



TECHNISCHE  
UNIVERSITÄT  
WIEN

## BACHELORARBEIT

# Implementation und Anwendung eines Hess-Smith Panelverfahrens in Python

ausgeführt am Institut für Strömungsmechanik und Wärmeübertragung  
der Technischen Universität Wien

unter der Anleitung von  
**Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Braun**

durch

**Leon Schwarzäugl**

Gumpendorfer Straße 137,  
1060 Wien

1. August 2022

---

Unterschrift StudentIn

# Zusammenfassung

Im Zuge der vorliegenden Arbeit wurde ein Panelverfahrens nach Hess-Smith zur Berechnung von ebenen Potentialströmungen um geschlossene Körper implementiert und auf verschiedene Probleme angewandt. Dazu wurde die Programmiersprache Python verwendet.

Zunächst wurde eine Klasse für Panele und ihre Eigenschaften entworfen. Im Zuge dessen wurde eine Routine entwickelt, die jene Parameter in eine .csv-Datei ausgibt. Danach wurde eine Klasse für Tragflächenprofile sowie Methoden zur Berechnung der ihnen zugehörigen Systemparameter implementiert. Mit den somit berechneten Systemparametern kann das sich ergebende lineare Gleichungssystem für die Quellstärken gelöst werden; im Anschluss wurden aus den ermittelten Quellstärken die Tangentialgeschwindigkeiten und Druckbeiwerte an den jeweiligen Panele, sowie unter Hinzunahme einer Wirbelbewegung der Auftriebsbeiwert für das Gesamtsystem ermittelt. Abschließend wurde eine Routine für die Ermittlung der Tangentialgeschwindigkeiten an beliebigen Punkten auf der Oberfläche des Profils entwickelt.

Die fertige Implementation wurde zunächst verwendet, um für die Panele verschiedener Tragflächenprofile Graphen zu Geometrien und Parametern zu gewinnen. Anschließend wurde anhand eines Kreiszylinders mit 8 Panele gleicher Seitenlänge der Vektor der Quellbelegung, sowie die Tangentialgeschwindigkeiten und Druckbeiwerte an den Panelmittelpunkten ermittelt sowie die Ergebnisse für den Druckbeiwert mit dem theoretischen Ergebnis aus der Potentialtheorie verglichen. Dabei wurde für die mittlere Abweichung in Abhängigkeit der Panelanzahl  $n$  die Abhängigkeit  $e_{c_{p_i}}(n) \propto 1/n^2$  ermittelt. Abschließend wurde das oben genannte System um eine Wirbelbelegung erweitert und der Auftriebsbeiwert unter verschiedenen Anstellwinkeln des Profils bei konstantem Angriffswinkel berechnet.

Letztlich wurde die Implementierung auf Joukowski-Profile angewandt und eine experimentelle Abschätzung der Fehlerordnung des Hess-Smith-Verfahrens durchgeführt. Für die relative Abweichung des Auftriebsbeiwerts des Hess-Smith Verfahrens vom theoretischen Resultat wurde 6.2% ermittelt.

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>1</b>
1.1	Zielsetzung . . . . .	1
1.2	Methodik . . . . .	1
<b>2</b>	<b>Grundlagen</b>	<b>2</b>
2.1	Hess-Smith-Panelverfahren . . . . .	2
2.1.1	Berücksichtigung einer Wirbelbelegung . . . . .	4
2.1.2	Berechnung der Systemmatrix . . . . .	5
<b>3</b>	<b>Bericht der Arbeit</b>	<b>8</b>
3.1	Vorbereitung der Daten . . . . .	8
3.2	Untersuchungen am Kreiszyylinder . . . . .	10
3.2.1	Achtseitiger Kreiszyylinder . . . . .	11
3.2.2	Untersuchungen an allgemeinen Kreiszyclindern . . . . .	12
3.3	Auswertung ausgewählter Profile . . . . .	15
3.3.1	NACA0012-Profil . . . . .	16
3.3.2	NASA NLF1015-Profil . . . . .	17
3.3.3	Taylor Zone-40-Profil . . . . .	18
3.4	Abschätzung der Fehlerordnung anhand des Joukowski-Profils . . . . .	19
	<b>Literatur</b>	<b>i</b>
	Anhang A: Lednicer- und Selig-Format . . . . .	ii
	Anhang B: Beispielausgabe der Methode <code>.write_panels()</code> . . . . .	iii
	Anhang C: Codeausschnitte . . . . .	iv

# 1 Aufgabenstellung

Die Berechnung der aerodynamischen Eigenschaften eines Tragflächenprofils.

## 1.1 Zielsetzung

Es sollen folgende Punkte implementiert werden:

- Eine Klasse für Panele und die sie definierenden Eigenschaften: Mittelpunkte auf der  $x$ - und  $y$ -Achse, Neigungswinkel  $\theta_i$  und Länge  $l_i$ . Jedes Panel enthält ebenfalls Möglichkeiten zur Speicherung der zugehörigen Quellstärken  $q_i$ , Tangentialgeschwindigkeiten  $v_i^{(t)}$  und Druckbeiwerten  $c_{p_i}$
- Eine Klasse für Trägerprofile, ihre Umfänge  $U$  und Tiefe  $t$  sowie mathematisch relevante Systemparameter  $\xi_{ij}$ ,  $\eta_{ij}$ ,  $I_{ij}$ ,  $J_{ij}$ ,  $A_{ij}^{(n)}$ ,  $A_{ij}^{(t)}$ ,  $M_{ij}$ .
- Eine Methode zur Lösung des linearen Gleichungssystems  $M\vec{q} = \vec{b}$  für den Vektor der Quellbewegung unter einer konstanten Anströmgeschwindigkeit  $V_\infty$  und einem Anstellwinkel des Profils  $\alpha$ , sowie der Berechnung der daraus resultierenden Tangentialgeschwindigkeiten und Druckbeiwerte.

Weiters soll mittels der Implementation eine Untersuchung der Abweichung der ermittelten Druckbeiwerte von den aus der Potentialtheorie berechneten Werten für einen achtseitigen Kreiszylinder erfolgen.

## 1.2 Methodik

Zur Lösung der Aufgabenstellung wurde die Programmiersprache Python verwendet. Für die Lösung der linearen Gleichungssysteme wird `linalg.solve`<sup>1</sup> aus der Python-Bibliothek NumPy verwendet.

Sämtliche Graphen wurden mithilfe der Python-Bibliothek Matplotlib erstellt. [7] Sofern analytische Lösungen zu den linearen Gleichungssystemen berechnet wurden, wurde dafür die Python-Bibliothek SciPy verwendet. [8]

---

<sup>1</sup>`linalg.solve` verwendet die LAPACK routine `_gesv` als Solver [5]

## 2 Grundlagen

### 2.1 Hess-Smith-Panelverfahren

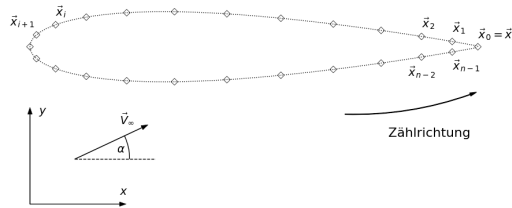


Abbildung 2.1: Zur Definition der Zählrichtung der Panels sowie der Anströmgeschwindigkeit und deren Winkel

Das Hess-Smith-Panelverfahren ist ein Verfahren zur Berechnung von ebenen Potentialströmungen um geschlossene Körper mit Auftrieb. Ein kontinuierlicher zweidimensionaler Körper mit der Umfangkurve  $\mathcal{C}$  wird dabei zunächst durch  $n+1$  diskrete Datenpunkte  $\vec{x}_i = (x_i, y_i)$  dargestellt. Die Zählrichtung dieser Punkte sei als im Uhrzeigersinn festgelegt (siehe Abb. 2.1). Für das Profil gilt somit die Periodizität  $\vec{x}_n = \vec{x}_0$ .

Die Kontur des Profils wird nun durch eine endliche Anzahl Verbindungsgeraden zwischen zwei benachbarten Punkten  $\vec{x}_i, \vec{x}_{i+1}$ , den Panels  $\mathcal{C}_i$  (Abb. 2.2) angenähert. Dabei wird jedes Panel durch die folgenden Parameter charakterisiert:

- Den Mittelpunkten auf beiden Achsen:

$$X_i = \frac{x_{n-i} + x_{n-i-1}}{2}, \quad Y_i = \frac{y_{n-i} + y_{n-i-1}}{2}, \quad (2.1)$$

- seinem Neigungswinkel:

$$\theta_i = \left( \frac{y_{n-i-1} - y_{n-i}}{x_{n-i-1} - x_{n-i}} \right), \quad (2.2)$$

- und seiner Länge:

$$l_i = \sqrt{(x_{n-i-1} - x_{n-i})^2 + (y_{n-i-1} - y_{n-i})^2}. \quad (2.3)$$

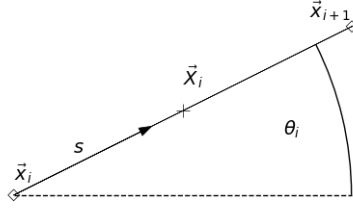


Abbildung 2.2: Zur Definition eines Panels

Im Hess-Smith-Panelverfahren sei nun jedes dieser Panels mit einer vorerst unbestimmten, konstanten Quelldichte  $q_i$  behaftet, welche an einem Punkt  $\vec{r}_i = \vec{x} - \hat{x}_i(s)$  zum Geschwindigkeitspotential den Beitrag

$$\phi_i^{(q)} = \frac{q_i}{2\pi} \int_{C_j} ds \ln r_i \quad (2.4)$$

liefert.  $s$  ist dabei die Koordinate der Bogenlänge, welche das gesamte Profil parametrisiert.

