ENAE311H Final Project Report

Fouad Ayoub, Eli Mirny, Riley Edgar

Introduction

In this project, our group was assigned to design a hypersonic waverider that would perform a freefall/glide from an altitude of 40 km beginning at Mach 10. This waverider must be designed to optimize distance traveled over the course of the glide while remaining under the volume, length, and density constraints given. We assumed inviscid and ideal air, as well as used a standard atmosphere model for our calculations.

We decided to model the shock wave that would be produced in the hypersonic flow as a conical shock wave being generated by a cone. This conical shock wave model allows for more flexibility in the waverider design, as the conical shock wave requires us to define a quasi-3D flow field, from which we can trace out the surfaces of the waverider. Additionally, using a conical shock wave would allow us to get experience numerically solving pressure and velocity distributions in a 3D flow field, which is described in a later section as well as in section 13.6 of Anderson.

Numerically Solving for Conical Shock Wave

Using the procedure outlined in section 13.6 of Anderson, we can numerically solve for the flow field properties after a shock wave. To begin, we define the coordinate system as shown below, with a velocity after the shock in components of V_θ and V_r directions. We initialize an arbitrary shock angle of 20°, and our goal is to determine the angle of a solid cone that would be generating this conical shock.

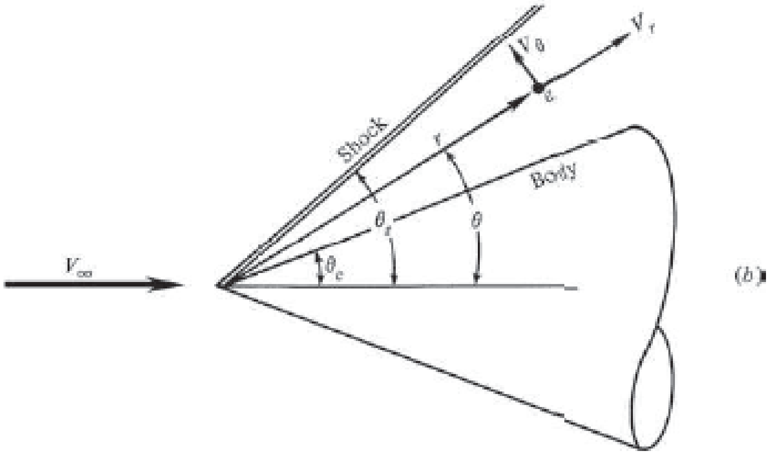

Figure 1: Defining the angles and reference frame for the velocity along streamlines coming out of the vertex of a conical shock wave, and along a conical body.

We define our free stream conditions, as perfectly uniform before the shock wave, with the flow properties from the standard atmospheric model and the Mach number at 10. Using these assumptions along with axisymmetric flow assumption, constant flow properties along rays from vertex (and irrotational), adiabatic flow, and conservation of mass, we get the Taylor Maccoll equation for the solution of conical flows.

$$\boxed{\begin{aligned}&\frac{\gamma-1}{2}\left[V_{\max}^2 - V_r^2 - \left(\frac{dV_r}{d\theta}\right)^2\right]\left[2V_r + \frac{dV_r}{d\theta}\cot\theta + \frac{d^2V_r}{d\theta^2}\right]\\&\qquad - \frac{dV_r}{d\theta}\left[V_r\frac{dV_r}{d\theta} + \frac{dV_r}{d\theta}\left(\frac{d^2V_r}{d\theta^2}\right)\right] = 0\end{aligned}} \tag{13.78}$$

Figure 2: The Taylor Maccoll equation, see Anderson, pg 359

This equation can be normalized with the quantity V_{max} and simplified to a second order differential equation in terms of V_r and θ by noting that the $V_\theta = dV_r/d\theta$. This brings us to the following ODE.

$$\begin{aligned}&\frac{\gamma-1}{2}\left[1 - V_r'^2 - \left(\frac{dV_r'}{d\theta}\right)^2\right]\left[2V_r' + \frac{dV_r'}{d\theta}\cot\theta + \frac{d^2V_r'}{d\theta^2}\right]\\&\qquad - \frac{dV_r'}{d\theta}\left[V_r'\frac{dV_r'}{d\theta} + \frac{dV_r'}{d\theta}\left(\frac{d^2V_r'}{d\theta^2}\right)\right] = 0\end{aligned} \tag{13.80}$$

Figure 3: The normalized and simplified version of the Taylor Maccoll equation. See Anderson, pg 359

We can further reduce this into first-order form by defining the vector V' = [V_r' V_θ'], taking its differential, and replacing $dV_\theta/d\theta$ with the right hand side of a second order ODE solved for $d^2V_r'/d\theta^2$. After doing so, we now have a first-order system of equations that we can numerically solve with Matlab's ode45() function. We then find the root of the solution to V_θ. This is, by definition, the angle at which a solid cone must be in order to generate a conical shock wave with the angle set initially.

