DLR-IB-AE-GO-2020-137

An Implementation of the Vortex Lattice and the Doublet Lattice Method
Version 1.02

Arne Voß

DLR Deutsches Zentrum für Luft- und Raumfahrt

Dokumenteigenschaften

Titel	An Implementation of the Vortex Lattice and the Doublet Lattice Method	
Betreff		
Institut	DLR Institut für Aeroelastik	
Erstellt von	Dr.-Ing. Arne Voß	Arne Voß
Beteiligte		
Geprüft von	Dr.-Ing. Markus Ritter	Markus Ritter
Freigabe von	Prof. Dr.-Ing. Wolf-Reiner Krüger	W. Krüger
Datum	7. October 2020	
Version	1.0	
Dateipfad	D:\work\voss_ar\Projekte\Loads Kernel\LoadsKernel_DLM.fodt	

Dokument-Historie

Version	Datum	Änderungen	Autor
1.00	07.10.2020	Version published as report DLR-IB-AE-GO-2020-137	A. Voß
1.01	02.09.2021	Typing mistakes corrected in equations (2.38) and (2.51). Thank you to Mr. Oriol Chandre Vila, Airbus/SUPAERO, for the careful reading and spotting. The implementation remains unchanged.	A. Voß
1.02	10.05.2022	Minor typing mistakes corrected in equations (2.48), (2.49) (2.51) and (2.52). More consequent use of indices +e and -e in equations (2.14) to (2.18). Thanks to Prof. Sujin Bureerat, Khon Kaen University, for pointing out these corrections. The implementation remains unchanged.	A. Voß

Summary

This report documents the implementation of a Vortex Lattice Method (VLM) and a Doublet Lattice Method (DLM) in Python. The aerodynamic influence matrices (AICs) obtained from this implementation are validated with respect to MSC.Nastran for both the parabolic and the quartic integration schemes of the DLM. The test cases include dihedral and sweep of the main wing, wing-empennage configurations with the horizontal tail planar to the main wing, near-planar and further away (e.g. T-tail) and with/without a vertical tail. The test cases have been inspected at different mach numbers and reduced frequencies. For all tested aircraft configurations the results were found to be equivalent to MSC.Nastran in a numerical sense. Using panels which are misaligned in y-direction provokes differences and errors.

Table of Contents

Nomenclature

Abbreviations

AIC	Matrix of aerodynamic influence coefficients
DLM	Doublet lattice method
VLM	Vortex lattice method

Software

Loads Kernel	Loads and aeroelastic analysis software developed in this thesis
MSC.Nastran	NASA structural analysis code, commercially distributed by MSC software
Numpy	Python library for scientific and technical computing
Python	High-level programming language for general-purpose programming

Notation Conventions

$\mathbf{Q}$	Matrix
$\mathbf{q}$	Vector
q, Q	Scalars
$\bar{()}$	Local coordinates, typically of the receiving point with respect to sending point

Sub- and Superscripts

$()^{\mathrm{desc}}$	Descriptive superscript
$()_k$	Center point of aerodynamic panel
$()_l$	One quarter point of aerodynamic panel
$()_j$	Three quarter point of aerodynamic panel
$()_r$	Receiving panel
$()_s$	Sending panel

$()_{x,y,z}$	Quantity in the direction of or about the x, y and z axis	
$()_{1,2}$	Planar and non-planar contributions	

Variables and Parameters

Latin:

$A_{1,2}$ to $E_{1,2}$	Coefficients in the parabolic (only A, B and C) and the quartic integration schemes, planar and non-planar, respectively	
$\mathbf{AIC}$	Matrix of aerodynamic influence coefficients	
c_p	Pressure coefficient	
c_{ref}	Reference length	m
D_{rs}	Normal wash factor, $D_{0,rs}$ is the steady part, $D_{1,2,rs}$ are the incremental unsteady parts, planar and non-planar, respectively	
e	Panel semi span width	m
F	Term used in the parabolinc and the quartic integration	
$I_{1,2}$	Integrals inside the kernel functions K, planar and non-planar, respectively	
j	Imaginary number	
$i_{0,r,a,b,c}$	Indices to identify panels with different conditions, e.g. planar, near-planaer, etc.	Boolean
$K_{1,2}$	"The" kernel functions, planar and non-planar, respectively	
k	Reduced frequency	
Ma	Mach number	
$P_{1,2}$	Result from the evaluation of the kernel functions, planar and non-planar, respectively	
$T_{1,2}$	Landahl's direction cosine matrices	
V	Free stream velocity	m/s
x, y, z	Distances in cartesian coordinates	m

Greek:

α	Angle of attack, term used to calculate $F_{\text{parabolic}}$	$\circ$
ϵ	Term used to calculate F_{quartic}	
Γ	Circulation strength	m^2/s
γ	Dihedral angle of a panel	rad

λ	Sweep angle of a panel	rad
ω	Circular frequency	$1/s$

1 Introduction

1.1 Background of this Implementation

The implementation of the vortex lattice and the doublet lattice method has been developed as a part of the Loads Kernel Software [19], which allows for the calculation of quasi-steady and dynamic maneuver loads, unsteady gust loads in the time and frequency domain as well as dynamic landing loads. The Loads Kernel Software was mainly developed and used within PhD thesis of the author [18] for the investigation of a flying wing configuration. In the meantime, the Loads Kernel has been applied to various aircraft configurations (e.g. DLR-F19, MULDICON, Discus 2c, FS35, ALLEGRA, HALO, XRF1,...) successfully. However, during the analysis of a High Altitude Platform (HAP alpha), differences with respect to MSC.Nastran were discovered that could be traced back to differences in the aerodynamic influence coefficients (AIC), which was the catalyst for a closer inspection, resulting in the need for a reliable, more generic implementation of the DLM that covers more geometrical conditions such as dihedral, sweep, and different tail configurations.

1.2 Literature on the Doublet Lattice Method

Most of the time, Albano and Rodden 1968 [1] is cited when referring to the DLM. It should be mentioned that the origins of compressible, unsteady aerodynamic theories date back to the early 1940s when for example Küssner published his General Airfoil Theory [9], referring again to Prandtl [13] who introduced the theory of a lifting surface based on potential accelerations instead of velocity fields in 1936. The translation of Küssners work into English language by NACA [10], shows that the development of the DLM was an international effort with contributions of scientists from multiple nations. At the time of formulation, its solution was only possible for some special cases and it took three decades until, in the late 1960s, Albano and Rodden had the computational power available for a general, numerical solution applicable to arbitrary three-dimensional wings. Therefore, Albano and Rodden didn't actually invent the DLM but were those bright minds who properly implemented the DLM for the first time. For further reading, a historical overview is given by R. Voss [20] and a very comprehensive work on the mathematical derivation is published by Blair [2].

Next to the first, original DLM publication by Albano and Rodden 1968 [1], there is a series of further publications introducing modifications and additions to the DLM, which are summarized below.

Albano and Rodden 1968 [1]

- The first, original DLM paper.
- Parabolic integration.
- Watkins approximation [21].
- Shortcomings for wing-tail combinations with small offset in z-direction, the near-planar case.

Rodden et al. 1971 [14]

- Refinement of the non-planar part with additional condition for the near-planar case.
- Planar part as before but with a rearrangement of formulas and new nomenclature.
- Laschka approximation [12] with 11 coefficients.
- Changed signs for $K_{1,2}$.
- Analytical solution for $K_{10,20}$.
- Probably the most useful publication with many details required when attempting an implementation.

Rodden et al. 1972 [15]

- Same as above but using a new formulation with $\delta_{1,2}$ that places the arctangent in the correct quadrant. In most programming languages arctan2 does the job, but for some reasons that is not used in the Nastran implementation.

Blair 1992 [2]

- Comprehensive work on the mathematical derivation.
- Planar surfaces only, therefore educational application only.

Rodden et at. 1998 [16]

- Quartic integration for higher reduced frequencies and/or higher allowable panel aspect ratios.
- Desmarais approximation [3,4] with 12 coefficients.

Rodden et al. 1999 [17]

- Numerical comparisons between parabolic and quartic integration schemes.
- Draft version of an unpublished report/manuscript?

There are only a few implementations of the DLM the author is aware of. They include different numbers of features from the publications listed above, a summary is given in Table 1.1. The DLM published by Kotikalpudi [7,8] is a Matlab code with its origins at the Universities of Iowa and Minnesota. The author used this implementation for a long time, has been in contact with Kotikalpudi and is very grateful for his publication. However, it has the shortcomings of Albano and Rodden 1968. At the DLR Institute of Aeroelasticity, there is a Fortran implementation by Ralph Voss, which can be used in combination with a Python wrapper but the quartic integration is missing. Another implementation at the DLR Institute of System Dynamics and Control is a Matlab code by Thiemo Kier, which is used within the Varloads framework. To the author's best knowledge, that implementation is very mature and includes most features. Both implementations are in-house tools and not publicly available. The original Nastran source code [22] by NASA is published on GitHub and includes the DLM implementation as well. However, the Fortran code is not documented, difficult to understand and the quartic integration is missing. Finally, there is the commercial Nastran version distributed by MSC. The AIC matrices may be extracted via a so-called DAMP alter in the OP4 format, which requires a OP4 reader and several additional steps when thinking of an efficient workflow. Licensing might be an issue as well. The considerations above as well as academic interest led to the present implementation, including all desired features as indicated in the last column of Table 1.1.

	Kotikal-pudi	R. Voss	T. Kier	NASA Nastran	MSC Nastran	This version
Wing-empennage planar or further away	x	x	x	x	x	x
Wing-empennage near-planar		x	x	x	x	x
Parabolic integration	x	x	x	x	x	x
Quartic integration			x		x	x
Watkins approx.	x					x
Laschka approx.		x	x	x	x	x
Desmarais approx.			x		x	x
License	Free	In-house	In-house	Free	Commer-cial	In-house (for now)
Language	Matlab	Fortran	Matlab	Fortran	Fortran	Python

Table 1.1: Overview of different DLM implementations

2 Implementation

2.1 Definitions

Both the VLM and the DLM are based on a matrix of aerodynamic influence coefficients $\mathbf{AIC}$, which depends on the Mach number Ma, the reduced frequency k defined by

$$k = \frac{\omega}{V} \tag{2.1}$$

and the geometry of the aircraft. Note that the "Nastran definition" of the reduced frequency adds $c_{\mathrm{ref}}/2$, leading to

$$k = \frac{c_{ref}}{2V} \cdot \omega \; . \tag{2.2}$$

The geometry is discretized using an aerodynamic panel mesh. The $\mathbf{AIC}$ matrix then relates an induced downwash $\mathbf{w}_j$ on each aerodynamic panel to a circulation strength Γ_j, which is translated to a complex pressure coefficient c_p.

$$\mathbf{\Delta c_p} = \mathbf{AIC}(Ma, k) \cdot \mathbf{w}_j \tag{2.3}$$

With $k = 0$ for the quasi static case, the solution of the DLM is equivalent to the VLM. Because the DLM involves numerical integration and approximation, the VLM is more accurate. In this implementation, the steady part of the DLM is subtracted from the solution and replaced later by the VLM solution and the DLM returns only the unsteady increment.

In the current implementation, the aerodynamic grid is stored in a python dictionary that contains the items as given in Table 1.1. For loads and aeroelastic analyses, the ¼, ½ and ¾ points are important, which have the index 'l', 'k' and 'j' respectively as sketched in Figure 2.1. In addition to that, for the corner points of the vortex / doublet filament are required, which have the index ‘1’ and ‘3’. The points $\mathbf{p}_1$, $\mathbf{p}_3$ and $\mathbf{p}_{0,j}$ as well as the panel normal vectors $\mathbf{N}$ and the remaining geometrical parameters are assumed as a given.

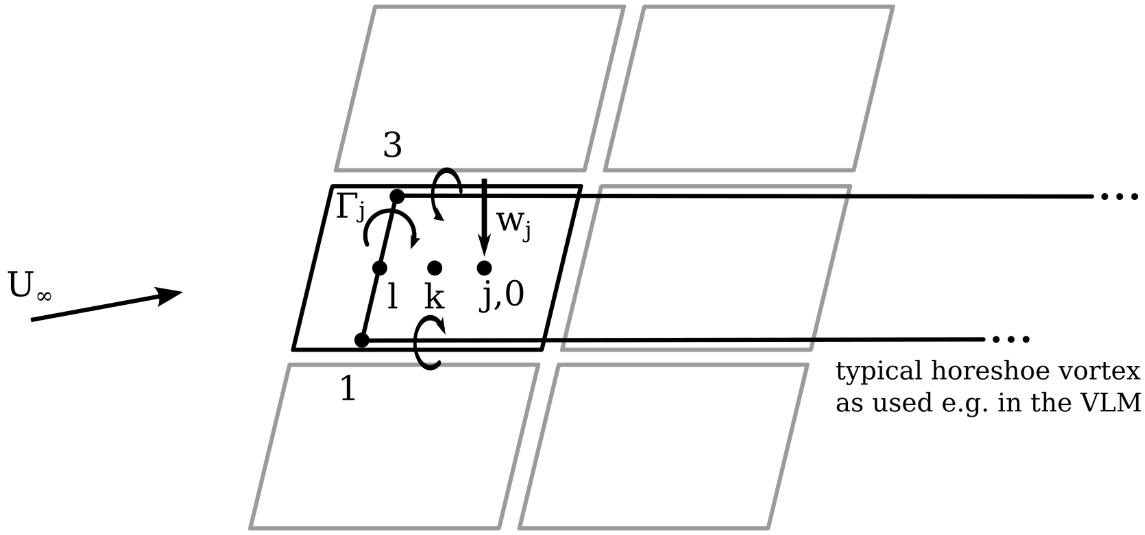

Figure 2.1: The geometrical mesh for the vortex lattice method

	Description	**Type**	**Shape**
aerogrid['offset_l']	¼ point	np.array	n x 3
aerogrid['offset_k']	½ point	np.array	n x 3
aerogrid['offset_j']	¾ point	np.array	n x 3
aerogrid['offset_P1']	Inner vortex / doublet point	np.array	n x 3
aerogrid['offset_P3']	Outer vortex / doublet point	np.array	n x 3
aerogrid['N']	Panel normal vector	np.array	n x 3
aerogrid['A']	Panel area	np.array	n
aerogrid['l']	Panel length	np.array	n
aerogrid['n']	Number of panels	int	-

Table 2.1: Definition of the aerodynamic grid

2.2 The Vortex Lattice Method

The formulation of the VLM used and described herin follows closely the derivation given by Katz and Plotkin [6] using horse shoe vortices. An implementation of the VLM is publicly available from Kotikalpudi [7,8]. That version is translated from Matlab to Python for performance and independence from commercial licenses. It is adapted to respect the dihedral of the wings, and the Prandtl-Glauert transformation is introduced. In addition, the computational process is split into more sub-functions to allow for the extraction of matrices at different stages, e.g. to obtain the circulation matrix $\mathbf{\Gamma}$ and the $\mathbf{AIC}$ matrices for both lift and drag.

