Using Numerical Methods to Approximate a Solution to the Prandtl-Glauert Equation

Blake Logan

Arizona State University, Tempe, AZ, 85202, United States

The objective of this investigation was to approximate a solution to the Prandtl-Glauert Equation using a numerical method. The validity of this solution was determined for two cases. A symmetric, circular-arc airfoil was analyzed at a Mach number of 0.85 for transonic similarity parameter values of 1.3 and 3. These values correspond to transonic and subsonic flow, respectively. A numerical method was constructed such that the perturbation potential field around the airfoil is modeled within a 100 by 100 mesh. Using a Jacobi Iteration Method, the value of perturbation potential was calculated at each point on this grid. The solution was achieved using an initial guess that updates to the current solution at each iteration. When the error between current and previous iteration is sufficiently small, the process is terminated. Using the perturbation potential field, the pressure coefficient distribution along the airfoil surface was calculated. From the pressure coefficient distribution, a drag coefficient value was determined. These solutions were compared to solutions produced via the Transonic Small Disturbance Equation. It was found that the numerical method using the Prandtl-Glauert Equation returned symmetric pressure distributions for both cases. The magnitudes of these pressure coefficient values were dependent on the thickness of the airfoil. For a similarity parameter of 1.3, the pressure coefficient magnitudes were much greater as the airfoil was thicker. The symmetric pressure distributions resulted in near-zero drag results for both cases. It was determined that the subsonic solution was valid for the subsonic case, although the transonic solution was more accurate. The subsonic small disturbance solution was incapable of accurately modeling transonic flow. The numerical method developed was proven sufficient for the Prandtl-Glauert Equation, although optimization may be required for the Transonic Small Disturbance Equation. Overall, it was found that numerical methods utilizing the Transonic Small Disturbance Equation and perturbation theory are a valid method of modeling transonic flow given that assumptions made are reasonable for the given model.

I. Nomenclature

a	=	Speed of sound
a_∞	=	Upstream speed of sound
C_D	=	Drag coefficient
C_p	=	Pressure coefficient
c	=	Chord length
I	=	Number of mesh points in x dimension
i	=	Iteration value (x dimension)
J	=	Number of mesh points in z dimension
j	=	Iteration value (z dimension)
K	=	Transonic similarity parameter
M	=	Mach number
M_∞	=	Upstream Mach number
R	=	Radius
t	=	Thickness
u	=	Horizontal perturbation velocity
v	=	Velocity
v_∞	=	Upstream velocity
w	=	Vertical perturbation velocity
Z	=	Vertical location of airfoil

Z'	=	Slope of airfoil surface
α	=	Angle of attack
β	=	Prandtl-Glauert Correction
Δx	=	Width of one cell in mesh
Δz	=	Hight of one cell in mesh
γ	=	Specific heat ratio
τ	=	Dimensionless thickness
ϕ	=	Full potential
φ	=	Perturbation potential
$\hat{\varphi}$	=	Perturbation potential at previous iteration

II. Introduction

In low-speed aerodynamics, thick, cambered airfoils are used to maximize the lift of an aircraft. This design causes the flow to increase in velocity on the upper surface of the airfoil, thus decreasing the pressure. This pressure differential is the lift force. This is a simple, well understood concept. However, when an aircraft reaches velocity near the speed of sound, the flow is much more complex. As discussed, the airfoil design will cause the flow to increase velocity along the upper surface. If the flow is initially close to a Mach number of one, the flow will reach supersonic speeds somewhere along the surface of the airfoil. As is well understood in this course, when at supersonic speeds, shockwaves become possible. As the flow speeds up along the upper surface, a "pocket" of supersonic flow will be created near the leading edge of the airfoil. However, there is still subsonic flow near the trailing edge of the airfoil. As the flow is traveling faster than the speed of sound (a disturbance), the signal to slow to subsonic is not received until the instant it occurs. This implies the existence of a shockwave. The normal shock wave created can be very strong and acts as the "barrier" between supersonic and subsonic flow on the surface of the airfoil. Refer to Figure 2.1 below.

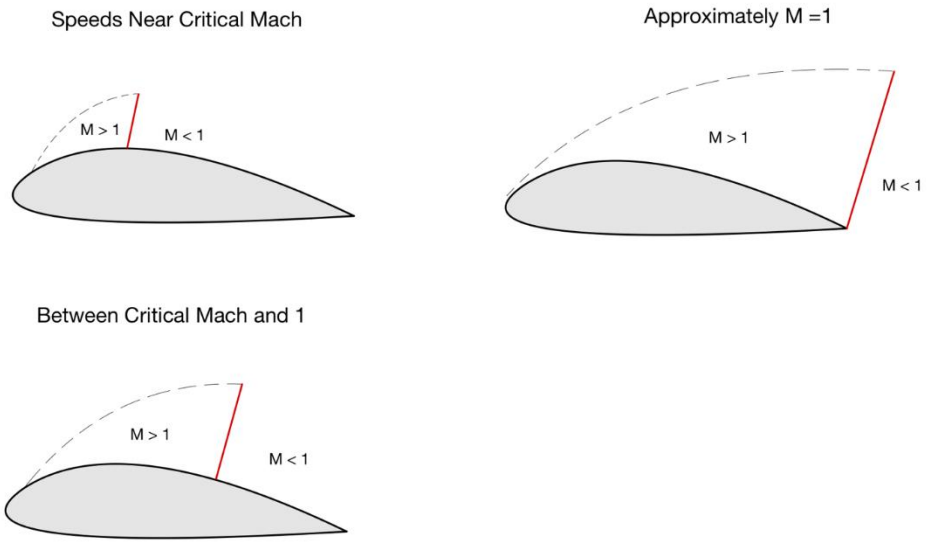


Figure 2.1: Formation of Shock in Transonic Flow

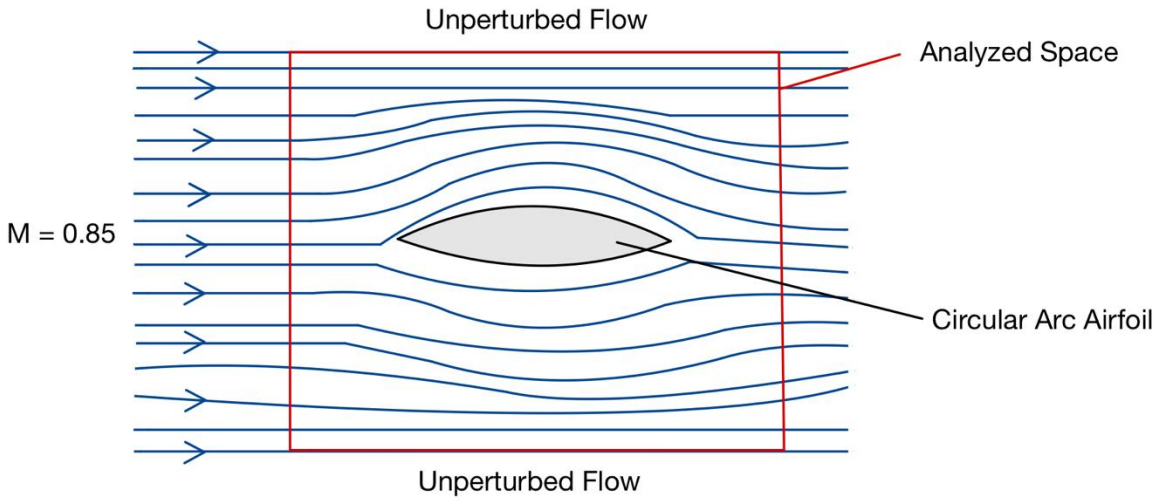
As can be seen in Figure 2.1, a normal shock wave will occur along the surface of the airfoil. The critical Mach number (around 0.8) is the Mach number in which supersonic flow occurs on the airfoil. For Mach numbers between critical and one, the shock occurs between the leading and trailing edge of the airfoil. As Mach number increases, this shock wave will become stronger. This will cause a significant amount of drag. In subsonic flow, there is only viscous and induced drag. Now, wave drag has been introduced due to the shock. However, wave drag is also a problem in supersonic flow. Nevertheless, supersonic flow produces less drag than transonic. This is

because the shock will cause boundary layer separation aft of itself. The boundary layer separation creates a low-pressure zone resulting in a pressure gradient that produces significant drag. Although the shock increases in strength as Mach number increases, the shock is pushed further back along the airfoil until it reaches the trailing edge. At this point, supersonic flow envelopes the entire airfoil and the drag will begin to decrease again. Therefore, transonic flow is a significant issue in aerodynamics. An aircraft must be capable of producing enough thrust to pass through the transonic regime into the supersonic regime. Hence, innovative designs such as supercritical airfoils were implemented.

