Lab 3: Airfoil Pressure Distribution

Gregory Golonka, Connor Hack, and Timothy Welch

April 2024

Abstract

The objective of this lab is to measure the surface pressure distribution across a symmetric airfoil. Analyzing the known distribution to calculate the lift, drag, and moment coefficients, to compare with thin airfoil theory. During this lab a NACA 0015 airfoil, with pressure taps at known locations, was connected to two 16-channel NetScanner pressure transducers, mapping out the airfoil. In the end, the pressure distribution followed the trends found in each theoretical case. Furthermore, the tests conducted found that as the angle of attack increased from 0 to 17.5°, the coefficient of lift increased from 0.039 to 1.215. At the same time, the drag coefficient increased from 0.001 to 0.527.

1 Introduction

The goal of this lab was to calculate the lift, drag, and moment coefficients using a collection of surface pressure measurements. To accomplish this goal, a NACA 0015 airfoil, equipped with 29 pressure taps, was used to measure the pressure distribution at different angles of attack. The pressure distribution was then used to calculate the axial and normal force coefficients as defined as:

$$c_n \equiv \frac{N'}{q_\infty c} \qquad c_a \equiv \frac{A'}{q_\infty c} \tag{1}$$

where q_∞ is the free stream dynamic pressure, A is the axial force, N is the normal force, and the $'$ indicates per unit span. Defining the axial direction as parallel to the chord line, and the normal perpendicular gives the following relationship to the lift and drag coefficients by the angle of attack, α

$$c_l = c_n \cos\alpha - c_a \sin\alpha \tag{2}$$

$$c_d = c_n \sin\alpha + c_a \cos\alpha \tag{3}$$

Finally, these characteristics of the airfoil were then compared to analytical results using XFOIL[1], which utilizes thin airfoil theory to calculate the different force and moment coefficients across the airfoil at different angles of attack.

2 Experimental Setup

2.1 Define Ambient Conditions

The ambient conditions of the laboratory were measured for the period in which the experiment was conducted. The ambient pressure and temperature were obtained from a Fisher Scientific Traceable digital barometer within the testing room. This data was taken at both the beginning and end of the lab duration. The temperature and pressure data can be used to calculate the air density with the Ideal Gas Law given in the equation

$$\rho = \frac{P}{RT} \tag{4}$$

where ρ is the density, P is the ambient pressure, R is the specific gas constant of air taken to be 287 $\frac{\text{J}}{\text{kg}\cdot\text{K}}$, and T is the temperature. The values of P, T, and ρ taken and calculated for both times, as well as their averages, are tabulated in Table 1.

Table 1: Ambient conditions of the lab before and after the experiments.

Time	Temperature, T (K)	Pressure, P (Pa)	Density, ρ ($\frac{\text{kg}}{\text{m}^3}$)
5:14pm	298.65	9.92E+04	1.12
6:02pm	298.15	9.92E+04	1.16
Average	298.51 ± 0.6	9.95E+04 $\pm$ 400	1.12 ± 0.01

The uncertainties in these measurements are important to the later analysis. The uncertainties in the average pressures, temperature, and density were calculated by propagating component uncertainties in quadrature. The component uncertainties from the above-stated measurement device are listed in Table 2.

Table 2: Error of pressure and temperature measurements.

Source	Pressure, ϵ_P (Pa)	Temperature, ϵ_T (K)
Reading	100	0.1
Systematic	400	0.4
Drift	100	0.4

The uncertainty in density was calculated from the quadrature addition of these uncertainties via the equation

$$\epsilon_\rho = \sqrt{\left(\frac{\epsilon_P}{RT}\right)^2 + \left(\frac{-\epsilon_T P}{T^2}\right)^2} \tag{5}$$

where ϵ_ρ is the uncertainty in density, ϵ_P is the uncertainty in pressure, and ϵ_T is the uncertainty in temperature. This was the same process used in Lab 1: Transducer Calibration & Pitot Wake Profiles.

2.2 Preparing Experiment

To measure the pressure data that was the focus of the lab, the NACA 0015 airfoil used in the experiment had 29 pressure taps across its surface. Tygon tubing was used to connect each of these pressure taps on the NACA 0015 airfoil to the pressure transducer used in the lab. It is assumed that throughout the tubes the static pressure is the same as the static pressure at the given tap location, allowing the following analysis and data processing to be considered true to life.

In order to ensure then that the data received from the pressure taps experienced minimal error, careful design and manufacturing of the taps was necessary. The diameter of the pressure tap should be twice the diameter of the Tygon tubing used and the ratio of the orifice depth to the diameter of the orifice should be between 0.1 and 1.75. Careful creation of the taps is also necessary as any burs or imperfections on the surface around the pressure taps could lead to differences between recorded pressure data and the true pressure.

Since the precise manufacturing of the pressure taps was crucial to the success of the lab as a whole, these pressure taps were created by the TAs prior to the conduction of the lab. Tabulated in Table 3 are the coordinates of these pressure taps over the airfoil.

Table 3: Pressure tap coordinates and indices.

x (mm)	$\|y\|$ (mm)	Upper	Lower
0.00	0.00		0
0.84	2.44	15	1
3.33	4.68	16	2
7.46	6.71	17	3
13.17	8.45	18	4
20.42	9.85	19	5
29.11	10.83	20	6
39.14	11.35	21	7
50.42	11.39	22	8
62.82	10.96	23	9
76.20	10.09	24	10
90.41	8.80	25	11
105.31	7.15	26	12
120.71	5.15	27	12
136.47	2.87	28	14

With the pressure taps created, the tubes connected to the orifice of the pressure taps were then connected to the pressure scanner used in the lab. Two NetScanner 9116 pressure scanners were utilized for this purpose, which featured 16 silicon piezoresistive pressure sensors to affix the tygon tubes to. Thus as will be discussed further in 3.1 NetScanner of the Experimental Procedure, the two NetScanner 9116 pressure scanners will both show up in the software for data gathering.

As with the pressure taps created prior to the lab, the tygon tubing was connected to the taps and the pressure transducers prior to lab. Thus all that was required for the experimental setup was to ensure that the connections between taps and tubing were strong, and to make sure that each tube was correctly connected to the corresponding pressure sensor.

2.3 Testing Aparatus

A brief discussion of the wind tunnel used in this lab is also relevant. The lab was conducted using a turbine-type wind tunnel found in the Hessert Laboratories on the University of Notre Dame campus. The tunnel fluctuates the freestream velocity as a percentage of the maximum motor speed. The flow velocity was held constant throughout the duration of the Airfoil Pressure Distribution Lab. The schematic for the wind tunnel can be found in Figure 1. The schematic for the wind tunnel comes from the Lab 2 Circulation around a 2-D Airfoil Measured with a Five-Hole Probe lab handout [2].