Ebenso liefert die Profilanströmung mit der konstanten Geschwindigkeit  $V_\infty$  unter dem Winkel  $\alpha$  ein Potential

$$\phi_\infty = V_\infty (\cos \alpha x + \sin \alpha y). \quad (2.5)$$

Aus der Summe über alle Panelbeiträge aus (2.4) und (2.5) ergibt sich das Gesamtgeschwindigkeitspotential zu

$$\phi(\vec{x}) = V_\infty (\cos \alpha x + \sin \alpha y) + \sum_{i=0}^{n-1} \phi_i^{(q)}. \quad (2.6)$$

Die Geschwindigkeit im Punkte  $\vec{x}$  ergibt sich zu

$$\vec{v}(\vec{x}) = \nabla \phi(\vec{x}). \quad (2.7)$$

Auf der Oberfläche des Profils muss die Strömung die Gleitbedingung erfüllen; damit lassen sich  $n$  Gleichungen aufstellen, um diese Bedingung zu modellieren. Es wird gefordert, dass die Normalkomponente des Geschwindigkeitsvektors,

$$v_j^{(n)} = \vec{v}(\vec{X}_j) \cdot \vec{n}_j, \quad (2.8)$$

verschwindet. Dabei ist  $\vec{n}_j$  der Normalvektor auf das Panel  $\mathcal{C}_j$ .  
In 0. Ordnung erhält man ein System von  $n$  Gleichungen

$$\sum_{j=0}^{n-1} M_{ij}^{(q)} q_j = b_i, \quad (2.9)$$

mit

$$M_{ij}^{(q)} = \frac{1}{q_j} \nabla \phi_j^{(q)}(\vec{X}) \cdot \vec{n}_i, \quad (2.10)$$

$$b_i = -V_\infty \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \cdot \vec{n}_i, \quad (2.11)$$

aus dessen Lösung die Quellstärken  $q_i$  für jedes Panel ermittelt werden können.  
Damit können die Tangentialkomponenten des Geschwindigkeitsvektors

$$v_j^{(t)} = \vec{v}(X_j) \cdot \frac{\vec{x}_{j+1} - \vec{x}_j}{|\vec{x}_{j+1} - \vec{x}_j|}, \quad (2.12)$$

bestimmt werden.

Aus der Bernoulli-Gleichung

$$p_\infty + \frac{1}{2} \rho V_\infty^2 = p + \frac{1}{2} \rho v^2, \quad (2.13)$$

definiert man den Druckbeiwert als das Verhältnis der Druckdifferenzen von Profilanströmung und dynamischem Druck

$$c_p = \frac{p - p_\infty}{\frac{1}{2} \rho V_\infty^2}. \quad (2.14)$$

Daraus kann der Druckbeiwert

$$c_{p_j} = 1 - \left( \frac{v_j^{(t)}}{V_\infty} \right)^2 \quad (2.15)$$

am Mittelpunkt des Panels  $\mathcal{C}_j$  bestimmt werden. [6] [4]

### 2.1.1 Berücksichtigung einer Wirbelbelegung

Will man den Auftrieb berücksichtigen, so reicht die reine Quellbelegung des Profils noch nicht aus.

Im Hess-Smith-Verfahren wird für das gesamte Profil eine einzige konstante Wirbelbelegung  $\gamma$  für alle Panels angenommen. Damit wird das Potential aus (2.6) erweitert um Beiträge

$$\phi_i^{(w)} = \frac{\gamma}{2\pi} \int ds \theta_i, \quad (2.16)$$

zu

$$\begin{aligned}\phi(\vec{x}) &= V_\infty(\cos \alpha x + \sin \alpha y) + \sum_{i=0}^{n-1} \left( \phi_i^{(q)} + \phi_i^{(w)} \right) \\ &= V_\infty(\cos \alpha x + \sin \alpha y) + \sum_{i=0}^{n-1} \int_{\mathcal{C}_i} \left( \frac{q_i}{2\pi} \ln r_i - \frac{\gamma}{2\pi} \theta_i \right) ds.\end{aligned}\quad (2.17)$$

Die Gleitbedingung bleibt erhalten, allerdings muss das Gleichungssystem entsprechend um den hinzugekommenen Geschwindigkeitsbeitrag erweitert werden. Für die neue Unbekannte  $\gamma$  fehlt also eine Gleichung. Diese ergibt sich aus der Kutta-Bedingung, welche besagt, dass es an der Hinterkante des Profils keine Umströmung gibt.

Zur Modellierung der Kutta-Bedingung wählt man

$$v_0^{(t)} = -v_n^{(t)}.\quad (2.18)$$

Daraus folgt, dass die Tangentialkomponenten des Geschwindigkeitsvektors in den Mittelpunkten der Panels, welche die Profilhinterkante bilden, vom Betrag gleich groß und gegenläufig gerichtet sein muss (siehe Abb. 2.3).

Nach Lösung des erhaltenen linearen Gleichungssystems mit  $q_n = \gamma$  kann nun der Auftriebsbeiwert  $c_a$  ermittelt werden. Dieser ergibt sich mit  $t$ , der Profiltiefe und  $l_i$ , der Länge des Panels  $\mathcal{C}_i$ , approximiert in folgender Form:

$$c_a \approx \frac{2}{V_\infty t} \sum_{i=0}^{n-1} v_i^{(t)} l_i.\quad (2.19)$$

Für eine analytische Berechnung des Auftriebsbeiwerts muss nun noch die Systemmatrix bestimmt werden, welche das lineare Gleichungssystem (2.9) definiert. [6] [4] [2]

## 2.1.2 Berechnung der Systemmatrix

Die Systemmatrix zur Berechnung der Normalgeschwindigkeiten  $v_i^{(n)}$  in den Panelmittelpunkten für eine stückweise konstante Quellenbelegung der Panels ergibt sich aus der Gleitbedingung. Wendet man diese direkt auf 2.6 an, muss ein kompliziertes Integral gelöst werden. Dieses kann theoretisch direkt mit Computermethoden gelöst werden, für den Berechnungsaufwand ist es allerdings gerade bei Profilen mit einer hohen Anzahl an Panels günstiger, dieses analytisch zu berechnen.

Für die Systemmatrix und ihre Parameter gelten:

$$M_{ij} = A_{ij}^{(n)}, \quad i = 0, 1, \dots, n-1; \quad j = 0, 1, \dots, n-1.\quad (2.20)$$



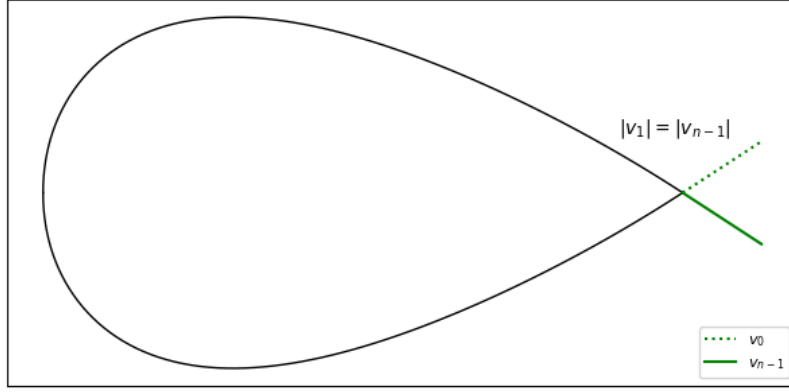


Abbildung 2.3: Zur Kutta-Bedingung am Beispiel eines NACA0012-Profiles - am Punkt der Hinterkante sind die Tangentialgeschwindigkeiten genau entgegengerichtet

mit der Einflussmatrix der Quellbelegung

$$A_{ij}^{(n)} = -\sin(\theta_i - \theta_j)I_{ij} + \cos(\theta_i - \theta_j)J_{ij} \quad (2.21)$$

sowie dem geometrischen Integral der Normalgeschwindigkeit

$$I_{ij} = \begin{cases} \frac{1}{4\pi} \ln \left[ \frac{(l_j + 2\xi_{ij})^2 + 4\eta_{ij}^2}{(l_j - 2\xi_{ij})^2 + 4\eta_{ij}^2} \right] & i \neq j \\ 0 & i = j \end{cases} \quad (2.22)$$

und dem geometrischen Integral der Tangentialgeschwindigkeit

$$J_{ij} = \begin{cases} \frac{1}{2\pi} \arctan \left[ \frac{l_j - 2\xi_{ij}}{2\eta_{ij}} \right] + \frac{1}{2\pi} \arctan \left[ \frac{l_j + 2\xi_{ij}}{2\eta_{ij}} \right] & i \neq j \\ \frac{1}{2} & i = j \end{cases} \quad (2.23)$$

$$\xi_{ij} = (X_i - X_j) \cos \theta_j + (Y_i - Y_j) \sin \theta_j \quad (2.24)$$

$$\eta_{ij} = -(X_i - X_j) \sin \theta_j + (Y_i - Y_j) \cos \theta_j \quad (2.25)$$

Mit der Systemmatrix und den Inhomogenitäten  $b_i$  wird nun das lineare Gleichungssystem aus (2.9) berechnet.

Für die Tangentialkomponente der Geschwindigkeit ergibt sich der Ausdruck

$$v_i^{(t)} = \sum_{j=0}^{n-1} A_{ij}^{(t)} q_j + V_\infty \cos(\alpha - \theta_i), \quad (2.26)$$

mit der Einflussmatrix der Wirbelbelegung

$$A_{ij}^{(t)} = \cos(\theta_i - \theta_j)I_{ij} + \sin(\theta_i - \theta_j)J_{ij}. \quad (2.27)$$

Die Systemmatrix  $M_{i,j}$  aus (2.20) berücksichtigt allerdings noch keine Wirbelbelegungen. Im Falle einer konstanten Wirbelbelegung  $\gamma$  auf allen Panels erweitert sich die Dimension von  $M_{ij}$  von  $n \times n$  auf  $(n+1) \times (n+1)$ . Dadurch müssen folgende zusätzliche Elemente bestimmt werden:

$$M_{in} = \sum_{j=0}^{n-1} A_{ij}^{(t)}, \quad i = 0, 1, \dots, n-1; \quad (2.28)$$

$$M_{nj} = A_{0j}^{(t)} + A_{n-1,j}^{(t)}, \quad j = 0, 1, \dots, n-1; \quad (2.29)$$

$$M_{n,n} = -\sum_{j=0}^{n-1} \left[ A_{0,j}^{(n)} + A_{n-1,j}^{(n)} \right]; \quad (2.30)$$

$$b_n = -V_\infty [\cos(\alpha - \theta_0) + \cos(\alpha - \theta_{n-1})]; \quad (2.31)$$

Gleichung (2.26) erweitert sich damit zu

$$v_i^{(t)} = \sum_{j=0}^{n-1} A_{ij}^{(t)} q_j - \gamma \sum_{j=0}^{n-1} A_{i,j}^{(n)} + V_\infty \cos(\alpha - \theta_i). \quad (2.32)$$

[6] [4]

## 3 Bericht der Arbeit

Im Folgenden werden die wesentlichen Erkenntnisse der Arbeit präsentiert.

