UNIVERSITY OF COLORADO - BOULDER

ASEN 3802: AEROSPACE SCIENCES LABORATORY II

MAY 2, 2024

Lab 3: Aerodynamics

Author:
LUKAS WRIGHT

Author:
BRADEN NELSON

Author:
JARED STEFFEN

Instructor:
Samantha Sheppard

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO BOULDER

I. Introduction

In this lab, we meticulously compare the vortex panel method with thin airfoil theory and experimental NACA data. We focus on how airfoil thickness and camber influence the zero-lift angle of attack and sectional lift coefficient. Additionally, we analyze the wing performance of a Cessna 150 using Prandtl Lifting Line Theory and the vortex panel method. We aim to gain insights into aerodynamic behavior for improved aircraft design and performance optimization.

II. Methodology

In order to analyze airfoils via the vortex panel method for questions 1-3, a MATLAB function was developed that could output the x and y coordinates of the airfoil surface given the inputs of the NACA airfoil naming parameters maximum camber (m), location of maximum camber (p), and thickness (t), as well as the chord length (c) and the number of desired panels to use in the vortex panel method. This function calculates the thickness distribution from the mean camber line via the equation:

$$y_t = \frac{t}{0.2}c\left[0.2969\sqrt{\frac{x}{c}} - 0.1260\left(\frac{x}{c}\right) - 0.3516\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1026\left(\frac{x}{c}\right)^4\right]$$

as well as the formula for the mean camber line via the equation:

$$y_c = \begin{cases} m\frac{x}{p^2}\left(2p - \frac{x}{c}\right) & 0 \le x < pc \\ m\frac{c-x}{(1-p)^2}\left(1 + \frac{x}{c} - 2p\right) & pc \le x \le c \end{cases}$$

The coordinates from these equations for the upper (x_U and y_U) and the lower (x_L and y_L) can then be determined from:

$$\xi = \arctan\frac{dy_c}{dx}$$
$$x_U = x - y_t \sin\xi$$
$$x_L = x + y_t \sin\xi$$
$$y_U = y_c + y_t \cos\xi$$
$$y_L = y_c - y_t \cos\xi$$

Which can then be concatenated as $[x_L x_U]$ and $[y_L y_U]$ to receive the x and y coordinates of the whole airfoil surface. The x and y coordinate outputs could then be used in the vortex panel method function provided by Kuethe and Chow from their textbook [1]. These results are then compared to experimental data from Abbot and Doenhoff's collection of NACA data [2] as well as thin airfoil theory.

For questions 4 and 5, another MATLAB function had to be developed that could output the span efficiency factor, and coefficients of lift and induced drag given the wing span, number of odd terms included in the summation series (N), as well as root and tip lift slopes, zero-lift angles of attack, geometric angles of attack, and chord lengths. It also linearly interpolates the values between the root and tip of the wing, so it assumes the wing is trapezoidal in shape. This function utilizes Prandtl Lifting Line Theory in order to achieve these goals:

$$\alpha(\theta) = \frac{4b}{\alpha_0(\theta)c(\theta)}\sum_{n=1}^{\infty} A_n \sin n\theta + \alpha_{L=0}(\theta) + \sum_{n=1}^{\infty} nA_n \frac{\sin n\theta}{\sin\theta}$$

where:

$$\theta_i = \frac{i\pi}{2N}$$
$$i = 1, 2, ..., N$$

and truncating the even terms such that:

$$\Gamma(\theta) = 2bV_\infty \sum_{j=1}^{N} A_{(2j-1)} \sin\left((2j-1)\theta\right)$$

III. Results

A. Problem 1: Computation of the Lift Generated by a Thick Symmetric Airfoil

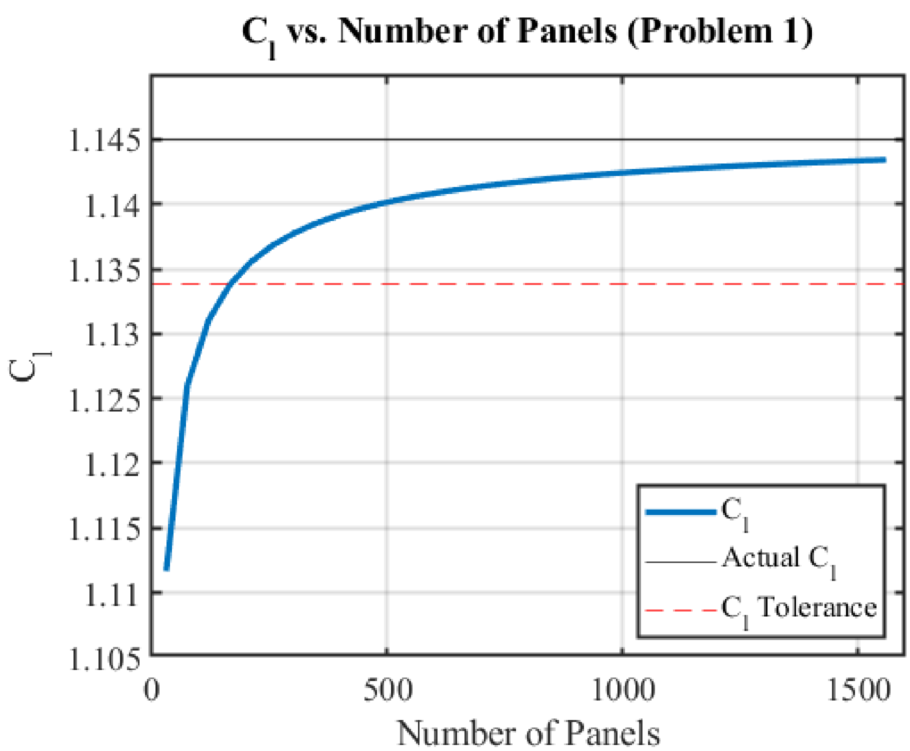

Fig. 1 Sectional Lift Coefficient Convergence

Problem 1 aims to determine the number of panels to accurately predict the sectional lift coefficient of a NACA 0006 airfoil at $\alpha = 10^o$ within 1% error. In order to find the actual sectional lift coefficient, the NACA Airfoil function developed was ran with 5000 panels, which had an output $c_l = 1.145$. In order to determine the minimum umber of panels needed to achieve a result within 1% of this, an if statement nested inside of a for loop ran the NACA Airfoils function, increasing the number of panels 45 at a time until the error between the calculated value with that number of panels was less than 1% of the actual value recorded at 5000 panels. The result came out to be 165 panels can accurately predict the sectional lift coefficient to be 1.137 for a NACA 0006 airfoil at $\alpha = 10^o$. The results for this test can be seen in Figure 1.

B. Problem 2: Study of the Effect of Airfoil Thickness on Lift

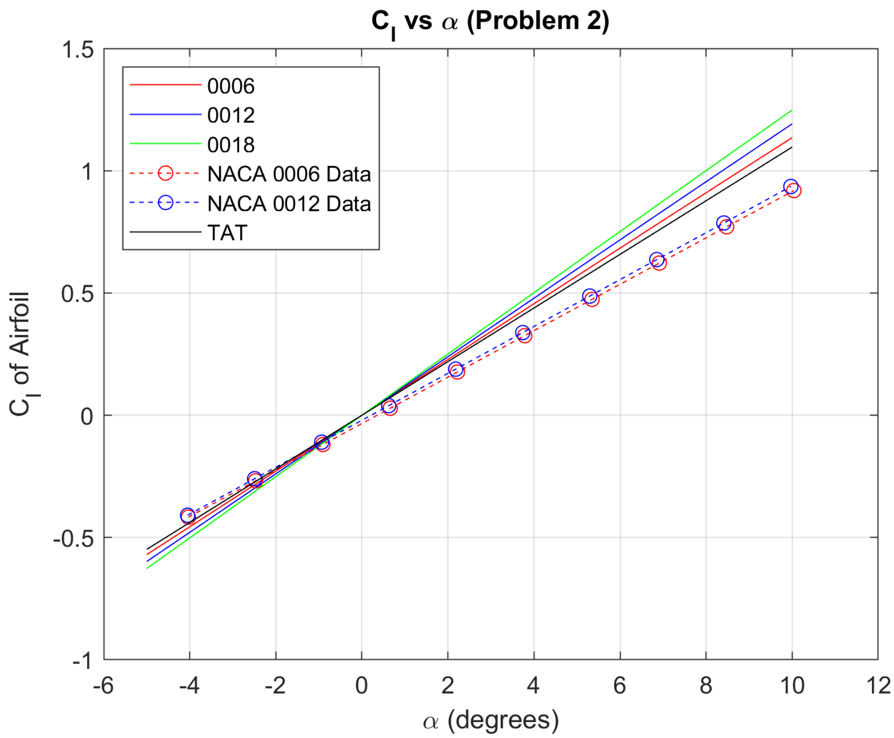

Fig. 2 Sectional Lift Coefficient vs Angle of Attack for Symmetric Airfoils with Varying Thickness

The second problem states to determine the effect of airfoil thickness on the sectional lift coefficient and compare it to thin airfoil theory and experimental NACA data. The three airfoils analyzed in this problem were NACA 0006 (thin airfoil), NACA 0012 (moderately thick airfoil), and NACA 0018 (thick airfoil), however the data for the NACA 0018 airfoil was not included in Abbot and Doenhoff's collection of NACA data, so only the NACA 0006 and NACA 0012 experimental data was compared. The NACA 0006 airfoil, the analytical results yielded $\alpha_{L=0} = 0^o$ and a lift slope of 0.113, the NACA 0012 yielded the same $\alpha_{L=0}$ and a lift slope of 0.119, and the NACA 0018 also yielded the same $\alpha_{L=0}$ and a lift slope of 0.125. Thin airfoil theory says that for a symmetric airfoil, $\alpha_{L=0} = 0$ and a lift slope of 0.109. Experimental NACA results yield a $\alpha_{L=0} = 0$ for the NACA 0006 and NACA 0012 airfoils as well as lift slopes of 0.0834 and 0.0957 respectively. Please note that the experimental data was pulled off of the chart visually using a web plot digitizer and is subjected to larger error, as the lift slopes for all of the following airfoils is known to be 2π. The collective results are tabulated below in Tables 1-3 and can be seen in Figure 2.

