Cooper Union for the Advancement of Science and Art

Final Project - Analysis of Albatross Flight

ME-422: Fundamentals of Aerodynamics

Authors:
Preston Xu
Neil Sawhney
Gary Kim

Professor:
Professor Wootton



May 10, 2024

Contents

1 Albatross Background

The albatross is a large seabird known for its large wingspan and ability to efficiently glide and stay airborne for hours at a time, often without flapping. For this project, we analyzed the cruise flight characteristics of the albatross.

Airfoil wing area S, span b, aspect ratio AR, chord vs. position (c(y)), twist distribution, lift to drag slope m and zero lift angle of attack are obtained or estimated from the following papers: *CRC Handbook of Avian Body Masses*, *Antarctic Wildlife: The Birds and Mammals*, "The Biology of the Great Albatrosses, Diomedea Exulahs and Diomedea Epomophora," *Birds with the Biggest Wingspans*, and .

Due to the large variation in albatross wing dimensions, average values were taken across multiple species such as the Black-browed albatross, the wandering albatross, Antipodean Albatross, Amsterdam Albatross, Tristan Albatross, and Northern Royal Albatross.

- Cruise Speed: $V_\infty = 20\ m/s$
- Span Length: $b_{span} = 3.1\ m$
- Chord Length: $c_l = 0.3\ m$
- Wing Area: $S = 0.336 \cdot 2 = 0.672\ m^2$
- Maximum Take Off Weight: $MTOW = 11\ kg$
- Section Lift Slope: $m_0 = 0.729\ 1/rad$
- Aspect Ratio: $AR = b_{span}^2/S = 14.3$

2 Analysis

Throughout this analysis, calculations are made using Symbolic Python in a Jupyter Notebook. This makes it easy to perform calculations and display analytical results for any aircraft configuration. While our analysis will focus on a general overview of albatross birds, the jupyter notebook available for download here can be used to analyze any aircraft configuration. A pdf output of the notebook parameters and calculation results used during our analysis is also shown in Appendix **??** for convenience.

2.1 Airfoil Model

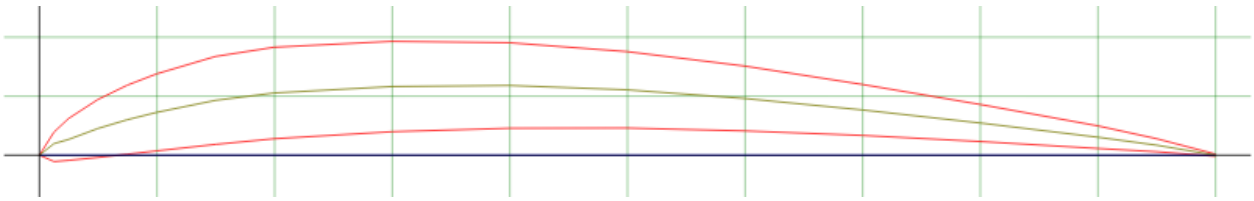


Figure 2.1: GOE 173 (ALBATROS 6020) Airfoil

The **GOE 173 (ALBATROS 6020)** airfoil was picked to represent the Albatross wing when fully extended during a glide. This airfoil was designed by the albatros company in a series of World War I aircrafts inspired by albatross birds. Unknown to them at the time, this style of thin cambered airfoil works well at low reynolds numbers for bird flight, but not well for fast flying aircraft[6]. For thin airfoils, turbulence can lead to sudden changes in lift and drag. When turbulent air flows over the airfoil, it can cause local pressure fluctuations. These pressure fluctuations can result in unsteady lift forces, making it challenging for the pilot to maintain stable flight. Thin airfoils are more sensitive to these fluctuations than thicker ones. Thin airfoils also tend to have a more abrupt stall behavior. The stall originates from the leading edge, with sudden separation of upper-side flow. Thicker airfoils stall in more benign ways, with separation starting from the trailing edge

and gradually moving forward[7]. Birds get around these issues by flying at comparably slower speeds and by altering the shape of their wings midflight with occasional flapping. The Albatros 6020 coordinates are available in many online databases. The GOE 173 airfoil and similar airfoils such as the GOE 174 is often used to represent the Albatross wing in several other resources and papers [8] [9].

2.2 Zero Lift Angle of Attack

Using the thin airfoil model, the coefficient of lift was calculated using **??**.

$$C_{lift} = 2\pi(\alpha + 0.1023) \tag{1}$$
$$C_{lift} = 6.29\alpha + 0.64 \tag{2}$$

With this in mind, the zero lift angle of attack can be easily determined by setting $C_{lift} = 0$, providing a zero lift angle of attack value calculated in **??**.

$$\alpha_{0L} = -0.1023 \text{ rad} \tag{3}$$
$$\alpha_{0L} = -5.86^\circ \tag{4}$$

The albatross is known for its ability to glide for long periods of time without flapping its wings. The zero lift angle of attack of -5.86 degrees is a good indicator of this, as it means that the albatross can generate lift with a low angle of attack, reducing drag and allowing it to glide for long periods of time.

2.3 Lift vs. Angle of Attack Slope

An equation for the coefficient of lift based on the angle of attack was previously determined using the thin airfoil model to determine the zero lift angle of attack. By putting this value into the full lift equation as shown in **??**, along with the cruise characteristics of the albatross, the lift vs angle of attack slope can be derived to be 1092 N/rad or about 19.06 N/deg.

$$L = \frac{1}{2}\rho v^2 S C_{lift} \tag{5}$$

$$L = \frac{1}{2}(1.293)(20)^2(0.672)(2\pi(\alpha + 0.1023)) \tag{6}$$

2.4 Estimated Lift Force

Based on finite wing theory, the coefficient of lift was calculated to be

$$C_L = (0.0189 * 14.3)\pi = 0.27\pi \tag{7}$$

at an angle of attack of -0.81 deg. This leads to a lift of around $L = 368\ N$. Based on 2d wing theory, the same coefficient of lift was calculated to be

$$C_L = 0.293\pi \tag{8}$$

at the same angle of attack. This leads to a lift of around $404N$.

Finite wing lift is lower compared to comparative 2d models due to the infinite wing assumption made in 2d modeling. It makes the assumption that lift is uniform along the entire length of the wing. This assumption neglects wingtip effects that act to reduce overall lift on the wing including wingtip vorticies that cause induced drag, downwash, and a reduction in the effective angle of attack of the wing, all acting to reduce the overall lift provided by the wing. Many wings include features to counteract or reduce the effects of wingtip vorticies such as winglets and raked wingtips. The albatross could be considered to have a raked wingtip, meaning the 2d model would be closer to the finite wing model, though quantifying by exactly how much would likely require finite element fluid dynamics.

2.5 Estimated Wing Drag

The induced drag of the albatross wing at 0 degree angle of attack was estimated to be 0.37 N at the cruise flight characteristics of 16 m/s using the finite wing model.

$$C_{D,i} = \frac{C_L^2(\delta + 1)}{\pi AR} = 0.6765\pi \tag{9}$$

$$D_i = C_{D,i} S q_\infty = 0.3694N \tag{10}$$

2.6 Lift vs Angle of Attack Curve

A coefficient of lift vs angle of attack curve was determined using xfoil, with a graph of the values shown in **??**. These calculations were performed with a Reynold's number of 325000.

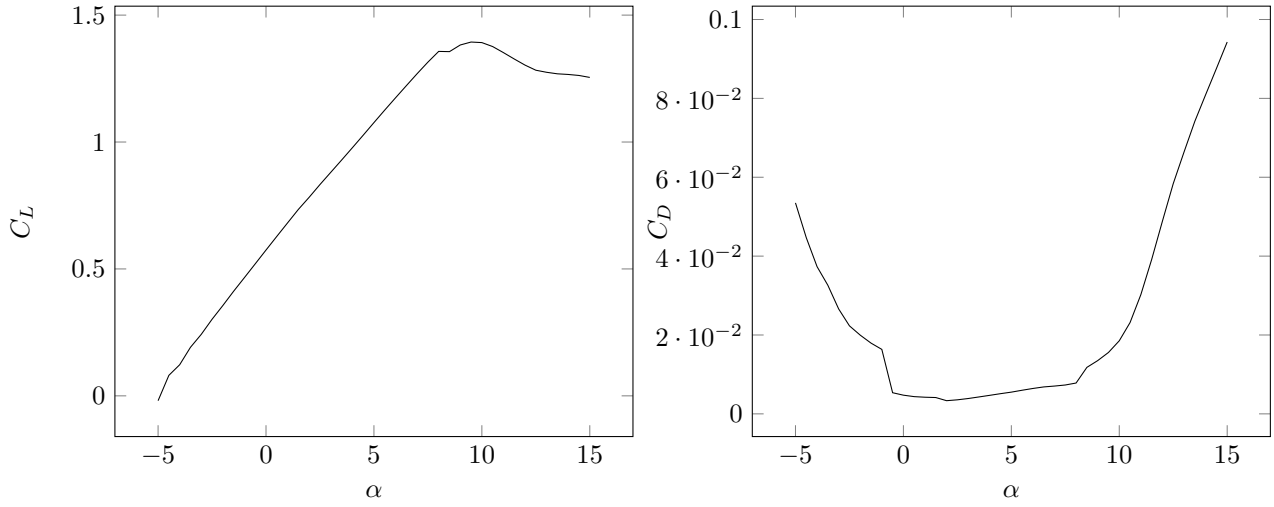


Figure 2.2: XFoil Lift and Drag Analysis

2.7 Stall Characteristics

As calculated using xfoil, stall occurs at around $\alpha_{crit} \approx 7\,\text{deg}$ with a maximum coefficient of lift of $C_{l,max} \approx 1$, varying a little depending on the exact Reynold's number and speed.

Since the wing that we are analyzing is that of a bird, there is much more potential dynamic control over wing shape in order to delay stall. The typical method of avoiding stalls by simply reducing angle of attack is much easier to do since as a living creature, the albatross likely has a much more acute sense of when airfoil over their wings are starting to detach and can adjust for this a lot more accurately than we can in human aircraft. In addition, muscles in the wing provide the ability to change the wing shape to effectively extend natural "slats" and "flaps" in flight, further increasing the maximum coefficient of lift. In addition, the actual airfoil shape changes with varying air pressure due to the compliance of the wing and feathers[10]. This is before we consider that the albatross can simply flap its wings if it is getting to close to a stall as well.

