



Aerodynamic Shape Optimization of UAV Fuselage and Fairings with Payload Integration

Luís da Natividade Pires Dias Pinheiro

Thesis to obtain the Master of Science Degree in
Aerospace Engineering

Supervisors: Prof. André Calado Marta
Eng. Nuno Miguel Bento de Matos

Examination Committee

Chairperson: Prof. Afzal Suleman
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Frederico José Prata Rente Reis Afonso

July 2025

Dedicated to my parents.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would first take this opportunity to express my sincere appreciation to my supervisors, Prof. André Calado Marta at Instituto Superior Técnico and Eng. Nuno Matos at TEKEVER, whose support, guidance, and hard work during this journey have made this achievement possible. Their advice and motivation played a key role in the completion of this work.

I also gratefully acknowledge the privilege of working with TEKEVER to execute this thesis in the industrial sector. The support, equipment, and trust given to me by the company played a critical role in facilitating and obtaining the final results. Special appreciation is extended to the Research and Development department at TEKEVER with whom I worked and who assisted me at various times.

To my parents, my sister and my grandparents, I want to express my sincere appreciation for your incomparable belief in me and my ability, and the unconditional support you provided in letting me chase my dreams. I could not have achieved this degree in Aerospace Engineering but for your sacrifices, patience and motivation. I would also want to extend my appreciation to a great friend of mine who became an important figure in my life within the last few years and assisted me through the final leg of this journey.

Lastly, I offer my appreciation to my entire family and friends, whose support, advice, and concern marked the duration of the entire process.

Resumo

A utilização de ferramentas de otimização avançadas está a tornar-se comum na resolução de problemas de engenharia, devido à crescente disponibilidade de recursos computacionais e à maturidade dos algoritmos numéricos. Neste trabalho, é utilizado um enquadramento de otimização de forma aerodinâmica de alta fidelidade baseado no método adjunto. O objetivo é manipular a forma de uma fuselagem genérica para minimizar a resistência aerodinâmica total, respeitando restrições de volume associadas à instalação de componentes internos.

A forma exterior da fuselagem é modificada através da técnica de deformação de forma livre (FFD), permitindo maior flexibilidade no processo de otimização. São estudadas diferentes abordagens de deformação aplicadas aos pontos de controlo da caixa FFD, incluindo deslocamentos na direção normal, nos eixos transversais e alterações de curvatura ao longo do eixo longitudinal.

Demonstra-se que a combinação destas deformações pode resultar numa redução de resistência aerodinâmica superior a 40% face à geometria de referência, mantendo as restrições de volume.

Para validar a abordagem, foram definidos dois casos inspirados em projetos da TEKEVER. O primeiro simula a integração de uma antena SATCOM num volume esférico, resultando numa redução de 4.6% na resistência aerodinâmica total. O segundo, com uma caixa inclinada representando um sistema de paraquedas, levou a um aumento de 25% no arrasto, devido a limitações geométricas e à expansão abrupta da fuselagem.

Palavras-chave: projeto de aeronaves, Otimização baseada em gradientes, Método adjunto, deformação de forma livre, Análise de alta fidelidade, Desempenho aerodinâmico

Abstract

The usage of powerful optimization tools is becoming common in solving many engineering problems due to available computational resources and mature numerical algorithms. In this work, an adjoint-based high-fidelity aerodynamic shape optimization framework is used to manipulate a generic aircraft fuselage shape to minimize the total aerodynamic drag with specific payload volume constraints. The external fuselage shape is modified using the free-form deformation (FFD) technique to allow greater flexibility. The impact of using different deformation approaches applied to the displacement of FFD control points is studied, including displacements along the normal direction, the transverse axis directions, and cambering along the fuselage longitudinal direction. It is demonstrated that the combination of flexible control point displacement in multiple FFD box cross-section planes together with a streamwise cambering can produce a very significant drag reduction, in excess of 40% compared to the selected baseline shape, while still satisfying volume constraints to account for internal or protruding payloads. Additionally, to complement the study, two real case scenarios inspired by TEKEVER UAV projects are tackled, representing realistic integration challenges within UAV fuselages. The first case handles the integration of a SATCOM antenna, represented by its enclosing spherical volume, while the second case deals with the integration of a recovery parachute system, represented by an inclined box. The final optimization cases highlighted the potential for aerodynamic improvement, with Case 1 achieving a 4.6% drag reduction driven by lower pressure drag, while the last case demonstrated a 25% drag increase due to stricter constraints and abrupt fuselage expansion.

Keywords: aircraft design, gradient-based optimization, adjoint method, free form deformation, high-fidelity analysis, aerodynamic performance

Contents

Acknowledgments	vii
Resumo	ix
Abstract	xi
List of Tables	xv
List of Figures	xvii
Nomenclature	xix
Glossary	xxiii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Objectives and Deliverables	3
1.3 Thesis Outline	4
2 Aircraft Analysis and Design Methodologies	5
2.1 Fuselage and Fairings Design	5
2.2 Aerodynamic Forces	7
2.3 Computational Fluid Dynamics (CFD)	12
2.4 Optimization Methods	15
2.4.1 Gradient-Based Methods	16
2.4.2 Adjoint Method	17
3 Aerodynamic Shape Optimization Framework	19
3.1 Overview	19
3.2 Meshing Process	20
3.2.1 Surface Mesh	21
3.2.2 Volume Mesh	21
3.2.3 Overset Mesh	23
4 Fuselage Shape Deformation Modeling	25
4.1 Parametrisation Method Overview	25
4.2 Fuselage Deformation using FFD	26
4.3 Deformation using Selected Points	28

4.4	Cambered Fuselages	29
4.5	Fuselage Constraints	31
5	Study Cases Using Different Parametrization Approaches	37
5.1	Selected FFD Points	38
5.2	Camber Function	39
5.3	Combined Deformation Strategies	41
5.4	Volume Constraints	41
5.5	Discussion of Results	43
6	Fuselage Shape Optimization on Real Case Scenarios	45
6.1	Problem Definition	46
6.2	Mesh Refinement	47
6.3	Baseline Analysis	50
6.4	Fuselage Shape Optimization with a Protruded Spherical Payload	52
6.5	Fuselage Shape Optimization with a Protruded Box Payload	59
7	Conclusions	65
7.1	Achievements	65
7.2	Future Work	66
	Bibliography	67

List of Tables

5.1	Optimization deformation strategy studies and corresponding drag coefficient reduction. .	43
6.1	Drag coefficient error in mesh refinement study.	49
6.2	ARX fuselage aerodynamic shape optimization parameters - Case 1	54
6.3	ARX fuselage aerodynamic aerodynamic shape optimization parameters - Case 3 (Symmetry-Enforced)	57
6.4	ARX fuselage aerodynamic shape optimization parameters - Case 4 (Symmetry-Enforced)	61
6.5	Comparison of drag coefficient variations and computational time.	61

List of Figures

1.1	Examples of UAV applications and industry adoption drivers [2, 3].	1
1.2	Forecast commercial drone market value	2
1.3	Example of TEKEVER UAS with different applications [3].	2
2.1	Evolution of fuselages	6
2.2	Aerodynamic forces	8
2.3	Categorization of drag	8
2.4	Detailed image of vortex formation in the junction between a fuselage and a wing.	10
2.5	The relative comparison between skin friction drag and pressure drag for various aerodynamic shapes and Reynolds numbers	11
2.6	Classification of optimization methods.	15
3.1	Overview of MACH-Aero optimization framework	19
3.2	Full CAD Model of the TEKEVER ARX Drone	21
3.3	TEKEVER ARX fuselage surface mesh	21
3.4	Zoomed-in view of overset grid near the fuselage within the farfield domain	22
3.5	Overset mesh cell roles after IHC.	24
4.1	Baseline simplified fuselage mesh	26
4.2	Effects of FFD box resolution	27
4.3	Different FFD control point displacement approaches	27
4.4	Different FFD control point strategies.	28
4.5	Camber parametrization for different parts of the fuselage	29
4.6	Camber displacement curve	30
4.7	Minimum distance between the surfaces	31
4.8	Calculate the closest pair of points, [101]	32
4.9	Intersected triangles test [104].	34
4.10	Projections of triangle vertices onto the intersection line L [104].	35
4.11	Triangulated surface constraint used in the optimization problem	36
5.1	Irregularities encountered using the directional deformation method	38

5.2	Comparison of resulting optimal shapes using different parametrization approaches (half-left is baseline, half-right is optimum).	39
5.3	Optimal shapes with camber deformation approach (half-left is baseline, half-right is optimum).	40
5.4	Optimal shapes using the triangulated surface constraint.	42
5.5	Comparison of optimal shapes with (red) or without (black) triangulated surface constraint.	42
6.1	Tekever ARX design.	45
6.2	Simplified TEKEVER ARX fuselage.	47
6.3	Simplified TEKEVER ARX fuselage domain discretization using three levels of mesh refinement.	48
6.4	Drag coefficient convergence study using Richardson's extrapolation	49
6.5	Pressure coefficient distribution on the baseline TEKEVER ARX fuselage.	50
6.6	Skin friction coefficient distribution on the baseline TEKEVER ARX fuselage.	51
6.7	First TEKEVER ARX fuselage optimization problem with an integrated sphere.	52
6.8	Example of a failed mesh in TEKEVER ARX fuselage optimization.	53
6.9	Pressure coefficient fuselage top view comparison of the two optimized shapes highlighting asymmetry.	55
6.10	Overlay of optimized fuselage shapes for Case 1 (red) and Case 2 (black), both integrating a protruded spherical payload	56
6.11	Pressure coefficient distribution on the optimized TEKEVER ARX fuselage (case 3).	57
6.12	Skin friction coefficient distribution on the optimized TEKEVER ARX fuselage (case 3).	58
6.13	Second TEKEVER ARX fuselage optimization problem with an integrated box.	59
6.14	Failed optimizations using different baseline fuselages	60
6.15	Pressure coefficient distribution on the optimized TEKEVER ARX fuselage (case 4).	62
6.16	TEKEVER ARX fuselage optimization problem with an integrated box (case 4)	63
6.17	Skin friction coefficient distribution on the optimized TEKEVER ARX fuselage (case 4).	63

Nomenclature

Greek symbols

α	Angle of attack
β	Angle of side-slip
Δs	Initial wall spacing
Δ	Difference in a quantity
δ_{ij}	Kronecker delta
ε	Dissipation rate
κ	Thermal conductivity coefficient
μ	Molecular viscosity coefficient
μ_t	Eddy viscosity
ν	Dynamic viscosity
$\tilde{\nu}$	Modified turbulent kinematic viscosity
π_1	Plane equation 1
π_2	Plane equation 2
ρ	Density
τ	Viscous shear stress tensor
τ	Shear stress
ω	Specific dissipation rate

Roman symbols

C_D	Coefficient of drag
C_f	Skin-friction drag coefficient
C_L	Coefficient of lift

C_M	Coefficient of moment
C_{D_0}	Parasite drag coefficient
$C_{D_{L\&P}}$	Drag coefficient due to leaks and protuberances
$C_{D_{misc}}$	Miscellaneous drag coefficient due to unique aircraft features such as landing gear, upswept fuselage, or flaps
C_{f_e}	Equivalent skin-friction coefficient
D	Drag
E	Total energy
$\vec{f}_e$	External forces
FF	Form factor
H	Enthalpy
$\mathbf{I}$	Unit tensor
k	Turbulent kinetic energy
L	Length (characteristic)
L	Lift
$\vec{M}$	Moment
p	Pressure
Q	Interference factor
q_H	Heat sources
$\vec{R}$	Aerodynamic resultant force
R	Residual function
S_{ij}	Strain rate tensor
S_{ref}	Reference area
S_{wet}	Wetted area
T	Flow temperature
$\vec{u}$	Velocity vector
u, v, w	Velocity Cartesian components
$\vec{v}$	Velocity vector

W_f Work due to applied forces
 y^+ Non-dimensional wall distance

Subscripts

∞ Free-stream condition
 i, j, k Computational indices
min Minimum
ref Reference condition
 x, y, z Cartesian components

Superscripts

T Transpose

Glossary

CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
CSM	Computational Structural Mechanics
DNS	Direct Numerical Simulation
FFD	Free-Form Deformation
IHC	Implicit Hole Cutting
KS	Kreisselmeier-Steinhauser
LES	Large Eddy Simulations
MDO	Multidisciplinary Design Optimization
PDE	Partial Differential Equations
RANS	Reynolds-Averaged Navier-Stokes
SA	Spalart-Allmaras
SLSQP	Sequential Least Squares Quadratic Programming
SQP	Sequential Quadratic Programming
URANS	Unstable Reynolds-Averaged Navier-Stokes
XDSM	eXtended Design Structure Matrix

Chapter 1

Introduction

This chapter provides an overview of the structure of the thesis, outlining the motivation, objectives and expected deliverables.

