Bilevel high-fidelity aero-structural optimization

A thesis submitted for the degree of
Master of Science

by

Eric Schaar

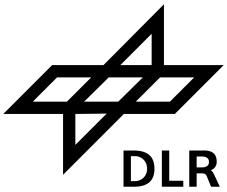
Supervisors:
Fabian Volle
DLR

Dr. Cian Conlan-Smith
Airbus

Dr. Joël Brézillon
Airbus

Examiners:
Prof. Dr. Stefan Görtz
Technische Universität
Braunschweig

Prof. Dr. Ingo Staack
Technische Universität
Braunschweig



January 2025

Master Thesis Project for Eric Schaar

Title: Extension of a high-fidelity multidisciplinary feasible aero-structural optimization framework by a bi-level formulation

Short description:

Solving multidisciplinary design optimization (MDO) problems using a multidisciplinary feasible (MDF) architecture is common practice in academia. This approach allows to retain all of the interdisciplinary interactions between the disciplines with the requirement of calculating the full derivative information.

For applications in industry, some disciplines may not be able to provide the full derivative information needed to realize a true MDF architecture. Oftentimes, embedded architectures are applied, where no interdisciplinary derivatives are needed. A bilevel optimization architecture, aimed to close the gap between both approaches, introduces a small set of interdisciplinary coupling variables to reduce the number of required interdisciplinary derivatives that need to be computed. Engineering knowledge is used for definition of the additional coupling variables in order to retain interdisciplinary interaction, to get optimization results comparable to the MDF approach.

The optimization problem to be used for this investigation is a simultaneous structural sizing and shape optimization of the DLR F25 high-aspect ratio research aircraft configuration applying a high-fidelity aero-structural performance prediction model as well as a lower fidelity loads model for structural sizing. The DLR code TAU is used for the aerodynamic analysis, while LANGRANGE from Airbus Defence and Space is used as a structural solver. The two tools are coupled via the HPC-capable simulation environment FlowSimulator, which DLR and Airbus use jointly.

In this Master thesis, the current process implementing an MDF architecture is to be reformulated into an embedded architecture, optionally followed by a bi-level architecture, using GEMSEO by IRT, a generic engine for multidisciplinary optimization which is used both at Airbus and at DLR.

The bi-level formulation will be applied on the industrially relevant aircraft configuration, the optimization of the DLR F25. Finally, the developed bi-level optimization method will be benchmarked against the existing MDF optimization architecture.

The following main steps are required:

1. Expose components of MDO chain to GEMSEO
2. Build processes capable of computing the coupling variables including their sensitivities
3. Optimization of the DLR F25
4. Comparison to results generated with an MDF architecture

Suggested Tasks:

Task No.	Task description	Est. period (weeks)
1	Literature review and familiarization with industrial MDO processes and requirements	2
2	Familiarization with GEMSEO	1
3	Familiarization with the existing MDF-type MDO chain and processes	2
4	Implement proposed concept for embedded architecture into GEMSEO using same disciplinary tools and components as for MDF	5

5	Implement existing concept for bi-level architecture in GEMSEO using bi-level plugin for strongly coupled aero-structural problems (optional)	4
6	Optimization of the DLR F25 and comparison of existing MFD results with results of embedded architecture (and bi-level, optional)	6
7	Postprocessing results according to industrial requirements and deploy GEMSEO process script in industrial environment with interfaces, but not including any plugins and domain-specific tools (no installation of tools)	2
8	Documentation	4

Braunschweig, 1.8.2024 

_____

Date & Signature Examiner

Eidesstattliche Erklärung

Ich Eric Schaar erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbstständig verfasst, sowie alle benutzten Quellen und Hilfsmittel vollständig angegeben habe und dass die Arbeit nicht bereits als Prüfungsarbeit vorgelegen hat.

Erklärung zur Nutzung von Künstlicher Intelligenz (KI)

Ich, Eric Schaar, erkläre hiermit transparent die Nutzung von Künstlicher Intelligenz (KI) im Rahmen meiner Masterarbeit.

Art der verwendeten KI-Systeme

Im Rahmen dieser Arbeit wurden folgende KI-Systeme verwendet:

- *Claude 3.5 Sonnet*
- *DeepL Write*

Umfang und Art der KI-Nutzung

Die KI-Systeme wurden für folgende Aspekte der Arbeit eingesetzt:

Textverarbeitung und -optimierung

- Generierung von Vorschlägen für alternative Formulierungen und Satzkonstruktionen
- Korrekturlesen auf Rechtschreibung, Grammatik und Stil
- Unterstützung bei der Quellenrecherche

Codegenerierung und -optimierung

- Generierung von Code Vorschlägen für kleine Programmteile, die keine Schnittstelle zu proprietärem Quellcode haben
- Fehlersuche in Programmteilen, die keinen proprietären Quellcode enthalten

Verantwortlichkeit

Ich erkläre, dass:

- Alle KI-generierten Inhalte von mir auf Richtigkeit überprüft und überarbeitet wurden
- Alle inhaltlichen Entscheidungen von mir getroffen wurden
- Keine geschützten Inhalte an die KI-Modelle weitergegeben wurden
- Die KI-Richtlinie der TU Braunschweig eingehalten wurde

Abstract

This thesis investigates the applicability of bilevel formulations for coupled aerostructural design optimization of aircraft wings. Bilevel optimization formulations separate an optimization problem into a system-level and lower-level optimization problem, allowing the lower-level problems to be solved independently of each other, while or the commonly used monolithic formulations, interdisciplinary gradient exchange is required , which is prohibited by existing disciplinary organizational structures. Thus, these formulations have recently become of particular interest for the aerospace industry. To expand the understanding of bilevel formulations, this thesis adapts an existing monolithic optimization process to enable the utilization of bilevel processes. The implementation is carried out in the novel optimization framework GEMSEO to facilitate the reusability of processes across different formulation types. The efficacy of the adapted process is evaluated through a comparative analysis with the results obtained from a multidisciplinary feasible formulation. This analysis is performed by testing the adapted process for a wing profile on the simple transonic wing and the DLR-F25 test cases. The findings indicate that bilevel formulations of various types can attain mission fuel burn reductions comparable to those of the multidisciplinary feasible formulation, at approximately 2.5% and 4%, respectively. Despite the similarity in the objective function, the optimized designs exhibit slightly different design characteristics, with the bilevel optimized design tending to be more aerodynamically optimal. To assess the robustness of the bilevel formulations, a starting point variation of the simple transonic wing is performed. Certain bilevel formulations show a dependence on the starting point. However, the use of an implicit wing root bending moment target successfully mitigates this tendency. The results demonstrate the viability of bilevel formulations for coupled aerostructural optimization, enabling industry to implement multidisciplinary optimization while respecting organizational constraints for a distributed aircraft design process.

Contents

List of Figures

List of Tables

Acronyms

General Terms

HPC	high performace computing
MLS	*moving-least-squares*
MPI	*Message Parsing Interface*
t	time
UPWing	*Ultra Performance Wing*

Aerodynamic and Aircraft Terms

α	angle of attack
α_i	induced angle of attack
AR	aspect ratio
AVL	*Athena Vortex Lattice*
b	wing span
C_D	coefficient of drag
$C_{D,i}$	induced coefficient of drag
CFD	computational fluid dynamics
CFRP	carbon-fiber reinforced polymer
c	chord
Γ	circulation
C_L	coefficient of lift
D	drag
δ	mean camber slope
e	span effectiveness factor
HARW	high aspect-ratio wing

L	lift
l	characteristic length
LD	lift over drag ratio
M	moment
Ma	Mach
Ma_{cr}	critical mach number
Ma_{div}	drag-divergence Mach
m_f	fuel mass
ϕ	dihedral angle
POD	proper-orthogonal-decomposition
q	dynamic pressure
R	range
RANS	*Reynolds averaged Navier-Stokes*
RBF	radial basis function
SA	*Spalart-Allmaras*
SVD	singular value decomposition
SAF	sustainable aviation fuel
S_{ref}	reference area
ϕ	sweep angle
T	thrust force
t	thickness
t/c	thickness-to-chord ratio
$TSFC$	thrust specific fuel consumption
V	velocity
V_∞	free stream velocity
VLM	vortex-lattice method
w	downwash
W	weight

Optimization Terms

BFGS	*Broyden-Fletcher-Goldfarb-Shanno* method
BL-BCD	*bilevel-block-coordinate-descent*
BL-EMB	*bilevel-embedded*
BL-IRT	*bilevel* formulation developed by the IRT Saint Exupery
COBYLA	*Constrained Optimization by Linear Approximation*
DOE	design of experiments

GP	*Gaussian process*
IP	*Interior-Point*
KKT	*Karush-Kuhn-Tucker* conditions
MDA	multidisciplinary analysis
MDF	*Multidisciplinary-Feasible*
MDO	multidisciplinary design optimization
ROM	reduced-order-model
SCBO	*scalable constrained bayesian optimization*
SQP	sequential quadratic programming
SSBJ	supersonic business jet
TuRBO	*trust region Bayesian optimization*
XDSM	extended design structure matrix

1 Introduction

The aviation industry is at a critical juncture in its evolution, driven by two competing forces: the increasing demand for air travel and the ambitious goal of achieving net carbon neutrality by 2050 at the latest, as agreed by all *ICAO* member states [1]. To meet these challenges, significant efforts are being made to replace fossil fuels with alternative energy sources such as sustainable aviation fuel (*SAF*) and hydrogen, as exemplified by *Airbus*'s *ZEROe* project. However, the widespread application of these technologies faces significant economic barriers, with *SAF* commanding substantial price premiums over conventional jet fuel [2] and liquid hydrogen requiring costly cryogenic storage infrastructure. Thus, significant improvements in aircraft efficiency are a key enabler for the economic deployment of any future fuel. the In response to this challenge, the *Ultra Performance Wing* (*UPWing*) project was launched in 2023 by a joint European consortium with the ambitious goal of reducing the total fuel consumption of a short-to-medium range aircraft by at least 30% compared to a state-of-the-art aircraft. A key innovation is the increased aspect ratio (AR) of 15.6 for the reference *DLR-F25* configuration, significantly higher than the AR of about 9 found in the comparable *Airbus A320* aircraft.

This increase in AR amplifies the interactions of aerodynamic and structural design. As wing span increases and chord length decreases to achieve higher aspect ratios, the elastic twisting of the wing increases, tightly coupling the aerodynamic and structural disciplines, resulting in a complex, multidisciplinary design task. Consequently, the project developed advanced multidisciplinary design optimization (*MDO*) methods to determine the optimal balance between improved aerodynamic performance and structural mass.

MDO methods have been a research topic since the late 1970s [3], with aircraft wings being among the first optimization subjects [3–5]. Within the industry, Airbus Defence and Space has developed the *LARGANGE* structural evaluation and optimization tool, which tightly integrates the load calculation and structural sizing process under consideration of external design requirements [6, 7]. Previous research has predominantly used monolithic *MDO* formulations, in particular the *Multidisciplinary-Feasible* (*MDF*) approach [8–10]. A

major reason for the focus on this class of algorithms is the fast convergence compared to distributed algorithms [11]. Despite extensive research and promising results, the aerospace industry has yet to fully integrate high-fidelity MDO methods into its design processes.

As pointed out by Spieck et al. [12], the challenges in adopting *MDO* are not purely technical, but include sociological aspects such as integration into existing, discipline-separated organizational structures. Monolithic architectures require a strong integration of the individual disciplines, especially for computing the sensitivities required for gradient-based optimization, which is most efficient for large-scale optimization problems [8]. Because of this organizational mismatch, recent research has re-focused on distributed architectures, despite their longer convergence times [11]. *Bilevel* formulations have emerged as particularly promising for both research and industrial applications [13, 14], as they preserve disciplinary decomposition and eliminate the need for coupled gradients by optimizing disciplines separately.

To advance research in these promising distributed algorithms, this thesis aims to perform aerostructural optimizations of industry-relevant test cases, including the HARW *DLR-F25* configuration. The research makes use of state-of-the-art methods and tools used by *Airbus* and *DLR*, including the *TAU* CFD solver, the *LAGRANGE* structural solver and the multidisciplinary coupling framework *FlowSimulator*. The optimization process developed in this thesis is implemented in the novel *GEMSEO MDO* framework [15] to improve the automation and reusability of the processes.

This thesis is structured as follows: Chapter 2 presents the theoretical foundations of the applied methods, including fundamental concepts of aircraft aerodynamics and aeroelasticity, along with brief explanations of numerical methods and multidisciplinary optimization algorithms. In Chapter 3, the assembly of the MDO processes from the different disciplinary tools is shown. The benchmark test cases to compare the different *bilevel* processes are presented in Chapter 4. The results of the studies are presented in Chapter 6 and discussed in Chapter 7. These results are discussed under the research question whether novel *bilevel* formulations enable the use of *MDO* in the aerospace industry in Chapter 7. Further improvements to the processes are proposed in Chapter 8.

2 Fundamentals of Aerodynamic Analysis and Multidisciplinary Optimization

This chapter establishes the theory behind a multidisciplinary aerodynamic and structural optimization of high aspect-ratio wings. First, the main concepts of aircraft aerodynamics relevant for this optimization qualitatively are introduced (Section 2.1). The numerical methods used to quantify the effects of these aerodynamic phenomena are then detailed, along with mathematical approaches for calculating the gradients of numerical simulations (Section 2.2). Finally, the chapter presents the theory underlying the algorithms used to formulate and solve the multidisciplinary optimization problem (Section 2.4). The implementation details are the subject of the following chapter (Chapter 3).

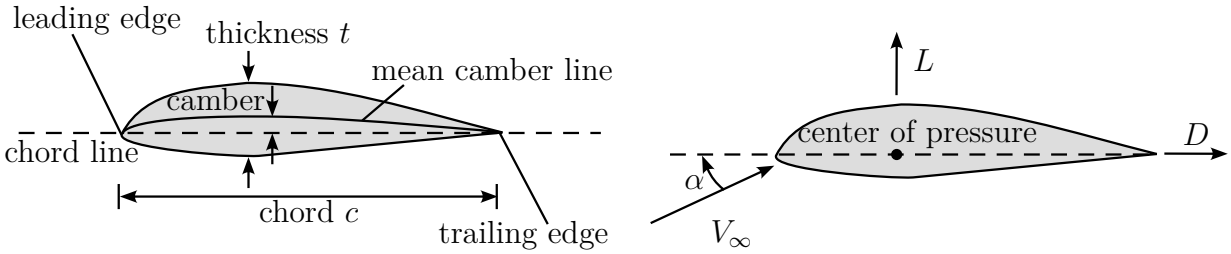
2.1 Aircraft Aerodynamics

This section provides a comprehensive foundation in aircraft aerodynamics, essential for understanding the optimization challenges and opportunities in modern aerospace design. The discussion is progressed systematically from fundamental principles to advanced concepts, establishing the theoretical framework that underlies our optimization study. It begins with the fundamental mechanisms of lift generation through an analysis of two-dimensional airfoils in Section 2.1.1, using potential flow theory. In Section 2.1.2, the analysis is extended to three-dimensional effects by examining wings of finite length. Here, it is highlighted, how spanwise flow and tip vortices influence lift distribution and generate induced drag - phenomena crucial for understanding the effect of high-AR wings. This analysis provides critical insights into the relationship between wing geometry and aerodynamic efficiency. The discussion advances to transonic aerodynamics in Section 2.1.3, addressing the effects of flow compressibility encountered in commercial aviation. It is qualitatively shown, how shock wave formation and interaction with boundary layers

impact aircraft performance at high subsonic speeds, introducing key considerations for contemporary transport aircraft design. The final Section 2.1.4, bridges the gap between aerodynamics and structures by analyzing static aeroelastic effects. We investigate how wing flexibility influences aerodynamic performance and explore the opportunities presented by aeroelastic tailoring. This section identifies crucial design parameters that enable the optimization of both structural and aerodynamic performance. Throughout these sections, strategic simplifications are employed to maintain focus on essential concepts while providing relevant mathematical formulations that enhance understanding.

2.1.1 Profile Aerodynamics

The study of airfoil characteristics serves as the foundation for understanding aerodynamic effects present on any aircraft. An airfoil represents the two-dimensional cross-sectional geometry of a three-dimensional wing, and its properties fundamentally influence the wing's aerodynamic behavior. The airfoil's geometry is defined by several key parameters illustrated in Figure 2.1a. The mean camber line, which represents the locus of points equidistant from the upper and lower surfaces, extends from the leading edge (the foremost point) to the trailing edge (the aftmost point). The chord line, a straight line connecting these points, serves as the primary reference for geometric measurements. The chord length, denoted as c, represents the characteristic dimensional parameter of the airfoil. The maximum perpendicular distance between the chord line and the mean camber line is defined the airfoil's camber, while the maximum separation between the upper and lower surfaces constitutes the airfoil thickness (t). The mean camber line describes the middle line between the upper and lower surface of the airfoil. It begins at the most forward point of the airfoil, the leading edge, and ends at its rearmost point, the trailing edge. Connecting these two points is the straight chord line. The length of the chord line is simply refered to as the chord (c) of the airfoil, while the maximum distance between the chord line and the mean camber line is called the camber of the airfoil. The airfoils thickness (t) is measured as the maximum distance between the upper and lower surface. Figure 2.1b depicts the main parameters relevant to characterize the flow around an airfoil and the resulting forces. The free stream velocity (V_∞) is the velocity of the undisturbed flow upstream of the airfoil. The angle between the free-stream and the chord line is refered to as the angle of attack (α). The interaction of the flow with the airfoil generates two forces, drag (D) facing the flow direction and lift (L) perpendicular to the flow. Depending on the reference points, these two forces induce a moment (M). The point at which the resultant M equals zero is designated as the center of pressure. For meaningful comparison and analysis, these aerodynamic forces and moments are typically expressed as non-dimensional coefficients. These coefficients are derived by normalizing the forces



(a) Airfoil nomenclature

(b) Flow nomenclature around two-dimensional airfoil

Figure 2.1: Basic nomenclature for two-dimensional airfoils in a flowfield

and moments with respect to dynamic pressure (q)

$$q = \frac{1}{2}\rho V^2, \tag{2.1}$$

and a reference area (S_{ref}). For fixed-wing aircraft S_{ref} is usually the planform area. To normalize moment coefficents, a characteristic length (l) is used, for airfoils this is c. Normalizing the forces and moment with those parameters results in the coefficents

$$c_D = \frac{D'}{qS_{ref}}, \quad c_L = \frac{L'}{qS_{ref}}, \quad c_M = \frac{M'}{qS_{ref}l}. \tag{2.2}$$

The generation of aerodynamic forces results from complex fluid-structure interactions, which cause pressure forces acting normal to the surface and shear forces acting tangetial to the surface. As the normal vectors on the airfoil surface predominantly point in the direction of lift, the pressure forces contribute mainly to lift, while shear forces cause drag. The generation of lift can be mathematically described through the concept of circulation (Γ)

$$\Gamma = -\oint_C \mathbf{V} \cdot d\mathbf{s}. \tag{2.3}$$

From the circulation strength, the coefficient of lift is calcuated with the *Kutta-Joukowski* theorem

$$c_L = \frac{2\Gamma}{V_\infty S_{ref}}. \tag{2.4}$$

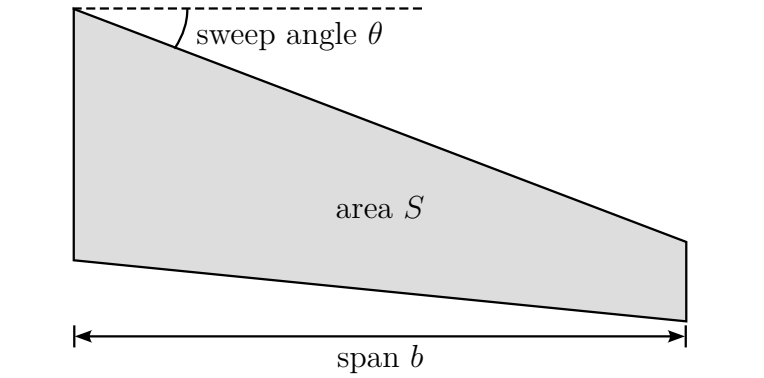
In incompressible flow regimes, drag forces primarily result from viscous effects, manifesting as skin friction and pressure drag due to flow separation. For three-dimensional applications, additional considerations include induced drag from finite wing effects and wave drag in transonic and supersonic flow regimes.

2.1.2 Finite Wing Aerodynamics

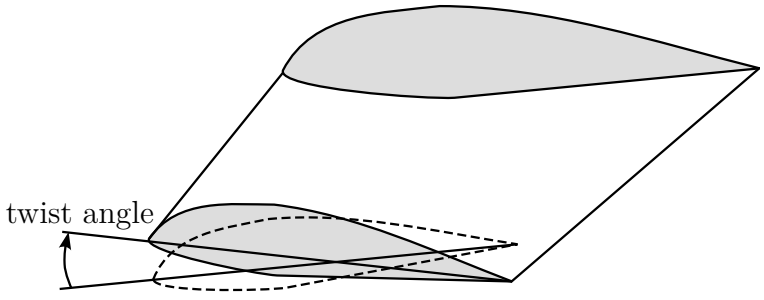
The transition from two-dimensional airfoil theory to three-dimensional wing analysis introduces several additional geometric parameters. A trapezoidal planform shape of a wing is outlined in Figure 2.2a. The sweep angle (ϕ) angle defines the angle between the wing's leading edge and the perpendicular of the aircraft's centerline. The wing span (b) represents the total horizontal distance from wingtip to wingtip. The aforementioned aspect ratio (AR), is derived from the relationship between the wing span and planform area (S), expressed as

$$AR = b^2/S. \tag{2.5}$$

This ratio serves as a crucial indicator of wing efficiency, that becomes of importance later in this explanation. Wing twist, another significant geometric characteristic, manifests in two forms: geometric twist, which is inherent in the wing's design, and aerodynamic twist, which results from structural deformation under aerodynamic loading (how the twist changes in in flight and its effects on the aerodynamic performance is explained in Section 2.1.4). The twist angle represents the relative angular displacement between wing sections measured with respect to their chord lines. The finite length of a three-



(a) Wing planform



(b) Wing twist angle

Figure 2.2: Simple wing nomenclature

dimensional wing introduces a significant aerodynamic phenomenon known as induced drag. This concept can be explained through the application of vortex filament theory

under the simplification of inviscid and incompressible flow conditions. A vortex filament, characterized by circulation strength velocity (V) induces a velocity-field according to Biot-Savart-law:

$$d\mathbf{V} = -\frac{\Gamma}{4\pi}\frac{d\mathbf{l}\times\mathbf{r}}{|\mathbf{r}|^3}. \tag{2.6}$$

According to HELMHOLTZ's vortex theorem, these filaments must maintain constant strength along their path and either extend to infinity or form closed loops, as they cannot terminate within the fluid volume. PRANDTL's lifting line theory provides a framework for understanding induced drag through the concept of a horseshoe vortex system. The theory models the wing's leading edge with a system of vortex filaments, which induce a downward velocity component known as downwash (w). For a single horseshoe vortex, the downwash distribution along the span is expressed as:

$$w(y) = -\frac{\Gamma}{4\pi b}\left(\frac{b/2+y}{(b/2+y)^2+z^2} + \frac{b/2-y}{(b/2-y)^2+z^2}\right) \tag{2.7}$$

This horseshoe vortex is displayed in Figure 2.3a together with the induced downwash. To overcome the singularity of a single horseshoe at the wing tips, where $y = b/2$, the theory employs infinitesimal horseshoes covering wing span increments dy like shown in Figure 2.3b. This furthermore allows the introduction of a spanwise lift distribution as Γ does not have to be constant over the span anymore. For an arbitrary point y_0 along the span of the wing, the velocity induced by the vortex sheet is

$$w(y_0) = -\frac{1}{4\pi}\int_{-b/2}^{b/2}\frac{\Gamma(y)}{y_0-y}\,dy. \tag{2.8}$$

Superpositioning this downward facing flow with V_∞ changes the direction of flow, adding an induced angle of attack (α_i) to the geometric α, which is

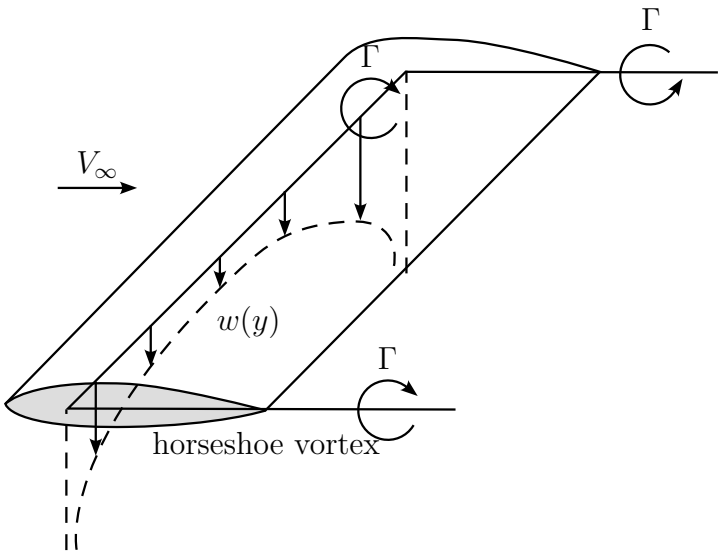
$$\alpha_i(y_0) = \tan^{-1}\left(\frac{-w(y_0)}{V_\infty}\right), \text{for small } \alpha_i : \alpha_i = \left(\frac{-w(y_0)}{V_\infty}\right). \tag{2.9}$$

As α is now increased by α_i, a part of the lift generated by the airfoil now faces against the direction of V_∞, creating the induced drag. This induced drag is simply:

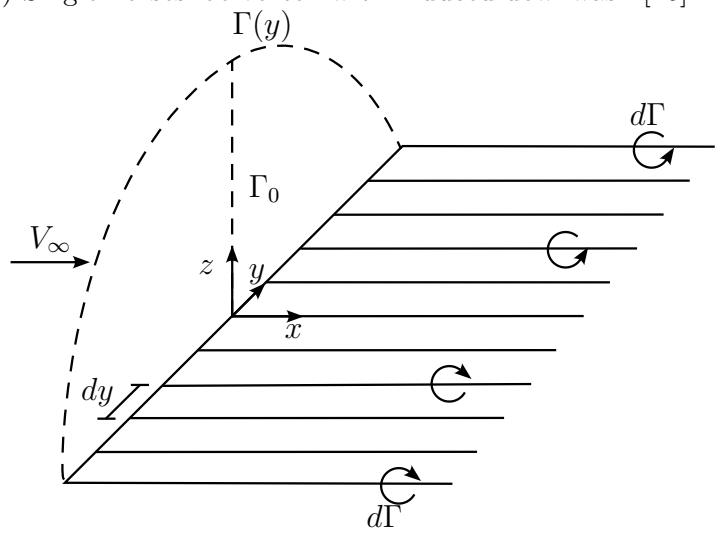
$$C_{D,i} = \sin\alpha_i \cdot C_L, \text{for small } \alpha_i\text{: } C_{D,i} = \alpha_i \cdot C_L \tag{2.10}$$

By inserting *Kutta-Joukowski* theorem, the induced coefficient of drag ($C_{D,i}$) can be calculated from the circulation (Γ):

$$C_{D,i} = \frac{1}{V_\infty S}\int_{-b/2}^{b/2}\Gamma(y)\alpha_i(y)\,dy. \tag{2.11}$$



(a) Single horseshoe vortex with induced downwash [16]



(b) Vortex sheet with elliptical lift distribution [16]

Figure 2.3: Horseshoe vortex modelling of a straight wing

When the lift distribution $\Gamma(y)$ is elliptical, like shown in Figure 2.3b

$$\Gamma(y) = \Gamma_0 \sqrt{1 - \left(\frac{2y}{b}\right)^2} \tag{2.12}$$

the induced downwash velocity and thus the induced angle of attack are constant along the wingspan

$$w(y) = \frac{-\Gamma_0}{2b}, \quad \alpha_i(y) = \tan^{-1}\left(\frac{-\Gamma_0}{2b}\right). \tag{2.13}$$

The unknown circulation strength Γ_0 can be obtained by inserting the elliptical lift distribution eq. (2.12) into the *Kutta-Joukowski* theorem eq. (2.4). From this result, α_i can also be obtained through the relation in eq. (2.15).

$$\Gamma_0 = \frac{2V_\infty S C_L}{b\pi} \tag{2.14}$$

$$\alpha_i = \frac{S C_L}{\pi b^2} \tag{2.15}$$

Recalling, that AR has been defined as $\frac{b^2}{S}$, thus by multiplying α_i by the coefficient of lift gives a result for $C_{D,i}$ eq. (2.10), which is inversely proportional to AR.

$$C_{D,i} = \frac{C_L^2}{\pi AR} \tag{2.16}$$

This relation is only valid for wings that have an elliptical lift distribution, but shows, that induced drag is reduced by increasing AR. For a non-elliptical lift-distribution the span effectiveness factor (e) measures the similarity of the actual distribution to the optimal ellipse.

$$C_{D,i} = \frac{C_L^2}{\pi e AR}. \tag{2.17}$$

This factor is equal to 1 for the elliptical lift distribution and $1 > e > 0$ for any other distribution. Thus, an elliptical lift distribution is optimal in terms of induced drag, as $C_{D,i}$ is minimal.

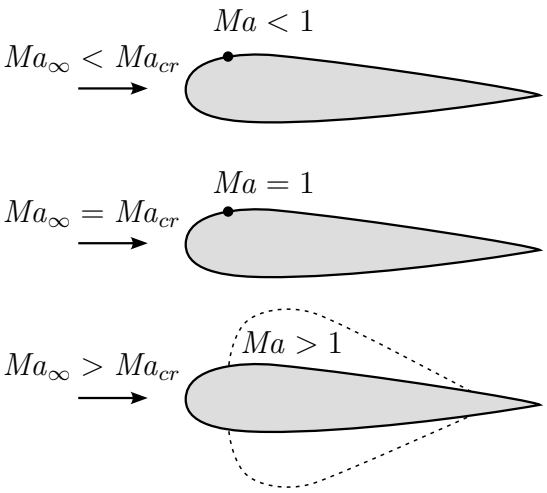
2.1.3 Pressure Drag at Transonic Airspeeds

In the previous sections, fundamental concepts of the aerodynamics of aircraft wings were introduced. However, this introduction has so far only discussed incompressible flows. The considered transport aircraft configurations fly close the transonic flow regime, where effects of compressibility have a crucial effect on the generation of drag. Inside the transonic flow regime, certain parts of the flow can exceed the local speed of sound. The relation of the airspeed and the local speed of sound is defined as the Mach (Ma)

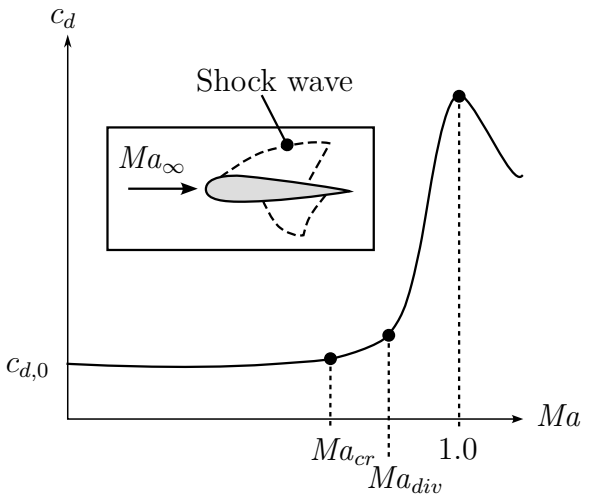
number, which depends on the gas constant R, the isentropic exponent of the gas κ, and temperature T

$$Ma = \sqrt{\kappa \cdot R \cdot T}. \tag{2.18}$$

When increasing the subsonic free-stream velocity of a flow around an arbitrary airfoil, at some chordwise location on the airfoil, the local velocity exceeds $Ma = 1$. The corresponding free-stream Ma number is the critical mach number (Ma_{cr}) number. Further increases in free stream velocity (V_∞) results in a supersonic flow region on the airfoil, illustrated in Figure 2.4a, causing a sudden increase in drag, after the drag-divergence Mach (Ma_{div}) threshold is surpassed. The drag increases up to the sound-barrier ($Ma = 1$), after which it decreases again, this trend is illustrated in Figure 2.4b [16]. At the point



(a) Qualititive drawing of airfoil near critical Ma number [16]



(b) Qualititive profile drag plot for increasing Ma number [16]

on the airfoil, at which $Ma = 1$ when $Ma_\infty = Ma_{cr}$, the pressure coefficient C_p is the critical pressure coefficient $C_{p,crit}$. This coefficient can be expressed as a function of Ma_{cr} and gas properties, allowing determination of Macr through the intersection of the $C_{p,crit}$ function with the minimum pressure coefficient curve for various Ma numbers. The airfoil's geometry, particularly its thickness, substantially influences its compressibility characteristics. Thicker airfoils generate stronger flow perturbations, resulting in more negative minimum pressure coefficients compared to thinner sections. For geometrically similar airfoils with different thickness-to-chord ratio (t/c), increased thickness correlates with lower critical Ma numbers, leading to earlier onset of wave drag in the transonic regime. When considering an airfoil of the same shape, except for varied t/c, the thicker airfoil will show a more negative minimum C_p than the thinner one. As can be see from the qualitative plot in Figure 2.5, this coincides with a lower Ma_{cr} number. Thus, this airfoil has higher wave drag at a given Ma number that is close enough to the transonic flow regime than the thinner airfoil. However, as the drag does not increase rapidly at Ma_{cr}, but at Ma_{div}. Thus, so called supercritical airfoils have been designed since the

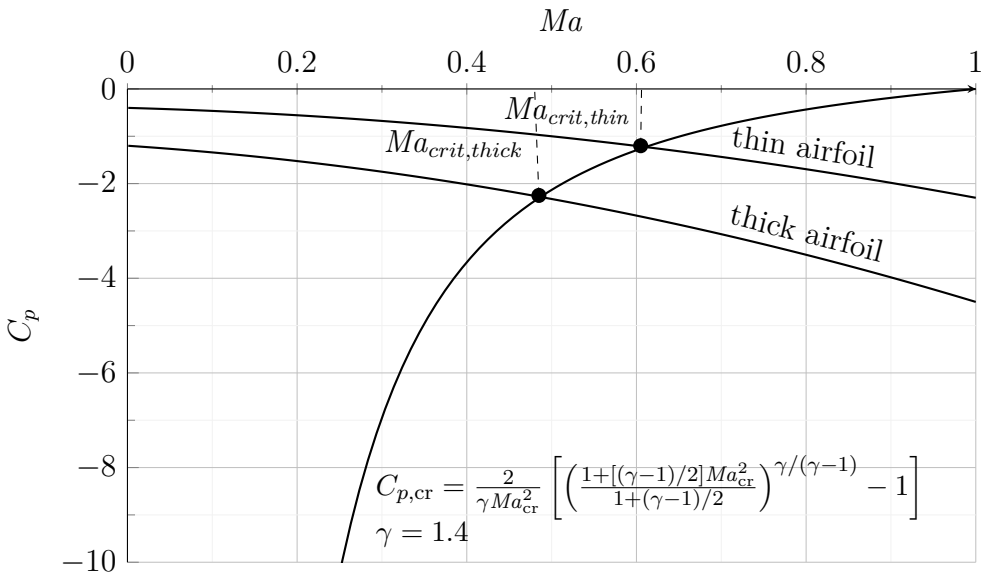


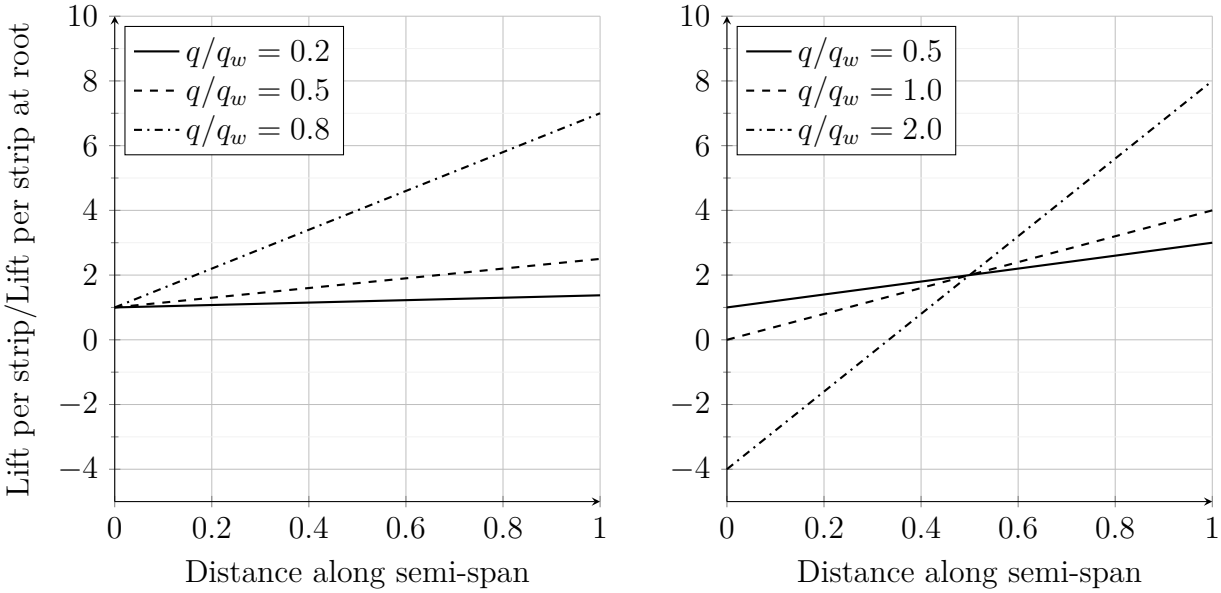
Figure 2.5: Minimum C_p and coinciding critical Ma number for airfoils of different thicknesses [16]

1960s with the goal to weaken the magnitude of the terminating shock. Another technique to reduce wave drag of wings is a sweep, typically backwards of the wings. By sweeping a wing with the same profile backwards, the air does not flow parallel to the original chord lines anymore, increasing the effective chord length of the wing. Even though the thickness is unchanged, the effective t/c ratio is reduced. This increases the value of Ma. To summarize this short introduction of wave drag and design choices for reducing it, key elements important for the design of jet transport aircraft wings are the thickness and the sweep angle of the wing, as these parameters increase the Ma_{cr} number and thus the onset of wave drag. Furthermore, the strength of the terminating shock can be reduced by choosing an appropriate transonic wing profile to increase the drag-divergence Ma_{div} number.