3 CFD Analysis

To achieve a more accurate analysis of albatross flight, a black-browed albatross was modelled using computational fluid dynamics. The hand calculation above are used to verify the CFD analysis and ensure the solver settings and boundary conditions are input correctly. Both the mathematical analysis and the CFD analysis work together to create a more complete understanding of albatross flight.

3.1 Geometry

The geometry for the CFD analysis was created in solidworks using the airfoil coordinates for the GOE 173 airfoil and is shown in **??**. The wing design was created to be a simple swept wing with a length of 3.1 meters, a chord length of 0.336 meters, and a planform area of 0.672 m^2 in accordance with the previous analysis. A uniform angle of attack of 7° was used across the entire wingspan. The wing design was also given a 10 degree bend angle 0.75 meters from the root of the wing, and a reverse bend angle of 10 degrees 1.25 meters from the root of the wing which was decided upon through the use of the reference photo shown in **??**.

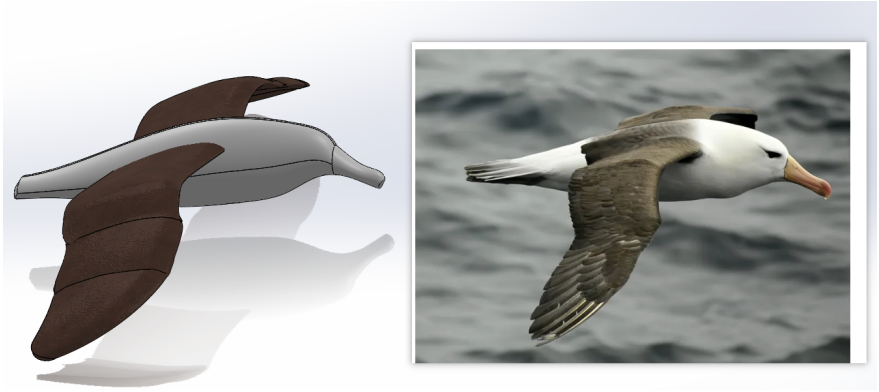
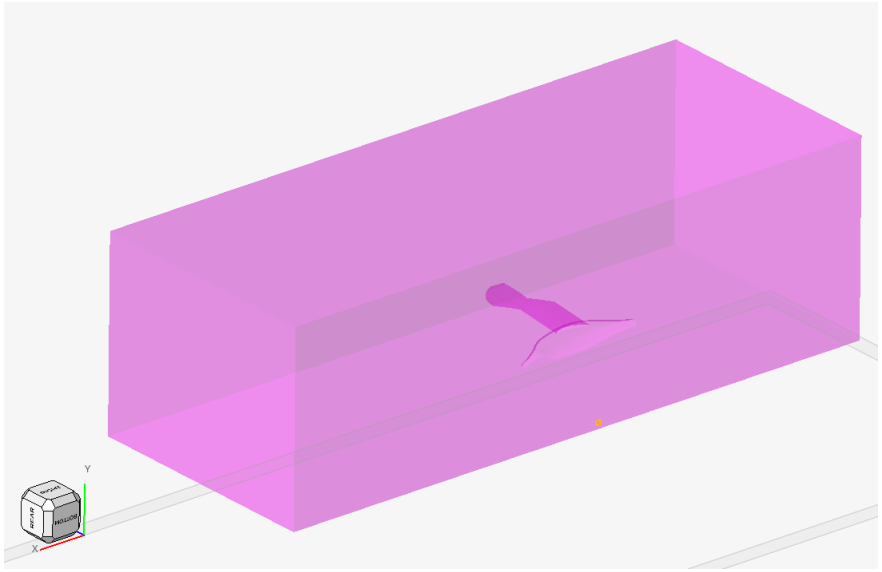


Figure 3.1: Geometry of a Black-browed Albatross on the left and the reference photo on the right

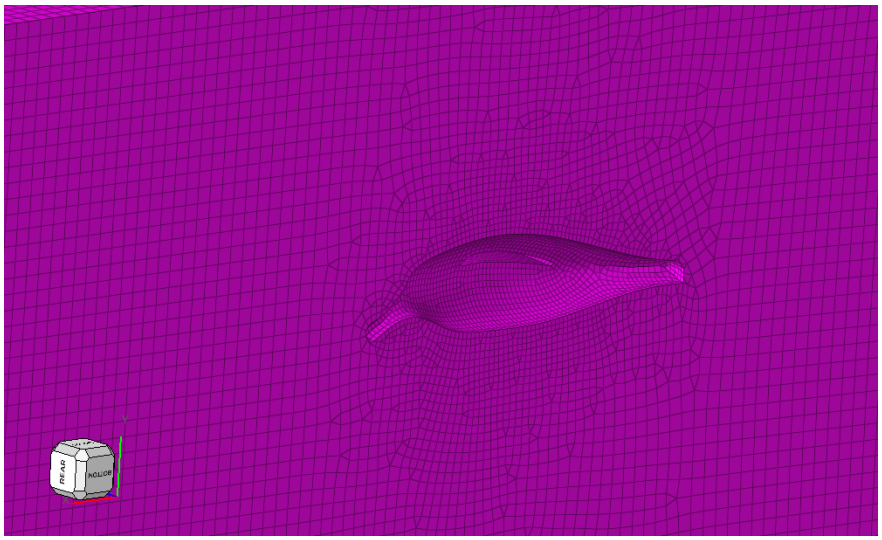
3.2 Mesh

The geometry was imported into HyperMesh to create a CFD tetracore mesh which is shown in **??**. Symmetry was employed on the x-y plane so that only half the bird can be modelled, decreasing computation time dramatically. The fluid volume around the bird has dimensions of 2m x 1m x 1 m with symmetry applied. This volume is intentionally much larger than the size of the bird to accurately capture wing effects, achieve a fully developed flow at the inlet, and ensure negligible backflow at the outlet. A CFD tetracore mesh was chosen over a fully tetrahedral mesh because it enforces cells of near perfect orthogonality at the wall boundaries. It also generates a smooth boundary layer with a uniform growth rate and a precise initial cell height. The mesh includes elements of roughly 30 mm in size with refinement down to 5 mm near the surface of the bird. The final mesh had a total of 1.4 million elements.

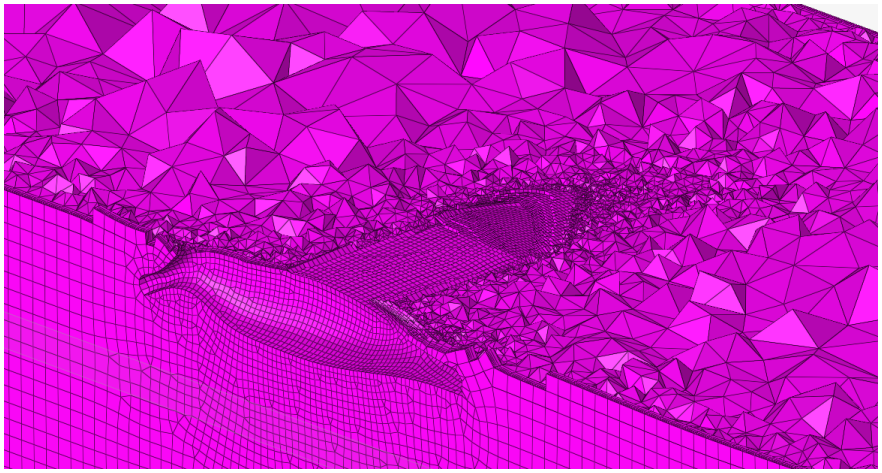
The boundary layer around the walls shown in **??** was created using a y+ value of 20. This is a relatively low y+ value which is ideal for capturing the boundary layer accurately. The initial layer height corresponding to this y+ value was around 0.4 mm.



(a) Translucent view of the CFD mesh



(b) Close-up view of the Wall



(c) Section Cut Showing the inner Tetra-core

Figure 3.2: CFD Mesh of the Albatross

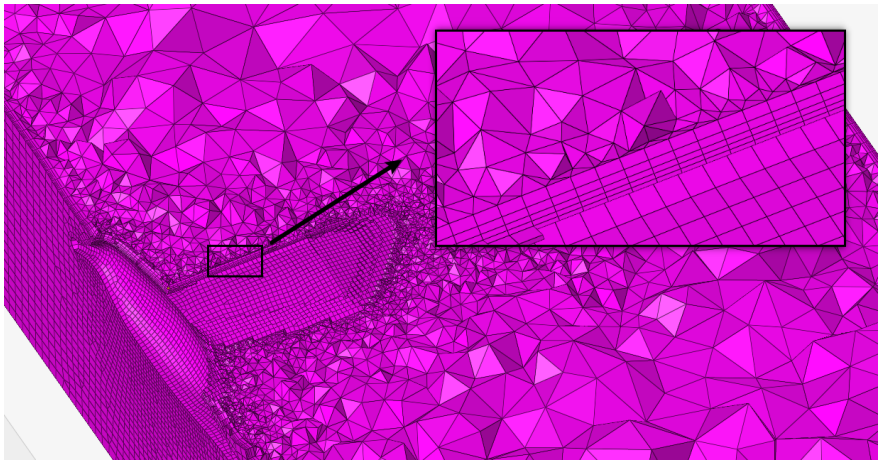


Figure 3.3: Boundary Layer on the CFD Mesh: The zoomed in view shows the boundary layer cells near the upper surface of the wing.

3.3 Boundary Conditions

The velocity at the inlet is set to the cruising speed of the albatross, 20 m/s. The fluid is set to air with a density of 1.293 kg/m3 and a dynamic viscosity of 1.81e-5 kg/ms. The pressure outlet is set to 0 Pa. The inlet and outlet specification in Ansys Fluent is shown in **??**.