### 3.1 Vorbereitung der Daten

Die Daten der Profile wurden der UIUC Airfoil Data Site<sup>1</sup> entnommen. Die Daten lagen dabei überwiegend im Selig- oder Lednicer-Format vor.<sup>2</sup>

Im Lednicer-Format wird in der ersten Zeile die Bezeichnung des Profils angegeben. In der zweiten Zeile wird die Anzahl der Koordinatenpaare der Ober- und Unterseite angegeben. Ab der dritten Zeile werden die Koordinaten der Panelenden  $(x_i, y_i)$  von  $x = 0$  bis  $x = 1$  für die Oberseite des Profils angegeben. Danach folgt eine Leerzeile. Danach werden die Koordinaten der Panelenden  $(x_i, y_i)$  von  $x = 0$  bis  $x = 1$  für die Unterseite des Profils angegeben.

Im Selig-Format wird in der ersten Zeile die Bezeichnung des Profils angegeben. Ab der zweiten Zeile werden die Koordinaten der Panelenden  $(x_i, y_i)$  im Gegenurzeigersinn, beginnend bei  $x = 1$  angegeben.

Es wurde eine Routine geschrieben, welche Daten im Lednicer-Format in das Selig-Format überführt. Dies wurde bewerkstelligt, indem die Koordinatenpaare der Oberseite invertiert wurden. Anschließend wurden aus den Daten die Header-Zeilen entfernt.

Unter Verwendung der Methode `.write_panels()` wurde für sämtliche Profile eine .csv-Datei mit den Werten  $X_i, Y_i, \theta_i, l_i$  pro Panel erstellt (eine Beispieldatei ist in Anhang A: Lednicer- und Selig-Format gezeigt). Ebenso wurde für jedes sich in der UIUC-Datenbank befindliche Trägerprofil ein Graph erzeugt, welcher zum einen die Geometrie des Profils, sowie auch die Neigungswinkel und Längen der einzelnen Panels darstellt. Zur einfacheren Betrachtung sind die Graphen der Neigungswinkel und Panellängen in die beiden Seiten des Profils aufgeteilt.

Ein solcher Graph ist in Abbildung 3.1 an einem NACA-0012-Profil gezeigt. Die Symmetrie des Profils spiegelt sich in der gewählten Darstellung der Neigungswinkel und Panellängen wieder.

---

<sup>1</sup>[https://m-selig.ae.illinois.edu/ads/coord\\_database.html](https://m-selig.ae.illinois.edu/ads/coord_database.html)

<sup>2</sup>Eine Gegenüberstellung der beiden Formate ist in Anhang A: Lednicer- und Selig-Format zu finden.

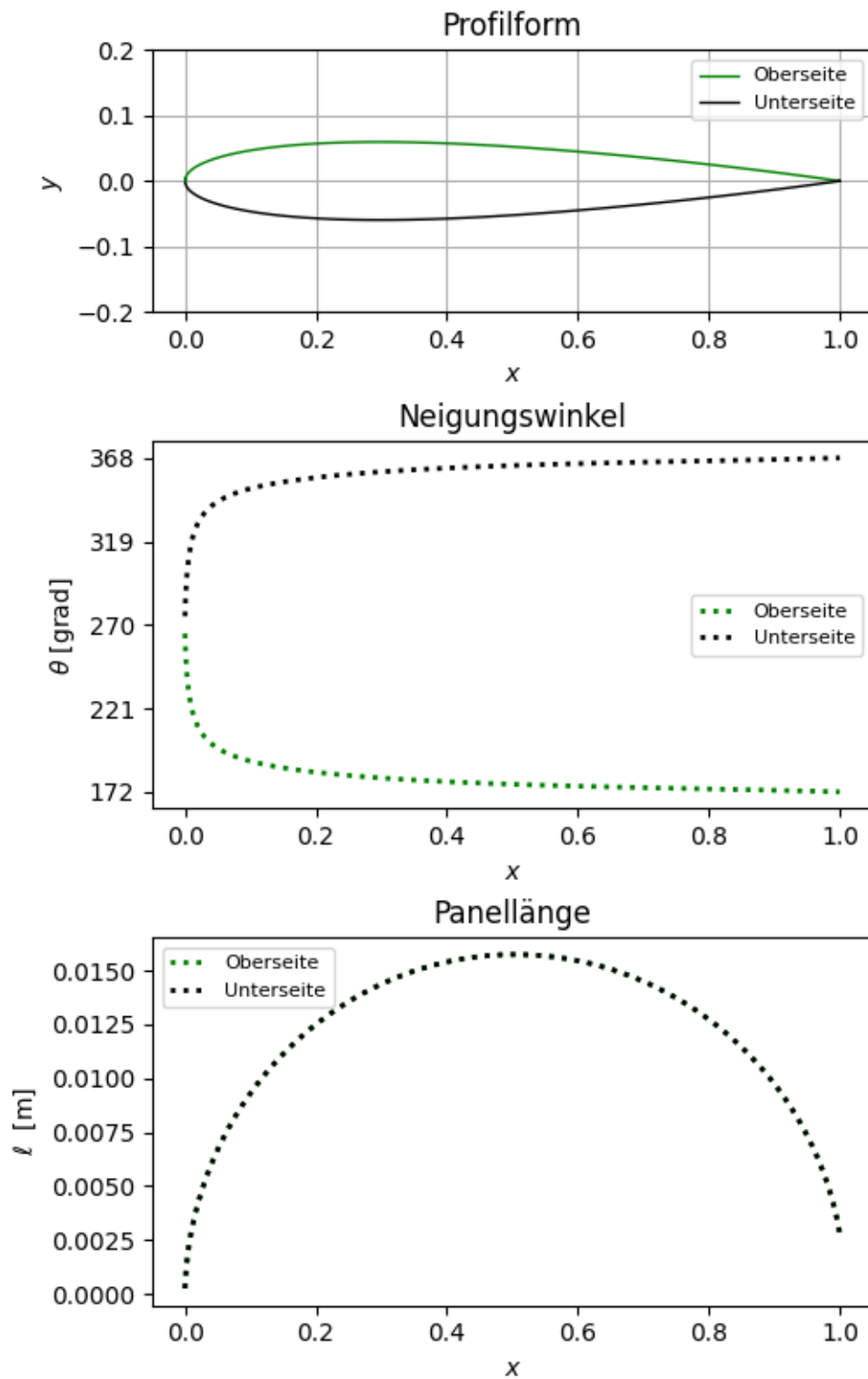


Abbildung 3.1: NACA-0012-Profil

$X_i$	$Y_i$	$\theta_i$	$l_i$
0.93	0.18	112.5	0.38
0.68	0.43	157.5	0.38
0.32	0.43	202.5	0.38
0.07	0.18	247.5	0.38
0.07	-0.18	292.5	0.38
0.32	-0.43	337.5	0.38
0.68	-0.43	22.5	0.38
0.93	-0.18	67.5	0.38

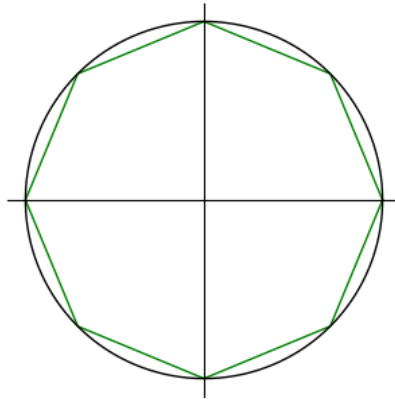


Abbildung 3.2: Achtseitiger Zylinder als Approximation eines Kreiszylinders sowie zugehörige Panelparameter

## 3.2 Untersuchungen am Kreiszylinder

Als nächstes wurden Untersuchungen an Kreiszylindern vorgenommen. Zur Generierung eines Kreiszylinders mit  $n$  gleich langen Seiten wurde die Funktion `make_cylinder` verwendet (siehe Anhang C: Codeausschnitte).