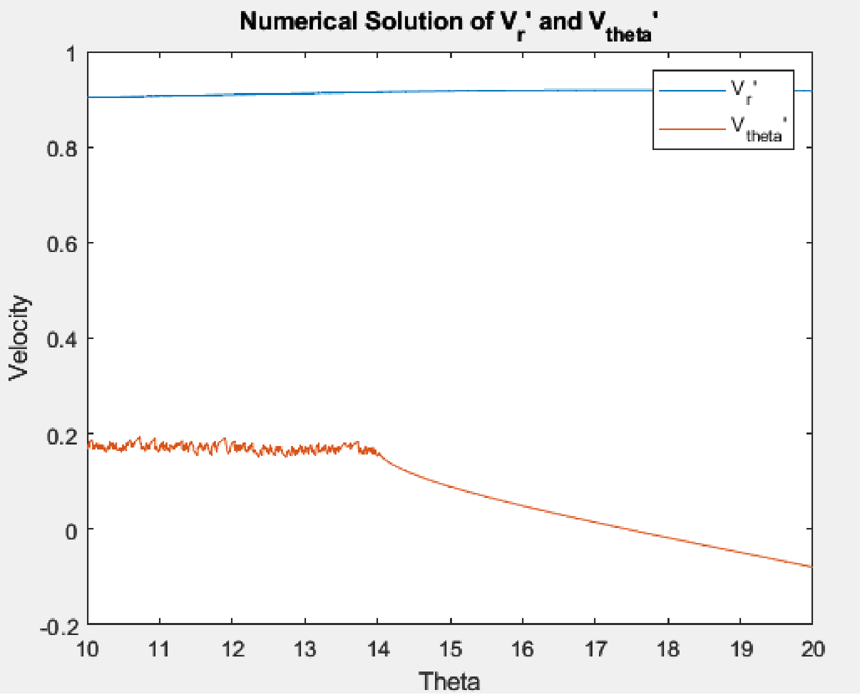

Figure 4: Numerical Solution of V_r‘ and V_θ’. Note that we are numerically solving *backwards*, from θ=20° to θ=10°, such that the initial conditions describe the state of the system at θ=20°.

We then solve for a Mach field in order to determine the properties of each ray stemming from the vertex of the cone, since any change in velocity from the immediate inside of the shock wave to the surface of the solid cone would be isentropic and solely dependent on the Mach numbers of each state.

Finally, we wrap this entire numerical cone solution as a function in Matlab that can be solved to produce a pressure distribution across our waverider shape. It will then return a resultant force quantity and an angle at which the force is acting on the waverider with respect to the vertical. The waverider shape remains constant throughout the changing regimes of speed, but the conical shock wave may taper or widen throughout the descent. Thus, the conical shock wave interior flow properties will need to be discretely solved for each time step of the waverider’s trajectory.

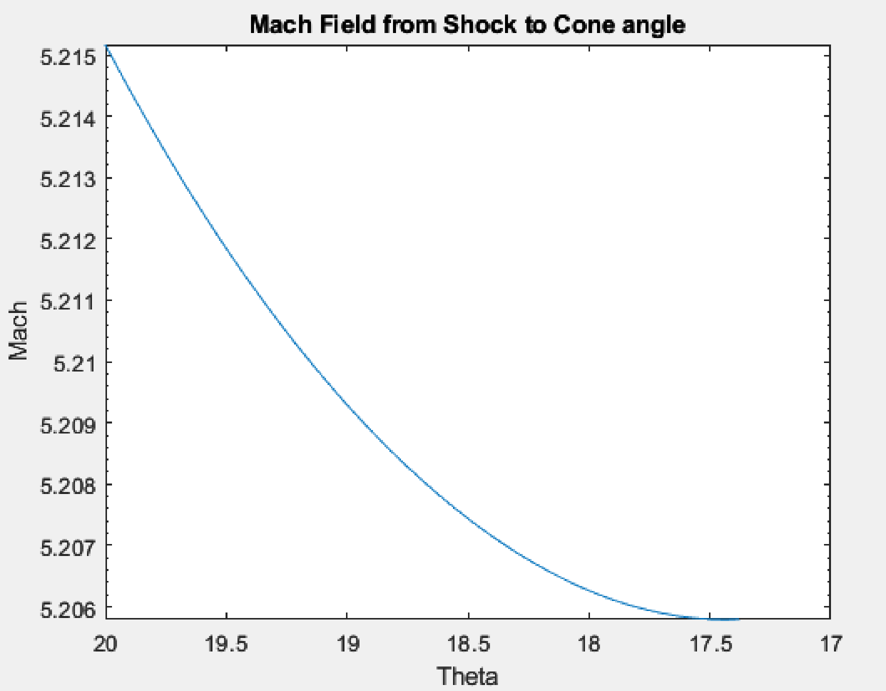

Figure 5: Mach field from the conical shock wave angle to the angle of the solid cone.