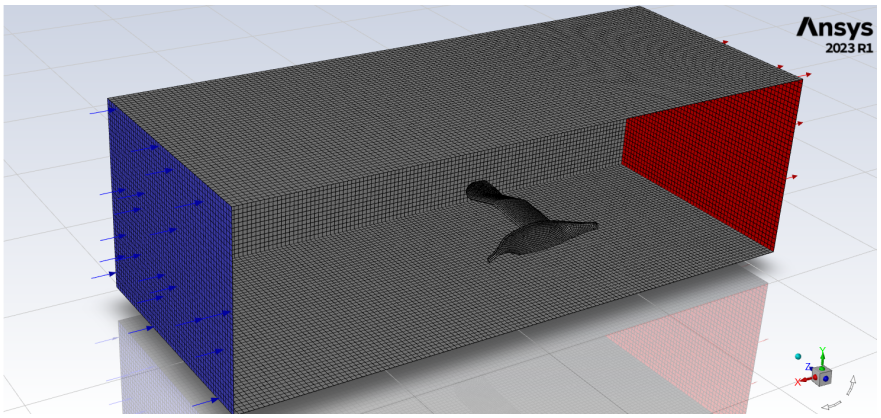


Figure 3.4: Setup Fluent

The hydraulic diameter is calculated using **??** for the inlet and outlet.

$$\begin{aligned} D &= \frac{4A}{P} \\ D &= \frac{4 * (0.95m)^2}{2 * 0.781m + 2 * 1.21m} \\ D &= 0.907m \end{aligned} \tag{11}$$

The Reynolds number is calculated using **??** for cruising speed of 20 m/s.

$$\begin{aligned} Re &= \frac{\rho V D}{\mu} \\ Re &= \frac{1.293kg/m^3 * 20m/s * 0.907m}{1.81e-5kg/ms} \\ Re &= 1.3e6 \end{aligned} \tag{12}$$

For the turbulent specification at the inlet and outlet, the turbulent intensity is calculated using **??**.

$$I = 0.16 * Re^{-1/8} \tag{13}$$

$$= 0.16 * (1.3e6)^{-1/8} \tag{14}$$

$$I = 0.110 \tag{15}$$

$$\tag{16}$$

3.4 Solver and Convergence

The mesh was imported into Ansys Fluent to complete the CFD analysis. The solver used was the k-omega Turbulence model with a coupled implicit steady state solver. Using the default convergence criteria, the model converged in just over 250 iterations. As shown in **??**, the outlet mass flow converged nicely and matches the inlet mass flow identically. This is a good sign that the fluid body was large enough behind the bird to achieve clean outflow. The pressure also converged quickly and stable as shown in **??**

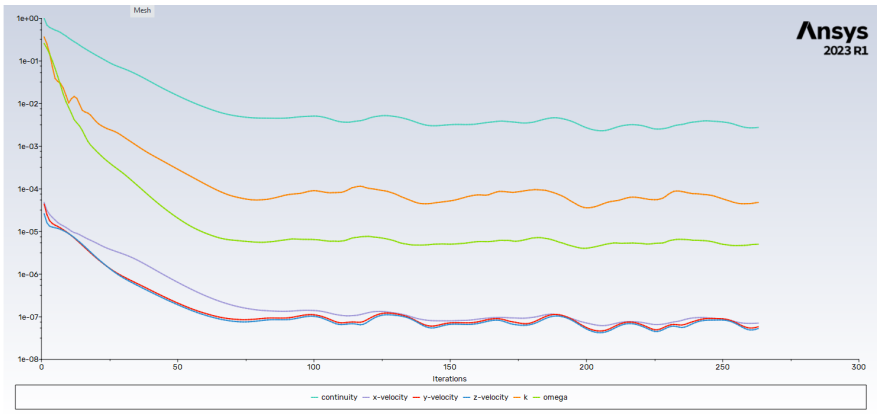


Figure 3.5: Convergence Residuals

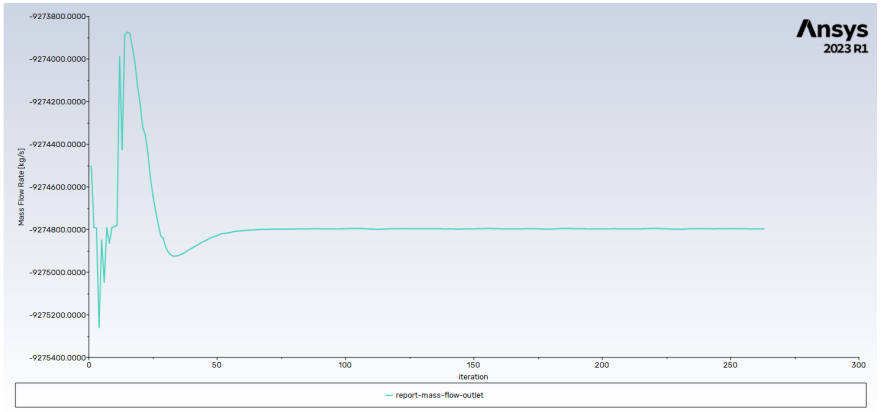


Figure 3.6: Convergence Outlet Mass Flow

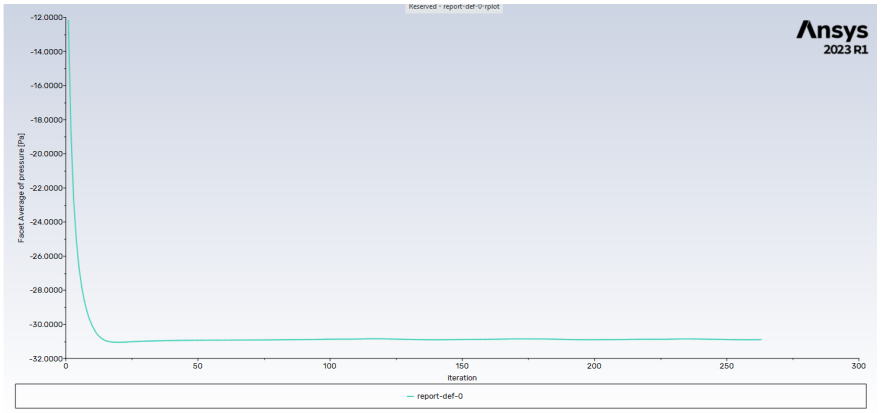


Figure 3.7: Convergence Pressure

3.5 Results

CFD results for lift and drag coefficients are reported and compared with results calculated using finite wing and thin airfoil models in Table **??**. Lift and Drag coefficients were calculated at an AOA of 3 degrees, an empirically observed cruising AOA in Albatrosses that we later verify.

The finite wing calculated C_D in Table **??** represents the total coefficient of drag (sum of viscous and skin drag). It is calculated using Sympy and can be found in Appendix **??**. The lift force results in Table **??** range from 133.36N (CFD Result) to 151.47N (Thin airfoil result). These are all close to and slightly larger than the actual weight of the albatross at 107.91N. This verifies that the chosen AOA is an appropriate estimate of the cruising AOA.

The pressure contour on the bird walls shown in **??** have a low pressure on the frontal upper surface of around -262 Pa and around 117 Pa on the frontal lower surface. This results in absolute pressure values of 101.063 kPa and 101.442 kPa respectively. The nose of the bird also experiences high pressure as expected.

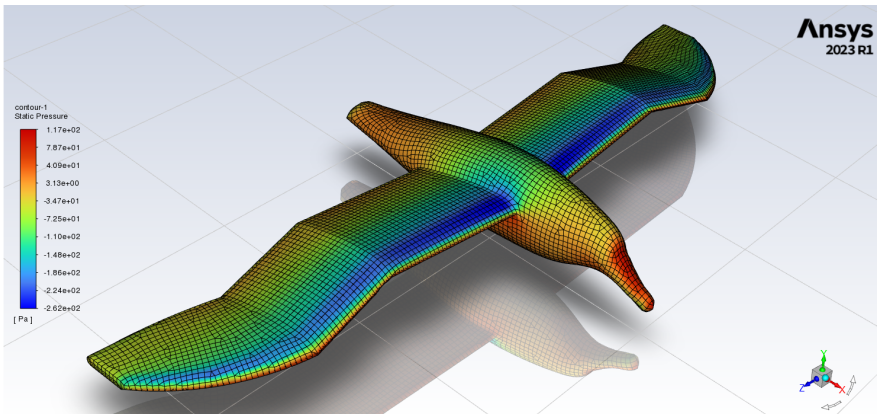


Figure 3.8: Results Pressure

The pathlines of the fluid particles which make contact with the bird are shown in **??**. They reveal higher velocity at the top surface of the bird and lower velocity below as expected. A surprising result is the major difference in velocity between the top and bottom surfaces of the fuselage indicating that the body of the bird itself contributes significantly to the overall lift. There is no observable turbulence behind the birds body and wings verifying that both the wing airfoil shape and fuselage airfoil shape at these flow speeds are of optimal shape to satisfy the kutta-condition.

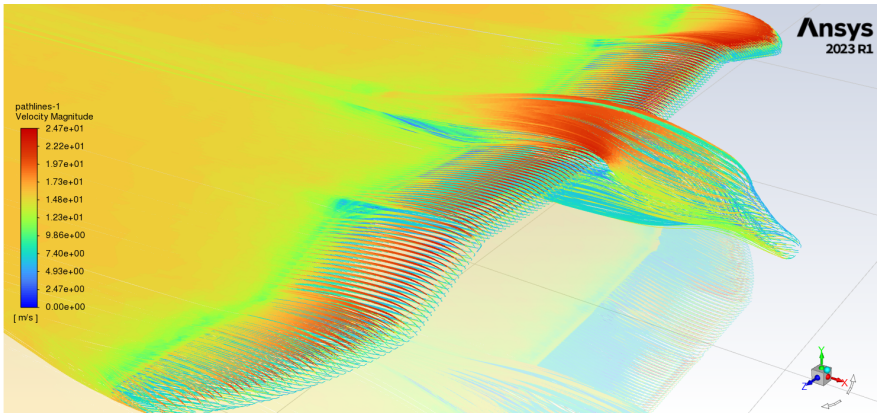


Figure 3.9: Results Pathlines Velocity

The pathlines of the fluid particles throughout the fluid reveal the wing tip vortices caused by the higher pressure air from beneath the wing curling around to the lower pressure area on top as shown in **??**. The

front view also reveals the downwash which is greatest near the wing tips and decreases toward the wing root.