Table 1 Effect of Thickness on Lift Results NACA 0006

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	0	0	0
Lift Slope [$\frac{1}{rad}$]	4.778	6.474	6.245

Table 2 Effect of Thickness on Lift Results NACA 0012

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	0	0	0
Lift Slope [$\frac{1}{rad}$]	5.48	6.818	6.245

Table 3 Effect of Thickness on Lift Results NACA 0018

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	–	0	0
Lift Slope [$\frac{1}{rad}$]	–	7.162	6.245

C. Problem 3: Study of the Effect of Airfoil Camber on Lift

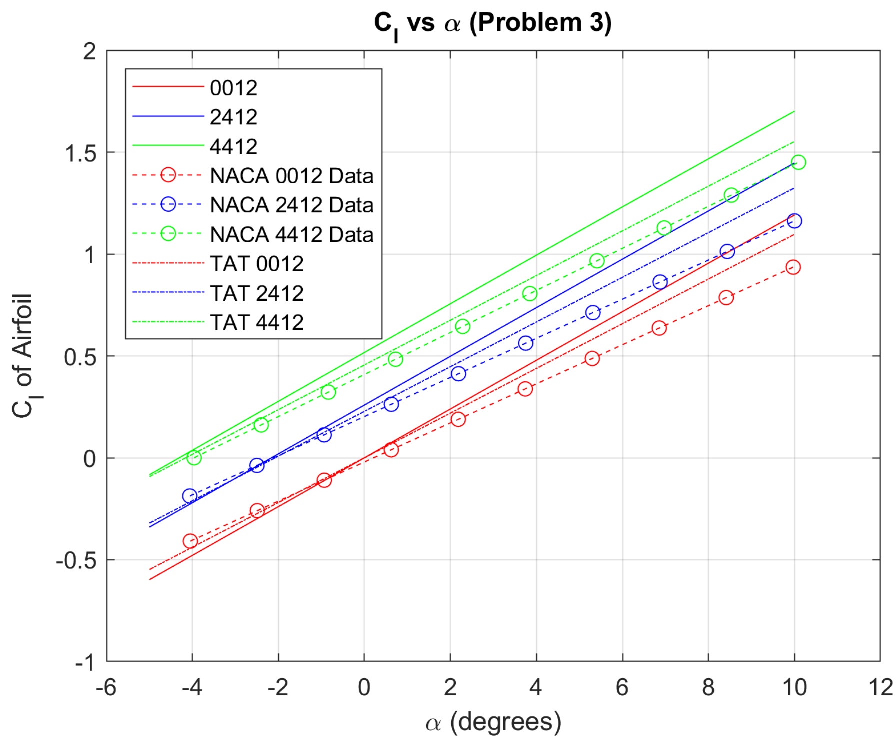

Fig. 3 Sectional Lift Coefficient vs Angle of Attack for Airfoils with Varying Camber

Problem 3 seeks to establish the effect of camber on the sectional lift coefficient of airfoils with the same thickness and compare it to experimental results and thin airfoil theory of cambered airfoils. In this section, the airfoils being analyzed and compared are the NACA 0012 (no camber), NACA 2412 (moderatly cambered), and NACA 4412 (highly cambered) airfoils. The methodology is largely the same as problem 2, however, the zero-lift angle of attack for cambered airfoils is no longer zero, but is determined by the equation:

$$\alpha_{L=0} = -\frac{1}{\pi}\int_0^{\pi} \frac{dy_c}{dx}(1-\cos\theta)d\theta$$

The derivation of $\frac{dy_c}{dx}$ can be found in Appendix A. The results for this derivation were combined with the equation above and utilized within MATLAB to integrate over the wingspan and determine the zero-lift angles of attack for each cambered airfoil. The results for the NACA 0012 airfoil and the thin airfoil theory corresponding to it are the same as in problem 2. For the NACA 2412 airfoil, the analytical data gave $\alpha_{L=0} = -2.12^o$ and a lift slope of 0.119, the experimental data gave $\alpha_{L=0} = -2^o$ and a lift slope of 0.096, and the thin airfoil theory gave $\alpha_{L=0} = -2.07^o$ and a lift slope of 0.109. For the NACA 4412 airfoil, the analytical data gave $\alpha_{L=0} = -4.24^o$ and a lift slope of 0.118, the experimental data gave $\alpha_{L=0} = -4^o$ and a lift slope of 0.091, and the thin airfoil theory gave $\alpha_{L=0} = -4.15^o$ and a lift slope of 0.109. Please note that the experimental data was pulled off visually using a web plot digitizer and is subjected to larger error, as the lift slopes for all of the following airfoils is known to be 2π. The results from each airfoil are tabulated in Tables 4-6 below and can be seen in Figure 3.

Table 4 Effect of Camber on Lift Results NACA 0012

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	0	0	0
Lift Slope [$\frac{1}{rad}$]	5.48	6.818	6.245

Table 5 Effect of Camber on Lift Results NACA 2412

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	-2	-2.12	-2.07
Lift Slope [$\frac{1}{rad}$]	5.500	6.818	6.245

Table 6 Effect of Camber on Lift Results NACA 4412

	Experimental	Analytical	Thin Airfoil Theory
$\alpha_{L=0}$ [degrees]	-4.24	-4	-4.15
Lift Slope [$\frac{1}{rad}$]	5.214	6.818	6.245

D. Problem 4: Prandtl Lifting Line Theory

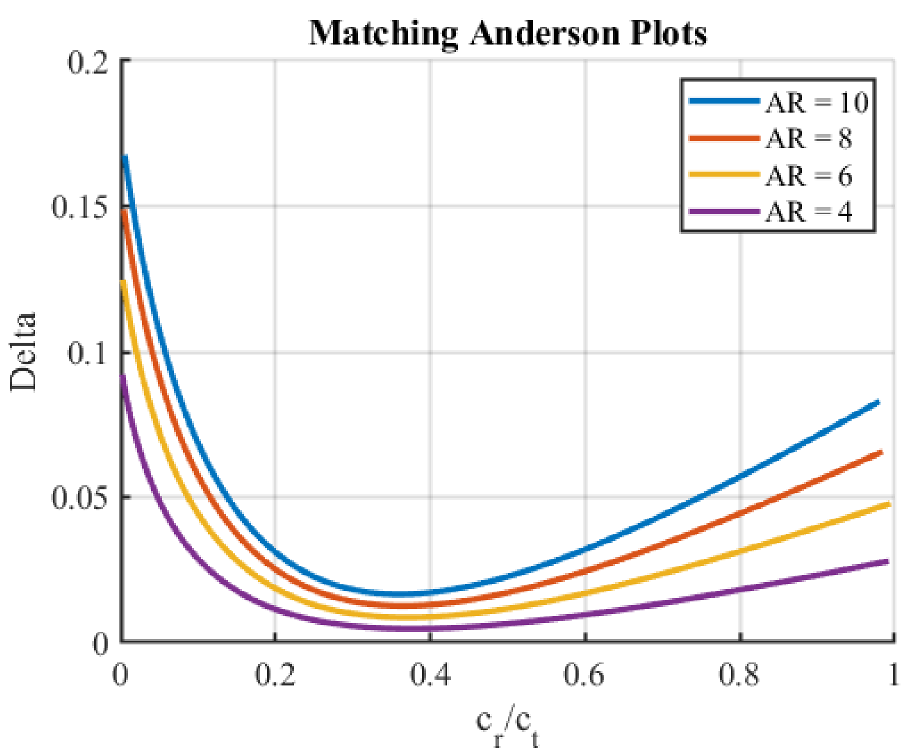

Fig. 4 Recreation of Plot from Anderson of δ vs Taper Ratio for Various Aspect Ratios

Problem 4 is the development and testing of the Prandtl Lifting Line Theory (PLLT) outlined in the Methodology section above. After the MATLAB function is developed, it is proven to work by redeveloping Figure 5.20 from Anderson' Fundamentals of Aerodynamics [3], which is shown in Figure 4.