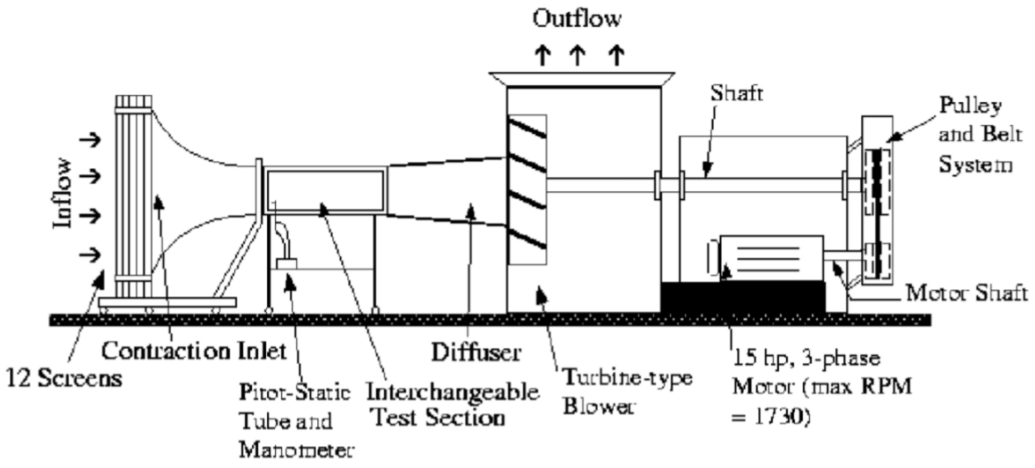


Figure 1: Informational graphic describing the different parts of a turbine-type wind turbine.

3 Experimental Procedure

The following procedure comes from the lab handout [3].

3.1 NetScanner

The entirety of this lab was conducted using the NUSS software, on the wind tunnel computer. To start the lab, ensure that the pitot tube is parallel to the tunnel walls and the airfoil is at zero degrees AoA (angle of attack). All data collected during this lab was at 45% wind tunnel speed. In order to collect data with the Netscanner 9116 pressure scanner, start the NUSS software. In the "Network Status" window:

(a) Two devices should appear under the 9116 tab: 9876 and 9897

(b) Select Group A if not already.

(c) Click "Connect" in the top toolbar, to ensure the two devices are connected.

After connecting the two 9116 scanners to the NUSS Software, the data can be collected as follows. In the "NetScanner™Unified Startup Software (NUSS)" window:

1. Click Run > Run (Group) resulting in three new windows appearing. The "Run Group A" window controls the data acquisition while the others only visually represent it.
2. In the "Run Group A" window, under "Record Controls", click "Rec" to begin recording data. Wait 10 seconds, then click "Rec" again to end the recording.
3. The data is recorded to the *.dat file listed in the white box. The directory is listed in the lower white box.

The NUSS Software automatically collects the pressure readings in PSI, relative the the ambient pressure in the room. Thus no calibration was needed to analyze the transducer data. To acquire a full data set, the NACA 0015 airfoil was then tested for angles of attack of 2.5, 5, 7.5, 10, 12.5, 15, and 17.5 degrees. Once all data files were collected, the *.dat files were then converted to *.csv format to import into MATLAB for data processing.

4 Results and Analysis

4.1 Pressure Coefficients

In order to visualize the pressure coefficients of the NACA 0015 airfoil over the chord, several data processing equations and techniques were implemented. Each of the 29 pressure taps on the airfoil took in pressure data during the 10-second duration that the NetScanner ran for. The data gathered during that time interval was then averaged out to obtain the mean pressure for that pressure tap at that angle of attack. These mean pressure readings were then used with the stagnation and freestream pressures to find the pressure coefficients of each pressure tap as defined:

$$C_p \equiv \frac{p - p_\infty}{q_\infty} \tag{6}$$

where p is the pressure at that pressure tap, p_∞ is the free stream pressure, and q_∞ is the free stream dynamic pressure. Rearranging using Bernoulli's equation such that:

$$q_\infty = p_0 - p_\infty \tag{7}$$

where p_0 is the stagnation pressure, Eq. 6 can then be rearranged to include the stagnation pressure, thereby ensuring that the pressure coefficient for each pressure tap can be calculated with the equation:

$$C_p \equiv \frac{p - p_\infty}{p_0 - p_\infty} \tag{8}$$

Using Eq. 8 the pressure coefficients for each pressure tap were calculated at 0, 2.5, 5, 7.5, 10, 12.5, 15, and 17.5-degree angles of attacks. Additionally, the uncertainty in the pressure coefficient is determined via the equation

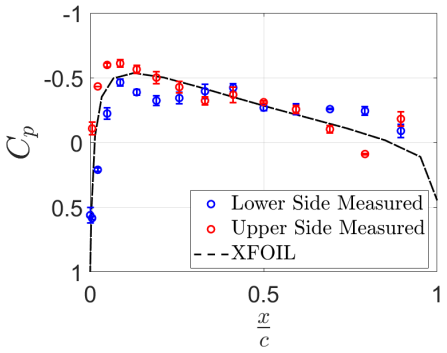
$$\epsilon_{C_p} = \sqrt{\left(\epsilon_p \frac{1}{p_0 - p_\infty}\right)^2 + \left(\epsilon_{p_\infty} \frac{p - p_0}{(p - p_\infty)^2}\right)^2 + \left(\epsilon_{p_0} \frac{p_\infty - p}{(p_0 - p_\infty)^2}\right)^2} \tag{9}$$

where ϵ_{C_p} is the uncertainty in the pressure coefficient, ϵ_{p_∞} is the uncertainty in freestream pressure, ϵ_{p_0} is the uncertainty in total pressure, and ϵ_p is the uncertainty in static pressure. All values of uncertainty in the various pressures come from the equation

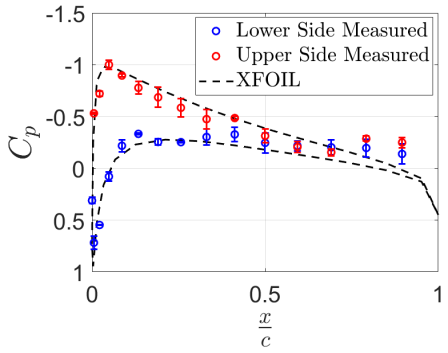
$$\epsilon_p = \sqrt{\epsilon_{p,sys}^2 + \epsilon_{p,read}^2 + \epsilon_{p,drift}^2} \tag{10}$$

where $\epsilon_{p,sys}$ and $\epsilon_{p,read}$ are the systematic and reading uncertainties in the pressure which come from the pressure transducer spec sheet [4], and $\epsilon_{p,drift}$ is the drift uncertainty in pressure across the 20 data points gathered for each pressure value for each angle of attack.

