ME-285: Turbomachine Theory, Assignment 1

Manish Sharma

October 2025

Problem Statement

Apply the vortex panel method to analyse the flow over a NACA0012 airfoil. The airfoil will be discretized into a specified number of panels, each with a continuous vortex sheet of strength (γ). The free-stream velocity U is inclined at an angle of attack α.

The solution is implemented in Python, and the code includes the ability to vary the number of panels for accuracy.

1 Introduction

The vortex panel method is a boundary element technique used to model inviscid, incompressible, and irrotational flow over airfoils. It is particularly effective for symmetric airfoils like NACA0012 at moderate angles of attack.

2 Airfoil Resampling and Panel Generation

Panels are need to be named in clockwise direction starting from Trailing edge $\rightarrow$ lower surface $\rightarrow$ Leading edge $\rightarrow$ upper surface $\rightarrow$ Trailing edge. A key requirement is the ability to vary the panel count. To achieve this robustly, a resampling function is used to generate m panels with optimal spacing.

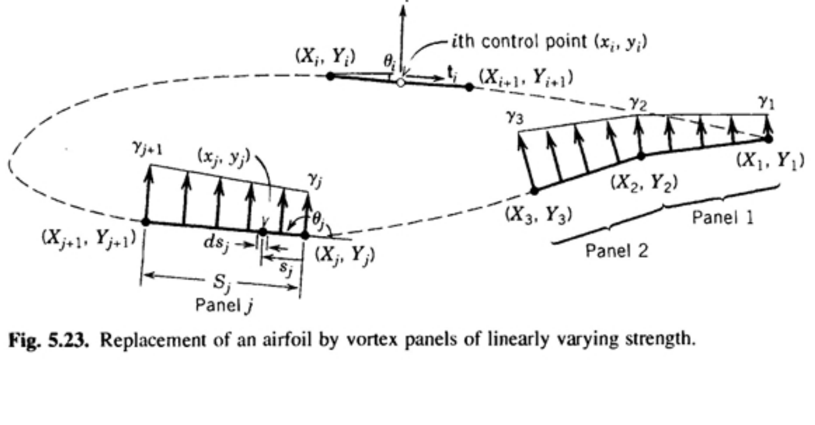

Fig. 5.23. Replacement of an airfoil by vortex panels of linearly varying strength.

To ensure accurate vortex panel analysis, the original **NACA0012 airfoil coordinates downloaded from https://www.engapplets.vt.edu/fluids/vpm/naca0012.txt(Virginia Tech) are resampled using cosine spacing**.

Reason for not using standard formula to generate coordinate file is they don't have sharp trailing edge because of which C_p over surface doesn't form a closed loop. In reality Sharp trailing edge is not possible but for such numerical method ,its valid to consider sharp trailing edge to ensure the kutta condition.

Explanation of the Resampling Procedure

1. **Identifying the Leading Edge:** The leading edge is located at the minimum x-coordinate:
$$\text{LE index} = \arg\min(X_{\text{orig}})$$
This divides the airfoil into two segments.

2. **Splitting Upper and Lower Surfaces:** The input coordinate file is assumed to follow the format:
$$\text{TE} \rightarrow \text{Lower Surface} \rightarrow \text{LE} \rightarrow \text{Upper Surface} \rightarrow \text{TE}$$

3. **Cosine Spacing:** For m_{new} total panels, each surface receives:
$$N_{\text{surf}} = \frac{m_{\text{new}}}{2} + 1$$
points with:
$$x(\theta) = \frac{1}{2}(1 - \cos\theta), \quad \theta \in [0, \pi]$$
This clusters points near the leading and trailing edges, improving accuracy.

4. **Interpolation Using Cubic Splines:** The original surfaces are interpolated using cubic spline interpolation (`interp1d` with `kind='cubic'`) to compute:
$$(X_{\text{lower,new}}, Y_{\text{lower,new}}), \quad (X_{\text{upper,new}}, Y_{\text{upper,new}})$$

5. **Reconstructing a Continuous Closed Airfoil Path:** To ensure proper vortex-panel traversal, the final coordinate order is:
$$\text{TE} \rightarrow \text{Lower Surface} \rightarrow \text{LE} \rightarrow \text{Upper Surface} \rightarrow \text{TE}$$

Effect of Resampling Using Cosine Spacing

- Higher resolution near the leading edge, improving accuracy of C_p peaks.
- Smooth transition around the trailing edge, aiding Kutta condition enforcement.