E. Problem 5: Analysis of Approximate Cessna 150 Wing Performance

The fifth problem is actually using the PLLT code in order to analyze the aerodynamic efficiency of a Cessna 150. It is given that the wing has a wingspan of 32 ft 8 in, a root chord of 5ft 2 in with a NACA 2412 airfoil, and a tip chord of 3 ft 10 with a NACA 0012 airfoil. The PPLT function is then able to linearly interpolate any necessary values along the wingspan. The Cessna 150 is modeled to be flying at 85 knots at 10,000 ft altitude with an angle of attack of 4^o, and a geometric twist such that it varies between 1^o at the root and 0^o at the tip. It is assume to be flying in standard atmosphere conditions. The first section of problem 5 asks for the number of odd terms in the series expansion for circulation to achieve lift and induced drag within 10%, 1%, and 0.1% error. To achieve the converged value for lift and induced drag, a value for 1000 odd terms was used in the PLLT function, which yielded $L = 1,363 lb_f$ and $D_i = 31 lb_f$. The number of odd terms for lift and induce drag to achieve 10%, 1%, and 0.1% error can be seen in Figures 5 and 6 as well as Table 7.

Table 7 Number of Odd Terms for Lift and Induced Drag to Converge for Various Percent Errors

Percent Error	Lift – Number of Odd Terms	Induced Drag – Number of Odd Terms
10%	2	2
1%	5	6
0.1%	15	19

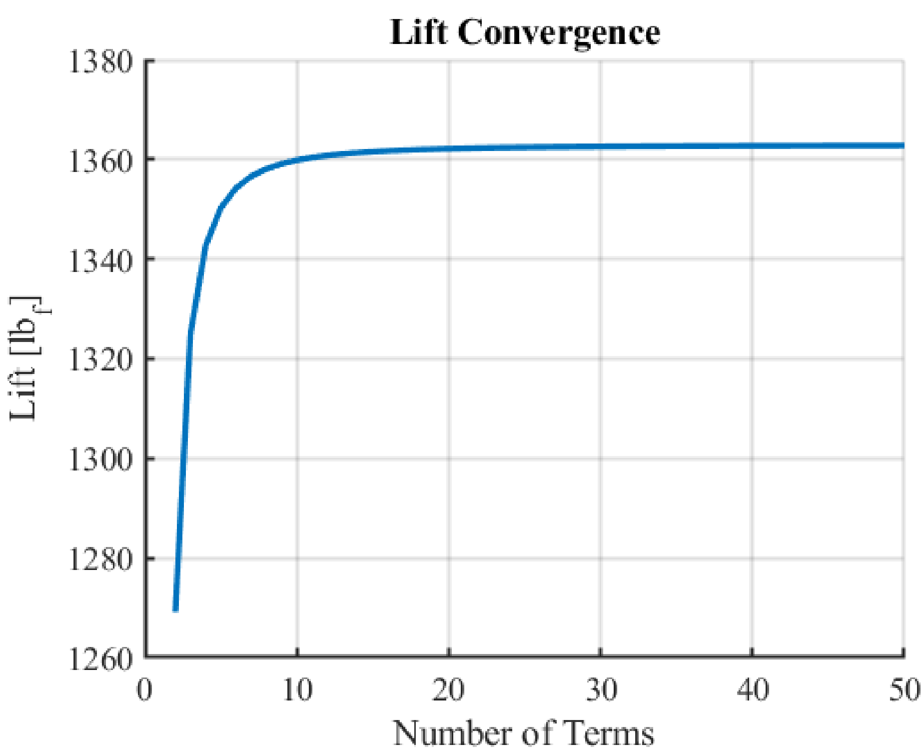

Fig. 5 Convergence of Coefficient of Lift

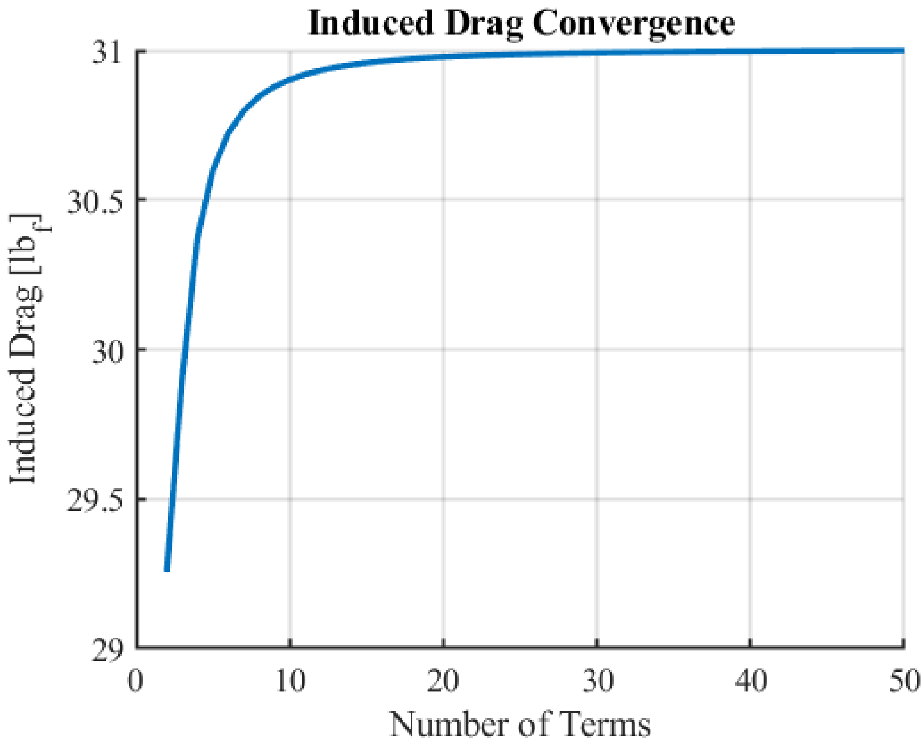

Fig. 6 Convergence of Coefficient of Induced Drag

The second part of problem 5 asks for the aerodynamic efficiency of the wing by giving the Lift-to-Drag ratio ($\frac{L}{D}$). In order to do this, the fact that the vortex panel method assumes inviscid flow must be address and corrected. In order to do this, the NACA charts from Abbot and Von Doenhoff were utilized once again to estimate c_d, the sectional drag coefficient. This was done by calculating the Reynolds Number of the given flight conditions, which comes out to be 3.2×10^6. This value is very close to one of the recorded values on the NACA chart of 3.0×10^6, so the data corresponding to that Reynolds Number was used. The value was read off of both the NACA 0012 and NACA 2412 charts and averaged between the two. Using a web digitzer, for a NACA 0012, $c_d = 0.006$ and for a NACA 2412, $c_d = 0.0065$, with an average $c_d = 0.00625$. The new coefficient of drag is now:

$$C_D = C_{Di} + c_d$$

Now, the Lift-to-Drag ratio was properly calculated using the highest number of panels that yielded a percent error of 0.1%. The Lift-to-Drag ratio then came out to be 28.61, meaning at the current flying conditions of the Cessna 150, the is $28.61x$ more lift and drag. The next step was then to provide a plot of how $\frac{L}{D}$ changes with angle of attack. In order to do this, a vector of the sectional drag coefficient had to be developed in order to have it change for the different sectional lift coefficients. To do this, three separate points were taken from the NACA charts using a web digitzer and then in MATLAB, a quadratic line was fit to those points. The average was then taken between the root and tip sectional drag coefficients for each point. This new vector was added to the induced drag vector and in order to get the total drag coefficient. For there, $\frac{L}{D}$ was calculated for a range of angles of attack and plotted. This plot can be seen in Figure 7.

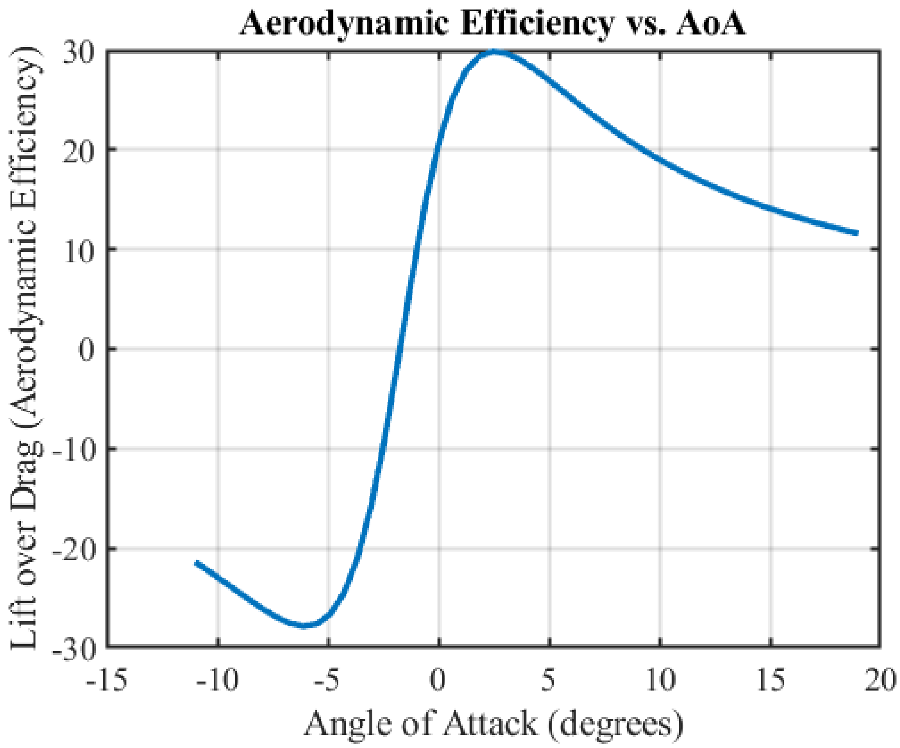

Fig. 7 Lift-to-Drag Ratio vs Angle of Attack

IV. Discussion

A. Problem 2: Study of the Effect of Airfoil Thickness on Lift

Zero-lift angle of attack is unaffected by airfoil thickness. This is to be expected, because in order to change $\alpha_{L=0}$ we need to induce lift at zero angle of attack. This requires a pressure difference between the upper and lower portions of the airfoil at $\alpha = 0$. Increasing thickness changes the profile of the top and bottom of the airfoil identically, and thus there is no pressure difference created on an uncambered airfoil regardless of thickness.

