Aerodynamic Performance of an Airfoil and Wake Analysis

Wyatt O. Welch[1]
San Diego State University, San Diego, CA

The purpose of this lab is to investigate the characteristics of a NACA 43012A airfoil within a wind tunnel at a constant speed, with recordings of surface pressure measurements and wake rake measurements at varying Angles of Attack. With this data, the experimenter aims to calculate lift, drag, and pitching moment coefficients, and plot the data to best describe the behavior. This will be compared with theoretical NACA data published from NACA Report 610, plotting the pressure coefficient flow distribution at each angle of attack [-5, 0, 5, 10, 15, 20]°. The wake will be compared with drag data, as well as with the wake velocity profiles. The outcomes of this lab will be to understand real world applications of Aerodynamic Principles, such as Coefficient derivation and Form drag vs. Skin drag. There are also real-world applications, and comparing results to theoretical data will ensure the process is followed correctly.

Nomenclature

T_{amb} = Ambient Temperature
P_{amb} = Ambient Pressure
R = Gas Constant
ρ = Air Density
q_∞ = Dynamic Pressure
P_{total} = Total Pressure
P_{static} = Static Pressure
U_∞ = Freestream Velocity
C_p = Pressure Coefficient
P_{local} = Local Pressure at a port
C_n = Normal Force Coefficient
C_a = Axial Force Coefficient
α = Angle of Attack
C_l = Lift Coefficient
C_d = Drag Coefficient
$\frac{dy_u}{dx}, \frac{dy_l}{dx}$ = Slope of upper and lower airfoil surfaces
$C_{m,ac}$ = Moment Coefficient about Aerodynamic Center
C_{mLE} = Moment Coefficient about Leading Edge
x_{ac}, y_{ac} = Coordinates of Aerodynamic Center
D' = Momentum Deficit Drag per Unit Span
u_2 = Local Velocity in the Wake
$C_{d,wake}$ = Drag Coefficient via Jones' Method
q_2 = Dynamic pressure in the wake
Re = Reynolds Number
c = Chord Length

[1] Aerospace Engineering Student, School of Engineering,

I. Introduction

THIS report is to measure the aerodynamic forces on an airfoil using a number of Pitots, and to then calculate and interpret the data against theoretical data from the NACA airfoil study. This will allow the students to investigate the variations in flow based on Angle of Attack and understand the concepts behind common coefficients used in the field. This lab was performed on SDSU grounds in the SDSU Low Speed Wind Tunnel Lab, by our TAs with the instruction of our professor. Thank you to the school, and faculty, for allowing us to learn through this lab.

II. Theory

The concepts, equations, and theory behind this lab seem daunting at first glance, but in reality, this is not a complex set of equations to understand, and with some explanation it will be easy to see how they are used in the wider context of the world. To start, we must calculate the Air Density using the ambient pressure and temperature. This, in conjunction with the Gas Constant, is an alternative form of the Perfect Gas Equation, which is assumed for the purposes of this lab.

$\rho = \frac{P_{amb}}{R \cdot T_{amb}}$ eq. 1

Density is derived from Perfect Gas equation for incompressible flow and is key for finding the freestream velocity. This would also be useful for Reynolds number if needed. It affects all forces and moments. The next equation to cover is Dynamic Pressure, which is the difference between the total and static pressures measured within the wind tunnel. It represents the kinetic energy per unit volume within the airflow.

$$q_\infty = P_{total} - P_{static} \quad (1)$$

This is useful for normalizing pressure coefficients and relates to lift and drag directly. Next, we put the previous two equations to use with the Freestream Velocity. This determines the flow speed of the fluid in the test section. This is useful for understanding what our Coefficients will mean in context of the experiment.

$$U_\infty = \sqrt{\left(\frac{2q_\infty}{\rho}\right)} \quad (2)$$

This also sets the Reynolds number, with density, and can be used to calibrate the tunnels speed, when compared with the q value in (inH2O). Now we are moving onto our first Coefficient. This one is quite useful, especially within this lab, since it can be found for every spot on the airfoil rather than as an overall value. You guessed it, the Pressure Coefficient! This handy tool nondimensionalizes surface pressure measurements. Without it, you would need to ask for much more information from someone to understand how their airfoil behaves. This compares local pressure to freestream (dynamic) pressure.

$$C_p = \frac{P_{local} - P_{static}}{q_\infty} \quad (3)$$

Where P_{local} is the pressure value at the pitot in question. This is used to show the acceleration and deacceleration over the airfoil when compared with other values, and is critical for lift/drag predictions, as you will soon see. The Normal Force Coefficient formula is derived in the handout and is used to calculate the net force perpendicular to the chord line.

$$C_n = \int_0^1 C_{p,lower} dx - \int_0^1 C_{p,upper} dx \quad (4)$$

This equation takes the area under the C_p under the airfoil and subtracts it with the C_p above the airfoil, then nondimensionalizes it. By using this formula, we neglect skin friction for thin airfoils at small α. This is a large contributor to lift, and is tied directly to pressure asymmetry, meaning it's the reason there is a greater pressure under the airfoil, which generates said lift. The Axial Force Coefficient is slightly more complex.

$$C_a = \int_0^1 C_{p,upper} \frac{dy_u}{dx}\, dx - \int_0^1 C_{p,lower} \frac{dy_l}{dx}\, dx \quad (5)$$

This works essentially the same as the normal force, but with switched signs for each, and accounts for pressure acting on the inclined surface, which means $\frac{dy}{dx}$ is the slope on either side of the airfoil. This calculates the force parallel to the chord line and is the major drag contributor. The next two are similar enough to be pair together, since their main difference lies in the trigonometry needed to derive it.

$$C_l = C_n cos\alpha - C_a sin\alpha \quad (6)$$

$$C_d = c_n sin\alpha + C_a cos\alpha \quad (7)$$

For the Lift coefficient, it converts body axis forces to lift and is the normal and axial coefficients times the trigonometric function and the angle of attack. The same goes for the Drag coefficient, with the difference being a switched sin and cosine, and adding the forces instead of subtracting, since the two work together for drag as opposed to lift. The final equation is a complex one but essentially has the nondimensionalized distance from the aerodynamic

center times the lift and drag coefficients and the angle of attack similar to the previous two equations. There is also a moment coefficient about the leading edge that is added as well, but that is simply calculated as the normal force times the horizontal distance, and the axial with the vertical distance. Adding these together gives us our next equation:

$$C_{m,ac} = C_{mLE} + C_l \frac{x_{ac}}{c} cos\alpha - C_d \frac{y_{ac}}{c} cos\alpha + C_l \frac{y_{ac}}{c} sin\alpha + C_d \frac{x_{ac}}{c} sin\alpha \quad (8)$$

This equation measures the tendency of the airfoil to pitch about the aerodynamic center. This is useful for stability analysis and also influences the control surface in its design. For the rake, there are two equations worth looking at, the first is the Momentum Deficit Drag, which is per unit span. It is based on the Momentum Theorem, where u_2 is the local velocity is in the wake.

$$D' = \rho \int_{-\frac{b}{2}}^{\frac{b}{2}} u_2(u_\infty - u_2)dy \quad (9)$$

This assumes 2D, incompressible, steady flow in the system, and measures the total drag without the surface instrumentation, which has countless applications. The second is the Drag Coefficient via Jones' Method. This equation nondimensionalizes the wake drag force which can then be used to compare with theory and other airfoils.

$$C_{d,wake} = \frac{2}{c} \int \left(\sqrt{\frac{q_2}{q_\infty}} - \frac{q_2}{q_\infty} \right) dy \quad (10)$$

These equations together give us enough background to complete the lab and present our data with full knowledge of what is happening in the wind tunnel and why.