The significance of drag in transonic flow is not the only quality that makes this regime of such interest in aerodynamics. For subsonic and supersonic flow, there are well known equations that can be used to solve for flow properties. However, transonic flow fits into neither of these categories. This makes it extremely difficult to solve for flow properties mathematically or to predict the flow behavior. Despite this, transonic flow is a necessary barrier between subsonic and supersonic speeds. Therefore, modeling and understanding transonic flow is crucial in the aerodynamic field. Due to the complexity discussed above, there is no analytic solution to the transonic equations. This means that numerical methods must be used to solve the aerodynamics of transonic flow. The purpose of this investigation is to develop a numerical method to solve the subsonic small disturbance equation. The transonic small disturbance is far more complex; however, this investigation will outline the errors that arise when predicting transonic flow with subsonic equations. Moreover, the method developed will be useful in understanding the method involved in solving the Transonic Small Disturbance Equation.

In this investigation, a model will be solved using the numerical method discussed above. The model is constructed such that small disturbances, inviscid, and small angle assumptions can be made. These assumptions allow for the problem to be solved in the simplest manner. In addition, a symmetric, circular-arc airfoil will be analyzed. This will be a very thin airfoil, allowing for the appropriate assumptions to be made. The precise thickness is determined by the transonic similarity parameter and upstream Mach number. To analyze transonic behavior, the Mach number will be modeled sufficiently close to one. Two transonic similarity parameters will be chosen such that both subsonic and transonic flow is modeled. The validity of this solution will be analyzed in comparison to accepted results produced via the transonic solution. A numerical method will be designed to solve for the perturbation potential field surrounding this airfoil.

Using a carefully chosen solution space, the perturbation potential field can be used to solve for the flow properties, such as; pressure coefficient and drag coefficient. Perturbation potential is a useful tool in solving complex flow problems. In this case, the perturbation potential at a point represents the change in velocity due to a small disturbance. In this investigation, the perturbation potential can be solved for using the Prandtl-Glauert Equation which is a subsonic small disturbance approximation. (More detail in Procedures). In this model, the small disturbance is the airfoil. This perturbation potential can be used to linearize the governing equations of complex flow, as will be discussed in the next section. Calculating the perturbation potential in a field around the object of



interest allows for the deduction of aerodynamic forces. Clearly, this is extremely useful in modeling complex flow scenarios. This investigation will use this linearized model along with a numerical algorithm to solve for the forces on the airfoil described above. Refer to Figure 2.2 below.

Figure 2.2: Physical Model for Investigation

III. Experimental Design and Procedure

In this experiment, numerical methods were used to solve the Prandtl-Glauert Equation. The Prandtl-Glauert Equation for subsonic flow operates under multiple assumptions. To achieve the final equation, one must begin with the Full Potential Equation in two dimensions, which is derived from the conservation of momentum and energy equations. The equation is shown below.

$$0 = \frac{\partial^2\phi}{\partial x^2}\left(1 - \frac{1}{a^2}\frac{\partial\phi}{\partial x}\right) + \frac{\partial^2\phi}{\partial z^2}\left(1 - \frac{1}{a^2}\frac{\partial\phi}{\partial z}\right) - \frac{2}{a^2}\left(\frac{\partial\phi}{\partial x}\frac{\partial\phi}{\partial z}\right)\frac{\partial^2\phi}{\partial x\partial x} \quad (1)$$

Equation (1) is a non-linear equation. To linearize, the full potential can be written in terms of the perturbation potential as:

$$\phi = v_\infty(x + \propto z) + \varphi \quad (2)$$

This linearization holds under the assumptions of small disturbances, thin airfoil, and small angles of attack. In addition, velocity can be written in terms of perturbation velocity as:

$$v = (v_\infty + u)^2 + (v_\infty + w)^2 \quad (3)$$

Implementing these substitutions to equation (1) and multiplying by a^2 yields:

$$0 = \frac{\partial^2\phi}{\partial x^2}(a_\infty^2 - (\gamma-1)v_\infty\frac{\partial\phi}{\partial x} - (v_\infty + \frac{\partial\phi}{\partial x})^2) + \frac{\partial^2\phi}{\partial z^2}(a_\infty^2(\gamma-1)v_\infty\frac{\partial\phi}{\partial x} - \left(v_\infty \propto + \frac{\partial\phi}{\partial z}\right)^2) - 2\frac{\partial^2\phi}{\partial x\partial x}(v_\infty + u)(v_\infty + w) \quad (4)$$

$$\bar{u} = \frac{u}{a_\infty} \quad (5)$$

$$\bar{w} = \frac{w}{a_\infty} \quad (6)$$

Under the assumption that all small terms squared are negligible relative to a_∞, equation (4) can be written as:

$$0 = \frac{\partial^2\phi}{\partial x^2}(1 - (\gamma-1)M\bar{u} - M_\infty^2) + \frac{\partial^2\phi}{\partial z^2}(1 - (\gamma-1)M\bar{u}) - 2\frac{\partial^2\phi}{\partial x\partial x}M\bar{w} \quad (7)$$

For Mach numbers less than 5, (excluding hypersonic flow) it can be assumed that $M\bar{u}$ and $M\bar{w}$ are negligible. With this assumption, equation (7) becomes:

$$(1 - M_\infty^2)\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0 \quad (8)$$

Equation (8) is known as the Prandtl-Glauert Equation. This equation, along with a central differencing numerical method will be used to solve the perturbation potential field containing a thin airfoil. The space in which the perturbation field is solved will be carefully chosen and outlined in the remaining procedure. It should be noted that this equation is only valid for subsonic flow and operates under small disturbance assumptions.

To solve this equation, a central difference accuracy model will be used. A central difference approximation operates by estimating the derivative at a point using the points just ahead and behind. When points are evenly spaced, the slope at a point can be approximated with a general rise-over-run formula. Hence, the formula is the difference between the function value at the points behind and ahead of the point of interest over the x-distance between them. Refer to Figure 3.1 below.

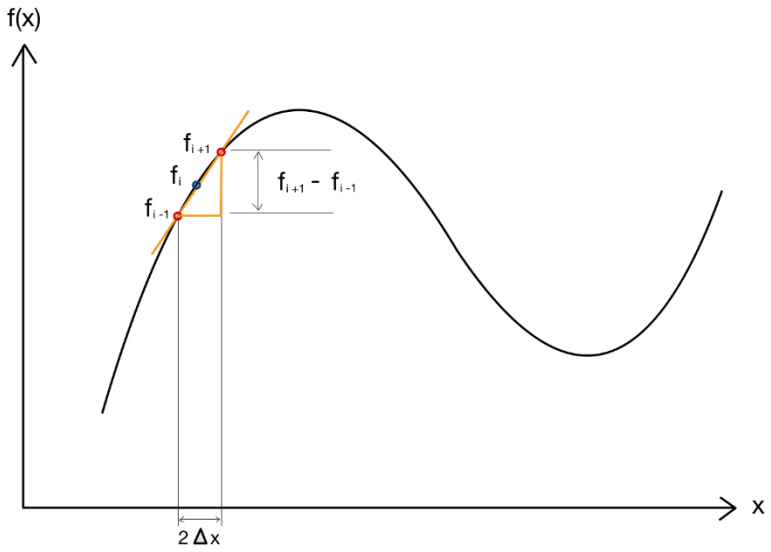


Figure 3.1: First Order Central Differencing

In one dimension, the general form is as follows:

$$f' = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \tag{9}$$

Equation (9) is for the simplest case; a first order, one-dimensional approximation. To solve the Prandtl-Glauert Equation, central differencing will be used to solve a second order, two-dimensional approximation. The second order central difference formula is derived from the Taylor Series Expansion. The derivation is performed as follows:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \cdots \tag{10}$$