Determining Shape of Waverider

We decided to make the waverider a conical-based shape, with equations of cones defining the upper and lower surfaces. A conical shaped wave rider has a better volumetric efficiency compared to a carat waverider, which is a measure of how well the surface area of the waverider is being utilized for generating lift and minimizing drag. Additionally, we ensured that the waverider was no longer than 1 meter by constraining its leading edge to 5 meters downstream of the vertex of the shock wave, and its trailing backside to 1 meter behind that. The actual equations for the conical surfaces are tilted downwards slightly in order to fit on top of the waverider, so the length of the waverider is slightly less than 1 meter. When calculating the exact length of the waverider using the distance from the leading edge to the trailing edge, we get that it is 0.8189 meters.

To design the waverider, we first determine the slope of the cones that will define the surface to ensure their leading edges are both aligned with each other and the conical shock wave at Mach 10 itself. Then, we vary the eccentricity of each conical surface and perform an optimization with respect to the resulting pressure that the surface generates. We could not change the eccentricity too much because of the limitations required on the leading edges.

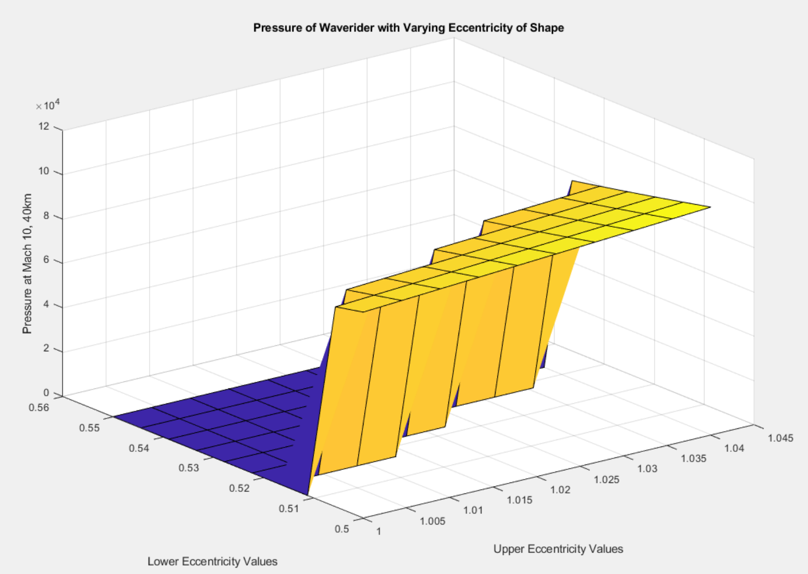

Figure 6: Optimization surface of the eccentricity values of the bottom and upper surface, with the resultant pressure at Mach 10 being the cost function.

Once we get the values for eccentricity that both maximize the resultant pressure generated by both surfaces and ensure the leading edges meet the conical shock wave and each other, we can then define the final upper and lower surfaces of the wave rider.

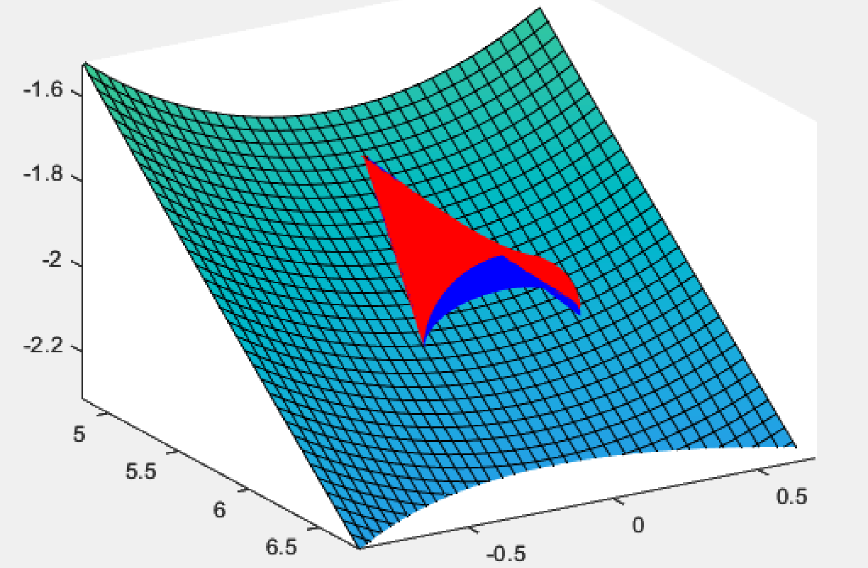

Figure 7: Final waverider shape, shown “riding the wave” of the conical shock wave produced during hypersonic travel.

Volume Calculation with Monte Carlo Method

To calculate the volume of the wave rider we used the Monte Carlo method in three dimensions. The Monte Carlo method is characterized by choosing random points in 3D space, checking whether or not the points are contained within a selected region defined by bounds on the three dimensions, and using the ratio of the points inside the region over the points outside the region to approximate the volume of the region.