Example calculations are a helpful tool in showing the instructor that the data has been properly understood and used to the best of its ability. From the equations described earlier in this section, and with predefined constants and values gathered from this lab, the following are the sample calculations from the initial pressure readings to the Coefficient of Moment about the Aerodynamic Center, with all Coefficients needed to calculate it within:

Constants

$P_{amb} = 30.11\ inHg = 101{,}900\ Pa$
$T_{amb} = 79.5°F = 299.5\ K$
$R = 287 \frac{J}{kg \cdot K}$

Airfoil Geometry

$C = 1$ (normalized)
$(x_{ac}, y_{ac}) = (0.238c, 0.07c)$
$Lower: Port\ 9\ (\frac{x}{c} = 0.306, \frac{y}{c} = -0.0353)$
$Upper: Port\ 26\ (\frac{x}{c} = 0.30, \frac{y}{c} = 0.0857)$

Lab Data

$P_{static} = -353.0\ Pa$
$P_{total} = 888.3\ Pa$
$P_9 = -867.4\ Pa$
$P_{26} = -743.8\ Pa$

$$\rho = \frac{P_{amb}}{R \cdot T_{amb}} = \frac{101{,}900}{287 \cdot 299.5} = 1.184 \frac{kg}{m^3}$$

$$q_\infty = P_{total} - P_{static} = 888.3 - (-353.0) = 1{,}241\ Pa$$

$$U_\infty = \sqrt{\left(\frac{2q_\infty}{\rho}\right)} = \sqrt{\frac{2 \cdot 1{,}241}{1.184}} = 45.80 \frac{m}{s}$$

$$C_{p,9} = \frac{P_9 - P_{static}}{q_\infty} = \frac{-867.4 - (-353.0)}{1{,}241} = -0.4145$$

$$C_{p,26} = \frac{P_{26} - P_{static}}{q_\infty} = \frac{-743.8 - (-353.0)}{1{,}241} = -0.3149$$

$$\frac{dy_l}{dx} \approx \frac{-0.0353 - (-0.0325)}{-.300 - 0.250} = -0.0571$$

$$\frac{dy_u}{dx} \approx \frac{0.0857 - 0.0922}{0.300 - 0.250} = -0.130$$

$$C_n = \int_0^1 C_{p,lower} dx - \int_0^1 C_{p,upper} dx = -0.2485$$

$$C_a = \int_0^1 C_{p,upper} \frac{dy_u}{dx}\, dx - \int_0^1 C_{p,lower} \frac{dy_l}{dx}\, dx = 0.0186$$

$$C_l = C_n cos\alpha - C_a sin\alpha = -0.2485$$

$$C_d = c_n sin\alpha + C_a cos\alpha = 0.0186$$

$$C_{m,ac} = C_{mLE} + C_l \frac{x_{ac}}{c} cos\alpha - C_d \frac{y_{ac}}{c} cos\alpha + C_l \frac{y_{ac}}{c} sin\alpha + C_d \frac{x_{ac}}{c} sin\alpha = -0.0132$$

$$D' = \rho \int_{-\frac{b}{2}}^{\frac{b}{2}} u_2(u_\infty - u_2) dy = 113.1 \frac{N}{m}\ Across\ rake$$

$$C_{d,wake} = \frac{2}{c} \int \left(\sqrt{\frac{q_2}{q_\infty}} - \frac{q_2}{q_\infty} \right) dy = 0.0186$$

Coefficient	Value
C_l	-0.2485
C_d	-0.0186
$C_{m,ac}$	-0.0132
$C_{d,wake}$	0.0186

Table 1: Sample Calculation Results

III. Apparatus, Setup, and Procedure

The apparatuses to be used are many and complex between them. To start, there is the wind tunnel. This is a closed return subsonic wind tunnel located in the San Diego State University Engineering building. It has a test section volume of 45"x32"x67", and can reach speeds around 0-180 mph. It is propelled by a 4-blade propeller and contains pitot static pressure ports inside the test section and is capable of measuring the dynamic pressure with the ports installed in the wind tunnel itself.

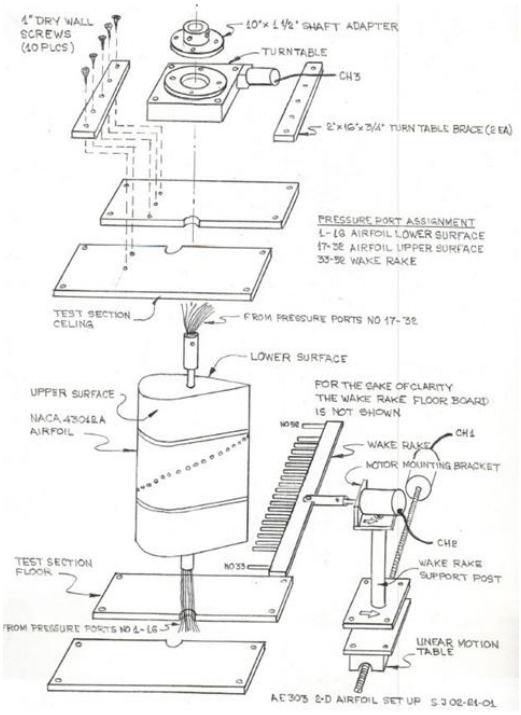

Figure 1: Detailed blueprint for assembly of airfoil and Wake Rake

Figure 2: Assembled in wind tunnel NACA43012A and Wake Rake

The data recorded in the lab is gathered using a Scanivalve ZOC33 Miniature Pressure Scanner. This allows dozens of pressure ports. Paired with the DSM4000, which interfaces between the scanner and the computer, allows for the data to be recorded and saved to a spreadsheet with minimal effort. To measure the speed within the wind tunnel, there is a Meriam 34FB8 Water Manometer, which measures the pressure differences in the wind tunnel in units of inH2O and has a max pressure reading of 10". Outside of the wind tunnel, for ambient pressure and temperature readings there are digital barometers and thermometers, which are used for every lab at every testing.

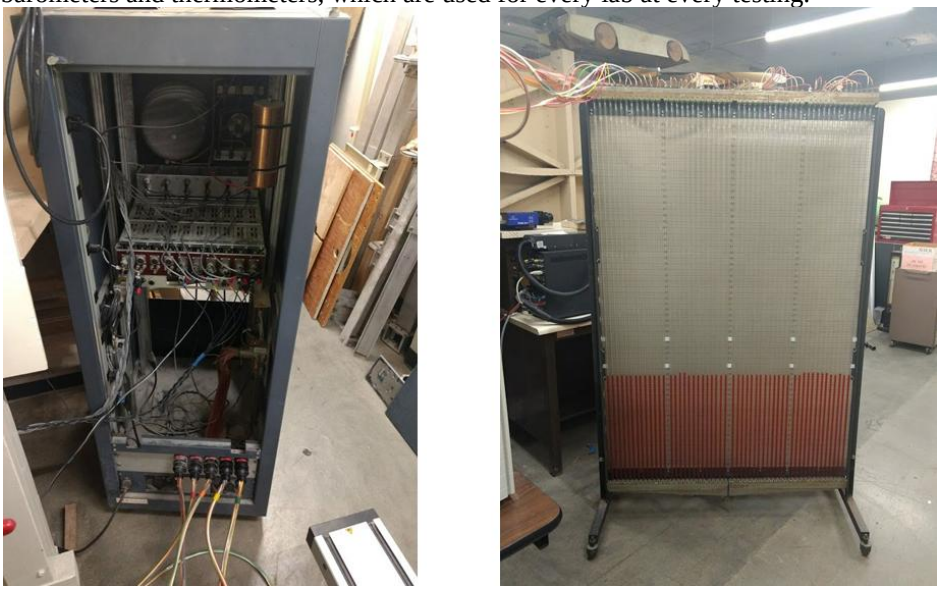

Figure 3: On left, ZOC33 Scanivalve with 64 ports, on right, Water manometer with 52 ports

The computer software is controlled through a ScanTel Program, which commands the DSM4000 as well as logs the pressure data. For this lab, a Compumotor Program is used in order to control the airfoil within the test area, and change the angle of attack precisely, as well as to reposition the rake as needed. This is a huge part of the lab and allows us to analyze the airfoil's behavior at different angles of attack.