As we increase wing thickness, sectional lift slope a_0 increases. A thicker airfoil will have a greater arc-length along the airfoil, and as a result the air will need to travel faster along this section to satisfy continuity. This greater velocity causes a greater pressure difference between the upper and lower parts of the wing. This causes a greater lift force (and thus C_l) for the high-thickness airfoil at a given angle of attack, thus a higher a_0.

The assumption of thin airfoil theory is most accurate for an infinitely thin airfoil, and its validity diminishes as thickness increases. As such, the thinnest airfoil (NACA 0006) is the most accurate with thin airfoil theory. The intermediate-thickness NACA 0012 is less accurate, and the very thick NACA 0018 is the least accurate. This can be seen visually in Figure 2. The thinnest airfoil (NACA 0006) has the closest a_0 to thin airfoil theory, and the thickest airfoil (NACA 0018) has the furthest prediction from thin airfoil theory. This supports the conclusion that as thickness increases, thin airfoil theory becomes less accurate.

Thin airfoil theory is closer to the experimental data than the vortex panel method. This may seem counter-intuitive at first, but inspection of the innate assumptions in both methods explains this. Both the vortex panel method and thin airfoil theory rely on the assumption of inviscid flow. The experimental data has viscous effects, and consequently the lift slope is less than both of our theoretical results. Let's consider the main assumption that separates thin airfoil theory from the vortex panel method. Thin airfoil theory assumes an infinitely thin airfoil, while vortex panel method accounts for thickness. We concluded above that lift slope increases with increasing thickness. As such, thin airfoil theory provides us with a shallower lift slope than any of the vortex panel method. The decrease in a_0 due to the thickness assumption in thin airfoil theory slightly accounts for the viscous effects experienced in real life, and as such is a closer approximation than the vortex panel method. Note that this only holds for airfoils that are reasonably thin, as thin airfoil data breaks down and becomes nonsensical for sufficiently thick airfoils.

The experimental data from NACA was gathered at specific Reynolds numbers. Recall the definition of the Reynolds number, which is the ratio of momentum forces to viscous forces $Re = \frac{\rho V L}{\mu}$. Both of our experimental methods assumed inviscid flow, which gives us an effectively infinite Reynolds number.

It is clear from Figure 2 that both thin airfoil theory and the vortex panel method are overestimates of the experimental value. This is to be expected because they are both idealized models which neglect effects like viscosity that would reduce lift in practice. This is evident in our Reynolds number, which is infinite. This physically means that the effect of momentum forces are infinitely more prevalent than viscous forces, which we know not to be true. We also assume small angles, and at $\alpha = 10°$ this assumption is reasonable, but still a source for error.

B. Problem 3: Study of the Effect of Airfoil Camber on Lift

Contrary to thickness, camber affects $\alpha_{L=0}$ but does not affect lift slope, a_0. Camber affects $\alpha_{L=0}$ because it changes the shape of the upper and lower surfaces of the airfoil such that it is asymmetric at $\alpha = 0$. Since there is asymmetry at $\alpha = 0$, there must be a pressure difference and therefore $L \neq 0$ at $\alpha = 0$. If there is nonzero lift at $\alpha = 0$, then $\alpha_{L=0} \neq 0$, differing from the uncambered airfoils in part 2.

Camber is known to shift where $\alpha_{L=0}$ lies without having an effect on lift slope. Camber effectively changes the angle at which a certain pressure differential exists between the top and bottom surfaces of the airfoil. However, it has no effect on how much lift changes with respect to angle of attack. When we change the angle of attack of an airfoil, we are changing the pressure differential between the upper and lower surfaces. The rate at which this pressure difference changes with angle of attack is unaffected by camber. Camber can change at what angle a certain pressure difference occurs (and thus a certain value of C_l occurs), but it can not change how quickly lift increases with angle of attack (i.e. it can not change the slope of C_l w.r.t α).

Since all three airfoils have the same thickness, the thin airfoil assumption applies equally well to all of them. We concluded above that the validity of thin airfoil theory relies solely on thickness, not on camber. All three of these airfoils have the same thickness and thus are as accurate as each other.

As discussed in part 2, thin airfoil theory has a closer lift slope to the experimental data due to the inviscid flow assumption. Both the vortex panel method and thin airfoil theory assume inviscid flow, meaning that they will over-predict lift compared to experimental values. Since we don't account for thickness in thin airfoil theory but we do in vortex panel method, then the vortex panel method will predict greater lift than thin airfoil theory. This under-prediction of lift from thickness accounts for some of the over-prediction from inviscid flow. As such, thin airfoil theory is closer to the experimental results.

As in problem 2, we assume inviscid flow such that $\mu = 0$. This provides us with an infinitely large Reynolds number.

A source of error is that we assume inviscid flow, when that is not the case in reality. We also assume small angles of attack. At $\alpha = 10°$ this assumption is not detrimental but still takes away from the accuracy of our results.

C. Problem 4: Prandtl Lifting Line Code

Since we are analyzing steady flight, the lift distribution must be symmetrical on both wings to ensure the aircraft is flying in a straight line. Consider the sinusoidal waves that make up the terms of our infinite series. The even terms will span N full periods (where N is the term number A_N), which results in zero net contribution. This is because for any positive contribution, there is an equal and opposite value in the negative direction that cancels any net circulation. The odd terms will make $2N - 1$ full periods, plus a half period. As a result, there is always a component that is not canceled out and contributes to our net circulation. A visualization of this can be seen in Figure 8.

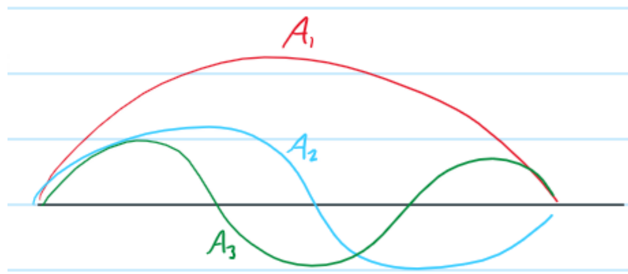

Fig. 8 Visualization of Contributions from Fourier Coefficients

A scenario in which we would include even and odd Fourier coefficients is when the aircraft has uneven loading on its wings, like a plane making a banked turn. The wing angled into the turn will have less lift than the wing on the outside of the turn. In this flight regime the even Fourier terms would not cancel themselves, and we would need to include them. We know that the most efficient wing geometry is an ellipse, and δ is a measure of how close a wing geometry is to reaching elliptical efficiency. The smaller the δ, the closer we are to approaching elliptical efficiency. Figure 4 shows that δ is minimized at $c_r/c_t \approx 0.35$. This implies that the most efficient geometry we examined has a taper ratio near $0.3 - 0.4$.This taper ratio minimizes δ for each aspect ratio we tested. However, it is also notable that a lower aspect ratio produces a smaller δ. As such, the condition we would design towards would be a taper ratio of $0.3 - 0.4$ and as small of an aspect ratio as we can manage.

D. Problem 5: Analysis of Approximate Cessna 150 Wing Performance

To make the wing more efficient, the ratio of lift to drag must increase. One method to increase aerodynamic efficiency would be to increase the aspect ratio of the wing. This causes the wing to lengthen and thin out, which may cause structural concerns. Despite this, the effects of increasing AR greatly aid in increasing aerodynamic efficiency, mainly through the reduction of induced drag. As the aspect ratio increases, the effect of the wingtip vortices is reduced because of the wing's geometry (long and thin). Reducing induced drag decreases the overall drag of the aircraft, which increases the L/D ratio. A very similar effect is seen when a geometric twist is added to the wing. By twisting the tip lower than the root, you decrease the angle of attack seen by the wing as you move away from the fuselage of the aircraft. Decreasing the angle of attack also decreases the coefficient of lift, and since induced drag is proportional to C_l^2, decreasing the lift at the tips greatly diminishes induced drag. Despite reducing the lift acting on the wing, the aerodynamic efficiency still increases because the overall drag decreases more than the lift. The root bending moment is also affected in a very similar sense. Since the root moment is greatly impacted by the forces acting farther away from the fuselage, any decrease in these forces aids in the reduction of moments at the root. Therefore, the root moment is decreased due to twisting the wing tip lower than the root.

V. Conclusions

In this lab we used the vortex panel method and thin airfoil theory to model the lift and drag of standard NACA airfoils. We analyzed the effect of thickness and camber on airfoil properties. Results showed that camber solely affects $\alpha_{L=0}$, and thickness solely affects a_0. After finding the results (C_l and $C_{D,i}$) from both methods, we compared to experimental data from NACA. We found that both of our models over-predicted lift and concluded that it is primarily due to viscous effects in the real system. We then created a function to employ Prandtl's Lifting Line Theory, and used it to analyze a Cessna 150. We calculated C_L and $C_{D,i}$, and used skin-friction drag C_D from NACA charts to calculate total drag. We then calculated aerodynamic efficiency, $\frac{L}{D}$, as a function of angle of attack and found that our peak aerodynamic efficiency is at $\alpha \approx 2.5°$.