In a first step, the Prandtl-Glauert transformation with $\beta = \sqrt{(1 - Ma^2)}$ is applied to the geometry by division of the x-coordinates by β as suggested by Hedman [5]. The next steps are described in Katz and Plotkin [6] in section 10.4.5 and 10.4.7. The induced velocities $\mathbf{D}^1_{u,v,w}$ at $\mathbf{p}_0$ from the vortex line segment between $\mathbf{p}_0$ and $\mathbf{p}_3$ are given by

$$\mathbf{D}^1_{u,v,w} = \frac{\mathbf{\Gamma}}{4\pi \left|\mathbf{r}_1 \times \mathbf{r}_2\right|^2} \cdot \left(\frac{\mathbf{r}_0 \cdot \mathbf{r}_1}{|\mathbf{r}_1|} - \frac{\mathbf{r}_0 \cdot \mathbf{r}_2}{|\mathbf{r}_2|} \right) \tag{2.4}$$

where

$$\mathbf{r}_0 = \mathbf{p}_3 - \mathbf{p}_1 \tag{2.5}$$

$$\mathbf{r}_1 = \mathbf{p}_0 - \mathbf{p}_1 \tag{2.6}$$

$$\mathbf{r}_2 = \mathbf{p}_0 - \mathbf{p}_3 \tag{2.7}$$

and with an assumed circulation strength $\mathbf{\Gamma}$ of unity. Singular cases arise when $|r_1| = 0$, $|r_2| = 0$ or $\left|\mathbf{r}_1 \times \mathbf{r}_2\right|^2 = 0$. In these cases, the induced velocity is set to 0. The induced velocities $\mathbf{D}^2_{v,w}$ at $\mathbf{p}_0$ from the inner horseshoe vortex are calculated with a simplified procedure

$$\mathbf{D}^2_v = \frac{-\sin(\gamma)}{4\pi \mathbf{d}} \cdot (\cos(\beta_1) - \cos(\beta_2)) \tag{2.8}$$

$$\mathbf{D}^2_w = \frac{-\cos(\gamma)}{4\pi \mathbf{d}} \cdot (\cos(\beta_1) - \cos(\beta_2)) \tag{2.9}$$

where γ is the dihedral angle of each panel, the cosines are given by $\cos(\beta_1) = 1.0$ and $\cos(\beta_2) = -r_{1,x}/|r_1|$ and $\mathbf{d}$ is the distance between the horseshoe vortex and points $\mathbf{p}_0$. The induced velocities $\mathbf{D}^3_{v,w}$ at $\mathbf{p}_0$ from the outer horseshoe vortex are calculated as above but with the cosines given by $\cos(\beta_1) = r_{2,x}/|r_2|$ and $\cos(\beta_2) = -1.0$. Singular cases arise when $|r_1| = 0$, $|r_2| = 0$ or $|d| = 0$ and for these cases, the induced velocity must be to 0 as well.

The final induced velocities $\mathbf{D}^{1,2,3}$ are composed of

$$\mathbf{D}^{1,2,3} = \mathbf{D}^{1,2,3}_w \cdot \mathbf{N}_z + \mathbf{D}^{1,2,3}_v \cdot \mathbf{N}_y \tag{2.10}$$

where the velocity components $\mathbf{D}_{v,w}$ are weighted by the normal vector $\mathbf{N}$ to account for dihedral and vertical surfaces. Superposition of the induced velocities from all three component for the horseshoe vortex leads to

$$\mathbf{D} = \mathbf{D}^1 + \mathbf{D}^2 + \mathbf{D}^3 \,. \tag{2.11}$$

The Kutta-Joukowski theorem relates circulation with lift per unit span width. Multiplication with panel areas $\mathbf{A}$ and division by panel span widths $\mathbf{b}$ ensures that matrix $\mathbf{A}_{jj}$ maps induced velocities to pressure coefficients $\Delta\mathbf{c}_p$ instead of circulation strength $\mathbf{\Gamma}$.

$$\mathbf{A}_{jj} = \mathbf{D} \cdot \frac{\mathbf{A}}{2\mathbf{b}} \tag{2.12}$$

Finally, the matrix of aerodynamic influence coefficients $\mathbf{AIC}$ is defined as the negative inverse

$$\mathbf{AIC} = \mathbf{Q}_{jj} = -\mathbf{A}^{-1}_{jj} \tag{2.13}$$

to fit into equation (2.3) from the beginning.

Note that the VLM code is vectorized for higher performance (no for-loops), therefore scalar values typically turn into vectors and vectors typically turn into matrices.

2.3 The Doublet Lattice Method

The geometrical points used for the VLM (compare Figure 2.1) are reused for the DLM and translated to the nomenclature introduced by Rodden et al. 1968 [1]. The receiving point 'r' is equivalent to 'j', the sending point 's' is equivalent to 'l', and the inner and outer dipole points with index '-e' and '+e' are equivalent to points '1' and '3'. The formulas are arranged in a different sequence than in the original publications to be as close to the actual implementation as possible. Note that, for simplicity, the following formulas are given for one single aerodynamic panel, the code itself is vectorized and uses mainly vectors and matrices.

The semi span width of a panel is given by

$$e = 0.5 \cdot \sqrt{(p_{+e,z} - p_{-e,z})^2 + (p_{+e,y} - p_{-e,y})^2}\ . \tag{2.14}$$

The cartesian coordinates of a receiving point relative to a sending point are given by

$$x, y, z_{sr} = p_r - p_s\ . \tag{2.15}$$

The dihedral angle defined by $\gamma = \arctan(d_z/d_y)$ and the sweep angle defined by $\lambda = \arctan(d_x/d_y)$ yield

$$\sin(\gamma) = (p_{+e,z} - p_{-e,z})/(2 \cdot e)\ , \tag{2.16}$$

$$\cos(\gamma) = (p_{+e,y} - p_{-e,y})/(2 \cdot e)\ , \tag{2.17}$$

$$\tan(\lambda) = (p_{+e,x} - p_{-e,x})/(2 \cdot e)\ , \tag{2.18}$$

and a relative dihedral angle γ_{sr} is calculated between receiving and sending panels. The local coordinates $\bar{y}$ and $\bar{z}$ of the receiving point relative to the sending point are calculated with

$$\bar{y} = y_{sr} \cdot \cos(\gamma) + z_{sr} \cdot \sin(\gamma) \tag{2.19}$$

and

$$\bar{z} = z_{sr} \cdot \cos(\gamma) - y_{sr} \cdot \sin(\gamma)\ . \tag{2.20}$$

The DLM as in Rodden et al. 1971 [14] differentiates between three cases for the arrangement of wing and horizontal tail with respect to their distance in z-direction. The planar condition is identified by

$$i_0 = |\bar{z}|/e \leq 0.001\ , \tag{2.21}$$

which is adapted from the Nastran source code [22] and allows for small geometrical inaccuracies which might arise from a text-based input. For the non-planar part, a ratio

$$\text{ratio} = 2.0 \cdot e \cdot |\bar{z}| / (\bar{y}^2 + \bar{z}^2 - e^2) \tag{2.22}$$

is calculated. The co-planaer / close-by condition is given by

$$i_a = |\text{ratio}| \leq 0.3 \quad \& \quad |\bar{z}| / e > 0.001 \tag{2.23}$$

and the remaining panels, which are further away, are identified by

$$i_r = |\text{ratio}| > 0.3 \quad \& \quad |\bar{z}| / e > 0.001 \, . \tag{2.24}$$

2.3.1 Parabolic Integration

Based on the conditions from above, a term $F_{\text{parabolic}}$ is calculated for the planar case using

$$F_{\text{parabolic}}(i_0) = \frac{2.0 \cdot e}{(\bar{y}^2 - e^2)} \, , \tag{2.25}$$

for the near-planar case using (compare Rodden et al. 1971 [14] eq. 32)

$$F_{\text{parabolic}}(i_a) = \frac{2.0 \cdot e}{\bar{y}^2 + \bar{z}^2 - e^2} \cdot (1.0 - \alpha \cdot \frac{\bar{z}^2}{e^2}) \, , \tag{2.26}$$

with (compare Rodden et al. 1971 [14] eq. 33)

$$\alpha(i_a) = \frac{4 \cdot e^4}{(\bar{y}^2 + \bar{z}^2 - e^2)^2} \cdot \sum_{n=2}^{7} \frac{(-1)^n}{2n - 1} \cdot \text{ratio}^{(2n-4)}, \tag{2.27}$$

and for the rest using (compare Rodden et al. 1971 [14] eq. 31b)

$$F_{\text{parabolic}}(i_r) = \frac{1}{|\bar{z}|} \cdot \text{arctan2} \left(\frac{2 \cdot e \cdot |\bar{z}|}{\bar{y}^2 + \bar{z}^2 - e^2} \right) . \tag{2.28}$$

Note that it is important to use arctan2, which places the arctangent in the range from 0 to π. In a next step, the kernel functions described in section 2.3.3 are evaluated three times along the panel span width with $\bar{e} = -e$, $\bar{e} = +e$ and $\bar{e} = 0$, resulting in matrices $P_1(\bar{e})$ and $P_2(\bar{e})$. The following terms are used for the parabolic integration (compare Rodden et al. [14] eq. 28-30 and eq. 37-39). Note that Rodden has the habit of leaving out some brackets, a likely source for confusion.

$$A_1 = \frac{P_1(-e) - 2 \cdot P_1(0) + P_1(+e)}{2e^2} \tag{2.29}$$

$$B_1 = \frac{P_1(+e) - P_1(-e)}{2e^2} \tag{2.30}$$

$$C_1 = P_1(0) \tag{2.31}$$

$$A_2 = \frac{P_2(-e) - 2 \cdot P_2(0) + P_2(+e)}{2e^2} \tag{2.32}$$

$$B_2 = \frac{P_2(+e) - P_2(-e)}{2e^2} \tag{2.33}$$

$$C_2 = P_2(0) \tag{2.34}$$

Matrix $D_{1,rs}$ is described as the “planar” part of normal wash matrix, which is a somewhat misleading expression as $D_{1,rs}$ is evaluated for both planar and non-planar panels. Matrix $D_{1,rs}$ is given by (compare Rodden et al. 1971 [14] eq. 34)

$$\begin{aligned} D_{1,rs}(i_0) = \frac{\text{chord}}{8\pi} \cdot [((\bar{y}^2 - \bar{z}^2) \cdot A_1 + \bar{y} \cdot B_1 + C_1) \cdot F_{\text{parabolic}} \\ + (0.5 \cdot B_1 + \bar{y} \cdot A_1) \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2}\right) \\ +2e \cdot A_1] \end{aligned} \tag{2.35}$$

Matrix $D_{2,rs}$ is the “non-planar” part of normal wash matrix, and distinguishes between two (new) conditions

$$i_b = |1/\text{ratio}| \leq 0.1 \quad \& \quad |\bar{z}|/e > 0.001 \tag{2.36}$$

and

$$i_c = |1/\text{ratio}| > 0.1 \quad \& \quad |\bar{z}|/e > 0.001\,. \tag{2.37}$$

Matrix $D_{2,rs}$ is given by (compare Rodden et al. 1971 [14] eq. 40)

$$\begin{aligned} D_{2,rs}(i_b) = \frac{\text{chord}}{16\pi \cdot \bar{z}^2} \cdot \Bigg[((\bar{y}^2 + \bar{z}^2) \cdot A_2 + \bar{y} \cdot B_2 + C_2) \cdot F_{\text{parabolic}} \\ + \frac{1.0}{(\bar{y}+e)^2 + \bar{z}^2} \cdot \Bigg(((\bar{y}^2 + \bar{z}^2) \cdot \bar{y} + (\bar{y}^2 - \bar{z}^2) \cdot e) \cdot A_2 \\ + (\bar{y}^2 + \bar{z}^2 + \bar{y} \cdot e) \cdot B_2 + (\bar{y} + e) \cdot C_2\Bigg) \\ - \frac{1.0}{(\bar{y}-e)^2 + \bar{z}^2} \cdot \Bigg(((\bar{y}^2 + \bar{z}^2) \cdot \bar{y} - (\bar{y}^2 - \bar{z}^2) \cdot e) \cdot A_2 \\ + (\bar{y}^2 + \bar{z}^2 - \bar{y} \cdot e) \cdot B_2 + (\bar{y} - e) \cdot C_2\Bigg)\Bigg] \end{aligned} \tag{2.38}$$

and (compare Rodden et al. 1971 [14] eq. 41)

$$D_{2,rs}(i_c) = \frac{\text{chord} \cdot e}{8\pi \cdot (\bar{y}^2 + \bar{z}^2 - e^2)} \cdot \left[\frac{2.0 \cdot (\bar{y}^2 + \bar{z}^2 + e^2) \cdot (e^2 \cdot A_2 + C_2) + 4.0 \cdot \bar{y} \cdot e^2 \cdot B_2}{((\bar{y} + e)^2 + \bar{z}^2) \cdot ((\bar{y} - e)^2 + \bar{z}^2)} - \frac{\alpha}{e^2} \cdot ((y^2 + \bar{z}^2) \cdot A_2 + \bar{y} \cdot B_2 + C_2) \right] \tag{2.39}$$

Note that α can not be take from eq. 2.27 because the conditions are different. Instead, it should be reconstructed from eq. 2.26.

In a last step, the planar and non-planar parts are added up

$$D_{rs} = D_{1,rs} + D_{2,rs} \tag{2.40}$$

leading to the final downwash matrix with $D_{rs} = A_{jj}$. Note that in eq. 22 in Rodden et al. 1971 [14] there is a third term $D_{0,rs}$ which subtracts the steady contribution, which has already been subtracted inside the kernel function in this implementation.

2.3.2 Quartic Integration

The quartic integration scheme is introduced by Roddel et al. 1998 [16] as a “further refinement”, which aims to relax the restriction to panel aspect ratios < 3...4. This allows for fewer panels or to calculate higher reduced frequencies with a given aerodynamic mesh compared to the parabolic integration scheme. Figures 4 and 5 in [17] visualize the differences clearly and make the improvement very obvious. However, the formulas are more complex compared to the parabolic scheme and the kernel functions need to be evaluated five times along the panel span width instead of only three times, leading to slightly higher computational times per panel. In addition, some rearrangements have been made in the formulas, e.g. using a term ϵ instead of α and parameters $d_{1,2}$ are introduced to place the arctangent in the right quadrant instead of using arctan2. Still, we have to use that formulation as $d_{1,2}$ and ϵ will be used later on in the calculation of $D_{2,rs}$. Note that there is a difference and/or mistake (?) in eq. 23 in Roddel et al. 1998 [16] compared to eq. 30b in Roddel et al. 1972 [15]. The following values appear to be correct:

$$d_1 = 1.0 \text{ and } d_2 = 0.0 \text{ for } \bar{y}^2 + \bar{z}^2 – e^2 > 0.0 \tag{2.41}$$

$$d_1 = 0.0 \text{ and } d_2 = 0.5 \text{ for } \bar{y}^2 + \bar{z}^2 – e^2 = 0.0 \tag{2.42}$$

$$d_1 = 1.0 \text{ and } d_2 = 1.0 \text{ for } \bar{y}^2 + \bar{z}^2 – e^2 < 0.0 \tag{2.43}$$

Based on the conditions i_0, i_a and i_r from before, the term F_{quartic} is calculated for the planar case using

$$F_{\text{quartic}}(i_0) = \frac{d_1 \cdot 2.0 \cdot e}{(\bar{y}^2 - e^2)}\,, \tag{2.44}$$

and (compare Rodden et al. 1998 [16] eq. 22)

$$F_{\text{quartic}}(i_a, i_r) = \frac{d_1 \cdot 2.0 \cdot e}{\bar{y}^2 + \bar{z}^2 - e^2} \cdot \left(1.0 - \epsilon \cdot \frac{\bar{z}^2}{e^2}\right) + d_2 \cdot \frac{\pi}{|\bar{z}|}\,. \tag{2.45}$$

Note that eq. 2.44 is actually equivalent to eq. 2.45 but without the terms including $\bar{z}$ because they would become singular if $\bar{z} = 0$. For the near-planar case, ϵ is equivalent to α as given in eq. 2.27 (compare Rodden et al. 1998 [16] eq. 25)

$$\epsilon(i_a) = \alpha(i_a), \tag{2.46}$$

and for the rest ϵ is calculated with (compare Rodden et al. 1998 [16] eq. 24)

$$\epsilon(i_r) = \frac{e^2}{\bar{z}^2} \cdot \left(1.0 - \frac{\bar{y}^2 + \bar{z}^2 - e^2}{2 \cdot e \cdot |\bar{z}|}\right) \arctan\left(\frac{2 \cdot e \cdot |\bar{z}|}{\bar{y}^2 + \bar{z}^2 - e^2}\right). \tag{2.47}$$