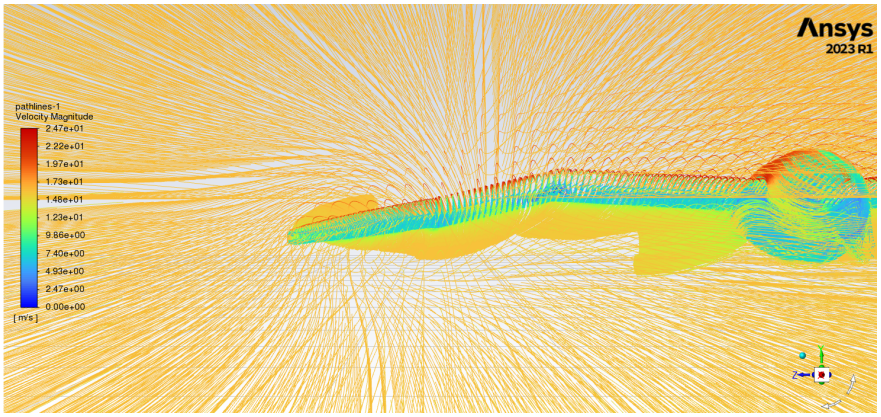


Figure 3.10: Results Pathlines Velocity (Wingtips)

4 Summary

	Thin airfoil Calculated Result	Finite Wing Calculated Result	CFD Result	XFoil Result
C_L	0.92	0.85	1.2	0.881
C_D	N/A	0.027	.032	0.0039
$Lift(N)$	151.47	139.94	133.36	145.05
$Drag(N)$	N/A	4.45	5.27	.642

Table 1: Tabulated Lift and Drag Coefficient Results

The CFD analysis showed that the albatross can generate enough lift to support its weight even without flapping its wings in a 20 mph headwind. The pressure contour on the bird walls showed a low pressure on the frontal upper surface and a high pressure on the frontal lower surface. The pathlines of the fluid particles showed higher velocity at the top surface of the bird and lower velocity below, with no observable turbulence behind the bird's body and wings. The pathlines of the fluid particles throughout the fluid revealed the wing tip vortices caused by the higher pressure air from beneath the wing curling around to the lower pressure area on top. The front view also revealed the downwash which is greatest near the wing tips and decreases toward the wing root.

Overall, the albatross is a remarkable species of bird that is able to generate enough lift to support its weight even without flapping its wings. The high aspect ratio of the albatross wing allows it to generate a large amount of lift with a small change in angle of attack, which is important for efficient gliding.

References

[1] J. B. Dunning Jr., *CRC Handbook of Avian Body Masses*, J. B. Dunning Jr., Ed. CRC Press, 2008, ISBN: 978-1-4200-6444-5.

[2] H. Shirihai, *Antarctic Wildlife: The Birds and Mammals*. Finland: Alula Press, 2002, p. 90, ISBN: 978-951-98947-0-6.

[3] W. L. N. Tickell, "The biology of the great albatrosses, diomedea exulahs and diomedea epomophora," in *Antarctic Bird Studies* (Antarctic Research Series), O. L. Austin, Ed., Antarctic Research Series. Washington, D. C.: American Geophysical Union, 2013, pp. 1–56, ISBN: 978-1-118-66914-3. DOI: 10.1029/ar012p0001.

[4] A. Name, *Birds with the biggest wingspans*, https://birdwatchershub.com/birds-with-the-biggest-wingspans/, Accessed: 2023-05-08, Year of Publication.

[5] https://web.archive.org/web/20090221101143/http://www.taxonomy.nl/Main/Classification/101869.htm, Accessed: 2023-04-06.

[6] *Albatros d.iii johannisthal oaw and oeffag variants*, Available online: [1][1]. [Online]. Available: https://porschecarshistory.com/wp-content/old/biblio4/79/Albatros_DIII_Johannisthal_OAW_and_Oeffag_variants.pdf.

[7] S. Kawai and K. Fujii, "Analysis and prediction of thin-airfoil stall phenomena with hybrid turbulence methodology," *AIAA Journal*, vol. 43, no. 5, pp. 953–961, 2005. DOI: 10.2514/1.11672. eprint: https://doi.org/10.2514/1.11672. [Online]. Available: https://doi.org/10.2514/1.11672.

[8] P. Dutta, O. P. Nagar, S. K. Sahu, R. R. Savale, and R. Gokul Raj, "Aerodynamic analysis of bionic winglet- slotted wings," *Materials Today: Proceedings*, vol. 62, pp. 6701–6707, 2022, International Conference on Materials, Mechanics, Mechatronics and Manufacturing, ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2022.04.752. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785322029182.

[9] M. Hassanalian, G. Throneberry, M. Ali, S. Ben Ayed, and A. Abdelkefi, "Role of wing color and seasonal changes in ambient temperature and solar irradiation on predicted flight efficiency of the albatross," *Journal of Thermal Biology*, vol. 71, pp. 112–122, 2018, ISSN: 0306-4565. DOI: https://doi.org/10.1016/j.jtherbio.2017.11.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306456517304175.

[10] A. Winter, J. Schoombie, and D. L. Smith, "Unlock the secrets of flight: Discovering the wonders of the grey-headed albatross," *University of Pretoria Aeronautical Research Group*, 2024, Available online: https://www.up.ac.za/aeronautical-group/news/post$_3$166786 – $unlock - the - secrets - of - flight - discovering - the - wonders - of - the - grey - headed - albatross$.

[11] P. L. Richardson and E. D. Wakefield, "Observations and models of across-wind flight speed of the wandering albatross," *Royal Society Open Science*, vol. 9, no. 11, p. 211 364, 2022. DOI: 10.1098/rsos.211364. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsos.211364. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsos.211364.

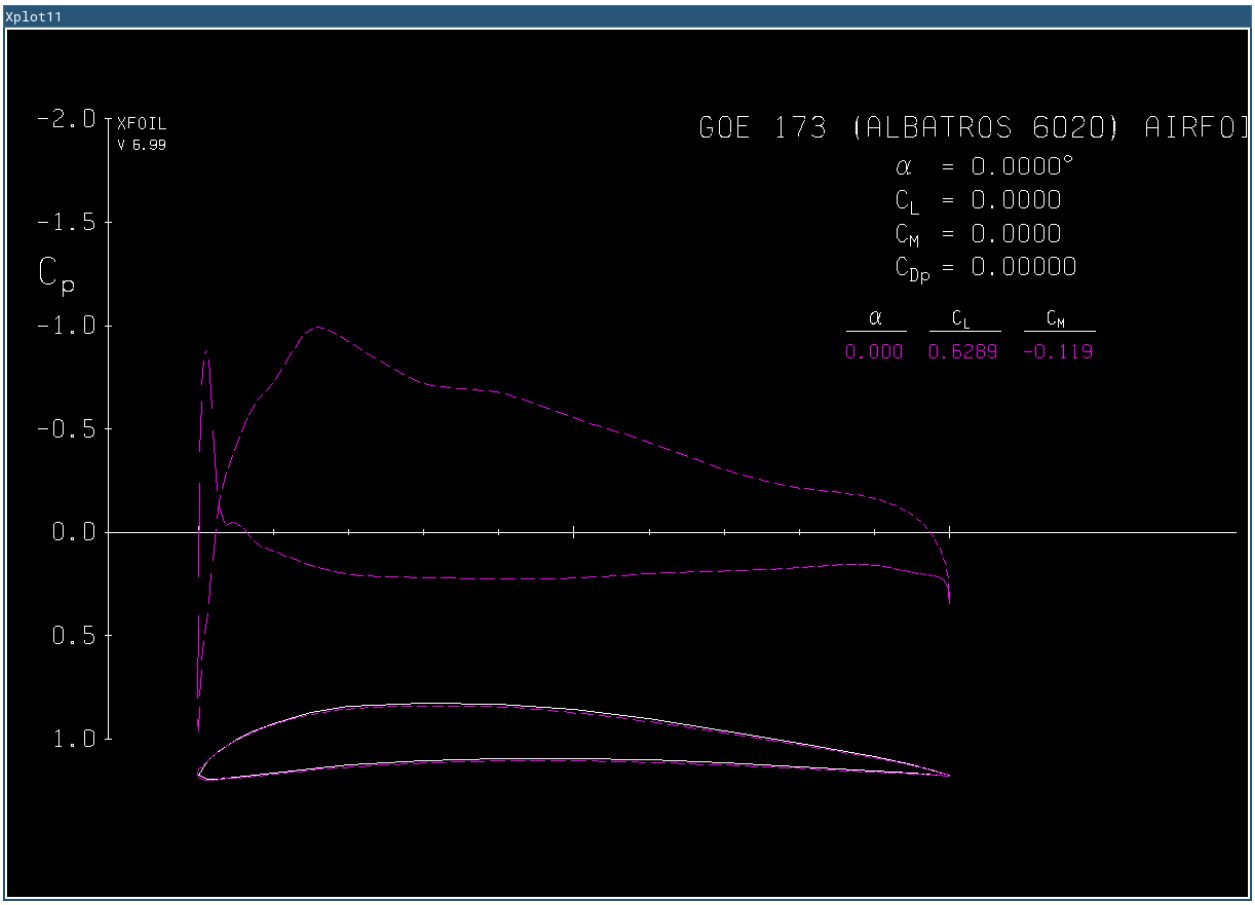
[12] W. An, F. Shi, S. He, W. Wang, H. Zhang, and L. Liu, "Aerodynamic design and optimization of bionic wing based on wandering albatross," in *Proceedings of the International Conference on Aerospace System Science and Engineering 2020*, Z. Jing and X. Zhan, Eds., Singapore: Springer Singapore, 2021, pp. 517–535, ISBN: 978-981-33-6060-0.