VI. Appendices

A. Appendix A: Derivation(s)

1. Cambered Thin Airfoil Theory Derivation

Cambered Thin Airfoil Derivation

$$x = \frac{c}{2}(1-\cos\theta)$$

$$z_c = \begin{cases} \frac{mx}{p^2}\left(2p - \frac{x}{c}\right) & 0 \le x < pc \\ m\frac{c-x}{(1-p)^2}\left(1 + \frac{x}{c} - 2p\right) & pc \ge x \le c \end{cases}$$

evaluating B.C.s

$$x = 0 \rightarrow \theta = 0$$
$$x = c \rightarrow \theta = \pi$$
$$x = pc \rightarrow \quad pc = \frac{c}{2}(1-\cos\theta)$$
$$\theta = \cos^{-1}(1-2p)$$

Upper part of piecewise.

$$z_{c_1} = \frac{2mxp}{p^2} - \frac{mx^2}{cp^2}$$

$$= \frac{2mx}{p} - \frac{mx^2}{cp^2}$$

$$\frac{dz_{c_1}}{dx} = \frac{2m}{p} - \frac{2mx}{cp^2}$$

plug in $x = \frac{c}{2}(1-\cos\theta)$

$$= \frac{2m}{p} - \frac{\cancel{2}m\frac{\cancel{c}}{\cancel{2}}(1-\cos\theta)}{\cancel{c}p^2}$$

$$= \frac{2m}{p} - \frac{m - m\cos\theta}{p^2}$$

$$= \frac{2mp^{\cancel{2}} - \cancel{p}m + \cancel{p}m\cos\theta}{p\,\cancel{3}^{2}}$$

$$= \frac{2mp - m + m\cos\theta}{p^2}$$

$$= \frac{m(2p - 1 + \cos\theta)}{p^2}$$

Lower part of piecewise:

$$z_{c_2} = \frac{m(c-x)}{(1-p)^2} + \frac{x}{c}\frac{m(c-x)}{(1-p)^2} - 2p\frac{m(c-x)}{(1-p)^2}$$

$$= \frac{mc - mx}{(1-p)^2} + \frac{mcx - mx^2}{c(1-p)^2} - \frac{2pmc + 2pmx}{(1-p)^2}$$

$$\frac{dz_c}{dx} = \cancel{\frac{-m}{(1-p)^2}} + \cancel{\frac{m}{(1-p)^2}} - \frac{2mx}{c(1-p)^2} + \frac{2pm}{(1-p)^2}$$

$$= \frac{-2mx\cancel{(1-p)^2} + 2cpm\cancel{(1-p)^2}}{c(1-p)^{\cancel{4}}}$$

$$= \frac{-2mx + 2cpm}{c(1-p)^2}$$

plug in $x = \frac{c}{2}(1-\cos\theta)$

$$= \frac{-\cancel{2}m\left[\frac{\cancel{c}}{\cancel{2}}(1-\cos\theta)\right] + 2\cancel{c}pm}{\cancel{c}(1-p)^2}$$

$$= \frac{-m + m\cos\theta + 2pm}{(1-p)^2}$$

$$= \frac{m(2p - 1 + \cos\theta)}{(1-p)^2}$$

$$\frac{dz_c}{dx} = \begin{cases} \frac{m(2p-1+\cos\theta)}{p^2} & 0 \le \theta < \cos^{-1}(1-2p) \\ \frac{m(2p-1+\cos\theta)}{(1-p)^2} & \cos^{-1}(1-2p) \le \theta \le \pi \end{cases}$$

$$\alpha_{L=0} = -\frac{1}{\pi}\int_0^{\pi} \frac{dz_c}{dx}(1-\cos\theta)\,d\theta$$

Plug into MATLAB

Fig. 9

B. Appendix B: References

References

[1] Kuethe, A. M., *Foundations of Aerodynamics: Bases of Aerodynamic Design*, J Wiley, 2000.

[2] Abbot, I. H., *Theory of Wing Sections*, Dover Publications, Inc., 1959.

[3] Anderson, J., *Fundamentals of Aerodynamics 6th Edition*, McGraw Hill Education., 2017.