2.1.4 Aeroelastic Coupling

The wings on modern, high aspect-ratio wing ($HARW$) configurations are flexible and deform when under aerodynamic loads. These deformations changes the flow states and therefore the aerodynamic properties of the wings. Hence the understanding this coupling is crucial for accurately predicting aircraft performance during cruise conditions. The most significant deformation effect is wing twist, which increases with airspeed and dynamic pressure. This relationship continues until reaching a critical point called divergence, characterized by the divergence dynamic pressure q_w. At this point, the lift forces overcome the wing's structural resistance, theoretically resulting in infinite

twist angles. While structural failure would occur before reaching this theoretical limit, the divergence dynamic pressure serves as a vital parameter for understanding a wing's aeroelastic behavior. This equation is important as it directly relates the lift distribution of the elastic wing to the divergence pressure q_w and thus its stiffness. To illustrate this relation, Figure 2.6 shows the spanwise lift distribution of the straight wing at different ratios of q/q_w. As can be clearly seen, the higher this quotient becomes, the more the lift is shifted outboards, creating a potentially undesireable lift distribution. This plot shows the behaviour of an elastic wing without trim, however, during the cruise flight state, the additional lift generated is compensated using trim to maintain a horizontal flight path. As the increasing incidence is compensated by decreasing α, the divergence speed of the untrimmed aircraft can be exceeded. Under these conditions, where $q/q_w > 1$, the lift becomes negative at the wing root. This is depicted in Figure 2.6b. As a generalized



(a) Untrimmed case

(b) Trimmed case

Figure 2.6: Spanwise lift distributions for different dynamic pressures [17]

summary, the finite stiffness of a wing influences the lift distribution of a wing, causing a general outboard shift of the lift distribution. This may cause undesireable lift distributions and increase the wing root bending moment. Measures that decrease this effect are to increase the structural stiffness, reduce the wing span or increase the sweep angle of the wing [17].

2.1.5 Breguet Range Estimation

On of the main objectives of aircraft design is the reduction of fuel consumption. While simulating the aircrafts power requirement over the whole flight envelope is an accurate method for determining the fuel consumption, it is computationally expensive, so estimation methods are of practical importance. Here, the *Breguet* range equation is derived. This derivation assumes a constant velocity during the considered flight segment. Thus, the range of an aircraft is simply velocity (V) multiplied by time (t) of the flight segment

$$R = V \cdot t. \tag{2.19}$$

A second assumption is constant fuel consumption throughout the flight, characterized by the thrust specific fuel consumption ($TSFC$) for a given thrust requirement. Under this assumption, the fuel massflow equals

$$\frac{dm}{dt} = -TSFC \cdot T. \tag{2.20}$$

In steady, horizontal flight, mass of an aircraft m is equal to lift (L) times the gravitational constant g and thrust force (T) is equal to D. By simply rearranging this relation, the thrust equals weight (W) divided by the lift over drag ratio (LD). This formula is substituted into eq. (2.20) and variables are seperated

$$\frac{LD}{m \cdot g} \cdot dm = -TSFC \cdot dt. \tag{2.21}$$

We integrate both sides of the equation, the left side from the initial mass m_i to the final mass m_f and the right hand side from 0 to t

$$\frac{LD}{g} \cdot \ln\left(\frac{m_i}{m_f}\right) = -TSFC \cdot t. \tag{2.22}$$

Substituting t from eq. (2.19) and reformulating the equation, range (R) can be calculated

$$R = \frac{V \cdot LD}{TSFC \cdot g} \cdot \ln\left(\frac{m_i}{m_f}\right). \tag{2.23}$$

For optimization purposes total fuel mass (m_f) burned on a specific mission is of key interest. As m_f is the difference between inital mass m_i and final mass m_f, the equation can be reformulated into

$$m_f = m_i \cdot \left(1 - e^{-\frac{R \cdot TSFC \cdot g}{V \cdot LD}}\right). \tag{2.24}$$

This relation only holds for horizontal flight. To calculate the fuel burn in flight segments, where the flight path angle γ is non-zero, Adler et al. [10] have extended this formula by changing the force balance

$$L = m \cdot g \cdot \cos(\gamma) \tag{2.25}$$

$$T = D + m \cdot g \cdot \sin(\gamma). \tag{2.26}$$

Following the same derivation as before, the fuel burn for the climb segment is

$$m_f = m_i \cdot \left(1 - e^{-\frac{R \cdot TSFC \cdot g}{V}\left(\frac{1}{LD}+\gamma\right)}\right). \tag{2.27}$$

Using this relationship as a target funtion for a surrogate-based multidisciplinary optimization of a *Boeing 737* aircraft, they achieved similar results using eq. (2.27) and a single evaluation point in cruise and climb, than with a mission-based optimization, integrating fuel consumption over the whole flight. This allows to use mission-based optimization even with high-fidelity methods, as the number of function evaluations is significantly reduced compared to the full mission analysis. Considering the climb segment can be of great importance for the total fuel burn, as for shorter missions up to 500 nm, the climb segment is responsible for more than half of the total mission fuel consumption, as is depicted in Figure 2.7

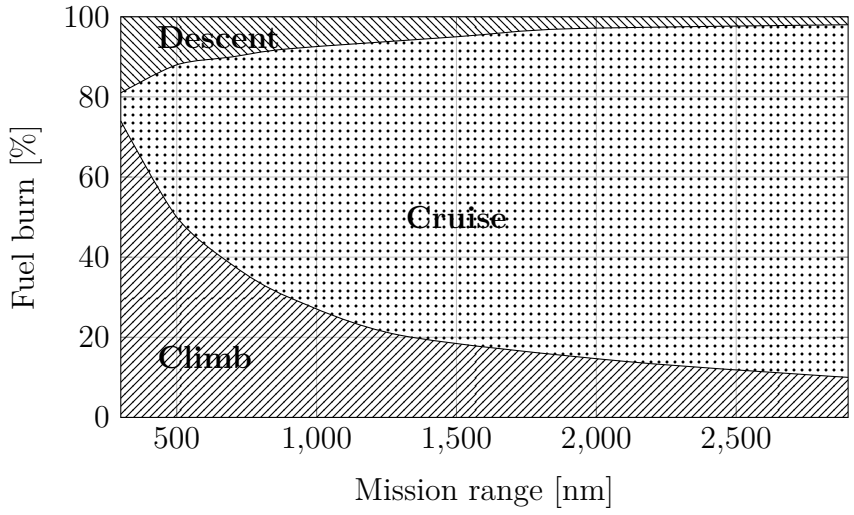


Figure 2.7: Ratio of fuel burn for climb, cruise, and descent for varying mission lengths [10]

2.2 Numerical Methods

After the introduction of the main concepts of aircraft aerodynamics relevant to coupled aerostructural optimizations in the previous section, this section explains the methodology behind the numerical methods employed for the optimization. The first concept investigateg are the *Reynolds averaged Navier-Stokes* (*RANS*) equation for computational fluid dynamics (*CFD*) in Section 2.2.1, which are a high fidelity method to compute the aerodynamic performance and loads of an aircraft. In the following Section 2.2.2, the *Finite-Element-Method* for structural simulation is briefly introduced, which is relevant for numerically evaluating structural stress and deformations. As both methods use different types of meshes, in Section 2.2.3 a method for interpolation of forces and displacements between different meshes is explained. When displacements are interpolated to a CFD volume mesh, all nodes in the fluid field have to be deformed accordingly, this process is explained in Section 2.2.4. While the *RANS* CFD method can produce high-fidelity solutions for the aerodynamic performance prediction, they are computationally expensive, especially for load evaluation processes. Thus in Section 2.2.5, the vortex-lattice method (*VLM*) is explained, which is a lower-fidelity model for calculating aerodynamic lift using the principles of potential theory.

In this section, exclusively the theoretical concepts behind these simulations are explained without detailing their implementation, as this is the focus of the next Chapter 3. However, to ensure consistency between the chapters and improve the readability, certain nomenclature used throughout this and the subsequent chapters is established in the following. Both numerical processes of *CFD* (Section 2.2.1) and structural simulation (Section 2.2.2) are mesh based, thus the index ∂M is defined for quantities in the aerodynamic mesh boundary and ∂S for the coupling surface of the meshes. Furthermore, the index F is used for quantities related to the flow, such as the flow residual $\mathbf{R}_F$, the index M for quantities regarding the aerodynamic mesh such as the mesh residual $\mathbf{R}_M$, and the index S for structural quanties, such as the structure residual $\mathbf{R}_S$.

2.2.1 Computational Fluid Dynamics (CFD)

Automated aerodynamic optimization of aircraft requires methods to compute the aerodynamic forces and moments. Within the past research on *MDO*, almost all high-fidelity compuations are executed uses *RANS CFD* methods. The *RANS* equations are a variation of the *Navier-Stokes* equations, that allow a coarser mesh resolution than other *CFD*,

methods, like the *Direct-Numerical-Simulation* or *Large-Eddy-Simulations*, as they resolve turbulence effects with a turbulence model. The *RANS* equations average fluctuating terms and are shown here in their incompressible form

$$\nabla \cdot (\rho \bar{\mathbf{u}}) = 0 \tag{2.28}$$

$$\frac{\partial(\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}} + \rho \langle \mathbf{u}' \otimes \mathbf{u}' \rangle) = -\nabla \bar{\mathbf{p}} + \mu(\nabla^2 \bar{\mathbf{u}} + \nabla(\nabla \cdot \bar{\mathbf{u}})) \tag{2.29}$$

Equation (2.29) consists of convective terms $\boldsymbol{\phi}_c$ and viscous terms $\boldsymbol{\phi}_v$, their respective gradients are

$$\nabla \boldsymbol{\phi}_c = \nabla \bar{\mathbf{p}} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}} + \rho \langle \mathbf{u}' \otimes \mathbf{u}' \rangle) \tag{2.30}$$

$$\nabla \boldsymbol{\phi}_v = \mu(\nabla^2 \bar{\mathbf{u}} + \nabla(\nabla \cdot \bar{\mathbf{u}})) \tag{2.31}$$

With these terms and using the residual form, eq. (2.29) can be summarized as

$$\mathbf{R}_F = \frac{\partial \mathbf{y}_F}{\partial t} + \nabla(\boldsymbol{\phi}_c - \boldsymbol{\phi}_v). \tag{2.32}$$

Equations (2.28) and (2.29) are not closed, meaning there are more unknown variables present than equations due to the addition of the Reynolds Stress term $\rho\overline{u_i' u_j'}$. This Reynolds Stress term has to be solved through a turbulence model. A class of commonly used turbulence models are the eddy turbulence models. Their basic assumption is statistical movement of fluid particles. In a sheared flow condition, this statistical movement results in a transfer of momentum in the direction of the velocity gradient. Thus the Reynolds Stress is assumed as

$$-\rho(\overline{\mathbf{u}' \otimes \mathbf{u}'}) = \mu_T \nabla \bar{\mathbf{u}}\,, \tag{2.33}$$

with the proportionality constant μ_T is refered to as the eddy or turbulent viscosity. μ_T is received through a turbulence model like the one equation *Spalart-Allmaras* (*SA*) turbulence model. In this model, for the turbulent kinematic viscosity $\nu_t = \rho\mu_t$, the linear representation $\tilde{\nu}_t$, which is valid outside the boundary layers of the flow, is resolved by a transport equation, here shown in its simplest form

$$\frac{\partial \tilde{\nu}_t}{\partial t} = c_{b1} S \tilde{\nu}_t + \frac{1}{\sigma}[\nabla \cdot (\tilde{\nu}_t \nabla \tilde{\nu}_t) + c_{b2}(\nabla \tilde{\nu}_t)^2] - c_{w1} f_w \left[\frac{\tilde{\nu}_t}{d}\right]^2. \tag{2.34}$$

In the first term, the generation of turbulence by shear flow is modeled. The second term describes diffusion of turbulence and the last term the destruction of turbulence at the boundaries of the flow. The constants contained in the equation are calibrated based on experimental results as $c_{b1} = 0.135$, $c_{b2} = 0.622$, $\sigma = 2/3$, $\kappa = 0.41$, and $c_{w1} = c_{b1}/\kappa + (1 + c_{b2})/\sigma$. From the linear representation $\tilde{\nu}_t$, the kinematic viscosity profile

ν_t is obtained by a quartic relationship [18]

$$\nu_t = \tilde{\nu} f_{\nu 1}, \qquad f_{\nu 1} = \frac{\chi^3}{\chi^3 + c_{\nu 1}^3}, \qquad \chi \equiv \frac{\tilde{\nu}}{\nu}. \tag{2.35}$$

There exist multiple methods for discretizing the *RANS* equations onto a computational mesh, each with distinct advantages and theoretical foundations. Within both legacy and modern *CFD* codes, the *Finite-Volume* discretization remains predominant due to its conservation properties. The fundamental concept of FVM works by partitioning the computational domain into discrete control volumes and integrating the governing equations over each volume. This approach naturally enforces mass and energy conservation laws. For the continuity equation (eq. (2.28)), this integration yields:

$$\int_V \nabla \cdot (\rho \bar{\mathbf{u}}) \, \mathrm{d}V = 0, \tag{2.36}$$

where V represents the control volume, ρ is the fluid density, and $\bar{\mathbf{u}}$ is the mean velocity vector. Using the *Divergence Theorem*, the volume integral can be reformulated as a surface integral over the control volume's boundary:

$$\int_V \nabla \cdot (\rho \bar{\mathbf{u}}) \, \mathrm{d}V = \int_S (\rho \bar{\mathbf{u}}) \cdot \mathbf{n} \, \mathrm{d}S, \tag{2.37}$$

where S represents the surface boundary of the control volume and $\mathbf{n}$ is the outward-pointing unit normal vector to the surface. In practical implementations, each mesh cell is treated as a control volume with a finite number of faces shared with neighboring cells. This allows the surface integral to be discretized as a sum of fluxes across each cell face:

$$\int_S (\rho \bar{\mathbf{u}}) \cdot \mathbf{n} \, \mathrm{d}S \approx \sum_{f=1}^{N_f} (\rho \bar{\mathbf{u}})_f \cdot \mathbf{n}_f A_f, \tag{2.38}$$

where N_f is the number of faces, A_f is the face area, and the subscript f denotes evaluation at the face [19].

The crucial step in *Finite-Volume* methods is the interpolation from cell-centered to face fluxes. Various interpolation schemes exist, ranging from linear interpolation to complex higher-order interpolation methods.

2.2.2 Structural Simulation

This section explains the linear *Finite-Element* method using the virtual work principle, which is applied in many engineering disciplines for calculating static continuum problems.

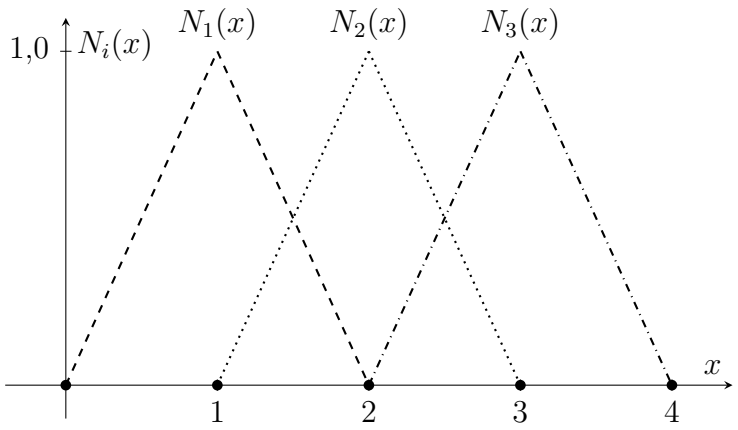


Figure 2.8: One-dimensional finite element discretization with linear shape functions [20]

We consider a general volume element subject to internal and external forces. The virtual work δW is defined as the product of force $\mathbf{F}$ and an infinitesimal discplacement $\delta\mathbf{v}$. This displacement is sufficiently small, so that the force $\mathbf{F}$ remains constant during the displacement.

This is summarized to the foundational equation of the *Finite-Element* method

$$\int_S \mathbf{t}(\mathbf{x}) \cdot \delta\mathbf{v} \, \mathrm{d}S + \int_V \mathbf{f}(\mathbf{x}) \cdot \delta\mathbf{v} \, \mathrm{d}V = \int_V \boldsymbol{\sigma}(\mathbf{x}) \cdot \delta\boldsymbol{\varepsilon} \, \mathrm{d}V. \tag{2.39}$$

To solve this equation numerically, the continuum is discretized using the Galerkin method. A one-dimensional example, such as a rod discretized into 4 elements with 5 nodes (vertices), including 2 boundary nodes, as illustrated in Figure 2.8 is considered. The virtual displacement field δv is approximated using shape functions $\mathbf{N}_n(\mathbf{x})$:

$$\delta\mathbf{v} = \sum_n \delta\mathbf{v}_n \mathbf{N}_n(\mathbf{x}) \tag{2.40}$$

When substituting the strain ϵ for the tensor product of the displacement $\epsilon = \nabla \otimes \delta\mathbf{v}_n$, the system of equations has full rank. This relation can be simplified for linear problems by formulating the force vector $\mathbf{F}$, which size is the product of problem dimension and number of vertices, as the matrix product of the stiffness matrix $\mathbf{K}$ and the displacement vector $\delta\mathbf{v}$ [20]

$$\mathbf{K} \cdot \delta\mathbf{v} = \mathbf{F}. \tag{2.41}$$

In further sections, the residual form of this equation is used and the deformation vector $\mathbf{u}$ is defined as the structural state $\mathbf{y}_S$ for consistency

$$\mathbf{R}_S = \mathbf{K} \cdot \mathbf{y}_S - \mathbf{F}_S = \mathbf{0}. \tag{2.42}$$

2.2.3 Interpolation

For simulating the structural deformation of a wing, the aerodynamic forces have to be mapped onto the structural mesh. Vice versa, to obtain the outer shape for aerodynamic simulation, the calculated deformations of the structural mesh have to be transfered to the *CFD* mesh. In general terms, the structural displacements present on the coupling surface of the structure mesh $\mathbf{u}_{\partial S}$ are mapped on the aerodynamic mesh to form the coupling surface $\mathbf{u}_{\partial M}$ by the coupling matrix $\mathbf{M}$

$$\mathbf{u}_{\partial M} = \mathbf{M}\mathbf{u}_{\partial S} \tag{2.43}$$

This relationship fulfills the energy conservation condition, if the structural forces $\mathbf{f}_{\partial S}$ are the transposed $\mathbf{M}^T$ multiplied by the aerodynamic forces $\mathbf{f}_{\partial F}$

$$\mathbf{f}_{\partial S} = \mathbf{M}^T\mathbf{f}_{\partial F}. \tag{2.44}$$

To obtain this desired interpolation matrix $\mathbf{M}$, Schuster et al. [21] have presented a method based on the *moving-least-squares* (*MLS*) method. In their approach, the coupling matrix $\mathbf{M}$ is deconstructed rowwise into linear polynomials, representing an aerodynamic surface mesh point $\mathbf{x}_j$

$$\mathbf{m}_j = \mathbf{p}(\mathbf{x}_j)^T \cdot \alpha_j \tag{2.45}$$

$$\mathbf{p}(\mathbf{x}_j)^T = [1, x_j, y_j, z_j] \tag{2.46}$$

The *MLS* optimization approach is now used to obtain the polynomical coefficients α_j for each mesh point $\mathbf{x}_j$ by defining a quadratic error functional J_j

$$J_j(\mathbf{x}_{F,j}, \mathbf{X}_S) = \sum_{i=1}^{N} \phi(\mathbf{x}_{F,j}, \mathbf{x}_{S,i})(\mathbf{p}(\mathbf{x}_{S,i})^T\boldsymbol{\alpha}_j - u_{S,i})^2, \tag{2.47}$$

where σ is the $\mathcal{C}^2$ radial basis function (*RBF*) by Wendland

$$\phi(\mathbf{x}_{F,j}, \mathbf{x}_{S,i}) = (\max(0, 1-r))^4(4r+1) \quad \text{with} \quad r = \frac{\|\mathbf{x}_{F,j} - \mathbf{x}_{S,i}\|_2}{\delta_j}. \tag{2.48}$$

Solving this optimization problem results in the polynomial factors, from which the interpolation matrix $\mathbf{M}$ can be constructed. The factor δ_j in the *RBF* is called the support radius and defines the radius around the point $\mathbf{x}_j$, in which N_δ mesh nodes have influence on the deformation of the node at position $\mathbf{x}_j$. The authors suggest a value for $30 - 100$ for the number of influential nodes N_δ.

2.2.4 Mesh Deformation

The *MLS* method provides a solution to transfer the deformations of the structure coupling surface $\mathbf{u}_{\partial S}$ to the aerodynamic boundary ∂M. To obtain a deformed CFD mesh, the volume mesh inside the fluid has to be deformed accordingly. Remeshing the aerodynamic model after each geometric change is theoretically possible, but causes changes in the discretization error, adding noise to the quantities of interest [22]. Thus, an algorithm for deforming the existing volume mesh is favourable. A robust mesh deformation method is proposed by Dwight [23] using an analogy from structure mechanics, where the volume mesh is considered to be a solid body. The problem is formulated analogous to the linear elastic static problem in Section 2.2.2, with a *Dirichlet* boundary condition on the surface mesh points. In residual form, the equation is

$$\mathbf{R}_M = \mathbf{K}_M \cdot \mathbf{y}_M - \mathbf{M}_S = \mathbf{0}, \tag{2.49}$$

where $\mathbf{K}_M$ denotes the stiffness matrix, $\mathbf{y}_M$ the mesh state and $\mathbf{M}_S$ the boundary points on the aerodynamic surface mesh. This system then is solved for the given finite element discretization. In [23], the authors show that this technique is robust even for large displacements, such as deployment of high-lift devices.

2.2.5 Vortex-Lattice Method

While the methods explained in Section 2.2.1 allow high-fidelity computations of the aerodynamic properties of an aircraft, they are computationally expensive. For some applications, like calculating the aerodynamic loads, this high-fidelity is not required. For these applications, the *VLM* can be applied, requiring less computational resources. Similarly to the assumptions in Section 2.1.2, for the *VLM* inviscid, incompressible flow is assumed. In Section 2.1.2, Prandtl's lifting line theory is used to calculate the lift of a straight wing. However, this theory can not provide an accurate load prediction for more complex wing shapes, like swept wings or wings with a dihedral angle. Similarly to the lifting line theory, the *VLM* uses horseshoe-vortices along the wing span and the chord line, forming a lattice of horseshoe vortices on the wing. Per definition, the trailing vortices are continued to infinity. This lattice forms N trapezoidal panels, in each panel a control point is placed at the $\frac{3}{4}c$ location. Figure 2.9 shows this panel lattice for $N = 8$ panels. The induced velocity at a specific control point m by the horseshoe vortex n is the sum of induced velocities from the bound and the trailing vortices of the horseshoe n. The velocity $\mathbf{V}$ at each vortex segment is calculated by the integral of *Biot-Savart* law

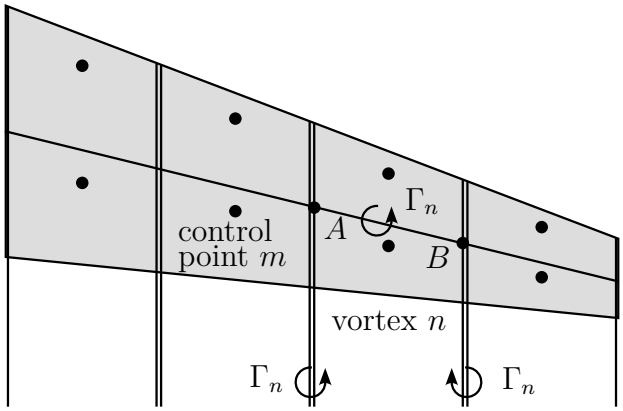


Figure 2.9: Panel lattice on a swept wing with $N = 8$ panels

(eq. (2.6))

$$\vec{V} = \frac{\Gamma_n}{4\pi} \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|^2} \left[\mathbf{r}_0 \cdot \left(\frac{\mathbf{r}_1}{r_1} - \frac{\mathbf{r}_2}{r_2} \right) \right]. \tag{2.50}$$

For the bound vortex segment, these values are inserted for r_0, r_1, and r_2:

$$\mathbf{r}_0 = (x_{2n} - x_{1n})\hat{i} + (y_{2n} - y_{1n})\hat{j} + (z_{2n} - z_{1n})\hat{k} \tag{2.51}$$

$$\mathbf{r}_1 = (x - x_{1n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k} \tag{2.52}$$

$$\mathbf{r}_2 = (x - x_{2n})\hat{i} + (y - y_{2n})\hat{j} + (z - z_{2n})\hat{k}. \tag{2.53}$$

For the trailing vortices, the formulas are shortened, as the x component approaches $-\infty$: To obtain the total induced velocity at the control point m, the velocities induced by all $2N$ vortices are summed. Using these relations, the induced velocity can be computed at a control point b from Γ of the vortices. For solving for the circulation (Γ), the wing's surface is defined as a streamline of the flow. From this boundary condition, a tangency relation using the mean camber slope (δ) and dihedral angle (ϕ) is derived

$$-u_m \sin\delta \cos\phi - v_m \cos\delta \sin\phi + w_m \cos\phi \cos\delta + V_\infty \sin(\alpha - \delta) \cos\phi = 0. \tag{2.54}$$

By inserting the induced velocity components for each of the M control points into the equation, the resulting system of M equations can be solved for each Γ_n, as $N = M$. The lift distribution can be interpolated from the values for Γ and coefficient of lift (C_L) and $C_{D,i}$ can be evaluated using the methods from Section 2.1.2.

2.2.6 Reduced-Order-Modeling (ROM)

The previous sections all dealt with in-the-loop numerical techniques. However, certain applications may prefer to use pre-recorded data instead. For example, reduced-order-

models (*ROM*s) techniques are used in MDO processes to replace a CAD-in-the-loop system. This section explains the mathematical background of *ROM*s using the proper-orthogonal-decomposition (*POD*) method. How this technique is implemented within the MDO process to replace a CAD model is detailed in Section 3.1.3.

To create a *ROM*, initial sample data is required. These data samples $\mathbf{x}_i \in \mathbb{R}^{m\times 1}$ are collected at points $\mathbf{p}^j$ of the input space and composed into the matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1(\mathbf{p}^1) & \mathbf{x}_2(\mathbf{p}^2) & \cdots & \mathbf{x}_n((\mathbf{p}^n)) \\ | & | & & | \end{bmatrix}. \tag{2.55}$$

The matrix $\mathbf{X}$ is decomposed with the singular value decomposition (*SVD*) into

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{2.56}$$

The matrix $\mathbf{U} \in \mathbf{R}^{m\times m}$ contains the left singular vectors of $\mathbf{X}$, these vectors are the orthonormal eigenvectors of the matrix $\mathbf{X}\mathbf{X}^T$, while the matrix $\mathbf{V}$ contains the right singular vectors of $\mathbf{X}$, i.e. the eigenvectors of $\mathbf{X}^T\mathbf{X}$. Within Σ, the singular values of X σ are on the main diagonal. The singular values σ are the square roots of the eigenvalues of the matrices $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$, there are as many non-zero entries as the rank l of $\mathbf{X}$. Their order on the main diagonal is sorted by their magnitude, so $\sigma_1 > \sigma_2 > ... > \sigma_l$.

Due to their sorting, the first entries of $\mathbf{\Sigma}$ are the most important for $\mathbf{X}$. These most important values σ correspond to a number of columns in $\mathbf{U}$, called the POD modes $\mathbf{\Phi} \in \mathbf{R}^{m\times r}$, $r << n$. The number of POD modes is truncated to the r most significant modes. If the original relationship was $\mathbf{x}(\mathbf{p})$, it can be transformed into the POD coefficients $\mathbf{a}(\mathbf{p})$ using a *Galerkin* projection onto the POD subspace $\mathbf{\Phi}$.

$$\mathbf{a}(\mathbf{p}) = \mathbf{x}(\mathbf{p})^T\mathbf{\Phi}. \tag{2.57}$$

The data $\mathbf{x}(\mathbf{p})$ at the data point $\mathbf{p}$ can now be evaluated using a lower dimensional representation

$$\mathbf{x}(\mathbf{p}) \approx \sum_{i=1}^{r} a_i(\mathbf{p})\phi_i. \tag{2.58}$$

2.3 Computational Differentiation

For optimization purposes, which are investigated in Section 2.4, computational methods to obtain the gradients of the different disciplines are required. Computing the gradients of large-scale problems can be costly, it is therefore of major importance to apply the correct techniques for computing the gradients of an optimization problem. Within this section, the methods of finite differences (Section 2.3.1) and the direct and adjoint implicit analytic methods (Section 2.3.2) are explained. In Section 2.3.3, the adjoint method is used to compute the gradients of a coupled aerostructural problem.

2.3.1 Finite Differences

In this section, the simplest technique for numerically computing gradients, the finite difference method is explained. The computational effort of this technique is $\mathcal{O}(n)$ where n is the number of input parameters. The idea is to pertube the function $f(x)$ by a small value ϵ in the direction e_i. This is defined as a one-sided approximation of the derivative:

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + \epsilon e_i) - f(x)}{\epsilon} \tag{2.59}$$

This approximation has an error smaller than the maximum value of the maximum second derivative of f in the bounds of $(x, x + \epsilon)$ times $\epsilon/2$

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x)}{\epsilon} \leq ||\nabla^2 f(\cdot)|| \cdot \frac{\epsilon}{2} \tag{2.60}$$

By approximating the function with a polynomial of second order, the derivative can be approximated to second order. The formula for the central derivative, where the function is perturbed by $-\epsilon$ and ϵ is

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} \tag{2.61}$$

This approximation is usually more costly to compute, as two perturbations of the function $f(x)$ have to be evaluated, however, the error is decreased to a term of third order:

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} \leq ||\nabla^3 f(\cdot)|| \cdot \frac{\epsilon^2}{3}. \tag{2.62}$$

The selection of the step size ϵ involves a fundamental trade-off. While decreasing ϵ reduces the truncation error shown in eq. (2.60) and eq. (2.62), it simultaneously amplifies

the impact of floating-point arithmetic errors. The total error in the central difference approximation, accounting for both truncation and roundoff errors, is bounded by:

$$\frac{\partial f}{\partial x_i}(x) = \frac{f(x + \epsilon e_i) - f(x - \epsilon e_i)}{2\epsilon} \leq ||\nabla^3 f(\cdot)|| \cdot \frac{\epsilon^2}{3} + \frac{h}{\epsilon}. \tag{2.63}$$

2.3.2 Direct and Adjoint Implicit Analytic Methods

While the finite-difference method explained earlier is applicable to any type of function, whether or not it is exposed to the code, the underlying equations are often known. Thus it is desirable to use knowledge of these equations to improve the accuracy of the resulting gradients and to reduce the computational time required. One class of methods, implemented in many tools related to aerodynamic analysis, are implicit analytical methods in direct or adjoint form. For their derivation, consider a set of general objective functions $\mathbf{f} : \mathbb{R}^{n+m} \to \mathbb{R}^f$, which computes a scalar objective value from the vector of design variables $\mathbf{x} \in \mathbb{R}^n$ and a number of states $\mathbf{y} \in \mathbb{R}^m$. The states $\mathbf{y}$ are an implicit function of the design variables $\mathbf{x}$, which can be expressed in their residual form

$$\mathbf{R}(\mathbf{y}; \mathbf{x}) = \mathbf{0}. \tag{2.64}$$

This relation could be, for example, the discretized *RANS* equations (eqs. (2.28) and (2.29)) for aerodynamic analysis. We can obtain the total derivative of the objective function $\frac{df}{d\mathbf{x}}$ from the partial derivative of the function with respect to the design variables $\frac{\partial f}{\partial \mathbf{x}}$ and the partial derivative of the function with respect to the states $\frac{\partial f}{\partial \mathbf{y}}$ multiplied by the total derivative of the states by the design variables $\frac{d\mathbf{y}}{d\mathbf{x}}$.

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{y}} \frac{d\mathbf{y}}{d\mathbf{x}}. \tag{2.65}$$

The total derivative of the states with respect to the design variables can be obtained by applying the same partial differentiation scheme to the residual equation (eq. (2.64))

$$\frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{y}} \frac{d\mathbf{y}}{d\mathbf{x}} = \mathbf{0}, \tag{2.66}$$

and rearrange the terms so that

$$\frac{d\mathbf{y}}{d\mathbf{x}} = -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{y}}\right)^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.67}$$

By reinserting this relation into eq. (2.65), the total derivative $\frac{df}{d\mathbf{x}}$ becomes a function of only partial derivative terms

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{y}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{y}} \right)^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.68}$$

Assuming that this equation is directly solvable, which is unlikely due to the existence of the inverse, a computational advantage over the finite difference method is the total derivative requires only partial derivatives, thus avoiding the often costly recalculation of the states $\mathbf{y}$. To actually solve eq. (2.68) two methods are available, the direct or the adjoint method. Both methods solve a system of equations to remove the inverse of eq. (2.68), but the terms chosen are different, as illustrated in Figure 2.10.

In the direct method, the total derivative of the states is solved by rearranging eq. (2.67) and solving the system of equations for the total derivative $d\mathbf{y}/\mathbf{x}$.

$$-\left(\frac{\partial \mathbf{R}}{\partial \mathbf{y}} \right) \frac{d\mathbf{y}}{d\mathbf{x}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.69}$$

The computational cost of this operation also scales linearly with the number of design variables ($\mathcal{O}(n)$), but probably with a lower proportional constant than for finite differences, since state recomputation is spared. Obtaining the total derivative of the objective function then consists only of substituting the total derivative back into eq. (2.65). This operation is easy to compute since no solver is needed.

For the adjoint method, the adjoint state vector $\bar{\mathbf{y}}$ is defined as the transposed of the solution of the system of equations highlighted in Figure 2.10, so that the adjoint vector can be solved from the *adjoint equation*.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{y}}^T \bar{\mathbf{y}} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}^T \tag{2.70}$$

The computations required to solve the adjoint equations are not dependent on the number of input variables, but depends on the number of objective functions. Thus the computational effort scales with $\mathcal{O}(f)$. In many practical cases, the number of objective functions is much smaller than the number of inputs, making the *adjoint* method the preferred way to solve the analytic derivatives for aerodynamic optimization. Again, the total derivative of the objective function only requires reinserting the adjoint state into eq. (2.65) and calculating the result by matrix multiplication and subtraction

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} - \bar{\mathbf{y}}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.71}$$

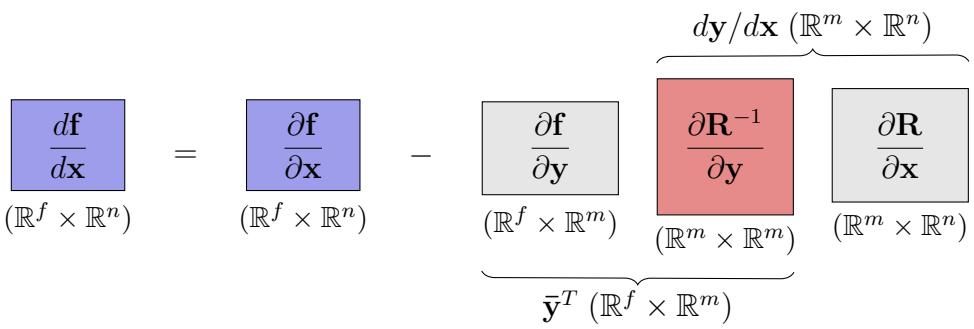


Figure 2.10: Visualization of the total derivative equation (eq. (2.68)) and the differential choice of the direct and adjoint state variables [24]

2.3.3 Coupled Aero-Structural Adjoint

The implicit *adjoint* analytical method for calculating the gradients of an objective function of interest, explained in the previous section, is so far applicable only to monodisciplinary analyses. However, in the case of a coupled aero-structural optimization, the gradients of interlinked processes must be computed through the coupled aero-structural *adjoint*. The mathematical background of this coupling is explained based on the work of Volle [8], without discussing further details of the implementation.