The two different ways of calculating F_{quartic} and $F_{\text{parabolic}}$ should lead to equivalent results in a numerical sense, thus comparing both results is a good cross-check. In a next step, the kernel functions described in section 2.3.3 are evaluated five times along the panel span width with $\bar{e} = -e$, $\bar{e} = -e/2$, $\bar{e} = +e$, $\bar{e} = +e/2$ and $\bar{e} = 0$, resulting in matrices $P_1(\bar{e})$ and $P_2(\bar{e})$. The following terms are used for the quartic integration (compare Rodden et al. 1998 [16] eq. 15-19 and eq. 28-32):

$$\begin{aligned}
A_1 &= \frac{-1}{6e^2} \cdot [P_1(-e) \quad -16P_1(-e/2) + 30P_1(0.0) \quad -16P_1(+e/2) + P_1(+e)] \\
B_1 &= \frac{+1}{6e} \cdot [P_1(-e) \quad -8P_1(-e/2) \quad +8P_1(+e/2) - P_1(+e)] \\
C_1 &= P_1(0.0) \\
D_1 &= \frac{-2}{3e^3} \cdot [P_1(-e) \quad -2P_1(-e/2) \quad +2P_1(+e/2) - P_1(+e)] \\
E_1 &= \frac{+2}{3e^4} \cdot [P_1(-e) \quad -4P_1(-e/2) + 6P_1(0.0) \quad -4P_1(+e/2) + P_1(+e)]
\end{aligned} \tag{2.48}$$

and

$$\begin{aligned}
A_2 &= \frac{-1}{6e^2} \cdot [P_2(-e) \quad -16P_2(-e/2) + 30P_2(0.0) \quad -16P_2(+e/2) + P_2(+e)] \\
B_2 &= \frac{+1}{6e} \cdot [P_2(-e) \quad -8P_2(-e/2) \quad +8P_2(+e/2) - P_2(+e)] \\
C_2 &= P_2(0.0) \\
D_2 &= \frac{-2}{3e^3} \cdot [P_2(-e) \quad -2P_2(-e/2) \quad +2P_2(+e/2) - P_2(+e)] \\
E_2 &= \frac{+2}{3e^4} \cdot [P_2(-e) \quad -4P_2(-e/2) + 6P_2(0.0) \quad -4P_2(+e/2) + P_2(+e)]
\end{aligned} \tag{2.49}$$

The “planar” part of normal wash matrix $D_{1,rs}$ is given by (compare Rodden et al. 1998 [16] eq. 20):

$$\begin{aligned}
D_{1,rs}(i_0) = \frac{\text{chord}}{8\pi} \cdot \Bigg[&[(\bar{y}^2 - \bar{z}^2)A_1 + \bar{y}B_1 + C_1 + \bar{y}(\bar{y}^2 - 3\bar{z}^2)D_1 \\
&+ (\bar{y}^4 - 6\bar{y}^2\bar{z}^2 + \bar{z}^4)E_1] \cdot F_{\text{quartic}} \\
&+ [0.5B_1 + \bar{y}A_1 + 0.5(3\bar{y}^2 - \bar{z}^2)D_1 \\
&+ 2\bar{y}(\bar{y}^2 - \bar{z}^2)E_1] \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2}\right) \\
&+ 2e\left[A_1 + 2\bar{y}D1 + (3\bar{y}^2 - \bar{z}^2 + \frac{1}{3}e^2)E_1\right]\Bigg]
\end{aligned} \tag{2.50}$$

The “non-planar” part of normal wash matrix $D_{2,rs}$ is given by (compare Rodden et al. 1998 [16] eq. 33 and 34):

$$
\begin{aligned}
D_{2,rs}(i_b) = \frac{\text{chord}}{16\pi \cdot \bar{z}^2} \cdot \Bigg[& F_{\text{quartic}} \cdot \Bigg((\bar{y}^2 + \bar{z})A_2 + \bar{y}B_2 + C_2 + \bar{y}(\bar{y}^2 + 3\bar{z}^2)D_2 \\
& \quad + (\bar{y}^4 + 6\bar{y}^2\bar{z}^2 - 3\bar{z}^4)E_2 \Bigg) \\
& + \frac{1.0}{(\bar{y}+e)^2 + \bar{z}^2} \cdot \Bigg(((\bar{y}^2 + \bar{z}^2)\bar{y} + (\bar{y}^2 - \bar{z}^2)e)A_2 \\
& \quad + (\bar{y}^2 + \bar{z}^2 + \bar{y}e)B_2 + (\bar{y}+e)C_2 + (\bar{y}^4 - \bar{z}^4 + (\bar{y}^2 - 3\bar{z}^2)\bar{y}e)D_2 \\
& \quad + ((\bar{y}^4 - 2\bar{y}^2\bar{z}^2 - 3\bar{z}^4)\bar{y} + (\bar{y}^4 - 6\bar{y}^2\bar{z}^2 + \bar{z}^4)e)E_2 \Bigg) \\
& - \frac{1.0}{(\bar{y}-e)^2 + \bar{z}^2} \cdot \Bigg(((\bar{y}^2 + \bar{z}^2)\bar{y} - (\bar{y}^2 - \bar{z}^2)e)A_2 \\
& \quad + (\bar{y}^2 + \bar{z}^2 - \bar{y}e)B_2 + (\bar{y}-e)C_2 + (\bar{y}^4 - \bar{z}^4 - (\bar{y}^2 - 3\bar{z}^2)\bar{y}e)D_2 \\
& \quad + ((\bar{y}^4 - 2\bar{y}^2\bar{z}^2 - 3\bar{z}^4)\bar{y} - (\bar{y}^4 - 6\bar{y}^2\bar{z}^2 + \bar{z}^4)e)E_2 \Bigg) \\
& + \left(\bar{z}^2 \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2} \right) \right) D_2 \\
& + 4\bar{z}^2 \left(e + \bar{y} \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2} \right) \right) E_2 \Bigg]
\end{aligned} \tag{2.51}
$$

and

$$
\begin{aligned}
D_{2,rs}(i_c) = \frac{\text{chord} \cdot e}{8\pi \cdot (\bar{y}^2 + \bar{z}^2 - e^2)} \cdot \Bigg[& \frac{1.0}{((\bar{y}+e)^2 + \bar{z}^2) \cdot ((\bar{y}-e)^2 + \bar{z}^2)} \\
& \cdot \Bigg(2(\bar{y}^2 + \bar{z}^2 + e^2) \cdot (e^2 A_2 + C_2) + 4\bar{y}e^2 B_2 \\
& \quad + 2\bar{y}(\bar{y}^4 - 2e^2\bar{y}^2 + 2\bar{y}^2\bar{z}^2 + 3e^4 + 2e^2\bar{z}^2 + \bar{z}^4)D_2 \\
& \quad + 2(3\bar{y}^6 - 7e^2\bar{y}^4 + 5\bar{y}^4\bar{z}^2 + 6e^4\bar{y}^2 + 6e^2\bar{y}^2\bar{z}^2 \\
& \qquad - 3e^2\bar{z}^4 - \bar{z}^6 + \bar{y}^2\bar{z}^4 - 2e^4\bar{z}^2)E_2 \Bigg) \\
& - \left(d_1\epsilon + \frac{e^2}{\bar{z}^2}\left(1 - d_1 - d_2\frac{\pi}{\text{ratio}}\right) \right) \frac{1}{e^2} \cdot ((\bar{y}^2 + \bar{z}^2)A_2 \\
& \quad + \bar{y}B_2 + C_2 + \bar{y}(\bar{y}^2 + 3\bar{z}^2)D_2 + (\bar{y}^4 + 6\bar{y}^2\bar{z}^2 - 3\bar{z}^4)E_2) \Bigg] \\
+ \frac{\text{chord}}{8\pi} \cdot \Bigg[& \frac{D_2}{2} \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2} \right) \\
& + 2\left(e + \bar{y} \cdot \log\left(\frac{(\bar{y}-e)^2 + \bar{z}^2}{(\bar{y}+e)^2 + \bar{z}^2} \right) \right) E_2 \Bigg]
\end{aligned} \tag{2.52}
$$

In a last step, the planar and non-planar parts are added up like before in eq. 2.40.

2.3.3 Kernel Functions

In this section, “the” kernel functions of the DLM are calculated. Again, Rodden has the habit of leaving out some brackets in his formulas. This applies to eq. 7, 8 and 11 where it is not very clear which parts belong to the denominator, however, that is clarified in the following. In a first step, some variables are defined, compare eq. 4, 9, 10, 11 and 12 in Rodden et al. [14].

$$r_1 = \sqrt{(\bar{y}-\bar{e})^2+\bar{z}^2} \tag{2.53}$$

$$\beta^2 = 1 - M^2 \tag{2.54}$$

$$R = \sqrt{(\bar{x}-\bar{e}\cdot\tan(\lambda))^2+\beta^2\cdot r_1^2} \tag{2.55}$$

$$u_1 = \frac{M\cdot R-\bar{x}+\bar{e}\cdot\tan(\lambda)}{\beta^2\cdot r_1} \tag{2.56}$$

$$k_1 = k\cdot r_1 \tag{2.57}$$

Here, the reduced frequency is defined by $k = \omega/V$, compare eq. 2.1 and 2.2. The exponential function

$$\mathrm{ejku} = e^{-j\cdot k_1\cdot u_1} \tag{2.58}$$

is pre-multiplied as it is used several times and is computationally expensive. The direction cosine matrices are given by (compare eq. 5 and 21 in Rodden et al. [14])

$$T_1 = \cos(\gamma_{sr}) \tag{2.59}$$

and

$$T_2 = \bar{z}\cdot(\bar{z}\cdot\cos(\gamma_{sr})+(\bar{y}-\bar{e})\cdot\sin(\gamma_{sr})) \tag{2.60}$$

The solution of the kernel functions $K_{1,2}$ is originally given by Landahl [11], compare also eq. 7 and 8 in Rodden et al. 1971 [14]. Note that the signs of terms for $K_{1,2}$ are switched in Rodden et al. 1971 [14] compared to Albano and Rodden 1968 [1], this implementation stays with the 1968 convention as we don't want to mess with the steady part from the VLM.

$$K_1 = -I_1 - \frac{\mathrm{ejku}\cdot M\cdot r_1}{R\cdot\sqrt{1.0+u_1^2}} \tag{2.61}$$

$$\begin{aligned}K_2 =& 3.0 \cdot I_2 + \frac{j \cdot k_1 \cdot \text{ejku} \cdot M^2 \cdot r_1^2}{R^2 \cdot (1.0 + u_1^2)^{0.5}} \\ &+ \frac{\text{ejku} \cdot M \cdot r_1}{R^2 \cdot (1.0 + u_1^2)^{1.5}} \cdot \left(\frac{(1.0 + u_1^2) \cdot \beta^2 \cdot r_1^2}{R^2} + 2.0 + \frac{M \cdot r_1 \cdot u_1}{R} \right)\end{aligned} \tag{2.62}$$

The evaluation of $K_{1,2}$ involves integrals $I_{1,2}$, which are more difficult to solve and require a numerical approximation as explained in section 2.3.4. The analytical solution of $K_{1,2}$ at $k = 0.0$ is given by (compare eq. 15 and 16 in Rodden et al. 1971 [14])

$$K_{10} = -1.0 - \frac{\bar{x} - \bar{e} \cdot \tan(\lambda)}{R} \tag{2.63}$$

and

$$K_{20} = 2.0 + \frac{(\bar{x} - \bar{e} \cdot \tan(\lambda)) \cdot (2.0 + \beta^2 \cdot r_1^2 / R^2)}{R} \,. \tag{2.64}$$

During the numerical evaluation, singularities may arise when $r_1 = 0.0$. In these cases, $K_1 = -2.0$ and $K_2 = +4.0$ when $\bar{x} \geq 0.0$ and $K_1 = 0.0$ and $K_2 = 0.0$ when $\bar{x} < 0.0$.

The final matrices of the kernel functions $P_1(\bar{e})$ and $P_2(\bar{e})$ are given by (compare eq. 27B and 36b in Rodden et al. 1971 [14])

$$P_1(\bar{e}) = -(K_1 \cdot e^{-j \cdot k \cdot (\bar{x} - \bar{e} \cdot \tan(\lambda))} - K_{10}) \cdot T_1 \tag{2.65}$$

and

$$P_2(\bar{e}) = -(K_2 \cdot e^{-j \cdot k \cdot (\bar{x} - \bar{e} \cdot \tan(\lambda))} - K_{20}) \cdot T_2 \,. \tag{2.66}$$

The steady parts K_{10} and K_{20} are directly subtracted from the solution, as the steady contribution will be added later from the VLM.

2.3.4 Integral Approximation

Integrals I_1 and I_2 (compare eq. 13 and 14 in Rodden et al. [14]) require an approximation to be solved in a computationally efficient way. There are a number of methods, e.g. by Watkins [21] (used in Albano and Rodden 1968 [1]), by Laschka [12] (used in Rodden et al. [14]) and by Desmarais [3] (used in Rodden et al. 1998 [16]). In the following, the last two are summarized.

The first step is to evaluate integral

$$I_0 = e^{j \cdot k_1 \cdot u_1} \cdot \int_{u_1}^{\infty} \left[1 - \frac{u_1}{(1 + u_1^2)^{0.5}} \right] e^{-j \cdot k_1 \cdot u_1} du_1 \,. \tag{2.67}$$

Using the approach

$$1 - \frac{u_1}{(1 + u_1)^{0.5}} = \sum_{n=1}^{11} a_n \cdot e^{-n \cdot c \cdot u_1} \tag{2.68}$$

as proposed by Laschka [12], leads to (compare eq. A.4 in Rodden et al. 1971 [14])

$$I_0(u_1, k_1) = \sum_{n=1}^{11} \frac{a_n \cdot e^{-n \cdot c \cdot u_1}}{n^2 \cdot c^2 + k_1^2} \cdot [n \cdot c - j \cdot k_1] \,, \tag{2.69}$$

where $c = 0.372$ and the coefficients a_n are as given in Table 2.2. The values are difficult to read in Rodden et al. 1971 due to the low quality of the digital reproduction of the original paper but can be found as well in Blair 1992 [2], page 89.

n	a_n
1	+0.24186198
2	-2.7918027
3	+24.991079
4	-111.59196
5	+271.43549
6	-305.75288
7	-41.183630
8	+545.98537
9	-644.78155
10	+328.72755
11	-64.279511

Table 2.2: Coefficients used in the Laschka approximation

The second integral is

$$J_0 = e^{j \cdot k_1 \cdot u_1} \cdot \int_{u_1}^{\infty} u_1 \cdot \left[1 - \frac{u_1}{(1 + u_1^2)^{0.5}} \right] e^{-j \cdot k_1 \cdot u_1} du_1 \,. \tag{2.70}$$

Using the same approach leads to (compare eq. A.8 in Rodden et al. 1971 [14])

$$J_0(u_1,k_1) = \sum_{n=1}^{11} \frac{a_n \cdot e^{-n\cdot c\cdot u_1}}{(n^2\cdot c^2+k_1^2)^2} \cdot [n^2\cdot c^2 - k_1^2 + n\cdot c\cdot u_1 \cdot (n^2\cdot c^2+k_1^2) \\ - j\cdot k_1 \cdot (2.0\cdot n\cdot c + u_1\cdot(n^2\cdot c^2+k_1^2)] \quad . \tag{2.71}$$

Then, I_1 is calculated with (compare eq. A.1 in Rodden et al. 1971 [14])

$$I_1 = \left[1.0 – \frac{u_1}{(1.0+u_1^2)^{0.5}} - j\cdot k_1\cdot I_0\right] \cdot e^{-j\cdot k_1\cdot u_1} \tag{2.72}$$

and I_2 is calculated with (compare eq. A.6 in Rodden et al. 1971 [14])

$$I_2 = \frac{1}{3} \cdot \left[(2.0 + j\cdot k_1\cdot u_1)\cdot\left[1.0 - \frac{u_1}{(1.0+u_1^2)^{0.5}}\right] \\ - \frac{u_1}{(1.0+u_1^2)^{1.5}} - j\cdot k_1\cdot I_0 + k_1^2\cdot J_0\right]\cdot e^{-j\cdot k_1\cdot u_1} \quad . \tag{2.73}$$

For compatibility with eq. 2.62, I_2 is divided by 3 and the inner. Note that the author believes there is a mistake in eq. A.6 in Rodden et al. 1971 [14], which is correcte in the following way: brackets are added for term $(1.0+u_1^2)^{0.5}$ and the inner, closing square brackets is put after the fraction. The integrals $I_{1,2}$ are only evaluated for positive values of u_1. When $u_1 < 0.0$, then (compare eq. A.5 and A.9 in Rodden et al. 1971 [14])

$$I_1(u_1<0.0) = 2.0\cdot \mathrm{Re}(I_1(0.0,k_1)) - \mathrm{Re}(I_1(-u_1,k_1)) + j\cdot \mathrm{Im}(I_1(-u_1,k_1)) \tag{2.74}$$

and

$$I_2(u_1<0.0) = 2.0\cdot \mathrm{Re}(I_2(0.0,k_1)) - \mathrm{Re}(I_2(-u_1,k_1)) + j\cdot \mathrm{Im}(I_2(-u_1,k_1)) \, . \tag{2.75}$$

Desmarais [3] proposed a similar approach but with higher accuracy

$$1 - \frac{u_1}{(1+u_1)^{0.5}} = \sum_{n=1}^{12} a_n \cdot e^{-2.0^{n/m}\cdot b\cdot u_1}, \tag{2.76}$$

where $b = 0.009054814793$, $m = 1.0$ and the coefficients a_n are as given in Table 2.3. Note that Desmarais investigated a number of different approximations in his work, the selected approach is referred to as D12.1.

n	a_n
1	0.000319759140
2	-0.000055461471
3	0.002726074362
4	0.005749551566
5	0.031455895072
6	0.106031126212
7	0.406838011567
8	0.798112357155
9	-0.417749229098
10	0.077480713894
11	-0.012677284771
12	0.001787032960

Table 2.3: Coefficients used in the Desmarais approximation

That leads to a modification of terms

$$I_0(u_1,k_1) = \sum_{n=1}^{12} \frac{a_n \cdot e^{-2.0^{n/m} \cdot b \cdot u_1}}{(2.0^{n/m})^2 \cdot b^2 + k_1^2} \cdot [2.0^{n/m} \cdot b - j \cdot k_1] \tag{2.77}$$

and

$$\begin{aligned} J_0(u_1,k_1) = &\sum_{n=1}^{12} \frac{a_n \cdot e^{-2.0^{n/m} \cdot b \cdot u_1}}{((2.0^{n/m})^2 \cdot b^2 + k_1^2)^2} \cdot \Big[(2.0^{n/m})^2 \cdot b^2 - k_1^2 \\ &+ 2.0^{n/m} \cdot b \cdot u_1 \cdot ((2.0^{n/m})^2 \cdot b^2 + k_1^2) \qquad . \\ &- j \cdot k_1 \cdot (2.0 \cdot 2.0^{n/m} \cdot b + u_1 \cdot ((2.0^{n/m})^2 \cdot b^2 + k_1^2) \Big] \end{aligned} \tag{2.78}$$

2.4 Performance Considerations

The code is vectorized and uses Numpy matrix operations. Compilation of the Python code using Cython or Numba showed no significant increase in speed. The most time-consuming part is the Laschka and/or the Desmarais approximations where a series of 11 or 12 terms are evaluated repeatedly.

