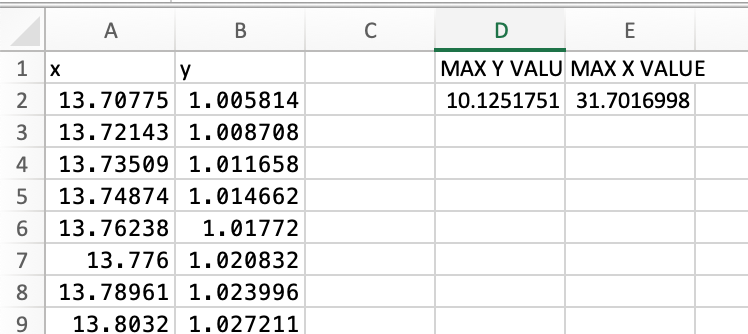
****

Figure 2

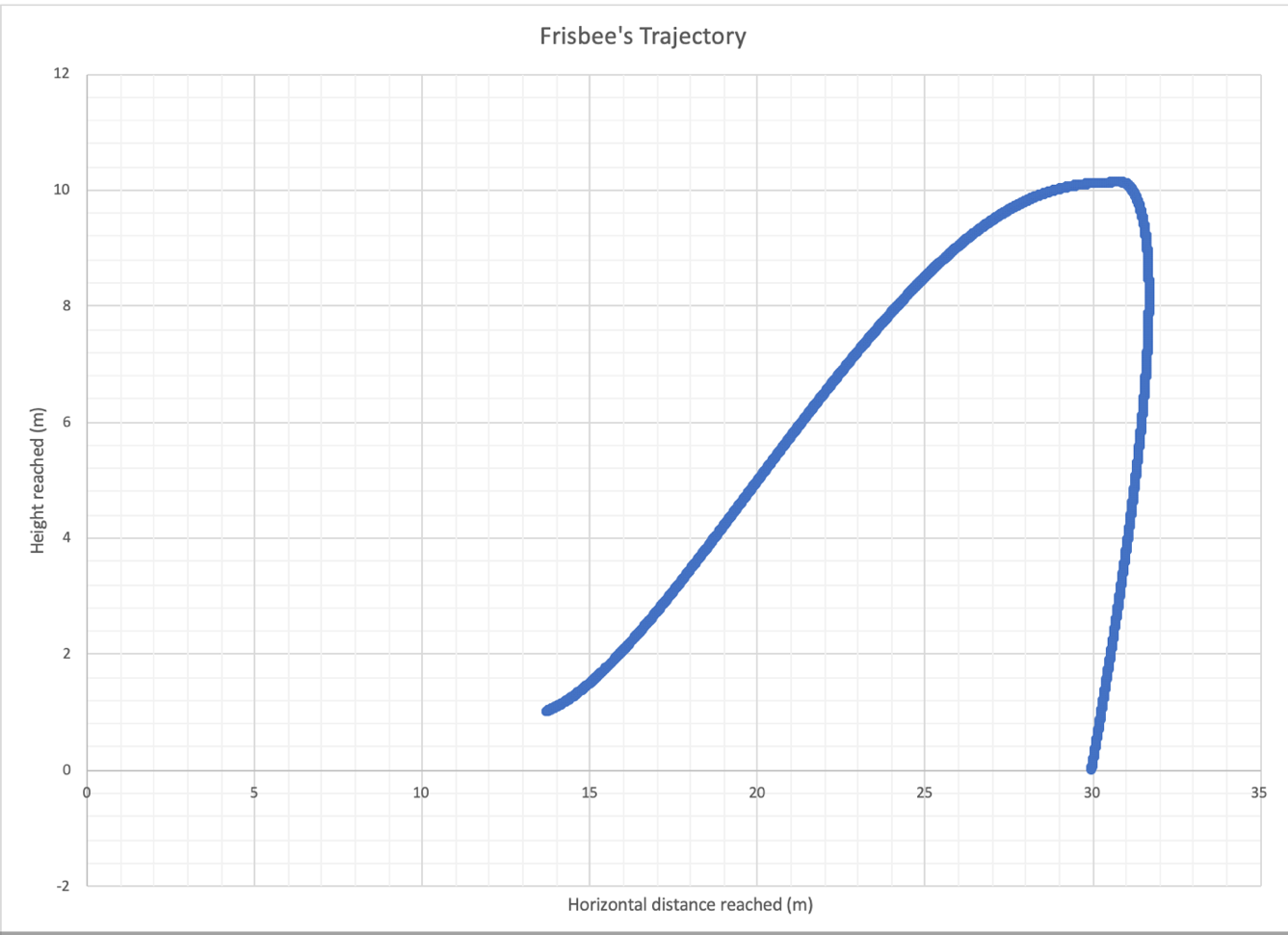


Figure 3

**Discussion**

The discrepancy between our results and the article’s suggests that there was a problem with our program which were not able to fix. However, our hypothesis which stated that the optimal angle would be in the range of 10 to 15 degrees was confirmed as it was found to be 12 degrees. The results we obtained were somewhat close to the ones states in *The Physics of Frisbees* as the maximum distance reached by the frisbee when thrown at a 12 degrees angle was of 40 meter while achieving a height of 7.7 meters. Our findings were relatively close to theirs. A java code was provided in the article, and I would say that we took way more components into account then they did and that is probably the reason why we got slightly different results along with the fact that we could not fix a problem in our program.

**References**

Morrison, V.R. “The Physics of Frisbees”. *Electronic Journal of Classical Mechanics and   
 Relativity.*Http://scripts.mit.edu/~womens-ult/frisbee\_physics.pdf.

1. Morrison, V.R. “The Physics of Frisbees”. *Electronic Journal of Classical Mechanics and Relativity.*Http://scripts.mit.edu/~womens-ult/frisbee\_physics.pdf. [↑](#footnote-ref-1)