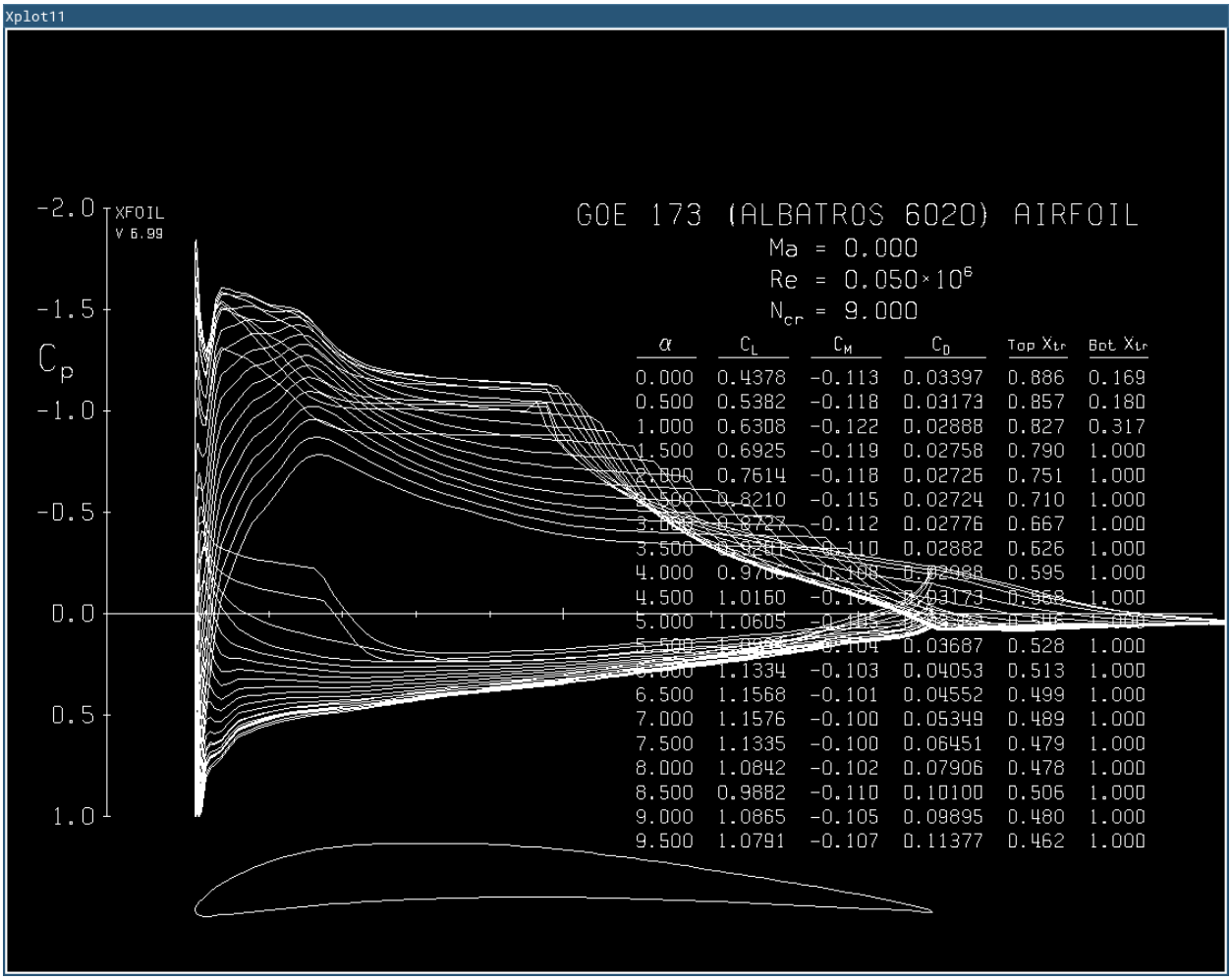
A Appendix

A.1 Used Constants

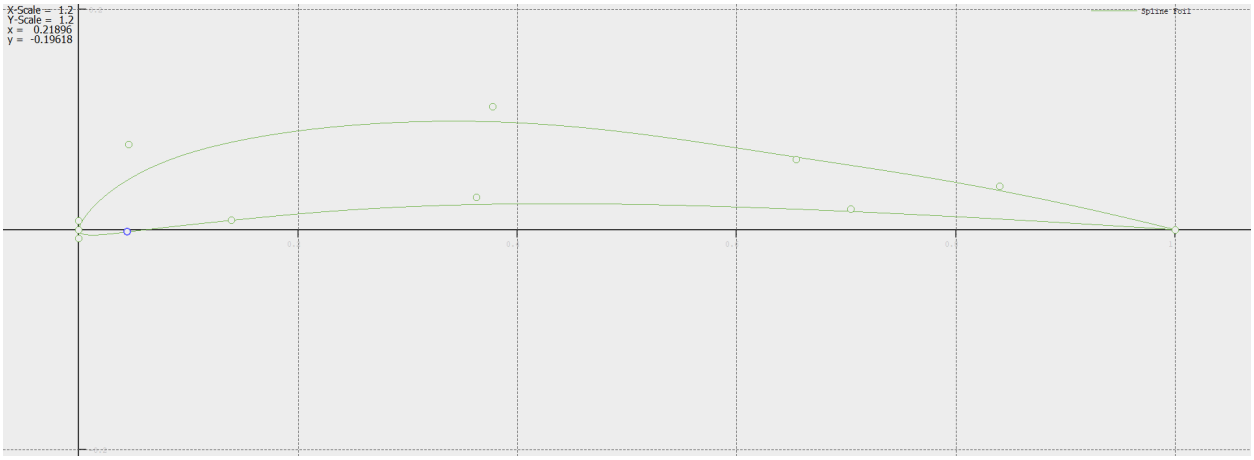
- $\rho_{air} = 1.293\ kg/m^3$
- $g = 9.8\ m/s^2$