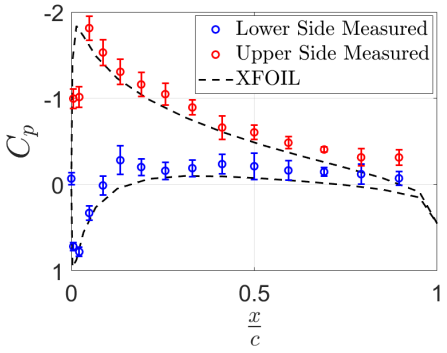
In addition to these experimental C_p values and their uncertainties, the software XFOIL was utilized to generate theoretical data as a comparison to the experimentally gathered results. To accomplish this, when XFOIL was launched and the airfoil option was selected, 29 source panels were chosen in order to accurately mirror what was produced in the lab. A discussion of the XFOIL results as compared to the experimentally gathered results can be found in Section 5.1 Pressure Coefficients. These results, both experimental and computational, are plotted as a function of the x position, non-dimensionalized by the chord length, for each angle of attack in Figure 2. Within each plot, the legend defines the plotted data in the following order: lower side measured, upper side measured, and XFOIL C_p at desired AoA.



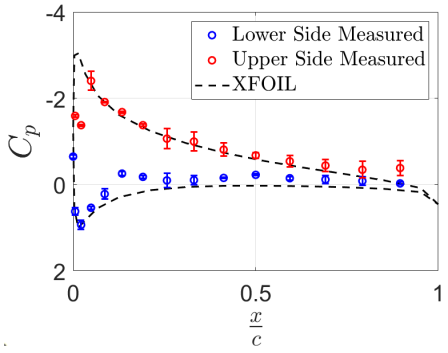
(a) Angle of attack: 0°



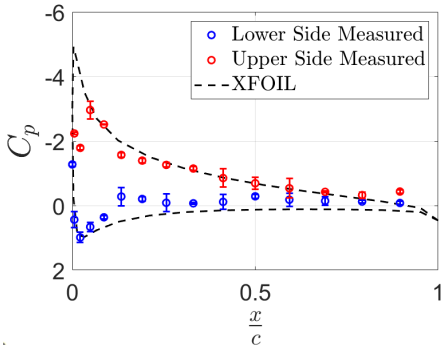
(b) Angle of attack: 2.5°



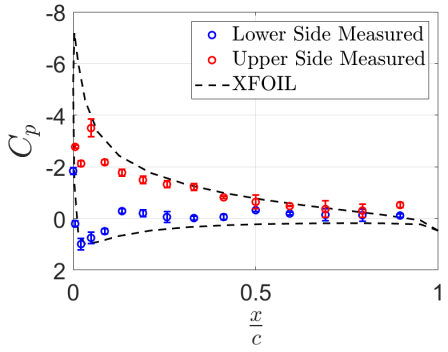
(c) Angle of attack: 5°



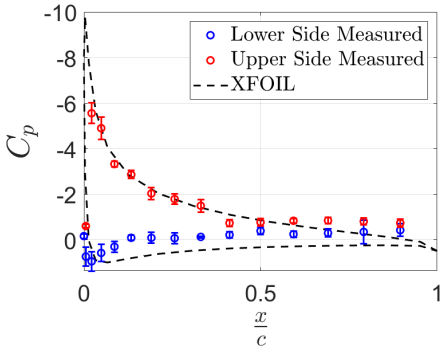
(d) Angle of attack: 7.5°



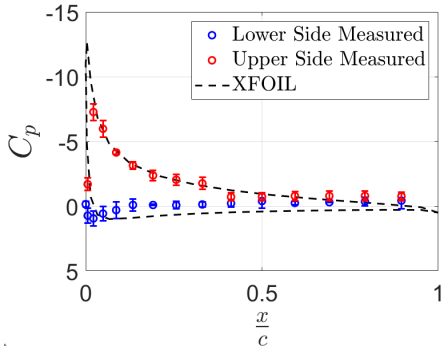
(e) Angle of attack: 10°



(f) Angle of attack: 12.5°



(g) Angle of attack: 15°



(h) Angle of attack: 17.5°

Figure 2: C_p distribution across the NACA 0015 airfoil at varying angle of attack.

4.2 Aerodynamic Forces and Moment Coefficients

The lift coefficient of the NACA 0015 airfoil could not be measured directly and therefore had to be calculated. Eq. 2 outlines the relationship between the axial and normal coefficients as well as the angle of attack and their effect on lift production. One important note, the normal and axial forces are totals across the entire span. However, the pressure is not known as a function of x and y, but rather measured at each tap location. As such, the integral forms of the equations for the normal, axial force coefficients can be approximated with a Riemann sum. Splitting across the upper and lower surfaces, the discretized coefficients therefore become:

$$c_n = \sum_{\text{lower}} \frac{1}{2}\left(c_{p_i} + c_{p_{i+1}}\right)\left(\frac{x_{i+1}}{c} - \frac{x_i}{c}\right) - \sum_{\text{upper}} \frac{1}{2}\left(c_{p_i} + c_{p_{i+1}}\right)\left(\frac{x_{i+1}}{c} - \frac{x_i}{c}\right) \tag{11}$$

$$c_a = \left[-\sum_{\text{lower}} \frac{1}{2}\left(c_{p_i} + c_{p_{i+1}}\right)\left(\frac{y_{i+1}}{c} - \frac{y_i}{c}\right)\right] - \left[-\sum_{\text{upper}} \frac{1}{2}\left(c_{p_i} + c_{p_{i+1}}\right)\left(\frac{y_{i+1}}{c} - \frac{y_i}{c}\right)\right] \tag{12}$$

In addition to finding the coefficients of normal and axial forces, the coefficient for the moment about the leading edge was also obtained. Similarly to the forces, the moment coefficient has to be discretized across the upper and lower surfaces to approximate. The following equation outlines the summation equation necessary to compute it, which was done in MATLAB

$$c_{m_{LE}} = \left[-\sum_{\text{upper}} \frac{1}{2}\left(c_{p_i}\frac{x_i}{c} + c_{p_{i+1}}\frac{x_{i+1}}{c}\right)\left(\frac{x_i}{c} - \frac{x_{i+1}}{c}\right) - \sum_{\text{upper}} \frac{1}{2}\left(c_{p_i}\frac{y_i}{c} + c_{p_{i+1}\frac{y_{i+1}}{c}}\right)\left(\frac{y_i}{c} - \frac{y_{i+1}}{c}\right)\right] \tag{13}$$

$$-\left[-\sum_{\text{lower}} \frac{1}{2}\left(c_{p_i}\frac{x_i}{c} + c_{p_{i+1}}\frac{x_{i+1}}{c}\right)\left(\frac{x_i}{c} - \frac{x_{i+1}}{c}\right) - \sum_{\text{lower}} \frac{1}{2}\left(c_{p_i}\frac{y_i}{c} + c_{p_{i+1}\frac{y_{i+1}}{c}}\right)\left(\frac{y_i}{c} - \frac{y_{i+1}}{c}\right)\right] \tag{14}$$

Using the equations for the coefficient of normal force, coefficient of axial force, and coefficient of moment about the leading edge, the values of each coefficient were obtained at the eight different angles of attack and tabulated in Table 4

Table 4: Calculated force and moment coefficients using Riemann sum approximation.