Before we worked on our own waverider, we analyzed the convergence of a Monte Carlo method for finding the volume of 3D shapes. We used the volume of a sphere with radius one as our test case, since we can easily determine the exact volume of such a sphere. We then ran the Monte Carlo method with a varying number of guesses, and plotted the relative percent error versus the number of guesses on a log-log scale graph to view the convergence rate. We find that the method converges with O($n^{4.5}$) complexity. A quadratic convergence is known to be desirable, so the algorithm having a better than quadratic convergence means it will work well enough for our purposes.

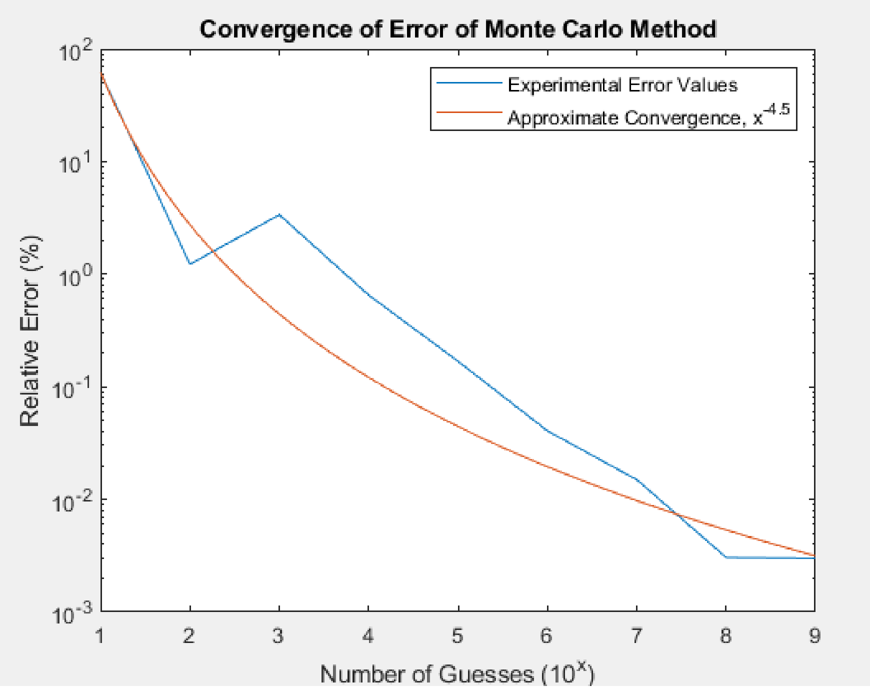

Figure 8: Convergence rate of the Monte Carlo method when calculating the volume of a sphere

When performing the Monte Carlo method on our own waverider body, we placed 100,000 random points in a 0.8 by 1 by 0.2 cube. Every point that we place is checked for whether it falls inside of the waverider. If it does, we count it and store it into a variable “inside”. To calculate the volume we divide the number of points that landed inside of the waverider by the total number of guesses we made and multiply it by the volume of 0.8*1*0.2=0.016m^3.

Initially, we would calculate the bounds of our volume each iteration of a guess, but we quickly realized that this would take too long. When completing further analysis of our code, we found that the time complexity of such an algorithm would be O(YXn), where n is the number of total guesses, and Y and X are the number of discretized points that make up the y and x boundaries, respectively. To allow for a better time complexity of the Monte Carlo volume algorithm, we decided to move all of the y bounds calculations to an offline computation that would be done one time. This reduces our online time complexity to a mere O(n), and our offline computation time complexity to O(XY). Now, we are ready to use this algorithm to calculate the volume of our waverider.

Our volume ended up being 0.0066m^3.