The set up for this procedure is as follows:

Setup

- Verify test section is clear.
- Place airfoil, rake into test section.
- Connect pitot static tubes to the airfoil and rake, ensure locked positions.
- Connect pitot static tubes to pressure scanners.
    - Leave ports #1-2 for dynamic/static pressures.
- Attach inclinometer to wake rake for angle calibration.
- Close and lock doors after ensuring the entire test section is ready for test.
- Leave one port open to atmosphere for reference (26)
- Power on ZOC33 and DSM4000 and allow 20 minutes for warmup.
- Launch ScanTel software and configure to record and save all 52 pressure ports.

Figure 4: Assembled Wake Rake with installed pitots

Test

- Record baseline test for q=0, AoA=0
    - Verify rake is vertical.
- Raise q to 5” H2O, set data to record.
- Set airfoil to desired AoA.
- Confirm airfoil angle matches desired angle per table.
- Adjust wake rake to specified angle.
- Allow 2-3 minutes for stabilization after setting speed before recording.
- Collect 4000 samples using Scanivalve.
- Repeat steps with various AoA until reaching 20 degrees.

Cleanup

- Return AoA to 0, and rake to vertical.
- Power down all recording software
- Export data for analysis.
- Clean the interior test area and take down any materials not required to keep in test area.

IV. Results

The work shown by previous sections does well to set up all the data we are to be analyzing within this section. Because we have already gone over sample calculations, we will not take the time to derive how these were created, since it is all a part of that previous section, as well as previous labs, making it trivial to show, as is the case with the Reynolds number. In Table 2, we have the Reynolds number for each angle of attack, represented in 5 significant figures, and with a preset multiplier of x10^6 for simplicity of viewing. With this we can see how accurate our flow calculations are.

Angle of Attack	Reynolds Number ($x10^6$)
-5	2.9054
0	2.9043
5	2.9047
10	2.9042
15	2.9050
20	2.9055

Table 2: Reynolds Number for each test

From the data, we can gleam the flow regimes across all the tests, and by taking the average, we can see that it lands at about 2.905. The variation in this flow is also extremely low, sitting at under 0.05%. This makes sense given that these measurements were all measured at q = 5inH2O, so the difference in the flow would be minimal and caused uncontrolled error, which will be discussed in the conclusion. This was calculated with C=1, since we are dealing with nondimensionalized figures in most cases throughout this lab.

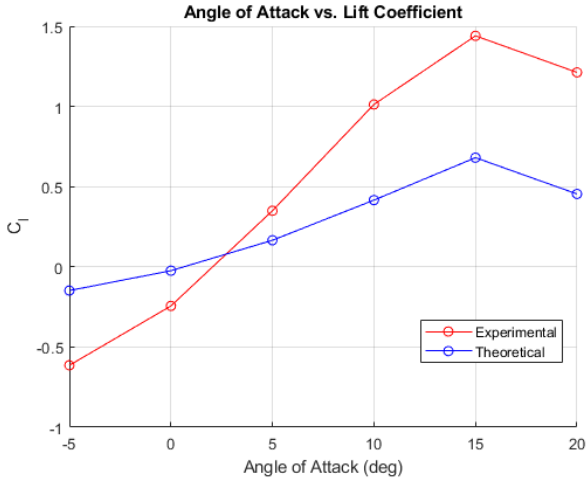

Figure 5: Experimental and Theoretical Lift Coefficients vs. Angle of Attack

With our first plot, we look at the Experimental and NACA theoretical Lift Coefficients at different angles of attack. It is clear to see that these follow a trend as you increase the angle of attack, so too does the coefficient of lift. This trend continues until roughly 20 degrees, at which point it begins to decrease. This makes sense conceptually, as you lift the fin, you create a larger angle that contributes more toward the normal forces, which produce most of the lift in an airframe.

The decline in lift after 15 degrees can also be explained by reasoning, as at some point in your flight you reach diminishing returns with your lift. At some point the airflow stops pushing up your wing and simply starts to push it. In real life, this would quickly lower your speed through extreme drag and cause all kinds of turbulence in the trialing edges, which we will be able to see later in the section. It is clear at least that the Experimental matches the Theoretical in most respects, the experimental seems to be climbing higher, which could be from an improperly set speed or a number of other variables. Again, this can and will be discussed later in the paper.

With an agreement in values at the lower Angles, it can validate the attached flow that would be seen from small angles of attack, and at higher angles, as the differences between the two lines begin to diverge, you can see an indication for flow separation, which comes from more turbulent behaviors. The theoretical curves may have assumed a more inviscid flow, which would dampen the effects of stall.

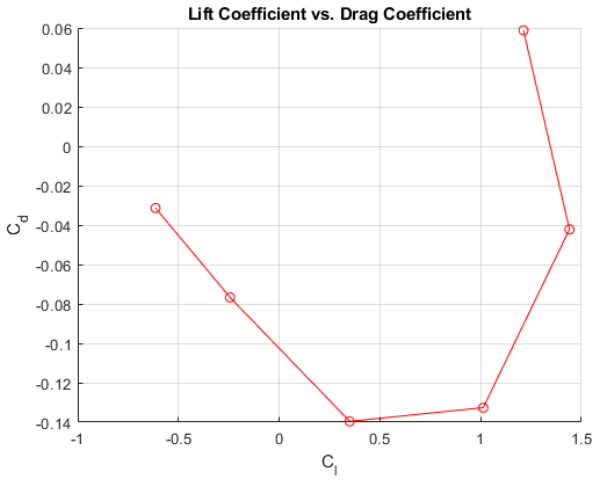

Figure 6: Experimental Lift Coefficient vs. Drag Coefficient for all Angles of Attack

Moving away from the theoretical, we have the lift and drag coefficients facing each other. It is sort of hard to see a trend here, except in one place. Drag seems to decrease when the lift is in the negative and begin quickly increasing afterward. The use of this graph is to have a look at the Lift to Drag ratio which is so greatly important for analyzing the usefulness of airfoils.

The numbers themselves seem to look good. For low angles of attack, it makes sense for the coefficient of drag to be small, since there would be little surface area to cause that drag. The sharp increase of drag at the stall is a sign of flow separation, which makes sense at that high of an angle. More surface areas, more turbulent air, and less stable boundary conditions are all likely to cause issues in the flow, which would increase drag throughout the system.

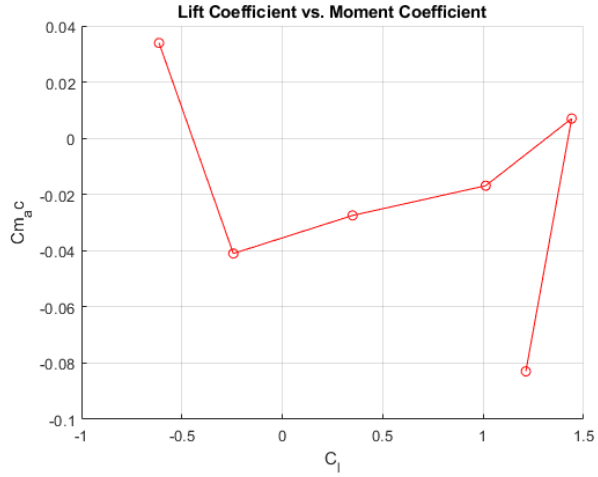

Figure 7: Experimental Lift Coefficient vs. Moment Coefficient for all Angles of Attack

By comparing the Lift coefficient and the Moment coefficient about the Aerodynamic center, which was provided in the documentation, we can see the stability characteristics of the airfoil. To interpret this, we check how much the moments vary with changes in the lift.