### 3.2.1 Achtseitiger Kreiszylinder

#### Berechnung der Systemmatrix

Für den in Abbildung 3.2 gezeigten Kreiszylinder wurde die Systemmatrix, gerundet auf zwei Dezimalstellen, zu folgenden Werten berechnet:

$$M_{i,j} = \begin{pmatrix} 0.5 & 0.056 & 0.064 & 0.065 & 0.065 & 0.065 & 0.064 & 0.056 \\ 0.056 & 0.5 & 0.056 & 0.064 & 0.065 & 0.065 & 0.065 & 0.064 \\ 0.064 & 0.056 & 0.5 & 0.056 & 0.064 & 0.065 & 0.065 & 0.065 \\ 0.065 & 0.064 & 0.056 & 0.5 & 0.056 & 0.064 & 0.065 & 0.065 \\ 0.065 & 0.065 & 0.064 & 0.056 & 0.5 & 0.056 & 0.064 & 0.065 \\ 0.065 & 0.065 & 0.065 & 0.064 & 0.056 & 0.5 & 0.056 & 0.064 \\ 0.064 & 0.065 & 0.065 & 0.065 & 0.064 & 0.056 & 0.5 & 0.056 \\ 0.056 & 0.064 & 0.065 & 0.065 & 0.065 & 0.064 & 0.056 & 0.5 \end{pmatrix}$$

#### Lösung des wirbellosen Gleichungssystems

Im Folgenden wurde die Lösung des linearen Gleichungssystems mit der obigen Systemmatrix und den Inhomogenitäten

$$b_i = -V_\infty \sin(\alpha - \theta_i), \quad i = 0, 1, \dots, n-1 \quad (3.1)$$

berechnet. Daraus wurde der Vektor der Quellenbelegung zunächst mit  $V_\infty$  und  $\alpha$  als Variablen berechnet:

$$\vec{q} = \begin{pmatrix} -0.16V_\infty \sin(\alpha - 5.89) - 0.17V_\infty \sin(\alpha - 5.11) - 0.17V_\infty \sin(\alpha - 4.32) - 0.17V_\infty \sin(\alpha - 3.53) - 0.16V_\infty \sin(\alpha - 2.75) - 0.12V_\infty \sin(\alpha - 1.96) + 2.13V_\infty \sin(\alpha - 1.18) - 0.12V_\infty \sin(\alpha - 0.39) \\ -0.12V_\infty \sin(\alpha - 5.89) - 0.16V_\infty \sin(\alpha - 5.11) - 0.17V_\infty \sin(\alpha - 4.32) - 0.17V_\infty \sin(\alpha - 3.53) - 0.17V_\infty \sin(\alpha - 2.75) - 0.16V_\infty \sin(\alpha - 1.96) - 0.12V_\infty \sin(\alpha - 1.18) + 2.13V_\infty \sin(\alpha - 0.39) \\ 2.13V_\infty \sin(\alpha - 5.89) - 0.12V_\infty \sin(\alpha - 5.11) - 0.16V_\infty \sin(\alpha - 4.32) - 0.17V_\infty \sin(\alpha - 3.53) - 0.17V_\infty \sin(\alpha - 2.75) - 0.17V_\infty \sin(\alpha - 1.96) - 0.16V_\infty \sin(\alpha - 1.18) - 0.12V_\infty \sin(\alpha - 0.39) \\ -0.12V_\infty \sin(\alpha - 5.89) + 2.13V_\infty \sin(\alpha - 5.11) - 0.12V_\infty \sin(\alpha - 4.32) - 0.16V_\infty \sin(\alpha - 3.53) - 0.17V_\infty \sin(\alpha - 2.75) - 0.17V_\infty \sin(\alpha - 1.96) - 0.17V_\infty \sin(\alpha - 1.18) - 0.16V_\infty \sin(\alpha - 0.39) \\ -0.16V_\infty \sin(\alpha - 5.89) - 0.12V_\infty \sin(\alpha - 5.11) + 2.13V_\infty \sin(\alpha - 4.32) - 0.12V_\infty \sin(\alpha - 3.53) - 0.16V_\infty \sin(\alpha - 2.75) - 0.17V_\infty \sin(\alpha - 1.96) - 0.17V_\infty \sin(\alpha - 1.18) - 0.17V_\infty \sin(\alpha - 0.39) \\ -0.17V_\infty \sin(\alpha - 5.89) - 0.16V_\infty \sin(\alpha - 5.11) - 0.12V_\infty \sin(\alpha - 4.32) + 2.13V_\infty \sin(\alpha - 3.53) - 0.12V_\infty \sin(\alpha - 2.75) - 0.16V_\infty \sin(\alpha - 1.96) - 0.17V_\infty \sin(\alpha - 1.18) - 0.17V_\infty \sin(\alpha - 0.39) \\ -0.17V_\infty \sin(\alpha - 5.89) - 0.17V_\infty \sin(\alpha - 5.11) - 0.16V_\infty \sin(\alpha - 4.32) - 0.12V_\infty \sin(\alpha - 3.53) + 2.13V_\infty \sin(\alpha - 2.75) - 0.12V_\infty \sin(\alpha - 1.96) - 0.16V_\infty \sin(\alpha - 1.18) - 0.17V_\infty \sin(\alpha - 0.39) \\ -0.17V_\infty \sin(\alpha - 5.89) - 0.17V_\infty \sin(\alpha - 5.11) - 0.17V_\infty \sin(\alpha - 4.32) - 0.16V_\infty \sin(\alpha - 3.53) - 0.12V_\infty \sin(\alpha - 2.75) + 2.13V_\infty \sin(\alpha - 1.96) - 0.12V_\infty \sin(\alpha - 1.18) - 0.16V_\infty \sin(\alpha - 0.39) \end{pmatrix}.$$

Nach Einsetzen der Werte  $V_\infty = 1$  und  $\alpha = 0$  vereinfacht sich der Vektor zu einem leichter zu handhabenden Wert. Die Ergebnisse für Quellenbelegung, Tangentialgeschwindigkeiten und Druckbeiwerte sind in Tabelle 3.1 zusammengetragen. Ebenfalls von Interesse ist die Größe  $\sum_{i=0}^{n-1} q_i l_i$ . Diese Summe sollte für einen geschlossenen Körper null ergeben, da der Körper sonst durch den Fluss seine Masse verändern würde (vgl. [3]). Tatsächlich ergibt sich:

$$\sum_{i=0}^{n-1} q_i l_i \approx 4.44 \cdot 10^{-16}$$

Tabelle 3.1: Ergebnisse für den achtseitigen Zylinder mit  $V_\infty = 1, \alpha = 0$

$i$	$q_i$	$v_i^{(t)}$	$c_{p_i}$
0	-2.19	0.77	0.41
1	-0.91	1.85	-2.41
2	0.91	1.85	-2.41
3	2.19	0.77	0.41
4	2.19	-0.77	0.41
5	0.91	-1.85	-2.41
6	-0.91	-1.85	-2.41
7	-2.19	-0.765	0.41

Dieser sehr kleine Fehler ergibt sich zum einen aus Fehlern, die bei der Berechnung von  $\vec{q}$  entstehen, und zum anderen aus Rundungsfehlern bei der Berechnung der Eckpunkte des Kreiszyinders.

### 3.2.2 Untersuchungen an allgemeinen Kreiszyindern

Die Potentialtheorie liefert ein exaktes Resultat für den Druckbeiwert auf der Oberfläche eines Zylinders:[4]

$$c_p^{\text{exakt}}(\varphi) = 2 \cos(2\varphi) - 1 \quad (3.2)$$

Dabei bezeichnet  $\varphi$  den Winkel der Flächennormale des Kreiszyinders welche ins Innere des Profils zeigt. Mit unserer Definition des Neigungswinkels  $\theta$  gilt

$$\varphi = \theta + \frac{\pi}{2}. \quad (3.3)$$

Abbildung 3.3a zeigt die Abweichung der Druckbeiwerte eines achtseitigen Zylinders vom theoretischen Resultat. Es ergibt sich eine Abweichung dadurch, dass die Mittelpunkte des achtseitigen Zylinders nicht auf der Kreisfläche liegen. Der Fehler ergibt sich zu:

$$c_p^{\text{exakt}}(\varphi) = 2 \cos(2\varphi) - 1 \quad (3.4)$$

$$= 1 - 4 \sin^2 \varphi \quad (3.5)$$

$$= 1 - 4 \left( \frac{Y}{R} \right)^2 \quad (3.6)$$

Der Fehler, der bei der Approximation entsteht, ist also

$$\Delta c_{p_i} = |c_{p_i} - c_p^{\text{exakt}}(\varphi)| \quad (3.7)$$

$$= |c_{p_i} - [1 - 4(Y_i/R)^2]| \quad (3.8)$$

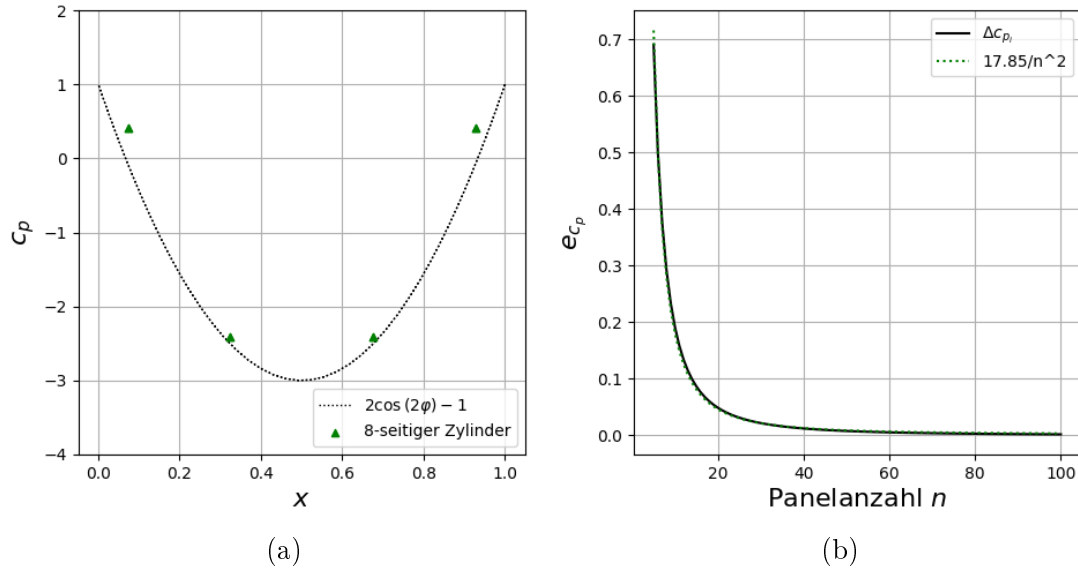


Abbildung 3.3: a) Abweichung der Druckbeiwerte eines achtseitigen Zylinders vom theoretisch ermittelten Wert b) Graph der Abweichung des mittleren Fehlers vom exakten Wert.

### Variation der Panelanzahl

Das Programm wurde nun für Kreiszylinder zwischen 5 und 100 Panels durchlaufen und die mittlere absolute Abweichung  $e_{c_p}$  für jeden Kreiszylinder festgehalten (Abbildung 3.3b). Aus den experimentellen Daten wurde mittels Kurvenanpassung für den absoluten Fehler in Abhängigkeit der Panelanzahl  $n$  die Beziehung

$$e_{c_p}(n) \approx \frac{17.85301}{n} + 0.00183 \quad (3.9)$$

ermittelt.