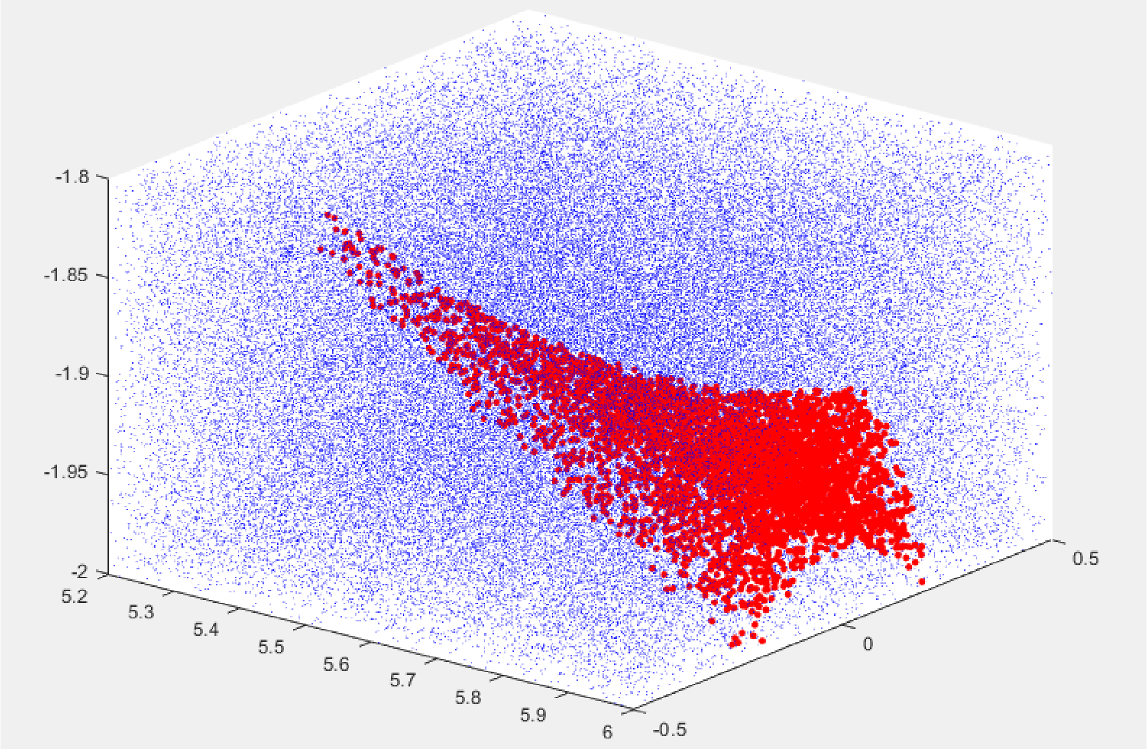

Figure 9: The Monte Carlo method will determine the volume based on the ratio of points inside the region versus outside the region and within the boundary box.

Equations of Motion

$$\ddot{y} = Lift / m - g$$
$$\ddot{x} = - Drag / m$$

Since our bottom surface is defined to be parallel to the flow we assume the resultant pressure force is going to be perpendicular to the flow. Based on those assumptions we define lift and drag based on the magnitude of the resultant force and the deflection angle of the flow.

$$D = f\ (altitude,\ Mach\ number)$$
$$L = f\ (altitude,\ Mach\ number)$$

```
[t,y] = ode45(@myodefun2,[0 300] ,[0, 10*sqrt(atmosisa(40000)*R*gam),
40000, 0],[], m, D, L, R, gam, g);

% y = [x dx y dy]
function ydot = myodefun2(t,y,m, D, L, R, gam, g)
    ydot(1,1) = y(2);
    ydot(2,1) = -D(sqrt(y(2)^2+y(4)^2)/sqrt(atmosisa(y(3))*R*gam),
y(3))/m;
    ydot(3,1) = y(4);
    ydot(4,1) = L(sqrt(y(2)^2+y(4)^2)/sqrt(atmosisa(y(3))*R*gam),
y(3))/m-g;
end
```

Results

After numerically solving the ODEs that describe the trajectory of the waverider, the results of the flight were compiled in the graphs below. We show altitude versus distance of the trajectory and the Mach number versus distance traveled to get a good idea of the behavior of the waverider.

In the altitude versus distance graph, we noticed a discontinuity in what would be an expected trajectory at around 700 km. We believe that this discontinuity might be a product of the multiple simplifications we used to optimize our code for a reasonable completion time, as well as the fact that all of our computations are numerically solved upon numerically solved values. These characteristics of our code mean that the functions that are involved in numerically solving the ODEs for the equations of motion have no guarantee of smoothness everywhere.