2.1 Resampling Function

```
def resample_airfoil_panels(X_orig, Y_orig, m_new):
    # Identify leading edge index
    le_index = np.argmin(X_orig)

    # Split into lower and upper surfaces
    lower_X = X_orig[:le_index+1]
    lower_Y = Y_orig[:le_index+1]
    upper_X = X_orig[le_index:]
    upper_Y = Y_orig[le_index:]

    # Cosine spacing from 0 to 1
    N_surf = m_new // 2 + 1
    x_dist = 0.5 * (1 - np.cos(np.linspace(0, np.pi, N_surf)))

    # Interpolate lower surface (flip for  L E TE )
    fl = interp1d(lower_X[::-1], lower_Y[::-1], kind='cubic')
    X_lower_new = x_dist
    Y_lower_new = fl(X_lower_new)

    # Interpolate upper surface
    fu = interp1d(upper_X, upper_Y, kind='cubic')
    X_upper_new = x_dist
    Y_upper_new = fu(X_upper_new)

    # Reorder: TE     LE     TE
    X_final = np.concatenate([X_lower_new[::-1], X_upper_new[1:]])
    Y_final = np.concatenate([Y_lower_new[::-1], Y_upper_new[1:]])

    return X_final, Y_final
```

Listing 1: Airfoil Resampling with Cosine Spacing

3 Governing Equations and Constants

3.1 Velocity Potential

The velocity potential at the i-th control point (x_i, y_i) in a uniform flow V_∞ at angle of attack α is given by:

$$\phi(x_i, y_i) = V_\infty(x_i \cos\alpha + y_i \sin\alpha) - \sum_{j=1}^{m} \frac{1}{2\pi} \int_{S_j} \gamma(s) \tan^{-1}\left(\frac{y_i - y(s)}{x_i - x(s)}\right) ds$$

where $\gamma(s)$ is the linearly varying vortex strength on panel j.

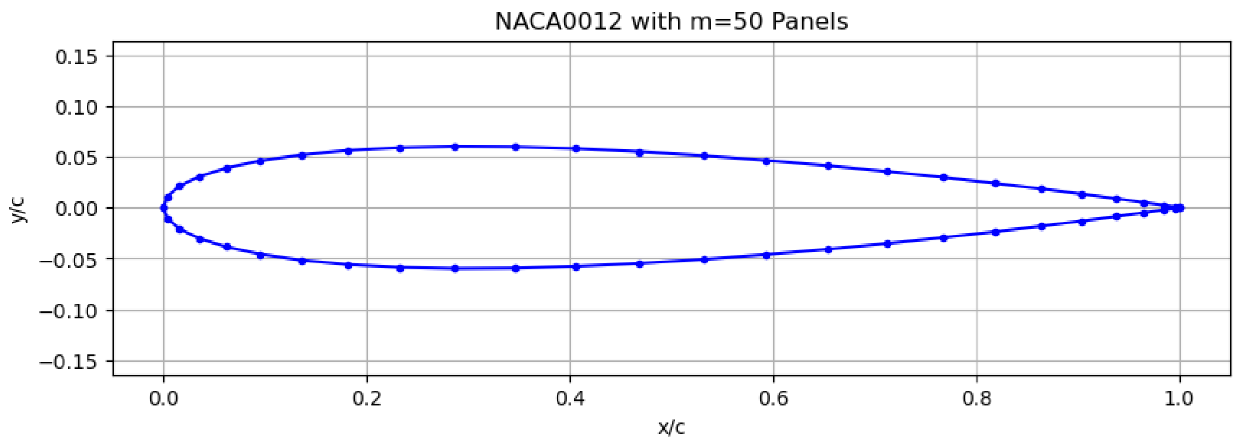

Figure 1: Resampled NACA0012 with $m = 100$ Panels (LE/TE Clustered)