C. Appendix C: Code

```
%% HK
clc;
close all;
clear;

%% Code

alpha_0006 = 10; % degrees

[m_0006, p_0006, t_0006] = NACAdata('0006');
[m_0012, p_0012, t_0012] = NACAdata('0012');
[m_0018, p_0018, t_0018] = NACAdata('0018');
[m_2412, p_2412, t_2412] = NACAdata('2412');
[m_4412, p_4412, t_4412] = NACAdata('4412');

N = 30:45:1575; % Iteration number -- 1575 used in report image
c = 1;
allow = 0.01;
error = 1;
c_l_0006_actual = 1.145;    % Number retreived when we input N=5000 --> we assume this is the
    asymptotic c_l
cl_tolerance = 0;
c_l_0006 = ones(1,length(N));
TAT_slope = 2*pi^2 / 180;

for i=1:length(N)

[x_b_0006, y_b_0006] = NACA_Airfoils(m_0006,p_0006,t_0006,c,N(i)); % Finding x and y of airfoil
    for a given # of Panels (Problem 1)
[c_l_0006(i)] = Vortex_Panel(x_b_0006,y_b_0006,alpha_0006);      % Finding Cl given certain array
    of panels

error = 1 - (c_l_0006(i) / c_l_0006_actual);

if(error < allow) && cl_tolerance == 0             % Checking if error is below allowable
    cl_tolerance = c_l_0006(i);                    % taking the values of n and cl that give us
        tolerable error
    n_tolerance = N(i);
end

end

alpha_vary = linspace(-5,10,100); % varying alpha for problem 2 and 3
c_l_0006_2 = ones(1,length(alpha_vary));
c_l_0012 = ones(1,length(alpha_vary));
c_l_0018 = ones(1,length(alpha_vary)); % Preallocating for speed
c_l_2412 = ones(1,length(alpha_vary));
```

```
c_l_4412 = ones(1,length(alpha_vary));
TAT_y = zeros(1,length(alpha_vary));

for i=1:length(alpha_vary)

[x_b_0006_2, y_b_0006_2] = NACA_Airfoils(m_0006,p_0006,t_0006,c,n_tolerance); % Finding x and y of airfoil using pervious n panels (Problem 2)
[c_l_0006_2(i)] = Vortex_Panel(x_b_0006_2,y_b_0006_2,alpha_vary(i));     % Finding Cl given certain alpha

[x_b_0012, y_b_0012] = NACA_Airfoils(m_0012,p_0012,t_0012,c,n_tolerance); % Finding x and y of airfoil using pervious n panels (Problem 2)
[c_l_0012(i)] = Vortex_Panel(x_b_0012,y_b_0012,alpha_vary(i));

[x_b_0018, y_b_0018] = NACA_Airfoils(m_0018,p_0018,t_0018,c,n_tolerance); % Finding x and y of airfoil using pervious n panels (Problem 2)
[c_l_0018(i)] = Vortex_Panel(x_b_0018,y_b_0018,alpha_vary(i));

[x_b_2412, y_b_2412] = NACA_Airfoils(m_2412,p_2412,t_2412,c,n_tolerance); % Finding x and y of airfoil using pervious n panels (Problem 3)
[c_l_2412(i)] = Vortex_Panel(x_b_2412,y_b_2412,alpha_vary(i));

[x_b_4412, y_b_4412] = NACA_Airfoils(m_4412,p_4412,t_4412,c,n_tolerance); % Finding x and y of airfoil using pervious n panels (Problem 3)
[c_l_4412(i)] = Vortex_Panel(x_b_4412,y_b_4412,alpha_vary(i));

TAT_y(i) = TAT_slope * alpha_vary(i);

% Symmetric Airfoils (Problem 2 and 3)
if(abs(c_l_0006_2(i)) < 0.005)
   alpha_L0_0006_2 = alpha_vary(i);
end

if(abs(c_l_0012(i)) < 0.005)
   alpha_L0_0012 = alpha_vary(i);  % finding zero lift angle of attack
end

if(abs(c_l_0018(i)) < 0.005)
   alpha_L0_0018 = alpha_vary(i);
end
% Cambered airfoils (Problem 3)
if(abs(c_l_2412(i)) < 0.005)
   alpha_L0_2412 = alpha_vary(i);  % finding zero lift angle of attack
end

if(abs(c_l_4412(i)) < 0.01)
   alpha_L0_4412 = alpha_vary(i);
end

end

% Lift Slops (Problems 2 and 3)
p_cl0006_2 = polyfit(alpha_vary,c_l_0006_2,1);
p_cl0012 = polyfit(alpha_vary,c_l_0012,1);
p_cl0018 = polyfit(alpha_vary,c_l_0018,1);
p_cl2412 = polyfit(alpha_vary,c_l_2412,1);
p_cl4412 = polyfit(alpha_vary,c_l_4412,1);
p_TAT = polyfit(alpha_vary,TAT_y,1);
a_0_0006 = p_cl0006_2(1);
```

```
a_0_0012 = p_cl0012(1);
a_0_0018 = p_cl0018(1);
a_0_2412 = p_cl2412(1);
a_0_4412 = p_cl4412(1);
a_0_TAT = p_TAT(1);

a_0_0006_rad = a_0_0006 * 180/pi;
a_0_0012_rad = a_0_0012 * 180/pi;
a_0_0018_rad = a_0_0018 * 180/pi;
a_0_2412_rad = a_0_2412 * 180/pi;
a_0_4412_rad = a_0_4412 * 180/pi;

% alpha_L0_0006_2 = -p_cl0006_2(2) / p_cl0006_2(1);
% alpha_L0_0012 = -p_cl0012(2) / p_cl0012(1);
% alpha_L0_0018 = -p_cl0018(2) / p_cl0018(1);   % Alternate method
% alpha_L0_2412 = -p_cl2412(2) / p_cl2412(1);   % Uses polyfit
% alpha_L0_4412 = -p_cl4412(2) / p_cl4412(1);

alpha_L0_0006_2_rad = alpha_L0_0006_2 * pi/180;
alpha_L0_0012_rad = alpha_L0_0012 * pi/180;
alpha_L0_0018_rad = alpha_L0_0018 * pi/180;
alpha_L0_2412_rad = alpha_L0_2412 * pi/180;
alpha_L0_4412_rad = alpha_L0_4412 * pi/180;

% TAT For Cambered Airfoils
dzdx1_2412 = @(x) (m_2412 * (2*p_2412 -1 + cos(x)))/p_2412^2;
dzdx2_2412 = @(x) (m_2412 * (2*p_2412 -1 + cos(x)))/(1-p_2412)^2;
dzdx1_4412 = @(x) (m_4412 * (2*p_4412 -1 + cos(x)))/p_4412^2;
dzdx2_4412 = @(x) (m_4412 * (2*p_4412 -1 + cos(x)))/(1-p_4412)^2;

x_theta = @(x) (cos(x) -1);

int1_2412 = @(x) dzdx1_2412(x).*x_theta(x);
int2_2412 = @(x) dzdx2_2412(x).*x_theta(x);
int1_4412 = @(x) dzdx1_4412(x).*x_theta(x);
int2_4412 = @(x) dzdx2_4412(x).*x_theta(x);

integ_2412 = integral(int1_2412,0,acos(1-2*p_2412)) + integral(int2_2412,acos(1-2*p_2412),pi);
integ_4412 = integral(int1_4412,0,acos(1-2*p_4412)) + integral(int2_4412,acos(1-2*p_2412),pi);

alpha_L0_TAT_2412 = -(1/pi) * integ_2412*(180/pi);
alpha_L0_TAT_4412 = -(1/pi) * integ_4412*(180/pi);

TAT_2412 = (2*pi^2)/180 * (alpha_vary - alpha_L0_TAT_2412);
TAT_4412 = (2*pi^2)/180 * (alpha_vary - alpha_L0_TAT_4412);

p_cl2412_TAT = polyfit(alpha_vary,TAT_2412,1);
p_cl4412_TAT = polyfit(alpha_vary,TAT_4412,1);

a_0_2412_TAT = p_cl2412_TAT(1);
a_0_4412_TAT = p_cl4412_TAT(1);

% Theory of Wings Sections from Abbot and von Doenhoff Data (TWS)
TWS_0006x = linspace(-4.033,10.04,10);
TWS_0006y = linspace(-0.4165,0.919,10);
TWS_0012x = linspace(-4.05,9.97,10);
TWS_0012y = linspace(-0.4089,0.9355,10);
TWS_2412x = linspace(-4.0625,10,10);
TWS_2412y = linspace(-0.1875,1.1625,10);
```

```
TWS_4412x = linspace(-3.9626,10.093,10);