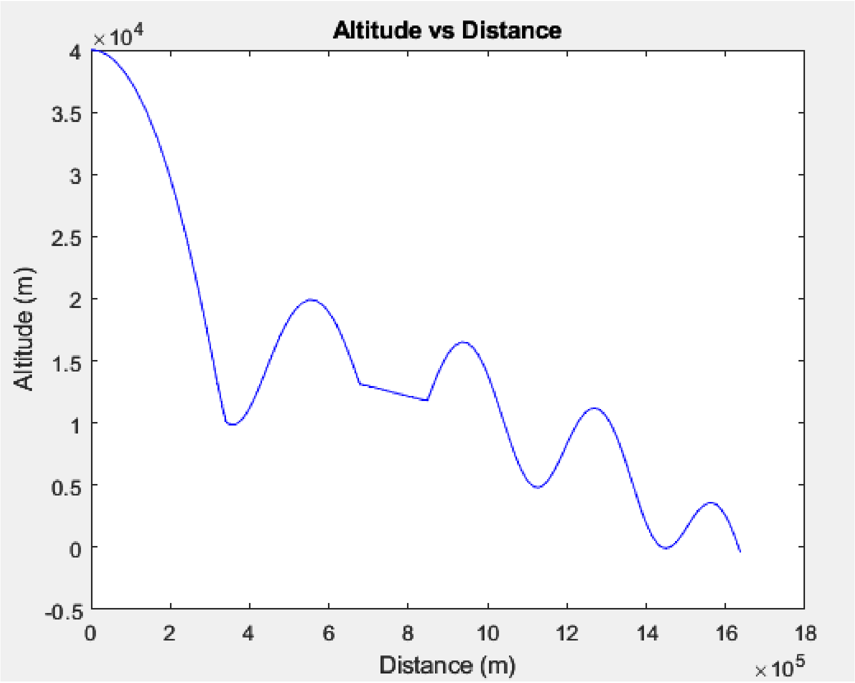

Figure 10: Altitude vs. distance graph of trajectory

The waverider starts at 40 km where air density is very low, so at the beginning of its path it rapidly falls until it reaches around 10 km. At that height, the higher density along with high speed creates a lift force strong enough to overcome the weight of the waverider, thus it rises in altitude again. This trend continues causing the wave rider's altitude to oscillate as it travels. Its total range ends up being around 1637 km.

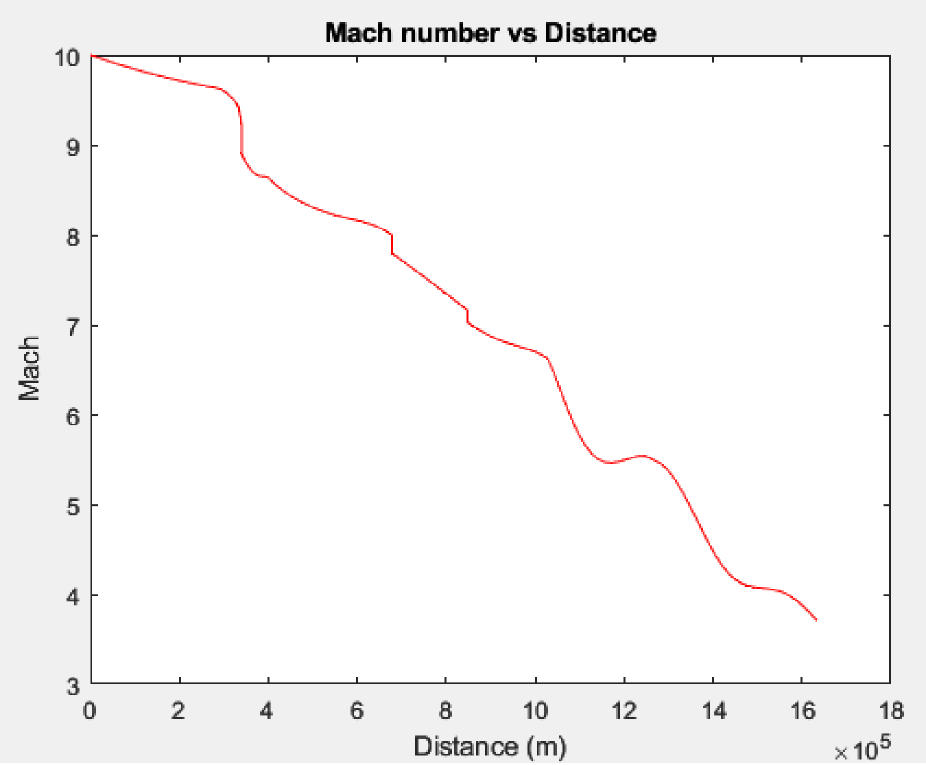

Figure 11: Mach number versus distance during flight

Our final Mach number ends up being around 3.7 after reaching the ground at t=13.65 minutes. The slope of the Mach number depends on the drag force and the wave rider's vertical speed. The rate of change of the altitude affecting the Mach number is most pronounced at around Mach 4 where we see a big drop. This is due to the rapid rise of the waverider’s altitude right around that Mach number, where its kinetic energy is being converted to potential energy, exchanging speed for height in the process.

When viewing the L/D during the trajectory, we notice that the waverider seems to perform better as the flight goes on. This may be due to the fact that as the waverider descends towards the ground, the density of the atmosphere increases, which causes an increase in lift.