3 Validation

In this section, the implementation of the VLM and the DLM is validated with respect to MSC.Nastran. The AIC matrices are exported from MSC.Nastran SOL145 via a so-called DMAP alter in the OP4 format and parsed again with an OP4 reader in Python in a next step. The command NASTRAN QUARTICDLM=0/1 is used to switch between the parabolic and the quartic DLM. Note that the default value is the parabolic version.

The comparison is performed in two steps. First, the AIC matrices are compared element-wise for numerical equality using the numpy command numpy.allclose(a,b). If the equation

$$|a-b| \leq (\mathrm{atol} + \mathrm{rtol} \cdot |b|) \tag{3.1}$$

is element-wise True, then allclose returns True [23]. A relative tolerance of $\mathrm{rtol} = 10^{-3}$ in combination with an absolute tolerance of $\mathrm{atol} = 10^{-4}$ are considered sufficient to assume numerical equality of the AIC matrices.

In a next step, the pressure distributions are compared. Therefore, a downwash vector $\mathbf{w}_j$ is assumed that corresponds to a 5° onflow normal to each panel. The following plots show the difference in pressure

$$\Delta\Delta\mathbf{c}_p = \mathbf{AIC}^{\mathrm{Nastran}}(Ma,k) \cdot \mathbf{w}_j j - \mathbf{AIC}^{\mathrm{LK}}(Ma,k) \cdot \mathbf{w}_j \tag{3.2}$$

for the real part and the imaginary part of$\Delta\Delta\mathbf{c}_p$ in a blue-white-red color map.

3.1 The Allegra Configuration

The Allegra configuration has a forward swept wing with positive dihedral and a T-tail with backward sweep and negative dihedral. There are no winglets but a vertical tail plane and some additional panels to account for the fuselage. The aircraft has a left and a right hand side, which are symmetrical with respect to the xz-plane. Therefore, the configuration includes most features of typical aircraft and is a comprehensive and representative test case. The tests have been performed against MSC.Nastran as described above for both the parabolic and the quartic version of the DLM. Summing up, all test cases were found to be equivalent in a numerical sense.

3.1.1 Typical Pressure Distribution

The 5° onflow condition results in the following pressure distributions $\Delta \mathbf{c}_p$. The real and imaginary parts are shown in Figures 3.1 and 3.2 respectively. The following sections will show only the differences $\Delta\Delta \mathbf{c}_p$.

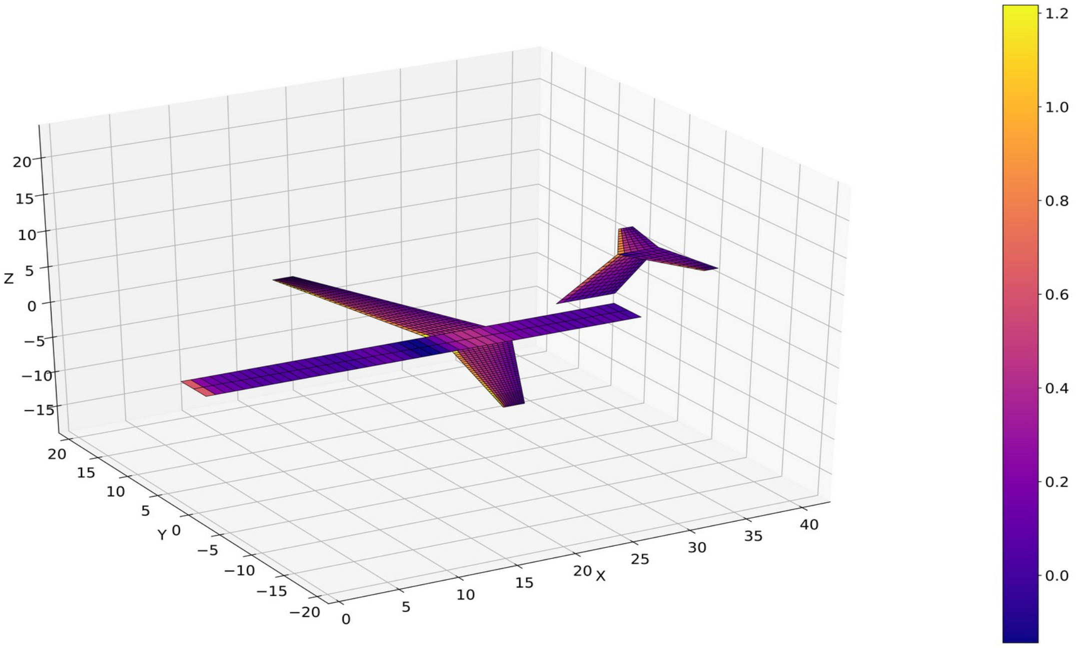

Figure 3.1: Real part of ΔCp at Ma=0.8, k$_{red}$*=0.6 with a 5° onflow*

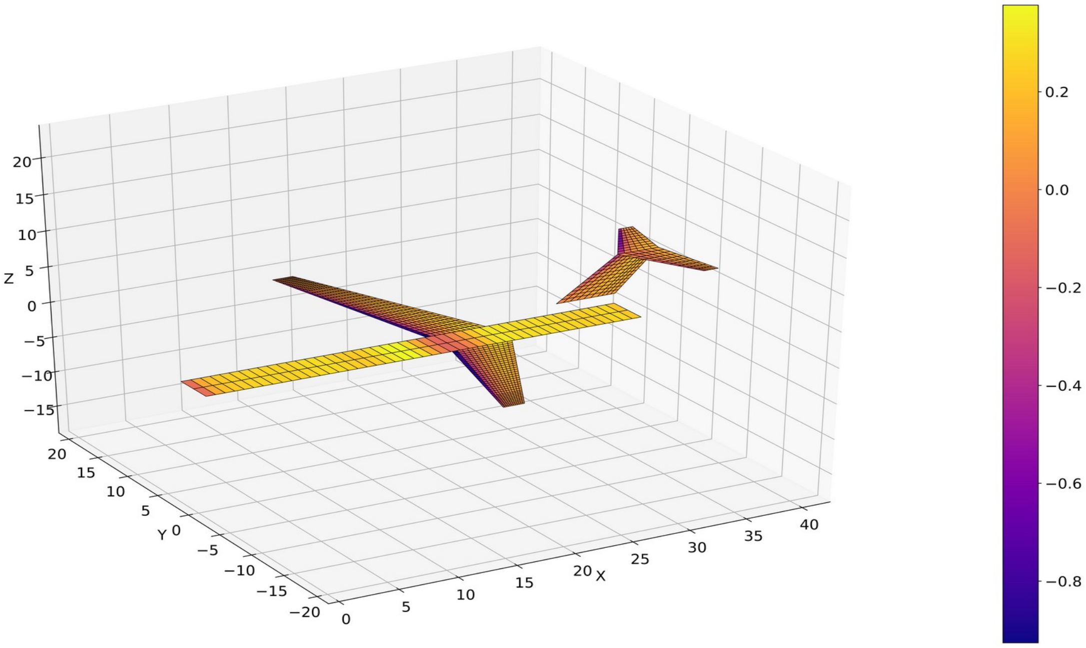

Figure 3.2: Imaginary part of ΔCp at Ma=0.8, k$_{red}$*=0.6 with a 5° onflow*

3.1.2 Parabolic DLM, Ma = 0.8, k_{red} = 0.001

Element-wise comparison of the AIC matrices: numerically equal. ✓

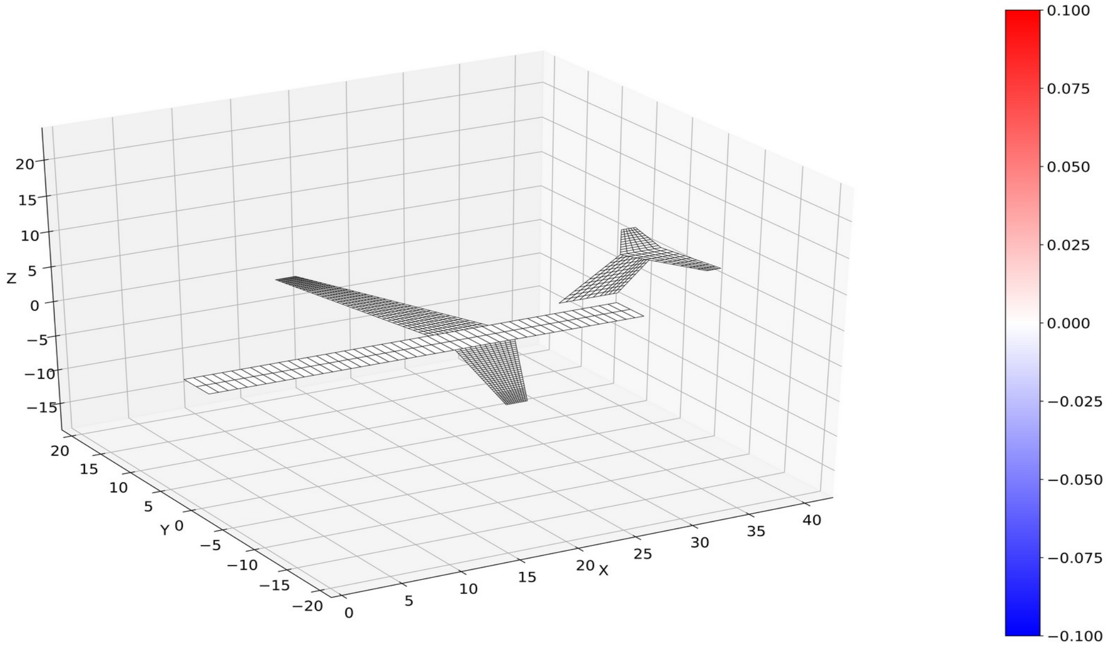

Figure 3.3: Real part of ΔΔCp at Ma=0.8, k_{red}=0.001 with a 5° onflow, parabolic DLM

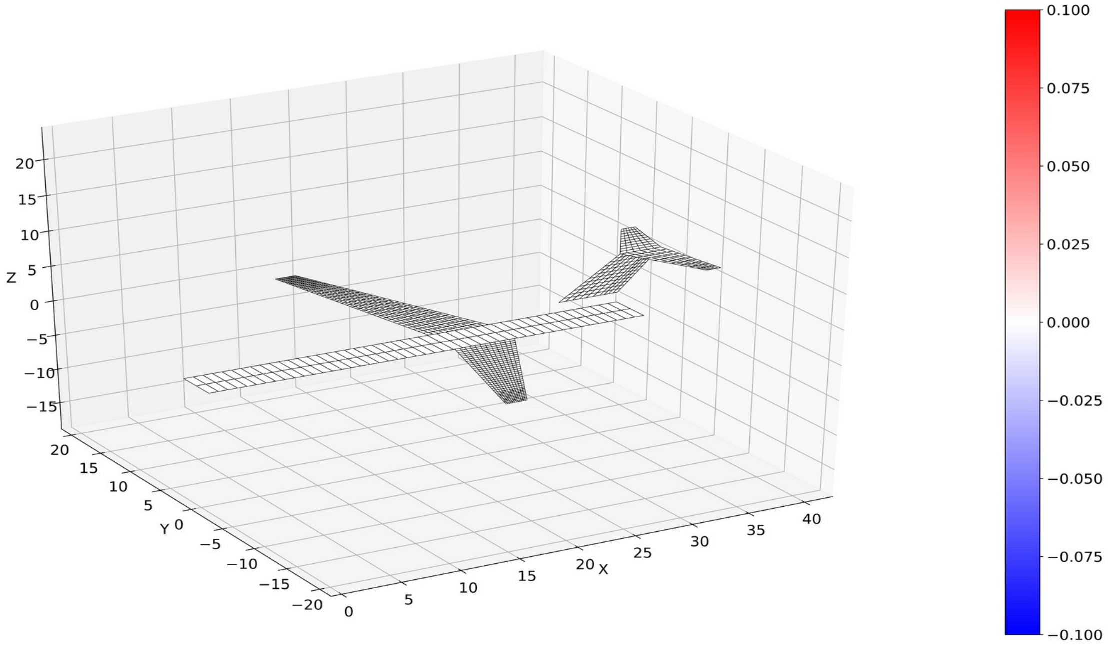

Figure 3.4: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=0.001 with a 5° onflow, parabolic DLM

3.1.3 Parabolic DLM, Ma = 0.8, k_{red} = 0.6

Element-wise comparison of the AIC matrices: numerically equal. ✓

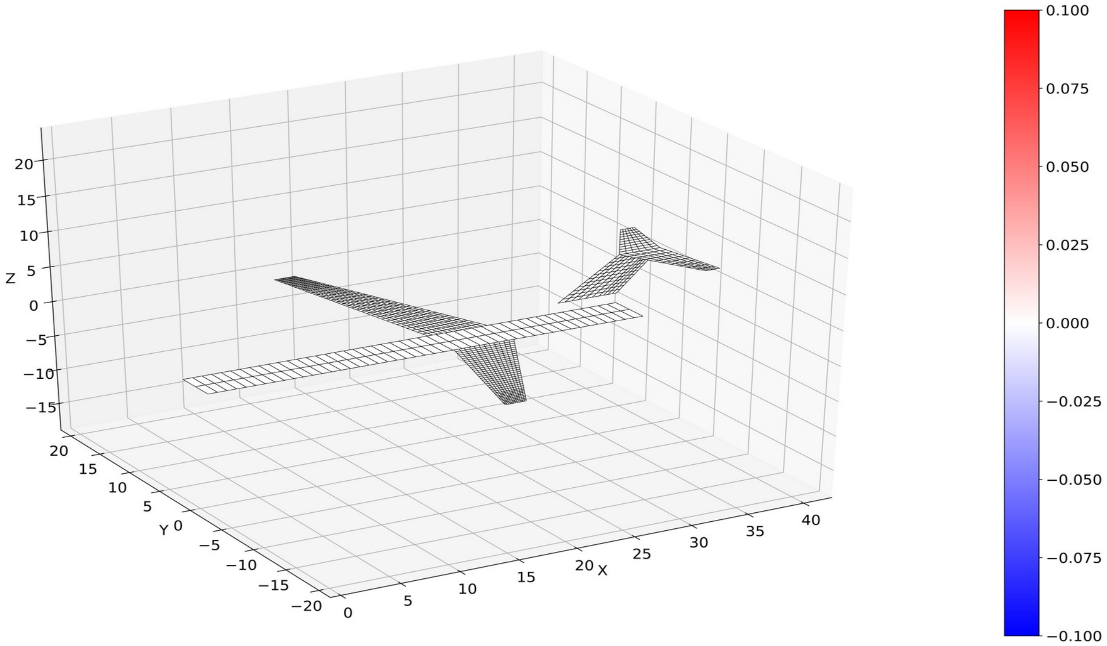

Figure 3.5: Real part of ΔΔCp at Ma=0.8, k_{red}=0.6 with a 5° onflow, parabolic DLM