1.1 Background and Motivation

In the last years, the demand for drones, or unmanned aerial vehicles (UAVs), has grown in many areas, including defense, farming, surveillance, warfare, and aerial photography. As shown in Fig. 1.1a, several industries are benefiting from UAVs for environmental monitoring, particularly during extreme weather events such as forest fires or other natural disasters.

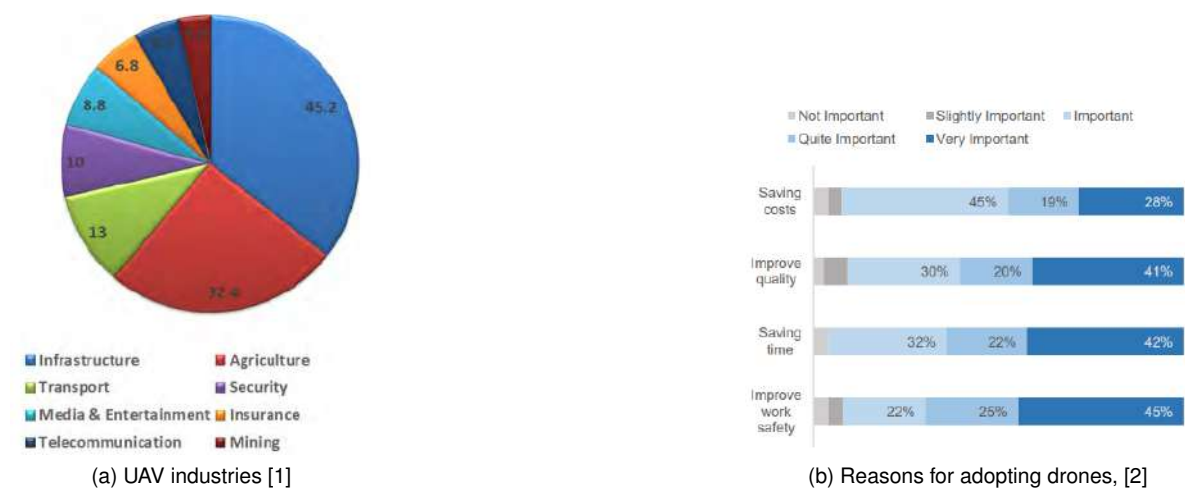


Figure 1.1: Examples of UAV applications and industry adoption drivers [2, 3].

- **Infrastructures:** UAVs are used, among other ways, for inspections and maintenance tasks related to crucial infrastructures, such as power lines, bridges, and pipelines. Drones can access hardly accessible areas and reduce risks and costs associated with traditional methods [1].
- **Agriculture:** UAVs find their application in agriculture for precision farming, which assists in pesticide application, irrigation control, and crop monitoring [4].

- Military: To make sure national security agencies have the most up-to-date equipment possible to stave off security breaches, regional nations are increasing their spending in military ammo [5].

From Fig. 1.1b, it can be concluded that the main reason for the use of drones by the majority of the operators is to save time and cost, and to also improving the quality and safety of the work [2, 6]. As these industries continue to grow, there is a need for more efficient, reliable, and high-performance drones. However, there are still challenges to overcome in developing drones that can fly longer distances and carry heavier loads.

As illustrated in Fig. 1.2, the global UAV market value is forecast to reach around 47.67 billion USD by 2032, growing at a compound annual growth rate (CAGR) of approximately 15.35% between 2024 and 2032. The market was valued at approximately 15.21 billion USD in 2023. This significant growth highlights the increasing demand and expanding applications of drones across different industries [7].

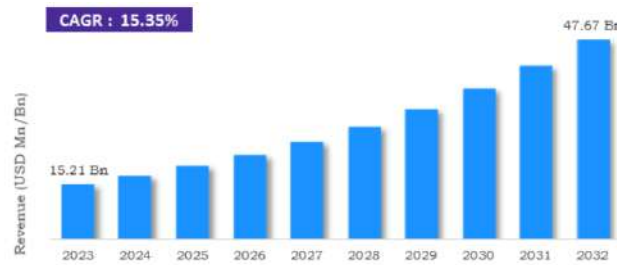


Figure 1.2: Forecast commercial drone market value [7]

TEKEVER is a Portuguese company among the drone industry that has consistently been at the top of the UAV market since 2010. The company is focused on developing next generation satellites and UAVs with the highest possible performance, pushing the boundaries of current technology by optimizing aerodynamic techniques. Currently, they offer two drone models: the AR3 and AR5. They are also working on a new, larger model called ARX, which will have unique features and is expected to significantly elevate the company's position in the industry [8].



(a) AR3



(b) ARX

Figure 1.3: Example of TEKEVER UAS with different applications [3].

The motivation behind this thesis, which was carried out in partnership with TEKEVER, was the reciprocal benefits that will result from the research, supporting the goals of the organisation as well as the progress of knowledge in this sector. For TEKEVER, the project offers the potential to enhance

their existing drone models, leading to more competitive products in the market. It provides a unique opportunity to combine theoretical knowledge within real-world applications, setting the new standard for aerodynamic shape optimizations in the UAV design business.

The motivation for this research was two sided: it would help the need to develop TEKEVER's capabilities in creating aerodynamic design that minimize resistance by developing computational algorithms and methods to optimize UAV fuselage surfaces; and it would allow me to apply and expand my expertise in aerodynamics and optimization in a real-world engineering application. The collaboration was expected to yield valuable insights that would not only advance TEKEVER's product development but also contribute to the broader field of UAV aerodynamic research.

1.2 Research Objectives and Deliverables

The main goal of this thesis is to explore and develop advanced aerodynamic optimization techniques for UAV fuselage design. This work addresses (1) the challenge of efficiently designing fuselage surfaces for a UAV subjected to internal components shape constraints and (2) shape-forming an existing fuselage surface to accommodate partial off-body components with efficient fairings.

To achieve these objectives, a study will be made in Computational Fluid Dynamics (CFD), design optimization, sensitivity analysis methods and fuselage and fairings design. After this revision, the goal is to define representative scenarios for typical aircraft missions and, based on those missions, identify performance metrics, geometric constraints and design variables. A more thorough study on adjoint-based numerical method and shape-deformation parametrization methods will help in the aerodynamic design of the fuselage and external shapes. This research will contribute to the development of robust algorithms that can be integrated into TEKEVER's existing design framework (TekAero), ultimately enhancing the performance and competitiveness of their UAV models and developing next-generation UAVs.

The current thesis has specific objectives that are tailored to meet this general objective. The key objectives for this study will be to:

- Minimize aerodynamic drag through optimization of fuselage form;
- Conduct a CFD analysis on the aircraft provided by TEKEVER;
- Identify key design variables and geometric constraints that can influence the overall performance of an UAV;
- Integrate the developed optimization techniques and algorithms into TEKEVER's existing UAV design and development processes.

This will be achieved through all the mentioned objectives, by using information and techniques provided by TEKEVER in order to develop this project in a more beneficial way for the company and the overall project goals.

On the other hand, the deliverables for this project will be centered around TEKEVER's framework by utilizing the already developed designs of AR5 and ARX. This way I will compare and validate different features around the fuselage of the drones. It is important to understand that to compare and develop different fuselages, I will have to create an optimization algorithm that integrates adjoint-based methods. This algorithm will improve the company's design process by offering more efficient and accurate optimization of UAV fuselage and fairings, reducing development time while ensuring high performance.

Finally, the outcome of the thesis will be a fully validated final design of the ARX fuselage and fairings, which can be directly applied to TEKEVER's future UAV models, supporting the company's goal of advancing its technology and maintaining its competitive edge in the UAV market.

1.3 Thesis Outline

The thesis is divided into seven chapters. Chapter 1 presents the motivation behind the work, the research aims and the collaboration context with TEKEVER. This also provides the general goal and deliverables of the project.

The second chapter begins with an overview of the evolution of the aircraft industry. It then covers fundamental aerodynamic concepts, including lift and the different types of drag. In the later half of this chapter, optimization techniques are discussed where a comparison between the various gradient-based and gradient-free techniques is provided.

Chapter 3 describes in detail the aerodynamic shape optimization framework with a focus on the MACH-Aero framework [9]. The key mesh generation concepts are also introduced in this chapter since they have major impact on the optimization results.

The fourth Chapter focuses on the fuselage shape deformation modeling. It starts with an overview of the parametrization methods where the Free-Form Deformation method is described in detail. Then the possible approaches chosen for this study are introduced.

Chapter 5 tries to use the approaches previously presented. Here, different problems will be addressed aiming to understand how the different approaches impact the overall optimization process.

In Chapter 6, two real-case scenarios inspired by TEKEVER projects are addressed. The integration of a spherical SATCOM antenna and the installation of a parachute housing box in the ARX will try to represent more realistic and challenging fuselage shape optimization problems.

Lastly, the conclusion and achievements of this work are given in the seventh chapter, together with some suggested future work.

Chapter 2

Aircraft Analysis and Design Methodologies

Optimization is a way of evolution, it is part of the human instincts. People tend to find the best possible solution for certain variables, changing the ones that can be controlled. In this chapter, the various methods used to design UAVs and achieve the most effective solution will be briefly described. This will include a discussion of existent techniques to reach the final design that complies to the problem's constraints.

2.1 Fuselage and Fairings Design

Since the beginning of the 20th century the aviation has gone through a lot of changes in the design concepts, wings, propulsion systems, and fuselages. The central body of an airplane, the fuselage, has seen extraordinary advances in response to evolving aerodynamic, structural, and technological demands. Early aircraft fuselages were simple, lightweight structures made primarily of wood and fabric, designed to minimize weight while providing enough structural integrity for flight. As aviation technology advanced, especially during and after World War I, metal construction became the standard, leading to the development of the monocoque and semi-monocoque designs, which offered enhanced strength and durability [10].

Evolution of Fuselages

After World War I, airplanes began to be seen as commercial assets, which led to improvements in fuselage design, materials, and aerodynamics. The Langley Memorial Aeronautical Laboratory's research influenced more streamlined designs [17], and the National Advisory Committee for Aeronautics (NACA) made significant advances in aircraft technology.

With the outbreak of World War II, tremendous progress was achieved in radar, jet engines, and aerodynamic design. Drag cleaning techniques were used in Langley's full-scale wind tunnel [18] to

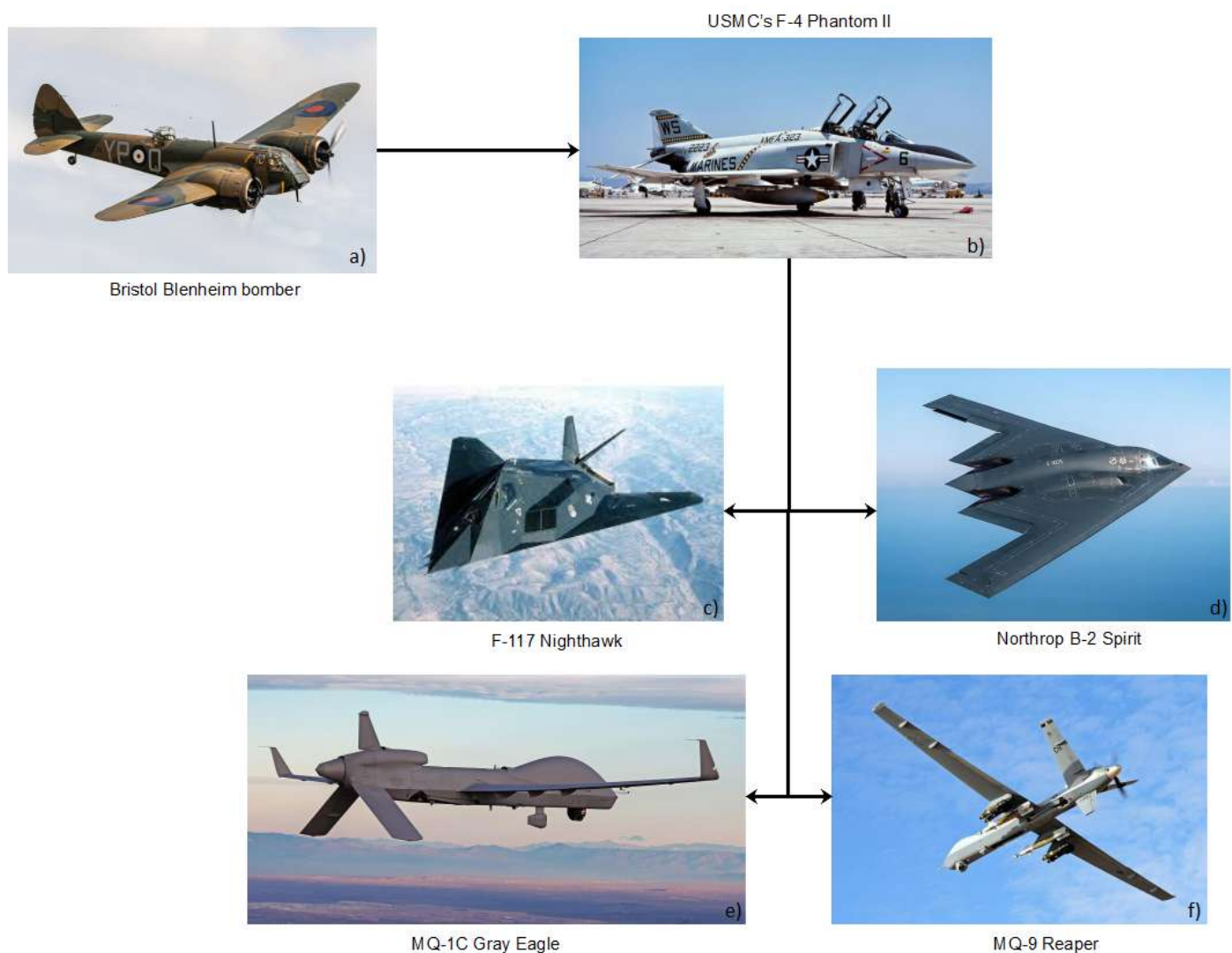


Figure 2.1: Evolution of fuselages [11–16]