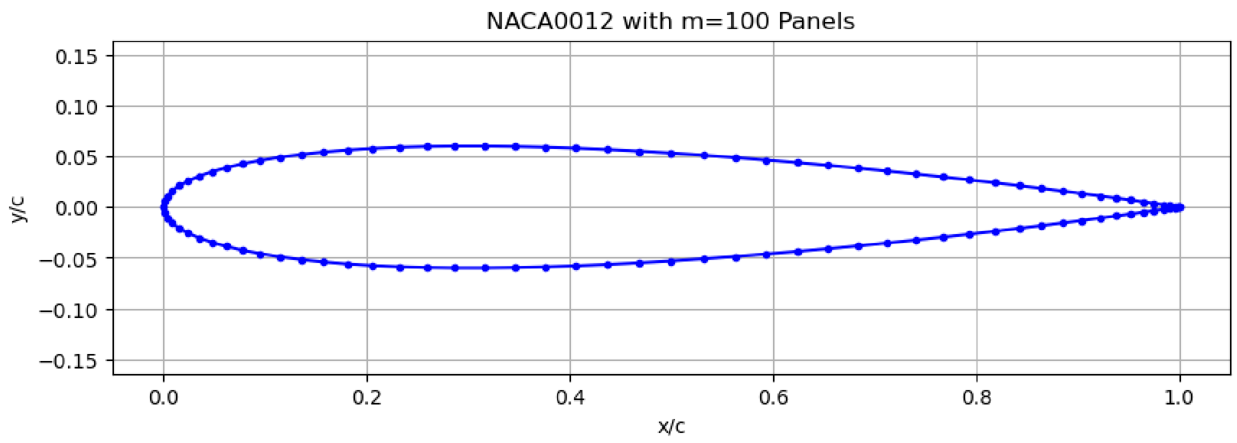

Figure 2: Resampled NACA0012 with $m = 50$ Panels (LE/TE Clustered)

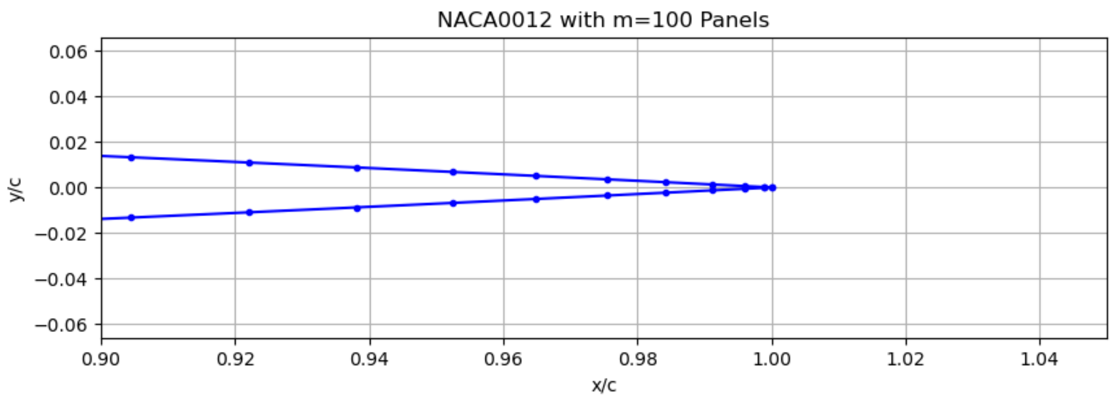

Figure 3: Sharp (Closed) Trailing edge)

3.2 Vortex Strength Distribution

The vortex strength on each panel varies linearly:

$$\gamma(s) = \gamma_j + (\gamma_{j+1} - \gamma_j)\,\frac{s}{S_j}$$

where s is the distance along the panel of length S_j.

3.3 No-Penetration Boundary Condition

The normal velocity at each control point must vanish:

$$\sum_{j=1}^{m} \left(C_{n1_{ij}}\gamma'_j + C_{n2_{ij}}\gamma'_{j+1}\right) = \sin(\theta_i - \alpha), \quad i = 1, 2, \ldots, m$$

wher $\gamma' = \frac{\gamma}{2\pi V_\infty}$ non dimensional circulation.

