**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.

**Model**

Since there is a lot of variables to consider, we will have to use many different equations. An equation for the drag force will be used

(1)

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

. (2)

Same thing with the lift forceand a lift coefficient *Cl* will need

to be calculated first

(3)

(4)

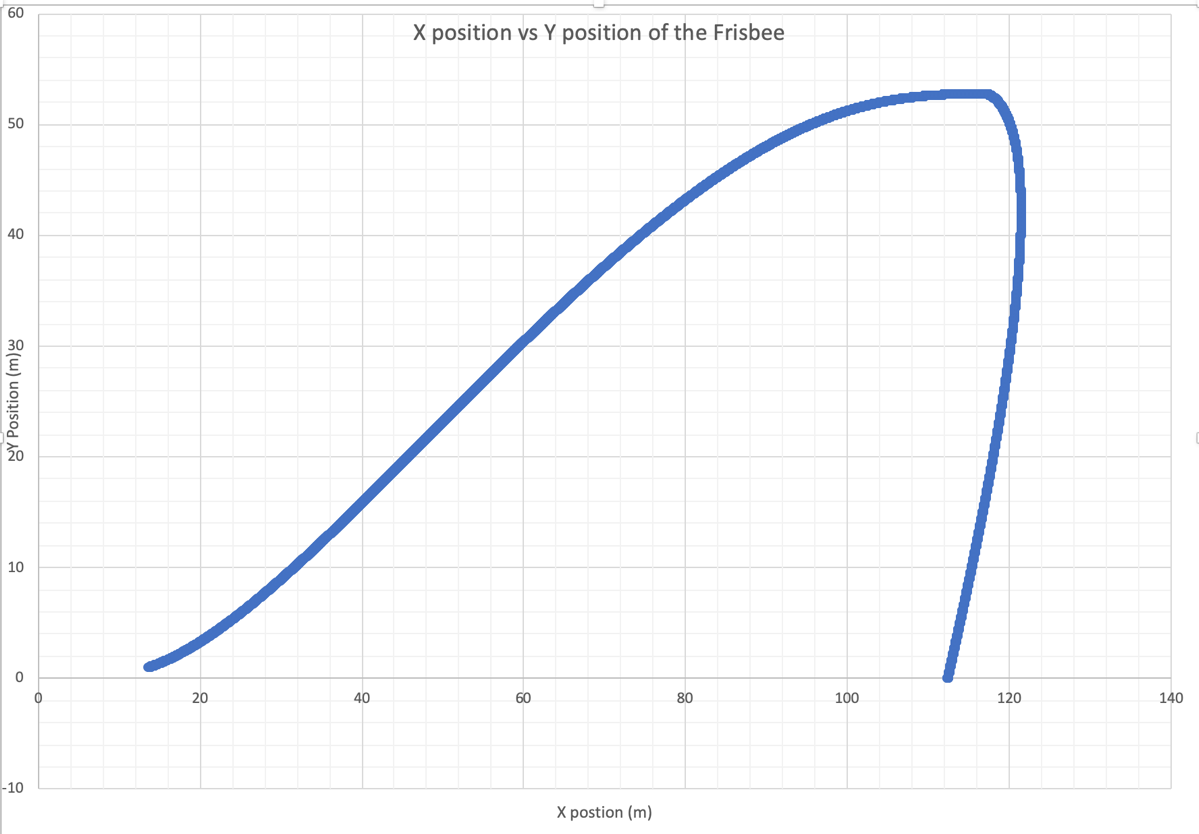
The Frisbee’s area, the air pressure, the density of the fluid which in this case will be air, the acceleration and the height of the fluid will have to be considered in these formulas too. The velocities and pressures are linked together, and they act inversely to each other. The average velocity of a Frisbee throw we will use is approximatively 14 m/s, the standard air density we will use in this case will be 1.23 kg/m3, the Frisbee’s area will be 0.0531m2 and as we might need it, the viscosity of air used will be 1.73x10-5N s/m2. When we will calculate the acceleration of the object, we will decompose it into two components (x and y) because it is a two-dimensional problem.

**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 and pressure, the induced drag, the slope of the graph of a linear function of angle of attack, the angle of attack that produces the less lift, and the initial velocity of average of a throw from the data we have looked at.

In one of our method, we have calculated the distance travelled by the frisbee and we used data such as the velocity, the x and y initial positions 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 distance. Since all of these values are always changing, this method is also used to update them each time the program is run. Also, to make sure we get the most accurate answer, a Golden Search method will be created to optimize the results to find the best angle of attack. Since we know the maximum (90o) and minimum (0o) angles at which we can throw a Frisbee, this method will be helpful to narrow down the possible angles to find the best one. Since this method is usually used to find a minimum value, we will have to use the inverse of our function because in this case, we want to find a maximum value. The Golden Search will also be useful because it is a unimodal function, meaning there is a single optimal value. 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**



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)