detect and reduce drag sources, resulting in significant aerodynamic performance improvements. Up until then, and as seen in Fig. 2.1 a), a lot of aircraft had design flaws that had a large effect on how well they performed aerodynamically. Projecting parts like engines and landing gear, as well as bulky fuselages, enhanced form drag and parasite drag. Air leakage through structural gaps and surface roughness were other common issues with aircraft fuselages, which further reduced performance [17]. Ineffective air-cooling systems that limited engine performance were prevalent, as were corrosion and durability problems. Flow separation was another serious issue since it interfered with the fuselage's natural airflow, raising drag and decreasing efficiency [17]. Drag cleanup tests tried to address these challenges by sealing and modifying aircraft designs to reduce drag and increase overall efficiency [19].

The invention of the jet engine marked the next significant development in aircraft design. Engineers faced new issues as a result of these developments. Jet engines made it possible to fly faster and higher, which naturally called for innovative fuselage designs to overcome aerodynamic heating, which occurs

as aircraft move at higher speeds and the resulting heat caused by the relative motion between a body and a fluid to be transferred to the structure of the aircraft [20, 21]. Additionally, engineers had to take into account problems like pressurization [22] and structural stability.

Richard Whitcomb's "area rule" [23] optimized fuselage shapes for transonic speeds by reducing cross-sectional area, resulting in streamlined aircraft with low wave drag [24]. The USMC's F-4 Phantom II, shown in Fig. 2.1 b), demonstrates how blended wing and fuselage integration reduces aerodynamic stress [25].

The introduction of stealth aircraft from 1980s to 2000s represented a breakthrough in fuselage design. Figures 2.1 c) and d) illustrate stealth fuselages, which used faceted and angular designs to deflect radar waves, significantly lowering radar cross-sections. These designs prioritised survivability by optimizing the fuselage for low radar reflections [26].

Drones and Unmanned Systems

One of the main advantages of UAVs compared to crewed aircraft is that pressurization is not required. Because there is no need to keep the cabin pressure constant and breathable, UAV designs can be optimized with lighter, simpler structures, which save weight and improve fuel efficiency. Consequently, it also allows long-duration and high-altitude missions.

Due to the multiple missions they must perform, the main challenge of UAVs is the integration of different payloads. UAVs may be required to carry environmental monitoring equipment, sensors, radars, or communication systems, each of which with different sizes and shapes. Radar systems, for instance, frequently need larger, more streamlined fairings to reduce drag, whereas smaller electro-optic sensors can fit into small gimbals that barely change the UAV's aerodynamics and centre of gravity. With the rapid advances in electronics technology, new payloads are expected more frequently than new developments in aircraft [27], highlighting the importance of re-shaping the fuselage.

By analyzing the two last images in Fig. 2.1, we can conclude that both have distinct advantages, the MQ-9 Reaper, Fig. 2.1 f), has a more compact fuselage providing a streamlined profile when unladen, however the drag caused by the external missiles and additional payloads below the wings will affect the overall drag of the UAV [24]. On the other hand, the MQ-1C Gray Eagle, shown in Fig. 2.1 e), has an internal payload bay, allowing it to maintain a consistent aerodynamic profile with reduced drag, even when fully equipped. Although this internal payload minimizes drag enhancing endurance and stability, it also restrains the size and configurations of the payloads. These design decisions provide a balance between cargo capacity and drag reduction according to mission requirements, reflecting various operational priorities.

2.2 Aerodynamic Forces

The study of aerodynamics examines the interactions between gases and moving objects. As a body interacts with the flow, several forces are generated. These forces are produced by the distribution of pressure (p) and shear stress (τ) across the surface of the body. The resultant aerodynamic force

($\vec{R}$) and moment ($\vec{M}$) acting on the body are the net effect of these two sources. The fluid velocity far upstream of the body is known as free-stream velocity, V_∞ . By definition, the parts of $\vec{R}$ that act parallel to V_∞ and perpendicular to it are the aerodynamic forces of drag (D) and lift (L), respectively, as seen in Fig. 2.2. One important efficiency parameter is the lift-to-drag ratio (L/D), where higher values denote better aerodynamic performance [21, 28].

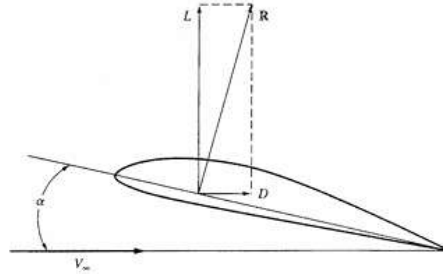


Figure 2.2: Aerodynamic forces [29].

The performance, fuel efficiency, range, and overall operating costs of an aircraft are all directly impacted by drag, making its study essential to the aircraft industry [30]. An aircraft's passage through the air is resisted by drag, which can be reduced to significantly improve energy consumption and emissions, two factors that are critical to the sustainability of the economy and the environment.

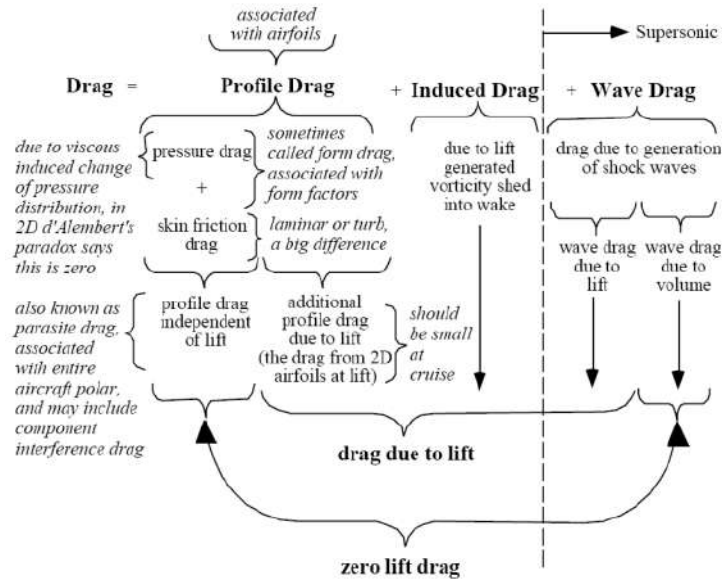


Figure 2.3: Categorization of drag, [31]

As observed in Fig. 2.3, there are several types of drag. Generally speaking, the drag is usually divided into three different categories, 1) Parasite drag consisting of form drag, skin friction, and interference drag; 2) Lift-induced drag; and 3) Wave drag [30]. It should be noted that in this thesis, the main focus will be on the first category, since the wave drag [32] is important only for transonic and supersonic speed. Lift-induced drag, on the other hand, arises due to lift generation. While certain fuselage shapes can contribute to lift, the configurations examined in this thesis are not designed with that purpose in mind. Accordingly, the lift produced by each geometry is minimal, and any associated lift-induced drag

can be considered negligible [33].

Based on [21], there are two methods to estimate the parasite drag C_{D_0} . The first method assumes that the aircraft's parasite drag is due to skin friction drag with a small contribution from pressure drag. This last type of drag results from the pressure imbalance in the drag direction caused by the separated flow, so it might be considered relatively small if there are small flow separations. The equivalent skin-friction coefficient (C_{fe}), which takes into consideration both drag components, is a representation of this relationship. An initial estimate of C_{D_0} is produced by multiplying (C_{fe}) by the ratio of the wetted area (S_{wet}) to the reference area (S_{ref}). This equation is a preliminary approach for subsonic analysis [34] and is given by:

$$C_{D_0} = C_{fe} \frac{S_{wet}}{S_{ref}} \quad (2.1)$$

The second method, the Component Buildup Method, adds together the contributions of each individual drag component and calculates the subsonic parasite drag coefficient (C_{D_0}). The Drag for each component is computed using the flat-plate skin-friction drag coefficient (C_f), an interference factor (Q) to account for component interaction effects, and a form factor (FF) to account for pressure drag from viscous separation. The wetted area (S_{wet}) and the reference area (S_{ref}) are then multiplied to get the total component drag:

$$(C_{D_0})_{\text{subsonic}} = \frac{\sum (C_f \cdot FF \cdot Q \cdot S_{wet})}{S_{ref}} + C_{D_{\text{misc}}} + C_{D_{L\&P}} \quad (2.2)$$

where $C_{D_{L\&P}}$ takes into account leaks and protuberances, ($C_{D_{\text{misc}}}$) takes into account various drag from unique characteristics like landing gear, an upswept fuselage, or flaps. Additional contributions from wave drag are included for supersonic situations. By taking into account the contributions of individual components and their aerodynamic interactions, this method offers a methodical approach to calculating parasite drag, providing a thorough study that is essential for optimizing aircraft performance [35].

Interference drag is the result of aerodynamic interactions between two or more body parts. In general, it can be separated into two categories: boundary layer interference and potential flow interference. The main problem in scenarios involving turbulent boundary layers is the category - boundary layer interference. Due to the fact that the flow at the junction between components differ significantly from the typical boundary layer flow that would form on each component separately, this kind of drag happens. This discrepancy results from the junction's strong secondary flow, which is defined by velocity components normal to the primary flow direction [36].

Two main effects, however, typically dominate this secondary flow in practical cases. When the approaching turbulent boundary layer goes around an obstruction, it first becomes distorted. This skewing causes streamwise vorticity in the junction by stretching and rotating the boundary layer's vortex lines. Secondly, the obstruction's existence produces a blockage effect that compels the approaching boundary layer to encounter steep negative pressure gradients close to the leading edge. In many cases, this trailing vortex becomes the dominant flow feature, changing the local flow dynamics and contributing to drag. In Fig. 2.4, it is clear how the wing and fuselage boundary layers interact, forming vortices near the trailing edge [37].

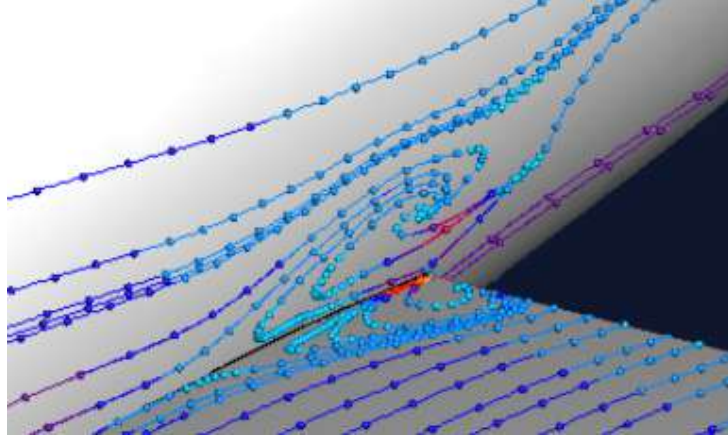


Figure 2.4: Detailed image of vortex formation in the junction between a fuselage and a wing [38].