For a conventional airfoil, the two should appear nearly constant with each other for low Angles of Attack. As it gets larger, there will be an increase in the moment gradually. This can be seen in the data besides the first and final points. Unfortunately, something has caused the data to seem askew for those two points, but the overall trend is a gradually increasing moment, along with an increasing lift. This can be expected, but the two points being this far off are causes for concern. It is possible that the flow separation disrupted the distribution, but that wouldn't explain the large error for the AoA=-5 test. However, a sudden slope as seen could simply just be a sign of stall, and it isn't out of the question that this is reasonable data for an experiment such as ours.

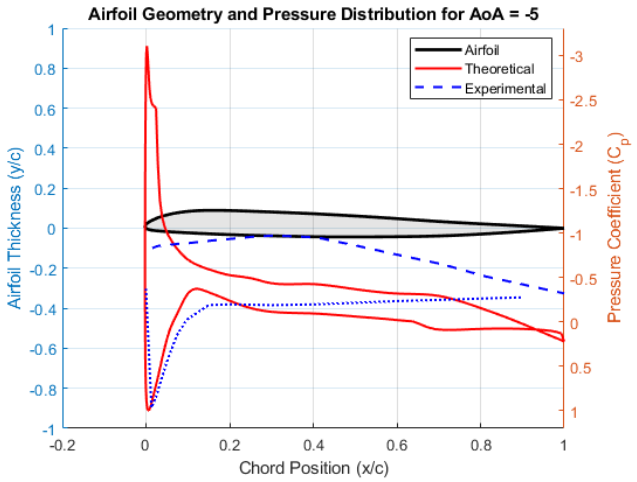

Figure 8: Airfoil with Theoretical and experimental Cp Values for AoA = -5

This graph, and the five to follow are going to be very similar in their analysis, since the only thing that changes is the angle of attack. And while that does lead to very different experimental and theoretical results, the overall headline is the error between the two methods to find the answer.

For AoA=-5, this plot seems to be following the Theoretical very closely. Especially on the lower side of the airfoil, there is a clear drop down at the beginning which makes sense for a downward pointing airfoil. The difference lies almost entirely with the upper edge of the Cp values. The theoretical data has a spike in pressure, which makes sense for downward pointed airfoils, since it is essentially lifting the wing downwards. Unfortunately, the experimental data does not reflect this accurately.

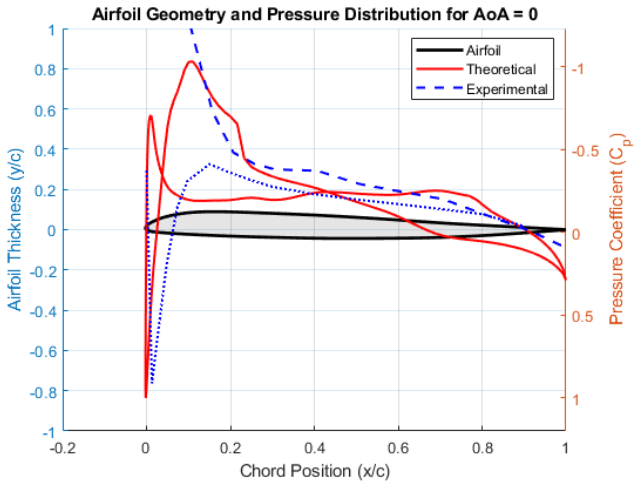

Figure 9: Airfoil with Theoretical and experimental Cp Values for AoA = 0

Once again, the Lower edge of the airfoil is matching almost exactly to the theoretical data. In this case, however, so too is the upper edge. The difference this time is that there is a seemingly much higher experimental peak in the upper edge, but the overall similarities are there, the two ramp up at roughly the same spot on the airfoil, and they decrease to the same point as well. Overall, this test is extremely effective at showing the similarities between theoretical and the experimental data. This is still in the small angles of attack, so it is no surprise that we see such close matches.

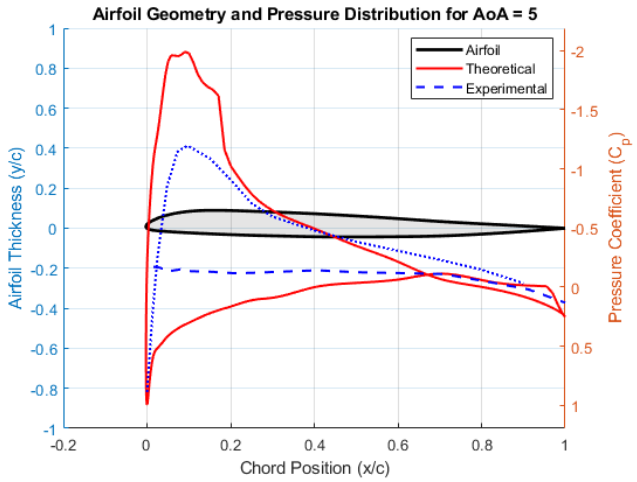

Figure 10: Airfoil with Theoretical and experimental Cp Values for AoA = 5

From this point onwards in this set of 6 figures, it is clear just how accurate the data can be. In this case, there is an issue with the lower edge matching up. It seems there is no pressure jump. This does seem strange with a positive Angle of Attack, since that is what would cause the jump in pressure, however we can take assurance in the accuracy by looking at the upper edge, and how closely it follows the jump with the theoretical.

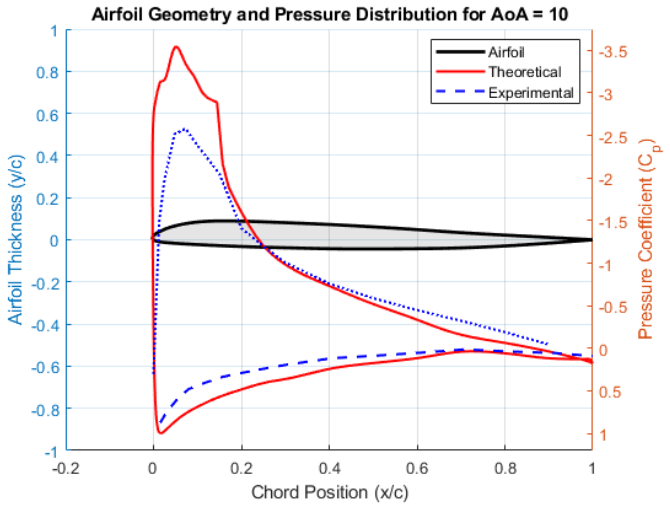

Figure 11: Airfoil with Theoretical and experimental Cp Values for AoA = 10

For this plot, we can see an extremely accurate portrayal of Experimental and Theoretical data being displayed together. There are spots where there should be greater pressure forces, but as spoken of earlier, this is something that can have countless sources of error and should not be closely counted as an issue.

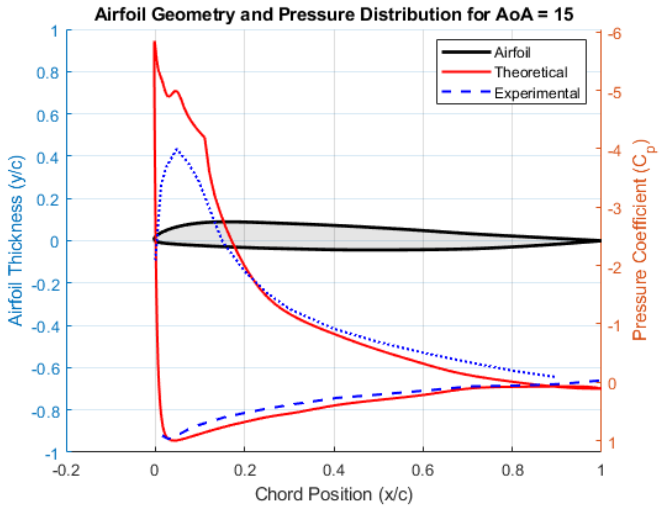

Figure 12: Airfoil with Theoretical and experimental Cp Values for AoA = 15

Figure 12 is likely the most accurate graph in the set, and much like Figure 11, there is not much to say about this one, there is beginning to get a flow separation, which can be seen my the subtle slope of the lower edge as opposed to the rapid drop from more neutral angles of attack. This is because the airfoil is tilting up such that there is a cleaner flow of the fluid over the lower side of the air foil, and a steeper upper section which suggests flow separation.