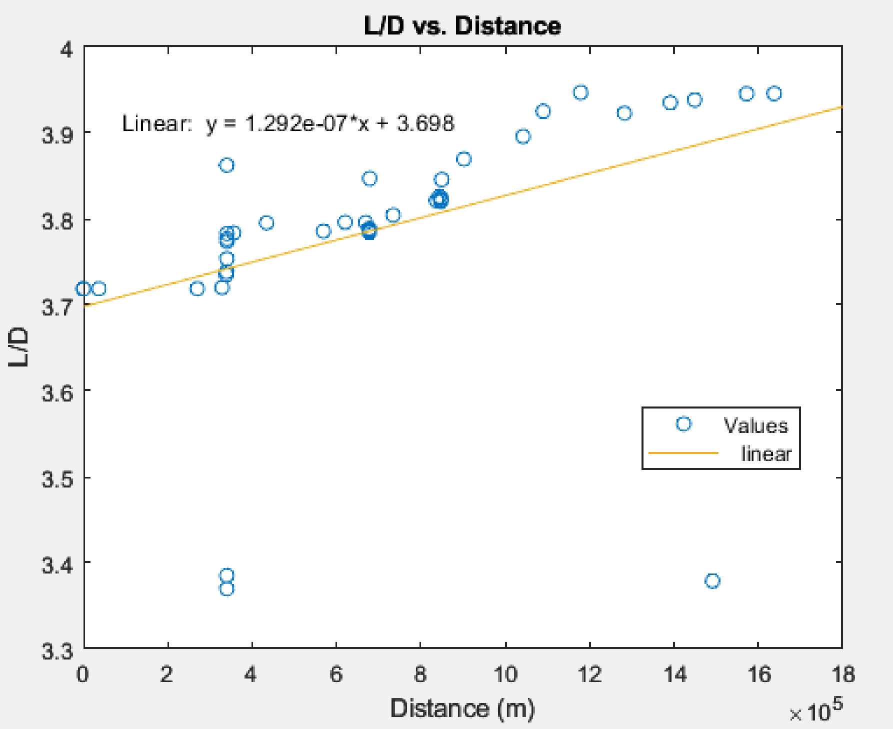

Figure 12: L/D versus distance during trajectory, along with a linear fitting to the data that shows the upward trend as the flight goes on.

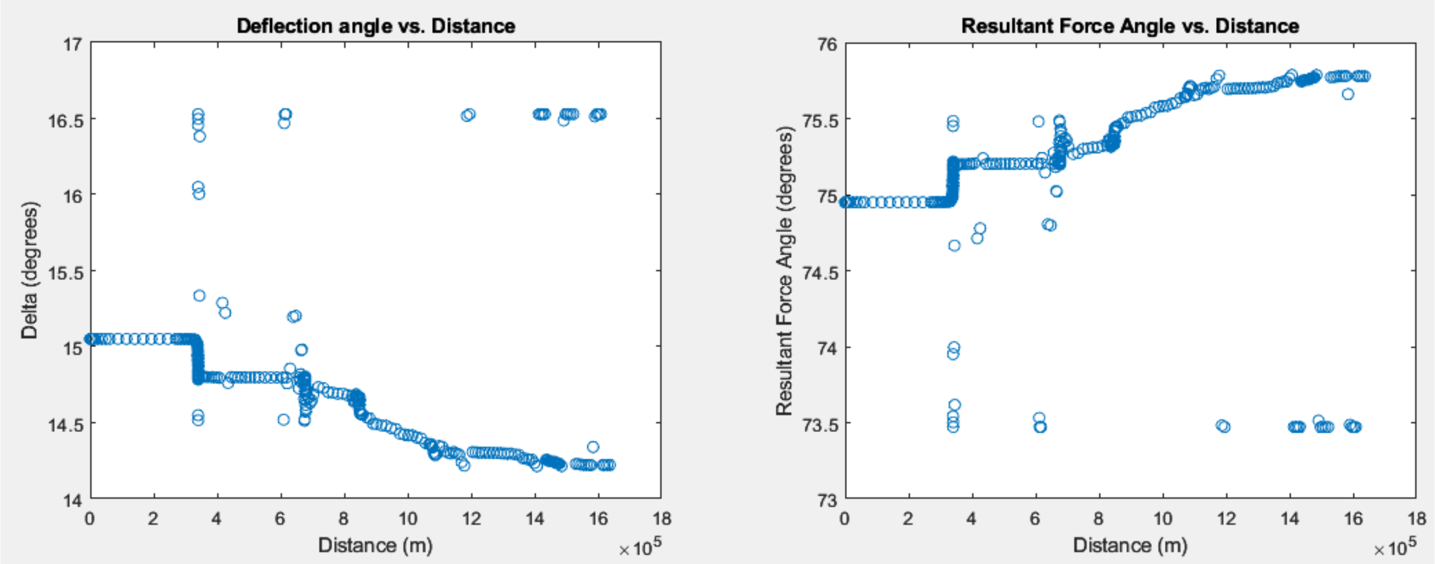

Figure 13: The left graph shows the deflection angle of the flow due to the conical shock wave as a function of distance traveled by the wave rider, and the right graph shows the corresponding resultant force vector angle, which is complementary to the deflection angle.

Another interesting measure is the deflection angle and resultant force vector angle throughout the trajectory. Since the waverider surface was designed such that it was parallel to the flow behind the shock wave, we assumed that the resultant force vector would be at an angle complementary to the deflection angle (the two angles sum up to 90°). Thus, plotting them side by side, we can see exactly how the deflection angle changes throughout the trajectory, as well as how it affects the force vector angle. We notice that there are a few large drops in the deflection angle, more specifically at around 400 km, 600 km, and 1100 km, which is supported by our altitude versus distance graph (Fig. 10), which shows that at those points, the waverider goes from losing altitude to gaining altitude.

Finally, for a better understanding of the pressure distribution behind the conical shock wave, we produced some plots of the pressure distribution with respect to the angle of the streamline at different points of the trajectory. Here, one can see the subtle variation in pressure behind the conical shock wave. It is worth noticing that the variance in pressure behind the shock actually seems to decrease as the altitude and Mach number decrease.