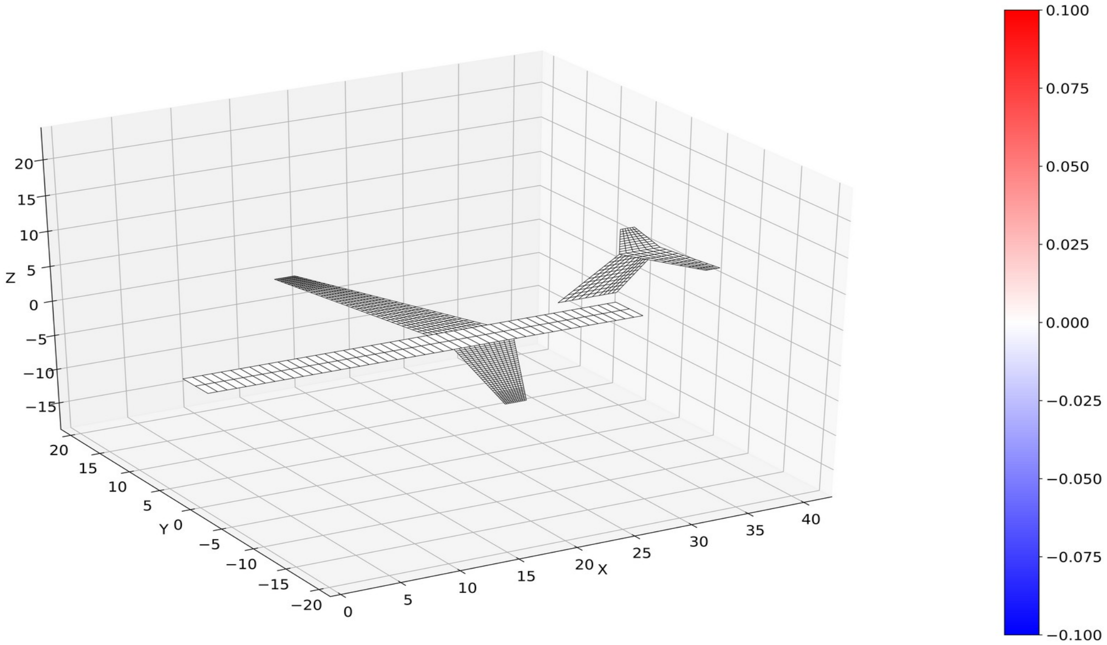

Figure 3.6: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=0.6 with a 5° onflow, parabolic DLM

3.1.4 Parabolic DLM, Ma = 0.8, k_{red} = 1.4

Element-wise comparison of the AIC matrices: numerically equal. ✓

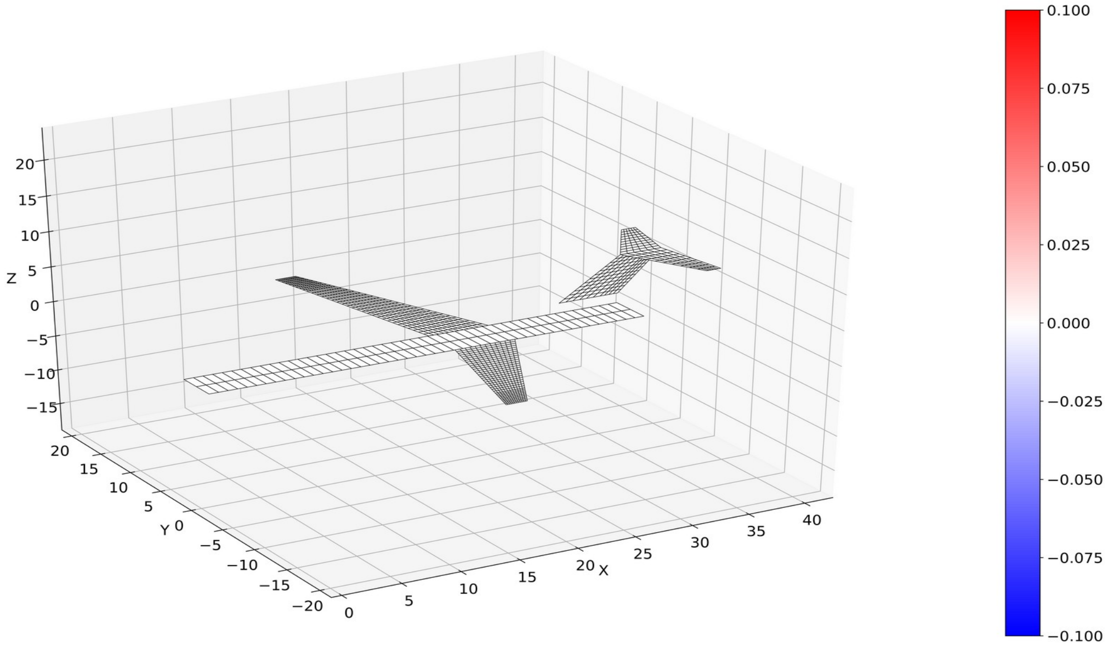

Figure 3.7: Real part of ΔΔCp at Ma=0.8, k_{red}=1.4 with a 5° onflow, parabolic DLM

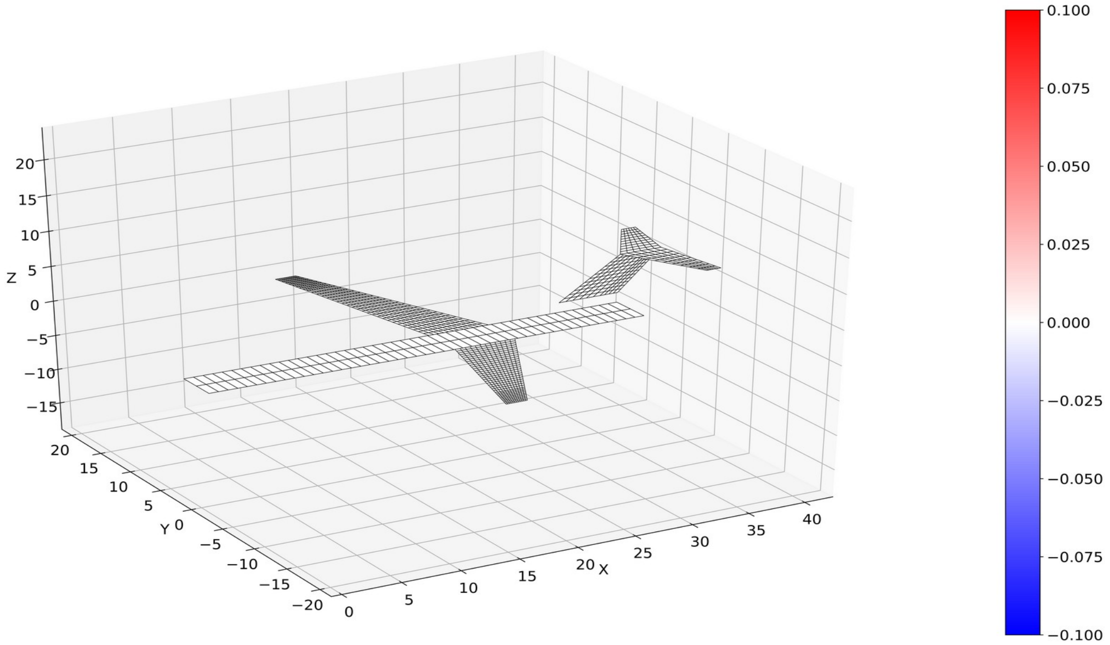

Figure 3.8: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=1.4 with a 5° onflow, parabolic DLM

3.1.5 Quartic DLM, Ma = 0.8, k_{red} = 0.001

Element-wise comparison of the AIC matrices: numerically equal. ✓

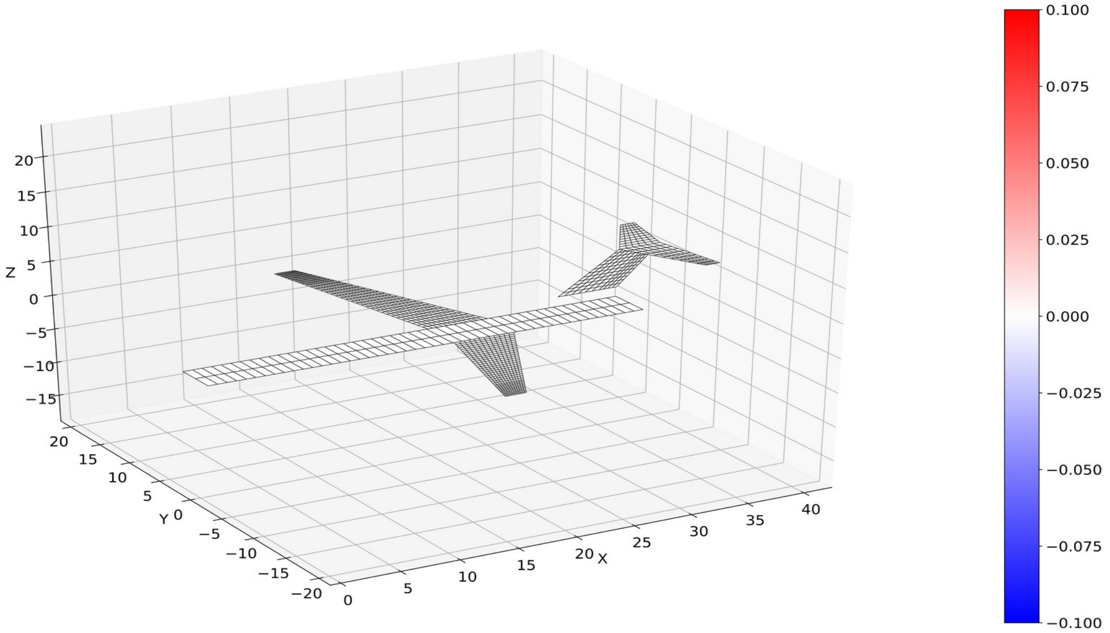

Figure 3.9: Real part of ΔΔCp at Ma=0.8, k_{red}=0.001 with a 5° onflow, quartic DLM

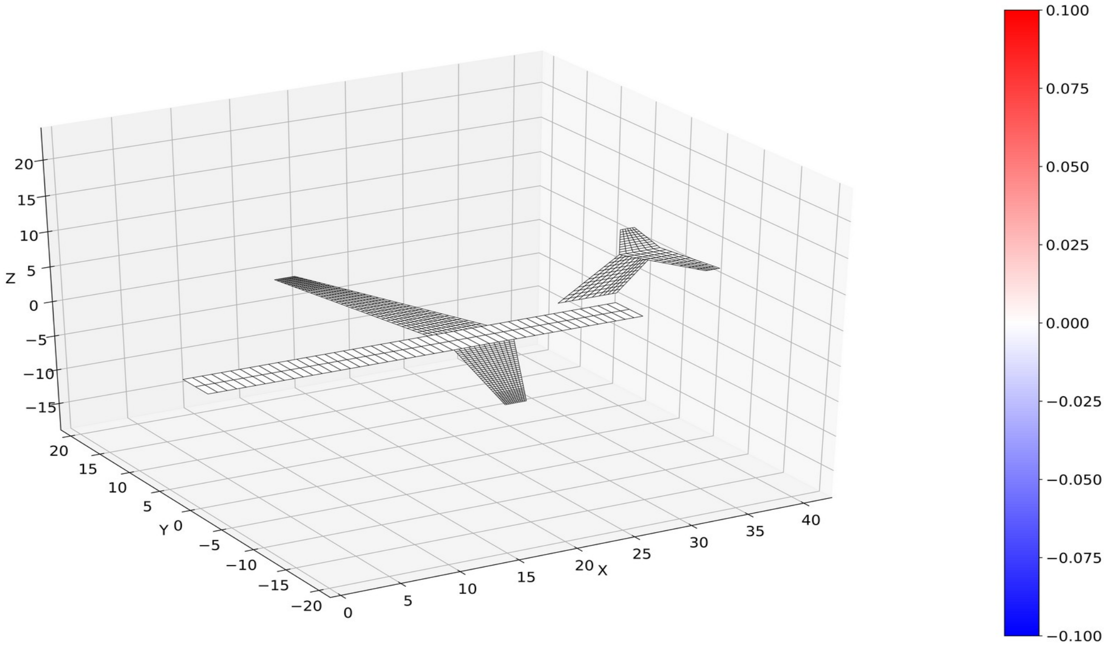

Figure 3.10: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=0.001 with a 5° onflow, quartic DLM

3.1.6 Quartic DLM, Ma = 0.8, k_{red} = 0.6

Element-wise comparison of the AIC matrices: numerically equal. ✓

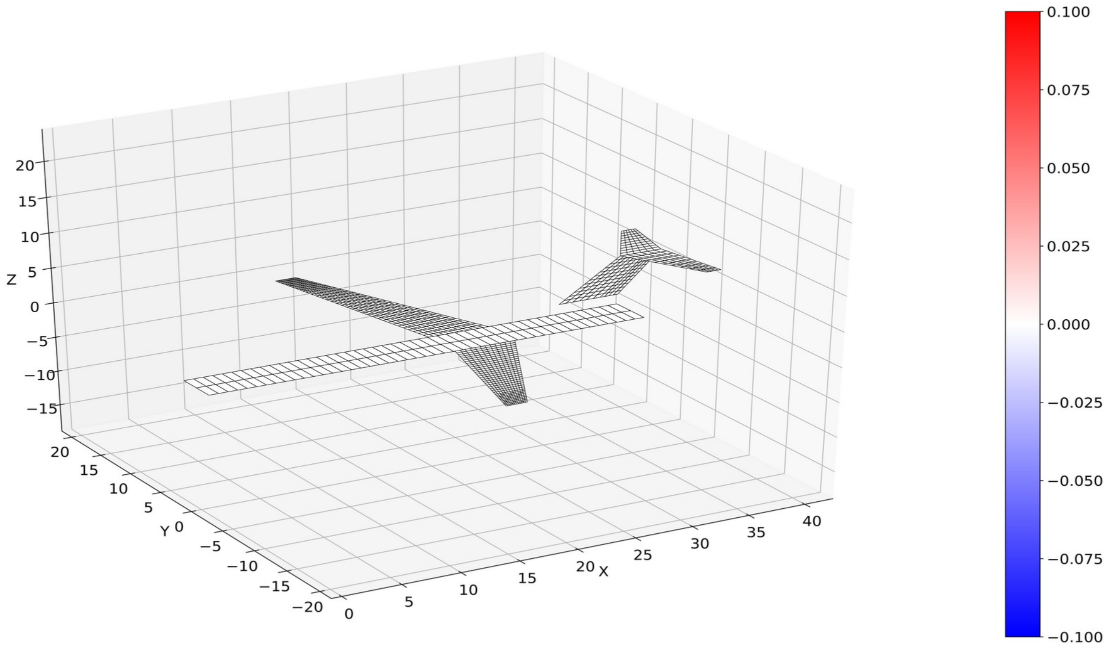

Figure 3.11: Real part of ΔΔCp at Ma=0.8, k_{red}=0.6 with a 5° onflow, quartic DLM