3.4 Kutta Condition

To ensure smooth flow at the trailing edge:

$$\gamma'_1 + \gamma'_{m+1} = 0$$

3.5 Linear System Formulation

Combining the boundary and Kutta conditions, we obtain a system of $m + 1$ equations:

$$\sum_{j=1}^{m+1} A_{ij}\gamma_j = \mathrm{RHS}_i, \quad i = 1, 2, \ldots, m + 1$$

with:

$$\text{For } i < m+1 : \begin{cases} A_{i1} = C_{n1_{i1}} \\ A_{ij} = C_{n1_{ij}} + C_{n2_{i,j-1}}, \quad j = 2, \ldots, m \\ A_{i,m+1} = C_{n2_{i,m}} \\ \mathrm{RHS}_i = \sin(\theta_i - \alpha) \end{cases}$$

$$\text{For } i = m+1 : \begin{cases} A_{m+1,1} = 1 \\ A_{m+1,m+1} = 1 \\ A_{m+1,j} = 0, \qquad j = 2, \ldots, m \\ \mathrm{RHS}_{m+1} = 0 \end{cases}$$

3.6 Constants for Influence Coefficients

Let:

$$\begin{aligned}
A &= -(x_i - X_j)\cos\theta_j - (y_i - Y_j)\sin\theta_j \\
B &= (x_i - X_j)^2 + (y_i - Y_j)^2 \\
C &= \sin(\theta_i - \theta_j) \\
D &= \cos(\theta_i - \theta_j) \\
E &= (x_i - X_j)\sin\theta_j - (y_i - Y_j)\cos\theta_j \\
F &= \ln\left(1 + \frac{S_j(S_j + 2A)}{B}\right) \\
G &= \tan^{-1}\left(\frac{ES_j}{B + AS_j}\right) \\
P &= (x_i - X_j)\sin(\theta_i - 2\theta_j) + (y_i - Y_j)\cos(\theta_i - 2\theta_j) \\
Q &= (x_i - X_j)\cos(\theta_i - 2\theta_j) - (y_i - Y_j)\sin(\theta_i - 2\theta_j)
\end{aligned}$$

Then the influence coefficients are:

$$\begin{aligned}
C_{n1_{ij}} &= 0.5DF + CG - C_{n2_{ij}} \\
C_{n2_{ij}} &= D + \frac{0.5QF}{S_j} - \frac{(AC + DE)G}{S_j} \\
C_{t1_{ij}} &= 0.5CF - DG - C_{t2_{ij}} \\
C_{t2_{ij}} &= C + \frac{0.5PF}{S_j} + \frac{(AD - CE)G}{S_j}
\end{aligned}$$

For $i = j$ (self-induced terms):

$$C_{n1_{ii}} = -1, \quad C_{n2_{ii}} = 1, \quad C_{t1_{ii}} = C_{t2_{ii}} = \frac{\pi}{2}$$

3.7 Tangential Velocity

The tangential velocity at the i-th control point is:

$$V_i = \cos(\theta_i - \alpha) + \sum_{j=1}^{m+1} \left(C_{t1_{ij}}\gamma_j + C_{t2_{ij}}\gamma_{j+1}\right)$$

3.8 Pressure Coefficient

This follows directly from Bernoulli's equation:

$$C_{p_i} = 1 - V_i^2$$

where V_i is non-dimensional velocity V/V_∞.

The pressure coefficient is calculated from the tangential velocity at each control point:

```
v[i] = np.cos(theta[i] - np.radians(alpha))
for j in range(m+1):
    v[i] += At[i, j] * gamma[j]

C_p[i] = 1 - v[i]**2
```