In the case of coupled aero-structure optimization, the objective functions of interest $\mathbf{f}$ are typically the aerodynamic performance coefficients (e.g., C_l, C_d, C_{mx}) and/or the structural mass m. The objective is modified by the design variables x and the states y, further separated into the aerodynamic state $\mathbf{y}_F$, the structural state $\mathbf{y}_S$, and the mesh state $\mathbf{y}_M$. The total derivative equation shown in eq. (2.65) is now the set of equations

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{y}_F} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{y}_M} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{y}_S} \end{bmatrix}^T \begin{bmatrix} \frac{d\mathbf{y}_F}{d\mathbf{x}} \\ \frac{d\mathbf{y}_M}{d\mathbf{x}} \\ \frac{d\mathbf{y}_S}{d\mathbf{x}} \end{bmatrix}. \tag{2.72}$$

Using the *adjoint* method to compute the derivatives is advantageous over the *direct* method, if there are few objective quantities of interest but likely many design parameters. For the coupled problem, the aerodynamic *adjoint* $\bar{\mathbf{y}}_F$, the structure *adjoint* $\bar{\mathbf{y}}_S$ and the mesh *adjoint* $\bar{\mathbf{y}}_M$ form a system of equations consisting of matrices and vectors.

$$\begin{bmatrix} \frac{\partial \mathbf{R}_F}{\partial \mathbf{y}_F} & \frac{\partial \mathbf{R}_F}{\partial \mathbf{y}_M} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{R}_M}{\partial \mathbf{y}_M} & \frac{\partial \mathbf{R}_M}{\partial \mathbf{y}_S} \\ \frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_F} & \frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_M} & \frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_S} \end{bmatrix}^T \begin{bmatrix} \bar{\mathbf{y}}_F \\ \bar{\mathbf{y}}_S \\ \bar{\mathbf{y}}_M \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \end{bmatrix}^T. \tag{2.73}$$

Since the structural state $\mathbf{y}_S$ has no direct effect on the aerodynamic residual $\mathbf{R}_F$ and the mesh residual $\mathbf{R}_M$ is not directly influenced by the fluid state $\mathbf{y_F}$, these terms are $\mathbf{0}$. The total derivative can then be calculated from the *adjoint* vectors as

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \begin{bmatrix} \bar{\mathbf{y}}_F \\ \bar{\mathbf{y}}_S \\ \bar{\mathbf{y}}_M \end{bmatrix}^T \begin{bmatrix} \frac{\partial \mathbf{R}_F}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{R}_S}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{R}_M}{\partial \mathbf{x}}. \end{bmatrix} \tag{2.74}$$

The terms $\frac{\partial \mathbf{R}_M}{\partial \mathbf{y}_S}$ and $\frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_F}$ are refered to as a cross derivatives, as the structure state only influences the aerodynamic performance through the in-flight state. However, Volle et al. [8] show that their availability significantly improves the outcome of simultaneous shape optimization and structure sizing processes.

2.4 Optimization Algorithms

In this section of the theory introduction, the theory behind optimization algorithms is explained [25]. The relevant optimization algorithms can be divided into two main groups, gradient-based and gradient-free optimization algorithms. Typically, when gradient information is provided, as with the methods in the previous section, gradient-based optimization is the more feasible approach for large-scale optimization problems. However, in cases where gradient information is not available or too costly, the use of gradient-free algorithms may be the only choice. We introduce both classes of algorithms with relevant example algorithms,

We use the sequential quadratic programming (*SQP*) and *Interior-Point* (*IP*) methods to illustrate the mathematical background of many gradient-based algorithms. For the gradient-free optimization algorithms, the *Constrained Optimization by Linear Approximation* (*COBYLA*) algorithm is presented as an example.

Starting from this introduction, the optimization problem is considered general nonlinear programming problem under equality and inequality constraints h and g, with the objective function $f(\mathbf{x})$ to be minimized

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0, \\ & \mathbf{h}(\mathbf{x}) = 0, \\ & \mathbf{x} \in \mathbb{R}^n. \end{aligned}$$

Considering only the unconstrained problem, a strict local optimum $\mathbf{x}^*$ is found if the gradient of the objective function $\nabla f(\mathbf{x}^*) = \mathbf{0}$ and the *Hessian* $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite. This sufficient condition only qualifies $\mathbf{x}^*$ as a local optimum of $f(\mathbf{x})$. If $f(\mathbf{x})$ is also convex, the local optimum is also the global optimum of the function. But when the problem is not convex, it is often not easy to decide whether a local optimum is a global optimum.

Streuber et al. found in [26] that the risk of multimodality (the presence of multiple local optima) is low for aerostructure optimizations at the detailed design level. However, for preliminary design problems with more geometric degrees of freedom, the risk of multimodality is significantly higher.

For the determination of the optimum of a constrained problem, further considerations

are necessary in order to include the fulfillment of the constraint in the condition. To determine the optimality of the constrained problem, the *Lagrangian* function combines the equality constraints $h(\mathbf{x})$ and the *Active Set* (the set of inequality constraints that are close to the constraint value) of the inequality constraints $(\mathcal{A}(\mathbf{x})|g = 0)$ with the objective function $f(\mathbf{x})$. By definition, a constraint is active if there is no distance to the boundary of the feasible subspace. This can be expressed with the slack parameter s, so that $g_j + s_j^2 = 0$. The *Lagrangian* function for this kind of problem is

$$\mathcal{L} = \mathbf{f}(\mathbf{x}) + \lambda_h^T \mathbf{h}(\mathbf{x}) + \lambda_g^T (\mathbf{g}(\mathbf{x}) + \mathbf{s} \odot \mathbf{s}) \tag{2.75}$$

Similar to the unconstrained case, a necessary condition for a local optimum is a stationary point of the function. The *Lagrangian* function, however, requires more partial derivatives:

$$\begin{aligned} \nabla_x \mathcal{L} &= \nabla \mathbf{f} + (\nabla \mathbf{h})^T \boldsymbol{\lambda}_h + (\nabla \mathbf{g})^T \boldsymbol{\lambda}_g = \mathbf{0} \\ \nabla_{\lambda \mathbf{h}} \mathcal{L} &= \mathbf{h} = \mathbf{0} \\ \nabla_{\lambda_g} \mathcal{L} &= \mathbf{g} + \mathbf{s} \odot \mathbf{s} = \mathbf{0} \\ \nabla_s \mathcal{L} &= \boldsymbol{\lambda_g} \odot \mathbf{s} = \mathbf{0} \end{aligned} \tag{2.76}$$

The conditions in eq. (2.76) are called *Karush-Kuhn-Tucker* conditions (KKT). The second-order sufficient conditions are

$$\begin{aligned} \mathbf{p}^T \nabla^2 \mathbf{p} > \mathbf{0} \quad \forall \mathbf{p} : \\ \nabla \mathbf{h} \mathbf{p} = \mathbf{0} \\ \nabla \mathbf{g} \mathbf{p} \leq \mathbf{0} \end{aligned} \tag{2.77}$$

2.4.1 Sequential quadratic programming (*SQP*) Method

A numerical method to solve these non-linear, constrained optimization problems is the *SQP* method [25]. Its concept is part of many open-source and commercial optimization algorithms like *SLSQP* [27] and *NLPQL* [28]. The idea behind the *SQP* is to model the optimization problems as a sequence of local, quadratic optimization problems. Quadratic problems have the desirable quality of convexity, making them easily solvable by algorithms. We formulate a quadratic optimization problem as

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \nabla^2 \mathcal{L} + \mathbf{q}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{b} \leq 0, \\ & \mathbf{C}\mathbf{x} + \mathbf{d} \leq 0 \end{aligned} \tag{2.78}$$

An iterative approach is desired to solve this problem. At every iteration of the solution, the optimizer takes a step $\mathbf{p}$, so that $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}$. As in the definition of the *KKT* conditions, the inequality constraints are converted into equality constraints by the *Working Set* of constraints as

$$\mathbf{C}_w = \mathbf{x}_k + \mathbf{d}_w = \mathbf{0}. \tag{2.79}$$

The *Working Set* must not be equivalent to the *Active Set*, as there is no necessity for *Working Set* constraints to be active. We can reformulate the initial definition of the quadratic problem from eq. (2.78) into:

$$\begin{aligned} \min \quad & \frac{1}{2}(\mathbf{x}_k + \mathbf{p})^T \nabla^2 \mathcal{L} + \mathbf{q}^T(\mathbf{x}_k + \mathbf{p}) \\ \text{subject to} \quad & \mathbf{A}(\mathbf{x}_k + \mathbf{p}) + \mathbf{b} \leq \mathbf{0}, \\ & \mathbf{C}_w(\mathbf{x}_k + \mathbf{p}) + \mathbf{d}_w = \mathbf{0} \end{aligned} \tag{2.80}$$

As $\mathbf{x}_k$ is the initial point of the optimization iteration, it is invariant, thus it can be removed from the optimization formulation and the *KKT* solution can be calculated as follows

$$\begin{bmatrix} \nabla^2 \mathcal{L} & \mathbf{A}^T & \mathbf{C}_w^T \\ \mathbf{A} & 0 & 0 \\ \mathbf{C}_w & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \lambda_h \\ \lambda_g \end{bmatrix} = \begin{bmatrix} -\mathbf{q} - (\nabla^2 \mathcal{L})^T \mathbf{x}_k \\ b \\ d \end{bmatrix}. \tag{2.81}$$

The solution of this system returns the search direction $\mathbf{p}$, however, a suitable magnitude for this step must still be determined. This can be done by means of a *Line Search* using a so-called merit function to determine the quality of an optimization step by means of its objective function and constraint violations. Commonly used merit functions use a norm to combine the objective function value and the constraint violations $\bar{\mathbf{g}}(\mathbf{x})$.

$$\boldsymbol{\phi}_1(\mathbf{x}, \mu) = \mathbf{f}(\mathbf{x}) + \mu \|\bar{\mathbf{g}}(\mathbf{x})\|_n. \tag{2.82}$$

A step α is considered as acceptable by the merit function if it fulfills the *Armijo* condition:

$$\boldsymbol{\phi}_1(x_k + \alpha_k \mathbf{x}_k; \mu_k) \leq \boldsymbol{\phi}_1(\mathbf{x}_k; \mu_k) + \eta \alpha_k D\boldsymbol{\phi}_1(\mathbf{x}_k; \mu_k; \mathbf{p}_k), \quad \eta \in (0, 1), \tag{2.83}$$

with $D\boldsymbol{\phi}_1(\mathbf{x}_k; \mu_k; \mathbf{p}_k)$ being the directional derivative of the merit function in the descent direction p_k:

$$D\boldsymbol{\phi}_1(\mathbf{x}_k; \mu; \mathbf{p}_k) = \nabla \mathbf{f}_k^T \mathbf{p}_k - \mu \|\mathbf{g}(\mathbf{x})|_1. \tag{2.84}$$

Another concept for determining the next step is the *Trust-Region* method, that constrains a trusted norm of the local function approximation and finds the miniumum within this trust region. The *Hessian* of the *Lagrangian* function $\nabla^2 \mathcal{L}$ is costly to compute in most cases. Thus, approximations like the *Quasi-Newton* methods are commonly used in

numerical optimizations. The method considered most effective for these approximations is the *Broyden-Fletcher-Goldfarb-Shanno* method (*BFGS*), which approximates the *Hessian* through its inverse. Inverting the formula back to obtain the direct *Hessian* approximation $\mathbf{B}_k$ results in:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} \tag{2.85}$$

$$\mathbf{s}_k = \alpha_k \mathbf{p}_k \quad \mathbf{y}_k = \nabla_x \mathcal{L}_{k+1} - \nabla_x \mathcal{L}_k \tag{2.86}$$

2.4.2 Interior-Point Methods

Another class of important algorithms for large-scale nonlinear optimization are the *Interior Point* methods. Like the *SQP* method, they can solve constraint nonlinear problems, but have distinct advantages and disadvantages. Generally speaking, *Interior Point* are considered more efficient for large-scale problems, where as *SQP* methods perform better for medium-scale problems [25]. For the interior point method, slack parameters $\mathbf{s}$ are used converting the inequality constraints to equality constraints and add a penalty to the objective function.

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) - \mu_b \sum_{j=1}^{n_s} \ln s_j \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) + \mathbf{s} = \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned} \tag{2.87}$$

The *Lagrangian* function for this problem is

$$\mathcal{L} = \mathbf{f}(\mathbf{x}) - \mu_b e^T ln(\mathbf{s}) + \mathbf{h}(\mathbf{x})^T \boldsymbol{\lambda}_h + (\mathbf{g}(\mathbf{x}) + \mathbf{s})^T \boldsymbol{\lambda}_g \tag{2.88}$$

To approch the original constrained problem, the problem must be solved in a sequence, where $\mu_b \to 0$. As previously done for the *SQP* method, the *KKT* conditions are defined in a matrix form

$$\begin{bmatrix} \nabla^2 \mathcal{L}_k & \nabla \mathbf{h}^T & \nabla \mathbf{g}^T & 0 \\ \nabla \mathbf{h} & 0 & 0 & 0 \\ \nabla \mathbf{g} & 0 & 0 & \mathbf{I} \\ 0 & 0 & \mathbf{I} & \mathbf{S}^{-1}\Sigma \end{bmatrix} \begin{bmatrix} \mathbf{p_x} \\ \mathbf{p}_{\lambda h} \\ \mathbf{p}_{\lambda g} \\ \mathbf{p}_s \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L} \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) + \mathbf{s} \\ \boldsymbol{\lambda}_g - \mu \mathbf{S}^{-1} e \end{bmatrix} \tag{2.89}$$

The matrix $\mathbf{S}$ is a diagonal matrix of the slack parameters $\mathbf{s}$. This matrix formulation differs to the *KKT* matrix of the *SQP* method by the fact that its size remains constant, thus

improving computational efficiency. However, the gradients of all constraints are required for each iteration of the *KKT* system. Similarly to the *SQP* method, a *Quasi-Newton* approximation of the *Hessian* of the *Lagrangian* can be used. The step size can also be set with a *Line-Search* [24].

2.4.3 Gradient-Free Optimization Algorithm *COBYLA*

Using a gradient-free optimizer can be advantageous if gradient computation is not feasible. Within this introduction, the *COBYLA* algorithm developed by Powell [29] is explained. We again consider a nonlinear optimization problem subject to inequality constraints of the known notation

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) < \mathbf{0}. \end{aligned}$$

We start the optimization at the initial point x_0 and define a merit function similar to the merit function for the SQP algorithm, with the only differences, that the $\mathcal{L}_\infty$ norm is used and the term containing the constraints is set to 0, if none of the constraints are violated. We note this operation using the *Föppl* bracket $\langle ... \rangle$.

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mu \left\langle \max\{g_i(\mathbf{x}) : i = 1, 2, \ldots, m\} \right\rangle \tag{2.90}$$

To start the optimization sequence, the optimizer needs an inital set of datapoints and their repective function value $f(x)$. This dataset contains exactly as many datapoints as there are design variables x_i and is called the simplex, with each datapoint refered to as a vertex of this simplex. The optimal vertex of this initial simplex is noted as $\mathbf{x}^{(0)}$. For the generation of the next step of $\mathbf{x}$, the algorithm optimizies the linearized problem for $\mathbf{x}^{(*)}$

$$\begin{aligned} \mathbf{f}(\mathbf{x}^{(*)}) &= \mathbf{f}(\mathbf{x}^{(0)}) + \mathbf{m}_f^T(\mathbf{x}^{(*)} - \mathbf{x}^{(0)}) \\ \mathbf{g}(\mathbf{x}^{(*)}) &= \mathbf{g}(\mathbf{x}^{(0)}) + \mathbf{m}_g^T(\mathbf{x}^{(*)} - \mathbf{x}^{(0)}). \end{aligned}$$

This new point $\mathbf{x}^{(*)}$ has to lie inside a trust region that regulates the maximum step size. Its first trust region radius is specified by the user as ρ_{beg}, so is the target ρ_{end} which terminates the optimization when the updates of $\mathbf{x}$ reach a sufficiently small step size. If the new point lies within the trust-region, the optimizer adjusts the trust-region radius ρ and increases the merit parameter μ. After this increase, the simplex is revised, whether the previous $\mathbf{x}^{(0)}$ is still the optimal vertex of the simplex. If the merit parameter is large enough, the values of the original target function and constraints are computed with the

new values for $\mathbf{x}$. This process is iterated until the reductions are small enough. If the trust-region radius is below the targeted ρ_{end}, the process is terminated. The detailed algorithm like described with in [29] is displayed in algorithm 1. For simplication purposes, these descriptions do not go into detail on the checking of simplex acceptability and simplex updating procedures.

Algorithm 1 *COBYLA* [29]

Require:

Initial point $\mathbf{x}^{(0)}$
Initial trust region radius ρ_{beg}
Final trust region radius ρ_{end}
Objective function $\mathbf{f}(\mathbf{x})$
Constraint functions $\mathbf{g}(\mathbf{x}) \geq \mathbf{0}$
Optimized solution $\mathbf{x}^*$
$\rho \leftarrow \rho_{\text{beg}}$
$\mu \leftarrow 0$
Create initial simplex: $\mathbf{x}^{(j)} = \mathbf{x}^{(0)} + \rho_{\text{beg}}\mathbf{e}_j, j = 1, \ldots, n$
Order vertices so $\mathbf{x}^{(0)}$ is best point
while $\rho > \rho_{\text{end}}$ **do**
    Check simplex acceptability
    if simplex is not acceptable **then**
        $\mathbf{x}^{(\Delta)} \leftarrow \mathbf{x}^{(0)} \pm \gamma\rho\mathbf{u}^{(\ell)}$
        Update simplex with $\mathbf{x}^{(\Delta)}$
    end if
    Solve linear subproblem for $\mathbf{x}^{(*)}$:
      $\min \mathbf{f}(\mathbf{x}^{(0)}) + \mathbf{m}_f^T(\mathbf{x} - \mathbf{x}^{(0)})$
      s.t. $\mathbf{g}(\mathbf{x}^{(0)}) + \mathbf{m}_g^T(\mathbf{x} - \mathbf{x}^{(0)}) \geq 0$
      $\|\mathbf{x} - \mathbf{x}^{(0)}\|_2 \leq \rho$
    Calculate merit function:
    $\mathbf{\Phi}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mu \langle \max\{-g_i(\mathbf{x}) : i = 1, \ldots, m\}\rangle$
    if $\|\mathbf{x}^{(*)} - \mathbf{x}^{(0)}\|_2 < \frac{1}{2}\rho$ or poor reduction **then**
        if $\rho > 3\rho_{\text{end}}$ **then**
            $\rho \leftarrow \rho/2$
        else
            $\rho \leftarrow \rho_{\text{end}}$
        end if
        Update μ if needed:
        $I \leftarrow \{i : g_i^{(\min)} < \frac{1}{2}g_i^{(\max)}\}$
        if $I \neq \emptyset$ **then**
            $\mu_{\text{new}} \leftarrow \frac{\max_j \mathbf{F}(\mathbf{x}^{(j)}) - \min_j \mathbf{F}(\mathbf{x}^{(j)})}{\min\{[g_i^{(\max)}]_+ - g_i^{(\min)} : i \in I\}}$
        end if
    else
    end if
end while
    return $\mathbf{x}^{(0)}$ or $\mathbf{x}^{(*)}$ (whichever is better)

2.5 MDO Formulations

This section introduces different algorithms, refered to as MDO formulations, to transform multidisciplinary optimization problems into nonlinear optimization problems, that can be solved with optimization algorithms. In a multidisciplinary problem, the vector of design variables $\mathbf{x} \in \mathbb{R}^n$ can be seperated into a shared design variable vector $\mathbf{x}_0 \in \mathbb{R}^{n0}$, and local design variable vectors $\mathbf{x}_i \in \mathbb{R}^{ni}$, belonging to discipline i.

Disciplines in *MDO* are typically coupled, meaning the residual $\mathbf{R}_i$ of discipline i can depend on the disciplinary outputs $\mathbf{y}_{\neq i}$ besides the shared design variables $\mathbf{x}_0$ and local design variables $\mathbf{x}_i$

$$\mathbf{R}_i(\mathbf{x}_0, \mathbf{x}_i, \mathbf{y}_{\neq i}, \mathbf{y}_i) = \mathbf{0}\,. \tag{2.91}$$

One of the primary challenges of *MDO* is to ensure feasibility of the couplings, meaning that for a multidisciplinary problem with p disciplines

$$\forall i \in [[1, p]] \quad \mathbf{y}_i - \boldsymbol{\phi}_i(\mathbf{x}_0, \mathbf{x}_i, \mathbf{y}_{\neq i}) = \mathbf{0}\,, \tag{2.92}$$

where $\boldsymbol{\phi}_i$ is the function calculating disciplinary output y_i [30].

The objective function of the *MDO* problem is $\mathbf{f} : \mathbb{R}^{n0} \times \mathbb{R}^m \to \mathbb{R}$, that calculates a scalar objective value from the shared design variables $\mathbf{x}_0$ and the output variables $\mathbf{y}$. Additionally both equality constraint functions $\mathbf{g} : \mathbb{R}^{n0} \times \mathbb{R}^m \to \mathbb{R}^{ce}$ and inequality constraint functions $\mathbf{h} : \mathbb{R}^{n0} \times \mathbb{R}^m \to \mathbb{R}^{ci}$ may be present in the optimization.

The optimization problem can therefore be summarized as

$$\begin{aligned} \min_{\mathbf{x}_0, \mathbf{x}} \quad & \mathbf{f}(\mathbf{x}_0, \mathbf{y}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_0, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}_0, \mathbf{y}) = \mathbf{0} \\ & \forall i \in [[1, p]] \quad \mathbf{y}_i - \boldsymbol{\phi}_i(\mathbf{x}_0, \mathbf{x}_i, \mathbf{y}_{\neq i}) = \mathbf{0}\,. \end{aligned} \tag{2.93}$$

There exist two classes of generalized *MDO* formulations, monolithic formulations and distributed formulations to solve this generalized problem. While monolithic formulations have been extensively researched and have demonstrated their capability for different types of coupled optimizations, new types of distributed *bilevel* formulations have recently received attention from research and the aerospace industry. The main difference between these formulations compared to monolithic formulations is that *bilevel* formulations hier-

archically seperate the monolithic optimization problem. Their capability of interest is defined by Gazaix et al. [13] as their ability to be integrated into established, disciplinary organizational structures. Because they remain separate in disciplines, maintainability, consistency, and collaboration can be improved.

To understand the methodology behind monolithic formulations, the *MDF* formulation is introduced in Section 2.5.1.

In this thesis, three different types of *bilevel* algorithms, called the *bilevel* formulation developed by the IRT Saint Exupery (*BL-IRT*), *bilevel-block-coordinate-descent* (*BL-BCD*), and *bilevel-embedded* (*BL-EMB*) formulations are introduced in Section 2.5.2, Section 2.5.3, and Section 2.5.4.

2.5.1 *MDF* Formulation

The smallest possible formulation of the optimization problem involving multiple disciplines is achieved through the use of a multidisciplinary analysis (*MDA*). For the definition of the *MDA*, it is assumed there exist a continuosly defined function $\mathbf{\Psi}(\mathbf{x}, \mathbf{x}_0) = \mathbf{y}$, so that the coupling condition Equation (2.92) is fulfilled. The purpose of the *MDA* is to resolve the coupling function, for each step of the optimizer so that the coupling condition in Equation (2.92) is fulfilled, thereby yielding a feasible solution after each iteration of the optimization process. Consequently, this formulation is referred to as the *Multidisciplinary-Feasible* (*MDF*) formulation. The *MDF* optimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{x}_0, \mathbf{x}} \quad & \mathbf{f}(\mathbf{x}_0 \mathbf{x}, \mathbf{\Psi}(\mathbf{x}_0, \mathbf{x})) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_0, \mathbf{x}, \mathbf{\Psi}(\mathbf{x}_0, \mathbf{x})) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}_0, \mathbf{x}, \mathbf{\Psi}(\mathbf{x}_0, \mathbf{x})) = \mathbf{0}\,. \end{aligned} \tag{2.94}$$

From an engineering perspective, an *MDA* can be interpreted as the integration of multiple disciplines into a unified, internally coupled discipline. The specific algorithm chosen to solve the *MDA* governs the execution sequence of the disciplines within the MDA. In the extended design structure matrix (*XDSM*) diagram of the *MDF* formulation in Figure 2.11, the *MDA* is solved using the *Gauss-Seidel* fixed-point algorithm. Other algorithms used to solve the *MDA* are Newton methods, such as the Newton-Raphson method, and quasi-Newton methods, such as the Broyden method. These methods tend to converge faster by using gradient information. The main advantage of the *MDF* formulation is its compactness, since the optimizer only has to control the design variables without

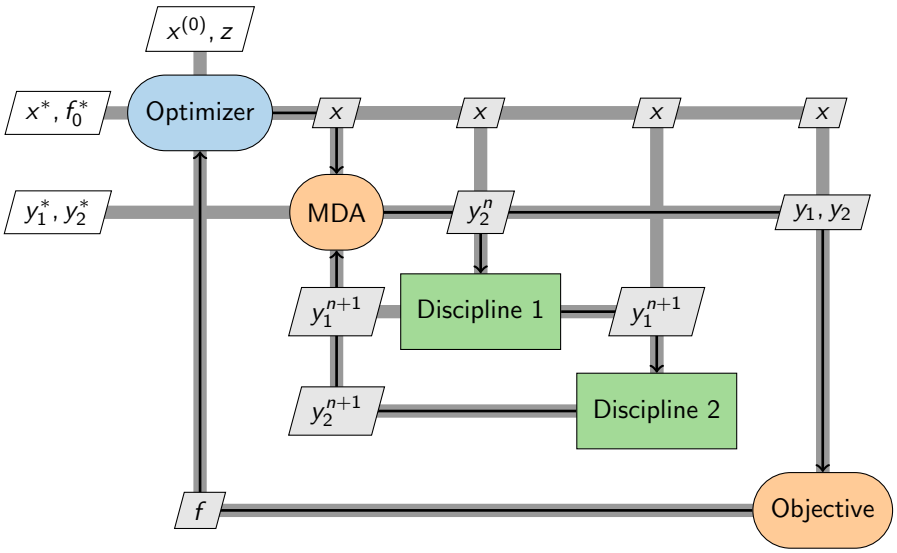


Figure 2.11: *MDF* formulation using *Gauss-Seidel* algorithm

enforcing any kind of consistency constraints. The *MDA* ensures inherent feasibility for each interation step. This can also be seen as a disadvantage, since the *MDA* must be reconverged for each iteration. Another major drawback is that not only disciplinary gradients are required, but also cross-disciplinary gradients if a gradient-based optimization process is used [31].

2.5.2 *BL-IRT* Formulation

The concept of each bilevel formulation is to recompose the optimization problem into a two-level problem, where the top-level optimizer controls the shared design variables $\mathbf{x}_0$, while the disciplinary design variables $\mathbf{x}_i$ are controlled by disciplinary optimizers. The constraints are also separated into the subset $\mathbf{g}_0$ of constraints that depend only on the shared variables $\mathbf{x}_0$ and the disciplinary constraints $\mathbf{g}_i$ that depend on the local variables $\mathbf{x}_i$. Only the subset $\mathbf{g}_0$ of constraints that depend only on the shared variables $\mathbf{x}_0$ is enforced by the system-level optimization. An *MDF* formulation from eq. (2.94) can be reformulated into the bilevel formulation as

$$\begin{aligned} \min_{\mathbf{x}_0} \quad & \mathbf{f}(\mathbf{x}_0, \mathbf{x}^*(\mathbf{x}_0), \mathbf{\Psi}(\mathbf{x}_0, \mathbf{x}^*(\mathbf{x}_0))) \\ \text{subject to} \quad & \mathbf{g}_0(\mathbf{x}_0) \leq 0 \\ & \mathbf{h}_0(\mathbf{x}_0) = \mathbf{0}\,. \end{aligned} \tag{2.95}$$

With the lower-level optimization as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}_0, \mathbf{x}, \boldsymbol{\Psi}(\mathbf{x}_0, \mathbf{x})) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_0, \mathbf{x}, \boldsymbol{\Psi}(\mathbf{x}_0, \mathbf{x})) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}_0, \mathbf{x}, \boldsymbol{\Psi}(\mathbf{x}_0, \mathbf{x})) = \mathbf{0}\,. \end{aligned} \tag{2.96}$$

For engineering applications, the lower-level problem of the *bilevel* optimization is typically decomposed into the different engineering disciplines, such as aerodynamics and structural mechanics. Mathematically, this decomposition is expressed as a partition into p blocks $x_i \in \mathbb{R}^{ni}, i = 1, ..., p$. The decomposed lower-level problem is then solved by d optimizers.

The subspace optimizations run in parallel, with no data exchange within the optimizations. For the *BL-IRT* formulation, two *MDA* loops compute the couplings before and after each subspace optimization [30]. An *XDSM* representation of the *BL-IRT* formulation is shown in Figure 2.12.

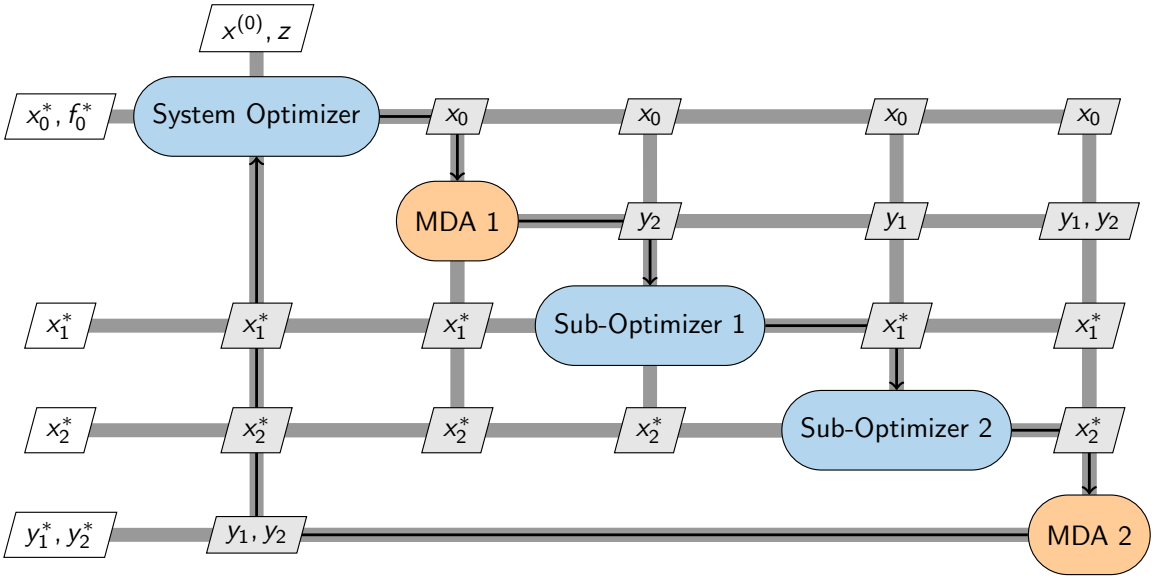


Figure 2.12: *XDSM* diagram of the *BL-IRT MDO* formulation

A gradient-free optimizer is to be used as the system-level optimizer, since the computation of coupled post-optimality gradients of the sub-scenarios is complex. Therefore, the set of top-level design variables must be limited to a sufficiently small set of variables.

David states in [30] that the covergence of this algorithm is strongly influenced by the chosen starting point of the optimization, since the coupling function is only updated before and after the subspace optimizations. Different coupling functions $\boldsymbol{\Psi}(\mathbf{x}_0, \mathbf{x}_{j\neq i})$ can therefore lead to a different subspace optimum $\mathbf{x}_i^*$. Furthermore, the top-level optimization

is subject to significant noise since it is not exposed to the changing local variables. However, he claims that these effects are less important for practical applications, since no top-level design point is visited twice, so the top-level function is only noisy, not random. Choosing an appropriate optimizer such as *COBYLA* can counteract the effects of noise, since its linear approximation is tolerant of noise.

2.5.3 *BL-BCD* Formulation

David [30] proposes the *BL-BCD* algorithm to improve the stability of the *BL-IRT* formulation. While two versions of this algorithm exist, the weakly coupled version of the algorithm is used here. In the *BL-BCD*, the lower-level optimization problem is solved using a *Block-Gauss-Seidel* method, alternating the optimization of the disciplines until a convergence criterion is met. An XDSM representation of this process is shown in Figure 2.13.

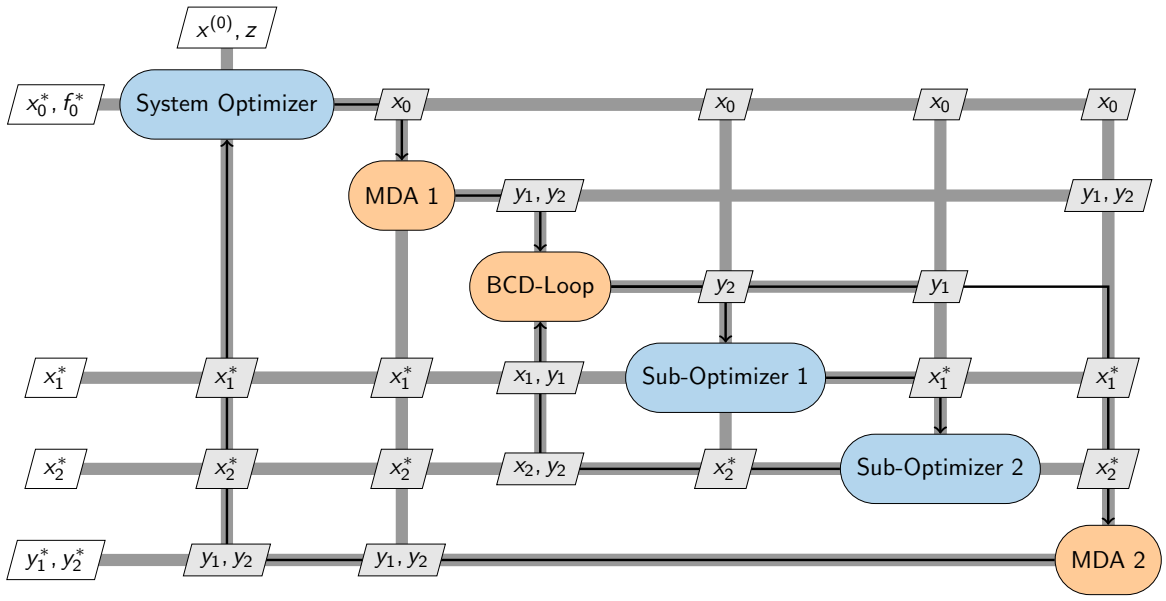


Figure 2.13: *XDSM* diagram of the *BL-BCD MDO* formulation

The convergence of the *BCD* algorithm relies on two assumptions, the *first-order separability* and the *unique block minizer*. A *first-order separable* problem is one in which, for each decomposed block, the effects of other blocks on the solution are negligible. The *unique block minimizer* property is satisfied if there is a unique minimizer and no local minima for each parameterization $\mathbf{x}_{\neq i} = [\mathbf{x}_{1:i-1}, ..., \mathbf{x}_{i+1:p}]$. In particular, a priori verification of the *first-order separability* is difficult for industrial problems [30].

For these cases, an alternative form of the *BL-BCD* formulation uses implicit target values set by the system-level optimizer and enforced on the disciplinary problems to ensure constant couplings [30]. The use of these implicit goal constraints instead of literal design variables is intended to separate tighter couplings.

Using the supersonic business jet (SSBJ) benchmark testcase, David shows a higher robustness of the *BL-BCD* algorithm compared to the *BL-IRT* formulation, but still not all starting points could be converged to the expected solution, as is the case for the *MDF* formulation.

2.5.4 *BL-EMB* Formulation

A modification of the general bilevel formulation is the formulation referred to here as the *bilevel-embedded* (*BL-EMB*) formulation. In the literature there exist a number of different names for this formulation, Abu-Zurayk et al. [14] called it the *two-optimizer* formulation, while Chittick et al. [32] refer to it as the *asymmetric suboptimization method*. The problem formulation of the *BL-EMB* formulation is similar to that of the *BL-IRT* and *BL-BCD* formulations, with the top-level optimization problem defined in eq. (2.95) and the disciplinary optimizations defined in eq. (2.96). The difference between the bilevel formulations of the *BL-EMB* and the *BL-IRT*/*BL-BCD* is the division of the set of design variables into the shared design variables $\mathbf{x}_0$ and the local design variables $\mathbf{x}$. For the *BL-EMB* formulation, the shared set of design variables $\mathbf{x}_0$ also contains all local disciplinary sets except for a single discipline. An example process with two sub-disciplines is shown in Figure 2.12.

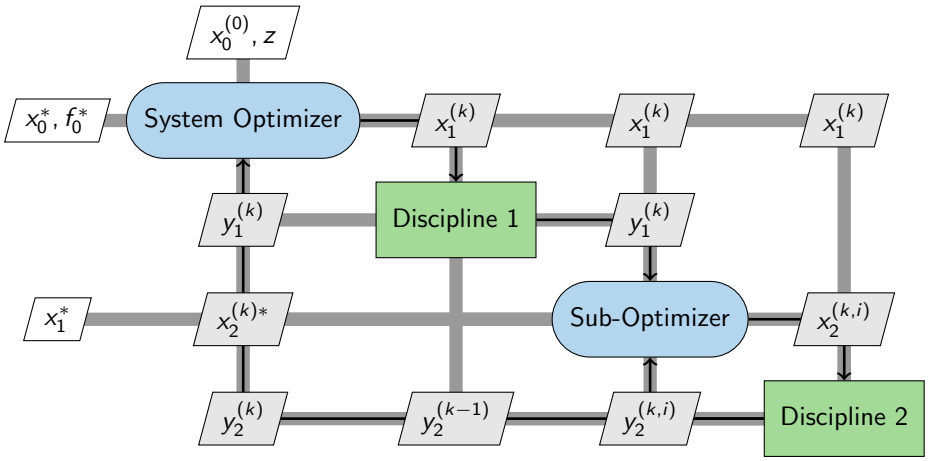


Figure 2.14: *XDSM* diagram of the *BL-EMB MDO* formulation

Changing the formulation to have only a single disciplinary sub-optimization allows the use of a gradient-based optimizer for system-level optimization, since only the post-optimality gradients of a single process need to be evaluated. A method for the compuation of the gradients for the embedded disicipline is proposed by Chittick et al. in [32]. They propose a coupled adjoint approach for the embedded structural optimization process.

