Introduction to Aerodynamics and Python

Project 0

Manbir Kaberwal

EAE 127

Professor Robinson

10/14/2023

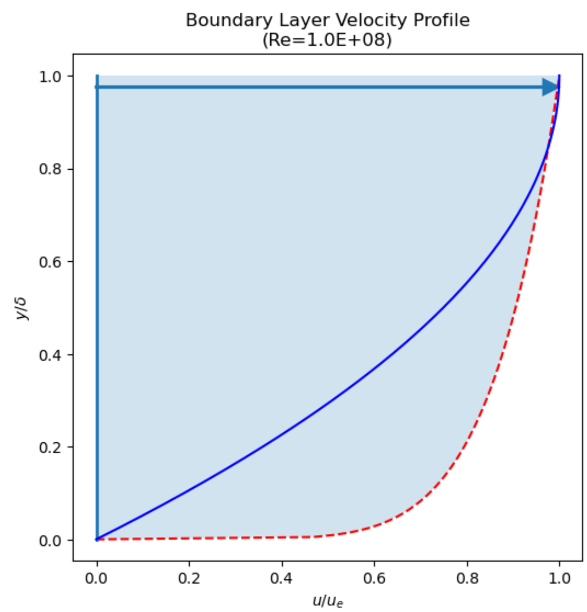
Contents

Introduction

Performed fives task to become more familiar with various aerodynamic concepts and how to connect lessons learned in the classroom to application with Python. While understanding the basics of Python syntax and development.

Results

1.) Boundary Layers and Numeric Integration



Here we can see the boundary lanes for both laminar and turbulent boundaries. Blue line representing Laminar and red representing turbulent layer. There was some issue when it came to comparing the generated graph versus the solution provided. But the root boundary layer profiles are visible. To plot both profiles we used the two governing equations below.

$$Laminar: \frac{u}{u_e} \approx (2\left(\frac{y}{\delta}\right) - (\frac{y}{\delta})^2)$$

$$Turbulent: \frac{u}{u_e} \approx (\frac{y}{\delta})^7$$

1.2) Boundary Layer Thickness

```
|-----------------------------------------------------------------|
|Displacement Thickness: 5.219319208243424 in (Re=1.0E+08, L=300)|
|-----------------------------------------------------------------|
BL Thickness: 41.45389476486636 in
```

With a visual of the boundary layers, we can discover how it translates to estimating the thickness. We used a Boeing 747 as and example at 300 ft, and Reynolds number of 1e8.

Using:

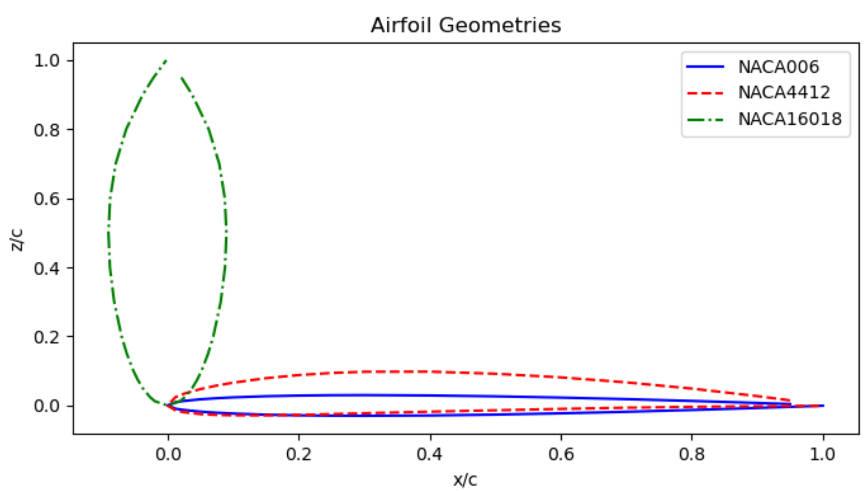
$$\delta(x) = \frac{0.16x}{Re_x^{\frac{1}{7}}}$$