### Rotation des achtseitigen Zylinders

Als nächstes wurden die Werte  $\gamma$ ,  $c_a$ , sowie die Druckbeiwerte für verschiedene Winkel im Bereich  $\alpha \in [-15^\circ, 15^\circ]$  bestimmt. Die Ergebnisse sind in Tabelle 3.2 aufgetragen.



Tabelle 3.2: Ergebnisse für den achtseitigen Zylinder mit  $V_\infty = 1, \alpha \in [-15^\circ, 15^\circ]$

$\alpha$	$\gamma$	$c_a$	$c_{p0}$	$c_{p1}$	$c_{p2}$	$c_{p3}$	$c_{p4}$	$c_{p5}$	$c_{p6}$	$c_{p7}$
-15	-0.51	-2.93	0.45	-1.26	-0.23	0.95	-1.88	-5.06	-3.26	0.45
-14	-0.48	-2.74	0.45	-1.34	-0.35	0.98	-1.68	-4.88	-3.22	0.45
-13	-0.44	-2.55	0.44	-1.42	-0.47	0.99	-1.49	-4.7	-3.18	0.44
-12	-0.41	-2.35	0.44	-1.5	-0.6	1.0	-1.3	-4.53	-3.13	0.44
-11	-0.38	-2.16	0.44	-1.58	-0.73	1.0	-1.12	-4.35	-3.08	0.44
-10	-0.34	-1.96	0.43	-1.66	-0.87	0.99	-0.95	-4.17	-3.03	0.43
-9	-0.31	-1.77	0.43	-1.74	-1.01	0.97	-0.78	-3.99	-2.98	0.43
-8	-0.28	-1.57	0.43	-1.82	-1.15	0.94	-0.62	-3.81	-2.92	0.43
-7	-0.24	-1.38	0.42	-1.9	-1.3	0.9	-0.46	-3.63	-2.86	0.42
-6	-0.21	-1.18	0.42	-1.97	-1.45	0.86	-0.32	-3.46	-2.81	0.42
-5	-0.17	-0.99	0.42	-2.05	-1.6	0.81	-0.18	-3.28	-2.74	0.42
-4	-0.14	-0.79	0.42	-2.12	-1.76	0.74	-0.04	-3.1	-2.68	0.42
-3	-0.1	-0.59	0.42	-2.2	-1.92	0.67	0.08	-2.93	-2.62	0.42
-2	-0.07	-0.39	0.41	-2.27	-2.08	0.6	0.2	-2.76	-2.55	0.41
-1	-0.03	-0.2	0.41	-2.34	-2.25	0.51	0.31	-2.58	-2.48	0.41
0	0.0	0.0	0.41	-2.41	-2.41	0.41	0.41	-2.41	-2.41	0.41
1	0.03	0.2	0.41	-2.48	-2.58	0.31	0.51	-2.25	-2.34	0.41
2	0.07	0.39	0.41	-2.55	-2.76	0.2	0.6	-2.08	-2.27	0.41
3	0.1	0.59	0.42	-2.62	-2.93	0.08	0.67	-1.92	-2.2	0.42
4	0.14	0.79	0.42	-2.68	-3.1	-0.04	0.74	-1.76	-2.12	0.42
5	0.17	0.99	0.42	-2.74	-3.28	-0.18	0.81	-1.6	-2.05	0.42
6	0.21	1.18	0.42	-2.81	-3.46	-0.32	0.86	-1.45	-1.97	0.42
7	0.24	1.38	0.42	-2.86	-3.63	-0.46	0.9	-1.3	-1.9	0.42
8	0.28	1.57	0.43	-2.92	-3.81	-0.62	0.94	-1.15	-1.82	0.43
9	0.31	1.77	0.43	-2.98	-3.99	-0.78	0.97	-1.01	-1.74	0.43
10	0.34	1.96	0.43	-3.03	-4.17	-0.95	0.99	-0.87	-1.66	0.43
11	0.38	2.16	0.44	-3.08	-4.35	-1.12	1.0	-0.73	-1.58	0.44
12	0.41	2.35	0.44	-3.13	-4.53	-1.3	1.0	-0.6	-1.5	0.44
13	0.44	2.55	0.44	-3.18	-4.7	-1.49	0.99	-0.47	-1.42	0.44
14	0.48	2.74	0.45	-3.22	-4.88	-1.68	0.98	-0.35	-1.34	0.45
15	0.51	2.93	0.45	-3.26	-5.06	-1.88	0.95	-0.23	-1.26	0.45

### 3.3 Auswertung ausgewählter Profile

Im Folgenden wurde das entwickelte Programm auf die drei Profile NACA0012, NASA NLF1015 und Taylor Zone-40 angewandt. Für jedes Profil wurden die Werte  $\gamma$  und  $c_a$  unter verschiedenen Anströmwindeln  $\alpha$  berechnet.

An dieser Stelle wurde das Programm um die Möglichkeit erweitern, den Geschwindigkeitsvektor an jedem Punkt des Profils zu berechnen. Dies spiegelt sich in (2.24) und (2.25) wieder, für welche dann

$$\xi_{ij} = (x - X_j) \cos \theta_j + (y - Y_j) \sin \theta_j \quad (3.10)$$

und

$$\eta_{ij} = -(x - X_j) \sin \theta_j + (y - Y_j) \cos \theta_j \quad (3.11)$$

gilt. (2.21) und (2.27) vereinfachen sich zu

$$A_{ij}^{(n)} = \sin(\theta_j) I_{ij} + \cos(\theta_j) J_{ij}, \quad (3.12)$$

$$A_{ij}^{(t)} = \cos(\theta_j) I_{ij} - \sin(\theta_j) J_{ij}. \quad (3.13)$$

Der Geschwindigkeitsvektor ergibt sich also zu:

$$\vec{v}(x, y) = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \sum_{j=0}^{n-1} A_{ij}^{(t)} q_j - \gamma \sum_{j=0}^{n-1} A_{ij}^{(n)} + V_\infty \cos(\alpha) \\ \sum_{j=0}^{n-1} A_{ij}^{(n)} q_j + \gamma \sum_{j=0}^{n-1} A_{i,j}^{(t)} + V_\infty \cos(\alpha) \end{pmatrix} \quad (3.14)$$

Diese Geschwindigkeitsvektoren wurden für die oben genannten Profile für jeden Punkt berechnet sind in Form einer Colormap, aufgeteilt auf die x- und y-Komponenten, dargestellt. Für den Anströmwindel wurde dabei  $\alpha = 5^\circ$  gewählt. Ebenfalls in Form einer Colormap wurden für die Panele die Werte für Quellbelegung und Druckbeiwert ermittelt und dargestellt. Erneut wurde für den Anströmwindel  $\alpha = 5^\circ$  gewählt.

### 3.3.1 NACA0012-Profil

$\alpha$	$\gamma$	$c_a$
-11	-0.32	-1.32
-7	-0.21	-0.84
-3	-0.09	-0.36
1	0.03	0.12
5	0.15	0.6
9	0.27	1.08
13	0.38	1.55
17	0.5	2.02

Tabelle 3.3: Ergebnisse für das NACA0012-Profil

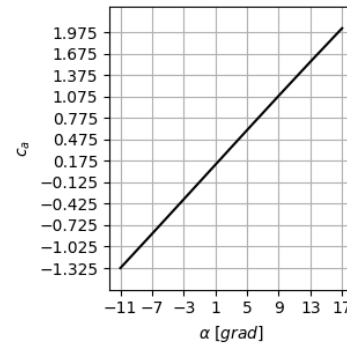
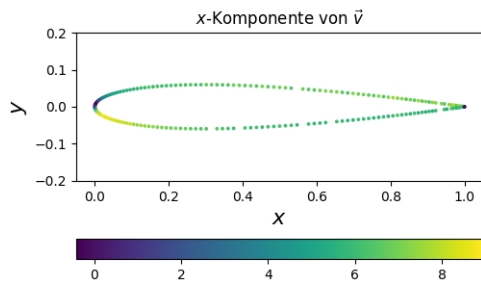
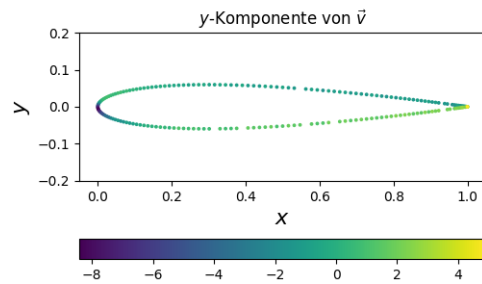


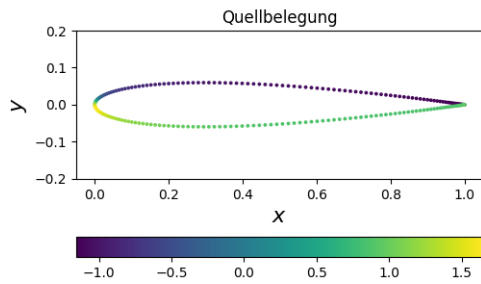
Abbildung 3.4: Graph des Auftriebsbeiwerts.