Abu-Zurayk et al. used this process for an optimization of the *Airbus XRF-1* configuration, where they embed the structural optimization into the overall aircraft design and aeroelastic optimization process [14]. In their case, they used a gradient based optimizer for the top-level optimization, so post-optimality gradients of the embedded optimization have to be computed or noise is added to the optimization. These post-optimality gradients can be computed by applying *finite differences* over the optimization, but this can become computationally expensive for optimizations with a large number of design variables. Braun et al. [33] present a different method for obtaining the post-optimality gradients without having to recalculate the entire optimization by using the *Lagrangian* multipliers provided by a gradient-based optimization algorithm to save effort.

3 Methodology

This chapter deals with the application of the theory described in Chapter 2 to MDO processes. The first Section 3.1 presents the main tools that implement the numerical theories of Section 2.2. These tools serve as building blocks for the optimization processes explained in Section 3.2.

3.1 Tools

The MDO process used for the studies in this thesis is a combination of several open source and proprietary tools. All tools share a common interface with the *Python* programming language, which serves as the control layer of the integration. We use several open source libraries to extend the capabilities of *Python*, in particular the multidisciplinary integration framework *GEMSEO*, whose functionality and core concept we explain in Section 3.1.5. Another coupling framework used is *FlowSimulator* (Section 3.1.4), developed and used by DLR, ONERA and Airbus. It serves as an in-memory data exchange interface for aerodynamic and structural analysis tools. The main tools used for this optimization study are the CFD solver *TAU* (Section 3.1.1), the structural solver and optimizer *LAGRANGE* (Section 3.1.2) and the CFD post-processing library *Antares* (Section 3.1.6).

3.1.1 *TAU*

For aerodynamic performance prediction we use the *DLR* developed *TAU CFD* code [34]. *TAU* uses the *finite volume* discretization with unstructured meshes. To solve the *RANS* equations, a number of 1- and 2-equation turbulence models are available, of which the *SA* turbulence model described in Section 2.2.1 is used. For optimization purposes, *TAU* provides the ability to perform adjoint calculations.

3.1.2 *LAGRANGE*

For the structural analysis and optimization components of the multidisciplinary optimization process, the *LAGRANGE* [7] tool is used. *LAGRANGE* is a structural solver tailored to the needs of the aerospace industry. It provides analytical sensitivity calculation and integrates linear aerodynamic solvers for aerodynamic load analysis. It also includes several optimization algorithms, such as NLPQL [28], so it can also serve as a structural optimization framework. *LAGRANGE* is capable of structural sizing large problems with many constraints and design variables by adjusting various design parameters, including shape, thickness, and laminate structure. It can also use various stability constraints such as stringer buckling and manufacturing constraints to constrain the optimization problem. Optimization in *LAGRANGE* uses dynamic constraint callbacks so that only the gradients of the constraints present in the optimizer's *working set* need to be calculated. This can significantly speed up the computation time, especially for very large optimization tasks with millions of constraints.

3.1.3 CAD-ROM

Aircraft optimization processes rely on a parametric model to reflect changes in the set of design variables onto the geometry and a method to provide the gradients of these changes. While fully differentiated CAD kernels exist, widely used CAD tools such as *CATIA* do not provide this functionality. In the process used for this work, this task is instead performed by a CAD-ROM model developed by Merle et al. [35]. The theory behind this model is explained in Section 2.2.6.

Each CAD-ROM model is created from a fully parametric CAD model. Based on the given parameterization, the changes in the surface geometry must be extracted based on the given parameter combination. In the approach of Merle et al. this is done using an external meshing tool.

To generate the *ROM* from the CAD data, a DOE sequence covering the entire input parameter space is generated using the *SMARTy* toolbox [36]. The resulting input parameter combinations $\mathbf{p}^j$ are inserted into the parametric CAD model and the POD input matrix $\mathbf{X}$ is created by column-wise stacking of the resulting displacement fields. By applying the POD algorithm, the POD basis $\mathbf{\Phi}$ and the POD coefficients $\mathbf{a}$ are obtained.

The displacement field can be extracted from the model as

$$\mathbf{y}_{CAD}(\mathbf{p}^j) = \sum_{i=0}^{n} a_i(\mathbf{p}^j) \cdot \phi_i \tag{3.1}$$

This relationship is only valid for input parameter combinations that are part of the initial DOE set. An *RBF* interpolation is applied to the POD coefficients **a** to obtain a continuous function.

3.1.4 *FlowSimulator*

The *FlowSimulator* [37] framework coordinates the exchange of data between different solvers and methods relevant to aircraft design, such as *CFD* and structural solvers. Instead of relying on file inputs and outputs from the different disciplinary tools, *FlowSimulator* provides a framework to process these tool inputs and outputs in memory, allowing different tools such as *CFD* solvers, mesh interpolation, and deformation methods to be coupled. To enable parallelization on high performace computing (*HPC*) clusters, *FlowSimulator* uses *OpenMPI* parallelization to distribute data between different processes.

In our optimization toolchain, we use the coupling capabilities of *FlowSimulator* to exchange data between the structural and aerodynamic analysis tools. An essential method for this coupling is the mesh interpolation method, which interpolates parameters between different meshes. Information is interpolated between different meshes using the *MLS* method of Section 2.2.3 implemented in the *FlowSimulator* environment by Schuster et al. [21].

Another important feature for the transfer of displacements calculated by *LAGRANGE* to the CFD mesh is the deformation of the volume mesh. To deform the CFD mesh according to the geometric changes, the linear-elastic deformation method explained in Section 2.2.4 is used, implemented and coupled to the *FlowSimulator* environment by Rempke [38].

3.1.5 *GEMSEO*

GEMSEO [15] is an open source *Python* library developed by IRT Saint Exupéry for the automation of MDO processes. The main goal of the software is to reduce setup time for automated optimization and analysis processes and to enable code exchange between partners via the common interface. It is used as the top-level integration framework of the aerostructure optimization study. This section explains the main concepts of *GEMSEO*

that are relevant for performing multidisciplinary optimization studies [1].

To facilitate reconfigurability of analysis and optimization tasks, each disciplinary analysis is wrapped in a class that inherits from `MDODiscipline`. Each class derived from `MDODiscipline` defines inputs and outputs, a `_run()` method for computing the outputs from the inputs, and a `_compute_jacobian()` method for computing the gradients of all output variables. Discplines can be arranged in a `Scenario` class.

Besides the *BL-IRT* formulation, the *MDF* and *IDF* formulations are available, as well as disciplinary optimizations. Other formulations can be created by the user by combining various objects inherited from the `MDODiscipline` class, such as the `MDOChain` or `MDF` objects. A separate proprietary plugin provides a more flexible implementation of the *BL-IRT* formulation and the *BL-BCD*, but these are not available in the open source software package.

When a `Scenario` is created, *GEMSEO* controls the order of execution of the disciplines and distributes the data. Before execution, this order can be checked using the `Scenario.xdsmize()` method, which automatically generates the *XDSM* diagram of the MDO process using the `pyXDSM` library. Two types of algorithms are provided for execution, optimization and design of experiments (*DOE*) algorithms from various open source packages.

After a `Scenario` has been run, several post-processing methods are provided to generate plots of the evolution of, for example, objective and design variables, constraint violations, or gradient sensitivities. Information about the execution of each problem can be stored in `Caches`, allowing post-processing after execution is complete, or restarting the optimization from a given starting point. The software package also provides several methods for parallelizing computational tasks, based on multiprocessing and multithreading. A *Slurm* job wrapper is also available for disciplines. However, at the time of writing, *OpenMPI* parallelization is not yet supported.

3.1.6 *Antares*

Antares is a *Python* library developed by Cerfacs for processing *CFD* data [39]. It can open different types of *CFD* data formats, such as *Tecplot* (`.plt`/`.dat`) or *Paraview* (`.vtu`/`.vtk`) files and load them into memory. *Antares* decomposes the various CFD data formats into their object-oriented structure, whose top-level object is the `Base` object. Each

[1] This description is based on *GEMSEO* v. 5.3.2. The functionality described may differ from other versions

base consists of different `Zone` objects representing different subspaces of the geometry, such as the different boundaries or the far field. The `Zone` objects themselves consist of one or more `Instance` objects. A `Instance` does not necessarily have to be a temporal instance, the `Instance` can also be used to store different iterations of an optimization process. These `Instance` objects then contain the CFD data, represented by *Numpy* arrays. The *Antares* data structures can be manipulated directly by the user by implementing *Python* methods, or by using the methods provided within *Antares*, called `Treatments`. Examples of `Treatments` are cutting planes, computing cell normal vectors, or merging `Zone` objects. In this thesis, *Antares* is used to plot the spanwise lift distribution, sectional views, and pressure distributions of wings.

3.2 Process Integration

After introducing the various tools used in the optimization study, this section describes the optimization toolchains created from these tools. It also shows how the tools are integrated into the `MDODiscipline` class of *GEMSEO*. First, in Section 3.2.1, we explain the *LAGRANGE*-based structural sizing process. In Section 3.2.2, the process setup for aeroelastic analysis with trim capability is defined. These two processes are the main building blocks used to generate the various *MDO* formulations, the evaluation of which is the main objective of this thesis.

3.2.1 Structural Sizing Process

The structural sizing process is adapted from the work of Volle et al. [8]. It controls the structural sizing design variables to satisfy various structural integrity constraints, such as strength, stability, and manufacturing constraints, while minimizing the finite element mass. Unlike the aerodynamic performance prediction process, it is itself an optimization process using the *LAGRANGE* internal *NLPQL* optimizer. The method is based on the low-fidelity *Athena Vortex Lattice* (*AVL*) load evaluation tool. Currently, the low-fidelity load model is unable to update changes in the wing planform. Similar to the aerodynamic performance prediction process, the process is trimmed for specific load cases using Equation (3.2). After trimming, the constraints are evaluated. For optimization, the gradients for the structural design variables are computed internally in *LAGRANGE*, dynamically switching between direct and adjoint automatic derivation. Shape design variable gradients are computed by adding the interpolated CAD-ROM gradients to the

structural coordinates and evaluating the load model. Similar to the power gradient calculation, the structural gradients are corrected for trimming. In addition, the ability to compute the post-optimality gradients with respect to the shape input parameters is added for gradient-based optimization in the embedded formulation.

3.2.2 Aerodynamic Performance Prediction

The aerodynamic performance prediction method is adapted from the work of Volle [8]. This method provides methods for gradient computation of performance parameters with respect to all shape and structural design variables using the adjoint approach. A flowchart of the process is shown in Figure 3.1. The inputs to the shape design variables are

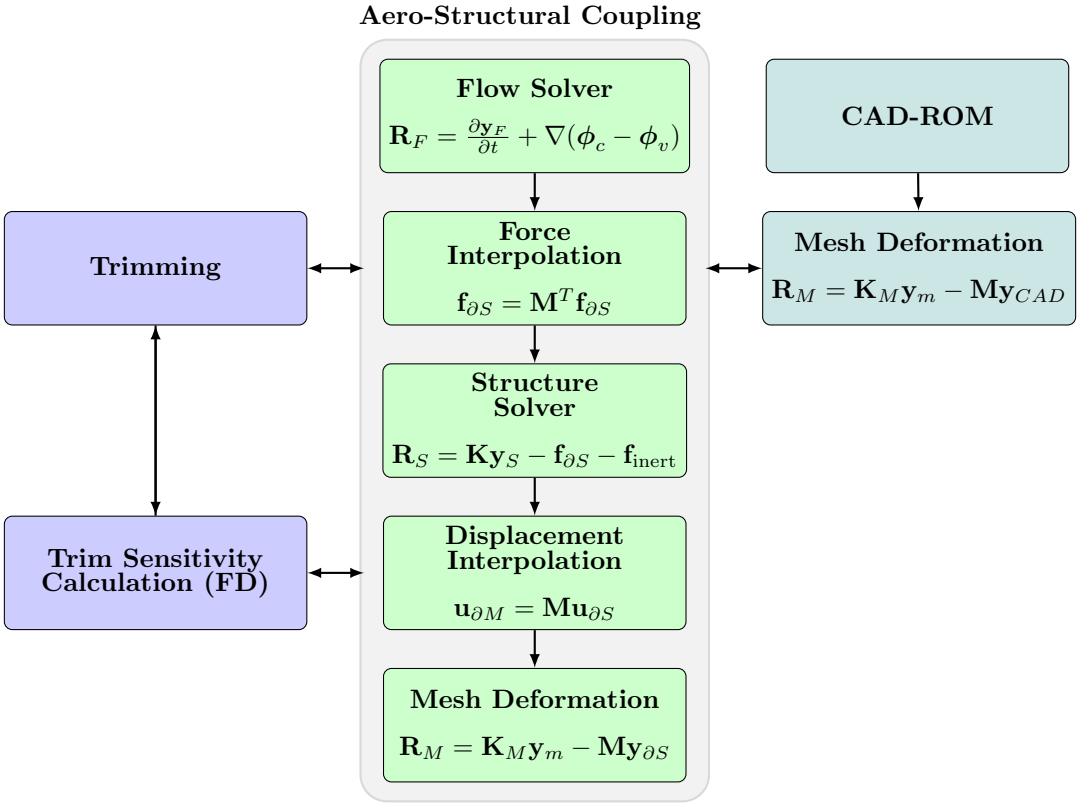


Figure 3.1: Flowchart of the aerodynamic performance prediction process using CAD-ROM geometry modeling

converted to surface mesh points by a fully differentiated CAD-ROM. The surface mesh is deformed using the Section 3.1.4 mesh deformation method and passed to the trimming loop. The trimming loop itself consists of two loops, the inner aero-structure loop resolves the aero-structure coupling, while the trimming loop itself adjusts the trim variables to achieve the desired flight state. *TAU* calculates the forces resulting from the flow on the CFD mesh, which are interpolated to the CSM mesh using the interpolation method of

Section 3.1.4. From the forces acting on the CSM mesh, *LAGRANGE* determines the structural displacements on the mesh, which are interpolated back to the CFD mesh. The CFD volume mesh is again deformed according to the displacements. Mathematically, this is described by the following fixed-point iteration scheme for the iteration step i

$$\begin{aligned}
\mathbf{f}_{\partial F}^{(i+1)} &= CFD(\mathbf{y}_M^{(i)}) \\
\mathbf{f}_{\partial S}^{(i+1)} &= \mathbf{M}^T \cdot \mathbf{f}_{\partial F}^{(i+1)}(\mathbf{y}_M^{(i)}) \\
\mathbf{y}_S^{(i+1)} &= \mathbf{K}^{-1} \cdot (\mathbf{f}_{inert} + \mathbf{f}_{\partial S}^{(i+1)}) \\
\mathbf{y}_M^{(i+1)} &= \mathbf{K}_M^{-1} \cdot \mathbf{M} \cdot \mathbf{y}_S^{(i+1)}
\end{aligned}$$

This inner aeroelastic loop is iterated until the $\mathcal{L}_2$-norm of the structural state variable changes is sufficiently small. Convergence is improved using the *Aitken* method [40]. For the outer trimming loop, a gradient-based approach is used to compute the next iteration of the trim variables δ by applying *finite differences* by perturbing the trim design variables. The resulting trim *Jacobian* is multiplied by the residual force vector $\mathbf{F}_R$.

$$\delta^{i+1} = \delta^i - \left(\frac{d\mathbf{F}}{d\delta}\right)^{-1} \mathbf{F}_R. \tag{3.2}$$

3.2.3 Aerodynamic Performance Sensitivity Calculation

The gradients of the coupled aeroelastic performance prediction process are computed using the coupled adjoint approach of Section 2.3.3 in a two-step process as shown in Figure 3.2. The first step of the process is to solve the coupled adjoint equations from Equation (2.73). Then, from the mesh and structural adjoints and the derivatives of the respective residuals, the power sensitivities are computed and trimmed. To solve the adjoint equations of Equation (2.74), all residual derivatives must be solved by the disciplinary flow, structural, and mesh deformation solvers. The mesh residual derivatives $\partial\mathbf{R}_M/\partial\mathbf{y}_M$ and $\partial\mathbf{R}_M/\partial\mathbf{y}_S$ are obtained from the mesh deformations according to Merle et al. [41] as the mesh deformation stiffness matrix $\mathbf{K}_M$ and the interpolation matrix $\mathbf{M}$:

$$\frac{\partial\mathbf{R}_M}{\partial\mathbf{y}_M} = \mathbf{K}_M \tag{3.3}$$

$$\frac{\partial\mathbf{R}_M}{\partial\mathbf{y}_S} = \mathbf{M}. \tag{3.4}$$

The partial derivatives of the structural residuum are calculated by

$$\frac{\partial \mathbf{R}_S}{\partial \mathbf{F}} = \frac{\partial \mathbf{R}_S}{\partial \mathbf{F}_{\partial S}} \frac{\partial \mathbf{f}_{\partial S}}{\partial \mathbf{f}_{\partial M}} \frac{\partial \mathbf{f}_{\partial M}}{\partial \mathbf{F}} = -\mathbf{M}^T \frac{\partial \mathbf{f}_{\partial M}}{\partial \mathbf{F}} \tag{3.5}$$

$$\frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_M} = \frac{\partial \mathbf{R}_S}{\partial \mathbf{f}_{\partial S}} \frac{\partial \mathbf{f}_{\partial S}}{\partial \mathbf{f}_{\partial M}} \frac{\partial \mathbf{f}_{\partial M}}{\partial \mathbf{y}_M} = -\mathbf{M}^T \frac{\partial \mathbf{f}_{\partial M}}{\partial \mathbf{y}_M} \tag{3.6}$$

$$\frac{\partial \mathbf{R}_S}{\partial \mathbf{y}_S} = \mathbf{K}. \tag{3.7}$$

A fixed-point iteration scheme according to Gastaldi et al. [42] is used to solve the system of adjoint equations for the fluid adjoint state $\bar{\mathbf{y}}_F$, the mesh adjoint state $\bar{\mathbf{y}}_M$, and the structural adjoint state $\bar{\mathbf{y}}_S$.

$$\begin{aligned} \frac{\partial \mathbf{R}_F^T}{\partial \mathbf{y}_F} \bar{\mathbf{y}}_F^{(i+1)} &= -\frac{\partial C_x}{\partial \mathbf{y}_F} + (\mathbf{M}^T \frac{\partial \mathbf{f}_{\partial M}}{\partial \mathbf{y}_F})^T \bar{\mathbf{y}}_S^{(i)} \\ \mathbf{K}_M^T \bar{\mathbf{y}}_M^{(i+1)} &= \frac{\partial C_x}{\partial \mathbf{y}_M} - \frac{\partial \mathbf{R}_F^T}{\partial \mathbf{y}_M} \bar{\mathbf{y}}_F^{(i+1)} + (\mathbf{M}^T \frac{\partial \mathbf{f}_{\partial \mathbf{M}}}{\partial \mathbf{y}_M})^T \bar{\mathbf{y}}_S^{(i)} \\ \mathbf{K}^T \bar{\mathbf{y}}_S^{(i+1)} &= -\mathbf{M}^T \bar{\mathbf{y}}_{\mathbf{M}}^{(i+1)}. \end{aligned} \tag{3.8}$$

Following the scheme of Equation (3.8), *TAU* is first run in adjoint mode to obtain the flow adjoint. Then the mesh deformation adjoint process is performed with the flow adjoint as input. The mesh adjoint is interpolated to the structural mesh where *LAGRANGE* calculates the adjoint displacements using the adjoint forces as boundary forces. The structural displacements are again interpolated to the CFD mesh. This loop is iterated until the $\mathcal{L}_2$ error convergence criterion for the structural adjoint change is reached. As in the original mode, the *Aitken* method is used to accelerate the convergence.
The performance sensitivities are determined from the adjoint mesh and structure states $\bar{\mathbf{y}}_M$ and $\bar{\mathbf{y}}_S$

$$\frac{dC_x}{d\mathbf{x}} = \bar{\mathbf{y}}_{\mathbf{S}} \frac{\partial \mathbf{R}_S}{\partial \mathbf{x}} + \bar{\mathbf{y}}_{\mathbf{M}} \frac{\partial \mathbf{R}_M}{\partial \mathbf{x}}. \tag{3.9}$$

While Volle computes the performance sensitivities for all design variables, including the structural sizing parameters, we only need the sensitivities with respect to the shape parameters because the structural parameters are held constant for each shape optimization step in the *bilevel* formulations. The derivative of the structure residual by the structural design variables consists of the change in the stiffness matrix $\mathbf{K}$ and the inertial forces $\mathbf{f}_{inert}$.

$$\frac{\partial \mathbf{R}_S}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{K}}{\partial \mathbf{x}} - \frac{\partial \mathbf{f}_{inert}}{\partial \mathbf{x}} \right) \tag{3.10}$$

Both terms do not depend directly on the shape design variables, but on the structural coordinates $\mathbf{X}_S$, which are interpolated from the CAD-ROM output coordinates $\mathbf{X}_R$ with

the interpolation matrix $\mathbf{M}_{SR}$, thus

$$\frac{\partial \mathbf{R}_S}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{K}}{\partial \mathbf{X}_S} - \frac{\partial \mathbf{f}_{inert}}{\partial \mathbf{X}_S} \right) \mathbf{M}_{SR} \frac{\partial \mathbf{X}_R}{\partial \mathbf{x}}. \tag{3.11}$$

The derivative of the coordinates $\frac{\partial \mathbf{X}_R}{\partial \mathbf{x}}$. The partial derivative term $\frac{\partial \mathbf{R}_M}{\partial \mathbf{x}}$ describes the change in the mesh residual due to perturbations in the shape design variables. This can be written as

$$\bar{\mathbf{y}}_M \frac{\partial \mathbf{R}_M}{\partial \mathbf{x}} = \bar{\mathbf{y}}_M \mathbf{M}_{MR} \frac{\mathbf{X}_M}{\partial \mathbf{x}}, \tag{3.12}$$

However, for performance reasons, the mesh adjoints are interpolated to the CAD-ROM coordinates

$$\bar{\mathbf{y}}_M \frac{\partial \mathbf{R}_M}{\partial \mathbf{x}} = (\mathbf{M}_{MR}^T \bar{\mathbf{y}}_M) \frac{\partial \mathbf{X}_R}{\partial \mathbf{x}} \tag{3.13}$$

Since the trimming algorithm changes the trim variables, which are not exposed to any optimizer, to keep the aircraft in a defined trim state, the gradients are corrected against the trim variables as described by Merle et al. [41]. The trim-corrected gradient for the flight performance output variable $\mathbf{y}_{aero}$ is the total derivative of the flight performance metric, which is a function of the design variables $\mathbf{x}$ and the trim variables δ. As a secondary condition, the derivatives of the trim constraints $\mathbf{C}_x$ with respect to the design variables $\mathbf{x}$ are equal to 0, since the trim keeps these values constant. Both equations for a system of two vector equations, which can be solved in

$$\frac{d\mathbf{C}_x}{d\mathbf{x}}|_{\mathbf{C}_{L,My}=const.} = \frac{\partial \mathbf{y}_{aero}}{\partial \mathbf{x}} - \frac{\partial \mathbf{y}_{aero}}{\partial \delta} \left(\frac{d\mathbf{C}_x}{d\delta} \right)^{-1} \frac{d\mathbf{C}_x}{d\mathbf{x}} \tag{3.14}$$

The derivative of the trimming constraints by the trimming design variables is computed with the mesh adjoint multiplied by the derivative of the surface points with respect to the trimming design variables.

$$\left(\frac{d\mathbf{C}_x}{d\delta} \right) = \mathbf{y}_M \frac{d\mathbf{X}_M}{d\delta}. \tag{3.15}$$

For control surface deflections and α as δ, the surface coordinate term $\mathbf{X}_M$ is computed by rigid body motion.

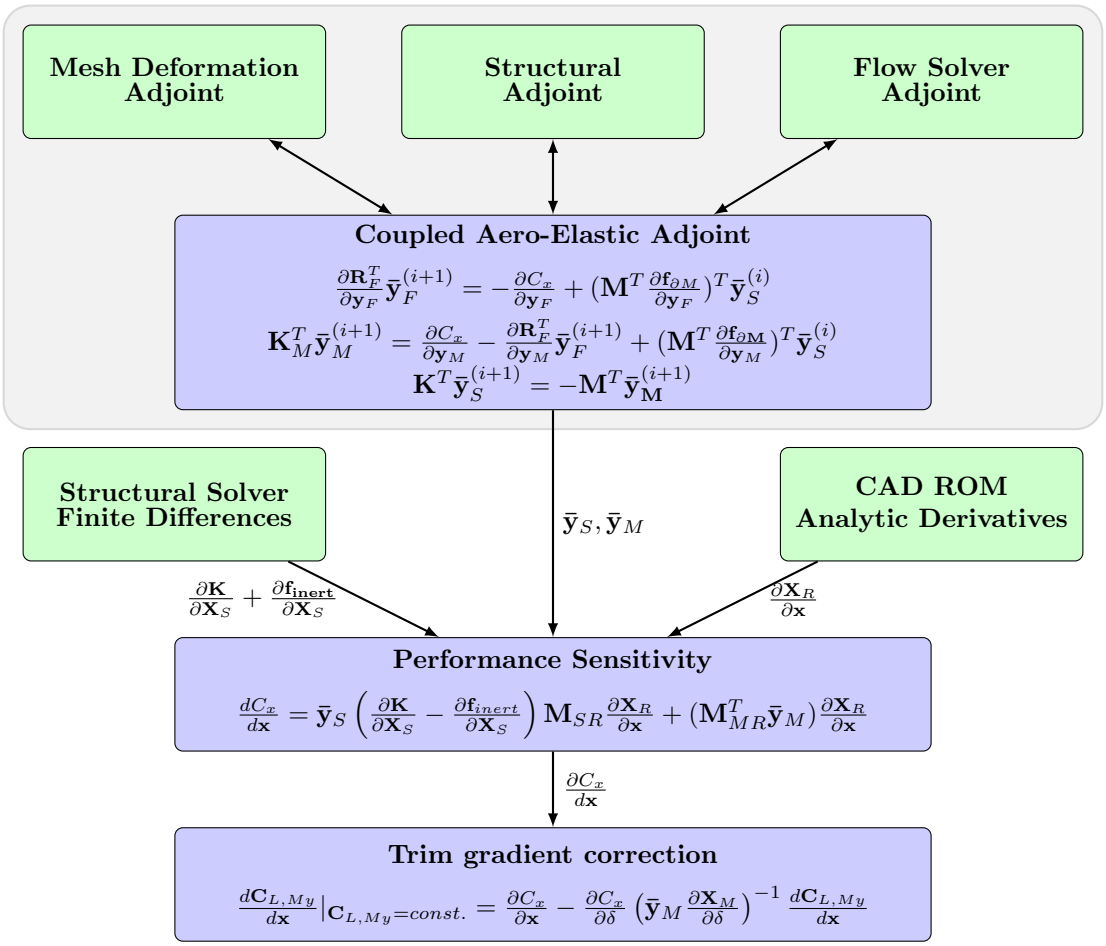


Figure 3.2: Flowchart of the aerodynamic performance prediction process gradient computation

3.2.4 Software Architecture

A significant part of the work to convert the pure *FlowSimulator* coupled process of Volle et al. into the *GEMSEO* integration environment was dedicated to parallelizing the process and implementing the required data transfer methods. This is necessary because *FlowSimulator* allows parallelization using *Message Parsing Interface* (*MPI*), which is not yet available in *GEMSEO*. Thus, a split process was developed to gain the formulation flexibility of *GEMSEO* while preserving the efficient coupling and data exchange methods of the processes of Volle et al. An overview of the architecture is given in Figure 3.3.

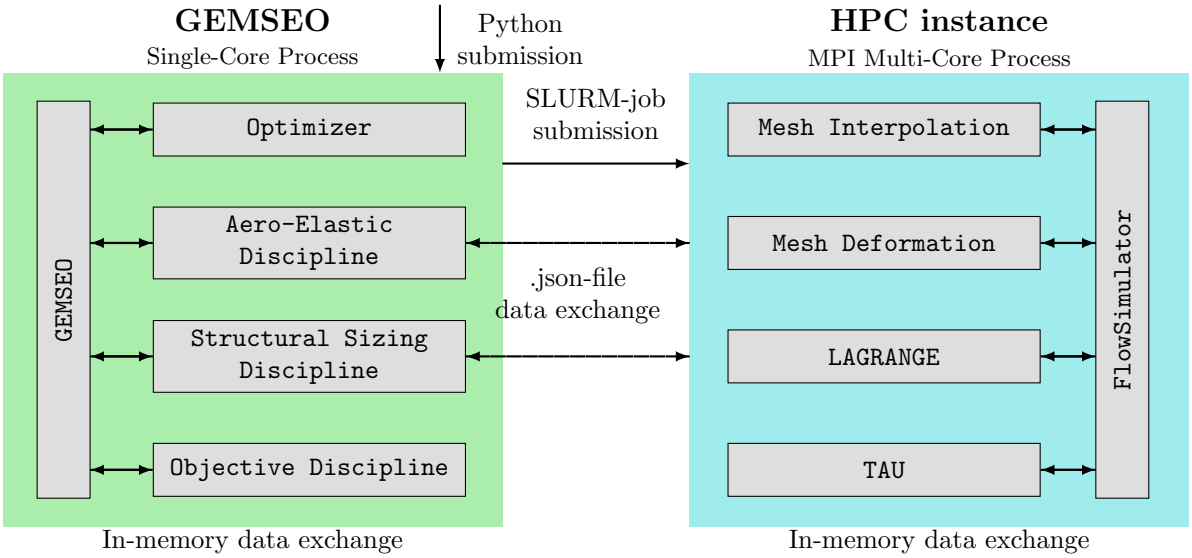


Figure 3.3: Software Architecture of the Integrated MDO Process

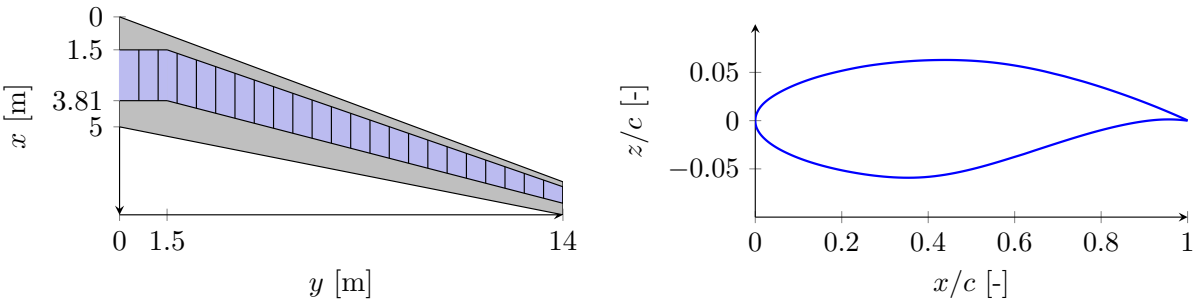
The implementation is split between a *GEMSEO* process running on a single process and a process running on an HPC node. Within the *GEMSEO* process, the optimizer passes design variables to the disciplines and reads their outputs to iterate to the next step. The disciplines themselves, with the exception of the `Objective Discipline`, do not perform direct computations, but submit jobs to the HPC process by passing `.json` files. These files are processed on the HPC cluster and either a cycle of the aerodynamic performance prediction process or the structural sizing process is run. The results are written to `.json` files and returned to the *GEMSEO* instance. Each HPC instance can only process a single job submission at a time, so the aerodynamic shape optimization and structural sizing must be run sequentially in the *bilevel* processes. However, it is possible to create multiple HPC instances to solve multiple flight conditions in a multi-point scenario.

4 Testcases

To test the different problem formulations defined in the previous Section 3.2, we need to provide appropriate test cases. The testcase defined within the *UpWing* and *NextAir* projects is the *DLR-F25* high aspect ratio, narrow-body jet transport aircraft. This is the main test case on which the tests will be performed. However, since this is a large testcase that requires long periods for aerodynamic and structural evaluation iterations, the much smaller benchmark process proposed by the University of Michigan [43] is used for the development of the aerostructure optimization process. Thus, we perform all studies on both test cases to also investigate the impact of problem scale on the performance of the embedded and *bilevel* formulations.

4.1 Simple Transonic Wing (University of Michigan)

Gray et al. [43] have published an open-source benchmark model for comparing high-fidelity RANS-CFD MDO frameworks. This model is a single wing whose planform is derived from the *Boeing 717* aircraft, with the wing profile described by the *RAE2822* airfoil. They provide meshes with different levels of resolution, ranging from 100,000 to 7,000,000 cells. The structural model of the wing is a wing box consisting of skin panels and a total of 23 ribs, the material of the wing is carbon-fiber reinforced polymer (*CFRP*) with a fixed ply-share. Figure 4.1a shows a top-down view of the wing planform and the internal structure of the wing box. The supercritical *RAE2822* airfoil is shown in Figure 4.1b. In their paper, Gray et al. propose three different optimization scenarios, a structural optimization with a fixed wing shape, a fuel burn optimization with a fixed planform, and a fuel burn optimization of a wing with a variable planform. In this study, only the fixed planform fuel burn optimization is performed because the load evaluation model in *LAGRANGE* does not update the planform during runtime. In this case, the set of planform design variables consists of the wing twist at 5 spanwise locations and a



(a) Wing planform projection with internal wing box structure

(b) *RAE2822*profile

Figure 4.1: Geometry of the simple transonic wing

total of 108 airfoil parameters. The airfoil parameterization used differs from the *free-form deformation* approach described in the paper. Instead of the described free-form deformation approach, a *CST* curve parameterization [44] controls the outer contour of the profile at 6 spanwise locations with 18 design parameters. For the structural optimization a number of design variables are given for the wing box, listed in Table 4.1. The structural

Table 4.1: Structural design variables of the simple transonic wing

Description	Count	Bounds
Panel length	111	$[0, \infty)$ mm
Panel thickness	111	$[2, 100]$ mm
Stiffener thickness	111	$[2, 100]$ mm
Stiffener height	111	$[2, 100]$ mm
Local stiffener pitch	23	$[50, 500]$ mm
Global stiffener spacing	4	$[50, 500]$ mm

optimization is subject to a total of $82,767$ constraints, including stability, strength, and manufacturing constraints. Additionally, the trim constraint of a C_L of 0.52 is satisfied by adjusting the angle of attack. Table 4.2 shows the distribution between the different groups of constraints. Two load cases are to be used for structural sizing, a push-down

Table 4.2: Structural constraints of the simple transonic wing

Constraint	Count per load case	Count total
Panel buckling	548	1,096
Stringer buckling, crippling	472	944
Strength	20.166	40.332
Manufacturing		63
Total		82,767

and a pull-up maneuver, both at sea level with a reduced airspeed of $Ma = 0.458$ and load

factors of -1 and 2.5 respectively. Other constraints proposed by the author relate to the geometry of the wing, but are not considered in this study. A number of estimates are made to obtain the total mass of the aircraft. The total mass of a single wing is estimated from the finite element mass m_{FE} of the wing box using a regression method by Mariens et al. [45]

$$m_{Wing} = 10.147 m_{FE}^{0.8162}. \tag{4.1}$$

To obtain the gross landing mass (m_{LGM}), add the mass of the fuselage m_{Frame}, the reserve fuel mass m_{RF}, and the payload mass m_{PL} to the wing masses.

$$m_{LGM} = m_{Frame} + m_{RF} + m_{PL} + 2 \cdot m_{Wing}. \tag{4.2}$$

The values of m_{Frame}, m_{RF}, and m_{PL} are listed in Table 4.3a. The mission fuel burn m_F is used as the optimization objective. It is calculated using the *Breguet* domain equation

$$m_F = m_{LGM} \cdot \left[\exp\left(\frac{D}{L} \cdot \frac{9.81 \cdot TSFC \cdot R}{V} \right) - 1 \right], \tag{4.3}$$

using the mission parameters from Table 4.3b. The total lift of the aircraft L is calculated by multiplying the lift produced by each wing by 2, the total drag is the sum of the drag of the wings and the drag of the airframe from Table 4.3b.

$$L = 2 \cdot L_{Wing} \tag{4.4}$$

$$D = 2 \cdot D_{Wing} + q_{Cruise} \cdot S_{ref} \cdot C_{D,Frame} \tag{4.5}$$

For simplicity, the climb segment is not included in the optimization as described in the publication.

The aerodynamic analysis of the baseline configuration, performed on the coarsest mesh available, shows that the lift distribution deviates from the aerodynamic optimum. As shown in Figure 4.2a, the distribution shows a slight outboard shift compared to the ideal elliptical distribution.