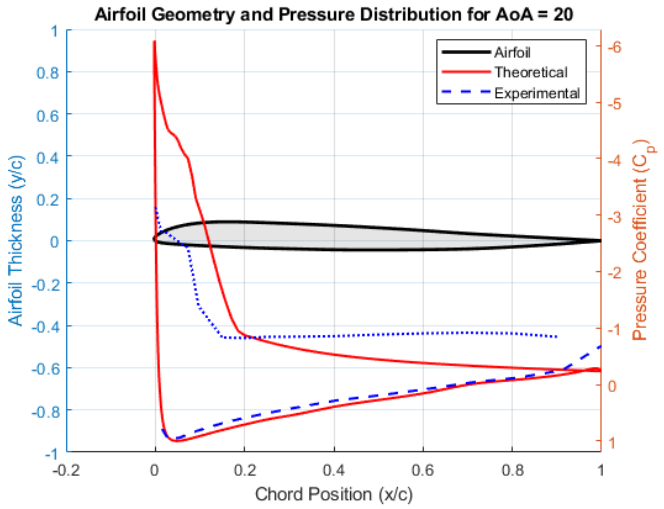

Figure 13: Airfoil with Theoretical and experimental Cp Values for AoA = 20

This figure is the perfect one for showing the flow separation in the data. The steep drop off at the upper edge suggests a flow separation that comes from the steep angle of attack, causing a smooth slope on the lower edge and a steep drop from the gap in the air that drops the pressure after the fluid is forced over the upper lip of the leading edge of the air foil.

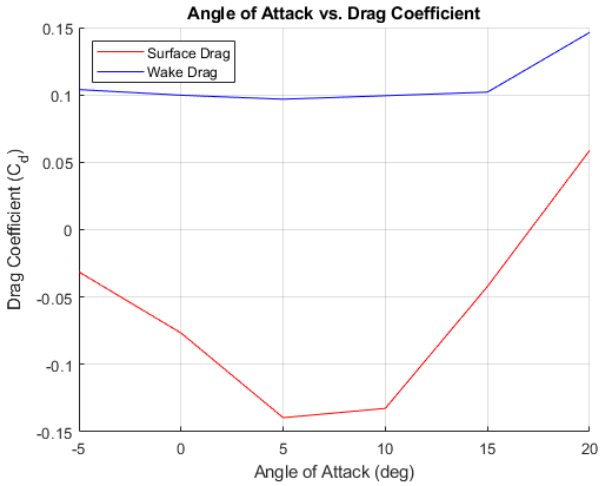

Figure 14: Drag Coefficient vs. Angle of attack, comparing Surface Drag to Wake Drag

Finally moving away from the last set of data, we can look at the Drag coefficient similarities between the Wake Drag and the Surface Drag. It is clear that the two have a gap between them, with one being mostly negatives and the other entirely positive. However, that is where the differences cease.

There is a clear u shape in the data, which suggests that increasing the angle of attack in either direction will increase the drag, and the reason that there is a delay in it changing around 5-10 degrees is not error but is instead based on the airfoil itself. A perfectly neutral airfoil would see this drop and rise take place exclusively on the 0-degree mark, but because it is built to be more effective for lift, there is less drag when increasing the angle of attack, and therefore more efficient to climb. This is a very interesting way to interpret the qualities of the air quality itself.

This work also validates the drag calculation method from earlier sections and data. The trend that the Wake drag is greater than the Surface drag has to do with the fact that wake rakes will have a better chance at capturing the separated flow losses in the system. On top of that, the pitots will not be able to read the turbulence that causes drag after the pitot itself, and even less so at higher angles of attack where the flow separation occurs. This is why the Drag coefficient is greater and matches the trends from the Surface drag.

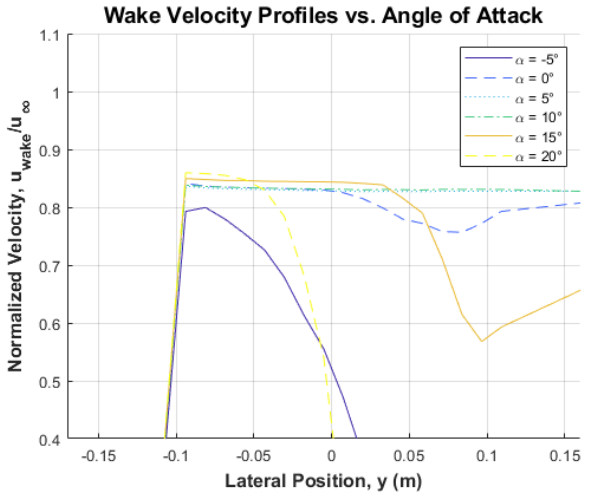

Figure 15: Wake Velocity Profiles for each Angle of Attack

For the final plot in the Results section, we are looking at the velocity profiles for each angle of attack. The readings taken here can be seen to increase in velocity and decrease as it trickles out. For many of the tests, compared to the first test it is clear that increasing the angle of attack increases the flow separation, and makes it take longer to return to zero. This cannot always be seen, however. The small angles of attack can also be seen to hold a constant value. This can be considered correct, since flow separation hasn't kicked in yet, so there's nowhere for the velocity to go but to remain constant.

This graph is honestly dubious and will remain one of my lease convincing data points. But the math seemed to check out so I would have to attribute it to some sort of error. To see where the work was applied, and to potentially figure out where the mistake could be, I would direct you to the Appendix to see the code used for this experiment.

V. Conclusion

This lab was a very interesting one to complete, and I believe that I have managed to show proper relationships between the Reynolds Numbers and the angle of attack, correctly associating that there was minimal variation due to the constant dynamic pressure in the system. I also identified that the Experimental and Theoretical life coefficients aligned closely at small Angles of Attack but diverged as it grew due to flow separation around the airfoil. There were small errors that stacked unfortunately in the moment coefficients graph in Figure 6, but the trend line of a subtle increase was still identifiable. Figures 8-13 were also properly described, with the development of flow separation, and the fundamental understanding of how a change in angle will affect the pressure coefficient above and below an airfoil. The surface drag and wake drag also captured the lower losses by the wake rake.

There were many things that may have gone wrong. While the flow fluctuation was minimal, it may be that the flow wasn't set perfectly, which would cause faster speeds and issues down the line in coefficients that compared speed with other components. There also is the issue of spiraling inconsistencies in calculations. By that, I mean that one small mistake in my many hundreds of lines of code, could easily result in massive mistakes long term, that I would need to spend hours debugging to uncover, If I could even identify an issue in the first place.

For future experiments, I think an easier to control dynamic pressure monitoring setup for the wind tunnel would minimize the largest opportunity for in lab error. Beyond that, I think having more simplified data at the beginning would go a long way to helping avoid issues in the code, since it would be one few aspects of the work to worry about. Overall, though, there isn't too much that needs to be changed. Despite the difficulty, it really is a straightforward lab.