Angle of Attack, α	c_n	c_a	$c_{m_{LE}}$
0	0.0038573	0.00099867	-0.028137
2.5	0.19903	0.0059026	0.041849
5	0.55291	0.027319	0.16032
7.5	0.71924	0.052242	0.20687
10	0.79016	0.078967	0.20982
12.5	0.82284	0.10042	0.20805
15	1.1655	0.090266	0.29788
17.5	1.3173	0.13754	0.31173

Due to the summation calculations of the uncertainties in the force and moment coefficients in Eqs. 11 through 14, the uncertainties in the coefficients are calculated in a simple manner, in which the errors in pressure are summed in quadrature after being biased by the steps in position, which are assumed to have no uncertainty. These uncertainty values are tabulated in Table 5.

Table 5: Uncertainties in force and moment coefficients.

Angle of Attack, α	ϵ_{c_n}	ϵ_{c_a}	$\epsilon_{c_{m_{LE}}}$
0	0.048657	0.048657	0.048657
2.5	0.043825	0.043825	0.043825
5	0.18907	0.18907	0.18907
7.5	0.083885	0.083885	0.083885
10	0.055176	0.055176	0.055176
12.5	0.12644	0.12644	0.12644
15	0.0469	0.0469	0.0469
17.5	0.046723	0.046723	0.046723

4.3 Lift, Drag, and Quarter-Chord Moment Coefficients

The lift and drag coefficients were calculated using the discretized axial and normal force coefficients. Recall, Eq. 2 and 3 show the relationship between these coefficients and the AoA. The quarter-chord moment coefficient, however, is calculated using the definition

$$c_{m_{c/4}} = c_{m_{LE}} + \frac{c_l}{4} \tag{15}$$

Meaning it comes from a combination of the discretized force coefficients as defined in Section 4.2 and the pressure coefficient across the dimensionless span. The aerodynamic coefficients were then tabulated for each angle of attack in Table 6.

Table 6: Calculated coefficients of lift, drag, and moment about the quarter-chord.

Angle of Attack, α	c_l	c_d	$c_{m_{c/4}}$
0	0.0038573	0.00099867	-0.027173
2.5	0.19858	0.014579	0.091495
5	0.54842	0.075404	0.29743
7.5	0.70626	0.14567	0.38344
10	0.76444	0.21498	0.40093
12.5	0.7816	0.27613	0.40345
15	1.1024	0.38883	0.57347
17.5	1.215	0.5273	0.61548

To determine the uncertainty of the aerodynamic forces, the errors of the individual parameters were propagated through quadrature. The uncertainties for the lift, drag, and quarter-chord moment coefficients are determined with the expressions

$$\epsilon_{c_l} = \sqrt{\left(\epsilon_{c_n} cos(\alpha)\right)^2 + \left(\epsilon_\alpha(-c_n sin(\alpha) - c_a cos(\alpha))\right)^2 + \left(\epsilon_{c_a} sin(\alpha)\right)^2} \tag{16}$$

$$\epsilon_{c_d} = \sqrt{\left(\epsilon_{c_n} sin(\alpha)\right)^2 + \left(\epsilon_\alpha(c_n cos(\alpha) - c_a sin(\alpha))\right)^2 + \left(\epsilon_{c_a} cos(\alpha)\right)^2} \tag{17}$$

$$\epsilon_{c_{m_{c/4}}} = \sqrt{\epsilon_{c_{m_{LE}}}^2 + \left(\frac{1}{4}\epsilon_{c_l}\right)^2} \tag{18}$$

where ϵ_{c_l}, ϵ_{c_d}, and $\epsilon_{c_{m_{c/4}}}$ are the uncertainties in the lift coefficient, drag coefficient, and moment coefficient about the quarter-chord, respectively, and ϵ_α is the uncertainty in the angle of attack which was taken to be the reading uncertainty of the wind tunnel's built-in protractor. The uncertainties obtained from these equations are tabulated in Table 7.

Table 7: Uncertainties in coefficients of lift, drag, and moment about the quarter-chord.

Angle of Attack, α	ϵ_{c_l}	ϵ_{c_d}	$\epsilon_{c_{m_{c/4}}}$
0	0.048657	0.048657	0.050154
2.5	0.043826	0.04386	0.045174
5	0.18907	0.18913	0.19489
7.5	0.083894	0.084111	0.086467
10	0.055208	0.055577	0.056876
12.5	0.12647	0.12663	0.13034
15	0.047023	0.047876	0.048351
17.5	0.046949	0.047911	0.048175

4.3.1 Lift Coefficient

To find the coefficient of lift at the varying angles of attack, the coefficients tabulated in Table 4 were implemented into Eq. 2 to yield the coefficients of lift at the eight different angles of

attack. These eight different coefficients of lift values were plotted with uncertainties shown against the angle of attack in Figure 3. Additionally, the theoretical coefficient of lift vs angle of attack curve was plotted alongside it for comparison and discussion in section 5.3 lift, drag, and quarter-chord moment coefficients.

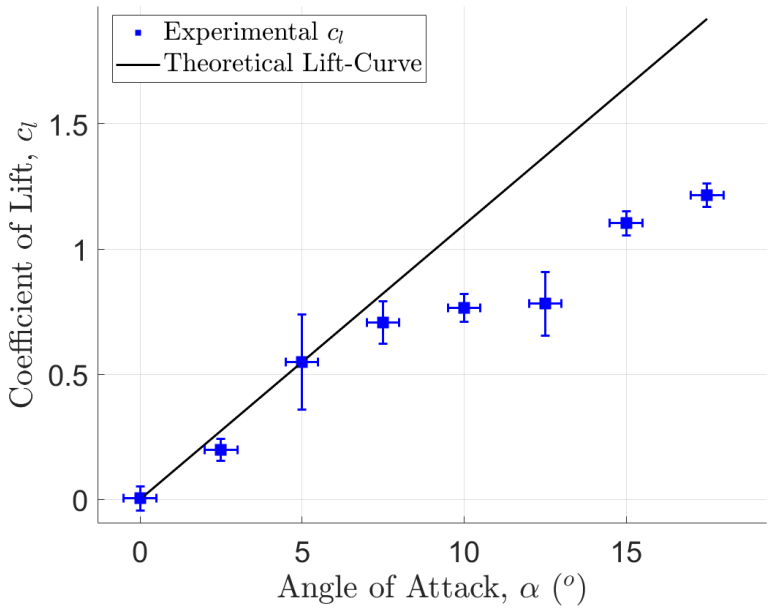


Figure 3: Coefficient of Lift vs Angle of Attack with Theoretical Curve

4.3.2 Drag Coefficient

In a similar process to finding the coefficient of lift, the coefficient of drag at each of the angles of attack was calculated using Eq. 3 and the tabulated values of Table 4. To visualize the coefficient of drag as a function of the angle of attack, the coefficients of drag along with accompanying uncertainty were plotted against the angle of attack in Figure 4.