The extra drag caused by this interaction and known as interference drag is commonly measured as the difference between the drag of the components taken into consideration individually and their combined drag. Mathematically, interference drag is $\Delta D = D_{1+2} - (D_1 + D_2)$. Three primary factors make up interference drag. As mentioned, the first is the distortion of velocity profiles in the junction brought on by the secondary flow and the change in surface shear stress brought on by the boundary layer merging. The vortex created by boundary layer separation and roll-up is the main contributor to the second, which is the induced drag brought on by energy losses to the secondary flow in the junction. The third component, is due to the modified growth of the viscous layer in the junction, which changes the pressure drag [37].

These effects are frequently seen at, for example, points where landing gear struts attach to the airframe or, as already shown in Fig. 2.4, where wings intersect the fuselage. Even when there is no direct contact between the components, the interaction can still happen when they are positioned closely together or behind each other [28]. It is difficult to calculate interference drag precisely because it requires intricate relationships between flow separation and boundary layers. The prediction of this type of drag is based on old experimental data [29].

As previously referred, the interference factor (Q) measures the increase in drag caused by airflow interactions between components. While well-designed arrangements, such filleted wings, usually have $Q \approx 1.0$, close-mounted components, like nacelles, have Q values of 1.3–1.5. By protecting components from increased dynamic pressure, favorable interference can occasionally lower drag [21]. These elements are essential for precise parasite drag estimation with Eq.(2.2).

Another important type of drag, the form drag, also referred as pressure drag, is the drag on a body moving through a fluid due to the pressure difference between the front and rear of the body. This type of drag is directly dependent on the shape, or "form", of the body, and is most pronounced when there is flow separation. As the fluid flows over a solid body such as a fuselage, it must go around deviations in the shape of the body. When the shape is poorly streamlined, the flow is not capable of remaining attached to the surface and is separated from the surface, creating a wake area that is a region of low pressure with increased turbulence. The larger and more chaotic this wake, the greater the pressure difference between forward-facing (high-pressure) and rear-facing (low-pressure) surfaces,

and the greater the drag force contrary to the direction of motion. Figure 2.5 shows how the geometry affects the predominant type of drag, and how the cylinder and the flat plate have much more drag than the other cases due to larger flow separation.

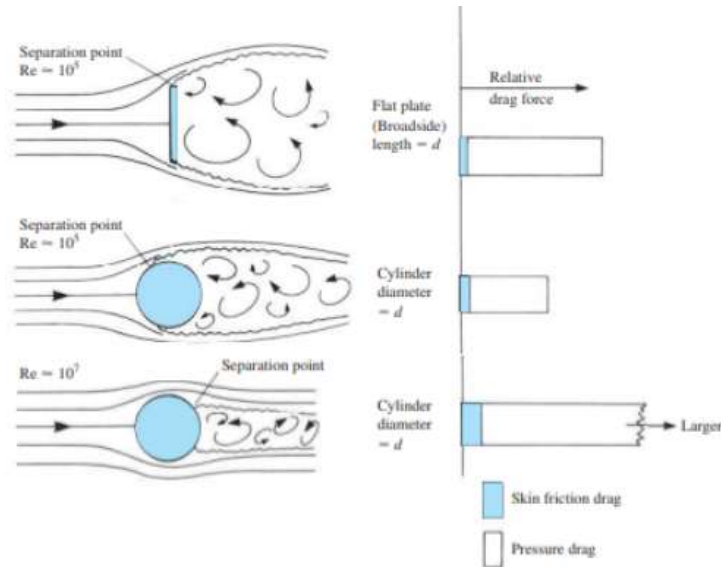


Figure 2.5: The relative comparison between skin friction drag and pressure drag for various aerodynamic shapes and Reynolds numbers adapted from [39].

In fuselage design, form drag is a significant component of the overall aerodynamic performance. The fuselage, having a relatively large area and curved shape, is very susceptible to flow separation if not effectively optimized. A fuselage with sudden changes or sharp geometric features will result in premature flow detachment, which enlarges the wake and hence the form drag. In contrast, a highly streamlined fuselage, where cross-sectional changes occur smoothly, encourages the fluid to stay attached to the surface as far aft as possible, minimizing wake size and pressure drag.

Note that, as the Reynolds numbers increase the importance of pressure drag becomes even greater, since the inertial forces outweigh viscous forces, making flow separation and wake formation more pronounced [40]. This is also shown in Fig. 2.5. This characteristic reinforces the need to design fuselage geometries that delay or mitigate flow separation under operational conditions.

In the specific case of this thesis, the optimization of fuselage shapes has exactly this goal in mind: by refining the fuselage contours and delivering smoother area transitions, more attached flow patterns are possible, and form drag is minimized, while the lift-to-drag ratio (L/D) is maximized. As it will be discussed in Secs. 5 and 6, the optimization process also considers the integration of a payload intersecting the fuselage, which will likely increase both the wetted area and volume. Nonetheless, the primary objective remains the minimization of drag coefficient.

In conclusion, form drag is an important aerodynamic factor in evaluating fuselage optimization results. Relative changes in form drag before and after shape optimization will provide a clear measure of success, demonstrating the advantage of geometrical advancements and directly translating into aircraft performance.

2.3 Computational Fluid Dynamics (CFD)

To model and analyse turbulent flows, which are frequently seen in real-world applications, engineers depend heavily on Computational Fluid Dynamics (CFD). To investigate turbulence, engineers have historically used experimental techniques, but these approaches are expensive, time-consuming, and have limited accuracy, particularly as engineering designs get more complicated. Direct analysis are complicated because turbulent flows are extremely unstable, three-dimensional, and contain a wide variety of length and time scales. With the help of CFD, engineers can get around these restrictions and design fuselages, turbines, and wings. Note, however, that CFD analyses still involve certain limitations, such as modelling assumptions, numerical errors, and high computational cost [41].

The Navier-Stokes equations express the fundamental laws of mass, momentum, and energy conservation in fluid flow. These equations are crucial to describe fluid dynamics, capturing a wide range of behaviour from laminar to turbulent flows. Their differentiated conservative form is given by:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) &= 0 \\ \frac{\partial(\rho \vec{v})}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v} \otimes \vec{v}) + \vec{\nabla} p \vec{I} - \vec{\nabla} \cdot \bar{\bar{\tau}} &= \vec{f}_e \quad , \\ \frac{\partial(\rho E)}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v} H - \bar{\bar{\tau}} \cdot \vec{v} - k \vec{\nabla} T) &= W_f + q_H \end{aligned} \quad (2.3)$$

where ρ denotes the density, p represents the pressure, $\vec{v}$ is the velocity vector, and E the total energy. The unit tensor is denoted by $\vec{I}$, $\bar{\bar{\tau}}$ represents the viscous shear stress tensor, k is the thermal conductivity, H is the enthalpy, and T denotes the flow temperature. The terms on the right-hand side capture source terms: $\vec{f}_e$ is the external forces acting on the flow, W_f denotes the work done by these forces, and q_H accounts for heat sources [42].

CFD approaches range from Reynolds-Averaged Navier-Stokes (RANS) for time-averaged flows to Large Eddy Simulation (LES) for larger turbulent structures, and Direct Numerical Simulation (DNS) for resolving all scales. To improve engineering capabilities and create effective designs, CFD is essential because it enables engineers to reach an equilibrium between accuracy and computational cost [41].

Techniques like Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES) can be used to simulate turbulence. Because DNS requires fine grids and small time steps, especially at high Reynolds numbers, it is computationally expensive for practical applications, despite being conceptually simple and extremely precise [43]. By simulating lower scales and concentrating on solving larger, more intense eddies [44], LES minimises processing work. However, the Reynolds-Averaged Navier-Stokes (RANS) method will be used for this thesis since it offers an effective and affordable way to simulate turbulent flows in engineering applications [45, 46].

By averaging the Navier-Stokes equations, the Reynolds-averaged Navier-Stokes (RANS) equations are created to simulate the effects of turbulence on the mean flow. The treatment of fluctuations, such as time-averaging, ensemble-averaging, or filtering, determines whether this averaging is unsteady or stable (removing all unsteadiness). Turbulence models approximate complicated interactions that are not closed, whereas unsteady RANS (URANS) capture predictable structures in flow in real-world ap-

plications. In order to balance computing efficiency and flow prediction accuracy, RANS models provide engineering approximations of turbulence.

In steady flow, every variable can be written as the sum of a time-averaged value $\bar{\phi}$ and a fluctuation about that value ϕ' as,

$$\phi(x_i, t) = \bar{\phi}(x_i) + \phi'(x_i, t), \quad (2.4)$$

such as velocity (u_i), into their time-averaged ($\bar{u}_i$) and fluctuating (u'_i) components [41, 47]. When applied to the Navier-Stokes equations, this decomposition, also referred to as the Reynolds decomposition, produces equations that explain how the mean flow fields change over time. Substituting these terms generates other quantities that reflect the effects of turbulence, including the Reynolds stresses ($\overline{u'_i u'_j}$), which measure the transfer of momentum caused by turbulent waves. This results in the additional terms

$$\bar{\tau}^R = -\overline{\rho \vec{u}'' \otimes \vec{u}''} \quad \text{and} \quad \bar{q}^R = -C_p \overline{\rho \vec{u}'' T''}$$

that represent a fluctuation in the velocity field and for energy equations, the turbulent heat flux vector, respectively [48, 49].

The RANS equations take the general form of the original Navier-Stokes equations but include these extra terms that require modeling for closure, which means that they contain more variables than equations,

$$\begin{aligned} \frac{\partial \bar{\rho}}{\partial t} + \vec{\nabla} \cdot (\bar{\rho} \vec{v}) &= 0 \\ \frac{\partial (\bar{\rho} \vec{v})}{\partial t} + \vec{\nabla} \cdot (\bar{\rho} \vec{v} \otimes \vec{v} + \bar{p} \bar{\mathbf{I}} - \bar{\tau} + \bar{\tau}^R) &= \rho \vec{f}_e \\ \frac{\partial (\bar{\rho} \tilde{E})}{\partial t} + \vec{\nabla} \cdot (\bar{\rho} \vec{v} \tilde{E} + \vec{v} \cdot \bar{p} + \vec{v} \cdot \bar{\tau}^R - \bar{q}^R) &= W_f + q_H \end{aligned} \quad (2.5)$$

To address this problem the use of turbulence models is essential (e.g., $k - \epsilon$, $k - \omega$, or Spalart-Allmaras models). These models approximate the effects of turbulence, allowing for a solution to be obtained in a realistic amount of time.

Turbulence Models

Many turbulence models have been created over time to forecast how turbulent flows would affect fluid dynamics. By striking a balance between accuracy and computational cost, these models allow engineers to describe intricate turbulence while controlling computational expenses.

Most researchers use the Boussinesq approximation with suitable generalization for compressible flows [49]. This approximation introduces the concept of eddy viscosity (μ_t), which acts as a proportionality factor linking the turbulent shear stress to the mean strain rate,

$$\bar{\rho} \tau_{ij} \equiv -\overline{\rho u'_i u'_j} = 2\mu_t \left(S_{ij} - \frac{1}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} k \delta_{ij}. \quad (2.6)$$

Note that the most important consideration in Eq.(2.6) is guaranteeing that the trace of τ_{ij} is equal to $-2k$, where $k = \frac{1}{2} \overline{u'_i u'_i}$.

One of the most recognized and used equations in nuclear and aerospace applications is the Spalart - Allmaras (SA) [50]. This is one of the most straightforward, yet efficient method for simulating turbulent flows. This model is based on the Boussinesq hypothesis and its "standard" equation is given as

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) \right] + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i} \quad (2.7)$$

where $\tilde{\nu}$ is the modified turbulent kinematic viscosity; C_{b1} , C_{b2} , and C_{w1} are model constants; f_{t2} and f_w are empirical functions used to account for compressibility and near-wall effects; σ and κ are model constants related to the turbulent Prandtl number and the von Kármán constant, respectively; S represents source terms for compressibility corrections; and d denotes the distance to the nearest wall.

The turbulent eddy viscosity is then computed from:

$$\mu_t = \rho \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}}, \quad \chi = \frac{\tilde{\nu}}{\nu}$$

where ρ is the fluid density, μ is the molecular dynamic viscosity, and $\nu = \mu/\rho$ is the corresponding molecular kinematic viscosity.

The Spalart-Allmaras model solves a single transport equation for a modified turbulent viscosity. It produces better results than algebraic models for separated flows and accurately predicts skin friction for attached boundary layers. The model presents no major numerical obstacles and is computationally efficient [49].

On the other hand, two-equation turbulence models, including the $k - \varepsilon$ and $k - \omega$ models, are frequently employed because they provide a balance between predictive power and complexity. A transport equation for the turbulent kinetic energy (k) and another for the dissipation rate (ε) or specific dissipation rate (ω) are solved independently. In contrast to one-equation or algebraic models, this offers a more thorough representation of turbulence [51].