Surface pressure analysis indicates the absence of shock waves across the wing, as shown by both the pressure projection plot (Figure 4.3) and the sectional pressure distributions (Figure 4.4). However, this smooth pressure field can be attributed to the limitations of the mesh resolution, as analyses on finer meshes captured a distinct shock formation on the upper surface of the wing. Also, due to the coarseness of the mesh, the extracted lift distribution shows discontinuities at the wing root. Therefore, this part of the wing is not shown in the plots of the lift distribution. The profile geometry is a *RAE2822* airfoil at all spanwise locations as shown in Figure 4.4. Consequently, the thickness-to-chord ratio (t/c) ratio is constant along the span. In flight, the wing is slightly twisted downwards,

Table 4.3: Objective function parameters of the simple transonic wing

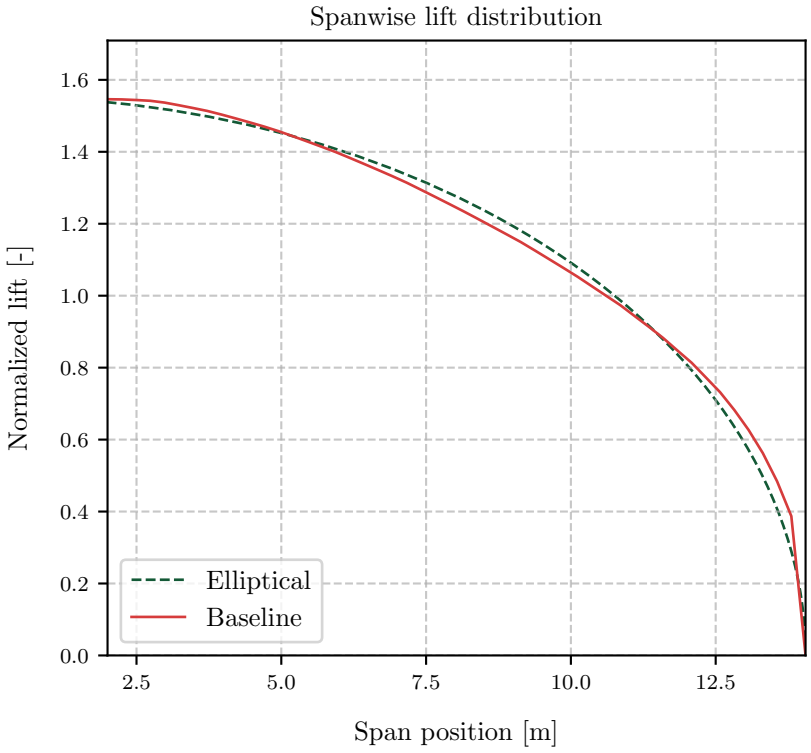
(a) Mass parameters of the simple transonic wing

Parameter	Description	Value
m_{Frame}	Operating empty mass (without wing)	$25,000\,\mathrm{kg}$
m_{PL}	Payload mass	$14,500\,\mathrm{kg}$
m_{RF}	Reserve fuel mass	$2,000\,\mathrm{kg}$

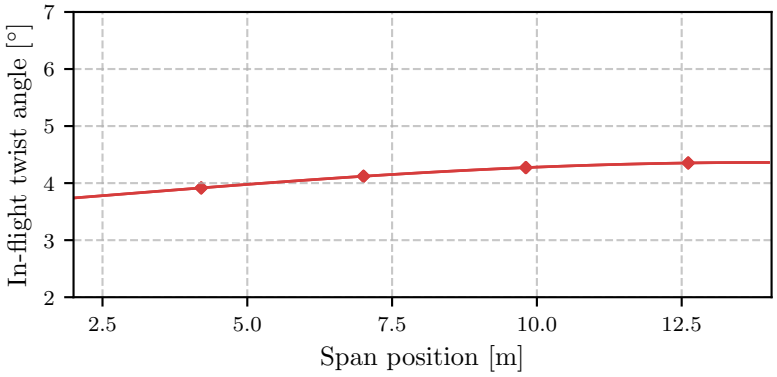
(b) Mission parameters of the simple transonic wing

Parameter	Description	Value
$C_{D,Frame}$	Coefficient of drag (C_D) of airframe	0.0152
H_{Cruise}	Cruise altitude	$10.400\,\mathrm{m}$
Ma_{Cruise}	Cruise Mach (Ma) number	0.78
R	Nominal range	$3815\,\mathrm{km}$
S_{ref}	Wing area	$45.5\,\mathrm{m}^2$
$TSFC$	Thrust-specific fuel consumption	$1.8{\cdot}10^{-5}\,\mathrm{kg\,N\,s}^{-1}$

starting with a pitch of about 3° at the wing root and increasing to a value of about 4.3° at the wing tip, as shown in Figure 4.2b. In its initial configuration, the coefficient of drag (C_D) of the wing and the added constant fuselage drag is 0.0332 with a dimensioned finite element mass m_{FE} of 815 kg. This results in an estimated fuel burn for the reference mission m_F of 9500 kg.



(a) Lift distribution of the baseline simple transonic wing



(b) Spanwise twist of the of the baseline simple transonic wing

Figure 4.2: Spanwise characteristics of the baseline simple transonic wing

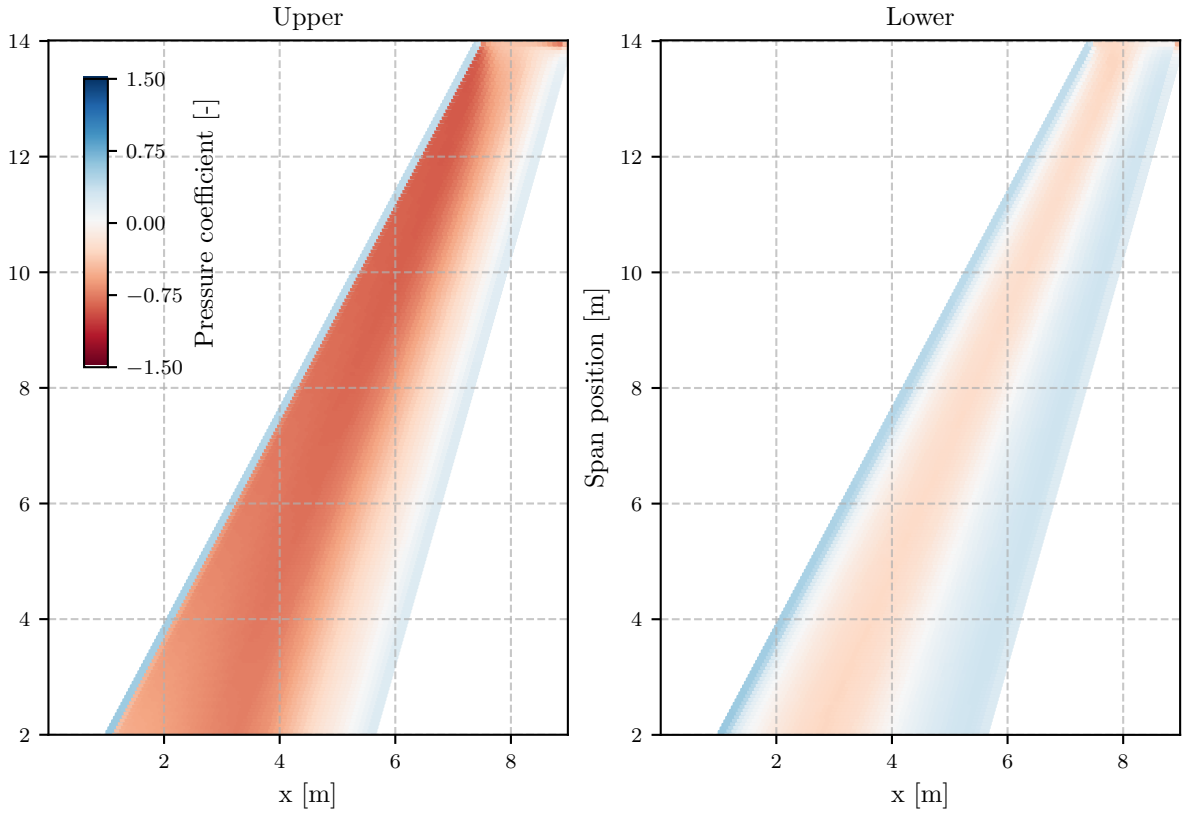


Figure 4.3: Wing surface pressure distribution of the baseline simple transonic wing

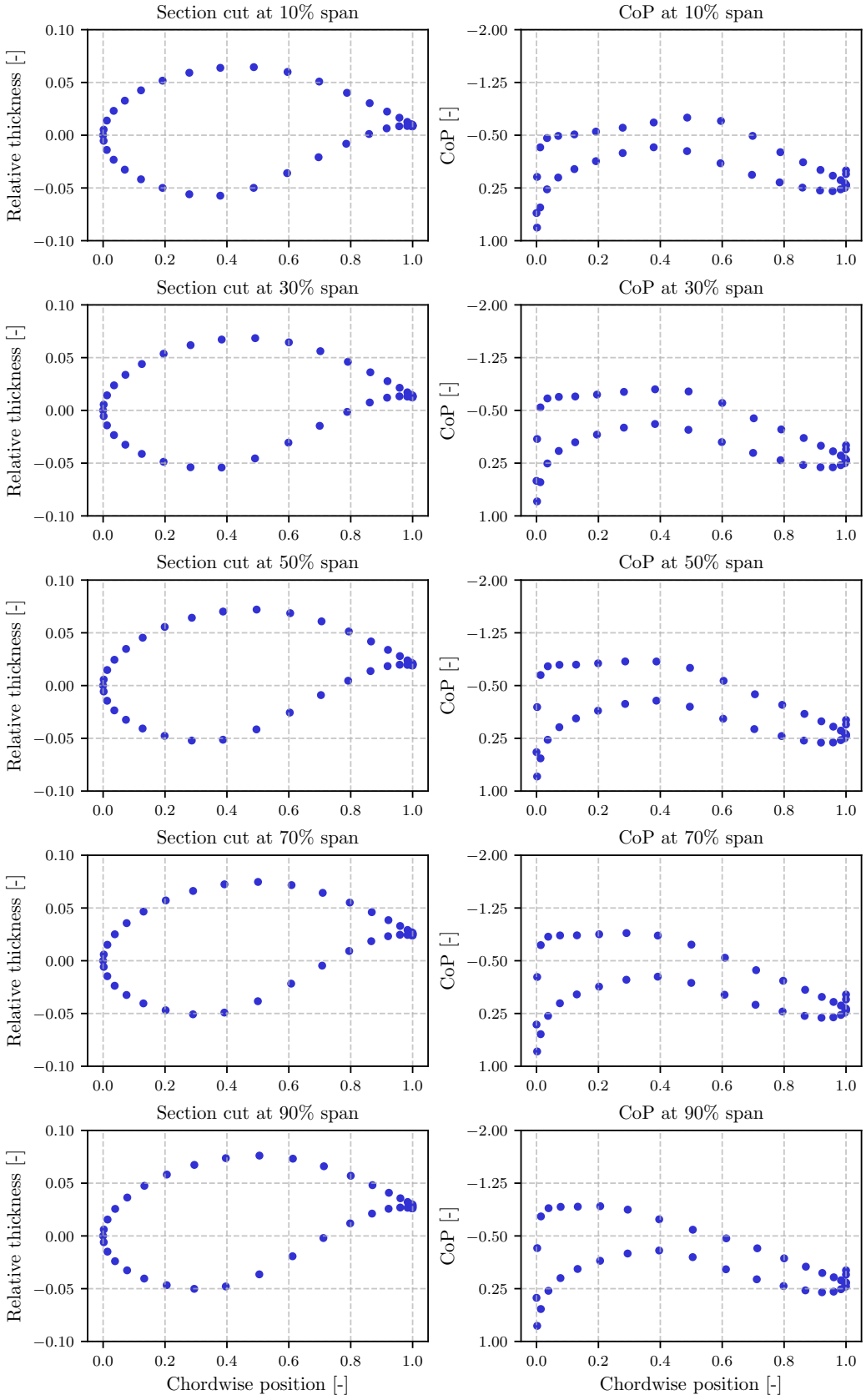


Figure 4.4: Section cuts of the baseline simple transonic wing

4.2 *DLR-F25*

The *DLR-F25* aircraft model is a model of a new generation short-to-medium range *HARW* aircraft developed by DLR as part of the *Virenfrei* project. It is used as a sample aircraft configuration for the *UpWing* and *NextAir* projects, which study *HARW*. The *AR* of the wing is 15.6, which is significantly increased compared to similar state-of-the-art aircraft.

The configuration is considered relevant for the development of novel MDO processes, as the next generation of aircraft is expected to be a short to medium range aircraft [46]. The *HARW* also allows a high optimization potential for coupled aero-structural effects. In this study, a wing-body configuration of the aircraft without a horizontal stabilizer is used. The aerodynamic surface mesh of this configuration is depicted in Figure 4.5. For

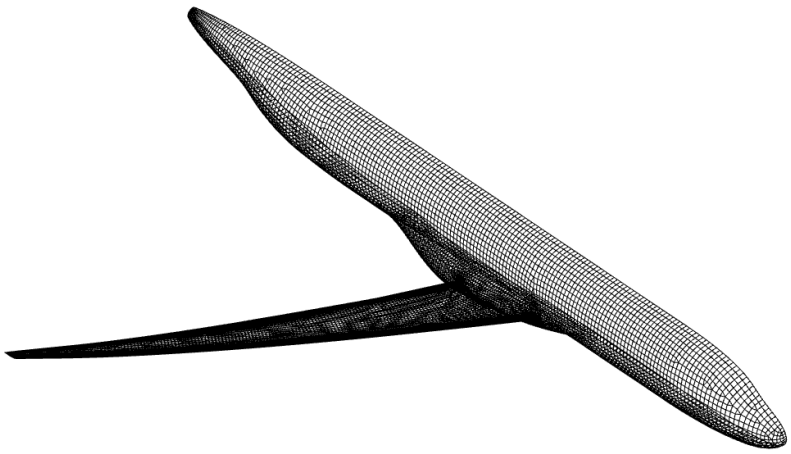
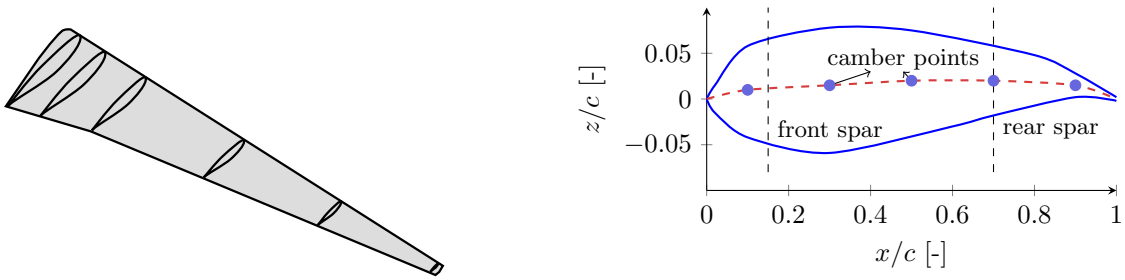


Figure 4.5: Aerodynamic surface mesh of the *DLR-F25*

this case, the design variables are the wing twist, the profile parameters and the structural sizing parameters. The profile parameters define 8 points on the camber line at 6 spanwise positions. This parameterization is shown in Figure 4.6. The mission fuel burn calculated with the *Breguet* range equation is used as the objective function of the optimization. Unlike the simple transonic wing model, the structural model contains the full mass model of the aircraft, so only the reserve fuel mass m_{RF} needs to be added to the finite element mass m_{FE}. The required reserve fuel mass m_{RF} consists of a fuel reserve sufficient for a holding time of 30 min and 3% contingency fuel. In total, the reserve fuel mass is estimated as $m_{RF} = 1,500\,\text{kg}$. The objective function of this test case is the mission fuel burn m_F,



(a) Illustration of the *DLR-F25* profile (b) Illustration of the *DLR-F25* profile parameterization paramters' spanwise distribution

Figure 4.6: *DLR-F25* profile parameters

calculated with the reformulated *Breguet* range equation

$$m_F = m_{FE} \cdot \left[\exp\left(\frac{D}{L} \cdot \frac{9.81 \cdot TSFC \cdot R}{V} \right) - 1 \right]. \tag{4.6}$$

All parameters included in the *Breguet* range equation are listed in Table 4.4b.

Table 4.4: Objective function parameters of the *DLR-F25*

(a) Mass parameters of the *DLR-F25*

Parameter	Description	Value
m_{PL}	Payload mass	$25,000\,\mathrm{kg}$
m_{RF}	Reserve fuel mass	$1,500\,\mathrm{kg}$

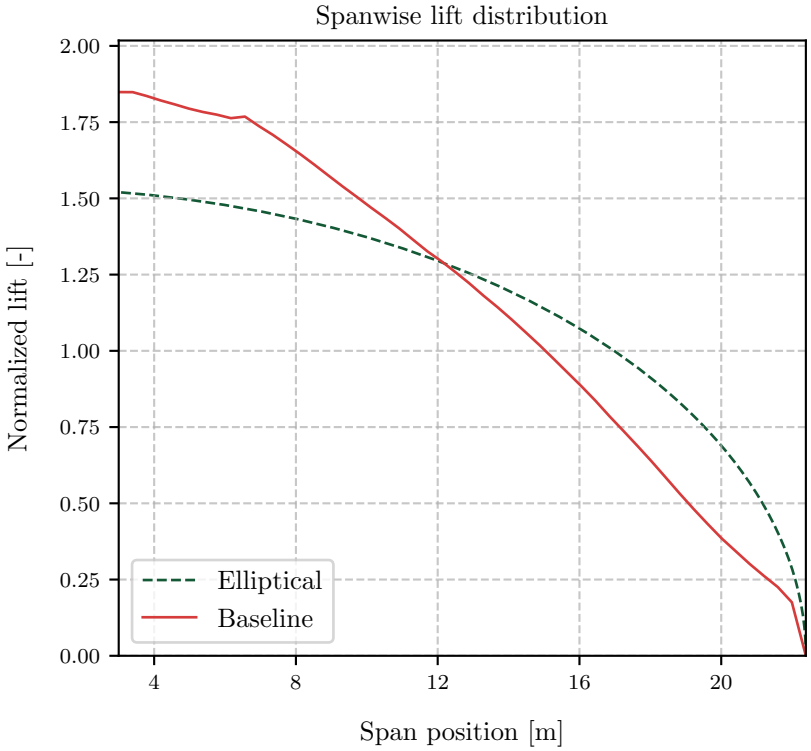
(b) Mission parameters of the *DLR-F25*

Parameter	Description	Value
H_{Cruise}	Cruise altitude	$12.192\,\mathrm{m}$
Ma_{Cruise}	Cruise Mach (Ma) number	0.78
R	Nominal range	$4,630\,\mathrm{km}$
S_{ref}	Wing area	$130.12\,\mathrm{m}^2$
$TSFC$	Thrust-specific fuel consumption	$1.35 \cdot 10^{-5}\,\mathrm{kg\,N\,s^{-1}}$

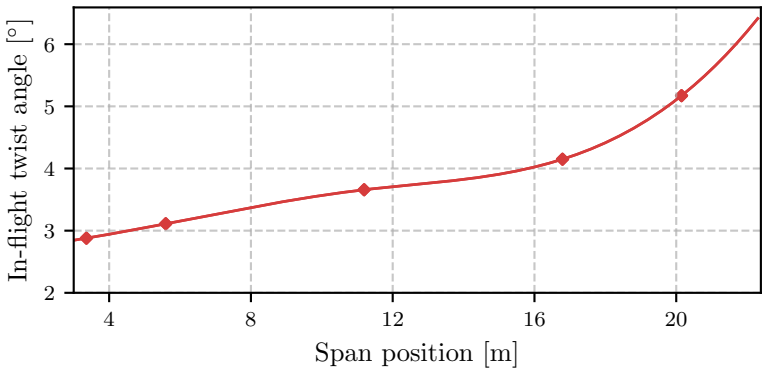
The aspect ratio (AR) of the *DLR-F25* wing is significantly higher than that of the simple transonic wing, so the lift is shifted further inboard to reduce the loads, as shown in Figure 4.7a. There are clearly visible pressure bumps on the wing surface, shown in the wing surface projection Figure 4.8 and the wing section cuts Figure 4.9. The thickness-to-chord ratio (t/c) ratio of the wing is about 0.16 at the wing root and decreases to 0.11 at the wing tip, which is shown in more detail in Figure 4.7c. The profile shape also changes along the span, as shown in Figure 4.9. In the in-flight shape, the twist at the wing root

is 2.9°, increasing to 6.4° at the wing tip.

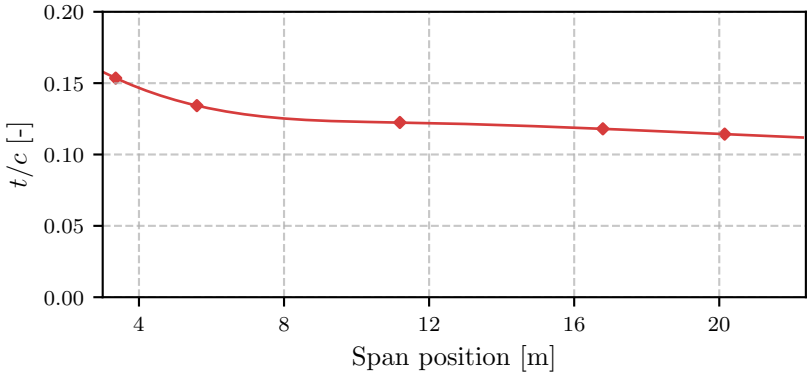
In its initial configuration, the *DLR-F25* has a coefficient of drag (C_D) of 0.0278, and a sized finite element mass m_{FE} of $75,821\,\mathrm{kg}$. On the reference mission, a mission fuel burn m_F of $13,446\,\mathrm{kg}$ is estimated.



(a) Spanwise lift distribution of the baseline *DLR-F25*



(b) Spanwise twist of the baseline *DLR-F25*



(c) Spanwise thickness-to-chord t/c ratio of the baseline *DLR-F25*

Figure 4.7: Spanwise characteristics of the baseline *DLR-F25*

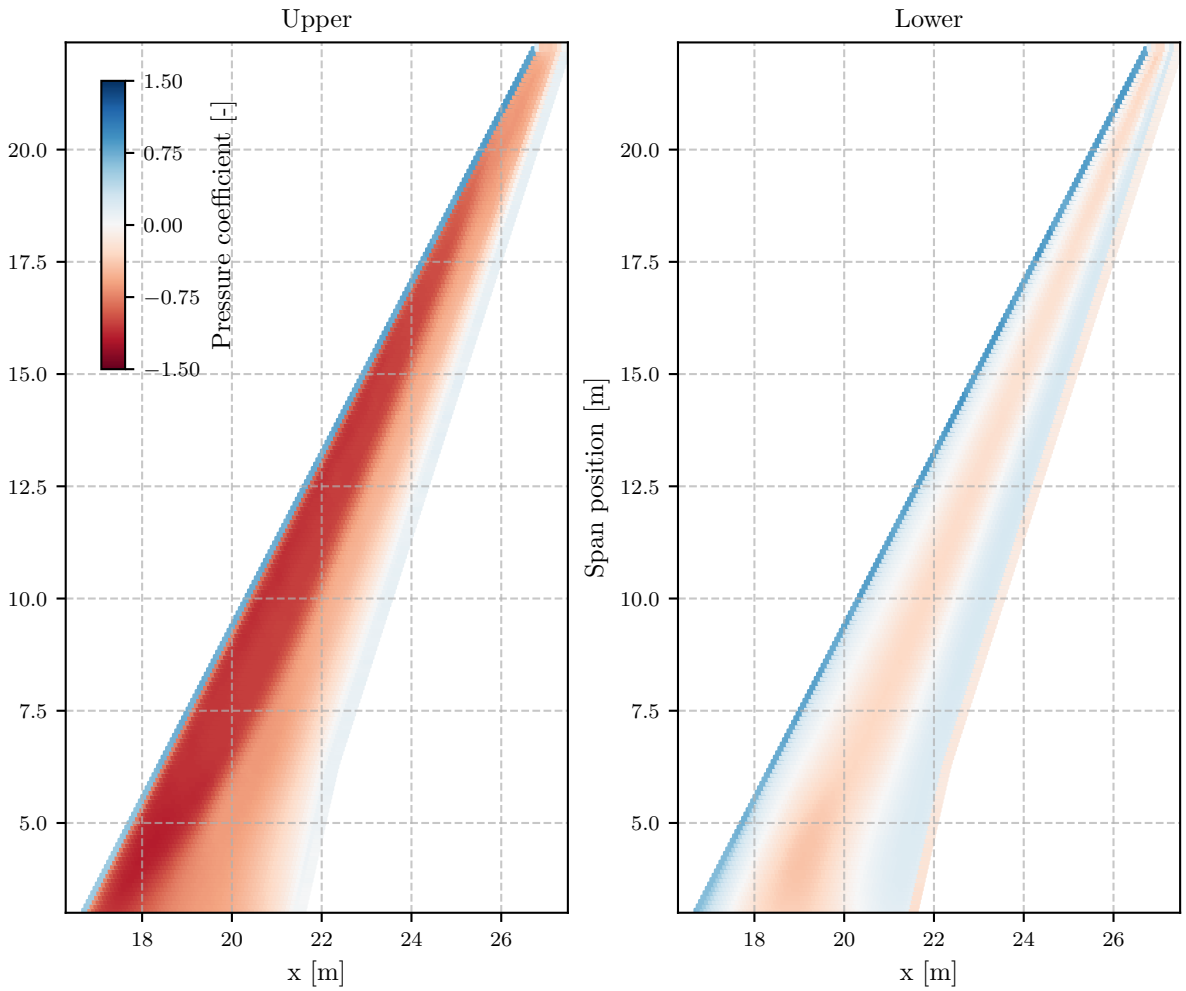


Figure 4.8: Wing surface pressure distribution of baseline *DLR-F25*

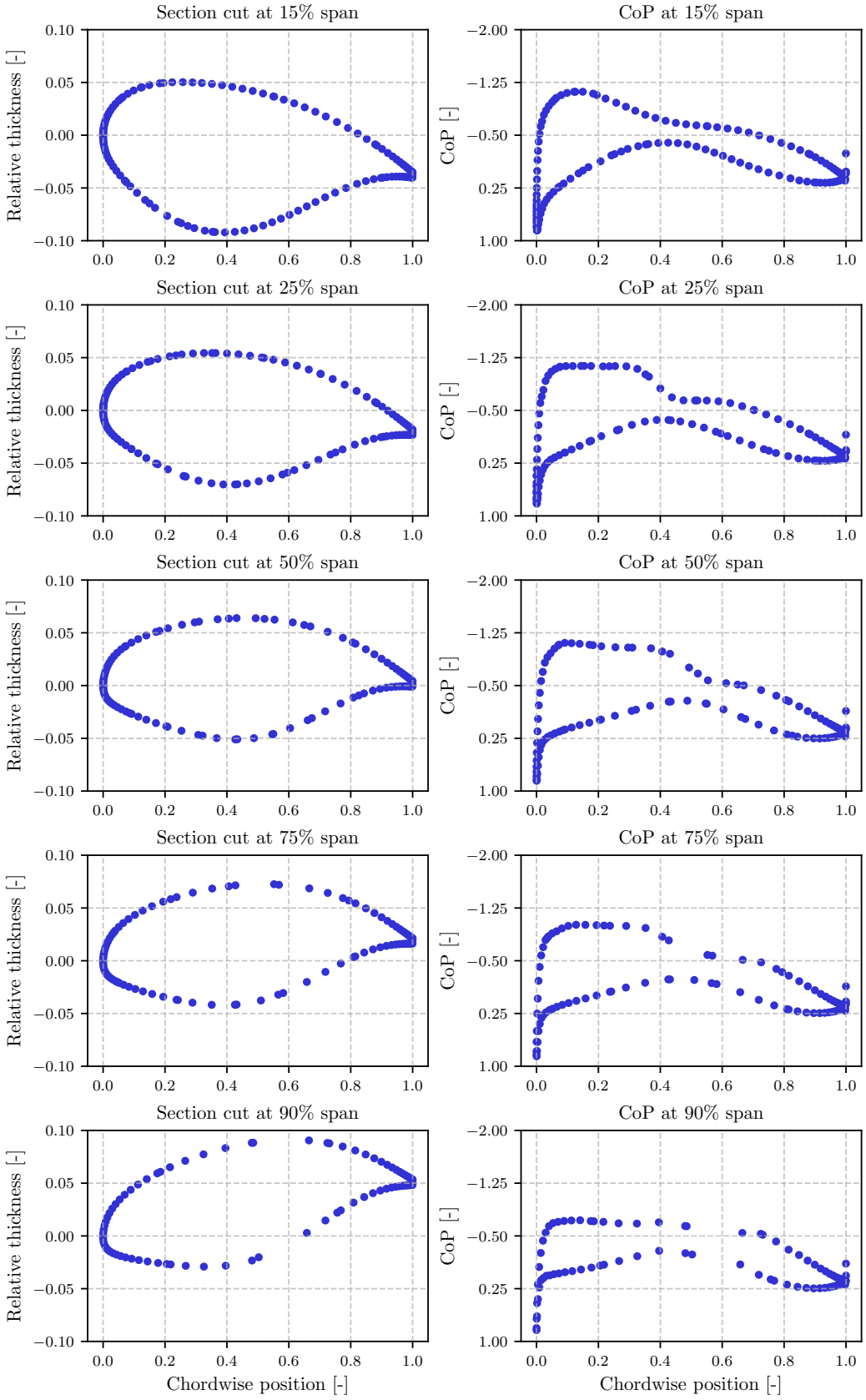


Figure 4.9: Section cuts of the baseline *DLR-F25*

5 MDO Formulations

In this section, the different MDO formulations of the processes from Section 3.2 are explained, starting with the *BL-EMB* formulation in Section 5.1. Section 5.2 shows how the *BL-IRT* formulation is implemented with the twist design variables as system-level design variables. In Section 5.3, the *BL-IRT* formulation uses a C_{mx} target as a system-level design variable. The last Section 5.4 shows the implementation of the *BL-BCD* formulation using a C_{mx}-target approach. While the *MDF* formulation serves as a reference for the comparisons, the reader is referred to Volle [8] for its detailed description. Notably, all formulations presented here retain their structure across test cases, requiring no case-specific modifications.

5.1 BL-EMB Formulation

A *XDSM* representation of the entire process is shown in Figure 5.1.

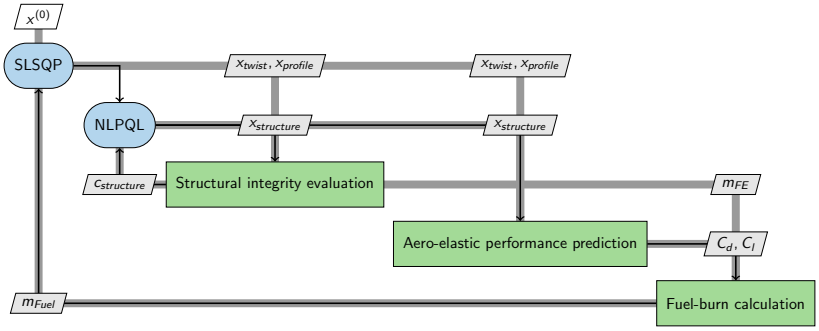


Figure 5.1: *XDSM* of the *BL-EMB* formulation

The *BL-EMB* uses the *SLSQP* optimizer to control all shape design parameters, specifically the wing twist and profile control parameters in both test cases. First, the structural sizing of Section 3.2.1 optimizes all structural design variables using the *LAGRANGE* internal

NLPQL optimizer. The objective function of the structure sizing process is the finite element mass m_{FE}, which is minimized while satisfying the structural constraints. Once the structure is sized given the current shape, the aerodynamic performance prediction process evaluates the drag for the given configuration.

The mission fuel burn m_F, obtained from Equation (4.3) for the simple transonic wing and from Equation (4.6) for the *DLR-F25*, serves as the objective function of the optimization process. Since the implementation relies on gradient-based optimization, each discipline must provide gradient information for its output variables. The objective function includes analytical derivatives with respect to all input parameters, while the aerodynamic performance prediction uses the adjoint method for sensitivity calculation described in Section 3.2.3.

To determine the structural mass gradients with respect to the shape design variables, the process uses one-sided, first-order *finite differences* applied to the structural sizing process. Since any parameter variation requires a complete resizing of the structure, the *finite difference* calculations are limited to the twist variables that have the most significant effect on the mass of the structure. While the structure sizing process affects the aerodynamic performance sensitivities, the corresponding gradients are not corrected for the structural influence to eliminate any cross-derivative information from the formulation.

5.2 BL-IRT Formulation

This section describes the *BL-IRT* process using the twist variables as the system-level design variables. The *XDSM* diagram of this process is shown in Figure 5.2.

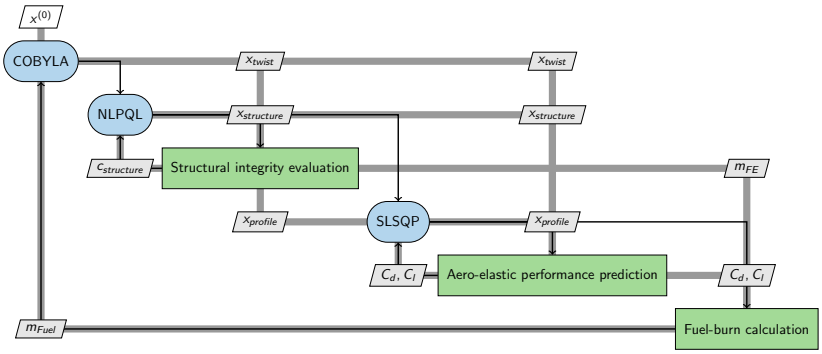


Figure 5.2: *XDSM* of the *BL-IRT* formulation

The twist variables are expected to have the strongest interdisciplinary effects, affecting both the aerodynamic performance and the load distribution and thus the structural mass,

and are therefore chosen as the system-level design variables. They are controlled by the gradient-free *COBYLA* optimizer. After the *COBYLA* optimizer has defined the values of the twist design variables $\mathbf{x}_{twist}$ for the current system-level iteration, the structural sizing process is performed for the current twist configuration of the wing. The structural sizing process optimizes the finite element mass m_{FE} under the structural constraints using the structural design parameters $\mathbf{x}_{structure}$. The dimensioned structural design variables are passed to the aerodynamic shape optimization process. This process uses the aerodynamic performance prediction process to obtain the drag of the current shape configuration. The shape design variables are controlled by the *SLSQP* optimizer. Since the twist design variables $\mathbf{x}_{twist}$ are controlled by the system-level optimizer, the shape design space consists of the profile shape parameters $\mathbf{x}_{profile}$. The aerodynamic optimization process is unconstrained, as trimming is handled internally by the aerodynamic performance prediction process.

The mission fuel burn m_F is the system-level objective of the optimization process and is calculated using the formulas Equation (4.3) for the simple transonic wing and Equation (4.6) for the *DLR-F25*.

5.3 BL-IRT Formulation with C_{mx}-Target Constraint

This section describes the *BL-IRT* process using a C_{mx}-target constraint as a system-level design variable. The *XDSM* diagram of the process is shown in Figure 5.3

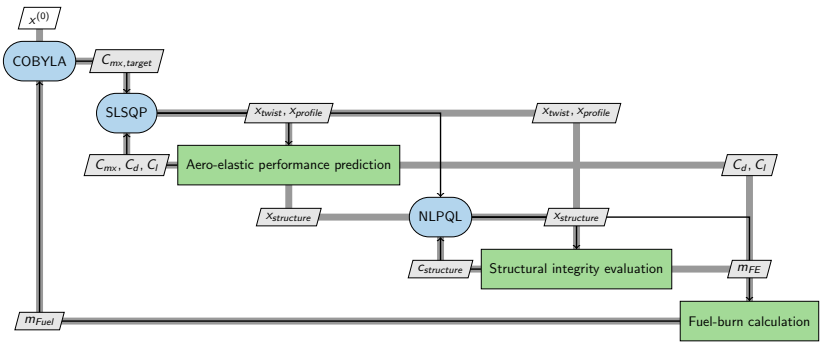


Figure 5.3: *XDSM* of the *BL-IRT* formulation with C_{mx}-target

When a C_{mx}-target has been set by the system level *COBYLA* optimizer, the aerodynamic shape optimization process is started. The *SLSQP* optimizer controls the shape design, which is composed of the twist design variables $\mathbf{x}_{twist}$ and the profile design variables $\mathbf{x}_{profile}$. This optimization minimizes drag while satisfying the C_{mx} target constraint, which is imposed as an equality constraint.

After the shape is optimized for the C_{mx} target, the structural sizing process minimizes the finite element mass m_{FE} with respect to the structural constraints. The mission fuel burn m_F is the system-level objective of the optimization process and is calculated using the formulas Equation (4.3) for the simple transonic wing and Equation (4.6) for the *DLR-F25*.

5.4 BL-BCD Formulation

This section describes the *BL-BCD* formulation, the *XDSM* diagram of the process is shown in Figure 5.4.