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) - \cdots \tag{11}$$

Adding equations (10) and (11) will cancel out the odd ordered terms and the higher order terms will be truncated. After rearranging to solve for the second derivative, equation (11) becomes:

$$f'' = \frac{f_{i+1} - 2f_i - f_{i-1}}{\Delta x^2} \tag{12}$$

Equation (12) is the second order central difference formula. In two dimensions, equation (12) simply becomes:

$$f''_{i,j} = \frac{f_{i+1,j} - 2f_{ij} + f_{i-1,j}}{\Delta x^2} + \frac{f_{i,j+1} - 2f_{ij} + f_{i,j-1}}{\Delta z^2} \tag{13}$$

Now, the problem must be constructed such that equation (13) can be used to iteratively converge to a solution of the Prandtl-Glauert Equation. In this investigation, the Prandtl-Glauert Equation is used to solve for the perturbation potential in a flow field with a circular-arc airfoil at zero angle of attack. To set up the problem, the geometry of the airfoil must be modeled. The equation for a circular arc profile is as follows:

$$(Z - z_c)^2 = R^2 - (x - x_c)^2 \tag{14}$$

Substituting the equations for the x and z coordinates at the center in terms of R, c, and t, and non-dimensionalizing all terms with respect to the chord length yielded:

$$Z = \pm 2\tau \left[\frac{1}{4} - (x - \frac{1}{2})^2\right] \tag{15}$$

For this problem, the flow is symmetric about the x-axis. Therefore, only one half of the airfoil needs to be considered to solve the Prandtl-Glauert equation. Hence, only the positive terms of Z were used. The problem was solved for two different transonic similarity parameter values. The transonic similarity parameter is described as:

$$K = \frac{(1-M_\infty^2)}{(M_\infty^2 \tau)^{2/3}} \tag{16}$$

The two parameter values were 1.3 and 3. For both cases, the problem was solved for a Mach number of 0.85. For most airfoils, this is above the critical Mach number. Meaning, the flow will speed up to a supersonic velocity at some point on the surface of the airfoil. However, a K value of 3 corresponds to subsonic flow, while a K value of 1.3 corresponds to transonic flow. This is possible as the dimensionless thickness is dependent on the similarity parameter. Hence, a K value of 3 corresponds to a very thin airfoil, allowing for the flow to remain subsonic. With an assumed Mach number and given similarity parameter, the dimensionless thickness was solved for as:

$$\tau = \sqrt[2/3]{\frac{(1-M_\infty^2)}{K}} \tag{17}$$

Refer to Figure 3.1 below.

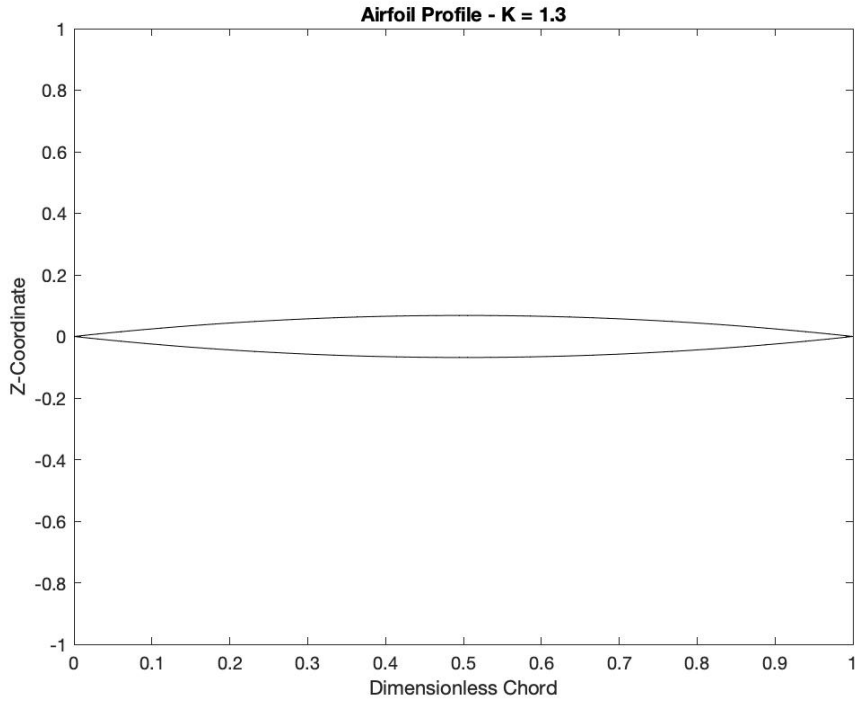


Figure 3.1: Airfoil Geometry – K = 1.3

The perturbation potential was calculated using central differencing at each point in a 100 by 100 mesh. To do so, an iterative method was used to converge to a solution. This means that an initial guess was placed for each

point in the mesh. Then, the code used the initial guess to calculate the current value of phi at each point. The new values of perturbation potential become the new initial guess. This occurs in a loop until the error between previous and current iteration falls below some threshold. This method is known as the Jacobi Iteration Method and was used to calculate the value of perturbation potential for the middle cells of the mesh. To use the Jacobi Method, the second derivatives shown in Equation (13) were substituted into the Prandtl-Glauert Equation. This yielded:

$$0 = (1 - M_\infty^2)\frac{\varphi_{i+1,j} - 2\varphi_{ij} + \varphi_{i-1,j}}{\Delta x^2} + \frac{\varphi_{i,j+1} - 2\varphi_{ij} + \varphi_{ij-1}}{\Delta z^2} \tag{18}$$

Rearranging to solve for the current perturbation potential value using the previous perturbation potential values behind and ahead of the point of interest yielded:

$$\beta = \sqrt{(1 - M_\infty^2)} \tag{19}$$

$$\varphi_{ij} = \beta^2\left(\frac{\hat{\varphi}_{i+1,j} + \hat{\varphi}_{i-1,j}}{\Delta x^2} + \frac{\hat{\varphi}_{i,j+1} + \hat{\varphi}_{ij-1}}{\Delta z^2}\right)\left(\frac{2\beta^2}{\Delta x^2} + \frac{2}{\Delta z^2}\right)^2 \tag{20}$$

This method works for the middle cells. However, boundary conditions must be specified for the perturbation potential to be calculated at every point. When sufficiently far from the disturbance of the airfoil, it can be assumed that the gradient is equal to zero. In addition, steady flow is assumed for the entire domain. The assumption that the gradient is zero implies that the derivative of phi with respect to x at the left and right-hand side of the mesh must also be zero. This is essentially defining "ghost cells", as the potential value at the first and third point are equal, allowing for the potential at the second point to be calculated. Refer to Figure 3.2 below.

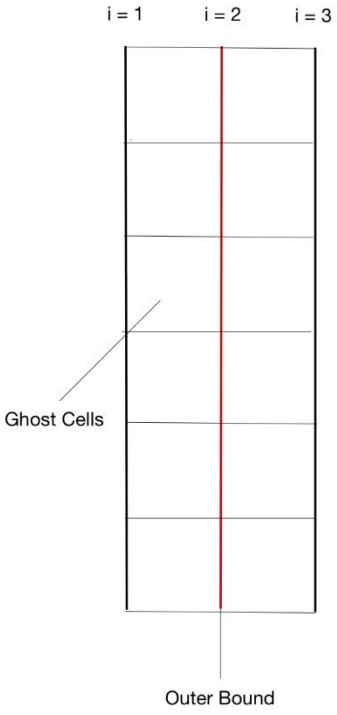


Figure 3.2: Definition of Boundary Conditions

This allows for the boundary conditions at the left and right hand-side to be written as:

$$\frac{\partial \varphi}{\partial x} = 0 \tag{21}$$

$$\frac{\varphi_{3,j}-\varphi_{1,j}}{2\Delta x}=0 \tag{22}$$

$$\varphi_{3,j}=\varphi_{1,j} \tag{23}$$

$$\varphi_{I,j}=\varphi_{I-2,j} \tag{24}$$

This allows for the Central difference to operate at the end points of the mesh even though there are no points outside the mesh to be used. The same method can be applied to the top of the grid, yielding the boundary condition:

$$\varphi_{i,J}=\varphi_{i,J-2} \tag{25}$$

The boundary condition for the bottom, however, is more complex as the airfoil is located at the bottom of the grid. Therefore, the above assumptions can be applied to points on the bottom of the grid only if the points are not located within the bounds of the airfoil. However, this condition is applied because of continuity. The bottom of the grid is an axis of symmetry; therefore, it can be assumed that the gradient is zero, but for a different reason. When the point of interest is within the bounds of the airfoil, a different condition must be applied. To specify this condition, the slope of the airfoil was determined as function of the x-coordinate. Taking the derivative of equation (6) yielded:

$$Z'=2\tau(1-2x) \tag{26}$$

When within the bounds of the airfoil, the gradient is not zero, rather, it is equal to the slope of the airfoil. Therefore, the gradient can be written as:

$$\frac{\partial\varphi}{\partial z}=\frac{\partial Z}{\partial x}=\frac{\varphi_{i,3}-\varphi_{i,1}}{2\Delta z} \tag{27}$$