The $k - \varepsilon$ model has been widely used in the past decades, because it can be applied to a wide range of flows. To effectively represent the viscous sublayer, it needs viscous adjustments and has trouble with flows with significant adverse pressure gradients. Because of these drawbacks, it is less accurate at predicting separated flows and frequently needs to be adjusted for particular uses. Since the $k - \omega$ model is substantially more accurate for two-dimensional boundary layers with both favourable and unfavourable pressure gradients, it replaced this model. Additionally, the model is easily integrated via the viscous sublayer without the need for any specific viscous corrections [49, 52].

The turbulence model used in this thesis is the Spalart-Allmaras model. Despite the benefits presented in the last paragraph, both approaches have disadvantages in some circumstances. Additionally, since the ADFlow solver only provides a differentiated version of the Spalart-Allmaras turbulence model, and gradient-based optimization will be used, this decision becomes essential for our research [53].

2.4 Optimization Methods

Computational design problems are becoming more common in manufacturing industries [54]. For complex systems like UAVs, the design is one of the most important characteristics to take into consideration. Optimizing to the specified objectives (goal functions) can lead to better, cheaper products and reduced production times. It is an engineers' job to formulate the correct problem and find the desired parameters according to the objectives. In mathematical language, the optimization task can be posed as

$$\begin{aligned}
 &\text{Minimize} && f(x) \\
 &\text{by varying} && x_i && i = 1, \dots, n_x \\
 &\text{subject to} && g_j(x) \leq 0 && j = 1, \dots, n_g, \\
 & && h_l(x) = 0 && l = 1, \dots, n_h, \\
 & && \underline{x}_i \leq x_i \leq \bar{x}_i && i = 1, \dots, n_x.
 \end{aligned} \tag{2.8}$$

and it basically demonstrates how an optimization problem works: by adjusting the design variables x_i within the boundaries of the constraints g_j and h_l , the objective function f can be minimized [55, 56].

Several methods can be used in optimization in different applications.

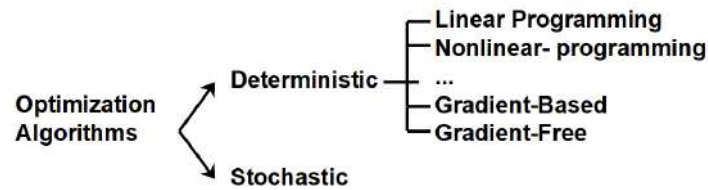


Figure 2.6: Classification of optimization methods [57].

Optimization Methods

As illustrated in Fig. 2.6, there are two main optimization groups, the stochastic optimization method that accounts for uncertainty and randomness in its process [58–60] and the deterministic methods which are known for their repeatable and reliable results when given specific inputs, forming the basis for decision making across engineering, economics, and several scientific fields.

The linear Programming (LP) is an optimization problem with a linear objective and linear constraints as it can be seen being used in this specific example of aircraft maintenance [61]. There are also the Gradient-Free and Gradient based methods. The first one, is an algorithm used when gradients are unavailable, as in the cases where black-box functions are used [62]. These, refer to functions whose internal operations or structure are too complicated to be directly examined. Although the function may be evaluated for inputs and will provide outputs, its derivative, gradient, and underlying equations are unknown. In general, gradient-free methods are simpler to implement but significantly less effective, especially when the problem's dimension grows. The fact that gradient-free algorithms do not presume

function continuity is one of their main advantages, making it useful for a wide range of problems [56, 63]. On the other hand, when gradient information is available, gradient-based techniques—which are covered in detail in this thesis—are effective instruments that provide effective optimal solutions.

2.4.1 Gradient-Based Methods

Although there are unconstrained gradient-based approaches, these will not be covered in-depth because constrained optimization is the main emphasis of this thesis as well as in the majority of engineering optimization problems. Note that constrained approaches are built on unconstrained gradient-based methods [56].

It is commonly known that gradient-based methods typically need much fewer function evaluations than gradient-free approaches [64]. The number of function evaluations needed in gradient-free methods tend to increase much faster than in gradient-based methods, where the growth is typically linear with the size of the problem [65]. In a general constrained optimization problem, as mathematically expressed in Eq. (2.8), the vector of inequality constraints is represented by $g(x)$ and the vector of equality constraints by $h(x)$. The lower and upper bounds on design variables are represented by $\underline{x}$ and $\overline{x}$, also referred to as bound constraints. Although objective and constraint functions might be nonlinear, gradient-based optimization techniques require that they must be continuously differentiable [56].

Common methods used in optimization problems include Sequential Quadratic Programming (SQP) and interior-point methods. Penalty methods were also once widely used, but they have been replaced by more effective approaches [66]. In this thesis, the focus will be on (SQP) methods, as they are widely used in problems similar to those addressed here [67, 68].

In the aerospace industry, Sequential Least Squares Quadratic Programming method (SLSQP) is widely used. This method is a variation of the SQP that combines elements of least squares optimization. It uses the Han–Powell quasi-Newton method with a BFGS update of the B-matrix and an L1-test function in the step-length algorithm. It approximates the Lagrangian to a quadratic function and linearizes both equality and inequality constraints. SLSQP is especially helpful because it is computationally efficient and can handle complex, nonlinear objectives and constraints. In order to approximate the answer iteratively, the approach divides the primary optimization problem into a series of quadratic subproblems. SLSQP updates the solution until convergence conditions are satisfied by minimizing a quadratic approximation of the Lagrangian, a function that takes into account both the objective and constraints, at each stage. A detailed comparison study between gradient-based and gradient-free methods as well as a more precise description on this optimization method (SLSQP) can be found in the work of Ma et al. [69] and Lyu et al. [65]. These were crucial to decide the methods used in this project. The complexity of the numerical software needed to perform the sequential quadratic programming approach in real time is one of its primary disadvantages; on the other hand, and to conclude, its benefits include flexibility in allowing non-quadratic cost functions, nonlinear models, and multiple objectives like singularity avoidance.

2.4.2 Adjoint Method

After selecting the optimization algorithm, it is important to select a suitable algorithm to effectively compute the derivatives based on the problem dimension.

In this thesis, the chosen method was the adjoint method. The adjoint method is considered to be an efficient and accurate method to evaluate the sensitivities of a cost function [70]. One of its key advantages is that its computational cost remains independent of the number of design variables, which contrasts with traditional methods like the Finite-difference method and the complex-step derivative approximation method that scale linearly or worse as the number of variables increases [56]. Future aerodynamic design applications rely on this feature which is of great interest to refine the optimization process.

There are two different approaches in this method, the discrete and the continuous ones. In the continuous approach, the initial continuous governing equations, such as Partial Differential Equations (PDE), are first linearized and then discretized. The discrete technique does not linearize the equations until they have been discretized into residual equations, $r(u) = 0$. Both have advantages and disadvantages but is considered, in general, that the discrete approach is easier to apply [71].

The *adjoint method* is primarily used to compute the *total derivative* of a function of interest f with respect to the design variables x , denoted as $\frac{df}{dx}$. In typical optimization problems, f depends not only on x , the design variables, but also on u , the state variables. These state variables u are implicitly related to x through a set of governing equations represented by the residual function $R(x, u) = 0$.

To compute the total derivative $\frac{df}{dx}$, the *chain rule* is applied,

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u} \frac{du}{dx}. \quad (2.9)$$

Since u depends on x through $R(x, u) = 0$, we can simply differentiate the residual equation to obtain $\frac{du}{dx}$.

$$\frac{dR}{dx} = \frac{\partial R}{\partial x} + \frac{\partial R}{\partial u} \frac{du}{dx} = 0. \quad (2.10)$$

This can be rearranged to obtain the linear system,

$$\frac{\partial R}{\partial u} \frac{du}{dx} = -\frac{\partial R}{\partial x}.$$

Then, we solve this equation repeatedly until all the columns in the $\frac{du}{dx}$ matrix are computed. However, this direct approach can be computationally costly, it is proportional to the number of design variables. Instead, the adjoint method introduces an adjoint vector ψ to simplify the expression. To eliminate the need for calculating $\frac{du}{dx}$ directly, we define the total derivative of F with respect to x as:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \underbrace{\frac{\partial f}{\partial u} \left[\frac{\partial R}{\partial u} \right]^{-1}}_{\Psi} \frac{\partial R}{\partial x}. \quad (2.11)$$

which is then simplified as

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \psi^T \frac{\partial R}{\partial x}, \quad (2.12)$$

where ψ is chosen such that

$$\left(\frac{\partial R}{\partial u} \right)^T \psi = \frac{\partial f}{\partial u}. \quad (2.13)$$

This equation is known as the adjoint equation. By solving for the adjoint vector ψ using the adjoint equation, it can then be used to evaluate the gradient of the cost function f with respect to all design variables x , without directly calculating $\frac{du}{dx}$, using Eq.(2.12)

It is important to note that the adjoint and direct methods scale differently with constraints and design variables. The adjoint method is effective in problems with a large number of design variables and few constraints since its cost is independent of the number of design variables as it is seen in Eq. (2.13). On the other hand, the direct method is better suited for situations with more constraints. Because of this, the adjoint method is best suited for large-scale issues like aerodynamic design problems [56, 72–74].

Chapter 3

Aerodynamic Shape Optimization Framework

In this chapter, the methodology used to execute the aerodynamic optimization is presented. The approach is based on the MACH-Aero Framework, an open-source multidisciplinary design optimization (MDO) framework developed by the MDO Lab at the University of Michigan [9].

3.1 Overview

This framework integrates several tools that work together to deform the geometry, generate quality meshes, carry out CFD simulations, and iteratively update the design based on the constraints and design variables defined in the problem. A summary of this process is illustrated in Fig. 3.1, representing how each tool drives the optimization cycle. One of the most important aspects of this workflow is the mesh generation step, which, as it will become clear, is required to achieve reliable and accurate simulation results. For this reason, the process of meshing has been explained in a dedicated section (Sec. 3.2).

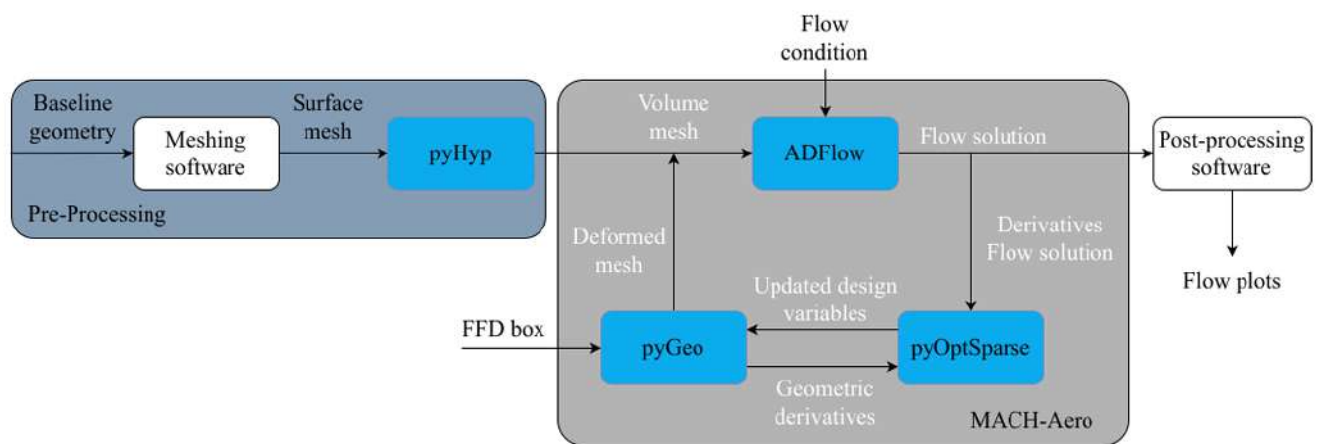


Figure 3.1: Overview of MACH-Aero optimization framework [75]

Each component of the framework shown in Fig.3.1 is briefly described next, starting with the geometry parameterization process.

In this work, the fuselages are parameterized using pyGeo, which maps the surface mesh of the geometry to a block of control points defined by the Free-Form Deformation (FFD) box [76]. This method, which will be explained in detail in Chapter 4, enables smooth and continuous shape changes by displacing the control points. This produces a flexible and efficient way to change the fuselage shape in the optimization process without significant computational cost.

Once the surface deformation is defined, it is propagated to the volume mesh using IDWarp [77], a MACH-Aero mesh-warping tool that preserves mesh quality through an inverse distance weighting algorithm. During the pre-processing stage, the initial mesh is generated through pyHyp, a hyperbolic mesh generation tool that employs a marching algorithm to create high-quality structured grids for CFD simulations. All of these tools work together to allow effective handling of complex geometries with good quality mesh and fast optimization cycles.