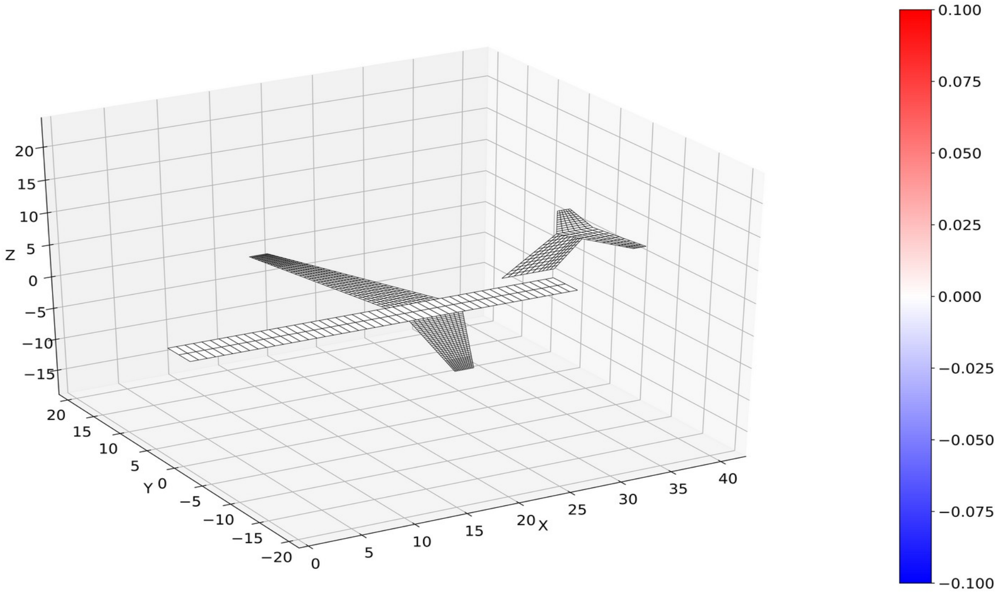

Figure 3.12: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=0.6 with a 5° onflow, quartic DLM

3.1.7 Quartic DLM, Ma = 0.8, k_{red} = 1.4

Element-wise comparison of the AIC matrices: numerically equal. ✓

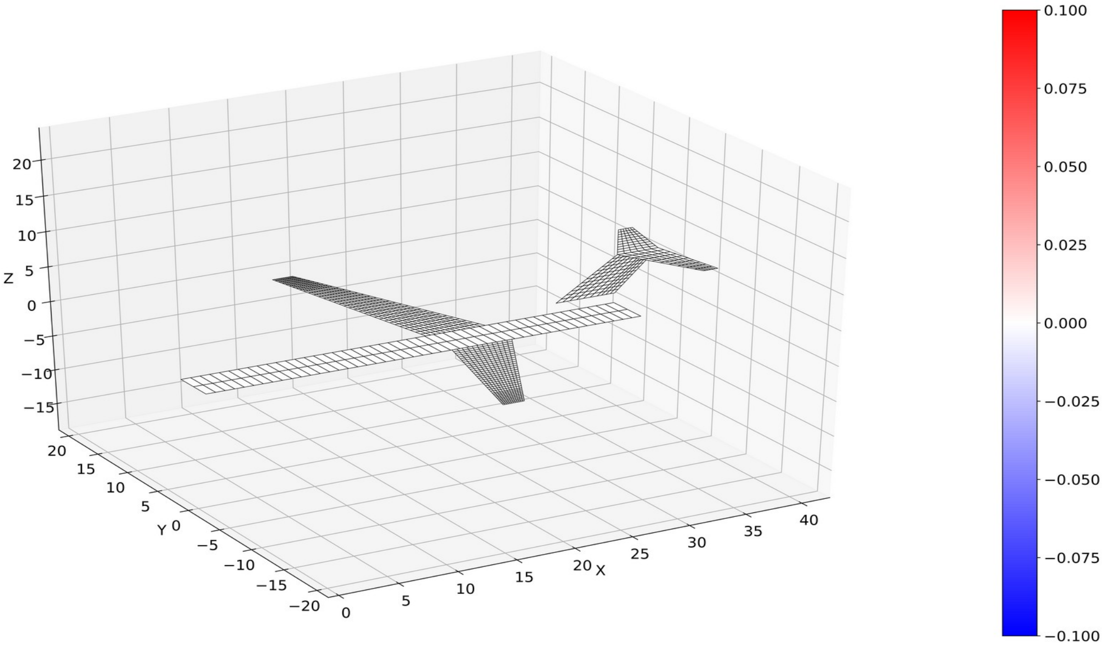

Figure 3.13: Real part of ΔΔCp at Ma=0.8, k_{red}=1.4 with a 5° onflow, quartic DLM

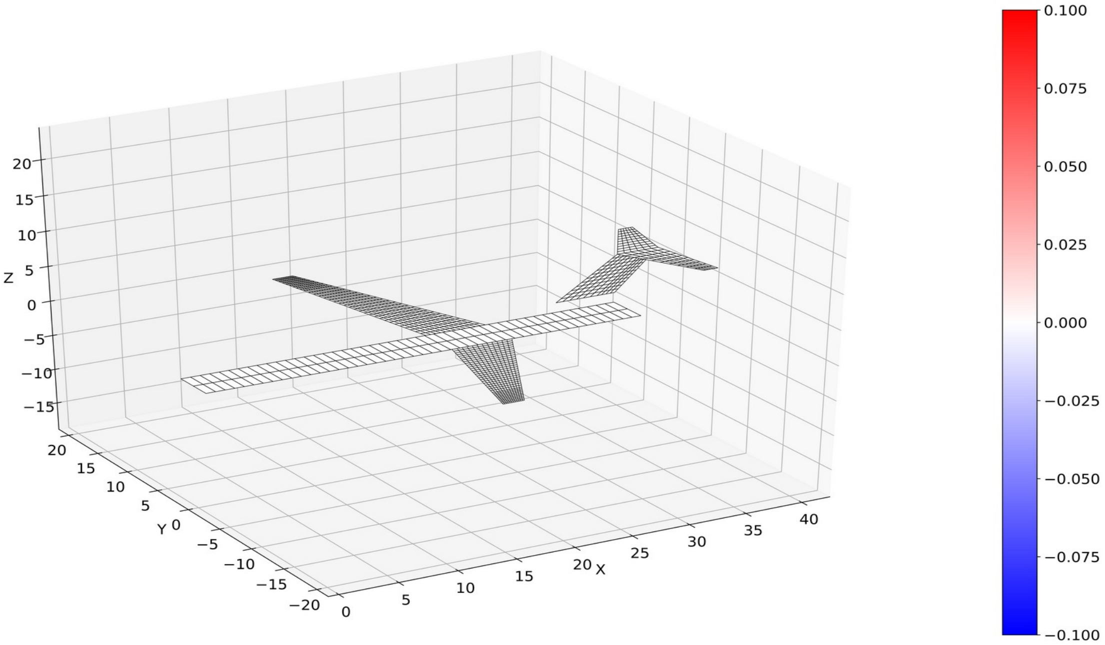

Figure 3.14: Imaginary part of ΔΔCp at Ma=0.8, k_{red}=1.4 with a 5° onflow, quartic DLM

3.2 Special Cases

In the following, some special test cases are inspected. The focus is on wing-empennage configurations with the horizontal tail

- planar to the main wing,
- near-planar and
- further away (e.g. T-tail).

This is an important test to validate all conditions as described in section 2.3. The offsets in z-direction between wing and horizontal tail plane are dz = 0.0 m, -0.1 m +1.9 m. The wing and the tail have both a span width of 3.0 m and are both discretized with 20 panels in span-wise direction so that the panels of wing and tail are aligned in y-direction. Finally, a (comparatively large) winglet is added to the wing because the previous example of the Allegra configuration had no winglets. The AICs are calculated at Ma=0.5, k_{red}=2.0 and with a 5° onflow condition.

3.2.1 Parabolic DLM, horizontal tail planar to the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

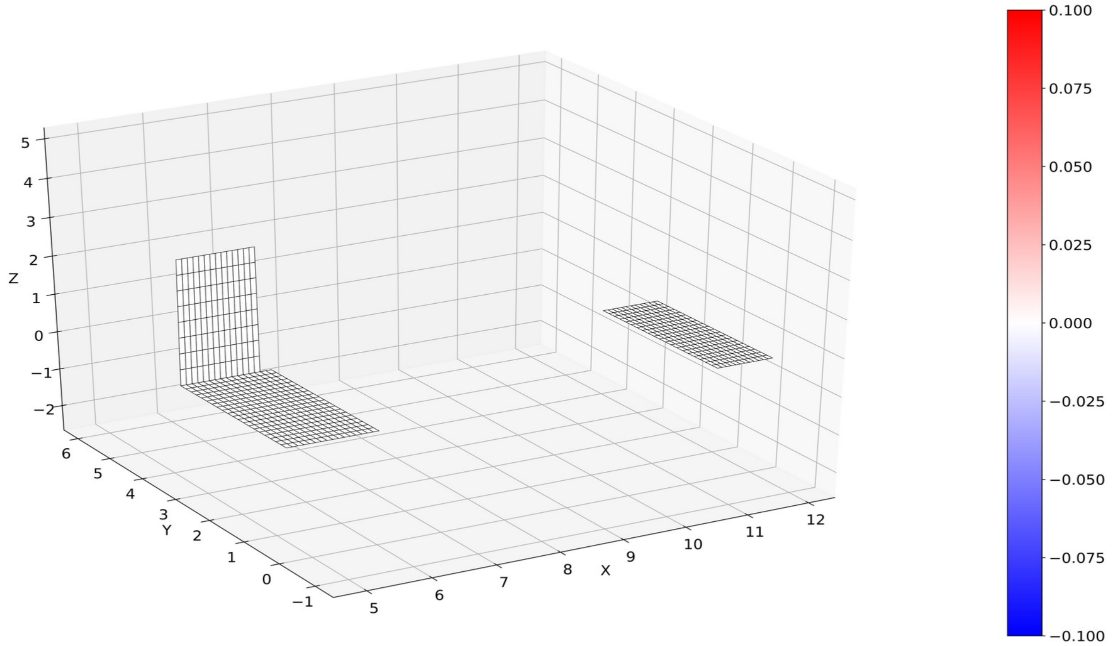

Figure 3.15: Real part of ΔΔCp, horizontal tail planar to the wing, parabolic DLM

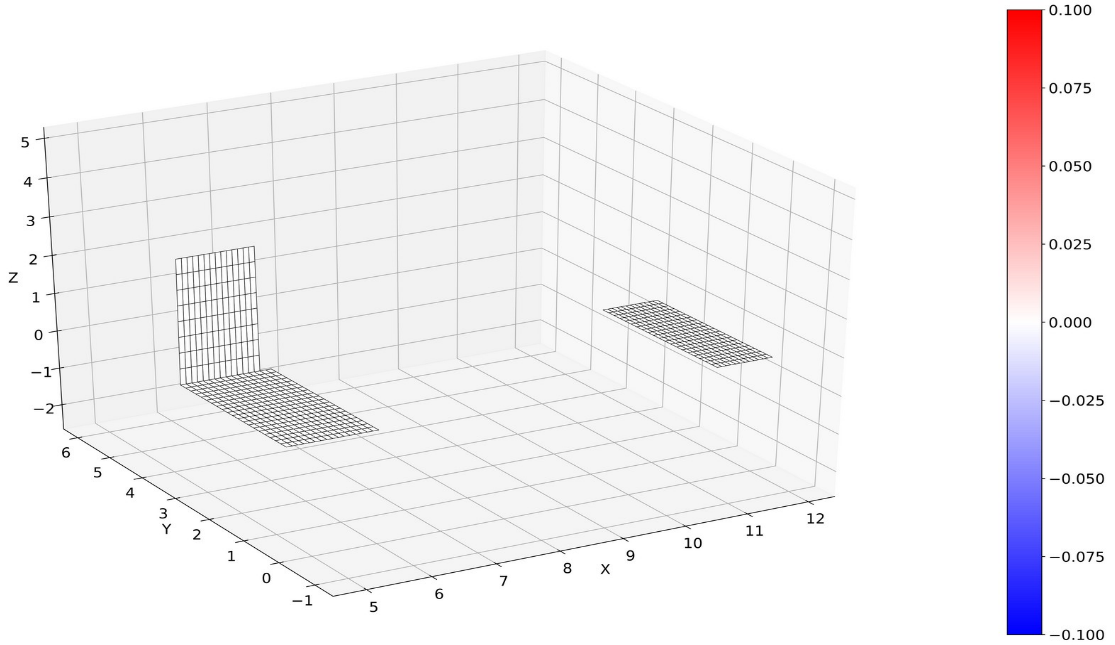

Figure 3.16: Imaginary part of ΔΔCp, horizontal tail planar to the wing, parabolic DLM

3.2.2 Parabolic DLM, horizontal tail near-planar to the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

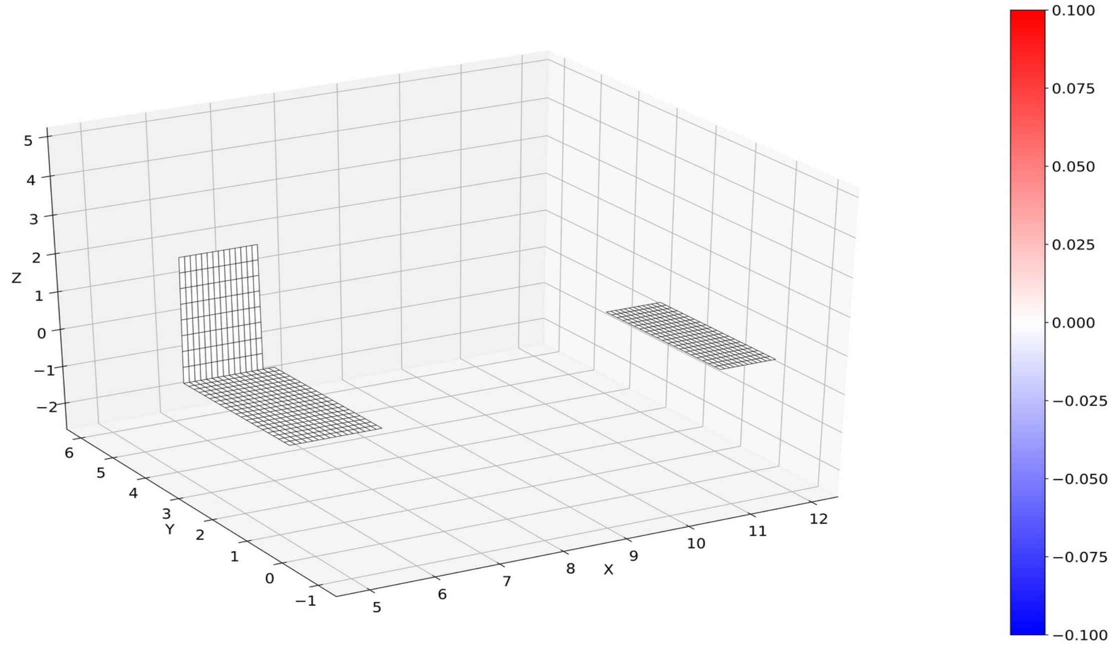

Figure 3.17: Real part of ΔΔCp, horizontal tail near-planar to the wing, parabolic DLM