After rearranging equation (27), the boundary condition becomes:

$$\varphi_{i,1}=\begin{cases}\varphi_{i,3} & off\ body\\ \varphi_{i,3}-2\Delta z Z' & on\ body\end{cases} \tag{28}$$

Using these boundary conditions, it is possible for a numerical algorithm to calculate the value of phi at every point in the grid. A series of if statements were constructed for the code to recognize when the boundary conditions applied. This was done by creating a vector ranging from -0.5 to 1.5 with a 0.02 length spacing for both x and z. Inside a while loop conditioned upon the error, it was possible for the code to check whether its current x or z coordinate meets one of the requirements for a boundary condition. If it does not, equation (20) is used. When the perturbation potential is calculated at each point in the space, the initial guess is updated with the current solution. Inside the while loop, the maximum difference between the current and previous iteration is calculated. When the error falls below some threshold, which is sufficiently small, the while loop terminates. With this method, a 100 by 100 perturbation potential field was calculated for both the subsonic and transonic case. Refer to Figure 3.3 below:

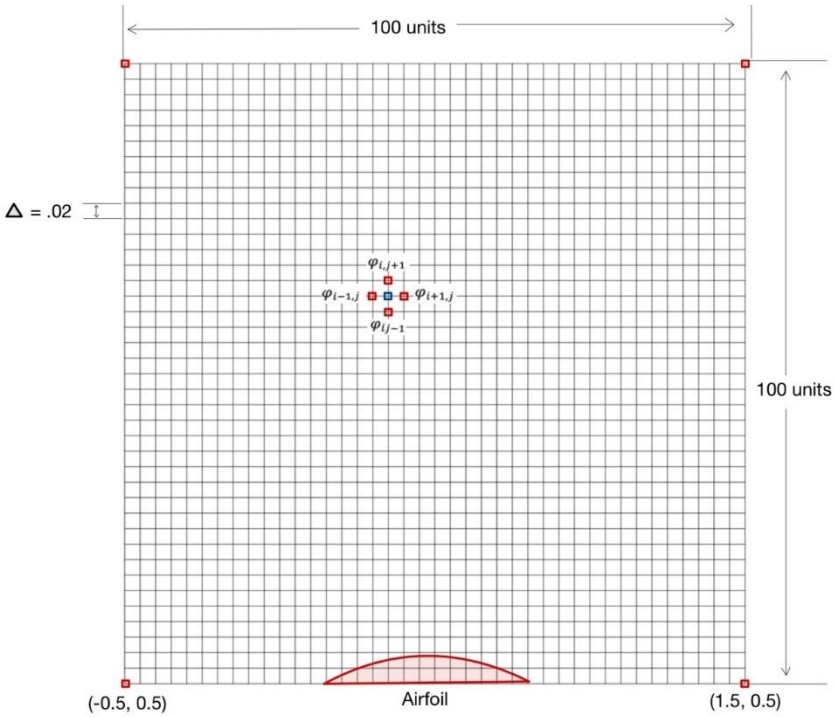


Figure 3.3: Mesh for Numerical Method Solution

The parameter space shown in Figure 3.2 was chosen as the effects of the airfoil disturbance will be fully captured. The space is large enough that the flow is unaffected at the outermost bounds. In addition, it is far more efficient to analyze only half of the airfoil. It should be noted that this process takes many iterations. In addition, it is not possible to converge to an exact solution, although the error may be exceedingly small. There are iteration methods more efficient than Jacobi, however, this method is sufficient for the problem at hand.

There were many assumptions made to create this numerical solution. The airfoil is assumed to be thin. However, for the larger transonic similarity parameter, the airfoil is quite thick. This means that its geometry extends farther than one cell in the z-direction. This is even though the boundary condition is only applied to the bottom-most points. This could cause somewhat inaccurate results. In addition, the numerical method solves the subsonic Prandtl-Glauert equation. However, the flow is transonic for one of the cases. This will cause quite inaccurate results as the solution is not capable of capturing the effects of shock waves. However, this error is understood and will be discussed in the next section.

It is possible to calculate the perturbation velocity in the x-direction using the perturbation potential. The formula for horizontal perturbation velocity is as follows:

$$u_{i,i} = \frac{\partial \varphi}{\partial x} \tag{29}$$

$$u_{i,j} = \frac{\varphi_{i+1,j}+\varphi_{i-1,j}}{\Delta x} \tag{30}$$

The perturbation velocity was calculated using equation (30) at the location of the airfoil (j = 2) from x-coordinates of 0 to 1. These are the x-coordinates in which the airfoil lies. Given the assumption of a thin airfoil, it was only necessary to calculate this value for the bottom-most grid points. Using this distribution of horizontal velocity, the dimensionless pressure coefficient was calculated as:

$$C_{p,i,j} = \frac{-2u_{i,j}}{\tau^{2/3}} \tag{31}$$

Next, the drag coefficient was calculated for each case. This calculation was performed as follows:

$$C_d = c_x \cos(\alpha) + c_z \sin(\alpha) \tag{32}$$

The angle of attack is zero degrees; therefore, equation (32) becomes:

$$C_d = c_x \tag{33}$$

$$c_x = \int C_p \, dz \tag{34}$$

As the airfoil is symmetric and only half the airfoil was analyzed, the result for drag coefficient can be multiplied by two. This yielded:

$$C_d = 2 \int C_p \, dz \tag{35}$$

Lastly, the critical pressure coefficient was calculated using the upstream Mach number as follows:

$$C_{p,crit} = \frac{1}{\tau^{2/3}} \left(\frac{2}{\gamma M_\infty^2} \left(\frac{2\left(1+\frac{\gamma-1}{2}M_\infty^2\right)}{\gamma+1} \right)^{\frac{\gamma}{\gamma-1}} - 1 \right) \tag{36}$$

In addition, data for the pressure coefficient calculated by the Transonic Small Disturbance Equation was used to calculate the value of drag coefficient using equation (35) for comparison.

IV. Results

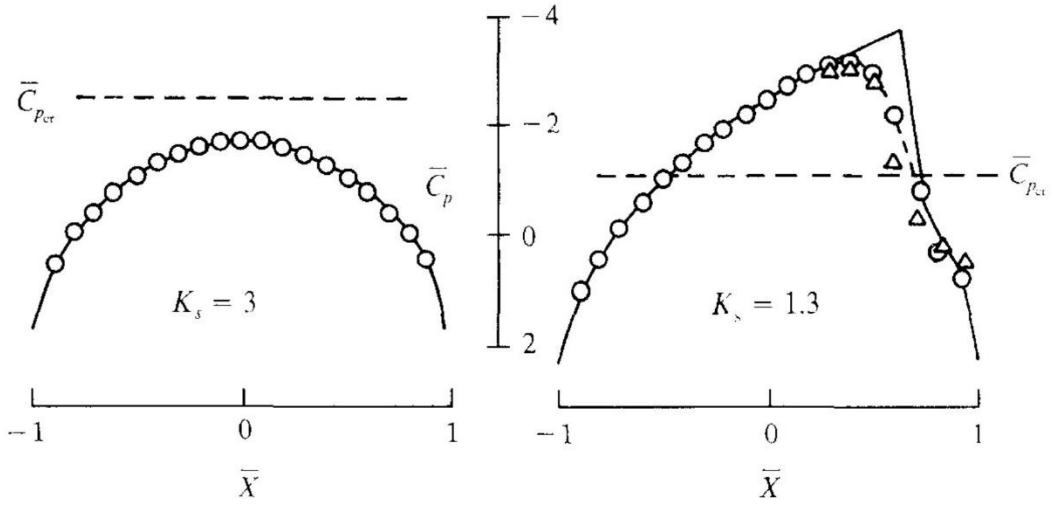


Figure 14.7, Modern Compressible Flow (John Anderson, p. 515).