Listing 2: Tangential Velocity and Pressure Coefficient

Pressure Coefficient Distribution

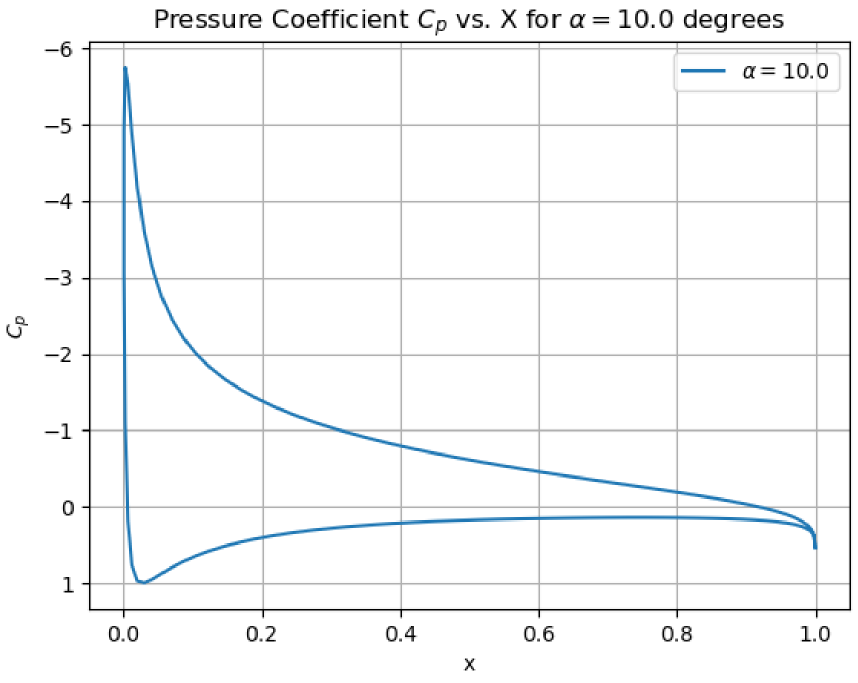

Figure 4: Pressure coefficient (C_p) distribution over the airfoil surface

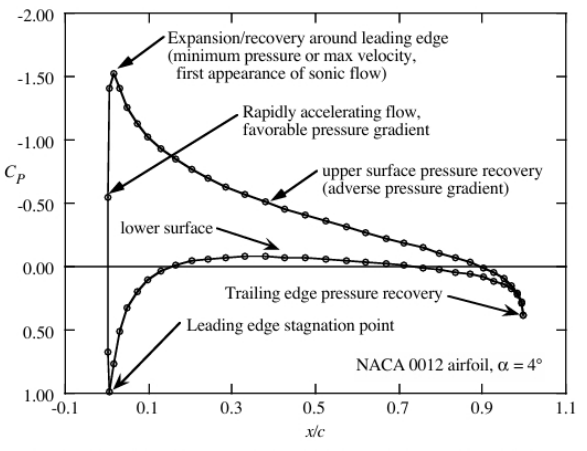

Figure 6-11. Key areas of interest when examining airfoil pressure distributions.