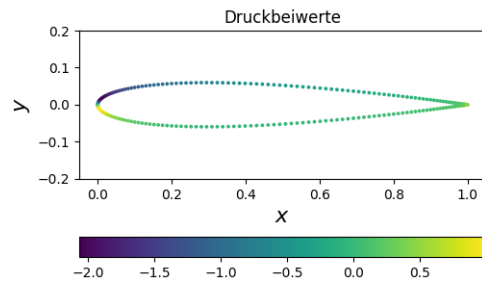
(a)



(b)



(c)



(d)

Abbildung 3.5: a) x- und b) y-Komponente des Geschwindigkeitsvektors. c Quellbelegung und d Druckbeiwerte pro Panel  $\mathcal{C}_i$  unter dem Anströmwinkel  $\alpha = 5^\circ$

### 3.3.2 NASA NLF1015-Profil

$\alpha$	$\gamma$	$c_a$
-11	-0.1	-0.43
-7	0.01	0.03
-3	0.13	0.5
1	0.24	0.96
5	0.35	1.41
9	0.47	1.86
13	0.57	2.3
17	0.68	2.73

Tabelle 3.4: Ergebnisse für das NASA NLF1015-Profil

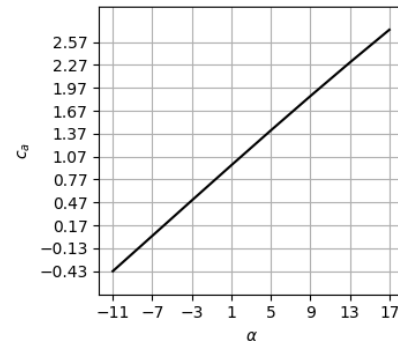
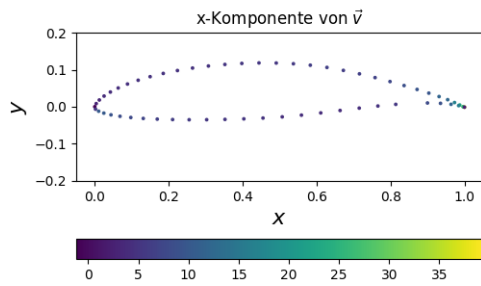
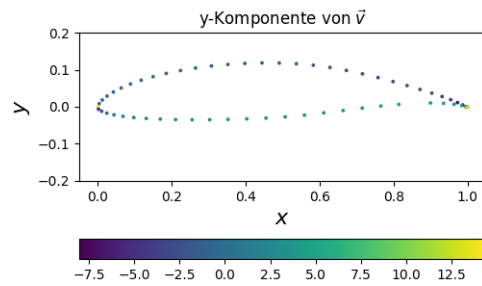


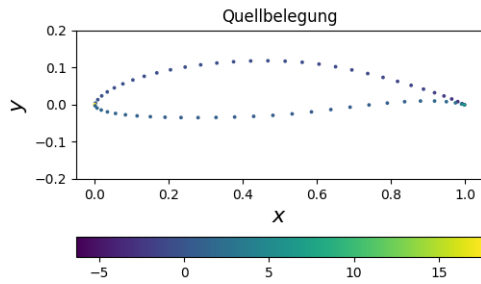
Abbildung 3.6: Graph des Auftriebsbeiwerts.



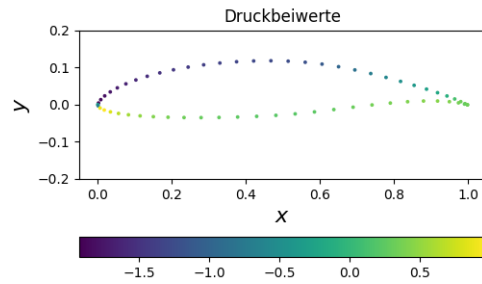
(a)



(b)



(c)



(d)

Abbildung 3.7: a) x- und b) y-Komponente des Geschwindigkeitsvektors. c) Quellbelegung und d) Druckbeiwerte pro Panel  $C_i$  unter dem Anströmwinkel  $\alpha = 5^\circ$

### 3.3.3 Taylor Zone-40-Profil

$\alpha$	$\gamma$	$c_a$
-11	-0.27	-1.07
-7	-0.16	-0.63
-3	-0.04	-0.18
1	0.07	0.27
5	0.18	0.71
9	0.29	1.16
13	0.4	1.59
17	0.51	2.02

Tabelle 3.5: Ergebnisse für das Taylor Zone-40-Profil

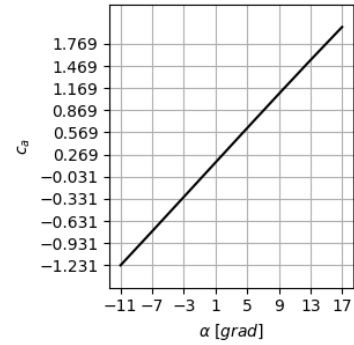
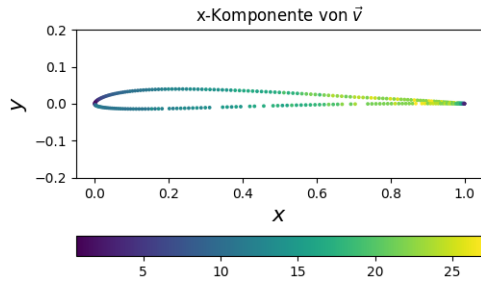
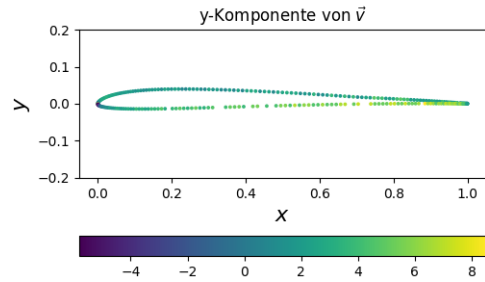


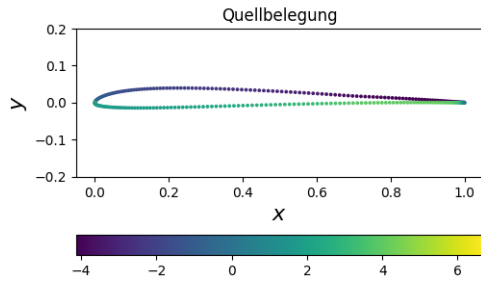
Abbildung 3.8: Graph des Auftriebsbeiwerts.



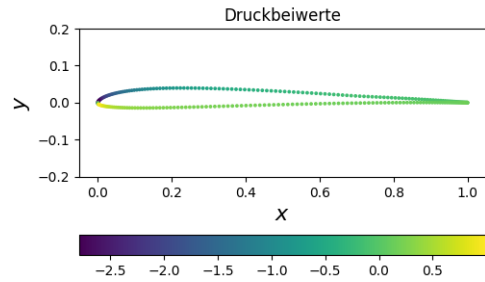
(a)



(b)



(c)



(d)

Abbildung 3.9: a) x- und b) y-Komponente des Geschwindigkeitsvektors. c Quellbelegung und d Druckbeiwerte pro Panel  $\mathcal{C}_i$  unter dem Anströmwinkel  $\alpha = 5^\circ$

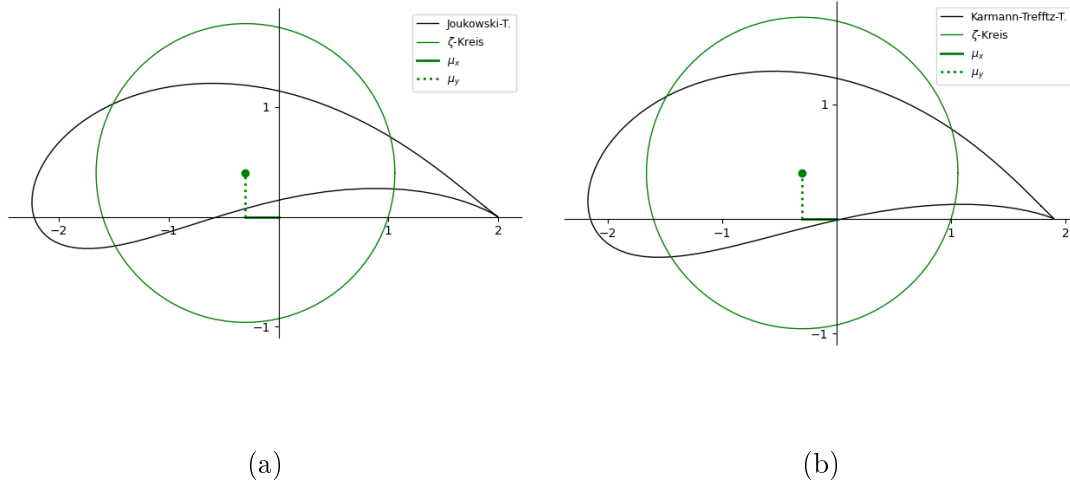


Abbildung 3.10: Vergleich der a) Joukowski-Transformation mit der b) Kármán-Trefftz-Transformation. Für beide Transformationen wurde  $\mu_x = 0.3, \mu_y = 0.4$  gewählt; für die Kármán-Trefftz-Transformation wurde  $n = 1.9$  gesetzt.

### 3.4 Abschätzung der Fehlerordnung anhand des Joukowski-Profiles

Die Joukowski-Transformation ist eine konforme Abbildung, deren Transformation als Ergebnis Tragflächenprofile liefert. Die Transformationsgleichung lautet:

$$z = \zeta + \frac{a^2}{\zeta}, \quad (3.15)$$

mit einem reellen Parameter  $a$  welcher im Folgenden  $a = 1$  gesetzt wird. Dabei ist  $z = x + iy$  eine komplexe Zahl im Bildraum und  $\zeta = \chi + i\eta$  eine komplexe Zahl im Ursprungsraum.

Ein Joukowski-Profil wird im  $z$ -Raum erzeugt durch die Anwendung der Joukowski-Transformation auf einen Kreis im  $\zeta$ -Raum. Einsetzen ergibt die realen und imaginären Komponenten

$$x = \frac{\chi(\chi^2 + \eta^2 + 1)}{\chi^2 + \eta^2}, \quad y = \frac{\eta(\chi^2 + \eta^2 - 1)}{\chi^2 + \eta^2}. \quad (3.16)$$

Durch Verschiebung des Mittelpunkts Kreises in der  $\zeta$ -Ebene können so verschiedenen Profile generiert werden. Die Abbildung hat allerdings an den Stellen  $z = \pm 1$

eine Singularität. Deshalb legt man einen der Punkte (hier  $z = -1$ ) ins Innere des Kreises und den Punkt  $z = 1$  auf den Kreis. In diesem Fall ergibt sich an der Hinterkante ein Winkel von  $0^\circ$ .