Figure 4.1: Transonic Solution with Comparison to Experimental Data

A. Pressure Coefficient Distribution for Subsonic Case

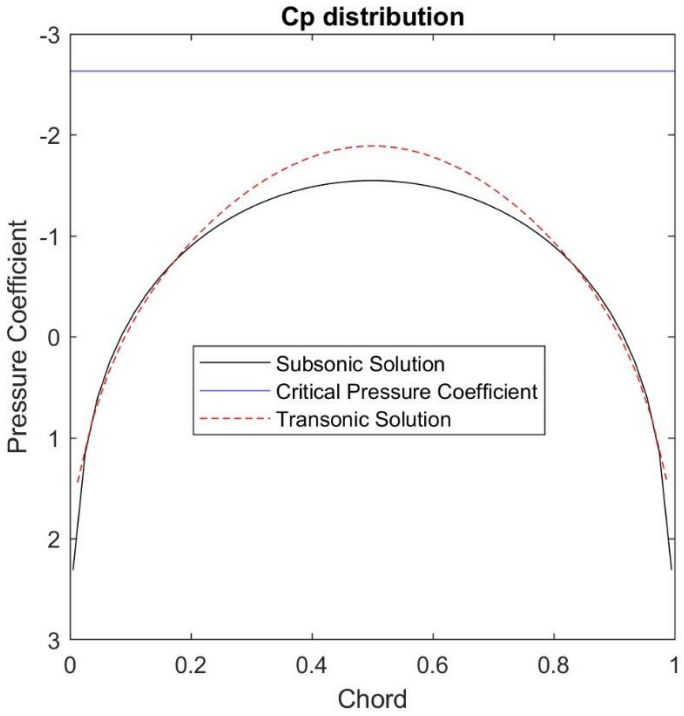


Figure 4.2: Pressure Coefficient Distribution for K = 3

Both pressure coefficient distributions are entirely symmetrical. The flow is subsonic everywhere and the airfoil geometry is symmetrical. Therefore, symmetry should be expected in the pressure distribution. Moreover, it can be seen that the critical pressure coefficient is never achieved. This implies that the flow did remain entirely subsonic. The solutions produced by both methods produced quite similar results. It should be expected, however, that the results are exactly the same. The flow is entirely subsonic, therefore, there should be no discrepancy. However, the results from the Transonic Small Disturbance Equation produced a steeper slope. The maximum (negative) pressure coefficient is slightly higher than the subsonic small disturbance solution approximated. Both solutions are approximations, hence, the numerical methods used to produce a solution will not produce the same results.

The transonic solution should be trusted more. Referring to Figure 4.1, the experimental data aligns perfectly with the approximation. Although the flow remains subsonic, compressibility effects are quite important at a Mach number of 0.85. The transonic solution captures these compressibility effects far more accurately. This is why it can produce results that align exactly with experimental data. The transonic solution better captures the variation of flow speed along the surface of the airfoil, which corresponds to a different variation of pressure coefficient. At higher subsonic speeds, the pressure coefficient will be more negative. The subsonic small disturbance equation essentially accounts for compressibility by scaling the incompressibility by a factor related to the Mach number. This will change the magnitude of the pressure coefficients, but not the manner in which the pressure varies along the surface. This is a linear approximation. For this reason, the subsonic small disturbance solution produces a slightly flatter, less steep pressure distribution. These differences account for the discrepancy between the two solutions.

It should be noted that the Mach number was chosen to be remarkably close to one to allow the transonic similarity parameter to be valid when compared to the transonic small disturbance solution. At a much lower Mach number, the subsonic solution would be far more accurate. As discussed above, the solution used scales the incompressible solution by some factor. This makes the pressure distribution very dependent on the airfoil geometry. Had a lower Mach number been used with the same similarity parameter, a much thicker airfoil would be analyzed. With the high Mach number, the airfoil is very thin. This causes the flow to not vary in velocity along the surface as much as it would for a thick airfoil. Hence, the pressure distribution is not as steep. The transonic solution captures the variation of Mach number due to compressibility effects in addition to the geometry of the airfoil. For these reasons, the transonic small disturbance solution produced more accurate results. Despite this, the numerical method used is still valid and produced results that could be quite useful.

B. Pressure Coefficient Distribution for Transonic Case

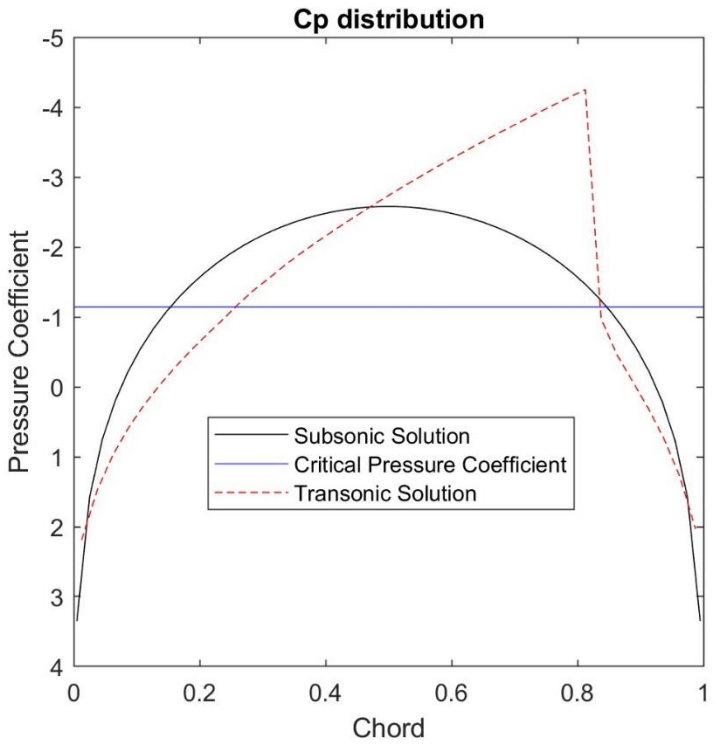


Figure 4.3: Pressure Coefficient Distribution for K = 1.3

For the similarity parameter pertaining to the transonic case, the discrepancy between the two solutions becomes far more prevalent. The transonic similarity parameter of 1.3 with a Mach number of 0.85 corresponds to a far thicker airfoil than the previous case. Hence, the flow will become supersonic along the surface of the airfoil. This can be seen as for both solutions, the pressure coefficient reaches a more negative value than the critical pressure coefficient. As was discussed above, the subsonic small disturbance solution scales the incompressible solution by a factor. This makes the pressure coefficient distribution dependent upon the upstream Mach number and the thickness of the airfoil. In this case, the Mach number is the same, but the airfoil is far thicker. This results in a larger magnitude of pressure coefficient, but the same general distribution. The pressure distribution is entirely symmetric as the compressibility effects cannot be captured. Hence, it becomes abundantly clear that the numerical method used to achieve this solution is only valid for subsonic flow. The linear approximation is entirely incapable of capturing the phenomena that occur in transonic flow.

As discussed in the introduction, the flow reaches supersonic speeds along the surface of the airfoil. The point where supersonic speeds are achieved is where the pressure distribution reaches the critical pressure

coefficient. The transonic small disturbance solution approximates this to be around one quarter of the chord. However, the flow is subsonic near the trailing edge of the airfoil. Therefore, a normal shock must occur for the flow to slow to subsonic speeds. The point at which subsonic speeds are achieved again is the second intersection of the critical pressure coefficient. This is around 0.8 chord and aligns perfectly with the location of the normal shock. This is also the approximate location of the shock found in the *Modern Compressible Flow* (John Anderson, 2021) solution. It is noticeably clear that the numerical method using the transonic small disturbance is far more capable of capturing the effects of compressibility and shock waves. The subsonic solution should not be trusted to predict behavior in transonic flow. However, the numerical algorithm used to solve the equation is still useful and can be implemented in solving the Transonic Small Disturbance Equation.

It is important to create a numerical method that aligns with the physics of the problem. This is one of the reasons that the solution from the subsonic method cannot be trusted. When the flow becomes supersonic, it becomes impossible for points behind the point of interest to influence the flow. This is because the flow is traveling faster than the speed of a small disturbance. Therefore, a central differencing method no longer makes sense. This is conveyed in the entirely inaccurate results. Instead, an upwind differencing method would need to be implemented. This means that only points ahead of the point of interest are used in the numerical method. This is how the transonic solution operates when the local Mach number is greater than one. Consequently, the results are far more accurate. Understanding how the numerical method operates and matching it to physics is crucial.