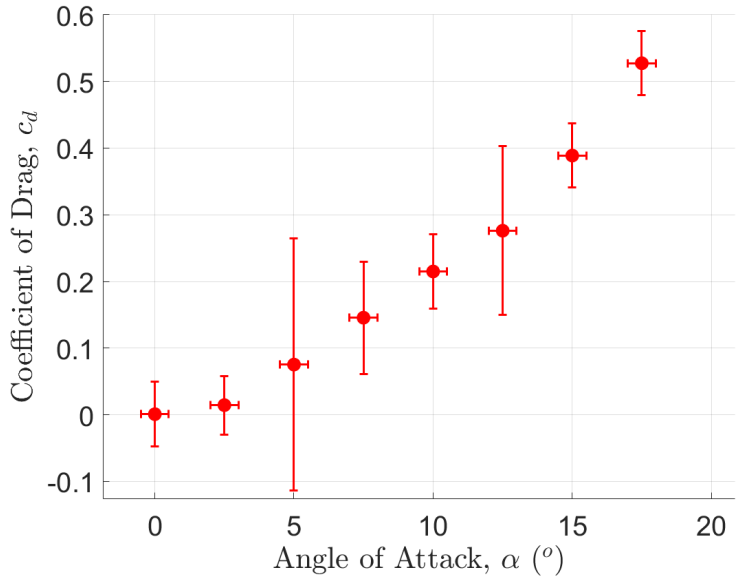


Figure 4: Coefficient of Drag vs Angle of Attack

In addition to visualizing the coefficients of drag as a function of the angle of attack, a drag polar plot was created, which plots the coefficient of drag as a function of the coefficient of lift. With the values already generated in Section 4.3.1, Figure 5 illustrates the drag polar complete with uncertainties at each point, which themselves are at each of the angles of attack used in the lab.

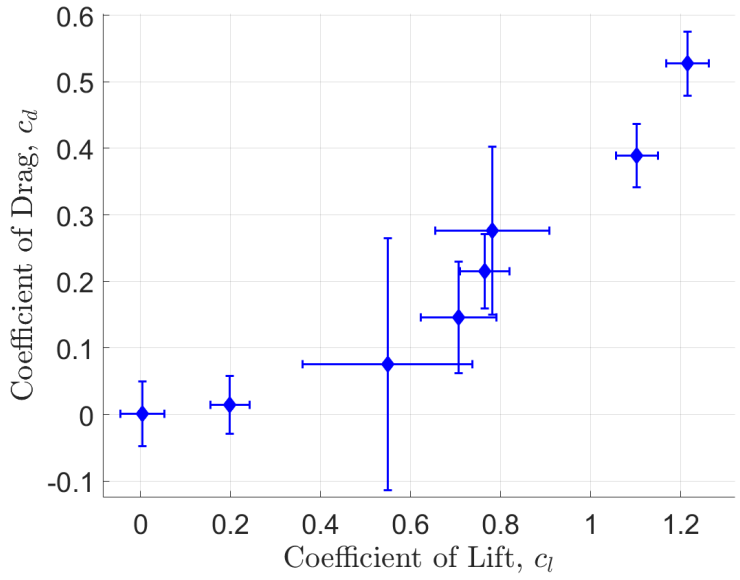


Figure 5: Coefficient of Drag vs Coefficient of Lift

4.3.3 Quarter-Chord Moment Coefficient

Using the results of the quarter-chord calculations, a plot versus the angle of attack was created in order to visualize its effect on the moment coefficient. The uncertainties were calculated using the process outlined in Section 4.3.

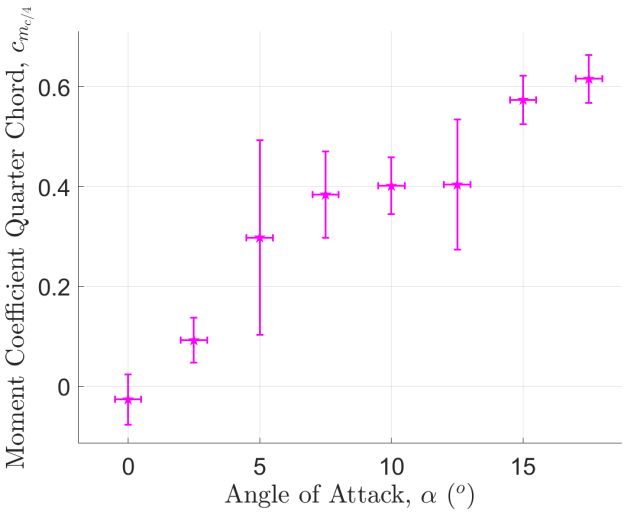


Figure 6: Quarter-Chord Moment Coefficient vs Angle of Attack

Another visualization of the quarter-chord moment coefficient was created that showed the quarter-chord moment as a function of the coefficient of lift. The data points used to plot this are themselves a function of the angle of attack, with each data point corresponding to one of the eight angles of attack used within the lab. The quarter-chord moment coefficients plotted as a function of the coefficient of lift are plotted along with their uncertainties in Figure 7.

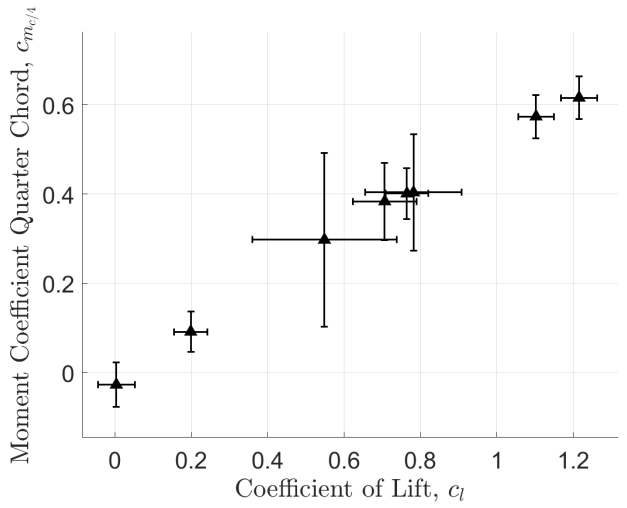


Figure 7: Quarter-Chord Moment vs Coefficient of Lift

5 Discussion

5.1 Pressure Coefficients

As seen in section 4.1 Pressure Coefficients, the experimental pressure coefficient curve over the airfoil matched very well with the theoretical XFOIL curve. The curves for experimental and theoretical coefficient of pressure aligned best at the leading edge of the airfoil as seen at the left of each of the graphs. The experimental coefficient of pressure differed most from the theoretical coefficient of pressure at the trailing edge, which is to be expected as the trailing edge would see the most impact from viscous forces which are neglected in the XFOIL theoretical coefficient of pressure curves.

In each of the theoretical coefficient of pressure graphs, there is a large negative spike near the leading edge on the upper surface which is not seen in the experimental coefficient of pressure points. This may be due purely to a lack of pressure taps that would sufficiently outline the correct pressure coefficient curve at the leading edge. Increasing the amount of pressure taps, particularly near the leading edge here may remedy that issue.