Additionally, ADflow is the computational fluid dynamics (CFD) solver responsible for analyzing the aerodynamic performance of each candidate geometry [53]. It receives the deformed mesh and solves the Reynolds-averaged Navier–Stokes equations with the Spalart-Allmaras turbulence model to compute flow variables such as pressure, velocity, and forces acting on the surface.

Finally, PyOptSparse [78], is an object-oriented framework for formulating and solving nonlinear constrained optimization problems in an efficient, reusable, and portable manner. It coordinates the design variable updates, uses gradient information provided by pyGeo and ADflow, and drives the iterative process towards an optimal solution under the defined constraints. When the convergence criteria are reached, this cycle stops and post-processing analysis to these final results can be done.

All of these tools together allow for efficient handling of complex geometries, maintaining mesh quality, and enabling fast optimization cycles.

Following the overview of the framework, the focus now shifts to the mesh generation process, which represents a critical step in guaranteeing the accuracy and robustness of the CFD simulations employed during the optimization.

3.2 Meshing Process

Mesh generation is a primary operation of the aerodynamic optimization approach because it has direct impact on CFD accuracy as well as the computational cost of simulations [79]. During this work, meshes were created to meet the requirements of the flow solver utilized within this project ADflow, which supports structured multi-block as well as overset 2D grids. Since these are the grid types accepted by the solver, the resulting meshes were exported in CGNS (CFD General Notation System) format [77]. In the following subsections, the methodology used to generate the surface mesh, volume mesh, and overset mesh, respectively, along with main tools and parameters involved in the respective step, are presented.



Figure 3.2: Full CAD Model of the TEKEVER ARX Drone

3.2.1 Surface Mesh

Surface mesh, shown in Fig. 3.3, forms the outer boundary of the computational domain and provides an initial point to create a volume mesh. For this purpose, structured surface meshes were generated from the fuselage CAD geometry, Fig. 3.2 using a multi-block topology. Through this, it will have greater control in mesh distribution at key locations such as the nose and tail with steeper pressure gradient and curvature. Once the outer surface was discretised, this surface mesh served as the basis for generating a high-quality volume mesh around the fuselage.

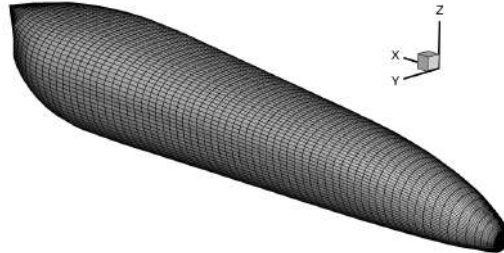


Figure 3.3: TEKEVER ARX fuselage surface mesh

3.2.2 Volume Mesh

Once the surface mesh was defined, the volume mesh was generated using pyHyp, a hyperbolic mesh generator integrated within the MACH-Aero framework. PyHyp starts with a surface corresponding to the geometry, in this case, of the ARX or the mesh used for the parametrization studies; and then extrudes the mesh in successive layers until it reaches the defined distance from the original surface, forming a structured volume mesh suitable for high-fidelity CFD analysis. Several parameters are necessary to be defined to complete this task. These include the first-layer height (s_0), the number of layers to extrude (N), and the total marching distance (marchDist), for example. Besides these, more parameters are available to change the algorithm that generates the mesh. However, default values were used, as they seemed appropriate to generate the fuselage due to their previous integration in similar shape optimization problems [80].

Furthermore, the computational domain was extended in the streamwise direction up to five chord lengths behind the fuselage to ensure proper resolution of the wake and pressure recovery region. Following this, the surface was extruded in all outward directions to generate the full volume mesh around the fuselage. This ensured sufficient space for flow development, accurate capture of near-wall effects, and proper representation of the farfield behaviour. This flow field can be observed below in Fig.3.4.

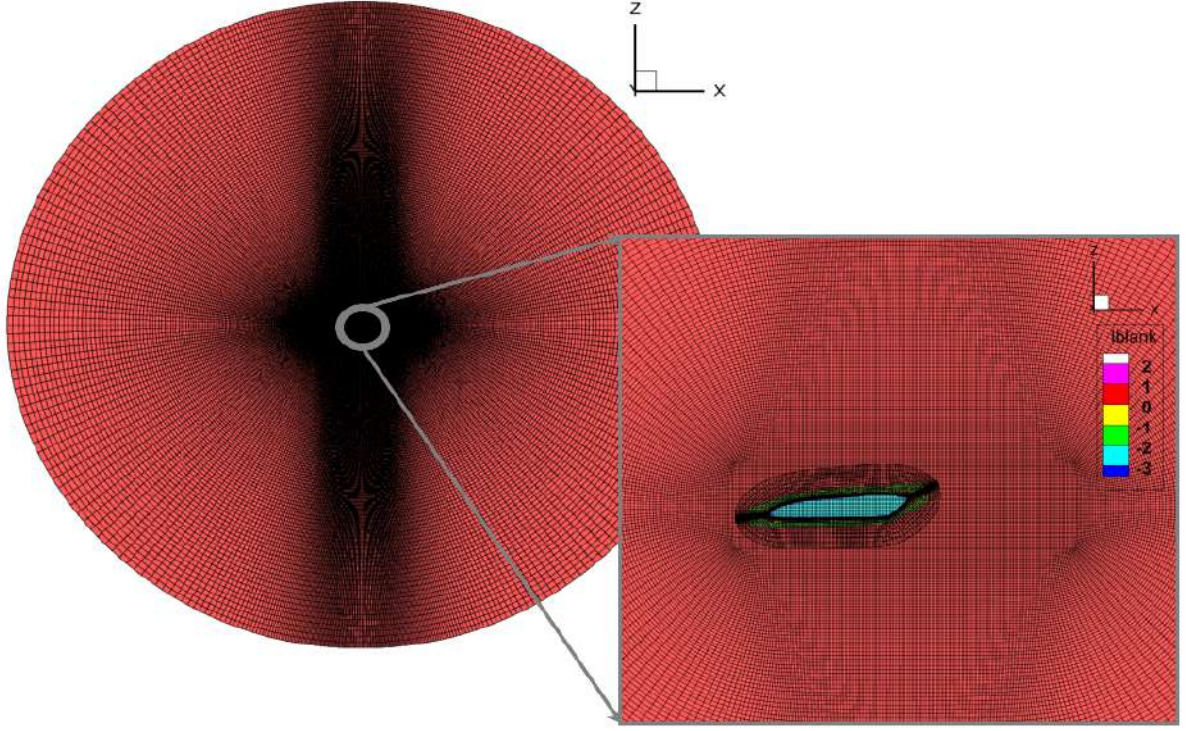


Figure 3.4: Zoomed-in view of overset grid near the fuselage within the farfield domain

A volume mesh qualifies as acceptable to ADflow if it contains no negative cell volumes. Besides that, ideally, the mesh should avoid cells that have poor quality in order to obtain a more accurate solution. Mesh quality is typically assessed using the normalized Jacobian determinant, which quantifies the deviation of each cell from an ideal hexahedral shape. In the mesh extrusion process, pyHyp provides real-time feedback in the terminal, indicating the minimum cell quality and a check for the presence of negative volumes. This information is critical for adjusting the mesh generation parameters to improve robustness and quality. Although pyHyp is a robust tool for mesh generation, a better tune of the parameters may facilitate this process.

To accurately resolve the boundary layer, the initial wall spacing (Δs) was estimated based on flat-plate turbulent boundary layer theory, targeting a non-dimensional wall distance of $y^+ \approx 1$ for use with the Spalart–Allmaras turbulence model [81]. The wall spacing was computed using

$$\Delta s = \frac{y^+ \mu}{\rho u_\tau}, \quad \text{where} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad \text{and} \quad \tau_w = \frac{1}{2} C_f \rho U_\infty^2. \quad (3.1)$$

The skin friction coefficient C_f was approximated using the empirical correlation for turbulent flow over a flat plate,

$$C_f = \frac{0.027}{Re_x^{1/7}}, \quad \text{with} \quad Re_x = \frac{\rho U_\infty x}{\mu}, \quad (3.2)$$

where, x corresponds to a characteristic fuselage length, which was equal to 6m in Chapter 5 and 4.4m in Chapter 6. U_∞ is the freestream velocity, which was set to 34.3m/s in both cases, since it is a typical cruise speed for ARX. The simulations were conducted at ground level (0m altitude), and ρ and μ represent the air density and dynamic viscosity under standard sea level conditions, since all simulations were conducted at ground level. In this study, an initial wall spacing of $\Delta s = 1.0 \times 10^{-6}$ m was applied in all volume mesh configurations. This resulted in a non-dimensional wall distance of $y^+ = 0.0842$ for the parametrisation cases and $y^+ = 0.0861$ for the final optimization case.

The growth ratio and number of layers were selected to produce a smooth mesh with high-quality cells. The mesh was continuously monitored during extrusion, using pyHyp's built-in quality metrics, to prevent negative volume cells and ensure suitability for CFD simulations in ADflow.

3.2.3 Overset Mesh

For more complex geometries, generating a high quality multiblock mesh is a very difficult exercise [82]. To mitigate this problem, an overset approach was applied. Basically, instead of having one complex mesh, this method allows to split the domain into separate and overlapping meshes. The information of these meshes is then interpolated in each iteration [83]. This process, as observed in Fig. 6.4, helps generating meshes with refined regions, such as near the trailing and leading edge of the fuselage. Another example of this implementation is shown in Fig. 3.4, where an example of a farfield mesh embedded in nearfield meshes can be observed.

To allow communication between these intersecting meshes, ADflow makes use of an Implicit Hole Cutting (IHC) algorithm [82, 84]. All the component meshes are merged into a single CGNS file. The IHC algorithm then assigns each cell in the domain a role based on where it is and what it is used for. These roles include compute cells (used in the flow solution), interpolate cells (which transfer information from donor blocks), blanked cells (excluded from computation but assist in defining overlap regions), and flooded cells (used to identify the interior of solid bodies) [84].

The classification of these cells is illustrated in Fig. 3.5. Red cells represent compute cells; green cells are interpolate cells, responsible for transferring flow variables from overlapping meshes; yellow cells are blanked cells, excluded from calculations but critical to controlling the flood process; light blue cells are flooded cells, which identify interior (non-physical) regions; and dark blue cells represent flood seeds, which initiate the flooding algorithm.

Flooding is a process used in the overset mesh process to identify interior regions of solid bodies, such as a fuselage, that should be excluded from the flow solution. It begins at specific cells, known as flood seeds, which are located near wall intersections on overlapping meshes. Then, compute cells are converted to flooded cells until the flooding is stopped by at least two layers of blanked or interpolated cells. This ensures that only the external flow field is resolved. For the flooding process to succeed, mesh resolution around the fuselage must be sufficient to prevent miss classification of cells that could

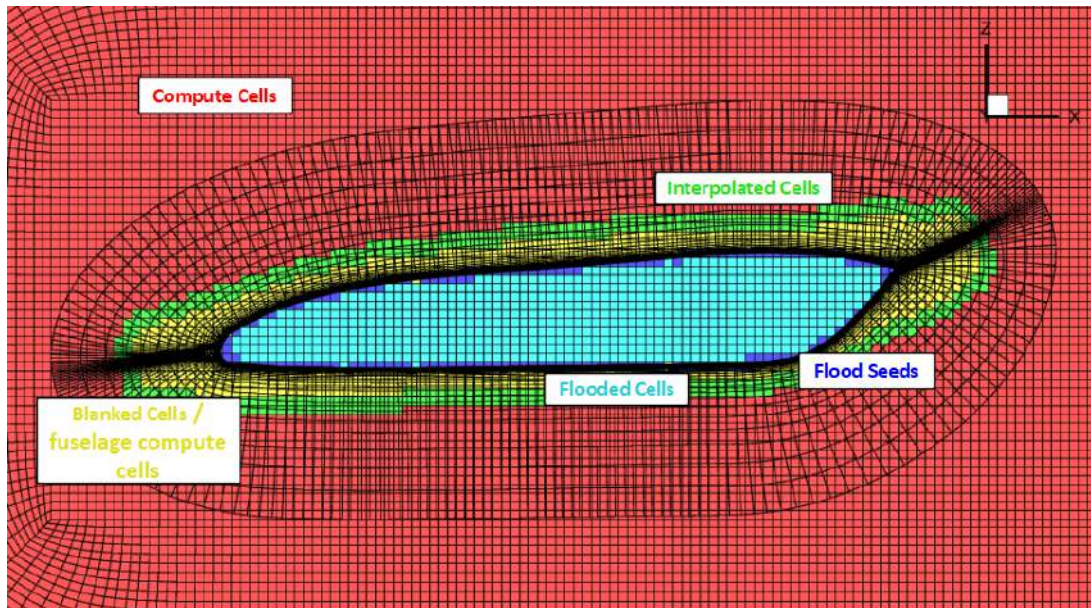


Figure 3.5: Overset mesh cell classification after implicit hole cutting.

compromise the solution.