Referring to Figure 4.1, the transonic small disturbance approximation does not exactly align with the experimental data. This is in contrast to the subsonic case. As discussed in the introduction, transonic flow is exceedingly difficult to model. Although this approximation is meant to capture these effects, it still cannot do so perfectly. There were many assumptions made in the development of the numerical solution. However, the approximation was still remarkably accurate. The solution located the shock and captured its effects very well. Although there is no analytic solution to the Transonic Small Disturbance Equation, the numerical method proves to be a useful tool in modeling the behavior of transonic flow.

C. Drag Coefficient

	Subsonic Case (K=3)	**Transonic Case (K = 1.3)**
Subsonic Solution	$5.3663 \cdot 10^{-6}$	$1.2331 \cdot 10^{-5}$
Transonic Solution	$7.4956 \cdot 10^{-5}$	0.1676

Table 4.1: Drag Coefficient Values for Subsonic and Transonic Solution

For the subsonic case, both solutions produced drag coefficients that are essentially zero. Referring to Figure 4.2, this would be expected. Despite the small discrepancy between the two curves, both plots are symmetric. This makes sense as the airfoil is symmetric, and the flow remains subsonic. Both solution methods assume that the flow is steady and inviscid. Therefore, the only drag should be caused by lift and wave drag. The flow is subsonic, so wave drag is not a factor. Hence, the small disturbance approximation should return a drag result of about zero. The subsonic small disturbance solution is scaling the incompressible solution by a factor related to the Mach number. Incompressible thin airfoil theory always assumes zero drag at zero angle of attack. Since the solution only scales the incompressible solution, the drag is also zero here. In a more mathematical sense, the symmetric airfoil and pressure distribution will result in near exact cancellation when integrated over the surface of the airfoil.

For the transonic case, the subsonic small disturbance approximation still returns a drag coefficient near zero. This is for the same reasons discussed above. The solution simply scales the incompressible solution. Therefore, it is incapable of capturing the effects of shocks. This means that despite a supersonic Mach number, this solution will always return zero drag for a symmetrical airfoil at zero angle of attack. However, as discussed in the introduction, the shock will produce significant amounts of wave drag. As the transonic small disturbance

approximation can capture the normal shock, it returns a significantly larger drag coefficient. This is despite the fact that viscous forces are neglected as they are minor relative to the wave drag.

C. Numerical Method Design Discussion

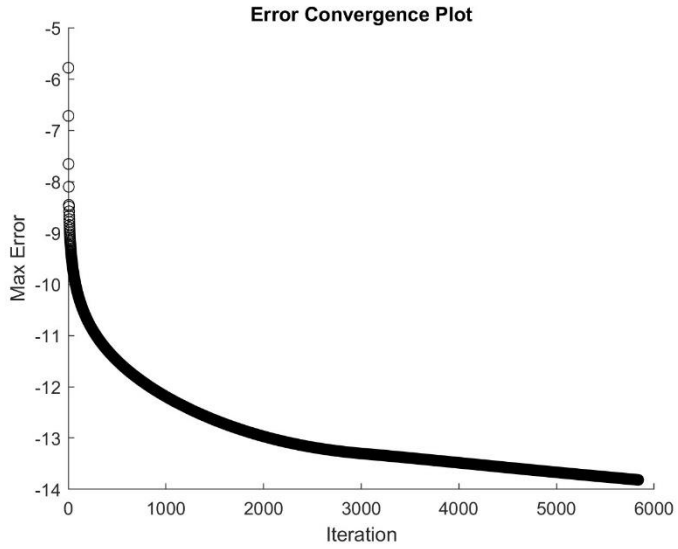


Figure 4.4: Maximum Error vs Iteration Number for K = 3

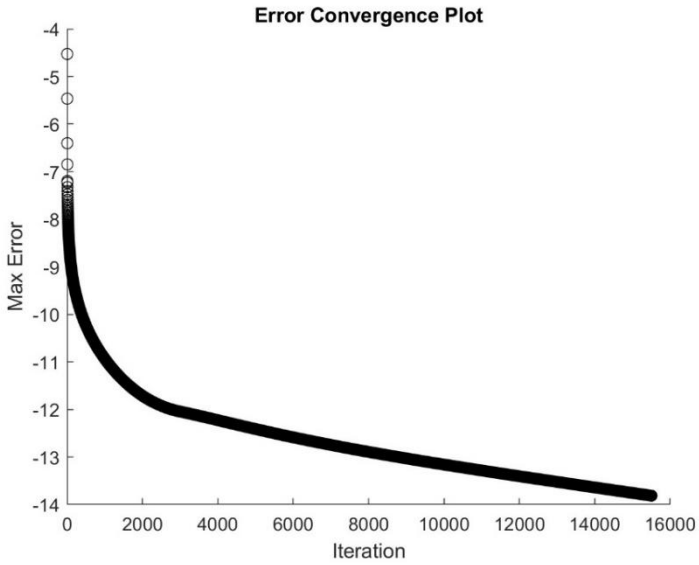


Figure 4.5: Maximum Error vs Iteration Number for K = 1.3

Referring to Figures 4.4 and 4.5, both numerical solutions converged to an error below 10^{-6}. (Note that the log of the error is plotted). The subsonic case was able to do this quite efficiently, in about 6,000 iterations. However, the transonic case required far more iterations to converge at just over 15,000 iterations. For both cases, a small disturbance equation is being solved. However, as discussed above, the value of K affects the thickness of the airfoil. This means that for a similarity parameter of 1.3, the airfoil is much thicker. Therefore, the disturbance is not

necessarily small. The larger disturbance results in a larger discrepancy between the previous and current iteration. As the error condition is equivalent for both cases, the transonic case requires far more iterations to converge.

In addition, the Jacobi iteration method was used to solve both cases. This involves using only previous iteration values to solve for the current iteration. There are other numerical methods, such as Gauss Seidel, that use a combination of the previous and current iteration to reach a solution. This can result in a far more efficient numerical iteration method that requires fewer iterations. However, the subsonic small disturbance equation is quite simple, and the Jacobi iteration method was sufficient. Despite the substantial number of iterations, the numerical method was designed in a manner that still allowed the algorithm to run very quickly. Hence, it was determined that another method of optimizing the algorithm was not necessary.

V. Conclusions

When compared to the experimental data found in *Modern Compressible Flow* (John Anderson, 2021), it was found that the numerical method used to solve the Prandtl-Glauert equation was quite accurate. The approximation had some error, but it captured the behavior of flow well. Had the upstream Mach number been lower, the numerical method would have been more accurate as the compressibility effects would be less relevant. It is important to understand that the Prandtl-Glauert Equation is a linear approximation. It operates by scaling the incompressible solution by a factor related to the Mach number. Hence, in transonic flow, the equation entirely falls apart. Due to the nature of this equation and the symmetric, zero angle of attack airfoil, the numerical method produced symmetrical pressure distributions for both cases. This symmetrical pressure distribution resulted in zero drag. This is a valid approximation at low Mach numbers, however, entirely invalid at transonic Mach numbers. This is due to the effects of the shock on the total drag.

The solution in *Modern Compressible Flow* (John Anderson, 2021) shows the formation of a shock on the airfoil. As a shock is present, it is no longer possible for the pressure distribution to be symmetric. Therefore, the drag does not integrate out to zero. This is in contrast to the solution that the Prandtl-Glauert approximation produced. Viscous drag is neglected; however, the total drag is dominated by wave drag. Wave drag cannot be captured at all with the subsonic small disturbance approximation. Nevertheless, the numerical method could be useful in solving the Transonic Small Disturbance Equation, which proved to provide accurate results even for complex transonic flow. The results produced by the numerical method make sense when the approximations are understood. The Prandtl-Glauert Equation is a linear approximation. Therefore, as it only scales the incompressible solution, the pressure distribution will always be symmetric for a symmetric airfoil at zero angle of attack. This will result in zero drag as well. While this is a valid solution for subsonic flow, it is not for transonic flow.

Unlike the Prandtl-Glauert Equation, the Transonic Small Disturbance Equation is not a linear equation. There is no analytical solution to this problem. This means that numerical methods are required to model transonic flow. Although the Prandtl-Glauert Equation is far simpler, the numerical method used to solve the equation is analogous to one that could be used to solve the Transonic Small Disturbance Equation. By modeling a potential field around the airfoil, the aerodynamic forces on an airfoil could be solved for in transonic flow. This is done through a numerical algorithm utilizing an initial guess that updates until a solution is found. It is not possible to solve for these forces directly, therefore, utilizing perturbations to deduce the forces is necessary. This is only possible with numerical methods. With increasing complexity in engineering, understanding numerical methods is vital in the modeling of aerodynamic scenarios. In addition, as technology evolves, the scenarios in which numerical methods can capture will evolve as well. The importance of computational fluid dynamics will only increase with time.