Finally, it can be seen from each of the eight different curves for the different angles of attack that the coefficients of pressure on the lower surface are generally less than those of the theoretical lower surface coefficient of pressure values. This may be due to viscous forces affecting the airspeed along the bottom surface of the airfoil. The coefficient of pressure equation as outlined by Eq.6 illustrates the inverse relationship between coefficient of pressure and dynamic pressure. Thus, when viscous effects decrease the speed and therefore dynamic pressure, the magnitude of the coefficient of pressure increases, accounting for the magnitude of the experimental coefficient of pressure values being greater than that of the theoretical coefficient of pressure.

5.2 Aerodynamic Forces and Moment Coefficients

As seen in Table 6, increasing the angle of attack led to increased coefficients of lift and coefficients of drag which was expected. An example of this phenomenon experienced in real life is in a commercial airline when taking off. To maximize the lift generated the plane will increase its angle of attack to just under its stall angle for maximum efficiency. This stall angle is the point in a viscous environment where the flow separates from the airfoil and decreases the lift. Furthermore, it can be seen from the same table that the coefficients of drag are all lower than the coefficients of lift for each angle of attack. These results look promising and will be further discussed in the subsequent discussion sections 5.3.1 Lift Coefficient and 5.3.2 Drag Coefficient utilizing the visualizations of Figure 3 and Figure 4 respectively.

5.3 Lift, Drag, and Quarter-Chord Moment Coefficients

5.3.1 Lift Coefficient

The coefficient of lift seems to match with the theoretical line for the low experimental angles of attacks until the 7.5-degree angle of attack. Beyond this point, the coefficient of lift line

does not follow the same slope as the theoretical coefficient of lift line. For the coefficient of lift curves as a function of angle of attack, there is typically an angle of attack at which the coefficient of lift curve drops off and is no longer linear. It is possible that this phenomenon may have occurred here, in which the coefficient of lift curve droops off early. However, another cause of this discrepancy may be due to the lower coefficient of normal forces caused by viscous forces keeping the flow running in the axial direction of the airfoil.

5.3.2 Drag Coefficient

The drag coefficient generally increases as a function of the angle of attack seen in Figure 4 which matches intuition. It is interesting to note that it appears to increase in an almost linear fashion with no one jump in angle of attack providing a massive jump in coefficient of drag.

The plot of the coefficient of drag as a function of the coefficient of lift illustrates that both the coefficient of lift and drag increase with the angle of attack, as each of the data points corresponds to one angle of attack. It is interesting to note that for higher angles of attacks, which yield higher coefficients of lift and drag as seen in Figure 3 and Figure 4 respectively, the coefficient of lift seems to increase more than the coefficient of drag. This is important because it means that when increasing at a small angle of attack, like when a plane is taking off, it will generate more lift than it will drag at these angles of attack which planes will generally go through when taking off.

5.3.3 Quarter-Chord Moment

It can be assessed from Figure 6 that the general trend for the quarter-chord moment coefficient increases as the angle of attack increases for small angles of attack. This finding is consistent with theory and intuition that posit an increased moment due to an increased force on the underside of the airfoil for larger angles of attack. A greater angle of attack creates larger magnitude normal forces on the airfoil's lower surface, thereby increasing the quarter-chord moment.

The secondary visualization for the quarter-chord moment coefficient that plots it as a function of the coefficient of lift similarly shows an increasing quarter-chord moment coefficient for increasing coefficients of lift. A trend line appears by inspection that shows a linearly increasing relationship between the two variables, which is to be expected. Both the coefficient of lift and quarter-chord moment coefficients increase linearly with the angle of attack. Thus, for each of the data points which are themselves dependent on the same angle of attack, a greater coefficient of lift has a correspondingly greater quarter-chord moment coefficient.

6 Conclusion

6.1 Summary

The Airfoil Pressure Distribution lab employed the NACA 0015 airfoils with 29 pressure taps connected to the NetScanner 9116 pressure scanners. The airfoil set within the wind tunnel experienced 45% tunnel speed at varying angles of attack ranging from 0 to 17.5 degrees with 2.5-degree increments. The NetScanner through the pressure taps collected pressure data that was then analyzed and used to plot pressure coefficient curves, lift curves, and drag curves. It can be seen from these curves that the experimentally gathered coefficients of pressure widely match that of the theoretical curves, and that the coefficients of lift and drag as plotted against the angle of attack are what is expected.

6.2 Recommended Improvements

Overall, the experiment saw success in several areas. However, there are still multiple improvements that can be made:

- The best way to decrease the uncertainties in all of the results and remove any noise or outlying data would be to acquire more data. The experiment provides 29 pressure taps to gather the pressure data utilized in this experiment for data analysis and discussion. More of these pressure taps would create more data points that would better illustrate the pressure coefficient curve found in section 4.1 Pressure Coefficients. Increasing the number of pressure taps would help to make the experimentally gathered pressure coefficient graph better match the theoretical data from XFOIL.

- Likewise, allowing the software to run for longer would provide more data points at each of the pressure taps and ensure that there were fewer outlying data points that would affect the pressure coefficients. Uncertainty, noise, and outlying data points can all serve to make the data seem worse than it is. In this case, running the NetScanner software for more than 10 seconds would provide an excess of data points to ensure that noise and outlying data points would not affect the mean pressure data as much.

- In addition to increasing the data gathered for the benefits discussed above, the NetScanner software used in the experiment had difficulty converting the .dat files into usable .csv files. The data when collected was recorded as .dat files and needed to be individually converted from that initial file type to a usable file type, each outside of the software. An issue would occur when sequentially exporting the .dat files as .csv files, the NetScanner software would crash and the process would need to be redone for that angle of attack test. If this process could be done from within the software, it would decrease the time necessary to obtain readable data and mitigate the crashing issue.

Together these improvements would make for a smoother experiment with more robust results.

References

[1] Drela, M., "XFOIL: Subsonic Development System," Massachusettes Institute of Technology, John Wiley. Xfoil 6.99 (Unix, Windows), Dec 23, 2013.

[2] AME 30333, *Lab 2: Circulation around a 2-D Airfoil Measured with a Five-Hole Probe*, University of Notre Dame, Notre Dame, IN, 2024.

[3] AME 30333, *Lab 3: Airfoil Pressure Distribution*, University of Notre Dame, Notre Dame, IN, 2024.

[4] TE Connectivity, *ETHERNET INTELLIGENT PRESSURE SCANNER: 9116 NetScanner System*, Sensor Solutions, 2017.