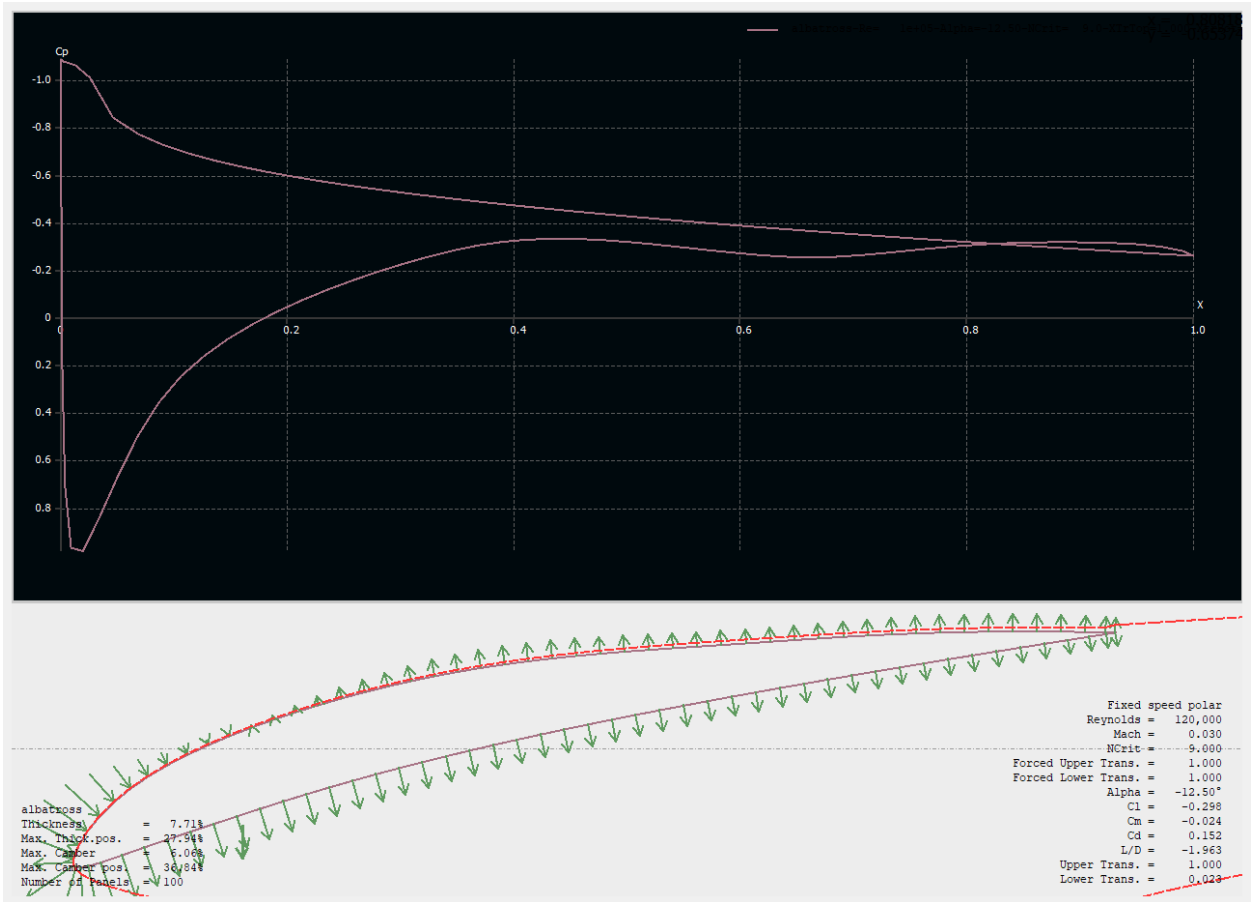
A.2 XFoil Screenshots

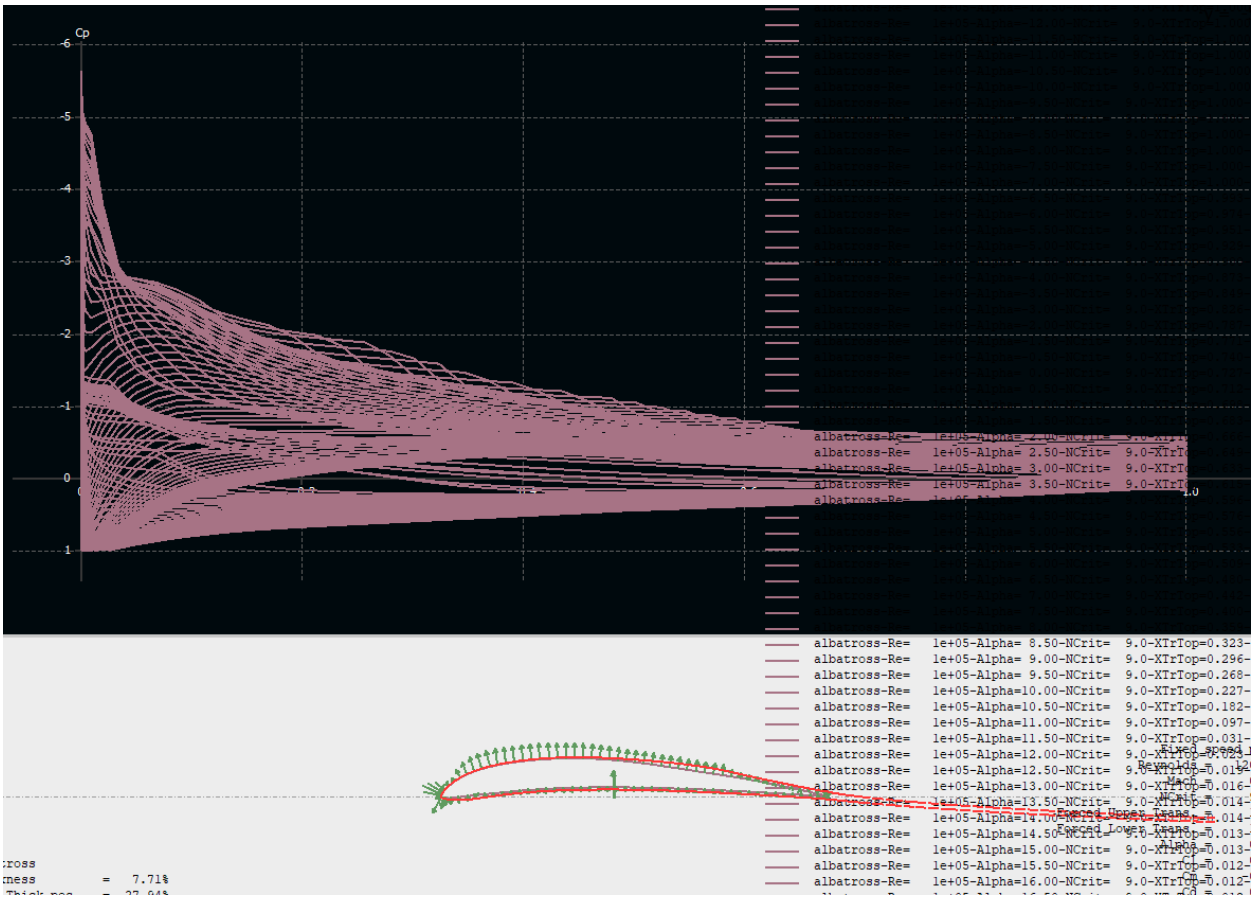


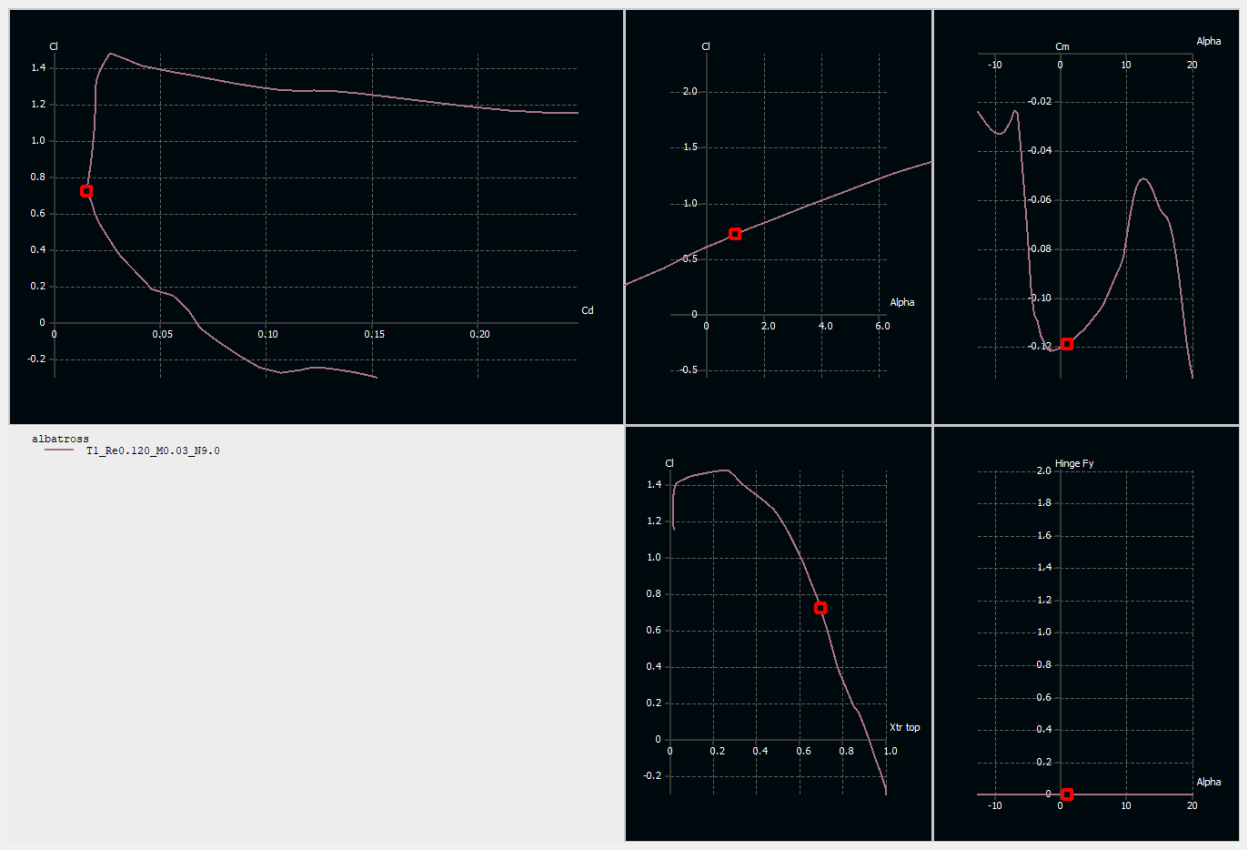


A.3 Xflr5 Screenshots


X-Scale = 1.2
Y-Scale = 1.2
x = 0.21896
y = -0.19618
Spline foil


Cp
-1.0
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
0
0.2
0.4
0.6
0.8
1.0
X
albatross
THickness = 7.71%
Max. Thick.pos. = 27.94%
Max. Camber = 6.06%
Max. Camber pos. = 30.84%
Number of Panels = 100
Fixed speed polar
Reynolds = 120,000
Mach = 0.030
NCrit = 9.000
Forced Upper Trans. = 1.000
Forced Lower Trans. = 1.000
Alpha = -12.50°
Cl = -0.298
Cm = -0.024
Cd = 0.152
L/D = -1.963
Upper Trans. = 1.000
Lower Trans. = 0.023

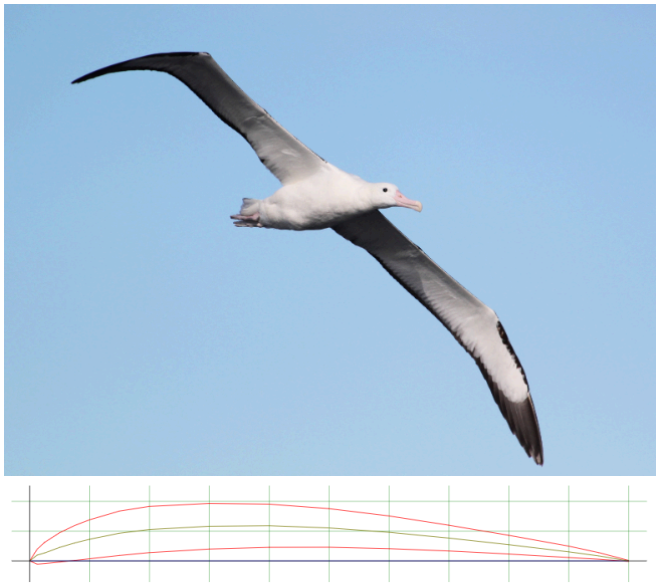

Cp
-6
-5
-4
-3
-2
-1
0
1
1.0
albatross-Re= 1e+05-Alpha= 2.50-NCrit= 9.0-XTrTop=0.640
albatross-Re= 1e+05-Alpha= 3.00-NCrit= 9.0-XTrTop=0.630
albatross-Re= 1e+05-Alpha= 3.50-NCrit= 9.0-XTrTop=0.620
albatross-Re= 1e+05-Alpha= 8.50-NCrit= 9.0-XTrTop=0.323
albatross-Re= 1e+05-Alpha= 9.00-NCrit= 9.0-XTrTop=0.296
albatross-Re= 1e+05-Alpha= 9.50-NCrit= 9.0-XTrTop=0.268
albatross-Re= 1e+05-Alpha=10.00-NCrit= 9.0-XTrTop=0.227
albatross-Re= 1e+05-Alpha=10.50-NCrit= 9.0-XTrTop=0.182
albatross-Re= 1e+05-Alpha=11.00-NCrit= 9.0-XTrTop=0.097
albatross-Re= 1e+05-Alpha=11.50-NCrit= 9.0-XTrTop=0.031
albatross-Re= 1e+05-Alpha=12.00-NCrit= 9.0-XTrTop=0.023
albatross-Re= 1e+05-Alpha=12.50-NCrit= 9.0-XTrTop=0.019
albatross-Re= 1e+05-Alpha=13.00-NCrit= 9.0-XTrTop=0.016
albatross-Re= 1e+05-Alpha=13.50-NCrit= 9.0-XTrTop=0.014
albatross-Re= 1e+05-Alpha=14.00-NCrit= 9.0-XTrTop=0.014
albatross-Re= 1e+05-Alpha=14.50-NCrit= 9.0-XTrTop=0.013
albatross-Re= 1e+05-Alpha=15.00-NCrit= 9.0-XTrTop=0.013
albatross-Re= 1e+05-Alpha=15.50-NCrit= 9.0-XTrTop=0.012
albatross-Re= 1e+05-Alpha=16.00-NCrit= 9.0-XTrTop=0.012
Fixed speed
Reynolds = 12
Mach = 0
Ncrit = 9
Forced Upper Trans. = 0
Forced Lower Trans. = 0
Alpha = 0
Cp = 0
tross
tness = 7.71%


Cl
Cd
Cl
Alpha
Cm
Alpha
albatross
T1_Re0.120_M0.03_N9.0
Cl
Xtr top
Hinge Fy
Alpha

A.4 Calculations - Jupyter Notebook

Analyzing an albatross wing

by: Neil Sawhney



Import the mean camber line from http://airfoiltools.com/airfoil/details?airfoil=goe173-il

In [ ]:
```
import pandas as pd

# Read the CSV file
df = pd.read_csv("albatross_foil-camber_line.csv")

# Display the first 5 rows
display(df.head())

x_foil = df["X(mm)"].tolist()
y_foil = df["Y(mm)"].tolist()
```

	X(mm)	Y(mm)
0	0.000	0.000000
1	1.241	0.978000
2	2.486	1.335821
3	4.978	2.261891
4	7.473	3.007482

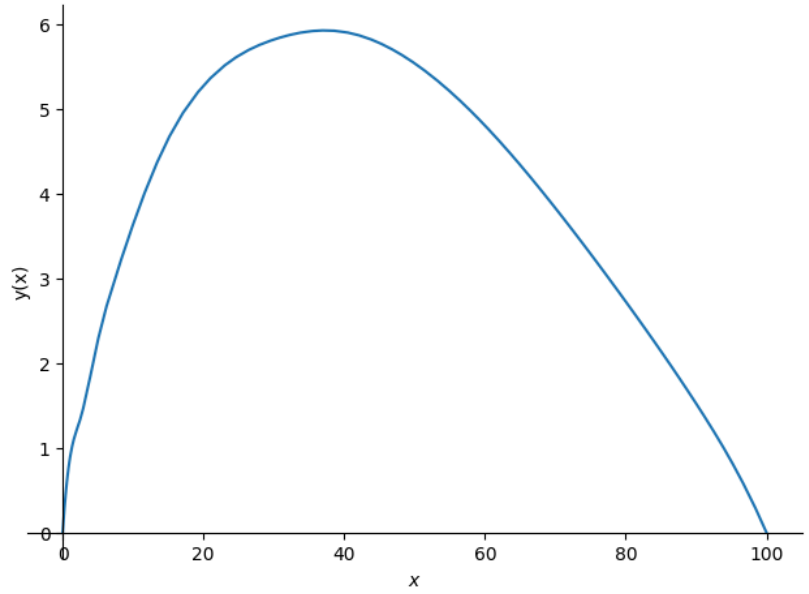
Interpolate the data into a function so that we can keep things analytical and symbolic