TWS_4412y = linspace(0,1.45,10);

% Problem 4 functions calls and variables
b=1;
a0_t = 2*pi;      % affects a0(theta)
a0_r = 2*pi;
c_t10 = 0.001;    % affects c(theta)
c_r10 = 0.199;
c_t8 = 0.001;     % affects c(theta)
c_r8 = 0.249;
c_t6 = 0.001;     % affects c(theta)
c_r6 = 0.332;
c_t4 = 0.001;     % affects c(theta)
c_r4 = 0.499;
aero_t = 0;       % affects a_L0(theta)
aero_r = 0;
geo_t = 4*pi/180; % affects a_geo(theta)
geo_r = 4*pi/180;
N2 = 50;

iterator10 = 1;
iterator8 = 1;
iterator6 = 1;
iterator4 = 1;

while(c_t10(iterator10)/c_r10(iterator10) <= 1)

[e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t10(iterator10),c_r10(iterator10),aero_t,aero_r,geo_t,geo_r,N2);
delta10(iterator10) = (1/e) - 1;
c_t10(iterator10 + 1) = c_t10(iterator10) + 0.001;
c_r10(iterator10 + 1) = c_r10(iterator10) - 0.001;
iterator10 = iterator10 + 1;

end

while(c_t8(iterator8)/c_r8(iterator8) <= 1)

[e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t8(iterator8),c_r8(iterator8),aero_t,aero_r,geo_t,geo_r,N2);
delta8(iterator8) = (1/e) - 1;
c_t8(iterator8 + 1) = c_t8(iterator8) + 0.001;
c_r8(iterator8 + 1) = c_r8(iterator8) - 0.001;
iterator8 = iterator8 + 1;

end

while(c_t6(iterator6)/c_r6(iterator6) <= 1)

[e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t6(iterator6),c_r6(iterator6),aero_t,aero_r,geo_t,geo_r,N2);
delta6(iterator6) = (1/e) - 1;
c_t6(iterator6 + 1) = c_t6(iterator6) + 0.001;
c_r6(iterator6 + 1) = c_r6(iterator6) - 0.001;
iterator6 = iterator6 + 1;

end

while(c_t4(iterator4)/c_r4(iterator4) <= 1)

[e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t4(iterator4),c_r4(iterator4),aero_t,aero_r,geo_t,geo_r,N2);
```

```
delta4(iterator4) = (1/e) - 1;
c_t4(iterator4 + 1) = c_t4(iterator4) + 0.001;
c_r4(iterator4 + 1) = c_r4(iterator4) - 0.001;
iterator4 = iterator4 + 1;

end

ct_cr10 = c_t10(1:end-1) ./ c_r10(1:end-1);
ct_cr8 = c_t8(1:end-1) ./ c_r8(1:end-1);   % Adjusting while loop vectors to match delta vec
ct_cr6 = c_t6(1:end-1) ./ c_r6(1:end-1);
ct_cr4 = c_t4(1:end-1) ./ c_r4(1:end-1);


% Problem 5
N3 = 2:50;
ct5 = 3 + 10/12;     % tip chord [ft]
cr5 = 5 + 2/12;      % root chord [ft]
b5 = 32 + 8/12;      % wingspan [ft]
geo_r5 = 5 * pi/180; % geometric AoA root [rad]
geo_t5 = 4 * pi/180; % geometric AoA tip [rad]
a0_r5 = a_0_2412_rad;    % a0 root
a0_t5 = a_0_0012_rad;    % a0 tip
aero_r5 = alpha_L0_2412_rad; % a_L0 root
aero_t5 = alpha_L0_0012_rad; % a_L0 tip
S5 = 0.5 * (ct5 + cr5) * b5;
rho5 = 17.56E-4;
mu5 = 3.534E-7;
V5 = 85 * 1.68780986; % knots to ft/s
cl_tolerance_PLLT2 = 0;
cDi_tolerance_PLLT2 = 0;
cl_tolerance_PLLT3 = 0;
cDi_tolerance_PLLT3 = 0;
cl_tolerance_PLLT4 = 0;
cDi_tolerance_PLLT4 = 0;
cL_SS = 0.513066830173821; % c_L steady state = 0.513066830173821
cDi_SS = 0.011671104290211; % c_Di steady state = 0.011671104290211

% Desired percent errors
allow2 = 0.1;
allow3 = 0.01;
allow4 = 0.001;

%[e00,c_L5,c_Di5] = PLLT(b5,a0_t5,a0_r5,ct5,cr5,aero_t5,aero_r5,geo_t5,geo_r5,1000);

for i=1:length(N3)

[~,c_L5,c_Di5] = PLLT(b5,a0_t5,a0_r5,ct5,cr5,aero_t5,aero_r5,geo_t5,geo_r5,N3(i));

% percent error
error_cL = abs(cL_SS - c_L5 )/ cL_SS;
error_cDi = abs(cDi_SS - c_Di5)/ cDi_SS;

if(error_cL < allow2) && cl_tolerance_PLLT2 == 0 % Checking if error is below allowable
    cl_tolerance_PLLT2 = c_L5;                                  % taking the values of n and cl that give us tolerable error
    n_tolerance2L = i+1;           % number of elements
end

if(error_cDi < allow2) && cDi_tolerance_PLLT2 == 0 % Checking if error is below allowable
```

```
    cDi_tolerance_PLLT2 = c_Di5;                                  % taking the values of n and cDi that give us tolerable error
    n_tolerance2Di = i+1;           % number of elements
end


if(error_cL < allow3) && cl_tolerance_PLLT3 == 0 % Checking if error is below allowable
    cl_tolerance_PLLT3 = c_L5;                                   % taking the values of n and cl that give us tolerable error
    n_tolerance3L = i+1;           % number of elements
end

if(error_cDi < allow3) && cDi_tolerance_PLLT3 == 0 % Checking if error is below allowable
    cDi_tolerance_PLLT3 = c_Di5;                                  % taking the values of n and cDi that give us tolerable error
    n_tolerance3Di = i+1;           % number of elements
end


if(error_cL < allow4) && cl_tolerance_PLLT4 == 0 % Checking if error is below allowable
    cl_tolerance_PLLT4 = c_L5;                                   % taking the values of n and cl that give us tolerable error
    n_tolerance4L = i+1;           % number of elements
end

if(error_cDi < allow4) && cDi_tolerance_PLLT4 == 0 % Checking if error is below allowable
    cDi_tolerance_PLLT4 = c_Di5;                                  % taking the values of n and cDi that give us tolerable error
    n_tolerance4Di = i+1;           % number of elements
end

end

% Lift/Drag
L_1 = 0.5*rho5*V5^2*cl_tolerance_PLLT2*S5;
Di_1 = 0.5*rho5*V5^2*cDi_tolerance_PLLT2*S5;
L_2 = 0.5*rho5*V5^2*cl_tolerance_PLLT3*S5;
Di_2 = 0.5*rho5*V5^2*cDi_tolerance_PLLT3*S5;
L_3 = 0.5*rho5*V5^2*cl_tolerance_PLLT4*S5;
Di_3 = 0.5*rho5*V5^2*cDi_tolerance_PLLT4*S5;

% data for plots
for i=1:length(N3)

[~,c_L5_vec(i),c_Di5_vec(i)] = PLLT(b5,a0_t5,a0_r5,ct5,cr5,aero_t5,aero_r5,geo_t5,geo_r5,N3(i));

L5 = 0.5*rho5*V5^2*c_L5_vec*S5;
Di5 = 0.5*rho5*V5^2*c_Di5_vec*S5;

end

% to estimate c_d, calculate Re for given flight conditions, pull from NACA charts
Re5 = (rho5*V5*((cr5+ct5)/2))/mu5; % use average chord length
c_d_0012 = 0.006;      % Taken from experimental data at Reynold's number = 3*10^6 (similar to our Re5 calculated number)
c_d_2412 = 0.0065;

c_d_avg = (c_d_0012 + c_d_2412)/2;