Appendix A - MATLAB Data Analysis Code

4/9/24 12:43 AM C:\Users\tjwel\Desktop...\dataAnalysis.m 1 of 7

```
% Lab # Analysis

% by Timothy Welch

clear; clc; close all

% Colors for Plots
plotColors{1} = '#0000FF'; % blue
plotColors{2} = '#FF0000'; % red
plotColors{3} = '#7E2F8E'; % purple
plotColors{4} = '#77AC30'; % green
plotColors{5} = '#FFA500'; % orange

%% Load Data
fileNames = {'45_10s_0Cleaned.csv','45_10s_2point5Cleaned.csv','45_10s_5.↙
csv','45_10s_7point5.csv','45_10s_10.csv','45_10s_12point5.csv','45_10s_15Cleaned.↙
csv','45_10s_17point5Cleaned.csv'};
% Load Data: P(AoA,transducer#) [psi]
for i=1:length(fileNames)
    data = table2array(readtable(fileNames{i}));
    for j=1:32
        if class(data) == 'struct'
            ePDrift(i,j) = 0.5*(abs(max(data.data(:,j))-min(data.data(:,j)))); % [psi]
            P(i,j) = mean(data.data(:,j)); % [psi]
        else
            ePDrift(i,j) = 0.5*(abs(max(data(:,j))-min(data(:,j)))); % [psi]
            P(i,j) = mean(data(:,j)); % [psi]
        end
    end
end

P = P*6894.76; % [Pa]
P = [P(:,17:32),P(:,1:16)]; % P1-P28=T1-T28; P29=T0; P31=Pinf; P32=Ptot
Pinf = P(:,31);
Ptot = P(:,32);
P = [P(:,29),P(:,1:28)]; % P(n) = T(n-1)

ePReading = 2.5*0.00003*6894.76; % [Pa]
ePSys = 2.5*0.0015*6894.76; % [Pa]
ePDrift = ePDrift*6894.76; % [Pa]
eP = sqrt(ePSys^2+ePReading^2+ePDrift.^2);
eP = [eP(:,17:32),eP(:,1:16)];
ePinf = eP(:,31);
ePtot = eP(:,32);
eP = [eP(:,29),eP(:,1:28)];

alpha = 0:2.5:17.5; % [deg]
ealpha = 0.5*pi/180; % [rad]

x = [0,0.84,3.33,7.46,13.17,20.42,29.11,39.14,50.42,62.82,76.20,90.41,...
```

```
     105.31,120.71,136.47,0.84,3.33,7.46,13.17,20.42,29.11,39.14,50.42,...
     62.82,76.20,90.41,105.31,120.71,136.47]; % [mm]
y = [0,-2.44,-4.68,-6.71,-8.45,-9.85,-10.83,-11.35,-11.39,-10.96,...
     -10.09,-8.80,-7.15,-5.15,-2.87,2.44,4.68,6.71,8.45,9.85,10.83,...
     11.35,11.39,10.96,10.09,8.80,7.15,5.15,2.87]; % [mm]

c = 152.4; % [mm]
t = 22.86; % [mm]

xC = x/c;
yC = y/c;

%% Ambient Conditions
Ratm = 287; % [J/kg/K]
Tbar = 0.5*(25.5+25.1); % [*C]
Tatm = Tbar+273.15; % [K] conversion from *C
Pbar = 0.5*(996+992); % [hPa]
Patm = Pbar*100; % [Pa] conversion from hPa
ePatm = sqrt(100^2+400^2+0^2); % [Pa]
eTatm = sqrt(0.1^2+0.4^2+0.2^2); % [K]
rho = Patm/(Ratm*Tatm); % [kg/m^3]
erho = sqrt((ePatm/(Ratm*Tatm))^2+(-eTatm*Patm/(Ratm*Tatm^2))^2); % [kg/m^3]

%% Cp Calculations
Cp = (P-Pinf)./(Ptot-Pinf);
eCp = sqrt((eP./(Ptot-Pinf)).^2 +...
           (ePinf.*((P-Ptot)./(P-Pinf).^2)).^2 +...
           (ePtot.*((Pinf-P)./(Ptot-Pinf).^2)).^2);

XFOILData = load('XFOILData.csv');
XFOILx = XFOILData(:,1);
XFOILy = XFOILData(:,2);
XFOILCp = XFOILData(:,3:end);

for i=1:length(alpha)
    figure
    plot(xC(1:15),Cp(i,1:15),'bo',xC(16:end),Cp(i,16:end),'ro',...
        XFOILx,XFOILCp(:,i),'k--','Linewidth',1.5)
    hold on
    errorbar(xC(1:15),Cp(i,1:15),mod(eCp(i,1:15),0.1*min(Cp(i,:))),...
        'vertical','bo','Linewidth',1.5)
    errorbar(xC(16:end),Cp(i,16:end),mod(eCp(i,16:end),0.1*min(Cp(i,:))),...
        'vertical','ro','Linewidth',1.5)
    hold off
    set(gca,'fontSize',16,'YDir','reverse')
    grid on
    legend("Lower Side Measured C_p at "+num2str(alpha(i))+"^o",...
        "Upper Side Measured C_p at "+num2str(alpha(i))+"^o",...
        "XFOIL C_p at "+num2str(alpha(i))+"^o",'Location','northeast')
    xlabel('x/c')
```

```
    ylabel('Pressure Coefficient, C_p')
end
% close all

%% Discretized Coefficients
cn = zeros(length(alpha),1);
ca = zeros(length(alpha),1);
cm = zeros(length(alpha),1);

ecn = zeros(length(alpha),1);
eca = zeros(length(alpha),1);
ecm = zeros(length(alpha),1);

CpLow = Cp(:,2:15);
CpUp = Cp(:,16:end);
xCLow = xC(:,2:15);
xCUp = xC(:,16:end);
yCLow = yC(:,2:15);
yCUp = yC(:,16:end);

for j=1:length(alpha)
    cn(j) = 0.5*(Cp(j,1)+CpLow(j,1))*(xCLow(1))-0.5*(Cp(j,1)+CpUp(j,1))*(xCUp(1));
    ca(j) = 0.5*(Cp(j,1)+CpLow(j,1))*(yCLow(1))-0.5*(Cp(j,1)+CpUp(j,1))*(yCUp(1));
    cm(j) = (0.5*(Cp(j,1)*xC(1)+CpLow(1)*xCLow(1))*xCLow(1) +...
          0.5*(Cp(j,1)*yC(1)+CpLow(j,1)*yCLow(1))*yCLow(1)) - ...
          (0.5*(Cp(j,1)*xC(1)+CpUp(j,1)*xCUp(1))*xCUp(1) +...
          0.5*(Cp(j,1)*yC(1)+CpUp(j,1)*yCUp(1))*yCUp(1));
    for i=2:length(CpLow(j,:))-1
        cn(j) = cn(j) + 0.5*(CpLow(j,i)+CpLow(j,i+1))*(xCLow(i+1)-xCLow(i))-0.5*(CpUp↙
(j,i)+CpUp(j,i+1))*(xCUp(i+1)-xCUp(i));
        ca(j) = ca(j) + 0.5*(CpLow(j,i)+CpLow(j,i+1))*(yCLow(i+1)-yCLow(i))-0.5*(CpUp↙
(j,i)+CpUp(j,i+1))*(yCUp(i+1)-yCUp(i));
        cm(j) = cm(j) + (0.5*(CpLow(j,i)*xCLow(i)+CpLow(j,i+1)*xCLow(i+1))*(xCLow(i+1)-↙
xCLow(i)) +...
                        0.5*(CpLow(j,i)*yCLow(i)+CpLow(j,i+1)*yCLow(i+1))*(yCLow(i+1)-↙
yCLow(i))) -...
                        (0.5*(CpUp(j,i)*xCUp(i)+CpUp(j,i+1)*xCUp(i+1))*(xCUp(i+1)-xCUp↙
(i)) +...
                        0.5*(CpUp(j,i)*yCUp(i)+CpUp(j,i+1)*yCUp(i+1))*(yCUp(i+1)-yCUp↙