Variation with Angle of Attack

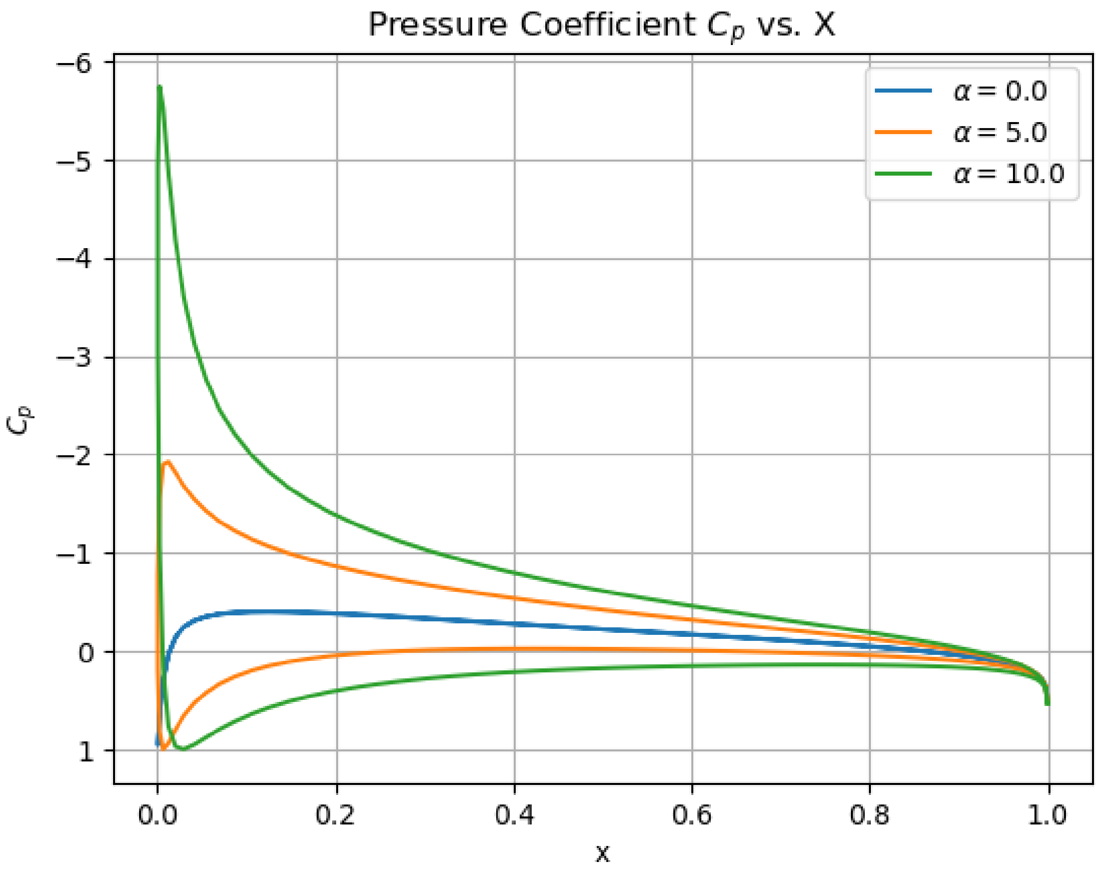

Figure 5: Variation of C_p with angle of attack

3.9 Lift Coefficient

The lift coefficient is computed by integrating the pressure difference between upper and lower surfaces:

```
Cl = np.trapz(CP_lower - CP_upper, x_mid)
```

Listing 3: Lift Coefficient Calculation

This approximates:

$$C_l = \int_0^1 (C_{p_{\text{lower}}} - C_{p_{\text{upper}}})\, dx$$

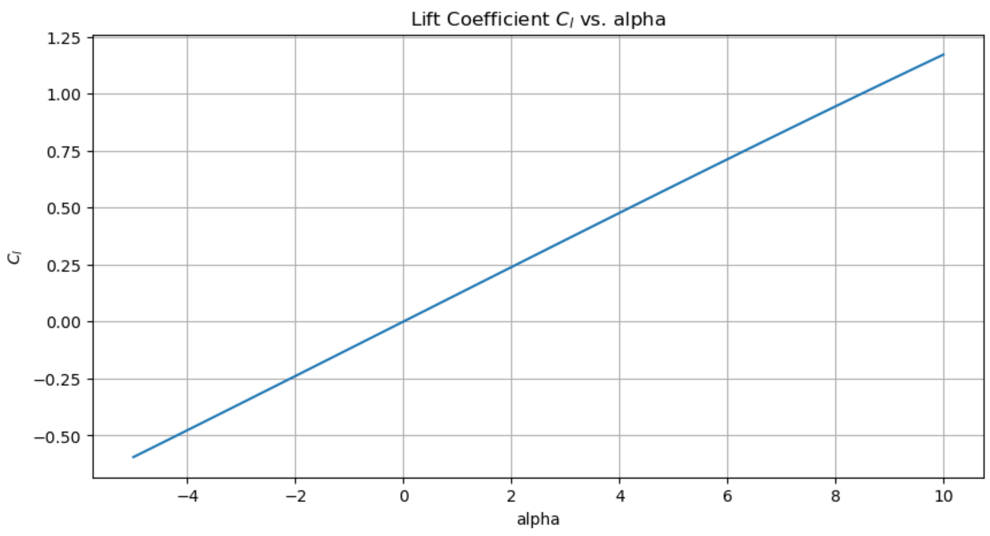

Figure 6: Co-effiecient of lift vs angle of attack

3.10 Center of Pressure and Torque

By definition center of pressure is the point where total aerodynamic forces act on the surface or the resultant pressure force acts at this point.

Therefore torque about CP is always zero.

The center of pressure and torque are computed from the pressure coefficient distribution.

This computes the moment coefficient about the leading edge and shifts it to the center of pressure using:

$$x_{cp} = -\frac{C_{m_{\mathrm{LE}}}}{C_l}, \quad C_{m_{cp}} = C_{m_{\mathrm{LE}}} + C_l \cdot x_{cp}$$

Center of pressure is not defined at alpha=0 as there is no lift for a symmetric airfoil. Therfore to avoid division by zero alpha=0 is excluded. In the vicinity of $\alpha = 0$,x_{cp} rises because of this reason but in reality it is ≈ 0.25 constant, which is the aerodynamic center according to thin airfoil theory.

```
x_moment_arm = x[mid:]  # x locations on upper surface
moment_integrand = (CP_lower - CP_upper) * x_moment_arm

Cm_LE = -np.trapz(moment_integrand, x_moment_arm)

if alpha == 0:
    x_cp = np.nan
    Cm_cp = 0
else:
    x_cp = -Cm_LE / Cl
    Cm_cp = Cm_LE + Cl * x_cp
```