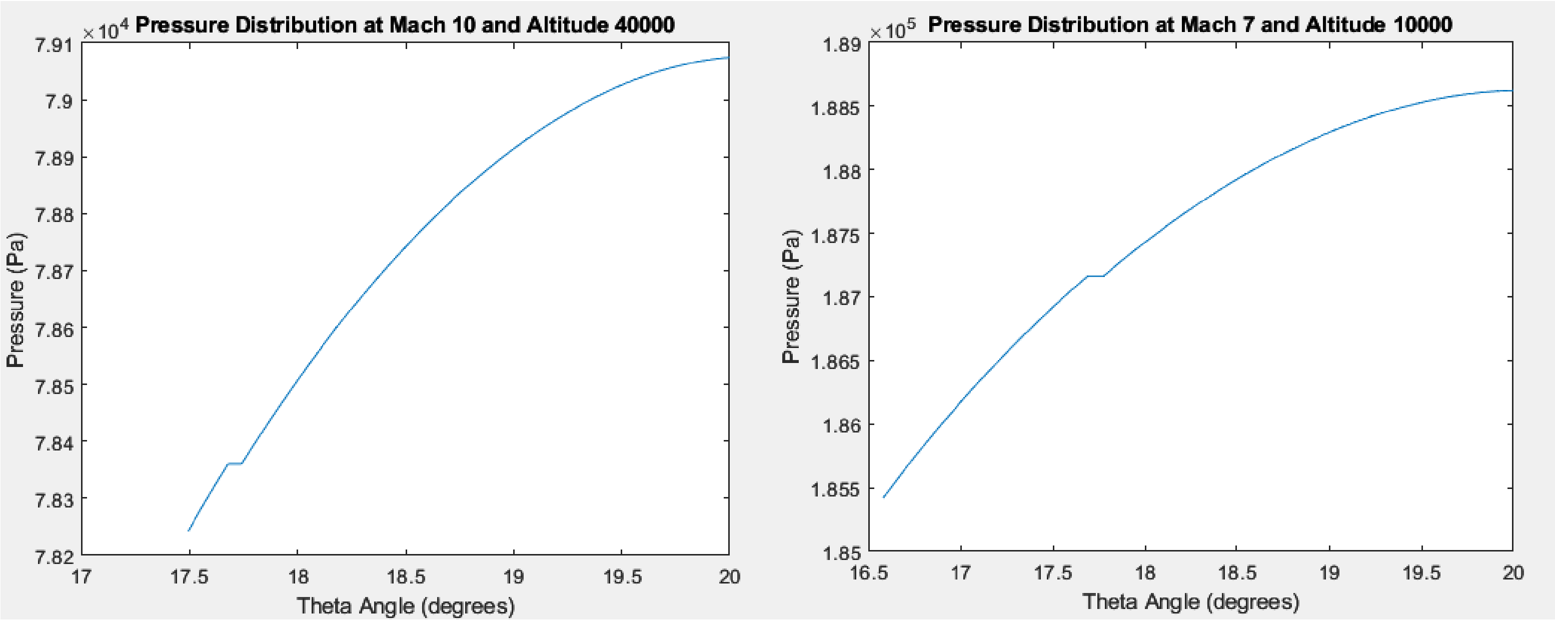

Figure 14: The left graph shows the pressure distribution behind a conical shock wave at Mach 10 and an altitude of 40 km. The right graph shows the pressure distribution at Mach 7 and an altitude of 10 km.

Conclusion

Over the course of this project our group constructed many different solvers using Matlab. These solvers include a conical shock wave solver, a Monte Carlo Solver that conducted the volume calculations, an optimization scheme for the waverider's shape, a CFD solver that would return the resultant force acting on the body, and a solver of dynamics on the wave rider. Throughout the process, we also had the opportunity to gain experience on and greaten our understanding of many of the aerodynamics concepts we learned in class. We learned how to make and justify simplifying assumptions that significantly improved the computational time of our code, how to numerically solve for conditions that occurred behind a conical shockwave, learned more about applying the Monte Carlo method leading us to learn about the convergence rate for the method, how to optimize our code through the use of an offline/online scheme of computation, and how to best depict the dynamics of the trajectory system in a way that highlighted important events in the flight. If we were to do this project again we would make a few changes to our project to better improve the accuracy of our results and make the process quicker. Most notably we would use a more comprehensive computational model for calculating the force as it would produce a more accurate trajectory that did not include discontinuities in its path. To do this we would use quadrature points for the pressure distribution across the surfaces of the waverider with problem-informed quadrature weights. However, we would also make changes to implement an offline/online scheme for the CFD computations so that the trajectory dynamics solver would run faster, and generate images of the pressure distribution around the surface of the waverider allowing us to better understand how the waverider is generating lift and therefore leading to a better intuition regarding the shape itself.