In [ ]:
```
import sympy as sp
from IPython.display import Markdown

alpha, theta, x = sp.symbols("alpha, theta, x")
chord_length_data = 100 # mm

# Construct an interpolating polynomial for x_foil and y_foil
y_func_x = sp.interpolating_spline(3, x, x_foil, y_foil)
display(Markdown("$y(x) = " + sp.latex(y_func_x) + "$"))
sp.plot(y_func_x, (x, 0, chord_length_data), ylabel="y(x)")
```

$$y(x) = \begin{cases} 0.0757438198145068x^3 - 0.48369211108273x^2 + 1.27168442585302x & \text{for } x \geq 0 \wedge x \leq 2.486 \\ -0.0164005768176812x^3 + 0.203520799000128x^2 - 0.436726868612968x + 1.41570349268081 & \text{for } x \geq 2.486 \wedge x \leq 4.978 \\ 0.0051175805847793x^3 - 0.117831363648218x^2 + 1.16296419705049x - 1.23871721561009 & \text{for } x \geq 4.978 \wedge x \leq 7.473 \\ -0.000365930224799395x^3 + 0.00510346519172705x^2 + 0.244272221129586x + 1.0497444964089 & \text{for } x \geq 7.473 \wedge x \leq 9.968 \\ -0.000199278194540061x^3 + 0.000119902878852579x^2 + 0.293948370264324x + 0.884687211550524 & \text{for } x \geq 9.968 \wedge x \leq 14.962 \\ 0.000245238196808445x^3 - 0.0198326598632167x^2 + 0.592478614011156x - 0.604182624096177 & \text{for } x \geq 14.962 \wedge x \leq 19.958 \\ 0.00012274930637512x^3 - 0.012498760037411 8x^2 + 0.446108641287747x + 0.369568014441756 & \text{for } x \geq 19.958 \wedge x \leq 29.956 \\ -3.90384183972492 \cdot 10^{-5}x^3 + 0.00204077921243152x^2 + 0.0105622035194419x + 4.71864437770423 & \text{for } x \geq 29.956 \wedge x \leq 39.956 \\ 2.88369943703519 \cdot 10^{-5}x^3 - 0.00609531076519523x^2 + 0.335647814665496x + 0.388937484720252 & \text{for } x \geq 39.956 \wedge x \leq 49.96 \\ 1.873961567553 \cdot 10^{-5}x^3 - 0.0045819156464153x^2 + 0.260038594531253x + 1.64808303068919 & \text{for } x \geq 49.96 \wedge x \leq 59.965 \\ 2.10662735920923 \cdot 10^{-5}x^3 - 0.00500046977231526x^2 + 0.28513719269084x + 1.14640388447584 & \text{for } x \geq 59.965 \wedge x \leq 69.972 \\ 6.97096097190038 \cdot 10^{-6}x^3 - 0.00204163812833513x^2 + 0.0781018248982628x + 5.97529680286998 & \text{for } x \geq 69.972 \wedge x \leq 79.98 \\ -1.90634019269676 \cdot 10^{-5}x^3 + 0.00420504690561922x^2 - 0.421508044117408x + 19.2948959108278 & \text{for } x \geq 79.98 \wedge x \leq 89.989 \\ -0.000149790157233174x^3 + 0.0394969568553698x^2 - 3.59739172858552x + 114.559761538028 & \text{for } x \geq 89.989 \wedge x \leq 100.0 \end{cases}$$



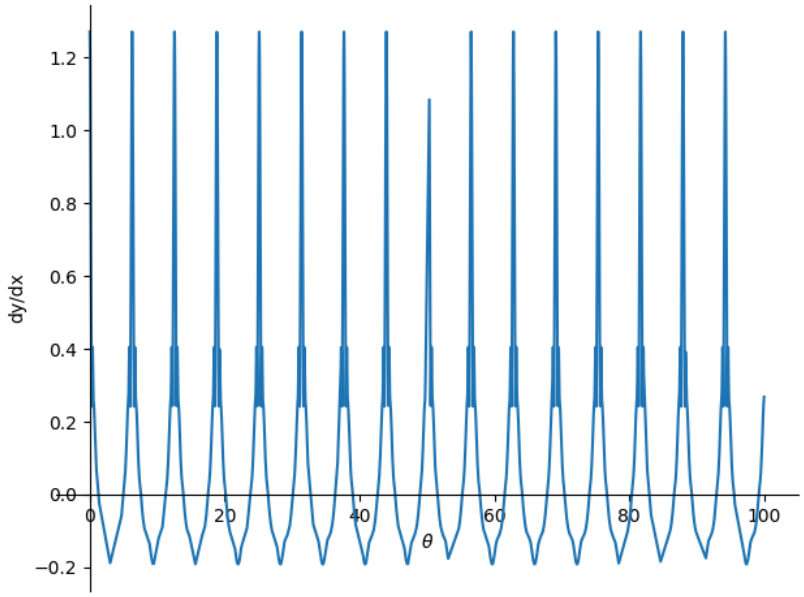
Out[ ]: <sympy.plotting.plot.Plot at 0x7f99db640410>

Now let's get derivatives and do a variable substitution to get the function in terms of θ instead of x

In [ ]:
```
# Differentiate y_func_x with respect to x
dydx_func_x = y_func_x.diff(x)

# Variable substitution from x to theta
dydx_func_theta = dydx_func_x.subs(x, chord_length_data / 2 * (1 - sp.cos(theta)))
display(Markdown("$\\frac{dy}{dx}(\\theta) = " + sp.latex(dydx_func_theta) + "$"))
sp.plot(dydx_func_theta, (theta, 0, chord_length_data), ylabel="dy/dx")
```

$$\frac{dy}{dx}(\theta) = \begin{cases} 568.078648608801(1 - \cos(\theta))^2 + 48.369211108273\cos(\theta) - 47.09752668242 & \text{for } 50.0\cos(\theta) - 50.0 \geq -2.486 \wedge 50.0\cos(\theta) \\ -123.004326132609(1 - \cos(\theta))^2 - 20.3520799000128\cos(\theta) + 19.9153530313998 & \text{for } 50.0\cos(\theta) - 50.0 \geq -4.978 \wedge 50.0\cos(\theta) \\ 38.3818543858447(1 - \cos(\theta))^2 + 11.7831363648218\cos(\theta) - 10.6201721677713 & \text{for } 50.0\cos(\theta) - 50.0 \geq -7.473 \wedge 50.0\cos(\theta) \\ -2.74447668599546(1 - \cos(\theta))^2 - 0.510346519172705\cos(\theta) + 0.754618740302291 & \text{for } 50.0\cos(\theta) - 50.0 \geq -9.968 \wedge 50.0\cos(\theta) \\ -1.49458645905046(1 - \cos(\theta))^2 - 0.0119902878852579\cos(\theta) + 0.305938658149582 & \text{for } 50.0\cos(\theta) - 50.0 \geq -14.962 \wedge 50.0\cos(\theta) \\ 1.83928647606334(1 - \cos(\theta))^2 + 1.98326598632167\cos(\theta) - 1.39078737231051 & \text{for } 50.0\cos(\theta) - 50.0 \geq -19.958 \wedge 50.0\cos(\theta) \\ 0.9206197978134(1 - \cos(\theta))^2 + 1.24987600374118\cos(\theta) - 0.803767362453429 & \text{for } 50.0\cos(\theta) - 50.0 \geq -29.956 \wedge 50.0\cos(\theta) \\ -0.292788137979369(1 - \cos(\theta))^2 - 0.204077921243152\cos(\theta) + 0.214640124762594 & \text{for } 50.0\cos(\theta) - 50.0 \geq -39.956 \wedge 50.0\cos(\theta) \\ 0.216277457777639(1 - \cos(\theta))^2 + 0.609531076519523\cos(\theta) - 0.273883261854027 & \text{for } 50.0\cos(\theta) - 50.0 \geq -49.96 \wedge 50.0\cos(\theta) \\ 0.140547117566475(1 - \cos(\theta))^2 + 0.45819156464153\cos(\theta) - 0.198152970110277 & \text{for } 50.0\cos(\theta) - 50.0 \geq -59.965 \wedge 50.0\cos(\theta) \\ 0.157997051940692(1 - \cos(\theta))^2 + 0.500046977231526\cos(\theta) - 0.214909784540686 & \text{for } 50.0\cos(\theta) - 50.0 \geq -69.972 \wedge 50.0\cos(\theta) \\ 0.0522822072892529(1 - \cos(\theta))^2 + 0.204163812833513\cos(\theta) - 0.12606198793525 & \text{for } 50.0\cos(\theta) - 50.0 \geq -79.98 \wedge 50.0\cos(\theta) \\ -0.142975514452257(1 - \cos(\theta))^2 - 0.420504690561922\cos(\theta) - 0.00100335355548548 & \text{for } 50.0\cos(\theta) - 50.0 \geq -89.989 \wedge 50.0\cos(\theta) \\ -1.1234261792488(1 - \cos(\theta))^2 - 3.94969568553698\cos(\theta) + 0.352303956951467 & \text{for } 50.0\cos(\theta) - 50.0 \geq -100.0 \wedge 50.0\cos(\theta) \end{cases}$$



Out[ ]: <sympy.plotting.plot.Plot at 0x7f99db4caf90>

Now we can get the lift coefficient!