Generally speaking, the overset mesh technique is a robust and flexible means of meshing complex geometries in CFD where local refinement or moving components are required. Breaking down mesh generation into small overlapping components and using techniques such as IHC and flooding to manage cell connectivity enables high-quality structured meshes to be used efficiently without the additional burden of constructing one conformal grid. This capability is required to achieve accurate and stable flow solutions in simulations using ADflow.

Chapter 4

Fuselage Shape Deformation Modeling

Shape deformation modeling is essential in aerodynamic optimization to allow accurate geometric adjustments to meet performance goals. The application of sophisticated computational methods for aerodynamic optimization is examined in this chapter, with an emphasis on shape deformation modeling.

4.1 Parametrisation Method Overview

There are several shape deformation techniques, such as basis vector, domain element, discrete, analytical, free-form deformation (FFD), partial differential equation (PDE), polynomial and spine, and computer-aided design (CAD) [85–87]. Among these, the free-form deformation and computer-aided design techniques stand out when compared to the others in efficiency and are overall more suitable for complex problems [88]. However, with today's tools, it still remains a challenge not only to parametrize a complex geometry with CAD systems but also produce models that are always suitable for automatic grid generation techniques.

The FFD method is a geometry modification technique that parametrizes the shape perturbation instead of the shape itself [76]. For example, Wentrup [89] explained that the control points of a volume were manipulated using the Free-Form Deformation (FFD) approach to change the shape of complex helicopter fuselage components. This allowed for smooth geometric changes throughout the optimization process. In the work developed by Kenway et al. [90], the Free-Form Deformation method is employed and used to perform shape deformation as well. Additionally, individual FFD boxes were used to modify the triangulated surface nodes and the structured surface meshes of each component based on the correspondent design variables [91]. More examples can be found in the work of Ekici and Juniper [92] and Zhao et al. [93]. This method allows externally supplied geometries in native formats, while simultaneously lowering the amount of design variables. Although the FFD technique offers unmatched flexibility in changing geometry, it does not provide direct control over certain aspects of the design. Nevertheless, it is still very useful for aerodynamic applications where considerable deformations are

necessary to obtain optimal performance because of its capacity to handle vast design space. Finally, the FFD method guarantees a compact representation of complex geometry, allowing for quick iterations and reliable solutions.

The FFD method serves as the parametrization approach in this thesis, and it is employed using tools from the open-source MACH-Aero optimization framework. Since FFD is already integrated into this framework, the implementation process is greatly facilitated.

4.2 Fuselage Deformation using FFD

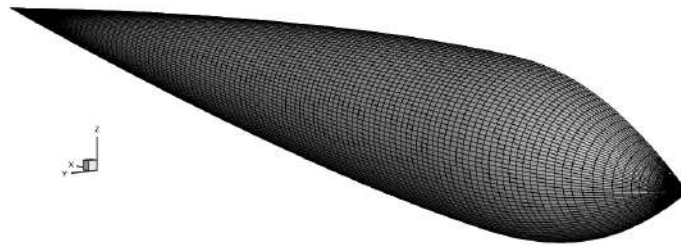


Figure 4.1: Baseline simplified fuselage mesh

A thorough overview of the fuselage and its computational mesh is given in Fig. 4.1, which shows the baseline geometry, before any change. This starting geometry acts as a benchmark, emphasising how the deformations affect the fuselage's aerodynamic performance as well as its shape. Figure 4.2 demonstrates the freedom of modeling the FFD box. The fuselage shape is directly changed by these FFD box adjustments, updating the 3D computational mesh. The FFD box works as a control structure that encloses the embedded fuselage, and any displacement applied to its control points results in a smooth deformation of the volume inside. Note that the nodes that are moved are located on the outer surface of the FFD box and are referred to as FFD control points. These nodes are mathematically linked to the mesh points inside the box. When the control points suffer any displacement, all the points within the FFD volume, including those on the fuselage surface, are automatically deformed based on their position relative to the box. By using the FFD method, the geometry of the object is embedded into a volume that can be manipulated by moving these outer control nodes, allowing for smooth and continuous shape changes. After receiving this morphed mesh, ADFlow is able to carry out the necessary CFD computations to evaluate aerodynamic performance for different configurations. These illustrations highlight the FFD method's adaptability and workflow integration in aerodynamic optimization [76, 88].

The resolution of an FFD box, driven by the number of its control points, defines the flexibility and accuracy of the deformation process. A coarse FFD box grid with a few control points efficiently captures global deformations. For projects where global optimization is to be performed or preliminary design studies, such a setting will be suitable, as it will enable large-scale geometry changes. This is demonstrated in Fig. 4.2b, in which a variation of one point of the FFD box alters a large portion of the fuselage. For cases where precise enhancements are not the goal, its simplicity ensures good modeling with a reduction of computational costs.

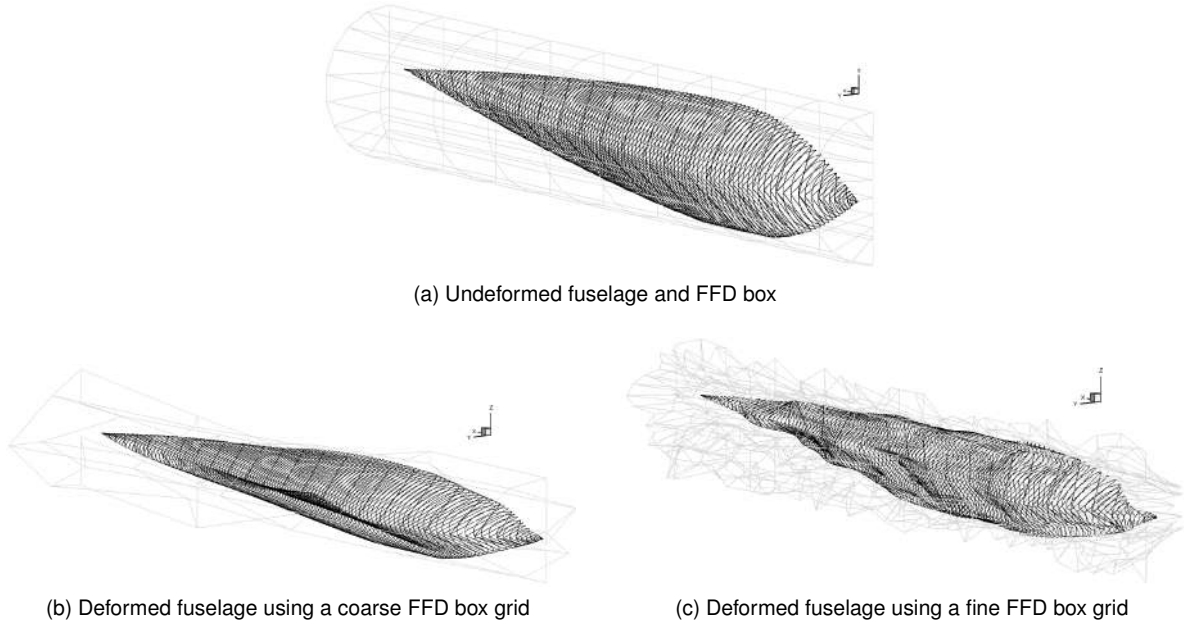


Figure 4.2: Effects of FFD box resolution

On the other hand, a fine FFD box grid allows more accurate and localised deformations. This increased resolution, as shown in Fig. 4.2c, is particularly useful when small-scale changes are needed. However, the computational cost is increased since the increased number of control points implies increased design space, which has implications in the computational cost of the optimization process. Strong management of the extra degrees of freedom is needed [89, 94].

As visible in Figs. 4.2b and 4.2c, there are different directions along which the fuselage may be deformed. Although the deformations are applied randomly in these two figures, there is a way to parametrize the control points of the FFD box to move in a specific direction.

The parametrization of the FFD box, and hence the fuselage, can be done by several approaches with different advantages, depending on the optimization problem. Figure 4.3 illustrates how the FFD box, and consequently the fuselage, can be deformed using three different approaches.

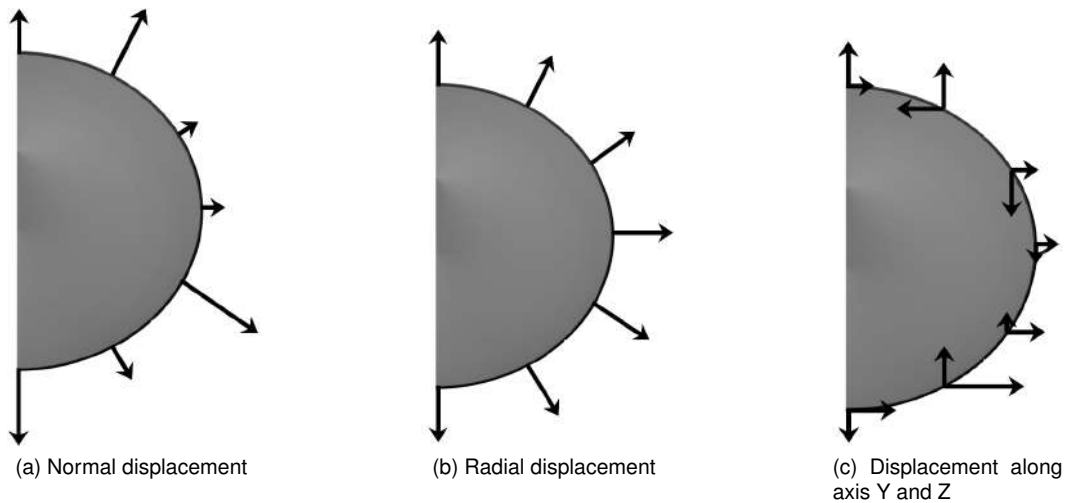


Figure 4.3: Different FFD control point displacement approaches

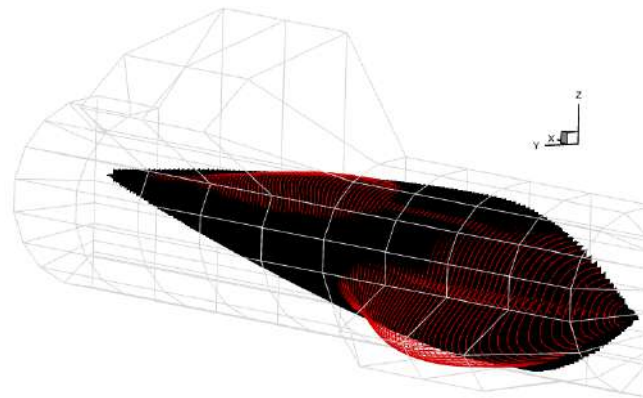
The first parametrization approach used in this work, Fig. 4.3a, utilizes the normal displacement of the FFD box points, such that the deformations are defined in different directions based on the geometry of the FFD box and fuselage, offering flexibility to the optimizer. This approach could be less intuitive for certain geometries, as the control points will move in different directions.

Additionally, radial displacements, Fig. 4.3b present an effective way to modify the fuselage shape. For this approach, it is essential to define a reference axis, which allows for the controlled scaling of different fuselage sections. By adjusting the fuselage's dimensions through scaling functions, this method makes it possible to expand or contract selected areas. However, for this study, it was not as effective since the goal was to reshape the geometry in study and not reduce its size [95].

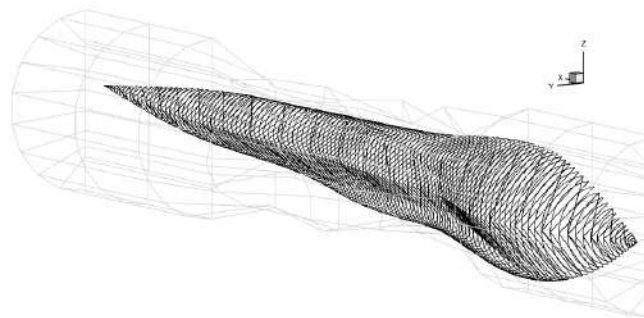
Lastly, there is the approach in which parameterizations along the X, Y, and Z axes are possible, as shown in Fig. 4.3c. This is a convenient approach to use when scaling, stretching, or compressing along the axes since it provides controlled shape variation in pre-determined directions. Limiting the deformations along the axes simplifies the optimization and provides the user with a clear understanding of the optimizer choices.