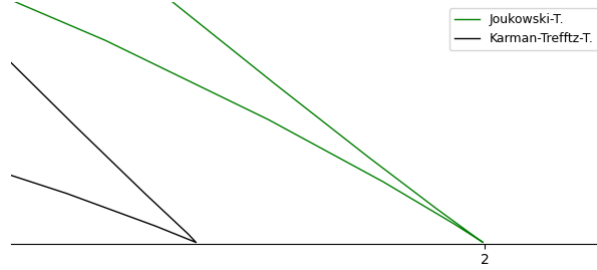


Abbildung 3.11: Vergleich der Hinterkantenwinkel des Joukowski- und Kármán-Trefftz-Profiles

Ein Profil mit einem von Null verschiedenen Winkel an der Hinterkante lässt sich durch eine verwandte Transformation, die Kármán-Trefftz-Transformation

$$z = nb \frac{(\zeta + b)^n + (\zeta - b)^n}{(\zeta + b)^n - (\zeta - b)^n}, \quad (3.17)$$

erzeugen, wiederum mit einem reellen Parameter  $b$  (im Folgenden  $b = 1$ ) und  $n \in [1, 2]$ , wobei sich für  $n = 1$  der Kreis in der  $\zeta$ -Ebene und für  $n = 2$  die Joukowski-Transformation ergibt. Die beiden Transformationen sind in Abbildung 3.10 unter gleichen Parametern dargestellt. Abbildung 3.11 gibt einen näheren Blick auf die Profilhinterkante.

Die analytische Lösung einer ebenen Potentialströmung um einen Kreiszylinder ist bekannt (siehe 3.2.2). Damit ist auch eine Lösung für Joukowski-Profile gegeben. Für die Zirkulation  $\Gamma = \sum_{i=1}^{n-1} \gamma l_i$  ergibt sich

$$\Gamma = 4V_\infty R \sin \left( \alpha + \arcsin \frac{\mu_y}{R} \right). \quad (3.18)$$

Mit dem Kutta-Jukowski-Theorem für den dynamischen Auftrieb  $F_a = \rho \Gamma V_\infty$  ist über den dynamischen Auftrieb mit  $c_a = \frac{F_a}{\rho t} = \frac{2F_a}{\rho V_\infty}$  einen Ausdruck für den Auftriebsbeiwert gegeben:

$$c_a = \frac{8RV_\infty \sin \left( \alpha + \arcsin \frac{\mu_y}{R} \right)}{V_\infty t} \quad (3.19)$$

Es wurde nun für Joukowski-Profile mit den Parametern  $\mu_x = 0.2, \mu_y = 0.1$  (siehe 3.13) für 10 bis 200 Panele der Auftriebsbeiwert berechnet und mit dem theoretischen Resultat aus (3.19) verglichen. Es ist ersichtlich, dass sich der Auftriebs-

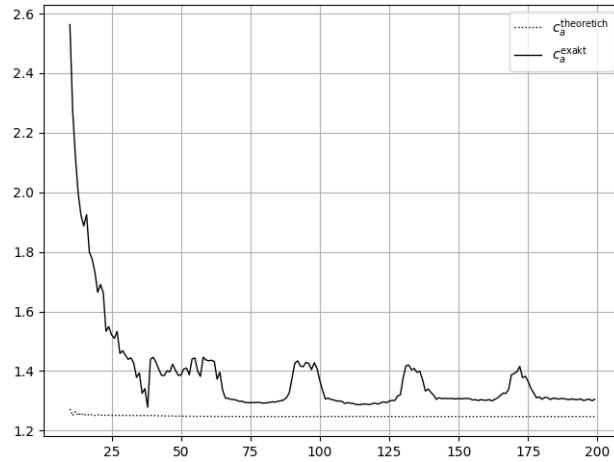


Abbildung 3.12: Vergleich des theoretisch ermittelten Werts für den Auftriebsbeiwert eines Joukowski-Profiles mit  $\mu_x = 0.2, \mu_y = 0.1$  und dem berechneten Wert.

beiwert ab einer Panelanzahl von ungefähr 75 stabil eingeschwungen hat, bis auf Schwankungen, welche Vermutlich auf der Diskretisierungsfunktion des Joukowski-Profiles beruhen. Für den mittleren relativen Fehler eines Joukowski-Profiles mit mindesten 75 Panels wurde der Wert

$$\frac{\Delta c_a}{c_a^{\text{theoretisch}}} = \frac{|c_a^{\text{theoretisch}} - c_a^{\text{exakt}}|}{c_a^{\text{theoretisch}}} = 0.062 = 6.2\%$$

ermittelt. [1] [3]

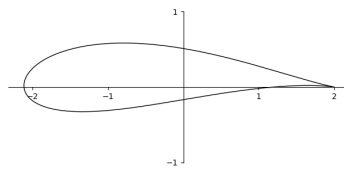


Abbildung 3.13: Joukowski-Profil,  $\mu_x = 0.2, \mu_y = 0.1$



# Literatur

- [1] Carlos Abello, Pedro Pablo Cárdenas Alzate und Jose Granada. „A survey of Joukowski airfoil and von Karman vortex street“. In: *Contemporary Engineering Sciences* 11 (Jan. 2018), S. 4921–4928. DOI: 10.12988/ces.2018.810540.
- [2] J.J. Alonso. *Hess-Smith Panel Method*. 2005. URL: [http://aero-comlab.stanford.edu/aa200b/lect\\_notes/lect3-4.pdf](http://aero-comlab.stanford.edu/aa200b/lect_notes/lect3-4.pdf) (besucht am 28.07.2022).
- [3] Lorena Barba und Olivier Mesnard. „Aero Python: classical aerodynamics of potential flow using Python“. In: *Journal of Open Source Education* 2.16 (2019), S. 45. DOI: 10.21105/jose.00045. URL: <https://doi.org/10.21105/jose.00045>.
- [4] T. Cebeci. *An Engineering Approach to the Calculation of Aerodynamic Flows*. 1. Aufl. Berlin Heidelberg: Springer, 1999. ISBN: 978-3-540-66181-8.
- [5] Charles R. Harris u. a. „Array programming with NumPy“. In: *Nature* 585.7825 (Sep. 2020), S. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [6] J.L. Hess und A.M.O. Smith. „Calculation of Potential Flow About Arbitrary Bodies“. In: *Progress in Aerospace Sciences* 8 (1966), S. 117–137.
- [7] J. D. Hunter. „Matplotlib: A 2D graphics environment“. In: *Computing in Science & Engineering* 9.3 (2007), S. 90–95. DOI: 10.1109/MCSE.2007.55.
- [8] Pauli Virtanen u. a. „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python“. In: *Nature Methods* 17 (2020), S. 261–272. DOI: 10.1038/s41592-019-0686-2.

## Anhang A: Lednicer- und Selig-Format

Es folgt eine Gegenüberstellung des Selig-Formats (links) mit dem Lednicer-Format (rechts) am Beispiel des NACA M13-Profiles.

NACA M13 AIRFOIL	NACA M13 AIRFOIL
1.00000000 0.00440000	17 17
0.95000000 0.01056000	0.000000 0.000000
0.90000000 0.01642000	0.012500 0.012780
0.80000000 0.02924000	0.025000 0.019370
0.70000000 0.04186000	0.050000 0.029640
0.60000000 0.05378000	0.075000 0.037910
0.50000000 0.06360000	0.100000 0.044680
0.40000000 0.06942000	0.150000 0.055420
0.30000000 0.07014000	0.200000 0.062760
0.20000000 0.06276000	0.300000 0.070140
0.15000000 0.05542000	0.400000 0.069420
0.10000000 0.04468000	0.500000 0.063600
0.07500000 0.03791000	0.600000 0.053780
0.05000000 0.02964000	0.700000 0.041860
0.02500000 0.01937000	0.800000 0.029240
0.01250000 0.01278000	0.900000 0.016420
0.00000000 0.00000000	0.950000 0.010560
0.00000000 0.00000000	1.000000 0.004400
0.01250000 -0.00812000	0.000000 0.000000
0.02500000 -0.00813000	0.012500 -0.008120
0.05000000 -0.00791000	0.025000 -0.008130
0.07500000 -0.00439000	0.050000 -0.007910
0.10000000 -0.00222000	0.075000 -0.004390
0.15000000 0.00192000	0.100000 -0.002220
0.20000000 0.00506000	0.150000 0.001920
0.30000000 0.00804000	0.200000 0.005060
0.40000000 0.00772000	0.300000 0.008040
0.50000000 0.00550000	0.400000 0.007720
0.60000000 0.00228000	0.500000 0.005500
0.70000000 -0.00064000	0.600000 0.002280
0.80000000 -0.00226000	0.700000 -0.000640
0.90000000 -0.00228000	0.800000 -0.002260
0.95000000 -0.00144000	0.900000 -0.002280
1.00000000 0.00000000	0.950000 -0.001440
	1.000000 0.000000