In [ ]:
```
# Calculate the lift coefficient
coeff_lift_albatross = (
    2
    * sp.pi
    * (
        alpha
        + (1 / sp.pi)
        * sp.integrate(dydx_func_theta * (sp.cos(theta) - 1), (theta, 0, sp.pi))
    )
).evalf()

# Display the equation
display(Markdown("$C_l = " + sp.latex(coeff_lift_albatross) + "$"))
```

$C_l = 6.28318530717959\alpha + 0.63959809779726$

Calculate lift for a speed v, with a span of b, chord_length of c, through air with density ρ

In [ ]:
```
cruising_velocity, span_length, density, chord_length, coeff_lift = sp.symbols("u_{\\infty}, b, rho, c, C_l")

# Calculate the lift
lift = sp.Rational(1, 2) * density * cruising_velocity ** 2 * span_length * chord_length * coeff_lift
display(Markdown("$L = " + sp.latex(lift) + "$"))
```

$L = \frac{C_l b c \rho u_{\infty}^2}{2}$

Find the angle of attack at steady altitude (α_0)

In [ ]:
```
albatross_weight = sp.symbols("W")

# Find the angle of attack at steady altitude
alpha_0 = sp.solve((lift - albatross_weight).subs(coeff_lift, coeff_lift_albatross), alpha, dict=True)[0][alpha]
display(Markdown("$\\alpha_0 = " + sp.latex(alpha_0) + "$"))
```

$\alpha_0 = \frac{0.31830988618379W}{bc\rho u_{\infty}^2} - 0.101795198856607$

Evaluating for properties of an Albatross moving at cruising speed

In [ ]:
```
albatross_data = {
    "W": 11*9.81,  # N
    "u_{\\infty}": 20,  # m/s
    "b": 1.5,  # m
    "rho": 1.225,  # kg/m^3
    "c": 0.3,  # m
    "C_l": coeff_lift_albatross,
```

```
}


def radians_to_degrees(radians):
    return 180 / sp.pi * radians


def degrees_to_radians(degrees):
    return sp.pi / 180 * degrees


# Compute the angle of attack at steady altitude in degrees
alpha_0_albatross = radians_to_degrees(alpha_0).evalf(subs=albatross_data)
display(Markdown("$\\alpha_0 = " + sp.latex(round(alpha_0_albatross, 2)) + "^\\circ $"))
```

$\alpha_0 = 3.09^\circ$

Lets compare the lift to a small aircraft with a 50 mph cruise speed, 6 ft span, and 1 ft chord length, using the NACA 4412 airfoil at 2 degrees angle of attack

In [ ]:
```
aircraft_data = {
    "u_{\\infty}": 22.352,  # m/s
    "b": 1.8288,  # m
    "rho": 1.225,  # kg/m^3
    "c": 0.3048,  # m
    "C_l": 6.28 * degrees_to_radians(2) + 25.18,
}

# Compute the lift at a 2 degree angle of attack for the albatross
lift_albatross = lift.subs(albatross_data).subs(alpha, degrees_to_radians(2)).evalf()
display(Markdown("Albatross: $L = " + sp.latex(round(lift_albatross, 2)) + " N$"))

# Compute the lift at a 2 degree angle of attack for the aircraft
lift_aircraft = lift.subs(aircraft_data).subs(alpha, degrees_to_radians(2)).evalf()
display(Markdown("Aircraft: $L = " + sp.latex(round(lift_aircraft, 2)) + " N$"))
```

Albatross: $L = 189.39N$

Aircraft: $L = 4332.52N$

Airfoil wing area S, span b, aspect ratio AR, chord vs. position (c(y)), twist distribution, lift to drag slope m and zero lift angle of attack are obtained / estimated from http://airfoiltools.com/airfoil/details?airfoil=goe173-il.

In [ ]:
```
import sympy as sp
from IPython.display import display, Markdown

(
    b_span,  # chord length
    V_infty,  # free stream velocity
    theta_0,  # specific spanwise coordinate parameter
    alpha,  # geometric AOA
    AR,  # aspect ratio
    S,  # planiform area
) = sp.symbols("b_{span}, V_{\\infty}, \\theta_{0}, \\alpha, AR, S")

vals = {
    b_span : 3.1,  # (meter)
    V_infty : 20,  # (meter/second)
    S : 2*.336,  # (meter^2)
}

derived_vals = {
    AR : (b_span**2/S).subs(vals),
}
vals.update(derived_vals)
```

In [ ]:
```
(
    c,  # chord length (m)
    m_0,  # section lift slope (1/rad)
    alpha_abs,  # absolute angle of attack [alpha - alpha_L0] (radians)
) = sp.symbols("c, m_0, \\alpha_{a}")

wing_vals = {
    # c : (3 + (b_span/8) * sp.cos(theta_0)).subs(vals),  # (meter)
    m_0 : (5.729),  # (1/rad)
    # alpha_abs : (-.1013 + .14 + .0872),  # (radian)
    # alpha_abs : (-.1013 + .0872),  # (radian)
    alpha_abs : (0.05),  # (radian)
}
vals.update(wing_vals)
display(
    Markdown(
        f'${sp.latex(vals)}$'
```

```
    )
)
```

$\{AR : 14.3005952380952,\ S : 0.672,\ V_\infty : 20,\ \alpha_a : 0.05,\ b_{span} : 3.1,\ m_0 : 5.729\}$

In [ ]:
```
(
    A_1,
    A_2,
    A_3,
) = sp.symbols("A_1, A_2, A_3")

A_n_eq = sp.Eq(
    alpha_abs,
    ((4 * b_span * sp.sin(theta_0)) / (m_0 * c) + 1) * A_1
    + (
        ((4 * b_span * sp.sin(2 * theta_0)) / (m_0 * c))
        + 2 * ((sp.sin(2 * theta_0)) / (sp.sin(theta_0)))
    )
    * A_2
    + (
        ((4 * b_span * sp.sin(3 * theta_0)) / (m_0 * c))
        + 3 * ((sp.sin(3 * theta_0)) / (sp.sin(theta_0)))
    )
    * A_3
)

display(A_n_eq)
```

$$\alpha_a = A_1 \cdot \left(\frac{4 b_{span} \sin(\theta_0)}{c m_0} + 1 \right) + A_2 \cdot \left(\frac{4 b_{span} \sin(2\theta_0)}{c m_0} + \frac{2 \sin(2\theta_0)}{\sin(\theta_0)} \right) + A_3 \cdot \left(\frac{4 b_{span} \sin(3\theta_0)}{c m_0} + \frac{3 \sin(3\theta_0)}{\sin(\theta_0)} \right)$$

In [ ]:
```
A_n_sol = sp.solve(
    [
        A_n_eq.subs(vals).subs(
            {
                theta_0: 1.47,
                c : 642.33e-3
            }
        ),
        A_n_eq.subs(vals).subs(
            {
                theta_0: 1.2,
                c : 512.53e-3
            }
        ),
        A_n_eq.subs(vals).subs(
            {
                theta_0: 0.98,
                c : 499.97e-3
            }
        ),
    ],
    [A_1, A_2, A_3],
    dict=True,
)[0]

vals[A_1] = A_n_sol[A_1]
vals[A_2] = A_n_sol[A_2]
vals[A_3] = A_n_sol[A_3]

display(
    Markdown(
        f'${sp.latex(A_1)} = {sp.latex(A_1.subs(vals))}$'
    )
)

display(
    Markdown(
        f'${sp.latex(A_2)} = {sp.latex(A_2.subs(vals))}$'
    )
)

display(
    Markdown(
        f'${sp.latex(A_3)} = {sp.latex(A_3.subs(vals))}$'
    )
)
```

$A_1 = 0.0189150712091736$

$A_2 = -0.00696214110305495$

$A_3 = 0.00407350397624457$

Calculate δ

In [ ]:
```
(
    delta,
    efficiency,
) = sp.symbols("\\delta, e")

efficiency_vals = {
    delta : (2*(A_2/A_1)**2 + 3*(A_3/A_1)**2).subs(vals)
}
vals.update(efficiency_vals)

efficiency_expr = 1/(1+delta)
vals[efficiency] = efficiency_expr.subs(vals)

display(
    Markdown(
        f'${sp.latex(efficiency)} = {sp.latex(efficiency_expr)} = {sp.latex(efficiency.subs(vals))}$'
    )
)
```

$e = \frac{1}{\delta+1} = 0.709172968837333$

Calculate $C_L, C_{D,i}$

In [ ]:
```
(
    C_L,
    C_D_induced,
) = sp.symbols("C_L, C_{D\,i}")

drag_coeff_vals = {
    C_L : (A_1 * sp.pi * AR).subs(vals),
}
vals.update(drag_coeff_vals)

C_D_induced_exprs = (C_L**2 * (1+delta)) / (sp.pi*AR)
vals[C_D_induced] = C_D_induced_exprs.subs(vals)

display(
    Markdown(
        f'${sp.latex(C_L)} = {sp.latex(C_L.subs(vals).evalf())}$'
    )
)

display(
    Markdown(
        f'${sp.latex(C_D_induced)} = {sp.latex(C_D_induced_exprs)} = {sp.latex(C_D_induced.subs(vals))}$'
    )
)

Re = albatross_data["rho"] * albatross_data["u_{\\infty}"] * albatross_data["c"] / 1.81e-5
c_f = 1.328 / sp.sqrt(Re)
c_d_viscous = 2*c_f
display(
    Markdown(
        r'$c_{d, viscous}$ = ' + sp.latex(c_d_viscous.subs(vals).evalf().round(5))
    )
)

c_d_total = C_D_induced + c_d_viscous

display(
    Markdown(
        r'$c_{d, total}$ = ' + sp.latex(c_d_total.subs(vals).evalf().round(5))
    )
)
```

$C_L = 0.849790688266456$

$C_{D,i} = \frac{C_L^2(\delta+1)}{\pi AR} = 0.00721469377513592\pi$

$c_{d,viscous}$ = 0.00417

$c_{d,total}$ = 0.02683

Calculate D_i

In [ ]:
```
(
    rho_air,
    q_infty,
```

```
    D_induced,
) =  sp.symbols("rho_{air}, q_{\\infty}, D_{i}")

drag_vals = {
    rho_air : 1.293e-3,  # (kg/meter^3)
    q_infty : .5 * rho_air * V_infty**2,  # (kg/(meter*second^2))
}
vals.update(drag_vals)

D_induced_exprs = C_D_induced * q_infty * S
vals[D_induced] = D_induced_exprs.subs(vals)

display(
    Markdown(
        f'${sp.latex(D_induced)} = {sp.latex(D_induced_exprs)} = {sp.latex(D_induced.subs(vals).evalf())}$'
    )
)

total_drag = D_f + D_induced
display(
    Markdown(
        f'$D = {sp.latex(total_drag.subs(vals).evalf().round(5))}$'
    )
)
```

$D_i = C_{D,i} S q_\infty = 0.00393881486849008$

$D = 0.00811$

Induced Drag Results

- 0 deg AOA
  - .37 Newtons
- 8 deg AOA
  - 29.45 Newtons