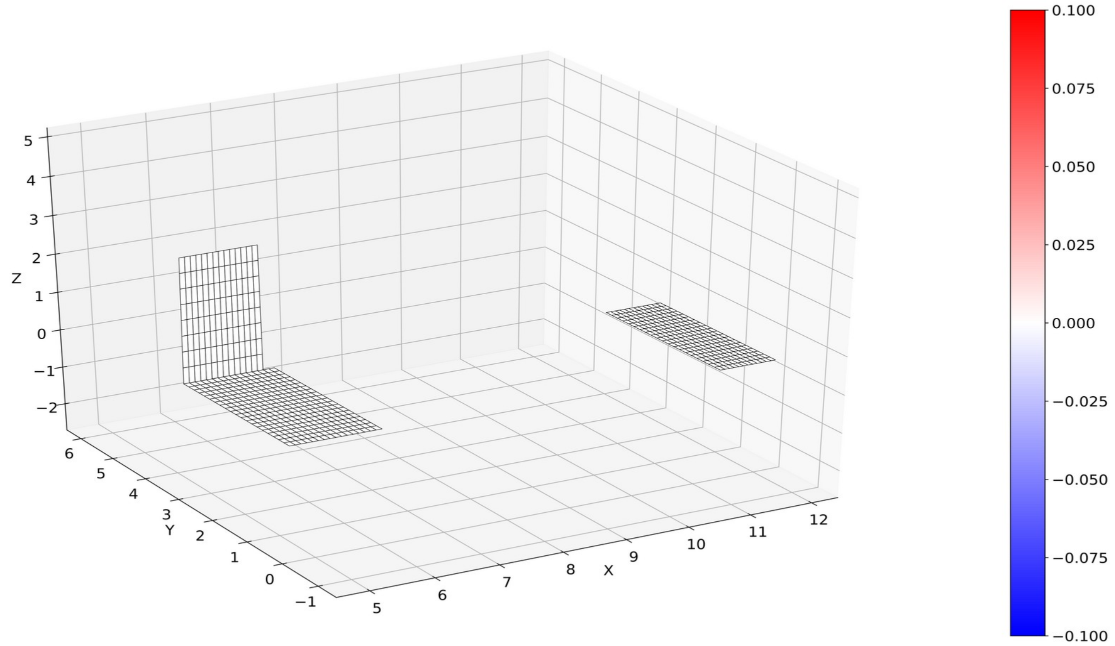

Figure 3.18: Imaginary part of ΔΔCp, horizontal tail near-planar to the wing, parabolic DLM

3.2.3 Parabolic DLM, horizontal tail further away from the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

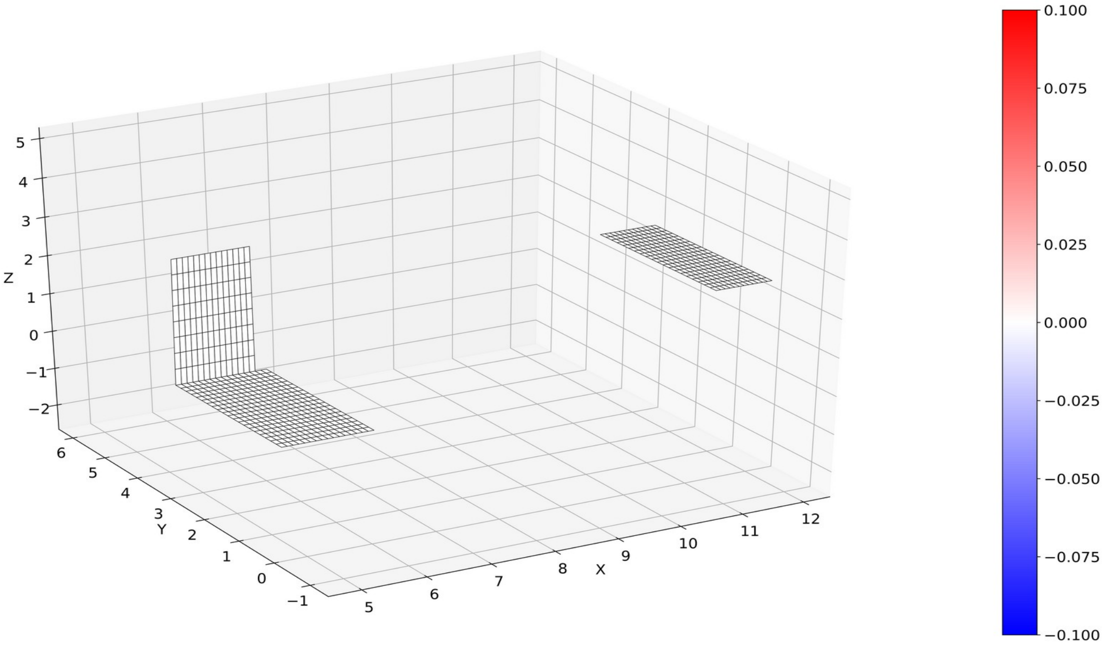

Figure 3.19: Real part of ΔΔCp, horizontal tail further away from the wing, parabolic DLM

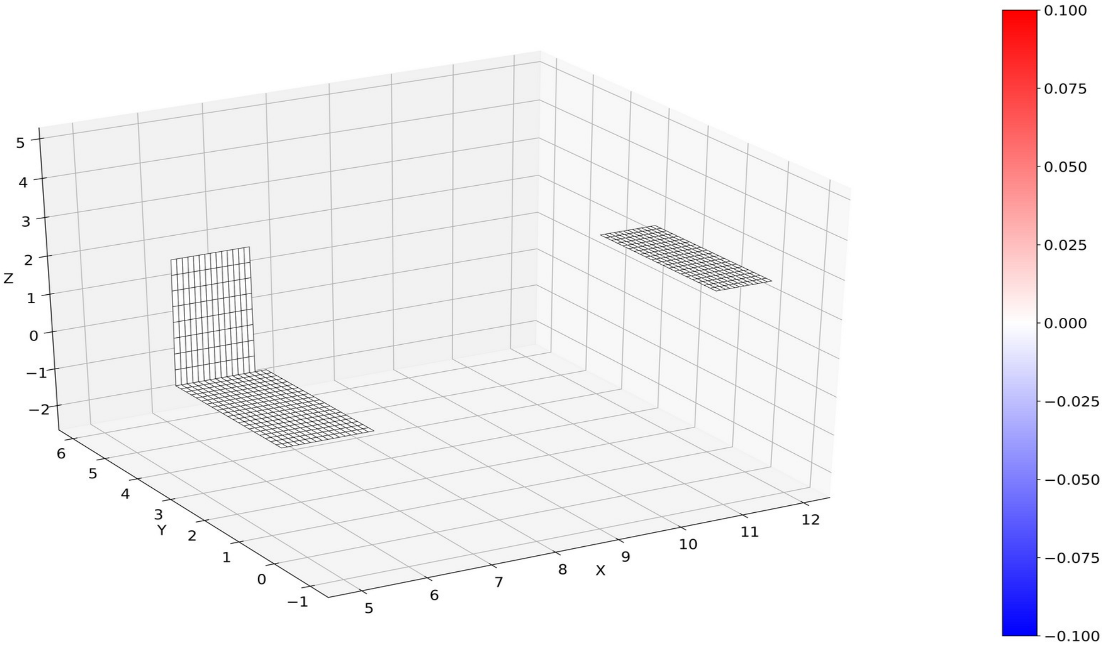

Figure 3.20: Imaginary part of ΔΔCp, horizontal tail further away from the wing, parabolic DLM

3.2.4 Quartic DLM, horizontal tail planar to the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

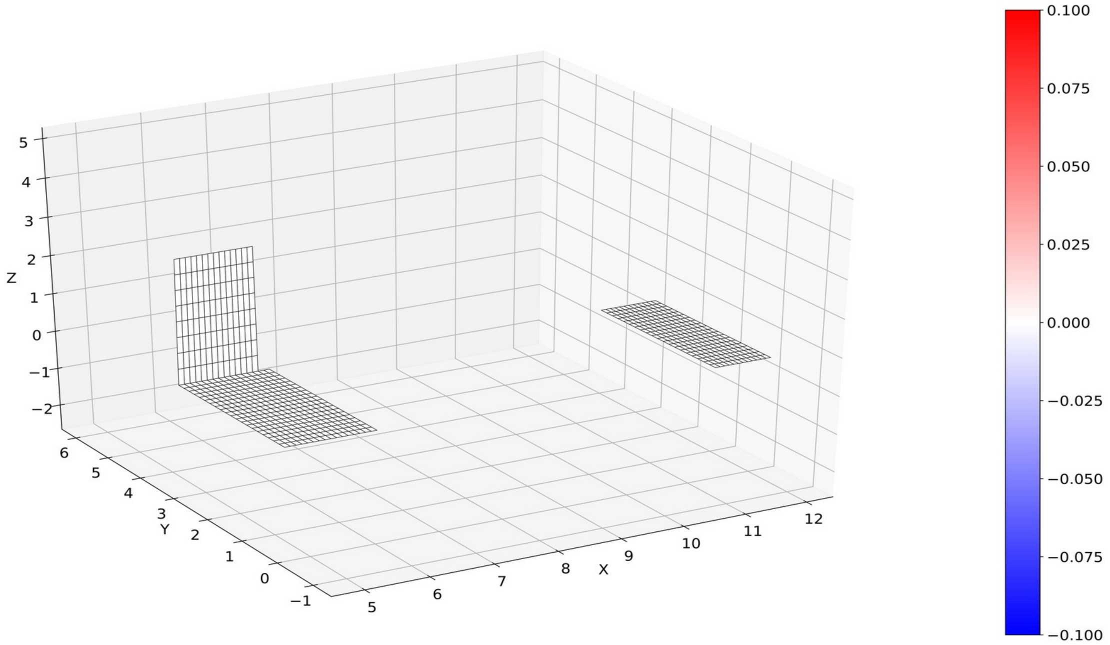

Figure 3.21: Real part of ΔΔCp, horizontal tail planar to the wing, quartic DLM

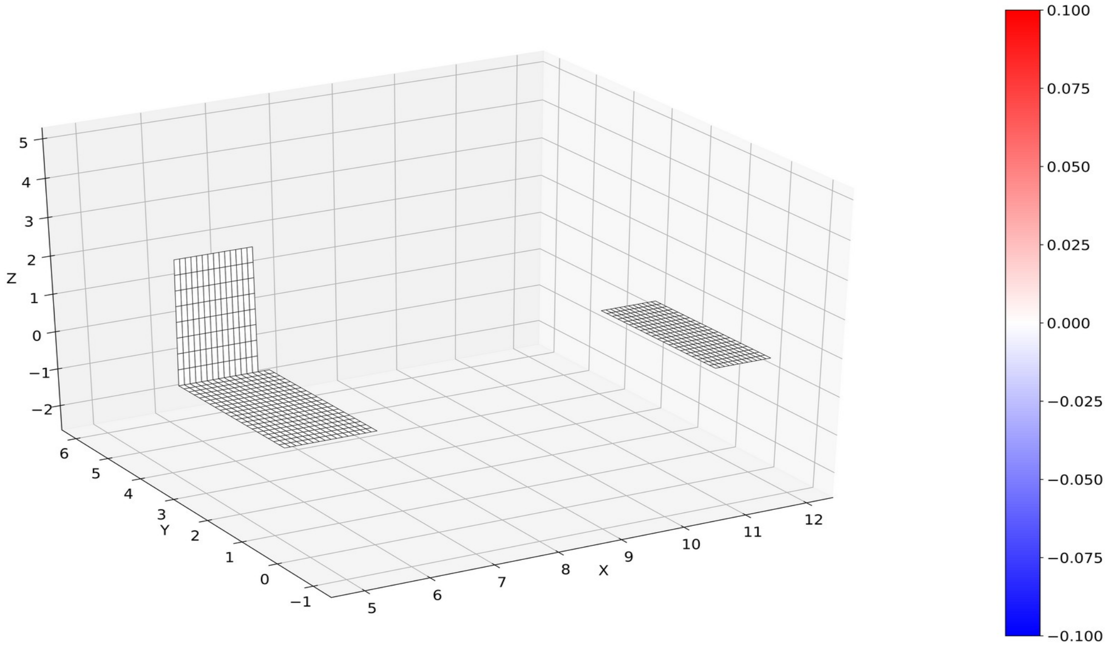

Figure 3.22: Imaginary part of ΔΔCp, horizontal tail planar to the wing, quartic DLM

3.2.5 Quartic DLM, horizontal tail near-planar to the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

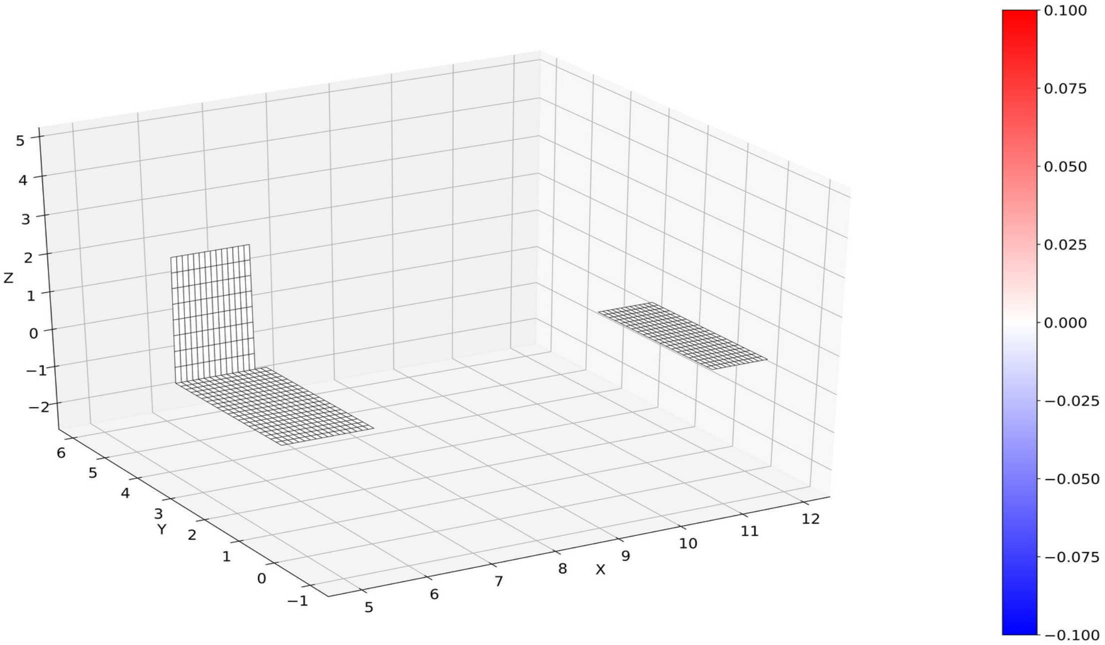

Figure 3.23: Real part of ΔΔCp, horizontal tail near-planar to the wing, quartic DLM

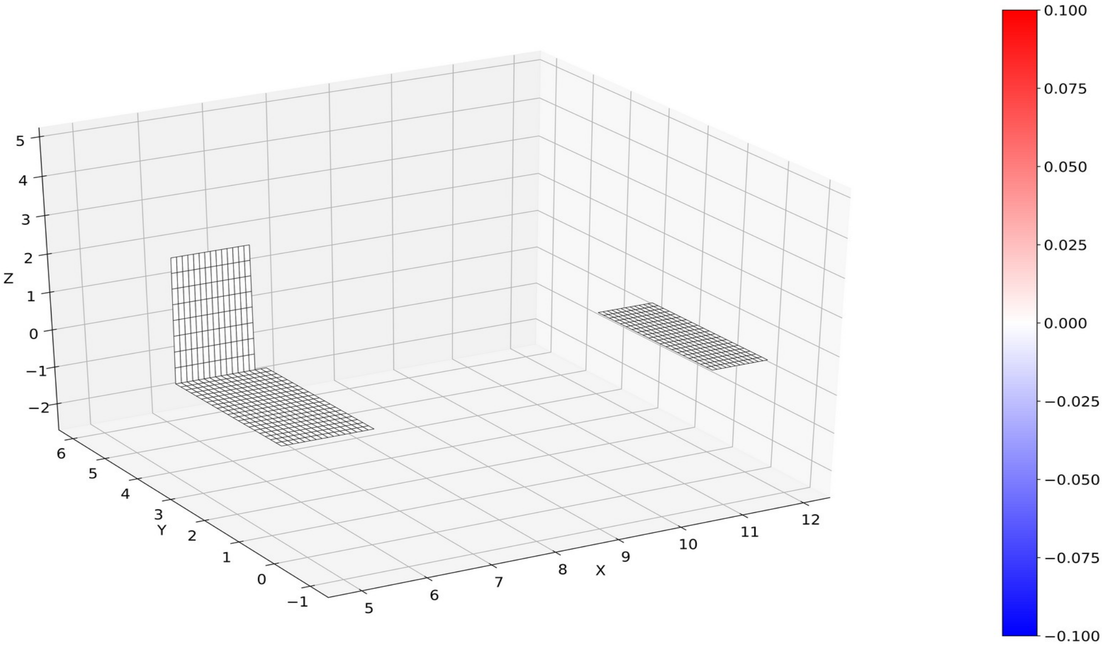

Figure 3.24: Imaginary part of ΔΔCp, horizontal tail near-planar to the wing, quartic DLM