Of course, this numerical method was only an approximation. The results are not entirely accurate. There are many ways in which this experiment could be improved for future study. The most obvious improvement is to implement the numerical method to solve the Transonic Small Disturbance Equation rather than the subsonic equation. However, with this implementation comes increasing complexity. As discussed previously, the Jacobi iteration method is not the most efficient. A new numerical method that utilizes both previous and current iterations in its solution could be used to increase efficiency. In addition, the boundary conditions were quite simple in this investigation. A future numerical method would implement boundary conditions that account for non-symmetric airfoils at different angles of attack. The boundary conditions used also assumed a thin airfoil. New boundary

conditions could be constructed to better represent the airfoil geometry and its effect on the perturbation potential. This would also require the solutions space to be edited. With an optimized numerical method, it would be possible to refine the size of the Mesh to solve for the perturbation potential at far more points. This of course could drastically increase the accuracy for more advanced models. Despite possible improvements, the Jacobi iteration method proves to be a useful tool in computation fluid dynamics and could be implemented and a wide range of models.

VI. References

[1] *High speed wind tunnel HSWT - lockheed martin*. (n.d.). Retrieved April 2, 2023, from https://lockheedmartin.com/content/dam/lockheed-martin/mfc/documents/high-speed-wind-tunnel-hswt/mfc-hswt-handbook.pdf

[2] NASA. (n.d.). *SP-367 Introduction to the Aerodynamics of Flight*. NASA. Retrieved April 23, 2023, from https://history.nasa.gov/SP-367/chapt5.htm

[3] *Transonic aerodynamics and shock waves*. aerodyn. (n.d.). Retrieved April 23, 2023, from https://aerodyn.org/shock/

[4] Anderson, J. D. (2021). *Modern compressible flow: With historical perspective*. McGraw-Hill.

[5] *Arizona State University*. TSDEdata1_3.txt. (n.d.). Retrieved April 23, 2023, from https://canvas.asu.edu/courses/143786/files/65281954?module_item_id=10656744

[6] *Arizona State University*. TSDEdata3.txt. (n.d.). Retrieved April 23, 2023, from https://canvas.asu.edu/courses/143786/files/65281975?module_item_id=10656747

VII. Appendices

A. Tabular Results

Appendix A-1: Transonic Solution Pressure Coefficient Data

Dimensionless Chord - K = 1.3	Pressure Coefficient – K = 1.3	Dimensionless Chord - K = 3	Pressure Coefficient – K = 3
0.0125	2.19209278935814	0.0125	1.44230731206533
0.0375	1.47539669359966	0.0375	0.811371247144665
0.0625	0.970813379250219	0.0625	0.371469391969161
0.0875	0.578642269494631	0.0875	0.0359300564384446
0.1125	0.250773593346509	0.1125	-0.237819250213733
0.1375	-0.0363329372931487	0.1375	-0.470993884469744
0.1625	-0.295215728389295	0.1625	-0.674849591384987
0.1875	-0.533386546827477	0.1875	-0.855844171176201

0.2125	-0.756048046254918	0.2125	-1.01794160999590
0.2375	-0.968104997365340	0.2375	-1.16368425631110
0.2625	-1.19179636121896	0.2625	-1.29472918248842
0.2875	-1.39428765971645	0.2875	-1.41213946867681
0.3125	-1.57011487157502	0.3125	-1.51655586506990
0.3375	-1.75442337289034	0.3375	-1.60830701210772
0.3625	-1.92151677540904	0.3625	-1.68748784371511
0.3875	-2.07966085163942	0.3875	-1.75402265683984
0.4125	-2.23267777394456	0.4125	-1.80772231172128
0.4375	-2.38137365529488	0.4375	-1.84834020436235
0.4625	-2.52604625719278	0.4625	-1.87562740074810
0.4875	-2.66705226269083	0.4875	-1.88938339056854
0.5125	-2.80469727523454	0.5125	-1.88949607964574
0.5375	-2.93917022097222	0.5375	-1.87596410408558
0.5625	-3.07059630458401	0.5625	-1.84889694078390
0.5875	-3.19910772237448	0.5875	-1.80849278737884
0.6125	-3.32487223871493	0.6125	-1.75499869481268
0.6375	-3.44808047567962	0.6375	-1.68865985456150
0.6625	-3.56891705389175	0.6625	-1.60966443337848
0.6875	-3.68753686006541	0.6875	-1.51808752455918
0.7125	-3.80405488383580	0.7125	-1.41383385371768
0.7375	-3.91854785630815	0.7375	-1.29657461739985
0.7625	-4.03106146909756	0.7625	-1.16566900014440

0.7875	-4.14161622430955	0.7875	-1.02005388886665
0.8125	-4.25016128984379	0.8125	-0.858072156509160
0.8375	-0.979107300470980	0.8375	-0.677181328926129
0.8625	-0.463196180845200	0.8625	-0.473417165873835
0.8875	-0.0795380043439975	0.8875	-0.240321379673829
0.9125	0.307427519328654	0.9125	0.0333627886899723
0.9375	0.742329492530045	0.9375	0.368853172178572
0.9625	1.28182395103828	0.9625	0.808728901709175
0.9875	2.03132847792900	0.9875	1.43967836979958

Appendix A-2: Subsonic Pressure Coefficient Data from MATLAB Algorithm

Dimensionless Chord - K = 1.3	**Pressure Coefficient – K = 1.3**	**Dimensionless Chord - K = 3**	**Pressure Coefficient – K = 3**
0.0051	3.3505	0.0051	2.3099
0.0253	1.5764	0.0253	1.1454
0.0455	0.7512	0.0455	0.6055
0.0657	0.2048	0.0657	0.2490
0.0859	-0.2099	0.0859	-0.0209
0.1061	-0.5453	0.1061	-0.2387
0.1263	-0.8264	0.1263	-0.4209
0.1465	-1.0674	0.1465	-0.5768
0.1667	-1.2769	0.1667	-0.7122
0.1869	-1.4611	0.1869	-0.8309
0.2071	-1.6239	0.2071	-0.9358
0.2273	-1.7684	0.2273	-1.0287

0.2475	-1.8969	0.2475	-1.1113
0.2677	-2.0111	0.2677	-1.1846
0.2879	-2.1124	0.2879	-1.2495
0.3081	-2.2020	0.3081	-1.3069
0.3283	-2.2806	0.3283	-1.3572
0.3485	-2.3491	0.3485	-1.4010
0.3687	-2.4079	0.3687	-1.4386
0.3889	-2.4575	0.3889	-1.4703
0.4091	-2.4984	0.4091	-1.4964
0.4293	-2.5307	0.4293	-1.5171
0.4495	-2.5548	0.4495	-1.5324
0.4697	-2.5708	0.4697	-1.5426
0.4899	-2.5787	0.4899	-1.5477
0.5101	-2.5787	0.5101	-1.5477
0.5303	-2.5708	0.5303	-1.5426
0.5505	-2.5548	0.5505	-1.5324
0.5707	-2.5308	0.5707	-1.5171
0.5909	-2.4984	0.5909	-1.4964
0.6111	-2.4576	0.6111	-1.4704
0.6313	-2.4079	0.6313	-1.4387
0.6515	-2.3491	0.6515	-1.4011
0.6717	-2.2807	0.6717	-1.3573
0.6919	-2.2020	0.6919	-1.3070

0.7121	-2.1125	0.7121	-1.2496
0.7323	-2.0111	0.7323	-1.1847
0.7525	-1.8969	0.7525	-1.1114
0.7727	-1.7684	0.7727	-1.0288
0.7929	-1.6239	0.7929	-0.9359
0.8131	-1.4612	0.8131	-0.8310
0.8333	-1.2770	0.8333	-0.7123
0.8535	-1.0675	0.8535	-0.5769
0.8737	-0.8265	0.8737	-0.4210
0.8939	-0.5454	0.8939	-0.2388
0.9141	-0.2100	0.9141	-0.0210
0.9343	0.2047	0.9343	0.2488
0.9545	0.7511	0.9545	0.6053
0.9747	1.5763	0.9747	1.1452
0.9949	3.3504	0.9949	2.3096