References

1. Liu, X. *2D Airfoil*. San Diego State Univ., San Diego, CA, Feb. 25, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16248169.
2. Liu, X. *About_Xfoil*. San Diego State Univ., San Diego, CA, Mar. 5, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16350451.
3. Liu, X. *AE_303_Lab_4_Updated_Setup*. San Diego State Univ., San Diego, CA, Mar. 12, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16447745.
4. Liu, X. *AE303_Lab4_Airfoil_2025*. San Diego State Univ., San Diego, CA, Mar. 5, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16350375.
5. Liu, X. *AE303_Lab4_Signup_Sheet_TuesdayClass*. San Diego State Univ., San Diego, CA, Mar. 4, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16350128.
6. Liu, X. *Lab 4 Data SP2025 Tuesday*. San Diego State Univ., San Diego, CA, Mar. 4, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16350127.
7. Liu, X. *Lab4_Airfoil_Port_Locations_etc*. San Diego State Univ., San Diego, CA, Mar. 5, 2025. https://sdsu.instructure.com/courses/165529/files/folder/Lab4_Airfoil?preview=16350547.
8. National Advisory Committee for Aeronautics. *NACA Report 6012: Tests of related forward-camber airfoils in the variable-density wind tunnel*. Washington, DC: NACA, .

Appendix

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% AE302 Lab 4 - Wyatt Welch
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc, clear all

% Load Data & Convert to Tables in Pa

load("q0a0.mat"); load("q5aN5.mat"); load("q5a0.mat"); load("q5a5.mat");
load("q5a10.mat"); load("q5a15.mat"); load("q5a20_1.mat");
load("q5a20_2.mat");

q0a0 = mean(table2array(q0a0),2) .* 6894.76;                q5aN5 =
mean(table2array(q5aN5),2) .* 6894.76 - q0a0;
q5a0 = mean(table2array(q5a0),2) .* 6894.76 - q0a0;         q5a5 =
mean(table2array(q5a5),2) .* 6894.76 - q0a0;
q5a10 = mean(table2array(q5a10),2) .* 6894.76 - q0a0;       q5a15 =
mean(table2array(q5a15),2) .* 6894.76 - q0a0;
q5a20_1 = mean(table2array(q5a20_1),2) .* 6894.76 - q0a0;   q5a20_2 =
mean(table2array(q5a20_2),2) .* 6894.76 - q0a0; % Pa
q0a0 = q0a0 - q0a0;

load("Theo_Vals.mat"); Theo = table2array(Theo_Vals);
N5x = Theo(:,1);    N5y = Theo(:,2);    N5Cp = Theo(:,3);
P0x = Theo(:,4);    P0y = Theo(:,5);    P0Cp = Theo(:,6);
P5x = Theo(:,7);    P5y = Theo(:,8);    P5Cp = Theo(:,9);
P10x = Theo(:,10);  P10y = Theo(:,11);  P10Cp = Theo(:,12);
P15x = Theo(:,13);  P15y = Theo(:,14);  P15Cp = Theo(:,15);
P20x = Theo(:,16);  P20y = Theo(:,17);  P20Cp = Theo(:,18);


Pa = 30.11 * 3386.39;               % Pa
Ta = (79.5 - 32) * (5/9) + 273.15;  % K
T_ref = 291.15;
mu_ref = 1.827e-5;
S = 120;
mu = mu_ref * (Ta/T_ref)^(3/2) * (T_ref + S)/(Ta + S);

alpha = [-5 0 5 10 15 20];

Xc_lower = [0.015, 0.029, 0.055, 0.080, 0.105, 0.157, 0.207, 0.257, 0.306,
...
    0.407, 0.507, 0.608, 0.708, 0.812, 0.912, 1.000];
Xc_upper = [0.000, 0.013, 0.025, 0.048, 0.073, 0.097, 0.150, 0.200, 0.250,
...
    0.300, 0.400, 0.500, 0.600, 0.700, 0.800, 0.900];
Yc_lower = [-0.0089, -0.0156, -0.0160, -0.0190, -0.0216, -0.0262, -0.0299,
...
    -0.0325, -0.0353, -0.0393, -0.0405, -0.0372, -0.0355, -0.0312, -
0.0136, 0.0000];
Yc_upper = [0.0000, 0.0394, 0.0514, 0.0678, 0.0794, 0.0868, 0.0933,
0.0927, ...
```

```
    0.0922, 0.0857, 0.0777, 0.0656, 0.0522, 0.0372, 0.0244, 0.0125];
xac_c = 0.238;
yac_c = 0.07;
c = 1;

Re = 827000;
R = 287;
rho = Pa / (R * Ta);

% Coefficent Calculations

qinf_5N5 = q5aN5(2) - q5aN5(1);  qinf_50 = q5a0(2) - q5a0(1);
qinf_55 = q5a5(2) - q5a5(1);     qinf_510 = q5a10(2) - q5a10(1);
qinf_515 = q5a15(2) - q5a15(1);  qinf_520 = q5a20_1(2) - q5a20_1(1);

taps_5N5 = q5aN5(3:34);  taps_50 = q5a0(3:34);
taps_55 = q5a5(3:34);    taps_510 = q5a10(3:34);
taps_515 = q5a15(3:34);  taps_520 = q5a20_1(3:34);

u_5N5 = sqrt((2 .* (qinf_5N5)) ./ rho);  u_50 = sqrt((2 .* (qinf_50)) ./
rho);
u_55 = sqrt((2 .* (qinf_55)) ./ rho);    u_510 = sqrt((2 .* (qinf_510)) ./
rho);
u_515 = sqrt((2 .* (qinf_515)) ./ rho);  u_520 = sqrt((2 .* (qinf_520)) ./
rho);

Re_5N5 = (rho * u_5N5 * c) / mu;    Re_50 = (rho * u_50 * c) / mu;
Re_55 = (rho * u_55 * c) / mu;       Re_510 = (rho * u_510 * c) / mu;
Re_515 = (rho * u_515 * c) / mu;    Re_520 = (rho * u_520 * c) / mu;
Re = [Re_5N5;Re_50;Re_55;Re_510;Re_515;Re_520];

Cp_5N5 = (taps_5N5 - q5aN5(1)) / qinf_5N5;  Cp_50 = (taps_50 - q5a0(1)) /
qinf_50;
Cp_55 = (taps_55 - q5a5(1)) / qinf_55;      Cp_510 = (taps_510 - q5a10(1))
/ qinf_510;
Cp_515 = (taps_515 - q5a15(1)) / qinf_515;  Cp_520 = (taps_520 -
q5a20_1(1)) / qinf_520;

Cp_lower_5N5 = Cp_5N5(1:16);  Cp_lower_50 = Cp_50(1:16);
Cp_lower_55 = Cp_55(1:16);    Cp_lower_510 = Cp_510(1:16);
Cp_lower_515 = Cp_515(1:16);  Cp_lower_520 = Cp_520(1:16);

Cp_upper_5N5 = Cp_5N5(17:32); Cp_upper_50 = Cp_50(17:32);
Cp_upper_55 = Cp_55(17:32);   Cp_upper_510 = Cp_510(17:32);
Cp_upper_515 = Cp_515(17:32); Cp_upper_520 = Cp_520(17:32);

Cn_5N5 = mean(trapz(Xc_lower, Cp_lower_5N5)) - mean(trapz(Xc_upper,
Cp_upper_5N5));
Cn_50 = mean(trapz(Xc_lower, Cp_lower_50)) - mean(trapz(Xc_upper,
Cp_upper_50));
Cn_55 = mean(trapz(Xc_lower, Cp_lower_55)) - mean(trapz(Xc_upper,
Cp_upper_55));
Cn_510 = mean(trapz(Xc_lower, Cp_lower_510)) - mean(trapz(Xc_upper,
Cp_upper_510));
```

```
Cn_515 = mean(trapz(Xc_lower, Cp_lower_515)) - mean(trapz(Xc_upper,
Cp_upper_515));
Cn_520 = mean(trapz(Xc_lower, Cp_lower_520)) - mean(trapz(Xc_upper,
Cp_upper_520));

dyu_dx = gradient(Yc_upper) ./ gradient(Xc_upper);
dyl_dx = gradient(Yc_lower) ./ gradient(Xc_lower);