## Anhang B: Beispielausgabe der Methode `.write_panels()`

Hier ist die Ausgabe der ersten 30 Zeilen der AirfoilProfile-Methode `.write_panels()` gezeigt, am Beispiel des NASA: HSNLF(1)-0213 Profils gezeigt.

```
X_i,Y_i,theta_i,l_i
0.9950043,-0.0009728,0.06055800732379477,0.010009748628212382
0.9850132,-0.0016043,0.06568588453475652,0.010012392145736168
0.97752025,-0.0021092000000000003,0.07047228200753602,0.005007529407801845
0.96253695000000001,-0.00330425,0.08141266954462605,0.025054484997700546
0.937568,-0.00552425,0.09593394112296426,0.02508172911124744
0.92258845,-0.0069828,0.10270780128934252,0.005019149972854041
0.910107,-0.0083076,0.10650475541673622,0.020084001420035736
0.8876414,-0.010771800000000002,0.11144687148830609,0.025116819156891672
0.8626817,-0.01365475,0.11853981264059243,0.025134786388787998
0.8377253,-0.01677495,0.13021388185551733,0.025167464110831598
0.82275295,-0.018755849999999998,0.13816384340639531,0.00503831234641125
0.81027805,-0.02055885,0.1448782150472314,0.020170819682402617
0.79032025,-0.0235841,0.15598759087809566,0.02020137326445906
0.77784795,-0.02556705,0.16438466111254837,0.005056667726873056
0.76288615,-0.0283095,0.184653165616612,0.02536632781405301
0.73795785,-0.03345505,0.22232106375561403,0.025550337335150823
0.7130314,-0.0387352,0.19509280338564264,0.025413502189190737
0.6905846,-0.0425474,0.13433154712117715,0.02014367279519793
0.6781074,-0.04417745,0.11216297598033555,0.005023767722536545
0.6631284,-0.0456803,0.09755918696978584,0.025085082682741986
0.63815875,-0.0478474,0.0755680809368733,0.02504497608084302
0.61318205,-0.0495071,0.05712775250964519,0.025020717455141107
0.58819985,-0.05077395,0.04420261223196785,0.025008928130969513
0.56321345,-0.05173975,0.03306342585001139,0.025001964705598602
0.53822330000000001,-0.05243635,0.02267137098916197,0.024998424208137613
0.51322960000000001,-0.0528728,0.012249641308583322,0.024997275443535843
0.48823320000000003,-0.0531112,0.006824603815373697,0.0249979821409649
0.4632347,-0.053202650000000004,0.0004920078324255504,0.02499960302584824
0.43823365000000003,-0.053107600000000005,6.275090293523316,0.025003319219855574
```

## Anhang C: Codeausschnitte

Hier sind ausgewählte Codeausschnitte gezeigt, welche zur Lösung der Problemstellungen geschrieben wurden.

### Die Klasse Panel

Die Klasse Panel modelliert die Panels  $\mathcal{C}_i$  eines gegebenen Profils. Gespeichert werden neben den charakteristischen Parametern  $X_i, Y_i, \theta_i, l_i$  auch der Normalwinkel des Panels  $\delta_i$ , welcher aus dem Profil herauszieht. Ebenso wird die Panelposition als Ober- oder Unterseite bestimmt (für einige besondere Profile ist auch ein Wert "vertical" für komplett senkrechte Panele möglich). Ebenso gespeichert wird die Quellbelegung  $q_i$ , die Tangentialgeschwindigkeit  $v_i^{(t)}$  unter gegebenen Anströmwinkel  $\alpha$  und -geschwindigkeit  $V_\infty$ , und der resultierende Druckbeiwert  $c_{p_i}$ .

**class** Panel:

```
def __init__(self, xa, ya, xb, yb):
    self.xa, self.ya = xa, ya # panel starting-point
    self.xb, self.yb = xb, yb # panel ending-point

    self.xm = (xa + xb) / 2 # center of panel on x axis
    self.ym = (ya + yb) / 2 # center of panel on y axis
    # panel length
    self.length = np.sqrt((xb - xa) ** 2 + (yb - ya) ** 2)

    self.theta = atan2(yb - ya, xb - xa) # angle of panel
    # if angle is negative, add 2pi to only have positive angles for
    # plotting
    if self.theta < 0:
        self.theta += 2 * np.pi

    # normal angle of panel
    self.delta = self.theta - np.pi / 2
    if self.delta < 0:
        self.delta += 2 * np.pi
    if self.delta > 2 * np.pi:
        self.delta -= 2 * np.pi

    # panel location (used for plotting)
    if np.pi / 2 < self.theta < 3 * np.pi / 2:
        self.loc = 'upper' # upper surface
```

```

elif self.theta == np.pi / 2 or self.theta == 3 * np.pi / 2:
    self.loc = 'vertical'
else:
    self.loc = 'lower' # lower surface

self.q = None # source strength
self.vt = None # tangential velocity
self.cp = None # pressure coefficient

```

## Die Klasse AirfoilProfile

Die Klasse AirfoilProfile modelliert ein gegebenes Profil. Sie speichert neben wählbaren Namen und der ihr zugewiesenen Panele auch die Profiltiefe  $t$  und sämtliche Systemparameter  $\xi_{ij}$ ,  $\eta_{ij}$ ,  $I_{ij}$ ,  $J_{ij}$ ,  $A_{ij}^{(n)}$ ,  $A_{ij}^{(t)}$ ,  $M_{ij}$ .

Durch einen Aufruf der Methode `.solve(V, a)` mit gegebenen  $\alpha$  und  $V_\infty$  werden für alle zugeordneten Panele die Quellstärken, Tangentialgeschwindigkeiten und Druckbeiwerte berechnet (und in den jeweiligen Klassenvariablen abgespeichert). Dabei wird ebenfalls die Genauigkeit der Approximation  $\sum q_i l_i$  berechnet. Die Methode kann mit dem Schlüsselwortargument `vortex=True` aufgerufen werden, wodurch ebenfalls der Auftriebsbeiwert  $c_a$ , sowie die Wirbelbewegung  $\gamma$  berechnet werden. Die Methode `.write_panels()` erzeugt eine .csv-Datei, welche für jedes Panel die Werte  $X_i, Y_i, \theta_i, l_i$  in eine Zeile schreibt (siehe Anhang A: Lednicer- und Selig-Format).

Die Methode `.compute_free_vt(x, y, V, a)` ermöglicht die Berechnung der Tangentialgeschwindigkeiten an jedem Punkt  $(x_i, y_i)$  des Profils. Diese werden als Tupel von der Methode zurückgegeben.

**class** AirfoilProfile :

```

def __init__(self, panels, name, vortex=True):
    self.panels = panels
    self.name = name
    self.len = len(self.panels)
    self.vortex = vortex

    self.U = sum([panel.length for panel in self.panels])
    self.xi = compute_xi(self.len, self.panels)
    self.eta = compute_eta(self.len, self.panels)
    self.I = compute_I(self.len, self.xi, self.eta, self.panels)
    self.J = compute_J(self.len, self.xi, self.eta, self.panels)
    self.An = compute_An(self.len, self.I, self.J, self.panels)
    self.At = compute_At(self.len, self.I, self.J, self.panels)

```

```

self.M = system_matrix(self.An, self.At, self.vortex)

self.x = [panel.xa for panel in self.panels]
self.y = [panel.ya for panel in self.panels]
self.t = abs(max(self.x) - min(self.x))
self.coords = [(panel.xa, panel.ya, panel.xb, panel.yb) for panel in
                self.panels]
self.lower = [panel for panel in self.panels if panel.loc == "lower"]
self.upper = [panel for panel in self.panels if panel.loc == "upper"]

self.ca = None
self.accuracy = None
self.gamma = None

def solve(self, V=1, a=np.radians(4.0)):
    b = compute_inhomogeneity(self.panels, self.vortex, V, a)
    qs = np.linalg.solve(self.M, b)

    for i, panel in enumerate(self.panels):
        panel.q = qs[i]
    if self.vortex:
        self.gamma = qs[-1]

    if self.vortex:
        vt = []
        for i in range(self.len):
            vt.append(
                sum([self.At[i][j] * self.panels[j].q for j in range(self.
                    len)]) - self.gamma * sum(
                    [self.An[i][j] for j in range(self.len)]) + V * np.cos(
                        (
                            a - self.panels[i].theta))
            )
        else:
            vt = []
            for i in range(self.len):
                vt.append(
                    sum([self.At[i][j] * self.panels[j].q for j in range(self.
                        len)]) + V * np.cos(
                            a - self.panels[i].theta))
    for i, panel in enumerate(self.panels):

```

```

        panel.vt = vt[i]

    for panel in self.panels:
        panel.cp = 1 - (panel.vt / V) ** 2

    self.ca = - 2 / (V * self.t) * sum([panel.vt * panel.length for panel
        in self.panels])
    self.accuracy = sum([panel.q * panel.length for panel in self.panels])

def write_panels(self):
    header = ["X_i", "Y_i", "theta_i", "l_i"]
    with open("data/vals/" + self.name + "_vals.csv", "w+", encoding='
        UTF8', newline="") as file:
        writer = csv.writer(file)
        writer.writerow(header)
        #file.write(f"Circumference: {self.U} \n")
        #file.write(f"X_i, Y_i, theta_i, l_i\n")
        for panel in self.panels:
            #file.write(f"{panel.xm}, {panel.ym}, {panel.theta}, {panel.
                length} \n")
            writer.writerow([panel.xm, panel.ym, panel.theta, panel.length
                ])

def compute_free_vt(self, x, y, V=1, a=np.radians(4.0)):
    eta = compute_eta_free(len(x), self.len, self.panels, x, y)
    xi = compute_xi_free(len(x), self.len, self.panels, x, y)
    I = compute_I_free(len(x), self.len, xi, eta, self.panels)
    J = compute_J_free(len(x), self.len, xi, eta, self.panels)
    An = compute_An_free(len(x), self.len, I, J, self.panels)
    At = compute_At_free(len(x), self.len, I, J, self.panels)

    vtx = np.empty(len(x))
    vty = np.empty(len(x))
    for i in range(len(x)):
        vtx[i] = sum([At[i][j] * self.panels[j].q for j in range(self.len)
            ]) - self.gamma * sum(
            [An[i][j] for j in range(self.len)]) + V * np.cos(a)

        vty[i] = sum([An[i][j] * self.panels[j].q for j in range(self.len)
            ]) + self.gamma * sum(

```

```

[At[i][j] for j in range(self.len)]) + V * np.sin(a)

return vtx, vty

```

### **make\_cylinder(r,n)**

Diese Funktion wurde verwendet, um die  $x$ - und  $y$ -Koordinaten eines Kreiszylinder mit Radius  $r$  und  $n$  gleich langen Panels zu generieren.

```

def cylinder(r=1, n=8):
    # generate a cylinder with n equidistant panels
    a = np.linspace(0, 360, num=n+1, endpoint=True) / 180 * np.pi

    x = r * np.cos(a)
    y = r * np.sin(a)

    return x, y

```