(i)));
    end
    ecn(j) = sqrt(sum(eCp(j,:)*(xC(2)-xC(1))^2));
    eca(j) = sqrt(sum(eCp(j,:)*(xC(2)-xC(1))^2));
    ecm(j) = sqrt(sum(eCp(j,:)*(xC(2)-xC(1))^2));
end

T1 = array2table([alpha',cn,ca,cm],'VariableNames',{'Angle of↙
Attack','c_n','c_a','c_m',});
table2latex(T1,'cCoeffs1')
```

```
eT1 = array2table([alpha',ecn,eca,ecm],'VariableNames',{'Angle of↙
Attack','c_n','c_a','c_m',});
table2latex(T1,'ecCoeffs1')

%% Lift, Drag, Quarter Chord
cl = cn.*cosd(alpha')-ca.*sind(alpha');
cd = cn.*sind(alpha')+ca.*cosd(alpha');
cq = cm+0.25*cl;

ecl = sqrt((ecn.*cosd(alpha')).^2+(ealpha'.*(-cn.*sind(alpha')-ca.*cosd(alpha'))).^2+↙
(eca.*sind(alpha')).^2);
ecd = sqrt((ecn.*sind(alpha')).^2+(ealpha*(cn.*cosd(alpha')-ca.*sind(alpha'))).^2+(eca.↙
*cosd(alpha')).^2);
ecq = sqrt(ecm.^2+(0.25*ecl).^2);

T2 = array2table([alpha',cl,cd,cq],'VariableNames',{'Angle of↙
Attack','c_l','c_d','c_mc4',});
table2latex(T2,'cCoeffs2')

eT2 = array2table([alpha',ecl,ecd,ecq],'VariableNames',{'Angle of↙
Attack','c_l','c_d','c_mc4',});
table2latex(T2,'ecCoeffs2')

%% Lift Coeff vs AoA
m0 = pi^2/90; % [/deg]
figure
scatter(alpha,cl,100,'bsquare','Filled')
hold on
plot(alpha,m0*alpha,'k-','Linewidth',1.5)
errorbar(alpha,cl,ecl,'vertical','bsquare','Linewidth',1.5)
errorbar(alpha,cl,ealpha*180/pi,'horizontal','bsquare','Linewidth',1.5)
hold off
set(gca,'fontSize',16)
grid on
xlabel(['Angle of Attack, \alpha (' char(176) ')'])
ylabel('Coefficient of Lift, c_l')
legend('Experimental c_l','Theoretical Lift-Curve')
% axis([-2.5,17.5,-0.2,2])

%% Drag Coeff Plots
figure
scatter(alpha,cd,100,'ro','Filled')
hold on
errorbar(alpha,cd,ecd,'vertical','ro','Linewidth',1.5)
errorbar(alpha,cd,ealpha*180/pi,'horizontal','ro','Linewidth',1.5)
hold off
set(gca,'fontSize',16)
grid on
xlabel(['Angle of Attack, \alpha (' char(176) ')'])
ylabel('Coefficient of Drag, c_d')
```

```
% axis([-2.5,17.5,-0.2,2])

figure
scatter(cl,cd,100,'bd','Filled')
hold on
errorbar(cl,cd,ecd,'vertical','bd','Linewidth',1.5)
errorbar(cl,cd,ecl,'horizontal','bd','Linewidth',1.5)
hold off
set(gca,'fontSize',16)
grid on
xlabel('Coefficient of Lift, c_l')
ylabel('Coefficient of Drag, c_d')
% axis([-2.5,17.5,-0.2,2])

%% Quarter Chord Coeff Plots
figure
scatter(alpha,cq,100,'mpentagram','Filled')
hold on
errorbar(alpha,cq,ecq,'vertical','mpentagram','Linewidth',1.5)
errorbar(alpha,cq,ealpha*180/pi,'horizontal','mpentagram','Linewidth',1.5)
hold off
set(gca,'fontSize',16)
grid on
xlabel(['Angle of Attack, \alpha (' char(176) ')'])
ylabel('Moment Coefficient Quarter Chord')
% axis([-2.5,17.5,-0.2,2])

figure
scatter(cl,cq,100,'k^','Filled')
hold on
errorbar(cl,cq,ecq,'vertical','k^','Linewidth',1.5)
errorbar(cl,cq,ecl,'horizontal','k^','Linewidth',1.5)
hold off
set(gca,'fontSize',16)
grid on
xlabel('Coefficient of Lift, c_l')
ylabel('Moment Coefficient Quarter Chord')
% axis([-2.5,17.5,-0.2,2])

%% Function(s)
function table2latex(T, filename)

    % Error detection and default parameters
    if nargin < 2
        filename = 'table.tex';
        fprintf('Output path is not defined. The table will be written in %s.\n',↙
filename);
    elseif ~ischar(filename)
        error('The output file name must be a string.');
    else
```

```
        if ~strcmp(filename(end-3:end), '.tex')
            filename = [filename '.tex'];
        end
    end
    if nargin < 1, error('Not enough parameters.'); end
    if ~istable(T), error('Input must be a table.'); end

    % Parameters
    n_col = size(T,2);
    col_spec = [];
    for c = 1:n_col, col_spec = [col_spec 'l']; end
    col_names = strjoin(T.Properties.VariableNames, ' & ');
    row_names = T.Properties.RowNames;
    if ~isempty(row_names)
        col_spec = ['l' col_spec];
        col_names = ['& ' col_names];
    end

    % Writing header
    fileID = fopen(filename, 'w');
    fprintf(fileID, '\\begin{tabular}{%s}\n', col_spec);
    fprintf(fileID, '%s \\\\ \n', col_names);
    fprintf(fileID, '\\hline \n');

    % Writing the data
    try
        for row = 1:size(T,1)
            temp{1,n_col} = [];
            for col = 1:n_col
                value = T{row,col};
                if isstruct(value), error('Table must not contain structs.'); end
                while iscell(value), value = value{1,1}; end
                if isinf(value), value = '$\infty$'; end
                temp{1,col} = num2str(value);
            end
            if ~isempty(row_names)
                temp = [row_names{row}, temp];
            end
            fprintf(fileID, '%s \\\\ \n', strjoin(temp, ' & '));
            clear temp;
        end
    catch
        error('Unknown error. Make sure that table only contains chars, strings or↙
numeric values.');
    end

    % Closing the file
    fprintf(fileID, '\\hline \n');
    fprintf(fileID, '\\end{tabular}');
    fclose(fileID);
```

```
end
```