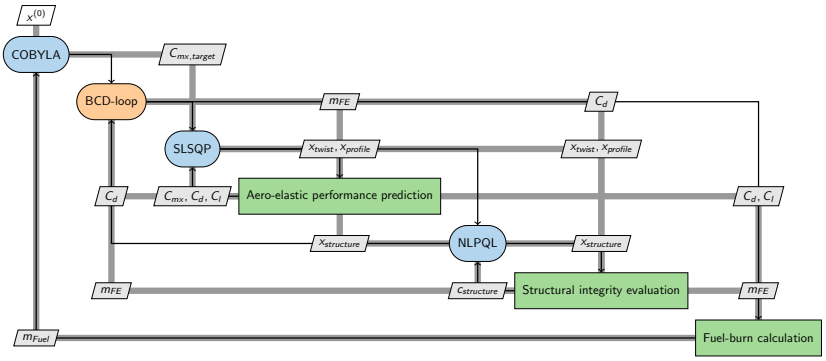


Figure 5.4: *XDSM* of the *BL-BCD* formulation

The *BL-BCD* formulation is similar to the *BL-IRT* approach, which uses a C_{mx} target as the top-level design variable. As in the *BL-IRT* formulation, the C_{mx} target is set by the *COBYLA* algorithm.

The disciplinary optimizations are identical to the disciplinary optimizations in the *BL-IRT* case with C_{mx}-target, but instead of being run once, they are run multiple times within the BCD loop. The BCD loop is a *Gauss-Seidel* fixed-point iteration. To determine the convergence of the algorithm, the structural sizing process passes the structural mass to the aerodynamic performance prediction, while the aerodynamic performance prediction passes C_D to the structural sizing process. If the changes in both parameters are within the convergence bounds, the loop terminates.

6 Results

This chapter presents the results of the different MDO formulations from Chapter 5 applied to the simple transonic wing in Section 6.1 and *DLR-F25* in Section 6.2.

After the presentation of the optimization results for both testcases, a starting point variation study is conducted on the simple transonic wing in Section 6.3 to determine the robustness of the optimization algorithms to perturbations of the starting point.

6.1 Simple Transonic Wing (University of Michigan) Optimization Results

This section presents the results of the coupled aero-structural optimization of the simple transonic wing with different *bilevel* formulations. In addition to the *BL-EMB*, *BL-IRT*, and *BL-BCD* formulations, an *MDF* optimization is conducted to create a reference to monolithic *MDO* formulations.

The optimization parameters utilized for the various MDO formulations are enumerated in Table 6.1. Every optimization is initiated with an initially sized structure, 0° of geometric twist across the wingspan, and profile parameters according to the baseline *RAE2822* airfoil. The baseline aerodynamic performance data is presented in Section 4.1.

The trend of all optimizations over wall clock time is plotted in Figure 6.1, displaying the evolution of the mission fuel burn m_F in Figure 6.1a, the coefficient of drag (C_D) in Figure 6.1b, and the finite element mass m_{FE} in Figure 6.1c. While the *MDF* formulation coverged in around 17 hours, the *bilevel* formulations required significantly more time, with the exception of the *BL-EMB* formulation, which converged in around 24 hours. The longest computation time is required by the *BL-BCD* formulation at 68 hours. Especially the *BL-IRT* formulation without a C_{mx}-target constraint shows still strong

Table 6.1: Optimization parameters for the simple transonic wing

(a) Top-level optimization parameters

Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target	*BL-EMB*	*BL-BCD*
Objective scaling	10^{-4}	1	1	10^{-3}	1
BCD convergence radius	—	—	—	—	10^{-6}
Max. BCD iterations	—	—	—	—	4

(b) Aerodynamic optimization parameters

Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target	*BL-EMB*	*BL-BCD*
Max. optimization iterations	—	25	25	25	25
Trim iterations	—	5	5	5	5
Aero-structural coupling iterations	20	20	20	20	20
Aero-structural adjoint coupling iterations	5	5	5	5	5
Primal CFD minimum residual	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Adjoint CFD minimum residual	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Objective convergence radius	—	10^{-5}	10^{-5}	—	10^{-5}
Design variable convergence radius	—	10^{-5}	10^{-5}	—	10^{-5}
Objective scaling	—	10^{2}	10^{2}	—	10^{2}

(c) Structure optimization parameters

Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target	*BL-EMB*	*BL-BCD*
Max. iterations	—	20	20	20	20
FD step size	—	—	—	0.5	—
Objective scaling	3	3	3	3	3

variations in the objective before the optimization is terminated due to an error in the used version of *OpenMPI*. This error terminates the execution of all *bilevel* formulations. The computationally most expensive *BL-BCD* formulation can only 7 system level iterations, before termination. Hence, further optimization potential could be expected, if the formulation was to converge further.

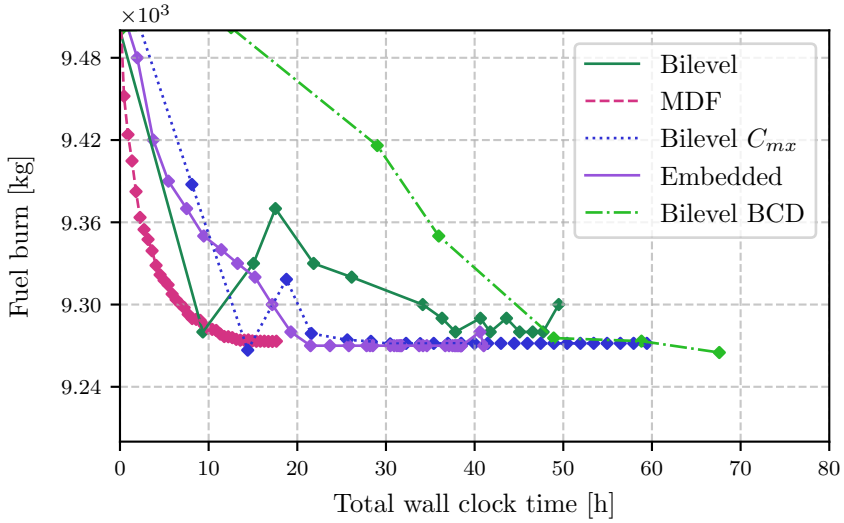
The converged mission fuel burn m_F^*, coefficient of drag C_D, and finite element mass m_{FE} are listed in Table 6.2. The lowest mission fuel burn is achived by the *BL-BCD* formulation at 9265 kg, while the highest is attained by the *BL-IRT* formulation at 9280 kg. With the exception of the *BL-IRT* formulation, all other *bilevel* optimizations result in a lower optimized mission fuel burn than the *MDF* optimization, despite being less converged. A notable observation is that the structural mass optimized by the *MDF* formulation is considerably lower compared to the *bilevel* scenarios, while the drag levels are notably higher.

Table 6.2: Optimal values of the simple transonic wing optimization

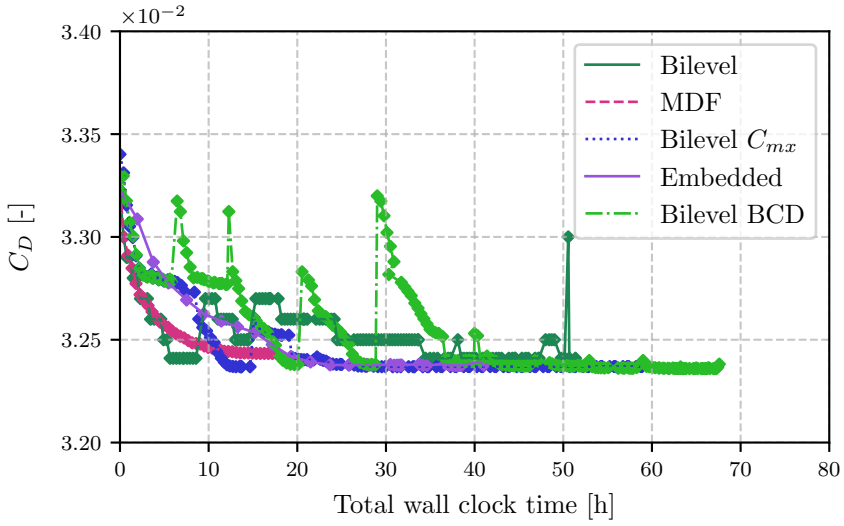
Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target	*BL-BCD*	*BL-EMB*
Mission fuel burn m^*_{fuel}	9273 kg	9280 kg	9272 kg	9265 kg	9269 kg
Coefficient of drag C^*_D	$3.243 \cdot 10^{-2}$	$3.239 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.238 \cdot 10^{-2}$
Finite element mass m^*_{FE}	828 kg	847 kg	847 kg	842 kg	843 kg

In Figure 6.2a, the optimized lift distributions from all formulations are depicted. It is evident that all formulations shifted the lift inboards compared to the baseline lift distribution. This outcome is to be expected, as the baseline distribution is shifted outboards from the elliptical distribution, whereas the combined aero-structural optimum is inboards of the elliptical distribution. While the optimized lift distributions are qualitatively similar, differences between the different formulations can be observed. The lift distribution optimized by both the *BL-EMB* and *BL-BCD* formulations exhibited a greater similarity to the baseline at the root compared to the distributions derived from the other formulations The lift distribution of the *BL-IRT* formulation demonstrated a high degree of similarity to the *MDF* distribution across the entire span width Near the wingtip, all of the different distribution align again, whilst significantly deviating from the baseline curve.

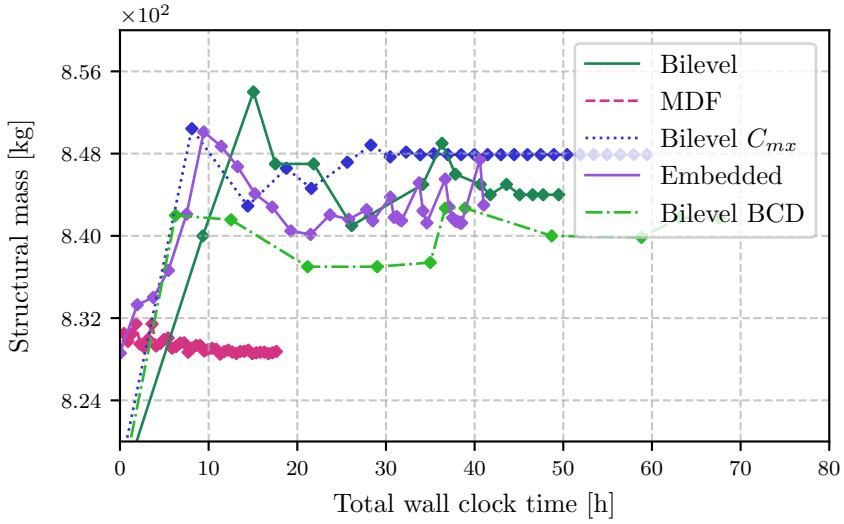
The in-flight twists of the wings are depicted in Figure 6.2b, with a nosedown pitch of the wing counted in a positive direction. Notable differences exist between the formulations in the obtained in-flight twists. While all optimizations decreased the twist at the wing root compared to the baseline configuration, the extent of this reduction varies across all



(a) Mission fuel burn m_F



(b) Coefficient of drag C_D



(c) Finite element mass m_{FE}

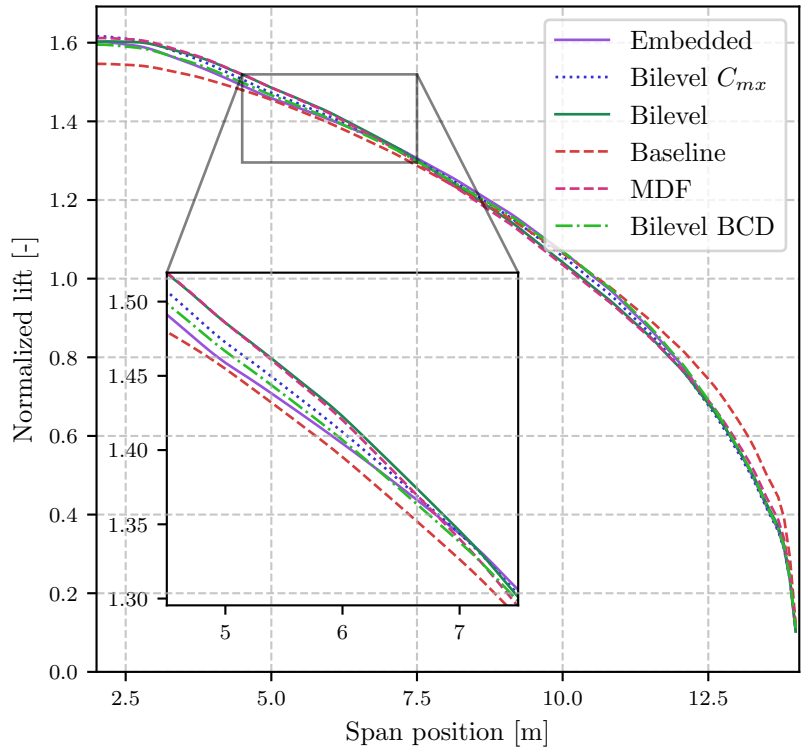
Figure 6.1: Optimization progress of the simple transonic wing

formulations. Notably, the *MDF* formulation exhibits the most significant reduction in twist, reaching approximately 0.8°. At the wingtip, the twists obtained from the *MDF* and *BL-IRT* are similar to the baseline configuration, while the *BL-EMB*, *BL-BCD*, and *BL-IRT* with C_{mx}-target formulations increase the twist by up to 1°.

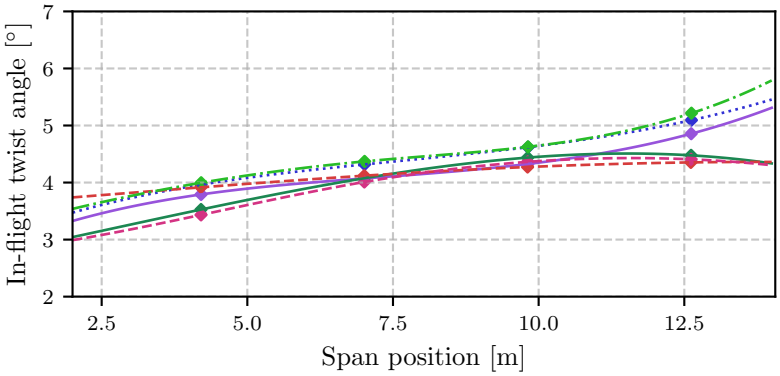
The thickness-to-chord ratio t/c of the wing is plotted in Figure 6.2c. All formulations reduce the thickness to chord ratio by approximately 0.025 uniformly across the wingspan. The obtained thicknesses are similar for all optimization formulations.

In Figure 6.3, section cuts of the optimized wings are depicted. A comparison of the obtained profiles reveals that they are similar in shape, but differ in their twist. This observation is reflected in the chordwise pressure distribution, which is nearly identical for all formulations at the examined spanwise locations. At all examined spanwise locations, the negative pressure coefficient on the upper side is further decreased near the leading edge and increased in the middle region. This equalization leads to a flatter chordwise pressure distribution, thereby removing the slight shock bump present in the baseline configuration at the wing root. On the lower side of the wing, an adverse effect can be observed, as the positive pressure coefficient is increased further in the middle chord region. The combined changes present on the upper and lower wing sides cause a shift of the lift generation from the upper to the lower side of the wing.

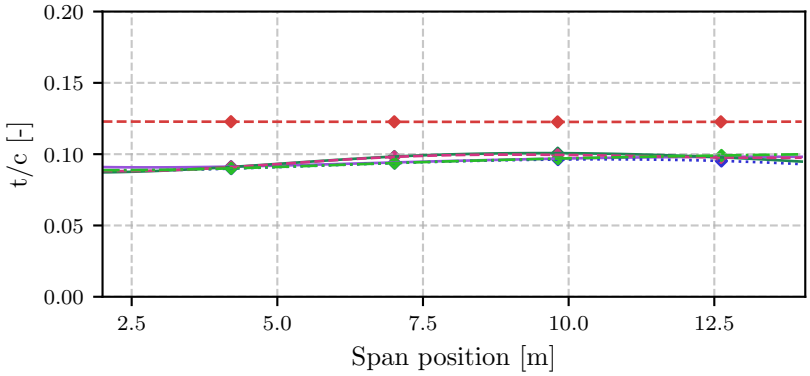
In summary, the optimization scenarios effectively optimized the simple transonic wing, with analogous qualitative alterations introduced to the design. However, these alterations vary quantitatively between the formulations. This is most notably evident in the spanwise lift distributions of the optimized wings and is reflected in the variation in in-flight twist shapes.



(a) Spanwise lift-distribution



(b) Spanwise twist



(c) Spanwise thickness-to-chord ratio

Figure 6.2: Different spanwise characteristics of the optimized simple transonic wing

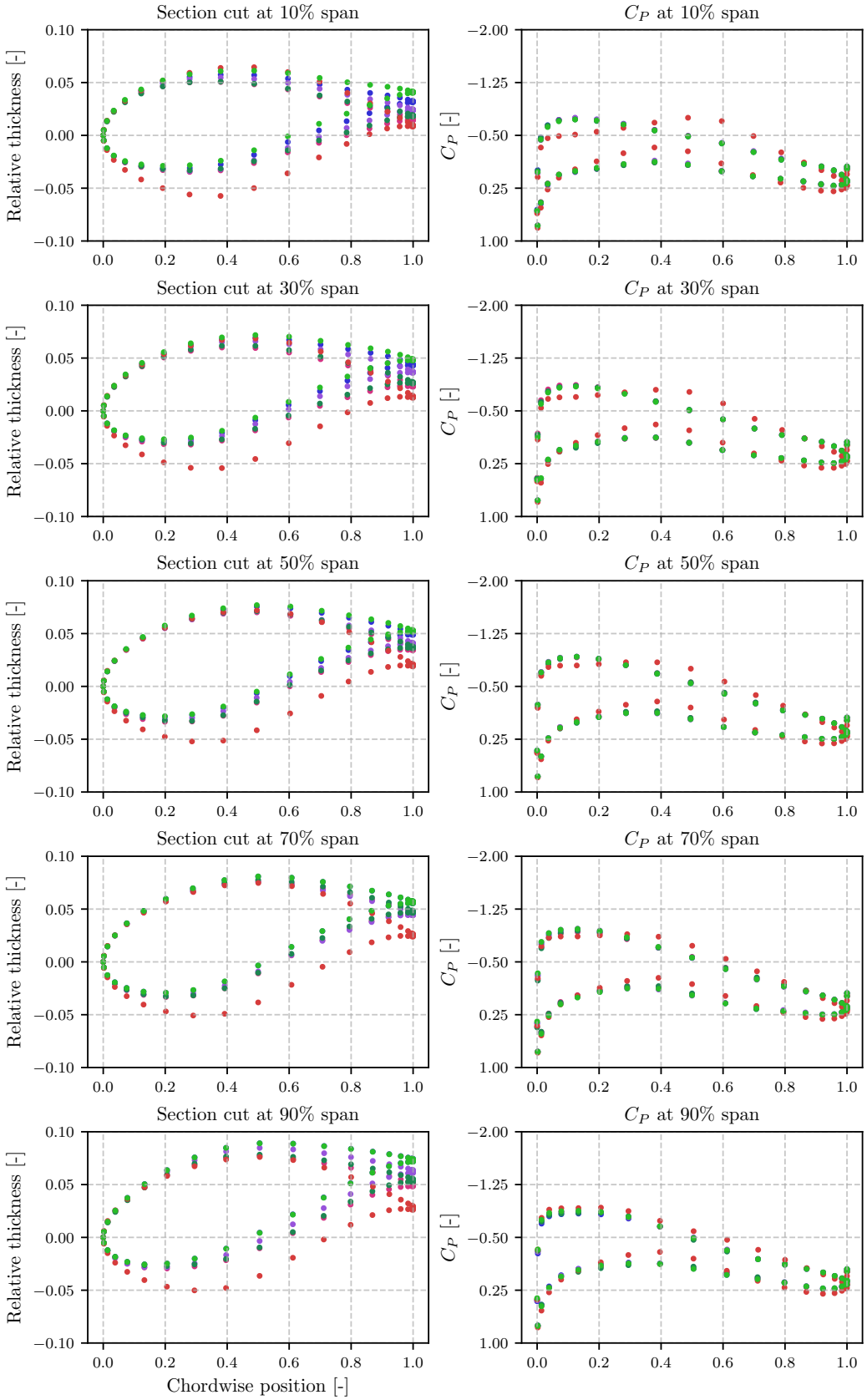


Figure 6.3: Optimized section cuts of the simple transonic wing

6.2 *DLR-F25* Optimization Results

This section presents the results of the coupled aero-structural optimization of the *DLR-F25*. The formulations studied are the *BL-IRT*, the *BL-IRT* with C_{mx} target constraint, and the *MDF* formulation as a reference. However, due to difficulties encountered in the recurrent deformation of the structural model required for the finite differencing of the structural sizing process, no representative results could be generated for the *BL-EMB*. All optimizations are started from the baseline configuration with an initially sized structure. The optimization hyperparameters are listed in Table 6.3.

The optimization progress of the three formulations is shown in Figure 6.4, with the mission fuel consumption m_F in Figure 6.4a, the drag coefficient C_D in Figure 6.4b, and the finite element mass m_{FE} in Figure 6.4c. The *MDF* optimization was run for about 175 hours, or about seven days, while the *BL-IRT* formulation was run for 320 hours, or about 13 days. In addition, the *BL-IRT* formulation with the C_{mx} target constraint was run for 280 hours, or 11 days. It is noteworthy that none of the formulations reached a formal convergence limit or the maximum number of optimization iterations. However, they were all aborted due to the same *OpenMPI* error observed in the simple transonic wing optimization.

The optimal values obtained by the different formulations are listed in Table 6.4. The optimal values for the mission fuel consumption m_F are within a $\pm 4\,\mathrm{kg}$ range. As in the simple transonic wing optimization, the *MDF* formulation yielded the lowest finite element mass m_{FE}, but also the highest drag. The *BL-IRT* formulation with the C_{mx} target constraint obtained the lowest mission fuel burn and drag values, but the highest structural mass of all formulations considered.

The optimized lift distributions are shown in Figure 6.5a. All lift distributions are shifted outboard from the baseline lift distribution, with the *MDF* and *BL-IRT* formulations being nearly identical. As expected from the higher mass and lower drag achieved by the *BL-IRT* with C_{mx} target constraint, the resulting lift distribution is shifted slightly more outboard.

The optimized in-flight twists are shown in Figure 6.5b. All formulations show a similar change compared to the baseline shape, as all optimizations increase the twist at the wing root by about 0.5°, while decreasing it near the wing tip. Since the parameterization of the wing does not include changes in the thickness profile, the spanwise thickness-to-chord t/c ratio remains unchanged.

Table 6.3: Optimization parameters for the *DLR-F25*

(a) Top-level optimization parameters

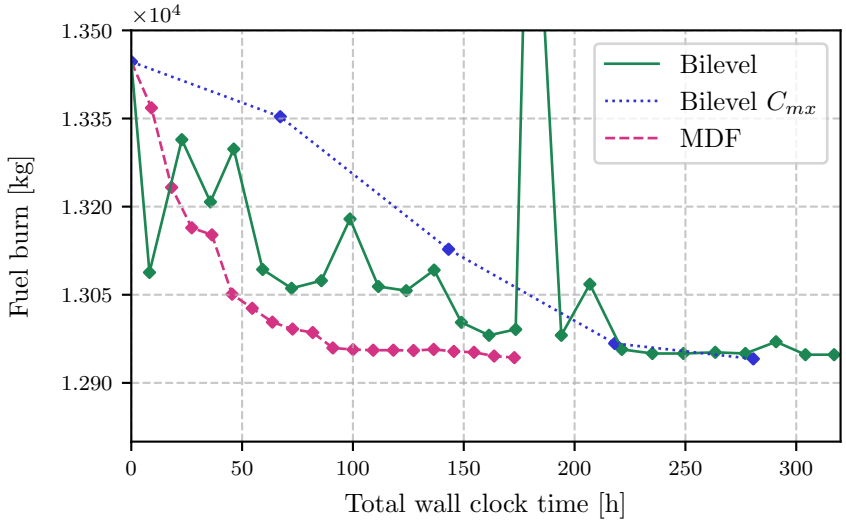
Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target
Objective scaling	10^{-5}	1	1

(b) Aerodynamic optimization parameters

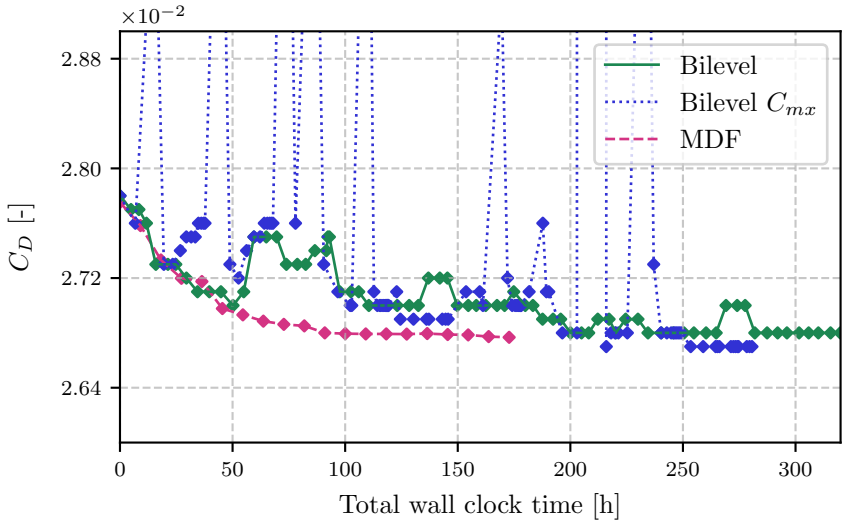
Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target
Max. optimization iterations	—	25	25
Trim iterations	5	5	5
Aero-structural coupling iterations	20	20	20
Aero-structural adjoint coupling iterations	4	4	4
Primal CFD minimum residual	10^{-5}	10^{-5}	10^{-5}
Adjoint CFD minimum residual	10^{-6}	10^{-6}	10^{-6}
Objective convergence radius	—	10^{-5}	10^{-5}
Design variable convergence radius	—	10^{-5}	10^{-5}
Objective scaling	—	1	10^{2}

(c) Structure optimization parameters

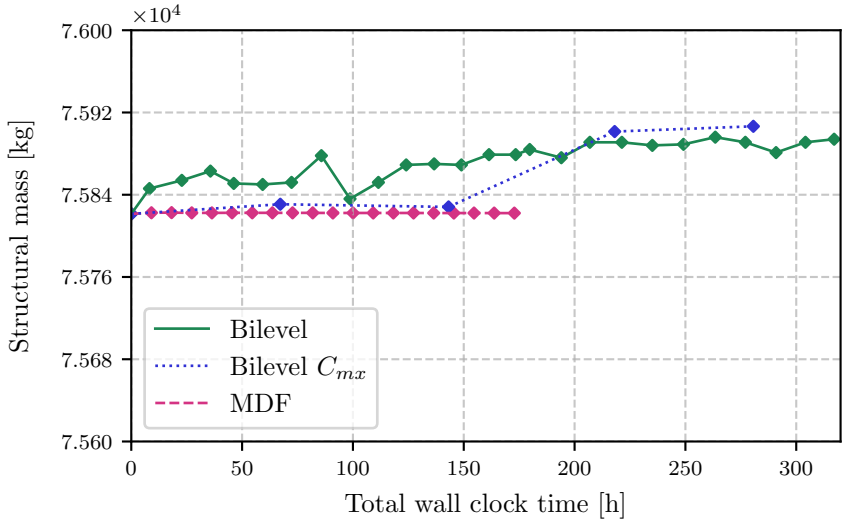
Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target
Max. iterations	—	20	20
FD step size	—	—	—
Objective scaling	—	10^{-3}	10^{-3}



(a) Mission fuel burn m_F



(b) Coefficient of drag C_D

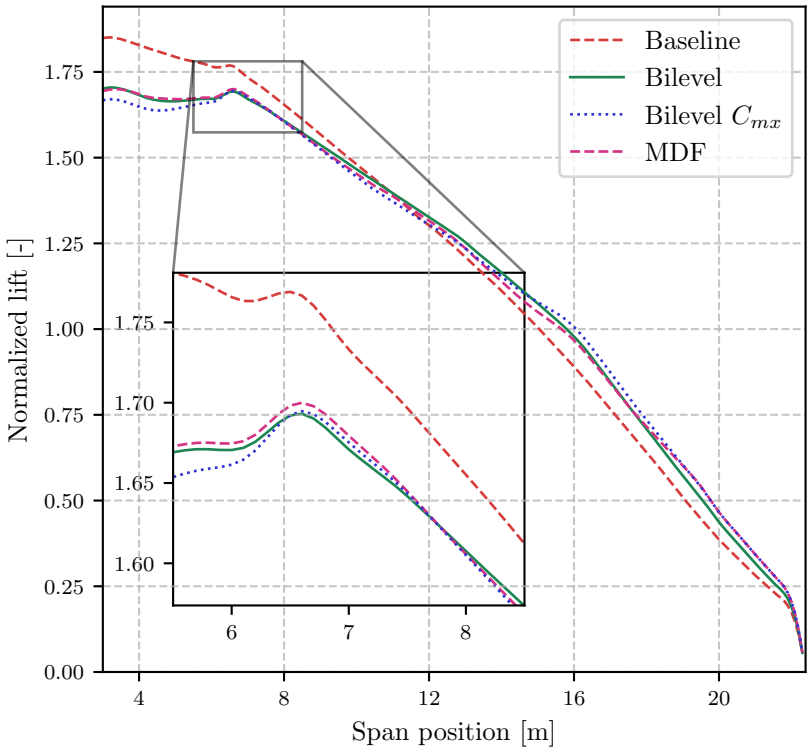


(c) Finite element mass m_{FE}

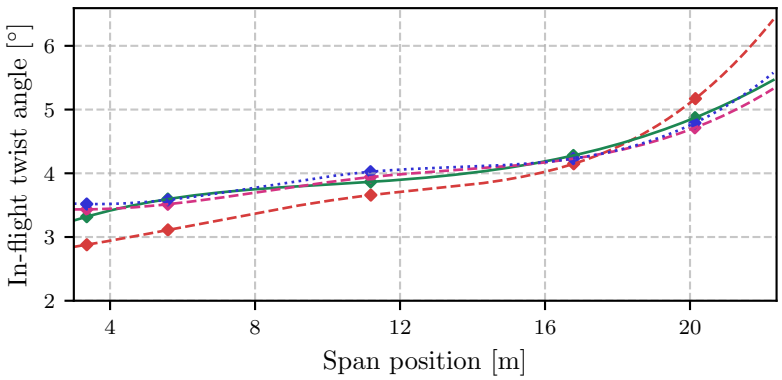
Figure 6.4: Optimization progress of *DLR-F25*

Table 6.4: Optimal values of the *DLR-F25* optimization

Parameter	*MDF*	*BL-IRT*	*BL-IRT* C_{mx}-target
Mission fuel burn m_F^*	12,943 kg	12,947 kg	12,941 kg
Coefficient of drag C_D^*	$2.676 \cdot 10^{-2}$	$2.676 \cdot 10^{-2}$	$2.674 \cdot 10^{-2}$
Finite element mass m_{FE}^*	75,821 kg	75,891 kg	75,907 kg



(a) Spanwise lift-distribution



(b) Spanwise twist

Figure 6.5: Different spanwise characteristics of the optimized *DLR-F25*

Figure 6.6 shows the optimized section cuts and chordwise pressure distributions of the different optimization formulations. Since the thickness of the airfoil is not included in the degrees of freedom of the optimization, the most notable difference between the optimized and baseline airfoils is the different twist shapes. Regarding the pressure distribution along the wing chord, all optimizations show similar results. For the section cuts at 15%, 25% and 50% span, the pressure coefficient at the trailing edge is reduced on the upper side, which removes the shock bump previously present at the half chord location. The pressure distribution on the underside remains almost unchanged. At the outboard location at 75% span, all optimizations decrease the pressure coefficient at the half chord, creating a slight negative bump. At the 90% spanwise location the most significant difference between the optimizations is visible, as the *MDF* and *BL-IRT* with C_{mx} target constraint optimizations significantly reduce the pressure coefficient at the leading edge of the upper wing side, while the *BL-IRT* optimization shows less of this effect.

During the optimization process it was noticed that the line search iterations of the aerodynamic optimization often fail and cause spikes in the C_D of the optimization step. Further investigations showed that the linearization accuracies for the *DLR-F25* case are significantly lower than for the simple transonic wing. The linearization accuracy describes the quotient of the decrease in the objective function value by the expected decrease based on the linearization.

$$\text{Linearization Accuracy} = \frac{\mathbf{f}_n - \mathbf{f}_{n-1}}{(\mathbf{x}_n - \mathbf{x}_{n-1})^T \cdot \nabla \mathbf{f}}. \tag{6.1}$$

This trend is plotted for the *MDF* optimizations of both cases in Figure 6.7.

This is a challenge for all formulations considered. In particular, the *BL-IRT* formulation is affected, since satisfying the C_{mx} objective equality constraint requires trustworthy gradient information. Thus, the objective scaling of the aerodynamic objective presented in Table 6.3 had to be reduced to a low value of 0.1 in order to reduce the step sizes of the optimization.

The C_{mx} target constraint residuals of the simple transonic wing and *DLR-F25 BL-IRT* optimizations with C_{mx}-target constraint are shown in Figure 6.8. The maximum residual for both problems was defined as $R_{Cmx} \leq 10^{-4}$, represented by the dashed line, showing that the optimization violated this constraint for all iterations for the *DLR-F25*, while satisfying it for the simple transonic wing.