3.2.6 Quartic DLM, horizontal tail further away from the wing

Element-wise comparison of the AIC matrices: numerically equal. ✓

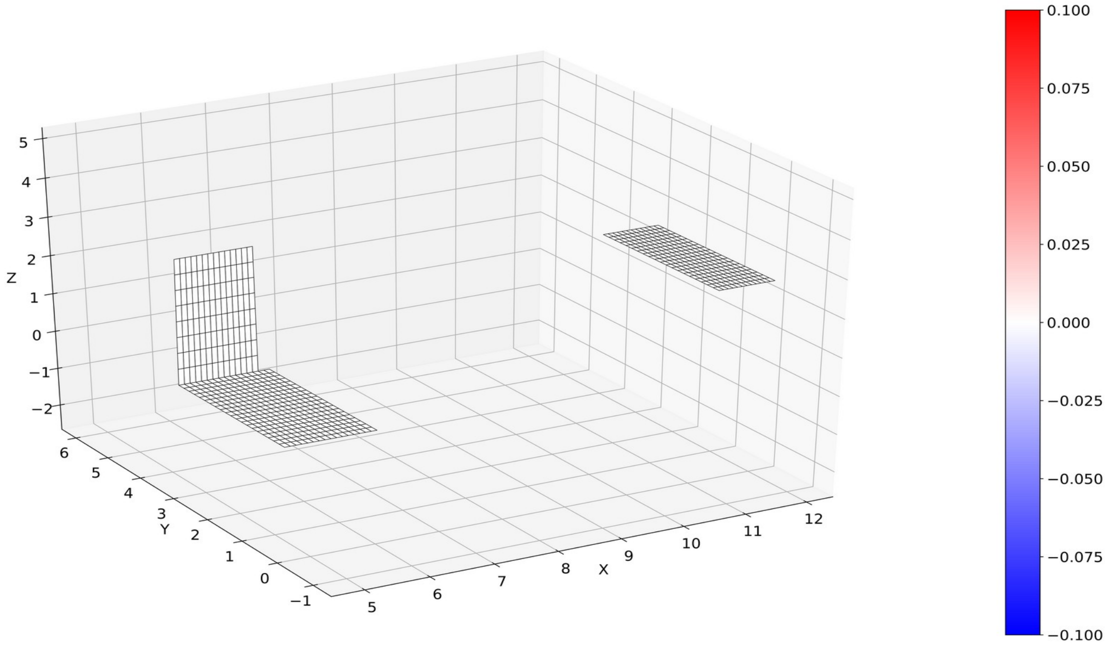

Figure 3.25: Real part of ΔΔCp, horizontal tail further away from the wing, quartic DLM

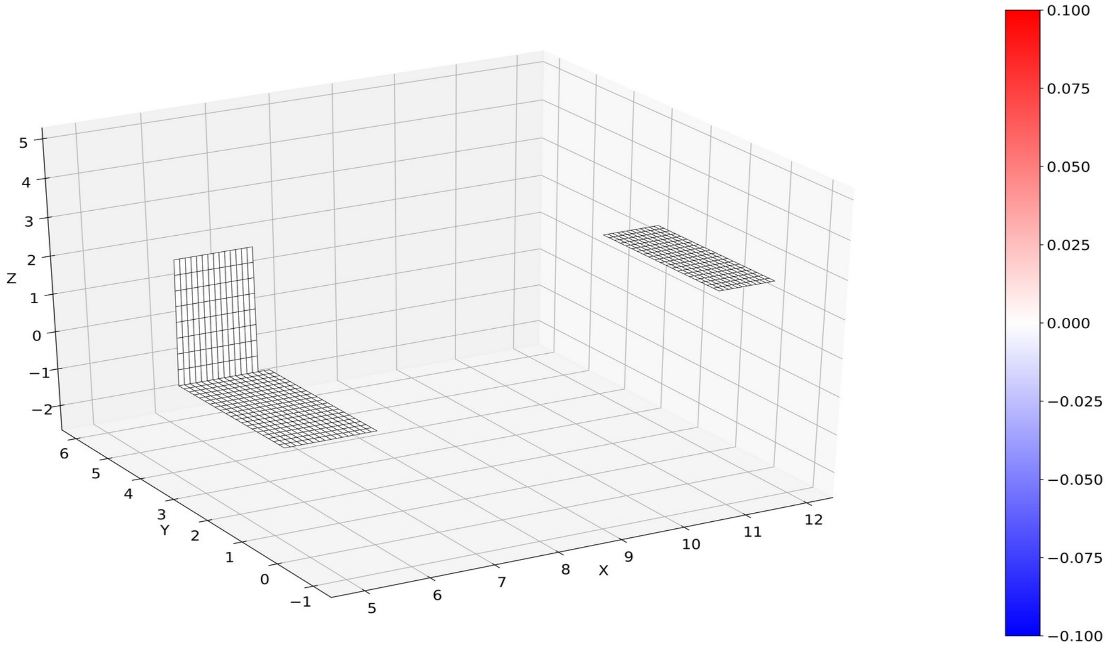

Figure 3.26: Imaginary part of ΔΔCp, horizontal tail further away from the wing, quartic DLM

3.2.7 Provoking differences and errors by misalignment of panels

Remember that the wing and the tail have both a span width of 3.0 m and are both discretized with 20 panels in span-wise direction so that the panels of wing and tail are aligned in y-direction. Rodden et al. [15] state that “One of the basic requirements of the Doublet Lattice Method is that stream-wise strip edges must be aligned for all surfaces in the same or nearly the same plane.” **Violating this requirement** for the near-planar case, e.g. by selecting only 8 panels for the wing but maintaining 20 for the tail, results in differences with respect to MSC.Nastran on several panels at the tail. However, both results should be wrong according to Rodden. Note that a misalignment in the planar case even results in singular AICs matrices in MSC.Nastran as well as in this implementation.

Element-wise comparison of the AIC matrices: NOT equal.

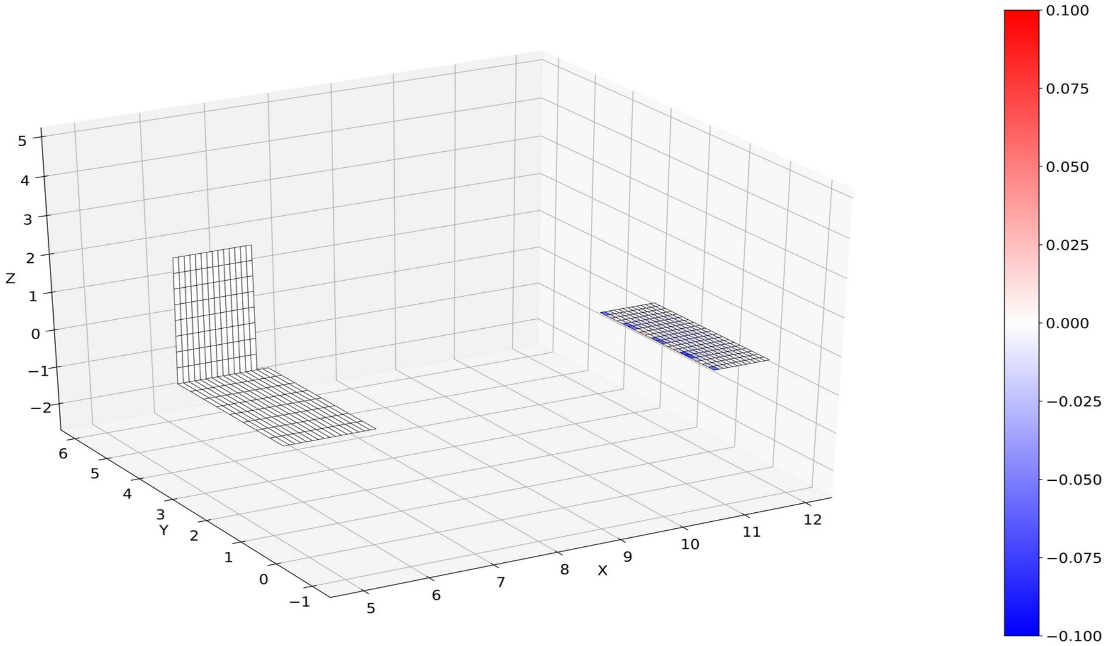

Figure 3.27: Real part of ΔΔCp, horizontal tail near-planar to the wing, quartic DLM, violated discretization requirements

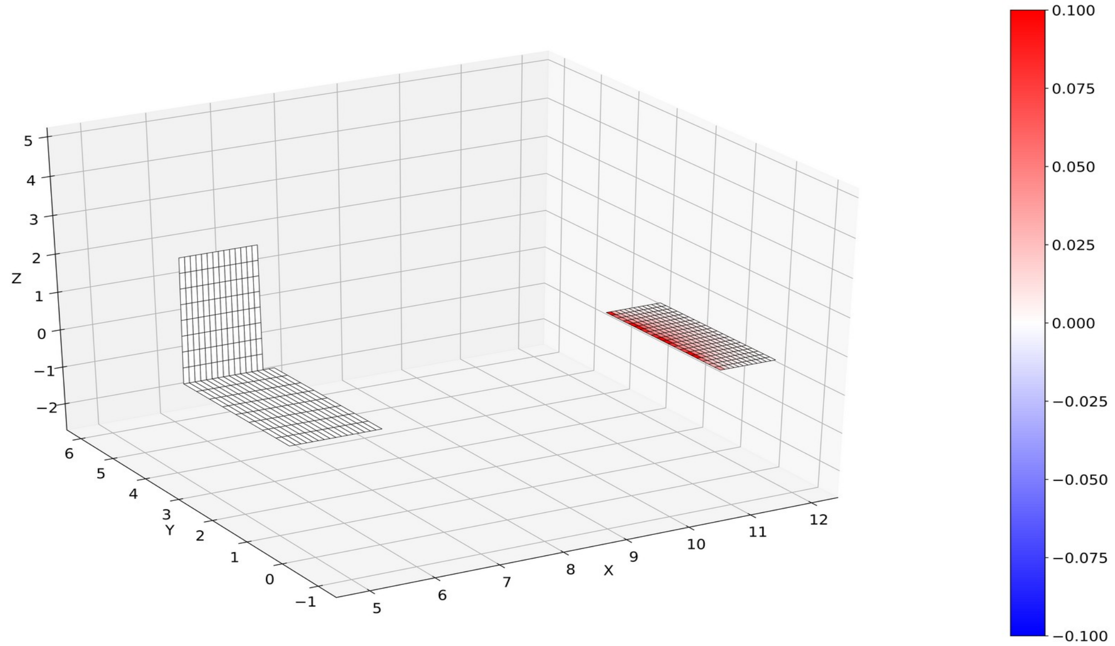

Figure 3.28: Imaginary part of ΔΔCp, horizontal tail planar to the wing, quartic DLM, violated discretization requirements

Bibliography

[1] Albano, E., and Rodden, W. P., “A Doublet Lattice Method For Calculating Lift Distributions on Oscillation Surfaces in Subsonic Flows,” in *AIAA 6th Aerospace Sciences Meeting*, New York, 1968.

[2] Blair, M., “A Compilation of the Mathematics Leading to The Doublet Lattice Method,” Airforce Wright Laboratory, Ohio, Technical Report WL-TR-92-3028, 1992.

[3] Desmarais, R., “An accurate and efficient method for evaluating the kernel of the integral equation relating pressure to normalwash in unsteady potentialflow,” presented at the 23rd Structures, Structural Dynamics and Materials Conference, New Orleans, LA, 1982, https://doi.org/10.2514/6.1982-687.

[4] Desmarais, R. N., “An accurate method for evaluating the kernel of the integral equation relating lift to downwash in unsteady potential flow,” Hampton, Virginia, NASA-TM-83281, Apr. 1982, https://ntrs.nasa.gov/search.jsp?R=19820015320.

[5] Hedman, S. G., “Vortex Lattice Method for Calculation of Quasi Steady State Loadings on Thin Elastic Wings in Subsonic Flow,” FFA Flygtekniska Försöksanstalten, Stockholm, Sweden, FFA Report 105, 1966.

[6] Katz, J., and Plotkin, A., *Low-speed aerodynamics: from wing theory to panel methods*. New York: McGraw-Hill, 1991.

[7] Kotikalpudi, A., “Body Freedom Flutter (BFF) Doublet Lattice Method (DLM),” *University of Minnesota Digital Conservancy*, 09-Sep-2014. [Online]. Available: http://hdl.handle.net/11299/165566. [Accessed: 12-Feb-2016].

[8] Kotikalpudi, A., Pfifer, H., and Balas, G. J., “Unsteady Aerodynamics Modeling for a Flexible Unmanned Air Vehicle,” in *AIAA Atmospheric Flight Mechanics Conference*, Dallas, Texas, 2015, https://doi.org/10.2514/6.2015-2854.

[9] Küssner, H. G., “Allgemeine Tragflächentheorie,” *Luftfahrtforschung*, vol. 17, no. 11/12, pp. 370–78, Dec. 1940, http://ntrs.nasa.gov/search.jsp?R=19930094437.

[10] Küssner, H. G., “General Airfoil Theory,” National Advisory Committee for Aeronautics, Washington, Technical Memorandum NACA-TM-979, Jun. 1941, http://ntrs.nasa.gov/search.jsp?R=19930094437.

[11] Landahl, M. T., “Kernel Function for Nonplanar Oscillating Surfaces in a Subsonic Flow,” *AIAA Journal*, vol. 5, no. 5, pp. 1045–1046, May 1967, https://doi.org/10.2514/3.55319.

[12] Laschka, B., “Das Potential und das Geschwindigkeitsfeld der harmonisch schwingenden tragenden Fläche bei Unterschallströmung,” *Z. angew. Math. Mech.*, vol. 43, no. 7–8, pp. 325–333, 1963, https://doi.org/10.1002/zamm.19630430704.

[13] Prandtl, L., “Beitrag zur Theorie der tragenden Fläche,” *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 16, no. 6, pp. 360–361, 1936, https://doi.org/10.1002/zamm.19360160613.

[14] Rodden, W. P., Giesing, J. P., and Kálmán, T. P., “New Developments and Application of the Subsonic Doublet-Lattice Method for Nonplanar Configurations,” *AGARD-CP-80-PT-2 - Symposium on Unsteady Aerodynamics for Aeroelastic Analyses of Interfering Surfaces*, no. Part 2, Jan. 1971.

[15] Rodden, W. P., Giesing, J. P., and Kalman, T. P., “Refinement of the nonplanar aspects of the subsonic doublet-lattice lifting surface method.,” *Journal of Aircraft*, vol. 9, no. 1, pp. 69–73, Jan. 1972, https://doi.org/10.2514/3.44322.

[16] Rodden, W., Taylor, P., and McIntosh, S., “Further Refinements of the Subsonic Doublet-Lattice Method,” *Journal of Aircraft*, vol. 35, no. 5, pp. 720–727, Oct. 1998.

[17] Rodden, W., Taylor, P., and McIntosh, S., “Improvements to the Doublet-Lattice Method in MSC/Nastran,” 1999.

[18] Voß, A., “Design and Structural Optimization of a Flying Wing of Low Aspect Ratio Based on Flight Loads,” Dissertation, Technische Universität Berlin, Berlin, Germany, 2020, https://doi.org/10.14279/depositonce-9858.

[19] Voß, A., “Loads Kernel User Guide,” Institut für Aeroelastik, Deutsches Zentrum für Luft- und Raumfahrt, Göttingen, Germany, Technical Report DLR-IB-AE-GO-2020-136, Nov. 2021, https://elib.dlr.de/140268/.

[20] Voss, R., “The Legacy of Camillo Possio to Unsteady Aerodynamics,” in *System Modeling and Optimization*, 2005, pp. 1–14, https://doi.org/10.1007/0-387-33006-2_1.

[21] Watkins, C. E. W., “A Systematic Kernel Function Procedure for Determining Aerodynamic Forces on Oscillating or Steady Finite Wings at Subsonic Speeds,” Jan. 1959, https://ntrs.nasa.gov/search.jsp?R=19980227841.

[22] “nasa/NASTRAN-95,” *GitHub*. [Online]. Available: https://github.com/nasa/NASTRAN-95. [Accessed: 04-May-2016].

[23] “numpy.allclose — NumPy v1.18 Manual.” [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.allclose.html. [Accessed: 27-May-2020].