Appendix A-3: Drag Coefficient Results

	Drag Coefficient for Subsonic Case (K = 3)	**Drag Coefficient for Transonic Case (K = 1.3)**
Subsonic Solution	$5.3663 \cdot 10^{-6}$	$1.2331 \cdot 10^{-5}$
Transonic Solution	$7.4956 \cdot 10^{-5}$	0.1676

B. Sample Calculations

Appendix B-1: Calculation of β

$$\beta = \sqrt{(1 - M_{\infty}^2)} \tag{19}$$

$$\beta = \sqrt{1 - 0.85^2}$$

$$\beta = 0.52678$$

Appendix B-2: Calculation of Dimensionless Thickness

$$\tau = \sqrt[2/3]{\frac{(1-M_\infty^2)}{K}} \tag{17}$$

$$\tau = \sqrt[2/3]{\frac{(1-0.85^2)}{3}}$$

$$\tau = 0.02813$$

C. MATLAB Code

Appendix B-1: MATLAB Code for K = 1.3

```
%% Lab 4 High Speed Aero
% Blake Logan

clc;
clear all;
D1 = importdata('TSDEdata1_3.txt');
M = 0.85; K = 1.3;
x = linspace(-0.5,1.5,100);
z = linspace(-.5,1.5,100);
tau = (1/M^2)*nthroot(((1-M^2)/K),2/3);
zp = 2*tau.*(1-2.*x);
B = sqrt(1-M^2);
k = 0;
I = length(x);
J = length(z);
del = 0.02;
phi0 = zeros(I,J);
phi = zeros(I,J);
u = zeros(I);
e = [];
max_diff = 1;
emax = [];
max_diff_vec = [];
while max_diff > 1*10^-6

    emax = abs(phi - phi0);
    k = k+1;

    for i = 1:I
        for j =1:J
            if (x(i) == -0.5)
                phi(1,j) = phi(3,j);  %Left

            elseif (x(i) == 1.5)
                phi(I,j) = phi(I-2,j); %Right
```

```
            elseif z(j) == 1.5
                phi(i,J) = phi(i,J-2); %Top

            elseif (z(j) == -.5) && not((x(i) >= 0) && (x(i) <= 1))
                phi(i,1) = phi(i,3);

            elseif (z(j) == -.5) && ((x(i) >= 0) && (x(i) <= 1))
                phi(i,1) = phi(i,3) - 2.*del.*(2*tau.*(1-2.*x(i)));

            else
                phi(i,j) = ((B^2)*((phi0(i-1,j) + phi0(i+1,j))/(del^2)) +
((phi0(i,j+1) + phi0(i,j-1))/(del^2)))*((((2*B^2)/(del^2)) + (2/del^2))^(-1));

            end

        end
    end

    for i = 25:75
        u(i) = (phi(i+1,2) - phi(i-1,2))/(2*del);
    end

   max_diff = max(max(abs(phi - phi0)));  % Calculate the maximum difference
   max_diff_vec = [max_diff_vec, max_diff];

    phi0 = phi;
end

kplot = 1:1:k;

figure(1)
scatter(kplot,log(max_diff_vec),'k');
xlabel('Iteration');
ylabel('Max Error');
title('Error Convergence Plot');

ex2 = D1.data(:,1);
cp2 = D1.data(:,2);
uplot = u(26:75);
u1 = u(26:75);
ex = x(26:75);
explot = linspace(0,1,48);
cpplot = (-2.*uplot)./(tau^(2/3));
cp1 = (-2.*u1)./(tau^(2/3));
z = (2*tau)*(.25 - (ex - .5).^2);
cd = 2*(trapz(z,cp1));

z2 = (2*tau)*(.25 - (ex2 - .5).^2);

cd2 = 2*(trapz(z2,cp2));
cpcrit = ((2./(1.4.*M.^2)).*( ((2.*(1+(.4/2).*M.^2))./(2.4)).^(1.4/.4)-
1))/(tau^(2/3));

figure(2)
plot(ex,cpplot,'k');
```

```
yline(cpcrit,'b');
hold on
plot(ex2,cp2,'r--');
hold off
axis ij
xlabel('Chord');
ylabel('Pressure Coefficient');
title('Cp distribution');
legend('Subsonic Solution','Critical Pressure Coefficient','Transonic Solution');
axis square

xx = linspace(0,1,100);
zx = (2*tau)*(.25 - (xx - .5).^2);

figure(3)

plot(xx,zx,'k');
hold on
plot(xx, -zx,'k');
hold off
xlim([0 1]);
ylim([-1 1]);
xlabel('Dimensionless Chord');
ylabel('Z-Coordinate');
title('Airfoil Profile - K = 3');
```

Appendix B-2: MATLAB Code for K = 3

```
%% Lab 4 High Speed Aero
% Blake Logan

clc;
clear all;
D1 = importdata('TSDEdata3.txt');
M = 0.85; K = 3;
x = linspace(-0.5,1.5,100);
z = linspace(-.5,1.5,100);
tau = (1/M^2)*nthroot(((1-M^2)/K),2/3);
zp = 2*tau.*(1-2.*x);
B = sqrt(1-M^2);
k = 0;
I = length(x);
J = length(z);
del = 0.02;
phi0 = zeros(I,J);
phi = zeros(I,J);
u = zeros(I);
e = [];
max_diff = 1;
```

```
emax = [];
max_diff_vec = [];
while max_diff > 1*10^-6

    emax = abs(phi - phi0);
    k = k+1;

    for i = 1:I
        for j =1:J
            if (x(i) == -0.5)
                phi(1,j) = phi(3,j);  %Left

            elseif (x(i) == 1.5)
                phi(I,j) = phi(I-2,j); %Right

            elseif z(j) == 1.5
                phi(i,J) = phi(i,J-2); %Top

            elseif (z(j) == -.5) && not((x(i) >= 0) && (x(i) <= 1))
                phi(i,1) = phi(i,3);

            elseif (z(j) == -.5) && ((x(i) >= 0) && (x(i) <= 1))
                phi(i,1) = phi(i,3) - 2.*del.*(2*tau.*(1-2.*x(i)));

            else
                phi(i,j) = ((B^2)*((phi0(i-1,j) + phi0(i+1,j))/(del^2)) +
((phi0(i,j+1) + phi0(i,j-1))/(del^2)))*((((2*B^2)/(del^2)) + (2/del^2))^(-1));

            end

        end
    end

    for i = 25:75
        u(i) = (phi(i+1,2) - phi(i-1,2))/(2*del);
    end

   max_diff = max(max(abs(phi - phi0)));  % Calculate the maximum difference
   max_diff_vec = [max_diff_vec, max_diff];

    phi0 = phi;
end

kplot = 1:1:k;

figure(1)
scatter(kplot,log(max_diff_vec),'k');
xlabel('Iteration');
ylabel('Max Error');
title('Error Convergence Plot');

ex2 = D1.data(:,1);
cp2 = D1.data(:,2);
uplot = u(26:75);
u1 = u(26:75);
```

```
ex = x(26:75);
explot = linspace(0,1,48);
cpplot = (-2.*uplot)./(tau^(2/3));
cp1 = (-2.*u1)./(tau^(2/3));
z = (2*tau)*(.25 - (ex - .5).^2);
cd = 2*(trapz(z,cp1));

z2 = (2*tau)*(.25 - (ex2 - .5).^2);

cd2 = 2*(trapz(z2,cp2));
cpcrit = ((2./(1.4.*M.^2)).*( ((2.*(1+(.4/2).*M.^2))./(2.4)).^(1.4/.4)-
1))/(tau^(2/3));

figure(2)
plot(ex,cpplot,'k');
yline(cpcrit,'b');
hold on
plot(ex2,cp2,'r--');
hold off
axis ij
xlabel('Chord');
ylabel('Pressure Coefficient');
title('Cp distribution');
legend('Subsonic Solution','Critical Pressure Coefficient','Transonic Solution');
axis square

xx = linspace(0,1,100);
zx = (2*tau)*(.25 - (xx - .5).^2);

figure(3)

plot(xx,zx,'k');
hold on
plot(xx, -zx,'k');
hold off
xlim([0 1]);
ylim([-1 1]);
xlabel('Dimensionless Chord');
ylabel('Z-Coordinate');
title('Airfoil Profile - K = 3');
```