Furthermore:

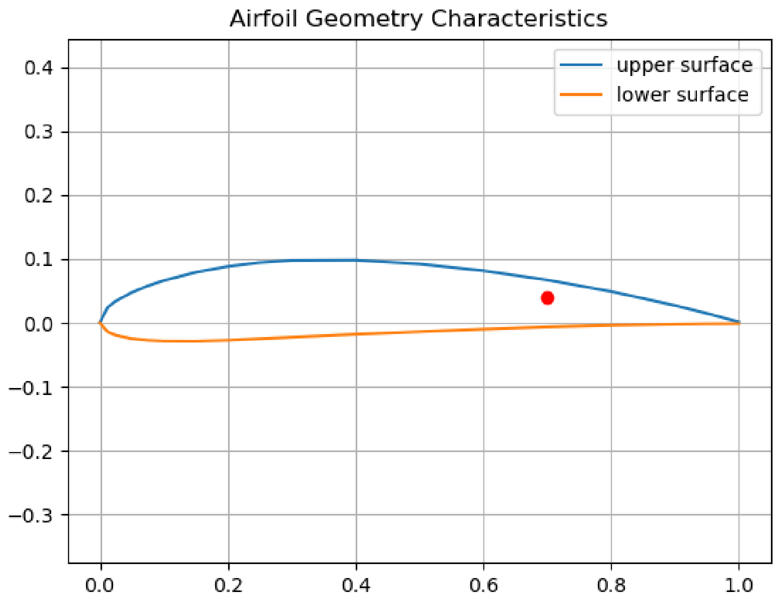
$$\delta(x) = \frac{\delta^*}{\delta}(x) \cdot \delta(x)$$

Using these equations, we can calculate Displacement thickness to be $\delta^* = 5.219\ in$ this indicates the potential flow is displaced 5.2 in away from fuselage surface.

2.) Airfoil Plotting and Line Integration



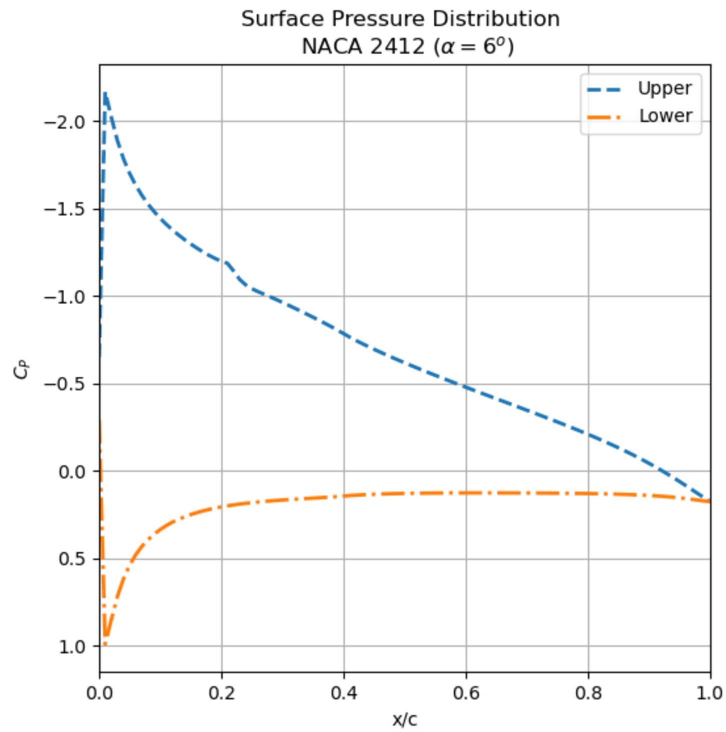
Here we plotted 3 airfoils as shown in the legend. We took .dat files from http://airfoiltools.com and plot the geometry of each airfoil. This gave an opportunity of added multiple data sets on one plot as well and challenge us to distinguish each data set uniquely. As well we plotted the Airfoil Geometry Characteristics for one of the airfoils from above. Next page you fill find the graph with the labeled geometry.