Ca_5N5 = mean(trapz(Xc_upper, Cp_upper_5N5 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_5N5 .* dyl_dx));
Ca_50 = mean(trapz(Xc_upper, Cp_upper_50 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_50 .* dyl_dx));
Ca_55 = mean(trapz(Xc_upper, Cp_upper_55 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_55 .* dyl_dx));
Ca_510 = mean(trapz(Xc_upper, Cp_upper_510 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_510 .* dyl_dx));
Ca_515 = mean(trapz(Xc_upper, Cp_upper_515 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_515 .* dyl_dx));
Ca_520 = mean(trapz(Xc_upper, Cp_upper_520 .* dyu_dx)) -
mean(trapz(Xc_lower, Cp_lower_520 .* dyl_dx));

Cl_5N5 = Cn_5N5 * cosd(-5) - Ca_5N5 * sind(-5); Cl_50 = Cn_50 * cosd(0) -
Ca_50 * sind(0);
Cl_55 = Cn_55 * cosd(5) - Ca_55 * sind(5);      Cl_510 = Cn_510 * cosd(10)
- Ca_510 * sind(10);
Cl_515 = Cn_515 * cosd(15) - Ca_515 * sind(15); Cl_520 = Cn_520 * cosd(20)
- Ca_520 * sind(20);
Cl = [Cl_5N5 Cl_50 Cl_55 Cl_510 Cl_515 Cl_520];

Cd_5N5 = Cn_5N5 * sind(-5) + Ca_5N5 * cosd(-5); Cd_50 = Cn_50 * sind(0) +
Ca_50 * cosd(0);
Cd_55 = Cn_55 * sind(5) + Ca_55 * cosd(5);      Cd_510 = Cn_510 * sind(10)
+ Ca_510 * cosd(10);
Cd_515 = Cn_515 * sind(15) + Ca_515 * cosd(15); Cd_520 = Cn_520 * sind(20)
+ Ca_520 * cosd(20);
Cd = [Cd_5N5 Cd_50 Cd_55 Cd_510 Cd_515 Cd_520];

dyu_dx = dyu_dx';
dyl_dx = dyl_dx';
x_rel = (Xc_upper - xac_c)';
y_rel_u = (Yc_upper - yac_c)';
y_rel_l = (Yc_lower - yac_c)';
Xc_upper = Xc_upper(:);

t1_5N5 = (Cp_upper_5N5 - Cp_lower_5N5) .* x_rel;    t2_5N5 = Cp_upper_5N5
.* dyu_dx .* y_rel_u;    t3_5N5 = Cp_lower_5N5 .* dyl_dx .* y_rel_l;
t1_50 = (Cp_upper_50 - Cp_lower_50) .* x_rel;       t2_50 = Cp_upper_50 .*
dyu_dx .* y_rel_u;      t3_50 = Cp_lower_50 .* dyl_dx .* y_rel_l;
t1_55 = (Cp_upper_55 - Cp_lower_55) .* x_rel;       t2_55 = Cp_upper_55 .*
dyu_dx .* y_rel_u;      t3_55 = Cp_lower_55 .* dyl_dx .* y_rel_l;
t1_510 = (Cp_upper_510 - Cp_lower_510) .* x_rel;    t2_510 = Cp_upper_510
.* dyu_dx .* y_rel_u;    t3_510 = Cp_lower_510 .* dyl_dx .* y_rel_l;
t1_515 = (Cp_upper_515 - Cp_lower_515) .* x_rel;    t2_515 = Cp_upper_515
.* dyu_dx .* y_rel_u;    t3_515 = Cp_lower_515 .* dyl_dx .* y_rel_l;
```

```
t1_520 = (Cp_upper_520 - Cp_lower_520) .* x_rel;    t2_520 = Cp_upper_520
.* dyu_dx .* y_rel_u;    t3_520 = Cp_lower_520 .* dyl_dx .* y_rel_l;

Cm_ac_5N5 = trapz(Xc_upper, t1_5N5 - t2_5N5 + t3_5N5);  Cm_ac_50 =
trapz(Xc_upper, t1_50 - t2_50 + t3_50);
Cm_ac_55 = trapz(Xc_upper, t1_55 - t2_55 + t3_55);      Cm_ac_510 =
trapz(Xc_upper, t1_510 - t2_510 + t3_510);
Cm_ac_515 = trapz(Xc_upper, t1_515 - t2_515 + t3_515);  Cm_ac_520 =
trapz(Xc_upper, t1_520 - t2_520 + t3_520);
Cm_ac = mean([Cm_ac_5N5; Cm_ac_50; Cm_ac_55; Cm_ac_510; Cm_ac_515;
Cm_ac_520],2)';

% Wake Work
yw_in = [0, 2, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10,
10.5, 11, 13];
yw = (yw_in - mean(yw_in)) * .0254;

qw_5N5 = q5aN5(35:54);  qw_50 = q5a0(35:54);
qw_55 = q5a5(35:54);    qw_510 = q5a10(35:54);
qw_515 = q5a15(35:54);  qw_520 = q5a20_1(35:54);

uw_5N5 = sqrt((2 * qw_5N5) / rho);  uw_50 = sqrt((2 * qw_50) / rho);
uw_55 = sqrt((2 * qw_55) / rho);    uw_510 = sqrt((2 * qw_510) / rho);
uw_515 = sqrt((2 * qw_515) / rho);  uw_520 = sqrt((2 * qw_520) / rho);

Dp_5N5 = rho * trapz(yw, uw_5N5 .* (u_5N5 - uw_5N5));    Dp_50 = rho *
trapz(yw, uw_50 .* (u_50 - uw_50));
Dp_55 = rho * trapz(yw, uw_55 .* (u_55 - uw_55));        Dp_510 = rho *
trapz(yw, uw_510 .* (u_510 - uw_510));
Dp_515 = rho * trapz(yw, uw_515 .* (u_515 - uw_515));    Dp_520 = rho *
trapz(yw, uw_520 .* (u_520 - uw_520));

Cdw_5N5 = 2 * trapz(yw,sqrt(qw_5N5/qinf_5N5) - qw_5N5/qinf_5N5);    Cdw_50
= 2 * trapz(yw,sqrt(qw_50/qinf_50) - qw_50/qinf_50);
Cdw_55 = 2 * trapz(yw,sqrt(qw_55/qinf_55) - qw_55/qinf_55);
Cdw_510 = 2 * trapz(yw,sqrt(qw_510/qinf_510) - qw_510/qinf_510);
Cdw_515 = 2 * trapz(yw,sqrt(qw_515/qinf_515) - qw_515/qinf_515);
Cdw_520 = 2 * trapz(yw,sqrt(qw_520/qinf_520) - qw_520/qinf_520);
Cdw = [Cdw_5N5 Cdw_50 Cdw_55 Cdw_510 Cdw_515 Cdw_520];


% xx_5N5 = ; xx_50 = ;
% xx_55 = ; xx_510 = ;
% xx_515 = ; xx_520 = ;

% Theoretical Interpretation
% Initialize

results = zeros(length(alpha), 5);

for i = 1:length(alpha)
    % Assign temporary variables (_t suffix)
    if alpha(i) == -5
        x_t = N5x; y_t = N5y; Cp_t = N5Cp;
```

```
    elseif alpha(i) == 0
        x_t = P0x; y_t = P0y; Cp_t = P0Cp;
    elseif alpha(i) == 5
        x_t = P5x; y_t = P5y; Cp_t = P5Cp;
    elseif alpha(i) == 10
        x_t = P10x; y_t = P10y; Cp_t = P10Cp;
    elseif alpha(i) == 15
        x_t = P15x; y_t = P15y; Cp_t = P15Cp;
    elseif alpha(i) == 20
        x_t = P20x; y_t = P20y; Cp_t = P20Cp;
    end