```

```
C_D = c_d_avg + cDi_tolerance_PLLT4;  % Calculating cd with estimated values and our lowest error
    cDi
D = 0.5*rho5*V5^2*C_D*S5;

LoD = L_3/D;     % For lowest error L/D problem 5

geor_vec = linspace(-10,20,50);
geor_vec_rad = geor_vec .* pi/180;
geot_vec = geor_vec - 1;
geot_vec_rad = geot_vec .* pi/180;

for i=1:length(geor_vec_rad)   % Finding L/D at varying angle of attack

[~,c_L2,c_Di2] = PLLT(b5,a0_t5,a0_r5,ct5,cr5,aero_t5,aero_r5,geot_vec_rad(i),geor_vec_rad(i),20);


% cd vector
% NACA data
cd0012 = [0.0136 0.0054 0.0136];
cl0012 = [-1 0 1];

cd2412 = [0.008 0.006 0.0091];
cl2412 = [ -0.65 0.2 1.08];

% quadratic fit
fit0012 = polyfit(cl0012,cd0012,2);
fit2412 = polyfit(cl2412,cd2412,2);

% average root and tip c_d to get c_d vec
c_d2 = (polyval(fit0012 ,c_L2)+polyval(fit2412 ,c_L2))/2;

% combining C_D
C_D2 = c_d2 + c_Di2;
LoD_vec(i) = c_L2 / C_D2;

end


%% CMD Line Prints
% Problem 1
fprintf('Problem 1: \n')
fprintf('\n')
fprintf("NACA 0006 Cl at 99 percent = %d \n", cl_tolerance)
fprintf("NACA 0006 - Number of Panels at 99 percent Cl = %d \n", n_tolerance)
fprintf('\n')

% Problem 2
fprintf('Problem 2: \n')
fprintf('\n')
fprintf("NACA 0006 lift slope in 1/rad = %d \n",a_0_0006_rad)
fprintf("NACA 0006 zero lift AOA in degrees = %d \n",alpha_L0_0006_2)
fprintf('\n')
fprintf("NACA 0012 lift slope in 1/rad = %d \n",a_0_0012_rad)
fprintf("NACA 0012 zero lift AOA in degrees = %d \n",alpha_L0_0012)
fprintf('\n')
fprintf("NACA 0018 lift slope in 1/rad = %d \n",a_0_0018_rad)
fprintf("NACA 0018 zero lift AOA in degrees = %d \n",alpha_L0_0018)
fprintf('\n')
fprintf("Symmetric TAT lift slope in 1/rad = %d \n",a_0_TAT*(180/pi))
```

```
fprintf("Symmetric TAT zero lift AOA in degrees = %d \n",0)
fprintf('\n')

% Problem 3
fprintf('Problem 3: \n')
fprintf('\n')
fprintf("NACA 0012 lift slope in 1/rad = %d \n",a_0_0012_rad)
fprintf("NACA 0012 zero lift AOA in degrees = %d \n",alpha_L0_0012)
fprintf("TAT NACA 0012 lift slope in 1/rad = %d \n",a_0_TAT*(180/pi))
fprintf("TAT NACA 0012 zero lift AOA in degrees = %d \n",0)
fprintf('\n')
fprintf("NACA 2412 lift slope in 1/degrees = %d \n",a_0_2412_rad)
fprintf("NACA 2412 zero lift AOA in degrees = %d \n",alpha_L0_2412)
fprintf("TAT NACA 2412 lift slope in 1/degrees = %d \n",a_0_2412_TAT*(180/pi))
fprintf("TAT NACA 2412 zero lift AOA in degrees = %d \n",alpha_L0_TAT_2412)
fprintf('\n')
fprintf("NACA 4412 lift slope in 1/rad = %d \n",a_0_4412_rad)
fprintf("NACA 4412 zero lift AOA in degrees = %d \n",alpha_L0_4412)
fprintf("TAT NACA 4412 lift slope in 1/rad = %d \n",a_0_4412_TAT*(180/pi))
fprintf("TAT NACA 4412 zero lift AOA in degrees = %d \n",alpha_L0_TAT_4412)
fprintf('\n')

% Problem 5
fprintf('Problem 5: \n')
fprintf('\n')
fprintf("Steady state lift in lb_f = %d \n", 0.5*rho5*V5^2*S5*cL_SS)
fprintf("Number of terms to get lift within 10%% error = %d \n",n_tolerance2L)
fprintf("Number of terms to get lift within 1%% error = %d \n",n_tolerance3L)
fprintf("Number of terms to get lift within 0.1%% error = %d \n",n_tolerance4L)
fprintf('\n')
fprintf("Steady state induced drag in lb_f = %d \n", 0.5*rho5*V5^2*S5*cDi_SS)
fprintf("Number of terms to get induced drag within 10%% error = %d \n",n_tolerance2Di)
fprintf("Number of terms to get induced drag within 1%% error = %d \n",n_tolerance3Di)
fprintf("Number of terms to get induced drag within 0.1%% error = %d \n",n_tolerance4Di)

%% Plotting

% Problem 1
figure(1)
plot(N,c_l_0006)
hold on
yline(c_l_0006_actual,'k')
yline(cl_tolerance,'--r')
hold off
xlabel("Number of Panels")
ylabel("C_l")
title("C_l vs. Number of Panels (Problem 1)")
legend('C_l','Actual C_l','C_l Tolerance','Location','southeast')
ylim([1.105 1.15])
grid on

% Problem 2
figure(2)
plot(alpha_vary,c_l_0006_2,'r')
hold on
plot(alpha_vary, c_l_0012,'b')
plot(alpha_vary, c_l_0018,'g')

plot(TWS_0006x,TWS_0006y,'o--r')
```

```
plot(TWS_0012x,TWS_0012y,'o--b')

plot(alpha_vary, TAT_y,'k')

ylabel("C_l of Airfoil")
xlabel("\alpha (degrees)")
title("C_l vs \alpha (Problem 2)")
legend('0006', '0012', '0018','NACA 0006 Data','NACA 0012 Data', 'TAT','Location','northwest')
grid on


% Problem 3
figure(3)
plot(alpha_vary, c_l_0012,'r')
hold on
plot(alpha_vary,c_l_2412,'b')
plot(alpha_vary, c_l_4412,'g')

plot(TWS_0012x,TWS_0012y,'o--r')
plot(TWS_2412x,TWS_2412y,'o--b')
plot(TWS_4412x,TWS_4412y,'o--g')

plot(alpha_vary,TAT_y,'-.r')
plot(alpha_vary,TAT_2412,'-.b')
plot(alpha_vary,TAT_4412,'-.g')

ylabel("C_l of Airfoil")
xlabel("\alpha (degrees)")
title("C_l vs \alpha (Problem 3)")
legend('0012', '2412', '4412','NACA 0012 Data','NACA 2412 Data','NACA 4412 Data' ,'TAT 0012','TAT
    2412','TAT 4412','Location','northwest')
grid on

% Problem 4
figure(4)
hold on
plot(ct_cr10,delta10)
plot(ct_cr8,delta8)
plot(ct_cr6,delta6)
plot(ct_cr4,delta4)
xlabel('c_r/c_t')
ylabel('Delta')
grid on
title("Matching Anderson Plots")
legend('AR = 10','AR = 8','AR = 6','AR = 4')
hold off

% Problem 5
figure(5)
hold on
plot(N3,L5)
grid on
ylabel('Lift [lb_f]')
xlabel('Number of Terms')
title('Lift Convergence')

figure(6)
hold on
plot(N3,Di5)
```

```
grid on
ylabel('Induced Drag [lb_f]')
xlabel('Number of Terms')
title('Induced Drag Convergence')

figure(7)
plot(geot_vec,LoD_vec)
grid on
ylabel('Lift over Drag (Aerodynamic Efficiency)')
xlabel('Angle of Attack (degrees)')
title('Aerodynamic Efficiency vs. AoA')


%% Functions

function [x_b, y_b] = NACA_Airfoils(m,p,t,c,N)

x = linspace(c,0,N); % Starting at TE going to LE

y_t = (t*c / 0.2) * (0.2969.*sqrt(x/c) - 0.126.*(x/c) - 0.3516.*((x/c).^2) + 0.2843.*((x/c).^3) -
    0.1036.*((x/c).^4));

% preallocate
y_c = zeros(1,length(x));
dy_c = zeros(1,length(x));
x_U = zeros(1,length(x));
x_L = zeros(1,length(x));
y_U = zeros(1,length(x));
y_L = zeros(1,length(x));

    for i=1:length(x)
    if x(i) <= p*c

    y_c(i) = m*(x(i)/p^2)*(2*p - x(i)/c);
    dy_c(i) = -2*m * (x(i) - c*p) / (c*p^2);

    elseif x(i) > p*c

    y_c(i) = m*((c - x(i)) / (1-p)^2) * (1 + x(i)/c - 2*p);
    dy_c(i) = -2*m*(x(i) - c*p) / (c * (p-1)^2);

    end

    squiggly = atan(dy_c);
    x_U(i) = x(i) - y_t(i)*sin(squiggly(i));
    x_L(i) = x(i) + y_t(i)*sin(squiggly(i));
    y_U(i) = y_c(i) + y_t(i)*cos(squiggly(i));
    y_L(i) = y_c(i) - y_t(i)*cos(squiggly(i));
    end

x_U = fliplr(x_U); % Flips vectors to make sure we are going clockwise
y_U = fliplr(y_U);

x_b = [x_L , x_U(2:end)];  % (2:end) so that we do not duplicated leading edge value
y_b = [y_L , y_U(2:end)];

x_b(isnan(x_b)) = 0;
y_b(isnan(y_b)) = 0;
```

```
end

function [m, p, t] = NACAdata(str)

m = str(1);
m = str2double(m) / 100;

p = str(2);
p = str2double(p) / 10;

t1 = str(3);
t2 = str(4);

t = strcat(t1, t2);
t = str2double(t) / 100;

end

function [e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t,c_r,aero_t,aero_r,geo_t,geo_r,N)

% Wing geometry
S = 0.5 * (c_t + c_r) * b;
AR = b^2/S;

top = pi/(2*N);
theta = linspace(top,pi/2,N);   % Making theta vec (left half of span)
y = -b/2 * cos(theta);         % Getting y vec

% preallocate
c = zeros(length(theta),1);
a0 = zeros(length(theta),1);
a_L0 = zeros(length(theta),1);
a_geo = zeros(length(theta),1);
alpha_vec = zeros(N,1);
A_vec = zeros(N,N);
A_n_math = zeros(N-1,1);

for i = 1:length(theta)
    c(i) = c_r + y(i) * (c_t - c_r) / (y(1) - y(end)); % linear interpolation
    a0(i) = a0_r + y(i) * (a0_t - a0_r) / (y(1) - y(end));
    a_L0(i) = aero_r + y(i) * (aero_t - aero_r) / (y(1) - y(end));
    a_geo(i) = geo_r + y(i) * (geo_t - geo_r) / (y(1) - y(end));
    alpha_vec(i) = a_geo(i) - a_L0(i); % Vector math stuff start
    for j = 1:length(theta)
        A_vec(i,j) = 4*b / (a0(i)*c(i)) * sin((2*j-1)*theta(i)) + (2*j-1) *
            sin((2*j-1)*theta(i))/sin(theta(i));
    end
end

A_n = A_vec\alpha_vec;

c_L = AR * pi * A_n(1);

    for k=2:length(A_n)
        A_n_math(k-1) = (2*k-1)*(A_n(k)/A_n(1))^2;
    end

    delta = sum(A_n_math);
    e = 1 / (1+delta);
```

```matlab
    c_Di = c_L^2 / (pi*e*AR);
end


function [CL] = Vortex_Panel(XB,YB,ALPHA)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:                      %
%                             %
% XB = Boundary Points x-location %
% YB = Boundary Points y-location %
% ALPHA = AOA in degrees      %
%                             %
% Output:                     %
%                             %
% CL = Sectional Lift Coefficient %
% improves efficiency by preallocating matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%
% Convert to Radians %
%%%%%%%%%%%%%%%%%%%%%%

ALPHA = ALPHA*pi/180;

%%%%%%%%%%%%%%%%%%%%%
% Compute the Chord %
%%%%%%%%%%%%%%%%%%%%%

CHORD = max(XB)-min(XB);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine the Number of Panels %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M = max(size(XB,1),size(XB,2))-1;
MP1 = M+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preallocate Matrices for Efficiency %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = zeros(1,M);
Y = zeros(1,M);
S = zeros(1,M);
THETA = zeros(1,M);
SINE = zeros(1,M);
COSINE = zeros(1,M);
RHS = zeros(1,M);
CN1 = zeros(M);
CN2 = zeros(M);
CT1 = zeros(M);
CT2 = zeros(M);
AN = zeros(M);
AT = zeros(M);
V = zeros(1,M);
CP = zeros(1,M);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Intra-Panel Relationships:                         %
```

```
%                                                    %
% Determine the Control Points, Panel Sizes, and Panel Angles %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
    IP1 = I+1;
    X(I) = 0.5*(XB(I)+XB(IP1));
    Y(I) = 0.5*(YB(I)+YB(IP1));
    S(I) = sqrt( (XB(IP1)-XB(I))^2 +( YB(IP1)-YB(I))^2 );
    THETA(I) = atan2( YB(IP1)-YB(I), XB(IP1)-XB(I) );
    SINE(I) = sin( THETA(I) );
    COSINE(I) = cos( THETA(I) );
    RHS(I) = sin( THETA(I)-ALPHA );
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inter-Panel Relationships:       %
%                                  %
% Determine the Integrals between Panels %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
    for J = 1:M
        if I == J
            CN1(I,J) = -1.0;
            CN2(I,J) = 1.0;
            CT1(I,J) = 0.5*pi;
            CT2(I,J) = 0.5*pi;
        else
            A = -(X(I)-XB(J))*COSINE(J) - (Y(I)-YB(J))*SINE(J);
            B = (X(I)-XB(J))^2 + (Y(I)-YB(J))^2;
            C = sin( THETA(I)-THETA(J) );
            D = cos( THETA(I)-THETA(J) );
            E = (X(I)-XB(J))*SINE(J) - (Y(I)-YB(J))*COSINE(J);
            F = log( 1.0 + S(J)*(S(J)+2*A)/B );
            G = atan2( E*S(J), B+A*S(J) );
            P = (X(I)-XB(J)) * sin( THETA(I) - 2*THETA(J) ) ...
              + (Y(I)-YB(J)) * cos( THETA(I) - 2*THETA(J) );
            Q = (X(I)-XB(J)) * cos( THETA(I) - 2*THETA(J) ) ...
              - (Y(I)-YB(J)) * sin( THETA(I) - 2*THETA(J) );
            CN2(I,J) = D + 0.5*Q*F/S(J) - (A*C+D*E)*G/S(J);
            CN1(I,J) = 0.5*D*F + C*G - CN2(I,J);
            CT2(I,J) = C + 0.5*P*F/S(J) + (A*D-C*E)*G/S(J);
            CT1(I,J) = 0.5*C*F - D*G - CT2(I,J);
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inter-Panel Relationships:      %
%                                 %
% Determine the Influence Coefficients %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
    AN(I,1) = CN1(I,1);
    AN(I,MP1) = CN2(I,M);
    AT(I,1) = CT1(I,1);
    AT(I,MP1) = CT2(I,M);
    for J = 2:M
        AN(I,J) = CN1(I,J) + CN2(I,J-1);
        AT(I,J) = CT1(I,J) + CT2(I,J-1);
```

```
    end
end
AN(MP1,1) = 1.0;
AN(MP1,MP1) = 1.0;
for J = 2:M
    AN(MP1,J) = 0.0;
end
RHS(MP1) = 0.0;

%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for the gammas %
%%%%%%%%%%%%%%%%%%%%%%%%

GAMA = AN\RHS';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Tangential Veloity and Coefficient of Pressure %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
    V(I) = cos( THETA(I)-ALPHA );
    for J = 1:MP1
        V(I) = V(I) + AT(I,J)*GAMA(J);
    end
    CP(I) = 1.0 - V(I)^2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Sectional Coefficient of Lift %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CIRCULATION = sum(S.*V);
CL = 2*CIRCULATION/CHORD;

end
```