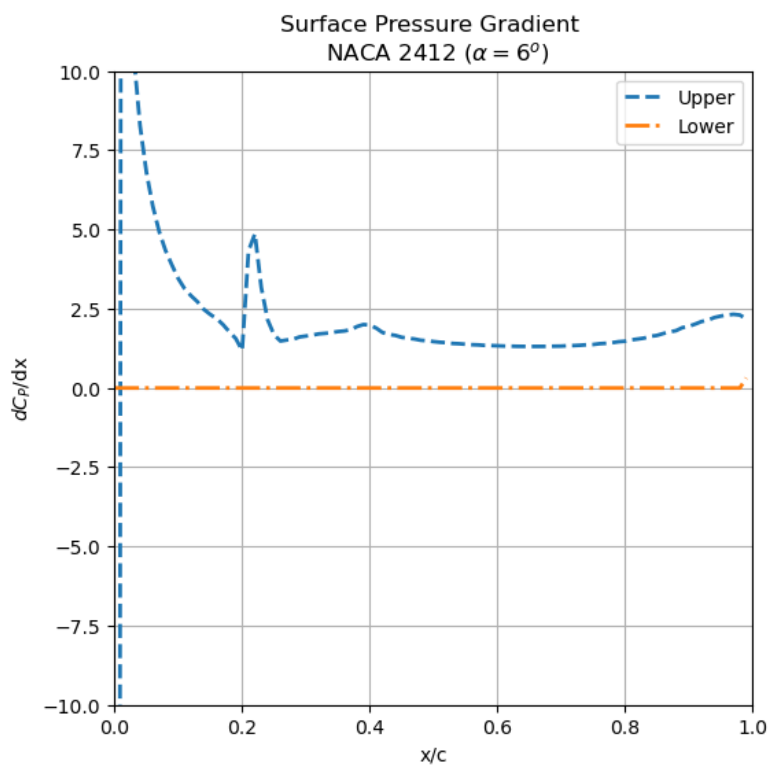
Airfoil Geometry Characteristics
upper surface
lower surface
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.2
0.4
0.6
0.8
1.0

3.) Airfoil Surface Pressure and Numeric Differentiation

3.1) Airfoil Surface Pressure



3.2) Surface Pressure Gradient and Numeric Differentiation



4.) Linear Algebra

$$
\begin{aligned}
4w + 2x + 3y + 2z &= 10 \\
-3w + 1x - 2y + 3z &= 9 \\
x + 2y + 1z &= -3 \\
3w + x - y - 2z &= -5
\end{aligned}
$$

The next exercise we were tasked with was getting familiar with using Python to solve matrix operations. With the system of equations from above we can form the following matrix:

$$
\underbrace{\begin{bmatrix} 4 & 2 & 3 & 2 \\ -3 & 1 & -2 & 3 \\ 0 & 1 & 2 & 1 \\ 3 & 1 & -1 & -2 \end{bmatrix}}_{A}
\underbrace{\begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}}_{\lambda}
=
\underbrace{\begin{bmatrix} 10 \\ 9 \\ -3 \\ -5 \end{bmatrix}}_{b}
$$