    % Split into upper and lower surfaces
    upper_mask_t = y_t >= 0;
    x_upper_t = x_t(upper_mask_t);
    y_upper_t = y_t(upper_mask_t);
    Cp_upper_t = Cp_t(upper_mask_t);

    lower_mask_t = ~upper_mask_t;
    x_lower_t = x_t(lower_mask_t);
    y_lower_t = y_t(lower_mask_t);
    Cp_lower_t = Cp_t(lower_mask_t);

    % Ensure ordered from LE to TE (flip upper surface)
    x_upper_t = flip(x_upper_t);
    y_upper_t = flip(y_upper_t);
    Cp_upper_t = flip(Cp_upper_t);

    % Combine into closed loop (upper -> lower)
    x_loop_t = [x_upper_t; x_lower_t];
    y_loop_t = [y_upper_t; y_lower_t];
    Cp_loop_t = [Cp_upper_t; Cp_lower_t];

    % Compute Cn (normal force coefficient)
    dy_t = diff(y_loop_t);
    Cn_t = -sum(Cp_loop_t(1:end-1) .* dy_t);

    % Compute Ca (axial force coefficient)
    dx_t = diff(x_loop_t);
    Ca_t = sum(Cp_loop_t(1:end-1) .* dx_t);

    % Compute Cm_ac (moment about AC)
    x_rel_t = x_loop_t(1:end-1) - xac_c;
    y_rel_t = y_loop_t(1:end-1) - yac_c;
    Cm_ac_t = -sum(Cp_loop_t(1:end-1) .* (x_rel_t .* dy_t - y_rel_t .*
dx_t));

    % Compute Cl and Cd (no radians, use cosd/sind)
    alpha_deg_t = alpha(i);
    Cl_t = Cn_t * cosd(alpha_deg_t) - Ca_t * sind(alpha_deg_t);
    Cd_t = Cn_t * sind(alpha_deg_t) + Ca_t * cosd(alpha_deg_t);

    % Store results
    results(i, :) = [alpha_deg_t, Cl_t, Cd_t, Cn_t, Cm_ac_t];
```

```
        % results(:,1) results(:,2) results(:,3) results(:,4) results(:,5)
end

% disp('Results: [Alpha(deg), Cl, Cd, Cn, Cm_ac]');
% disp(results);


% Plots


% AoA vs. Cl
figure(1)
hold on, grid on
plot(alpha, Cl, '-o', "Color", 'r')
plot(results(:,1), results(:,2), 'bo-')
xlabel('Angle of Attack (deg)')
ylabel("C_l")
legend("Experimental", "Theoretical", Location='best')
title("Angle of Attack vs. Lift Coefficient")


% Cl vs. Cd
figure(2)
hold on, grid on
plot(Cl, Cd, '-o', "Color", 'r')
xlabel("C_l")
ylabel("C_d")
title("Lift Coefficient vs. Drag Coefficient")


% Cl vs. Cm_ac
figure(3)
hold on, grid on
plot(Cl, Cm_ac, '-o', "Color", 'r')
xlabel("C_l")
ylabel("Cm_ac")
title("Lift Coefficient vs. Moment Coefficient")


% Cp for AoA = -5
figure(4)
yyaxis left;
hold on, grid on;

hAirfoil = plot(N5x, N5y, 'k-', 'LineWidth', 2);
fill(N5x, N5y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(N5x, N5Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(N5Cp)-0.2, max(N5Cp)+0.2]);
```

```
plot([Xc_lower' Xc_upper], [Cp_lower_5N5 Cp_upper_5N5], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = -5');
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');


% Cp for AoA = 0
figure(5)
yyaxis left;
hold on, grid on;

hAirfoil = plot(P0x, P0y, 'k-', 'LineWidth', 2);
fill(P0x, P0y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(P0x, P0Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(P0Cp)-0.2, max(P0Cp)+0.2]);

plot([Xc_lower' Xc_upper], [Cp_lower_50 Cp_upper_50], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = 0');
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');


% Cp for AoA = 5
figure(6)
yyaxis left;
hold on, grid on;

hAirfoil = plot(P5x, P5y, 'k-', 'LineWidth', 2);
fill(P5x, P5y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(P5x, P5Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(P5Cp)-0.2, max(P5Cp)+0.2]);

plot([Xc_lower' Xc_upper], [Cp_lower_55 Cp_upper_55], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = 5');
```

```
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');


% Cp for AoA = 10
figure(7)
yyaxis left;
hold on, grid on;

hAirfoil = plot(P10x, P10y, 'k-', 'LineWidth', 2);
fill(P10x, P10y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(P10x, P10Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(P10Cp)-0.2, max(P10Cp)+0.2]);

plot([Xc_lower' Xc_upper], [Cp_lower_510 Cp_upper_510], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = 10');
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');


% Cp for AoA = 15
figure(8)
yyaxis left;
hold on, grid on;

hAirfoil = plot(P15x, P15y, 'k-', 'LineWidth', 2);
fill(P15x, P15y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(P15x, P15Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(P15Cp)-0.2, max(P15Cp)+0.2]);

plot([Xc_lower' Xc_upper], [Cp_lower_515 Cp_upper_515], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = 15');
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');


% Cp for AoA = 20
figure(9)
yyaxis left;
```

```
hold on, grid on;

hAirfoil = plot(P20x, P20y, 'k-', 'LineWidth', 2);
fill(P20x, P20y, 'k', 'FaceAlpha', 0.1);
ylabel('Airfoil Thickness (y/c)');
ylim([-1 1]);

yyaxis right;
hCp = plot(P20x, P20Cp, 'r-', 'LineWidth', 1.5);
set(gca, 'YDir', 'reverse');
ylabel('Pressure Coefficient (C_p)');
ylim([min(P20Cp)-0.2, max(P20Cp)+0.2]);

plot([Xc_lower' Xc_upper], [Cp_lower_520 Cp_upper_520], "Color", 'blue',
'LineWidth', 1.5)

xlabel('Chord Position (x/c)');
title('Airfoil Geometry and Pressure Distribution for AoA = 20');
legend('Airfoil', '', "Theoretical", "Experimental", 'Location', 'best');



%Cdw vs. Cdp
figure(10)
hold on, grid on
plot(alpha, Cd, 'Color', 'r')
plot(alpha, Cdw, 'Color', 'b')
xlabel('Angle of Attack (deg)')
ylabel('Drag Coefficient (C_d)')
title('Angle of Attack vs. Drag Coefficient')
legend('Surface Drag', 'Wake Drag', Location='northwest')


% Wake Velocity Profiles
figure(11);
clf;
hold on;
grid on;

colors = parula(6);
line_styles = {'-', '--', ':', '-.', '-', '--'};

legend_labels = {};
q_all = {q5aN5, q5a0, q5a5, q5a10, q5a15, q5a20_1};
alpha_labels = {'-5°', '0°', '5°', '10°', '15°', '20°'};

for i = 1:6
    q_current = q_all{i};
    q_inf = q_current(2) - q_current(1);
    u_wake = sqrt(q_current(33:52)/q_inf);

    % Plot with style variations
    p = plot(yw, u_wake, ...
        'Color', colors(i,:), ...
```

```
        'LineStyle', line_styles{i}, ...
        'MarkerFaceColor', colors(i,:));

    legend_labels{end+1} = ['\alpha = ' alpha_labels{i}];
end

xlabel('Lateral Position, y (m)', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('Normalized Velocity, u_{wake}/u_{∞}', 'FontSize', 12,
'FontWeight', 'bold');
title('Wake Velocity Profiles vs. Angle of Attack', 'FontSize', 14,
'FontWeight', 'bold');

ylim([0.4 1.1]);
xlim([min(yw) max(yw)]);
legend(legend_labels)
```