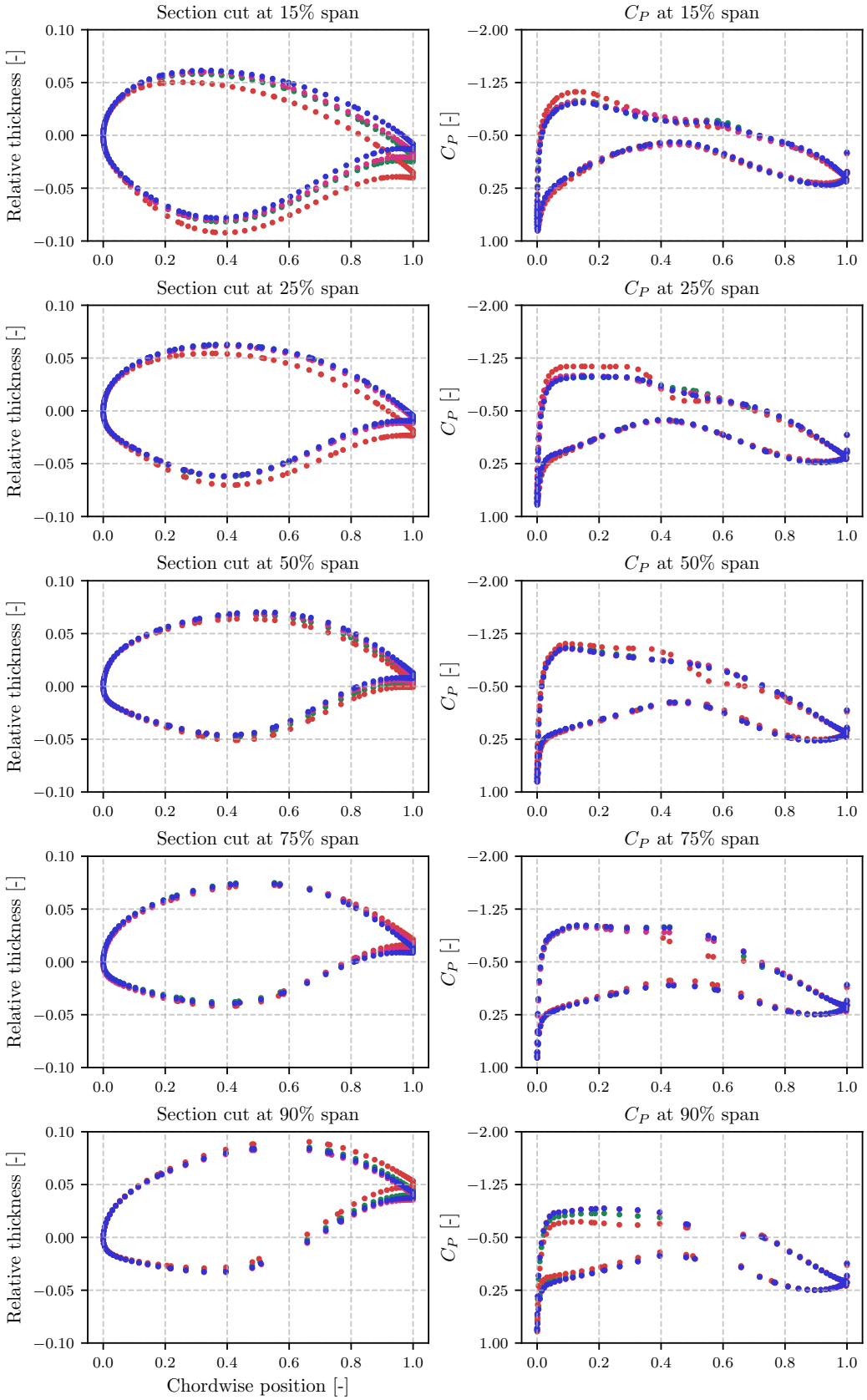


Figure 6.6: Optimized section cuts of the *DLR-F25*

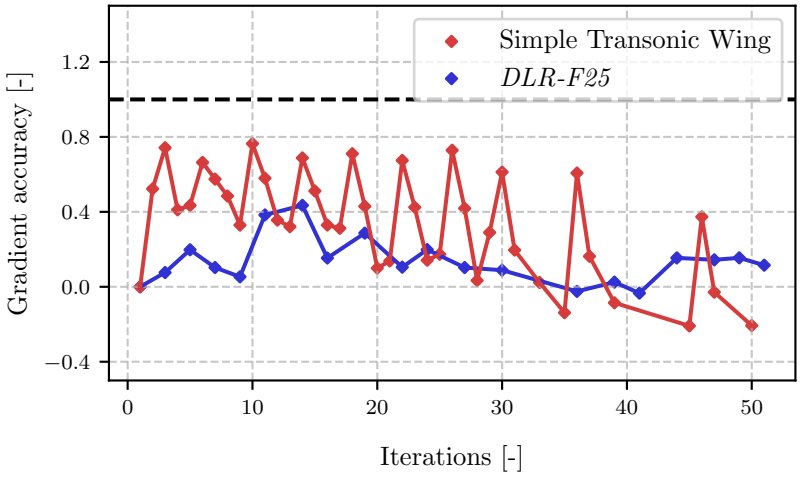


Figure 6.7: Linearization accuracy comparison between the simple transonic wing and *DLR-F25 MDF* optimizations, according to Equation (6.1)

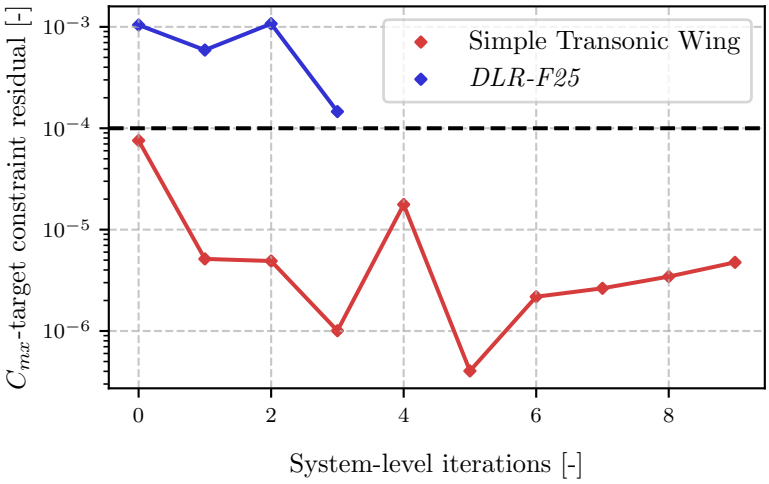


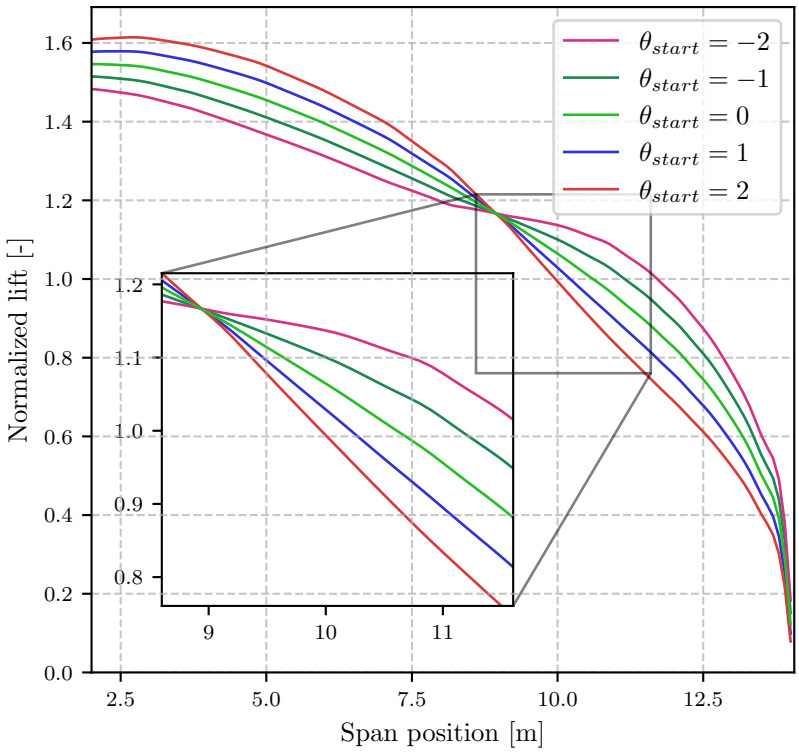
Figure 6.8: C_{mx}-target constraint residual comparison between the simple transonic wing and *DLR-F25 MDF* optimizations

6.3 Starting Point Variation of Simple Transonic Wing

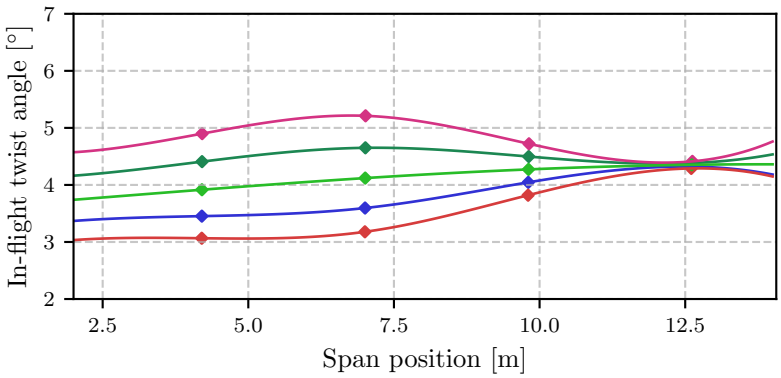
DAVID's research [30] reveals the limitation that the convergence of the *BL-IRT* formulation depends on the starting point of the optimization. This dependency poses significant challenges for industrial applications, as it requires either precise a priori knowledge of the approximate optimal design variables, or multiple optimization runs with different starting points to ensure that the objective function is sufficiently minimized. Neither approach is feasible for industrial applications. While the BL-BCD formulation successfully overcomes this dependence in his study, DAVID's numerical experiments cover only the analytical SSBJ test case. This study extends this investigation to high-fidelity problems to determine whether these convergence problems persist in industrial applications, using the simple transonic wing with varied initial twist variables. The initial parameter variation is performed for the *BL-IRT*, *BL-IRT* with C_{mx} target, and *BL-BCD* formulations.

To test the influence of starting point variation for high-fidelity MDO methods, the simple transonic wing is used as a benchmark test case, since it has been successfully optimized by all *bilevel* formulations. Due to the higher computational cost of this testcase compared to analytical formulas, the number of samples must be kept small. Therefore, instead of random parameter sampling, technically relevant changes in the initial configuration of the wing are to be introduced. The wing twist parameters are chosen as the varied input parameters because changing the twist of the wing directly affects the spanwise load distribution of the wing, thus introducing challenging coupling effects into the study. Since the test case consists of a single wing, changing the twist uniformly across the span has little effect as it is almost completely counteracted by the trim. Instead, the twist profile is changed from the wing root to the center of the wing while keeping the twist constant at the outboard locations. The wing root twist changes are $-2°$, $-1°$, $1°$, and $2°$, as well as the baseline configuration. The structure of the wing is initially sized according to the changes in the twist profile of the wing.

The changed geometric twists lead to the in-flight twist profiles in Figure 6.9b. The resulting changes in the lift distribution of the wing are shown in Figure 6.9a. For the scenarios where the twist is decreased, the lift is shifted outboard, while for the scenarios where the root twist is increased, the lift is shifted inboard. The changed lift conditions are reflected in the initial performance parameters, listed for each case in Table 6.5. The parameters considered are the drag coefficient C_D in cruise condition, the finite element mass m_{FE}, and the fuel burn m_{fuel} for the reference mission of the test case.



(a) Spanwise lift-distribution of different initial configurations of the simple transonic wing



(b) Spanwise twist of different initial configurations of the simple transonic wing

Figure 6.9: Inital configurations of different starting points for the simple transonic wing

Table 6.5: Performance parameters of different initial configurations of the simple transonic wing

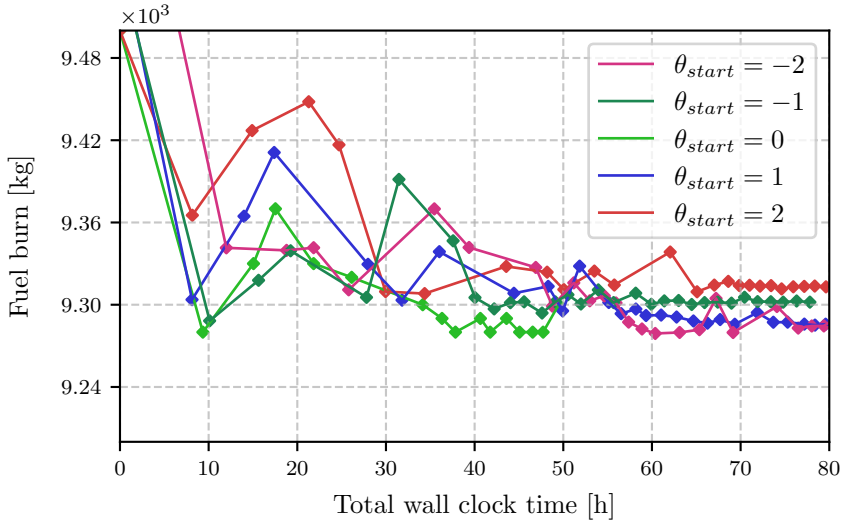
Parameter	$\theta_{start} = -2°$	$\theta_{start} = -1°$	$\theta_{start} = 0°$	$\theta_{start} = 1°$	$\theta_{start} = 2°$
Mission fuel burn m_{fuel}	9699 kg	9546 kg	9500 kg	9489 kg	9558 kg
Coefficient of drag C_D	$3.381 \cdot 10^{-2}$	$3.323 \cdot 10^{-2}$	$3.325 \cdot 10^{-2}$	$3.344 \cdot 10^{-2}$	$3.34 \cdot 10^{-2}$
Finite element mass m_{FE}	825 kg	817 kg	815 kg	804 kg	810 kg

6.3.1 Starting Point Variation of *BL-IRT* formulation

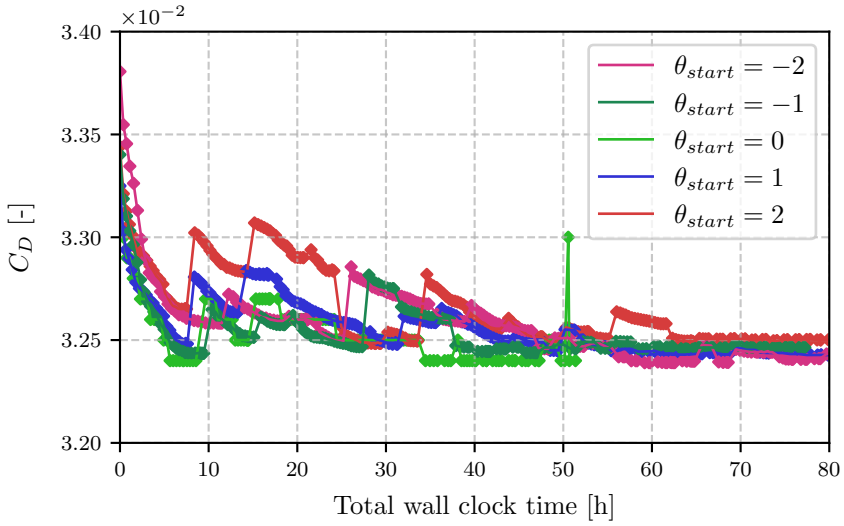
For the *BL-IRT* formulation, there are significant differences in the found optimal value between the starting points, as shown in Figure 6.12a. The maximum absolute difference between the starting points is about 32 kg in optimized mission fuel burn m_F^*, which is more than the maximum difference between the formulations found in Section 6.1. The different results in the attained goal are also present in the aerodynamic drag shown in Figure 6.10b and the structural mass shown in Figure 6.10c. All optimized performance parameters are listed for the different starting points in Table 6.6. To visualize the differences in aerodynamic shape and structural sizing parameters, the mean and standard deviation of the optimized in-flight twists are plotted over all tested start points in Figure 6.11. As can be seen, there are significant differences in in-flight shape between the optimized designs for the different start points.

Table 6.6: Optimal values of the *BL-IRT* optimization of the simple transonic wing with varied starting points

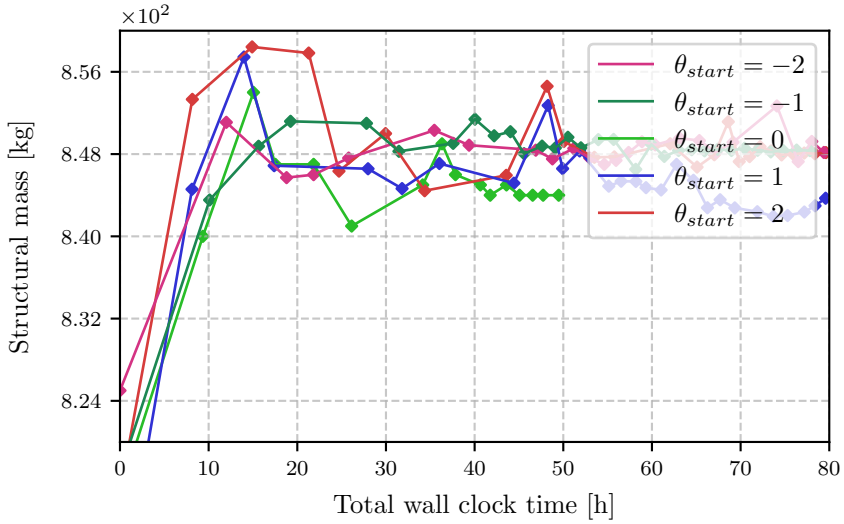
Parameter	$\theta_{start} = -2°$	$\theta_{start} = -1°$	$\theta_{start} = 0°$	$\theta_{start} = 1°$	$\theta_{start} = 2°$
Mission fuel burn m^*_{fuel}	9302 kg	9300 kg	9280 kg	9284 kg	9312 kg
Coefficient of drag C^*_D	$3.241 \cdot 10^{-2}$	$3.246 \cdot 10^{-2}$	$3.239 \cdot 10^{-2}$	$3.242 \cdot 10^{-2}$	$3.25 \cdot 10^{-2}$
Finite element mass m^*_{FE}	848 kg	848 kg	847 kg	843 kg	848 kg



(a) *BL-IRT* optimization objective function of the simple transonic wing with starting point variation



(b) *BL-IRT* optimization aerodynamic objective function of the simple transonic wing with starting point variation



(c) *BL-IRT* optimization structure objective function of the simple transonic wing with starting point variation

Figure 6.10: Starting point variation of *BL-IRT* optimization of the simple transonic wing

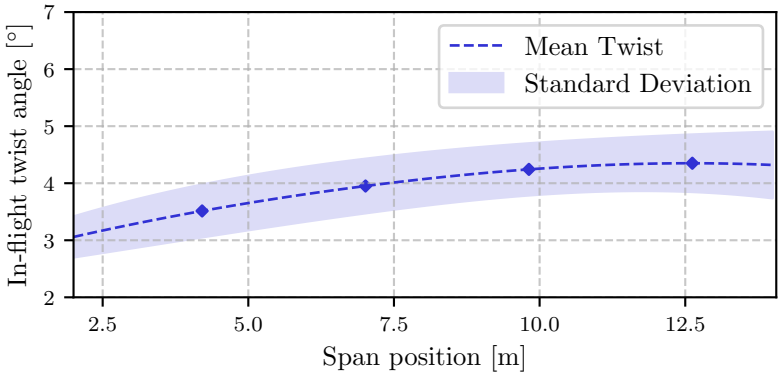


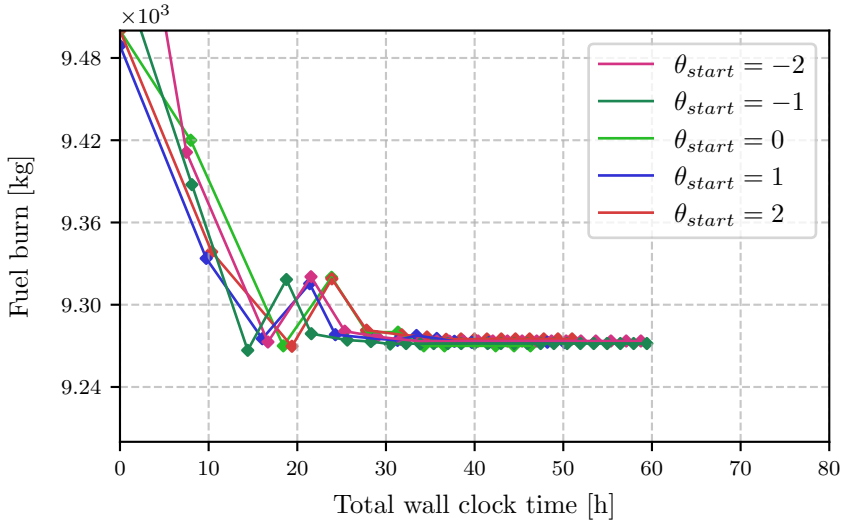
Figure 6.11: Mean and standard deviations of the *BL-IRT* optimized in-flight twists for varied starting points

6.3.2 Starting Point Variation of *BL-IRT* formulation with C_{mx}-target constraint

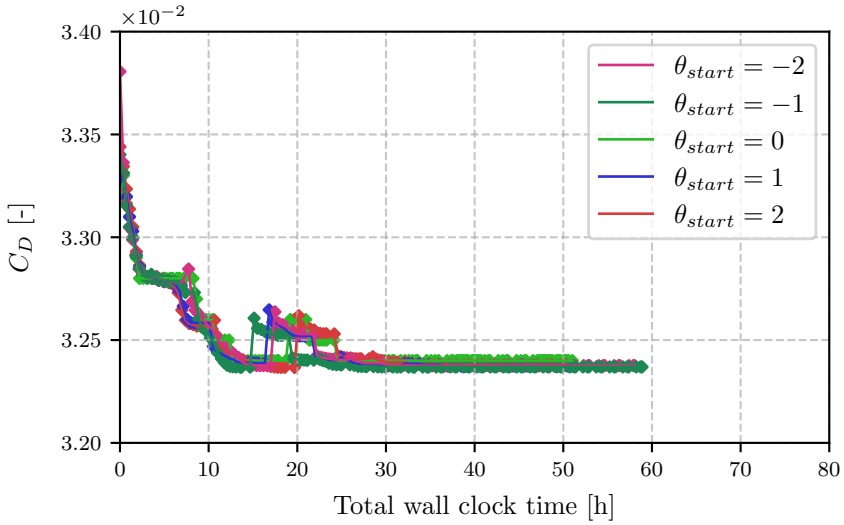
The use of a C_{mx} target constraint mitigates this effect for the *BL-IRT* formulation, as the optimal values found are more closely aligned and all lie within a 3 kg mission fuel burn m_F of each other, as shown in Figure 6.12a and Table 6.7. Both structural mass and aerodynamic drag are also much more similar, with only small differences between the initial values, as shown in Figure 6.12b and Figure 6.12c. The in-flight twists of the optimized solutions are depicted in Figure 6.13 with their mean value and the standard deviation. The band of standard deviation is significantly thinner compared to the in-flight shapes optimized without a C_{mx}-target in Figure 6.11.

Table 6.7: Optimal values of the *BL-IRT* optimization with C_{mx}-target of the simple transonic wing with varied starting points

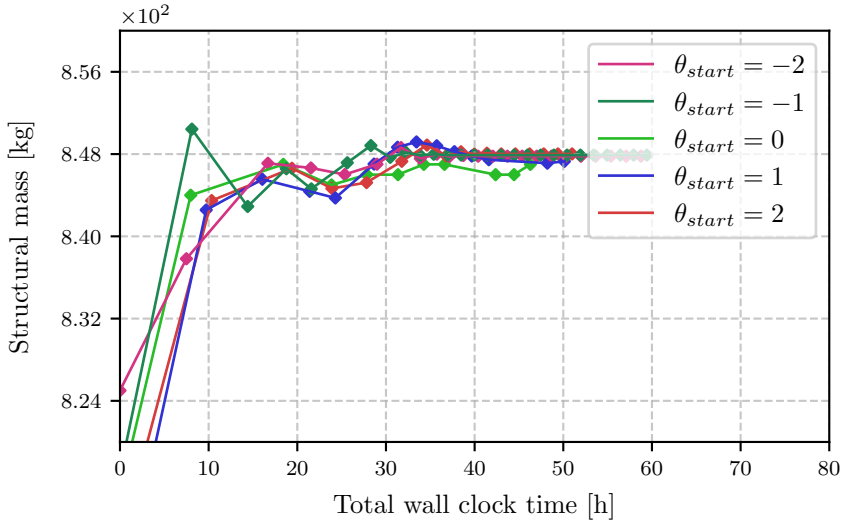
Parameter	$\theta_{start} = -2°$	$\theta_{start} = -1°$	$\theta_{start} = 0°$	$\theta_{start} = 1°$	$\theta_{start} = 2°$
Mission fuel burn m^*_{fuel}	9274 kg	9272 kg	9272 kg	9272 kg	9275 kg
Coefficient of drag C^*_D	$3.238 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$
Finite element mass m_{FE}	848 kg	848 kg	847 kg	847 kg	848 kg



(a) *BL-IRT* optimization with C_{mx}-target objective function of the simple transonic wing with starting point variation



(b) *BL-IRT* optimization with C_{mx}-target aerodynamic objective function of the simple transonic wing with starting point variation



(c) *BL-IRT* optimization with C_{mx}-target structure objective function of the simple transonic wing with starting point variation

Figure 6.12: Starting point variation of *BL-IRT* optimization with C_{mx}-target of the simple transonic wing

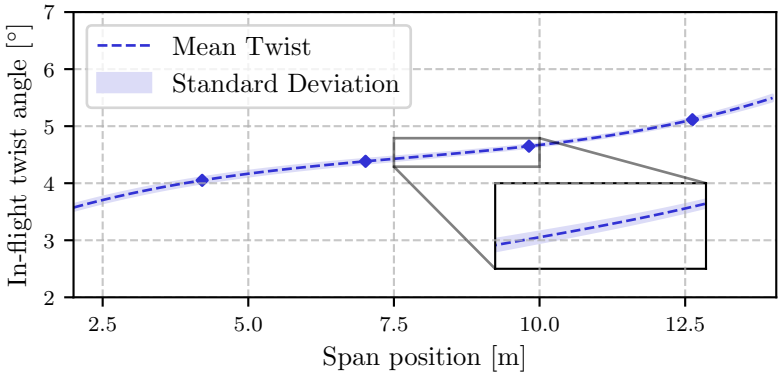


Figure 6.13: Mean and standard deviations of the *BL-IRT* with C_{mx}-target optimized in-flight twists for varied starting points

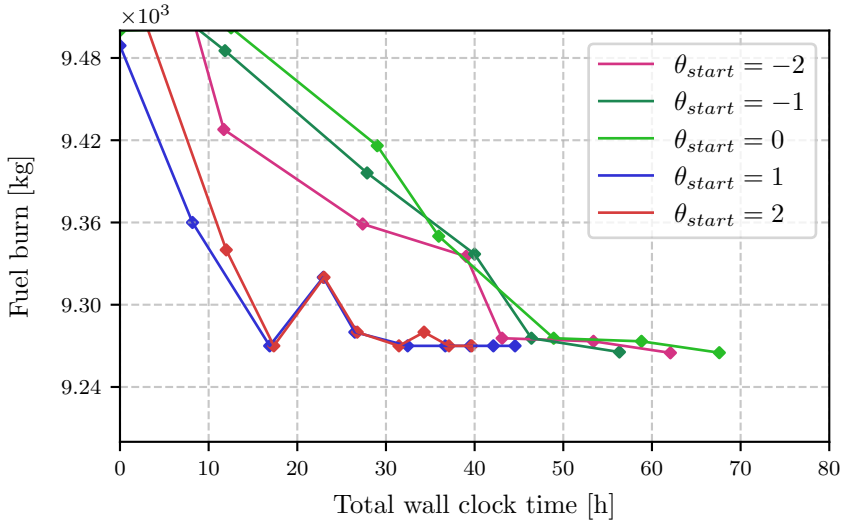
6.3.3 Starting Point Variation of *BL-BCD* formulation

The results for the *BL-BCD* formulation are plotted in Figure 6.14. A notable difference exists between the results of the positively twisted configurations and the negatively twisted results. There is also a notable difference between the fuel consumption obtained after the first system-level iterations, shown in Figure 6.14a. However, since the same C_{mx}-target constraint is applied there, a closer match should be expected, as is the case for the *BL-IRT* formulation with C_{mx}-target in Figure 6.12a. This indicates insufficient convergence of the BCD loops.

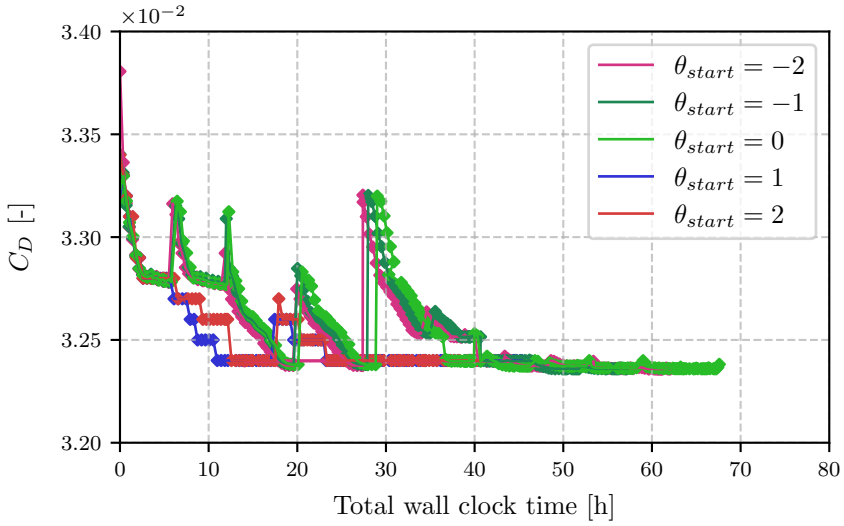
The optimized in-flight twists shown in Figure 6.15 have a similar standard deviation compared to the *BL-IRT* with C_{mx}-target optimized results from Figure 6.13.

Table 6.8: Optimal values of the *BL-BCD* optimization of the simple transonic wing with varied starting points

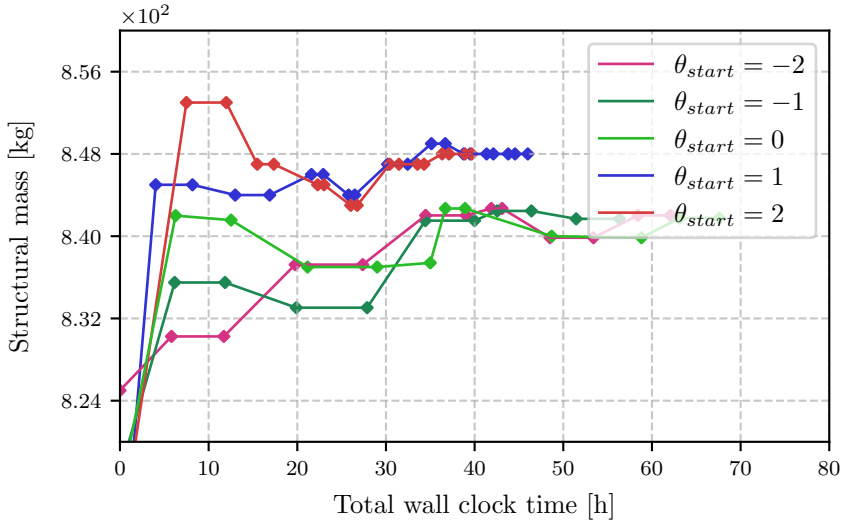
Parameter	$\theta_{start} = -2°$	$\theta_{start} = -1°$	$\theta_{start} = 0°$	$\theta_{start} = 1°$	$\theta_{start} = 2°$
Mission fuel burn m^*_{fuel}	9265 kg	9266 kg	9265 kg	9272 kg	9272 kg
Coefficient of drag C^*_D	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$
Finite element mass m^*_{FE}	842 kg	842 kg	842 kg	848 kg	848 kg



(a) *BL-BCD* optimization objective of the simple transonic wing with starting point variation



(b) *BL-BCD* optimization aerodynamic objective function of the simple transonic wing with starting point variation



(c) *BL-BCD* optimization structure objective function of the simple transonic wing with starting point variation

Figure 6.14: Starting point variation of *BL-BCD* optimization of the simple transonic wing

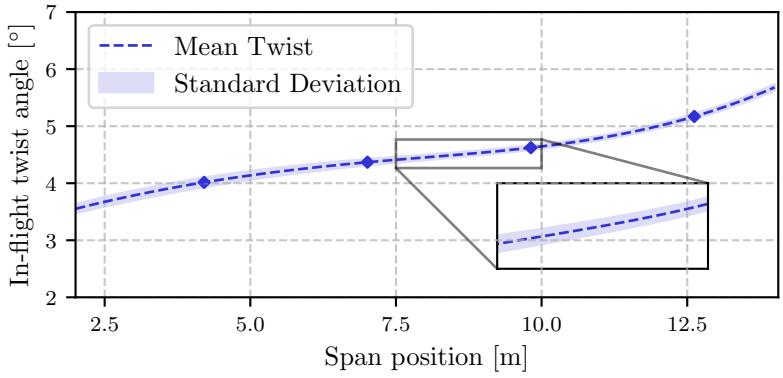


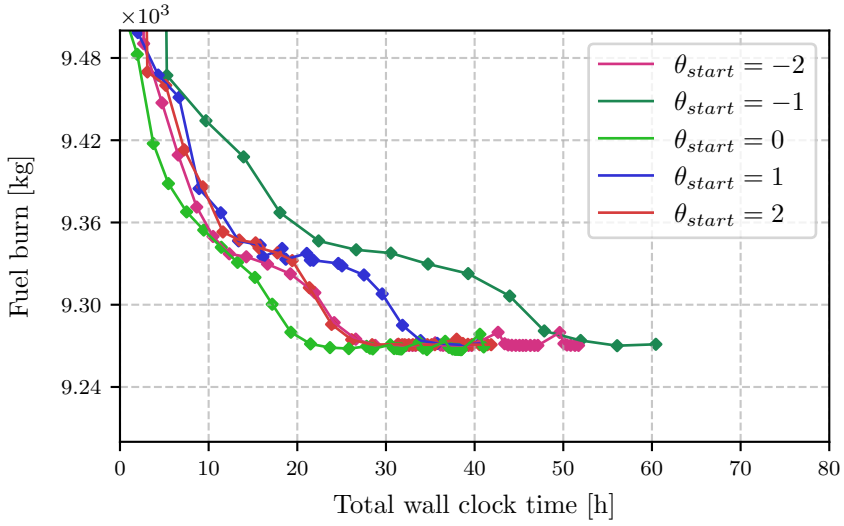
Figure 6.15: Mean and standard deviations of the *BL-BCD* optimized in-flight twists for varied starting points

6.3.4 Starting Point Variation of *BL-EMB* formulation

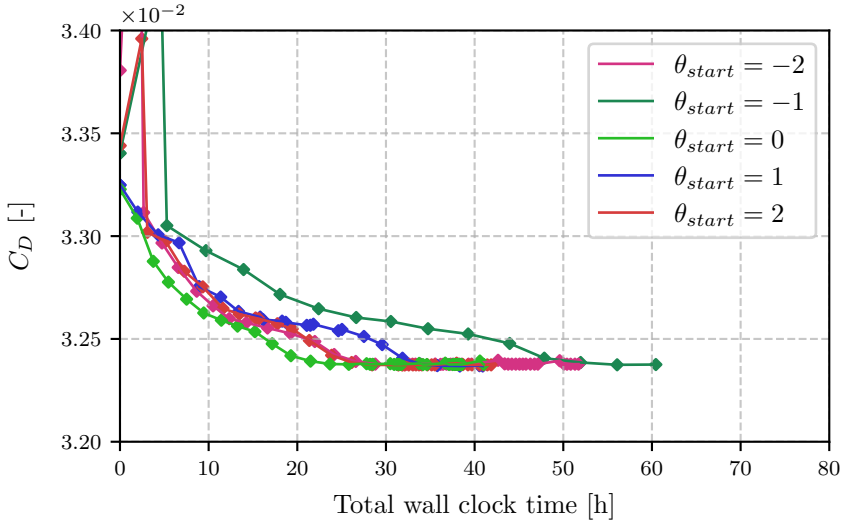
The results for varying the starting point of the *BL-EMB* formulation are shown in Figure 6.16a, with the optimal values listed in Table 6.9. The maximum difference in mission propellant consumption m_F is 2 kg – the smallest spread among all the formulations examined. As shown in Figure 6.17, only a small standard deviation is observed for the optimized in-flight twists.

Table 6.9: Optimal values of the *BL-EMB* optimization of the simple transonic wing with varied starting points

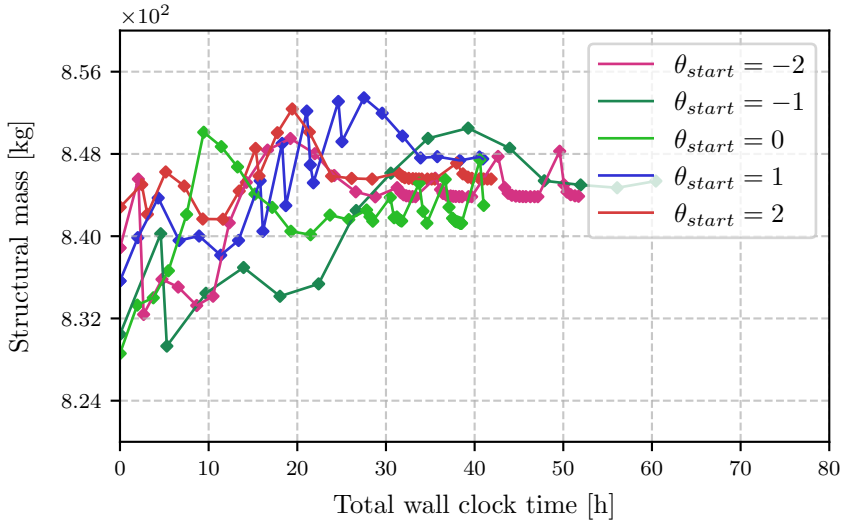
Parameter	$\theta_{start} = -2°$	$\theta_{start} = -1°$	$\theta_{start} = 0°$	$\theta_{start} = 1°$	$\theta_{start} = 2°$
Mission fuel burn m^*_{fuel}	9270 kg	9271 kg	9269 kg	9271 kg	9271 kg
Coefficient of drag C^*_D	$3.238 \cdot 10^{-2}$	$3.238 \cdot 10^{-2}$	$3.238 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$	$3.237 \cdot 10^{-2}$
Finite element mass m^*_{FE}	844 kg	845 kg	843 kg	848 kg	846 kg



(a) *BL-EMB* optimization objective of the simple transonic wing with starting point variation



(b) *BL-EMB* optimization aerodynamic objective function of the simple transonic wing with starting point variation



(c) *BL-EMB* optimization structure objective function of the simple transonic wing with starting point variation

Figure 6.16: Starting point variation of *BL-EMB* optimization of the simple transonic wing

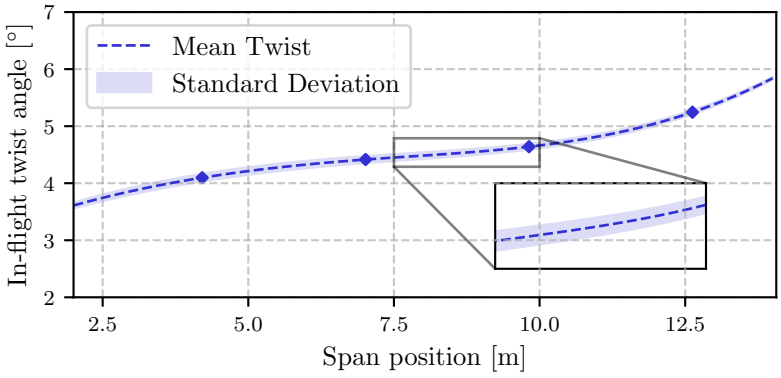


Figure 6.17: Mean and standard deviations of the *BL-EMB* optimized in-flight twists for varied starting points

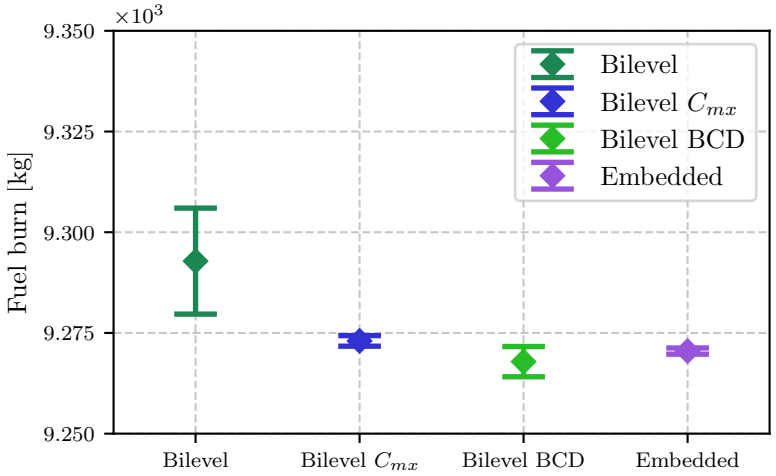
6.3.5 Summary of Results

A summary of these results is visualized in Figure 6.18, which shows the means and standard deviations in the optimized mission fuel burn m_F^*, drag coefficient C_D^*, and finite element mass m_{FE}^*.