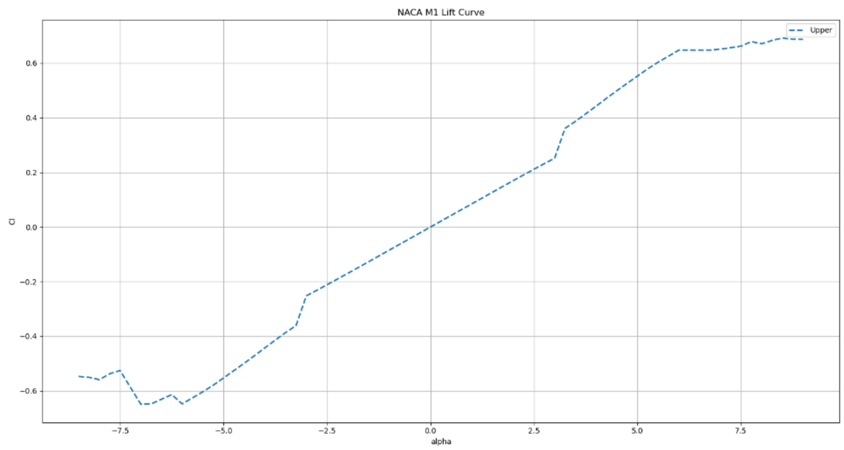
We can then set up our code to solve for Lambda with the result output being:

```
Results in:
w = 3.413333
x = -4.733333
y = -2.346667
z = 6.426667
```

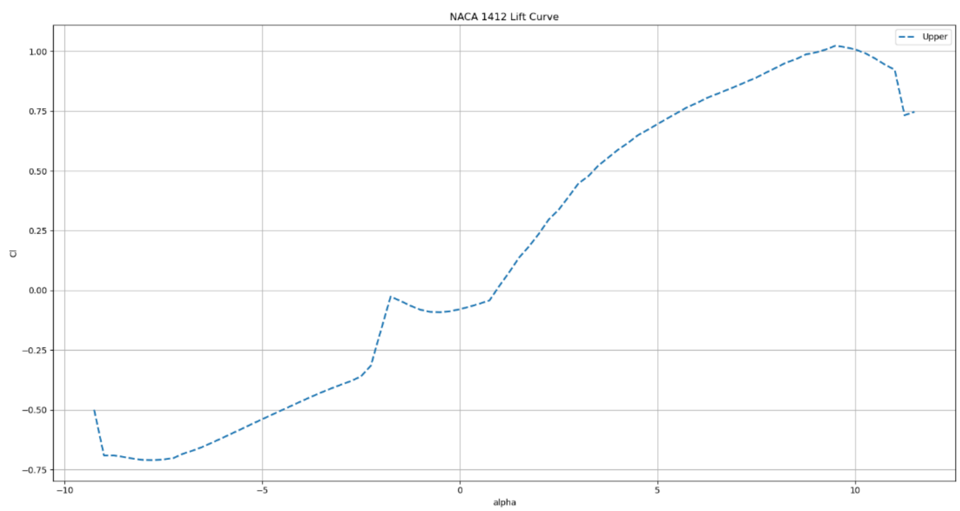
5.) Lift Curves and Linear Interpolation

In this section we retrieved coefficient of lift against angle of attack data to give us a lift curve for two air foils, a symmetric and cambered airfoil. With these plots we can analyze how the coefficient of lift compares between the airfoils at various angle of attacks.

Symmetric Airfoil:



Cambered Airfoil



5.2) Linear Interpolation

	Symmetric Airfoil	Cambered Airfoil
Alpha	5.65	5.65
Cl	0.616	0.756

With the graphs and data available we can interpolate coefficient values at a singular alpha value to see how each airfoil compares. At and alpha value of 5.65 degrees we can see the cambered airfoil has a greater coefficient of lift.

Conclusion

There is a present learning curve being introduced to Python just recently, but its application if visualizing aerodynamic data is quite unique and useful. Plots customization and variable definition seem to be tricky but taking a governing formula and data are relatively easy. Overall, this project helped get my toes in the water but there is still quite I bit I can learn more about. I attempted to use language models to help and come the codes with the given solution and unfortunately, I found the language model though helpful seems to be lacking, It was use is terms of learning what a possible function is capable of and how to implement it into my script. I wish to attempt and see how GitHub copilot is cable of.