Listing 4: Center of Pressure and Moment Coefficient

Center of Pressure

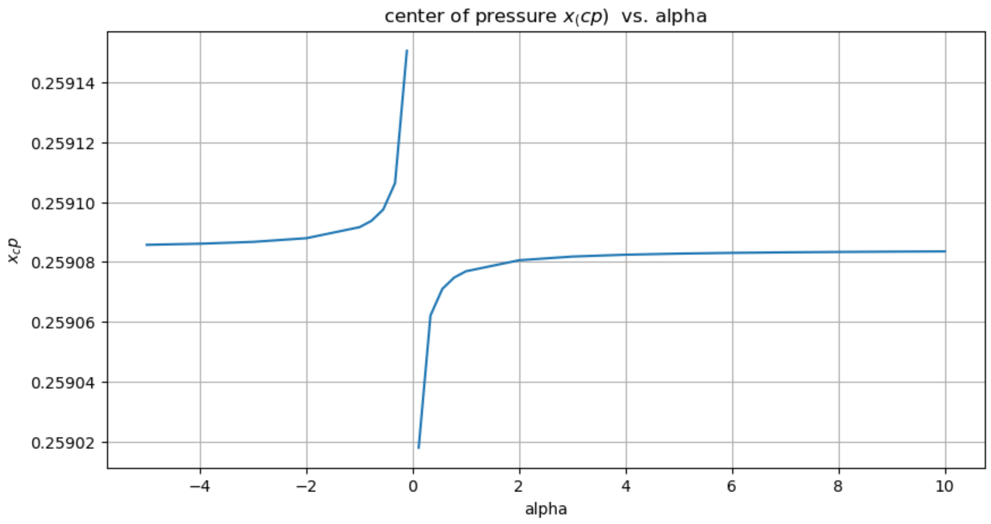

Figure 7: Variation of x_{Cp} with angle of attack

Torque Calculation

Torque about the center of pressure:

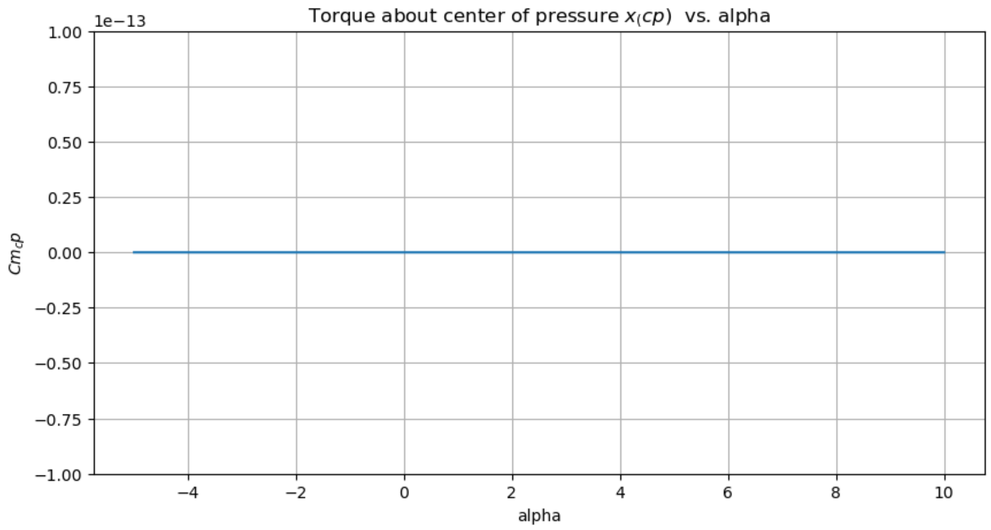

Figure 8: Torque about center of pressure vs angle of attack

]

4 Conclusion

The vortex panel method effectively models the flow over the NACA0012 airfoil. The pressure distribution and aerodynamic characteristics vary predictably with angle of attack.

References

- https://www.engapplets.vt.edu/fluids/vpm/naca0012.txt
- Arnold M. Kuethe ,Chuen-Yen Chow *Foundations of Aerodynamics*
- Anderson, J. D. (2010). *Fundamentals of Aerodynamics.*