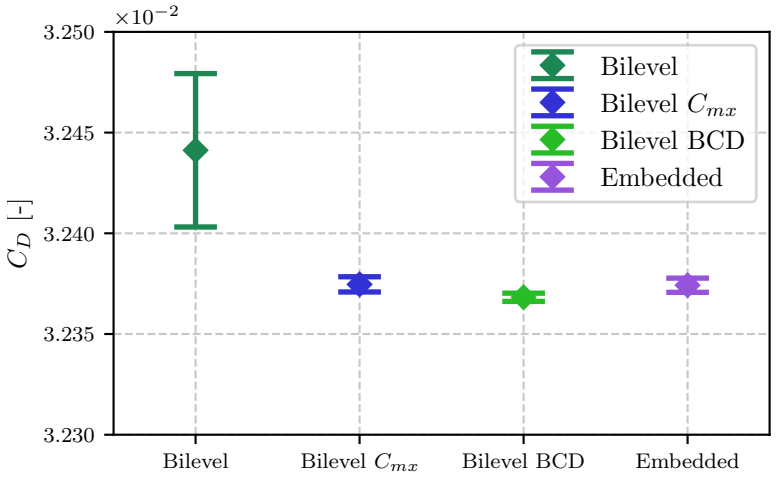
These results indicate a dependence of the optimal solution of the *BL-IRT* formulation on the initial conditions. The use of a C_{mx} target as a system-level design variable reduced this effect, while in this study the *BL-BCD* formulation had a negative effect on the similarity of results. Reasons for this unexpected behavior could be a too low degree of convergence for the subscenarios within the BCD loop. The *BL-EMB* formulation shows the most accurate result in terms of reproducibility of the objective function among all the formulations studied.

Using a C_{mx} target as the system-level design variable also leads to much more deterministic results, significantly reducing the standard deviation of the in-flight twists between the different starting points. All other formulations, except the *BL-IRT* formulation, also show similarly small margins of variability between the optimized twist profiles.

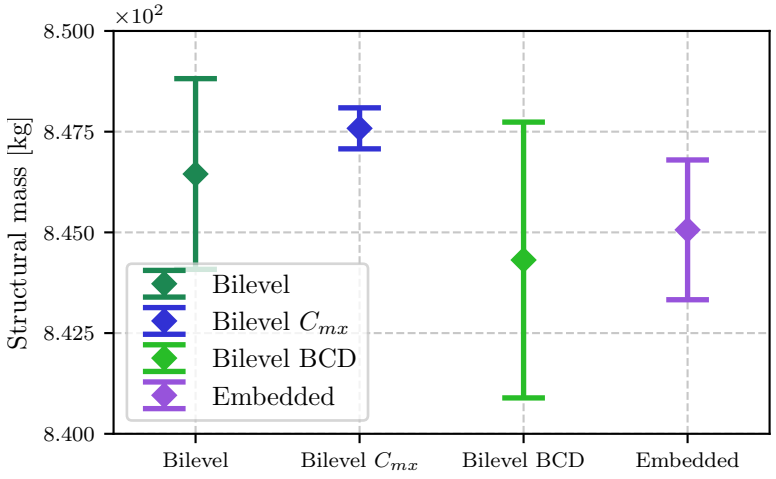
A concluding assumption that can be drawn from this study with limited sample size is that the use of an implicit system-level design variable can help to overcome the starting point dependency of *bilevel* formulations. With the exception of the *BL-IRT* formulation, the results of all formulations are interpreted to be within a reasonable margin of error for large aircraft optimizations. A lack of convergence for the BCD loop in the *BL-BCD* formulation is identified as a potential source of error leading to lower reproducibility of results compared to the *BL-IRT* formulation with C_{mx} target constraint. This issue is the subject of Section 8.4, where a change in the software architecture to ensure proper convergence is discussed.



(a) Standard deviation of the optimized mission fuel burn m_F^* for the tested starting point variation



(b) Standard deviation of the optimized coefficient of drag C_D^* for the tested starting point variation



(c) Standard deviation of the optimized finite element mass m_{FE}^* for the tested starting point variation

Figure 6.18: Mean optimized performance parameters with standard deviations

7 Discussion

The results of the previous chapter show that different *bilevel* formulations are able to achieve similar objectives as the benchmark *MDF* formulation for both the simple transonic wing and *DLR-F25* test cases. The *BL-IRT* formulation shows significant variation in the achieved optimum depending on the starting point of the optimization, which was reduced by applying a C_{mx} target constraint. Common to all *bilevel* formulations in the results were the increased computational times compared to the *MDF* formulation.

Considering these and all further properties obtained from the results in Chapter 6, the research question of this thesis is discussed in Section 7.1, whether *bilevel* formulations can be used for the application of MDO processes in industry. While high-fidelity methods are used in MDO processes, the test cases employed, especially the simple transonic wing, are of relatively small scale in terms of resolution and degrees of freedom in the design. Therefore, in Section 7.2 the scalability of the *bilevel* processes to larger optimization cases is theoretically investigated.

Since only wing profile optimizations are considered in this study, Section 7.3 discusses the application of *bilevel* optimizations to more diverse optimization tasks, especially wing planform and full aircraft configurations.

7.1 Interpretation of Results for Industrial Application

This section discusses the results of Chapter 6 under the research question of whether the considered *bilevel* formulations are an alternative to monolithic formulations like the *MDF* in industry whose organizational structure prohibits fully coupled optimization.

Within Airbus, the organization limits the centralization of the MDO process, so that each discipline must be optimized by its respective department. Therefore, cross-domain

gradient information, such as the influence of structural design on aerodynamic performance parameters, is not shareable between disciplines.

All considered *bilevel* formulations fulfill the condition that no gradients are exchanged between the aerodynamic shape optimization and the structural sizing process. Only the *BL-EMB* formulation requires post-optimality sensitivities of the structural sizing process with respect to the shape design variables. While these gradients were obtained using *finite differences* in this study, it proved to be a major robustness issue for the *BL-EMB* formulation, especially for the *DLR-F25* optimization. In Section 8.3 an analytical method for post-optimality sensitivity calculation is proposed, but whether this method still satisfies the gradient exchange restrictions imposed by the organizational structure has to be evaluated by the industrial partner.

For an MDO method to be adopted in aircraft design, the reliability of the results is critical. The study of starting point variation in Section 6.3 confirms the findings of [30] that the optimization results of the *BL-IRT* formulation are dependent on the starting point of the optimization. Applying a C_{mx} target constraint significantly reduced this effect, but the sample size studied is too small to draw general conclusions.

Contrary to expectations, the *BL-BCD* formulation with C_{mx}-target constraint did not outperform the *BL-IRT* formulation in terms of starting point reliability. This result is unexpected given the results in [30], but is likely due to the lower degree of system-level convergence and the improper convergence of the BCD loop. In general, the *BL-BCD* formulation is expected to be more robust, since the BCD loop reduces the noise to which the top-level optimizer is subjected.

While the author considers the results of the starting point variation study as promising, showing that the *BL-IRT* formulation with C_{mx} target constraint and the *BL-EMB* formulation show variations of less than 0.03% in the objective function, as well as nearly identical designs, even when starting from non-technical starting points. Further studies are however required on the initial condition dependency of the different formulations using industrial testcases in together with reliability requirement definitions from industry are required.

A disadvantage of the *bilevel* algorithms is their high computation time. In the different studies, all *bilevel* formulations required at least twice the time of the *MDF* optimization. While this is a negative attribute of the *bilevel* algorithms, it is to be expected, considering the *No Free Lunch* theorems of Wolpert [47], that a price has to be paid for the gain of reducing the required gradient information. Thus, the author considers the loss of computational efficiency and the dependence of the starting point on non-technically relevant scales as a worthwhile sacrifice to enable the application of MDO in the aerospace

industry. Staying with the implications of the *No Free Lunch Theorem*, that improvements in the performance of an optimization can only be made by increasing its tailoring to the underlying optimization problem, in Chapter 8 new methods are proposed to improve the performance of *bilevel* formulations for coupled aero-structural problems.

7.2 Scalability

While the tools used to perform the experiments on the different *bilevel* formulations use high fidelity methods, the test cases on which they were used are simplified. Therefore, the scalability of these formulations needs to be considered for industrial applications with potentially much larger models and more detailed parameterizations. A further enhancement of the structural sizing process could, for example, feature *CFRP* laminate tailoring, which would significantly increase the number of design variables. Furthermore, the addition of other disciplines such as an engine model or a flight path optimization may be of interest, which would increase the dimensionality of the coupling.

A limiting factor for the *BL-IRT* formulation is the use of a gradient-free optimizer for the common subset of design variables. The required function evaluations scale exponentially for a gradient-free optimization [48, 49]. Increasing the number of shared design variables would result in an exponentially growing number of disciplinary optimizations.

This scaling problem is mitigated by the implicit *BL-IRT* formulation with a C_{mx} target constraint, as this removes all direct design variables from the top-level optimizer and replaces them with the target constraint. If other disciplines are to be added to the *BL-IRT* formulation with target constraints, suitable target constraints must be found that capture the discipline tradeoff. Although the C_{mx} target has shown its effectiveness in separating the coupling, finding such an implicit variable requires a detailed understanding of the coupling and may not be possible for every optimization problem.

Similar considerations apply to the *BL-BCD* formulation, but in addition the number of coupling variables affects the convergence of the *BCD* loop. In [30], the scalability of the *BL-IRT* and *BL-BCD* formulations is tested for a scalable version of the *Sellar* problem. It is found that increasing the number of local and coupling variables does not affect the number of required function evaluations for the *BL-BCD* formulation, but does cause an increase for the *BL-IRT* formulation. However, this increase is still smaller than the increase for a gradient-free *MDF* formulation.

The scaling of the *BL-EMB* formulation is currently limited by the *finite differencing* of

the structural sizing process with respect to the shape design parameters. Even for the test cases used, the gradients are only calculated for the shape design variables that are expected to have the most significant influence on the structure sizing requirements. The cost of adding more shape parameters with derivatives scales linearly with the cost of the full structural sizing process.

The conclusion that can be drawn from this discussion is that especially the implicit formulation of the *BL-IRT* formulation has desirable convergence properties that would allow larger problems to be optimized with a similar number of function evaluations. In contrast, the explicit *BL-IRT* formulation is expected to scale unfavorably with the number of top-level design variables, due to the curse of dimensionality inherent in the gradient-free algorithms used. For the *BL-EMB* formulation, scalability is limited due to the required *finite differencing* of the embedded optimization. The use of post-optimality sensitivities extracted from the *Lagrangian* multipliers is a potential solution to this problem, that is further discussed in Section 8.3

7.3 Transferability

This section discusses the application of the *bilevel* formulations to other optimization problems. Special interest is given to wing planform optimizations, as this may be one of the next extensions of the MDO process.

For the *BL-IRT* formulation, a meaningful decomposition of the optimization design space into local and shared design variables must be found. The design variables with the most significant interdisciplinary effects should be in the shared design space. However, their number is limited by the use of the system-level gradient-free optimizer, as discussed in Section 7.2. When applying the *BL-IRT* formulation to a planform optimization, the number of shared design variables controlled by the system-level gradient-free optimizer increases by at least 4, given a minimal parameterization of span, sweep angle, and root and tip chords. Since the number of design variables already prevents proper convergence in a profile optimization configuration due to run-time constraints, further increasing the number of shared design variables may not be feasible.

Transferring the *BL-IRT* with C_{mx} objective formulation to another optimization problem requires the existence of a suitable implicit design variable to replace C_{mx}. A requirement for this implicit design variable is that the resulting disciplinary optimizations are free of multimodalities, i.e., there are no local minima. Streuber et al. [26] have judged the risk of multimodalities to be low for detailed aerostructure optimizations, with an increasing risk

for preliminary design problems. Therefore, extending the *BL-IRT* process to a planform optimization introduces the risk of creating a multimodal shape optimization, which requires further system-level specification of the disciplinary problem. The addition of a C_{my} target constraint for the wing could address this challenge, if it is indeed manifested, by adding further control over the longitudinal position of the center of lift. The same considerations apply to the *BL-BCD* formulation.

Applying the *BL-EMB* formulation to other optimization problems may be of interest if there is a significant difference in the optimization times of the disciplines involved. An extension to wing planform optimization would still meet this criterion, since the computational complexity of aerodynamic planform optimization is higher than that of structural design. In the current implementation, which uses *finite differences* to obtain the gradients of the finite element mass m_{FE} with respect to the most important shape design variables, the effort to obtain these gradients increases linearly with the number of added design variables, as previously discussed in Section 7.2.

8 Further Developments

This chapter discusses potential extensions to the existing optimization framework to improve various performance aspects of *bilevel* aerostructural optimization processes on industrial configurations. All proposed ideas aim at tailoring the aerostructural optimization process specifically to the *bilevel* processes by introducing new methods. In Section 8.1, the idea is to replace the gradient-free optimizers currently used in the *BL-IRT* and *BL-BCD* formulations with *Bayesian* optimization algorithms to reduce the required number of system-level optimization iterations.

In Section 8.2, a new objective function is proposed for the structural design process to account for changes in lift-induced drag due to passive load alleviation measures.

In order to improve the computational time for the *BL-EMB*, an approach for the computation of post-optimality derivatives of the structural sizing process is proposed, replacing the currently used *finite differences*.

The lack of adequate convergence of the BCD loop indicated by the results in Section 6.3 requires an adaptation of the underlying processes, namely the exposure of the structural design variables to *GEMSEO*, discussed in Section 8.4.

8.1 Bayesian Optimization

As shown in Chapter 6, the computational complexity of both the *BL-IRT* and *BL-BCD* methods is significantly higher than that of the *MDF* formulation for a given problem. This discrepancy can be attributed to the cost of each iteration of the gradient-free system-level optimizer, which involves optimization of both the aerodynamic shape and the structural design. Consequently, reducing the number of system-level optimization iterations could significantly reduce the required computational time. *Bayesian* optimization uses infor-

mation from all previous iterations by training a *GP* with the objective and constraint function data. In contrast, other gradient-free optimizers, such as *COBYLA*, use only the information from the current simplex directly. The use of this additional information by the optimization algorithm can lead to a significant improvement in its performance. In addition, *Gaussian process* (*GP*) algorithms have demonstrated their ability to handle noisy data, as shown by McHutchon et al. [50], in both the input and output domains, a property that extends to the suboptimization results of *bilevel* processes.

As shown by Eriksson et al. [51], the *scalable constrained bayesian optimization* (*SCBO*) algorithm has been shown to outperform numerous gradient-free optimizers, including the *COBYLA* algorithm used in this work. The *SCBO* algorithm is an extension of the *Bayesian* optimization concept, where a *GP* is trained on sampled data points $[\mathbf{x}_i, f(\mathbf{x}_i)]$ and predicts new samples that are candidates for the optimal value. The *SCBO* algorithm extends the *Bayesian* optimization framework by using a trust region method to define candidate points in conjunction with the predictions generated by the *GP*. To improve the quality of the data prior to fitting the *GP*, *bilog* transformations are applied to the constraints and a *Gaussian cupola* transformation is applied to the objective functions.

The use of these transformations can be particularly beneficial in scenarios where the test case lacks a clearly defined optimum, as shown in Section 6.1. In this section, comparable fuel consumption values were achieved by the *MDF* and *bilevel* formulations, even though these formulations are composed of significantly dissimilar performance parameters.

A version of the *SCBO* for unconstrained is also proposed in [51], called *trust region Bayesian optimization* (*TuRBO*). Replacing the system-level gradient-free optimizer in the *BL-IRT* and *BL-BCD* formulations with either the *SCBO* or *TuRBO* method could reduce the overall computational time required by reducing the number of system-level optimization iterations. Subsequent improvements in computational efficiency can be achieved by using the optimizer's batch capabilities to run multiple system-level optimization iterations simultaneously.

8.2 Unified Structural Sizing Objective Function

During the optimization process it was noticed that the structural sizing process, especially in the first *bilevel* iterations, applies aggressive passive load alleviation measures. While this is beneficial for reducing the aerodynamic loads on the wing and thus the structural mass, it degrades the aerodynamic properties of the wing by shifting the load inboard. This effect could be mitigated by using the global objective function as the objective

function for the structural sizing process. The coefficient of lift (C_L) and coefficient of drag (C_D) would have to be provided by the loads model of the structural sizing process for cruise flight. Since the low-fidelity loads model in *LAGRANGE* is based on *potential theory*, it cannot provide estimates for wave and pressure drag. These estimates would have to be provided by the previous aerodynamic optimization cycle, assuming that structural changes do not significantly affect these terms.

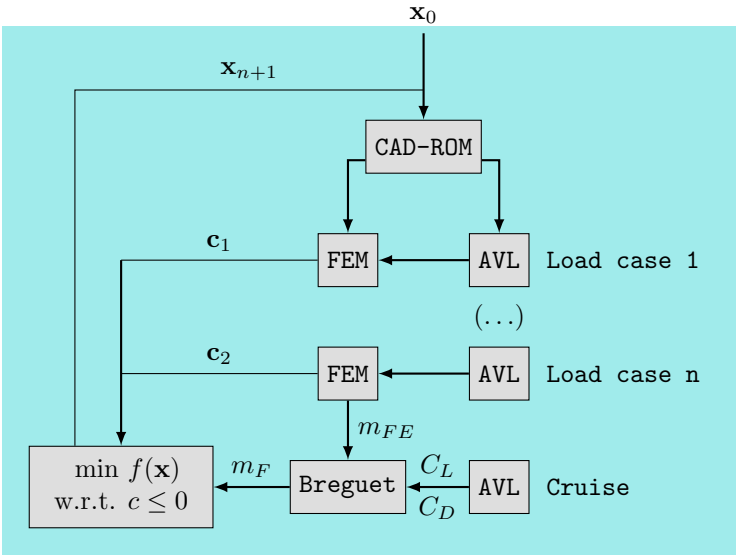


Figure 8.1: Draft of adapted structure sizing process

8.3 Analytical Post-Optimality Gradients for *BL-EMB* Formulation

In the current implementation of the *BL-EMB* formulation, the finite element mass m_{FE} is differentiated using one-sided, first-order *finite differences* to obtain the sensitivities with respect to the most influential shape design variables. This approach is computationally expensive and the computational effort scales linearly with the number of design variables.

An analytical post-optimality sensitivity calculation, as proposed by Braun et al [33], can reduce the computational effort to obtain the gradients of the structural sizing process. Post-optimality sensitivities describe the change in the objective function with respect to a previously fixed parameter, such as the shape parameters in the structural sizing process. Using the method of Braun [33], the sensitivity of the optimal value of the objective

function f^* with respect to a design parameter x_i is

$$\frac{df^*}{dx_i} = \frac{\partial f}{\partial x_i} + \boldsymbol{\lambda}^{*T} \frac{\partial \mathbf{c}}{\partial x_i}. \tag{8.1}$$

However, this method requires the provision of the *Lagrangian*, which is not provided by the *LAGRANGE* internal *NLPQL* optimizer. Recalling the *KKT* conditions for a quadratic problem as Equation (2.81), the optimal *Lagrangian* multipliers for the equality constraints $\boldsymbol{\lambda}_h^*$ and the working set of inequality constraints $\boldsymbol{\lambda}_h^*$ are defined by the conditions:

$$\mathbf{A}\boldsymbol{\lambda}_h^* = \nabla f \tag{8.2}$$

$$\mathbf{C}_w\boldsymbol{\lambda}_g^* = \nabla f. \tag{8.3}$$

Both matrices $\mathbf{A}$ and $\mathbf{C}_w$ are not necessarily square, so the system of equations has to be extended to the left by their transposed values.

$$(\mathbf{A}\mathbf{A}^T)\boldsymbol{\lambda}_h^* = \mathbf{A}^T\nabla f \tag{8.4}$$

$$(\mathbf{C}_w\mathbf{C}_w^T)\boldsymbol{\lambda}_g^* = \mathbf{C}_w^T\nabla f \tag{8.5}$$

Ideally these equation systems should be solved with a *TQ-factorization* or *QR-decomposition* to reduce error scaling, as the condition number of $\mathbf{A}\mathbf{A}^T$ is the square of the condition number of $\mathbf{A}$ [33].

8.4 Exposure of the Structural Sizing Process to *GEMSEO*

In the current software architecture of the *GEMSEO* processes, the *LAGRANGE*-based structural sizing process is implemented as a "black box", with no exposure to the *GEMSEO* integration layer of structural design variables and structural constraints.This section therefore discusses the potential implementation of the structural sizing process that would be exposed to *GEMSEO*. This initiative stems from the observed improper convergence of the BCD loops of the *BL-BCD* formulation, as detailed in Section 6.3. The cause of this convergence problem is the current invisibility of the structural design variables to *GEMSEO*. Consequently, they are not taken into account in the convergence criteria of the *Gauss-Seidel* algorithm. As a result, the algorithm can be terminated without ensuring the consistency of the structural design variables between the aerodynamic performance prediction and the structural design processes. Therefore, it is imperative to expose the structural design variables to ensure the effective execution of the *BL-BCD* processes.

While the implementation of *GEMSEO*, which provides optimization resources, would also allow the use of other *GEMSEO* methods, such as post-processing, certain key features are currently not available in *GEMSEO*, which hinders this development. A challenging feature of the structural sizing process is the large number of design variables and constraints. To address this challenge, *LAGRANGE* uses callbacks from the *NLPQL* optimizer to selectively compute the sensitivities of constraints within the *working set* of the optimizer. This approach reduces the computational complexity. It is noteworthy that such a method, as well as an optimizer that provides callbacks regarding which constraints require derivation, has not yet been implemented in *GEMSEO*. Consequently, the implementation of such a method would result in a significant increase in the computational time required for structural optimization.

An intermediate solution, providing the structural design variables as outputs of the structural sizing process implementation in *GEMSEO*, which are read as inputs by the aerodynamic performance prediction process, is therefore considered favorable for the currently used versions of *GEMSEO* and *LAGRANGE*.

9 Conclusion

In this study, the applicability of different *bilevel* formulations to high-fidelity MDO processes was investigated, as the aerospace industry demands MDO processes that can be implemented within their disciplinary organization structures. The MDO framework *GEMSEO* was utilized to embed the existing MDO process and facilitate changing the formulations of the process.

The *DLR-F25* and the simple transonic wing (University of Michigan) were selected as test cases for the study. Across both test cases, the different *bilevel* optimization formulations were found to achieve objective function values that were similar to the benchmark *Multidisciplinary-Feasible* (*MDF*) formulation. However, the *bilevel* formulations tended to result in higher structural mass and lower aerodynamic drag. For the simple transonic wing, the *bilevel-embedded* (*BL-EMB*) formulation, which embeds the structural sizing process within the aerodynamic shape optimization, and the *bilevel-block-coordinate-descent* (*BL-BCD*) formulation, which iterates a *Gauss-Seidel* algorithm for structural sizing and aerodynamic shape optimization, achieve optimal results that surpass the MDF optimum. It is noteworthy that the computational demands associated with *bilevel* optimizations are often substantially higher than those required for the *MDF* formulation. Specifically, the *BL-BCD* formulation required four times the computational time of the *MDF* formulation for optimizing the simple transonic wing.

In earlier research on *bilevel* formulations employing analytical test cases, the impact of the optimization starting point on the resulting optimized design was identified. In this study, the aforementioned research was expanded upon by applying various *bilevel* optimizations to the simple transonic wing, with modified twist profiles and mass optimized structure. The impact of the starting point on the final design was observed for the *bilevel* formulation developed by the IRT Saint Exupery (*BL-IRT*), which utilizes section twists as system-level design variables. By employing an implicit version of the *BL-IRT* formulation with a C_{mx}-target as the system-level design variable, which is enforced by the aerodynamic shape optimization, this effect was found to be considerably mitigated

to below technical scales. The *BL-EMB* formulation exhibited the least influence of the starting point among all the investigated formulations.

The optimization framework *GEMSEO* was found to be beneficial in studying the application of different MDO formulations using complex optimization toolchains, especially in terms of separating processes and formulations. The current depth of process exposure to *GEMSEO* is considered optimal in terms of computational efficiency. However, in order to improve the convergence of the *BL-BCD* formulation for tightly coupled aerostructural problems, it is necessary to expose the structural design variables to *GEMSEO*. Suggested enhancements for more efficient and larger scale processes include the use of a *Bayesian* optimizer to control system-level design variables and an analytical process to obtain post-optimality sensitivities of the structural sizing processes.

The findings from this thesis underpin the potential of *bilevel* formulations for application of MDO in the aerospace industry. Throughout the study, these formulations produced results comparable to the results obtained when using an *MDF* formulation.

Despite the intention to use this comparison to evaluate the performance of the *bilevel* formulations, it must be acknowledged that the problem is inherently ill-posed. This is due to the fact that applying MDF optimizations within aerospace companies that are characterized by a long-standing organizational structure is not a feasible proposition from the outset. At this time, when MDO methods play no role in the design processes of commercial aircraft, the only fair comparison is between the *bilevel* optimized results and those obtained without MDO, i.e. designs. If this comparison were to be carried out, it would highlight the fact that bilevel MDO methods improve the mission fuel burn calculated for conventionally designed *HARW* aircraft. Specifically, the DLR-F25 demonstrated a 4% improvement through pure optimization without introducing additional technical complexities to the aircraft. If the fuel consumption of the worldwide fleet of aircraft were to be reduced by 4%, the estimated savings in emissions would amount to 35 million tons per year.

This underscores the efficacy of *bilevel* methods in aero-structural optimization and their indispensable role in achieving the ambitious targets established to reduce global aviation emissions.

Bibliography

[1] ICAO. *States adopt net-zero 2050 global aspirational goal for international flight operations*. 13.12.2024. URL: https://www.icao.int/Newsroom/Pages/States-adopts-netzero-2050-aspirational-goal-for-international-flight-operations.aspx.

[2] Madelynn. J. Watson et al. "Sustainable aviation fuel technologies, costs, emissions, policies, and markets: A critical review". In: *Journal of Cleaner Production* 449 (2024), p. 141472. ISSN: 09596526. DOI: 10.1016/j.jclepro.2024.141472. URL: https://www.sciencedirect.com/science/article/abs/pii/S095965262400920X.

[3] Raphael T. Haftka. "Optimization of flexible wing structures subject to strength and induced drag constraints". In: *AIAA Journal* 15.8 (1977), pp. 1101–1106. ISSN: 0001-1452. DOI: 10.2514/3.7400.

[4] Bernard Grossman et al. "Integrated aerodynamic/structural design of a sailplane wing". In: *Journal of Aircraft* 25.9 (1988), pp. 855–860. ISSN: 0021-8669. DOI: 10.2514/3.45670.

[5] Bernard Grossman et al. "Integrated aerodynamic-structural design of a transport wing". In: *Journal of Aircraft* 27.12 (1990), pp. 1050–1056. ISSN: 0021-8669. DOI: 10.2514/3.45980.

[6] Hans Eschenauer, Gerd. Schuhmacher, and Johann Krammer. "Constructive design models for multidisciplinary optimization of fiber composite structures". In: *4th Symposium on Multidisciplinary Analysis and Optimization*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 1992. DOI: 10.2514/6.1992-4684.

[7] Gerd Schuhmacher et al. "Multidisciplinary airframe design optimisation". In: (2012), pp. 44–56.

[8] Fabian Volle et al. "Simultaneous Shape and Sizing Optimization for Improving Aircraft Performance Using the Adjoint Approach". In: *DLR eLib* (2023). URL: https://elib.dlr.de/199141/.

[9] Scott Delbecq, Marc Budinger, and Aurélien Reysset. “Benchmarking of monolithic MDO formulations and derivative computation techniques using OpenMDAO”. In: *Structural and Multidisciplinary Optimization* 62.2 (2020), pp. 645–666. ISSN: 1615-147X. DOI: 10.1007/s00158-020-02521-7.

[10] Eytan J. Adler and Joaquim R. R. A. Martins. “Efficient Aerostructural Wing Optimization Considering Mission Analysis”. In: *Journal of Aircraft* 60.3 (2023), pp. 800–816. ISSN: 0021-8669. DOI: 10.2514/1.c037096.

[11] Nathan P. Tedford and Joaquim R. R. A. Martins. “Benchmarking multidisciplinary design optimization algorithms”. In: *Optimization and Engineering* 11.1 (2010), pp. 159–183. ISSN: 1389-4420. DOI: 10.1007/s11081-009-9082-6.

[12] Martin Spieck et al. “MDO as a SOCIO-technical system–a small contribution to collaborative engineering”. In.

[13] Anne Gazaix et al. “Industrial Application of an Advanced Bi-level MDO Formulation to Aircraft Engine Pylon Optimization”. In: *AIAA Aviation 2019 Forum*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2019. ISBN: 978-1-62410-589-0. DOI: 10.2514/6.2019-3109.

[14] Mohammad Abu-Zurayk et al. “Comparing Two Multidisciplinary Optimization Formulations of Trimmed Aircraft Subject to Industry-relevant Loads and Constraints”. In: *AIAA AVIATION 2021 FORUM*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2021. ISBN: 978-1-62410-610-1. DOI: 10.2514/6.2021-3075.

[15] Francois Gallard et al. “GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation”. In: American Institute of Aeronautics and Astronautics, 2018. DOI: 10.2514/6.2018-0657.

[16] John D. Anderson. *Fundamentals of aerodynamics*. Sixth edition. McGraw-Hill series in aeronautical and aerospace engineering. New York, NY: McGraw-Hill Education, 2017. ISBN: 978-1259129919.

[17] Jan R. Wright and Jonathan E. Cooper. *Introduction to aircraft aeroelasticity and loads*. Vol. 18. John Wiley & Sons, 2008. ISBN: 9781118700440.

[18] Phillipp Spalart and Steven Allmaras. “A one-equation turbulence model for aerodynamic flows”. In: *30th Aerospace Sciences Meeting and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 1992. DOI: 10.2514/6.1992-439.

[19] Hrvoje Jasak. “Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows”. In: *Open Journal of Fluid Dynamics* (1996).

[20] Martin Bäker. *Numerische Methoden in der Materialwissenschaft*. Springer, 2024. ISBN: ISBN 3-936148-08-2. URL: https://link.springer.com/book/10.1007/978-3-658-45261-2.

[21] Andreas Schuster, Lars Reimer, and Jens Neumann. "A Mesh-Free Parallel Moving Least-Squares-based Interpolation Method for the Application in Aeroelastic Simulations With the Flow Simulator". In: *New Results in Numerical and Experimental Fluid Mechanics X*. Ed. by Andreas Dillmann et al. Vol. 132. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Cham: Springer International Publishing, 2016, pp. 573–583. ISBN: 978-3-319-27278-8. DOI: 10.1007/978-3-319-27279-550.

[22] Richard P. Dwight and Joël Brézillon. "Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization". In: *AIAA Journal* 44.12 (2006), pp. 3022–3031. ISSN: 0001-1452. DOI: 10.2514/1.21744.

[23] Richard P. Dwight. "Robust Mesh Deformation using the Linear Elasticity Equations". In: *Computational Fluid Dynamics 2006*. Ed. by Herman Deconinck. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 401–406. ISBN: 978-3-540-92778-5. DOI: 10.1007/978-3-540-92779-2-62.

[24] Joaquim R. R. A. Martins and Andrew Ning. *Engineering design optimization*. Cambridge: Cambridge University Press, 2022. ISBN: 9781108833417. URL: https://zbmath.org/1480.74002.

[25] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Second edition. Springer series in operation research and financial engineering. New York, NY: Springer, 2006. ISBN: 978-0-387-40065-5. URL: http://www.loc.gov/catdir/enhancements/fy0818/2006923897-d.html.

[26] Gregg M. Streuber and David. W. Zingg. "Evaluating the Risk of Local Optima in Aerodynamic Shape Optimization". In: *AIAA Journal* 59.1 (2021), pp. 75–87. ISSN: 0001-1452. DOI: 10.2514/1.J059826.

[27] Dieter Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. URL: https://books.google.de/books?id=4rKaGwAACAAJ.

[28] Klaus Schittkowski. "NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search - User's Guide, Version 3.0". In: (2006).

[29] Michael J. D. Powell. "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. Ed. by Susana Gomez and Jean-Pierre Hennart.

Dordrecht: Springer Netherlands, 1994, pp. 51–67. ISBN: 978-90-481-4358-0. DOI: 10.1007/978-94-015-8330-5-4.

[30] Yann David. "Architectures bi-niveau et algorithmes multi-fidélité pour l'optimisation multidisciplinaire en haute dimension." Dissertation. Toulouse: Université de Toulouse, 2024.

[31] Joaquim R. R. A. Martins and Andrew B. Lambe. "Multidisciplinary Design Optimization: A Survey of Architectures". In: *AIAA Journal* 51.9 (2013), pp. 2049–2075. ISSN: 0001-1452. DOI: 10.2514/1.J051895.

[32] Ian R. Chittick and Joaquim R. R. A. Martins. "Aero-structural optimization using adjoint coupled post-optimality sensitivities". In: *Structural and Multidisciplinary Optimization* 36.1 (2008), pp. 59–70. ISSN: 1615-147X. DOI: 10.1007/s00158-007-0200-9.

[33] Robert Braun, Ilan Kroo, and Peter Gage. "Post-optimality analysis in aerospace vehicle design". In: *Aircraft Design, Systems, and Operations Meeting*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 1993. DOI: 10.2514/6.1993-3932.

[34] Dieter Schwamborn et al. "Development of the TAU-Code for aerospace applications". In: *50th NAL International Conference on Aerospace Science and Technology*. 2008. URL: https://elib.dlr.de/55519/.

[35] Andrei Merle et al. "Adjoint high-dimensional aircraft shape optimization using a CAD-ROM parameterization". In: *CEAS Aeronautical Journal* 14.3 (2023), pp. 729–738. ISSN: 1869-5582. DOI: 10.1007/s13272-023-00660-3.

[36] Philipp Bekemeyer et al. "Data-Driven Aerodynamic Modeling Using the DLR SMARTy Toolbox". In: *AIAA AVIATION 2022 Forum*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2022. ISBN: 978-1-62410-635-4. DOI: 10.2514/6.2022-3899.

[37] Michael Meinel and Gunnar Einarsson. "The FlowSimulator framework for massively parallel CFD applications". In: *DLR eLib*. URL: https://elib.dlr.de/67768/1/para10-paper-44.pdf.

[38] Arne Rempke. "Netzdeformation mit Elastizitätsanalogie in multidisziplinärer FlowSimulator-Umgebung". In: *STAB Symposium* (2016). URL: https://elib.dlr.de/109263/.

[39] antares Development Team. *Antares Documentation Release 2.3.0*. 2024. URL: https://cerfacs.fr/antares/.

[40] Ulrich Küttler and Wolfgang A. Wall. "Fixed-point fluid–structure interaction solvers with dynamic relaxation". In: *Computational Mechanics* 43.1 (2008), pp. 61–72. ISSN: 0178-7675. DOI: 10.1007/s00466-008-0255-5.

[41] Andrei Merle, Arthur Stueck, and Arne Rempke. “An Adjoint-based Aerodynamic Shape Optimization Strategy for Trimmed Aircraft with Active Engines”. In: *35th AIAA Applied Aerodynamics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2017. ISBN: 978-1-62410-501-2. DOI: 10.2514/6.2017-3754.

[42] Alessando A. Gastaldi et al. *Simultaneous Airframe Shape and Sizing Optimization using Simultaneous Airframe Shape and Sizing Optimization using FlowSimulator and the Lagrange MDO suite suite*. Dresden, 2022.

[43] Alasdair C. Gray and Joaquim R. Martins. “A Proposed Benchmark Model for Practical Aeroelastic Optimization of Aircraft Wings”. In: *AIAA SCITECH 2024 Forum*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2024. ISBN: 978-1-62410-711-5. DOI: 10.2514/6.2024-2775.

[44] Brenda Kulfan and John Bussoletti. “"Fundamental" Parameteric Geometry Representations for Aircraft Component Shapes”. In: *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2006. ISBN: 978-1-62410-020-8. DOI: 10.2514/6.2006-6948.

[45] Jan Mariens, Ali. Elham, and Michael. J. L. van Tooren. “Influence of weight modelling on the outcome of wing design using multidisciplinary design optimisation techniques”. In: *The Aeronautical Journal* 117.1195 (2013), pp. 871–895. ISSN: 0001-9240. DOI: 10.1017/S0001924000008563.

[46] Joël Brézillon. *Deliverable D1.2.1-1 (Key): Use Case Definition for MDO*. Ed. by Clean Aviation. 2023.

[47] David H. Wolpert and William G. Macready. “No Free Lunch Theorems for Optimization”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82.

[48] Nikolaus Hansen et al. “Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009”. In: *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*. Ed. by Martin Pelikan and Jürgen Branke. New York, NY, USA: ACM, 2010, pp. 1689–1696. ISBN: 9781450300735. DOI: 10.1145/1830761.1830790.

[49] Ole Sigmund. “On the usefulness of non-gradient approaches in topology optimization”. In: *Structural and Multidisciplinary Optimization* 43.5 (2011), pp. 589–596. ISSN: 1615-147X. DOI: 10.1007/s00158-011-0638-7.

[50] Andrew Mchutchon and Carl Rasmussen. "Gaussian Process Training with Input Noise". In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc, 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/a8e864d04c95572d1aece099af852d0a-Paper.pdf.

[51] David Eriksson and Matthias Poloczek. *Scalable Constrained Bayesian Optimization*. 2020. DOI: 10.48550/arxiv.2002.08526.