4.3 Deformation using Selected Points

It is possible to select specific FFD box control points to modify the geometry in study. More precisely, `addLocalSectionDV`, which is a method integrated in `pyGeo` library (Fig. 3.1), allows to generate local design variables for specific sections within the FFD box.



(a) Deformation using specific control points,



(b) Deformation using point selection

Figure 4.4: Different FFD control point strategies.

Figure 4.4a demonstrates how, given a range of acceptable perturbations and certain chosen control points, it is possible to provide local shape deformations along the normal direction of the surface. This may be seen, in the figure, for the black fuselage, where no changes take place, to the red one, which highlights the shape modification due to the application of the FFD method. This example shows the capability of re-shaping particular parts of the fuselage geometry by using a few specific control points of the FFD box.

In MACH-Aero documentation different information regarding parametrization of shapes can be found, in this case, adapted to fuselages [95]. One of the functions used is the Point Selection. This method is useful in a way that specifies points of the FFD box that can be manipulated during the optimization process.

Figure 4.4b demonstrates how this influences the deformations in the FFD box and, by consequence, the fuselage. This focused strategy minimises computational cost and concentrates optimization on crucial regions by guaranteeing that only important sections of the geometry are impacted.

4.4 Cambered Fuselages

The addition of fuselage camber is an aerodynamic modification that can influence lift, drag, and stability characteristics. Traditionally, the fuselage is designed to contribute little to lift, but the aerodynamic characteristics can be modified by adding curvature to its longitudinal axis. This section describes the developed capability of adding fuselage camber, which was implemented, in the optimization framework, and the potential aerodynamic influence that it may have.

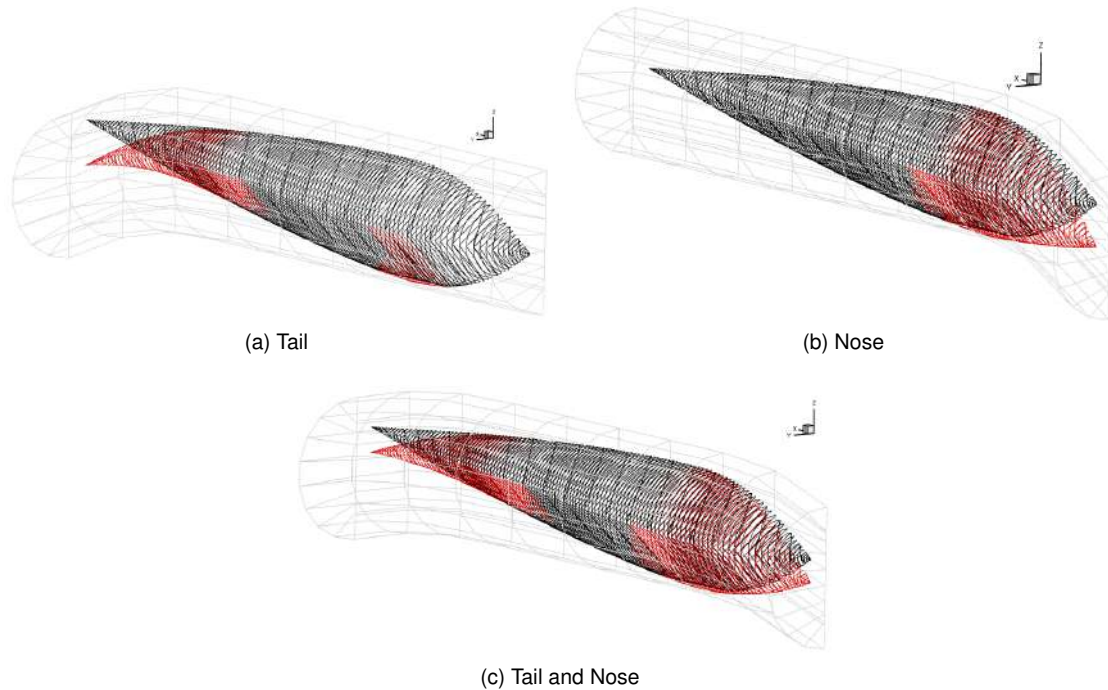


Figure 4.5: Camber parametrization for different parts of the fuselage

In the current developed framework, the introduction of nose camber, tail camber or any combination

of them may be introduced in it, or even to describe regions where only a limited range of camber variations is applied, such as the wing root. These configurations can be seen, respectively, in Figs. 4.5a, 4.5b and 4.5c. Even though this is exaggerated to emphasize fuselage parametrisation, a quadratic function, Fig. 4.6, is used in the optimization problems to introduce camber, enabling the independent upwards or downwards adjustment of both the nose and the tail in search of different aerodynamic configurations.

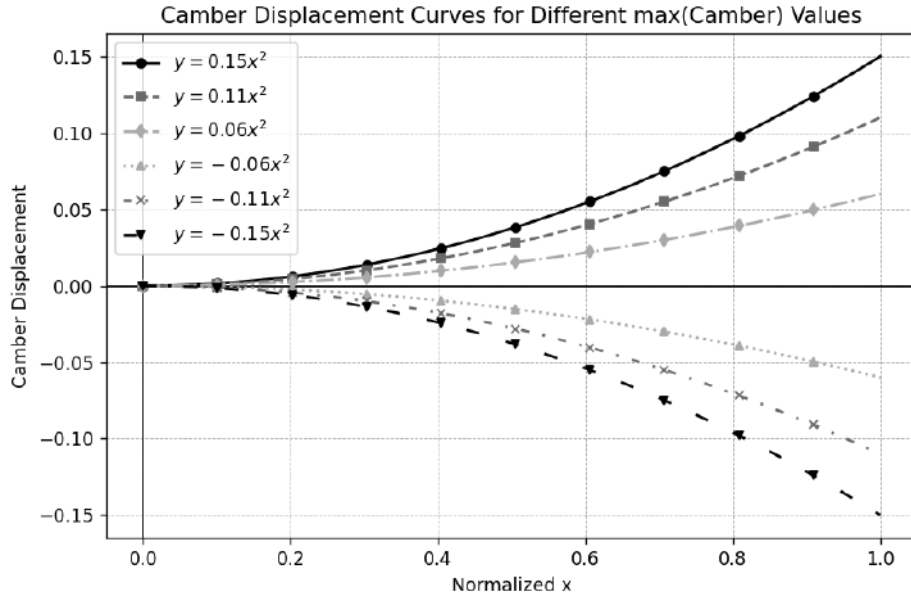


Figure 4.6: Camber displacement curve

A cambered fuselage can potentially generate positive lift at zero angle of attack when compared to a symmetrical fuselage [96, 97]. An increase in lift, with a relatively small increase in drag, results from the introduction of camber in the nose section. However, if camber is introduced into the tail section, it primarily influences lift and drag, but somewhat more strongly the effects of flow separation. Both nose and tail sections may simultaneously be subjected to camber, the result being variations in the fuselage aerodynamic forces. Besides lift and drag variations, camber has effect on stability and pitching moment. Nose camber introduces a nose-up pitching moment that can affect trim control. A cambered tail section adds to the changes in aerodynamic stability but also can introduce additional trim requirements. Adding nose camber together with a cambered tail section in the opposite direction can shift the position of the aerodynamic center and require compensatory trim changes.

It is important to note that adding camber to a fuselage influences wing-fuselage interaction because of the changes in the pressure distribution around the fuselage. Therefore, the nature of flow at the wing-fuselage junction can be affected. With the current developed framework, it is possible to parametrize a fuselage by zones. This way, problems like these might be avoidable by selectively excluding camber modifications in critical areas where aerodynamic disturbances could negatively impact performance. This kind of zone control permits the implementation of camber in regions where it is beneficial, like improved lift characteristics or stability adjustments, while making sure that areas like the wing-fuselage junction remain aerodynamically smooth.

4.5 Fuselage Constraints

The creation of high-fidelity multidisciplinary design optimization (MDO) frameworks necessitated more complex geometric constraints. In particular, ensuring sufficient internal volume for systems such as payloads or energy systems while maintaining aerodynamic efficiency was one of the main challenges. Classical constraint techniques, based on fixed points or prescribed volumes, are not the most adequate techniques to control the deforming geometries that are typical in shape optimizations [98]. This motivated the development of the triangulated surface constraints.

To effectively respond to these challenges, a geometric constraint method was needed that satisfied three essential criteria. Firstly, it had to be general, capable of accurately representing arbitrary surfaces, including complex and locally concave shapes, without relying on convexity assumptions. Secondly, the constraint metrics had to be differentiable so that gradient-based optimizers could efficiently use the information. Finally, the method needed to be efficient, scalable to handle large numbers of design variables and surface polygons without excessive computational cost.

To understand how the triangulated surface constraint works, it is important to first define the concept of containment between surfaces. Consider a component represented by surface A, that needs to stay within a bounding surface B. Using Fig. 4.11 as an example, where surface B, the fuselage, represents the outer mould, a closed, connected, and orientable surface defined in three-dimensional real space; while surface A, represented by the sphere, is a connected surface within this space. The simplest way to check if A is a subset of B would be to ensure that every point of A is within B. However, this is not computationally possible for an infinite number of points. To address this, the definition is made more pragmatic: it only needs to check that at least one point P from surface A is inside B, and also check that surfaces A and B do not intersect. But even this causes problems with gradient-based optimization, since it provides a simple yes or no without providing any useful gradient information. The problem is that, without the gradient, there is no way for the optimizer to tell in which direction it should move to fix a geometry break.

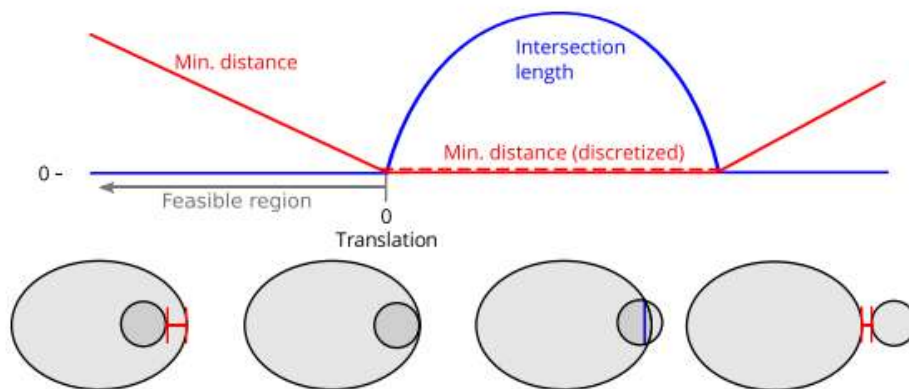


Figure 4.7: Minimum distance between the surfaces [98].

To overcome this problem, the method focuses on the minimum distance between the two surfaces, d_{min} , shown in Fig. 4.7, which must be greater than zero, while monitoring the length of any intersection

curve between the sphere and the fuselage, L , which must be equal to zero. Note that when intersections occur, this second parameter L is used to guide optimizer back into the feasible space and not stuck. However, if the optimizer takes excessively large steps into the infeasible region, it may still become trapped, as the local gradient information may no longer be reliable [99]. To avoid this risk, care must be taken while controlling the step sizes within the optimization so that progress is made while within the space where linearization of constraints exists and recovery is feasible. With this, the triangulated surface constraint ensures geometric feasibility by enforcing $d_{min} > 0$, $L = 0$, and keeping at least one point P on A inside B while avoiding large steps into infeasible regions.

In order to calculate the parameters introduced above, an efficient and compact representation of the surface geometry is essential. A triangulated surface model is employed, since this offers a favorable trade-off between generality, efficiency, and compatibility with typical CFD mesh topologies. Specifically, many CFD grids natively support triangular facets, particularly when based on tetrahedral elements, and structured grids are trivially triangulated surfaces [100]. Also, parametric surfaces such as splines can be easily triangulated, so the technique can be used anywhere irrespective of the geometry type.

The calculation of the minimum distance between two triangulated surfaces is a combination of point-triangle and edge-edge distance tests [98]. By examining this approach in more detail, there may be an infinite equally close points between two triangles, which would create multiple d_{min} , therefore, constraining only this parameter could generate bad results [101]. This supports the claim made by Brelje et al. [98].

Additionally, while theoretically it should be possible to use segment-triangle distance tests to calculate the nearest points, they are less efficient and computationally expensive [101]. Instead, it is well-established that the global minimum distance between two triangles will always fall either between an edge of each or between a vertex of one and a point projected inside the other triangle. Hence, the problem is reduced to computing the distances between all pairs of edges illustrated in Fig. 4.8a, one from each triangle, and the closest point on the opposite triangle for each vertex of each triangle, exemplified in Fig. 4.8b (this point is meant to be inside the triangle). This leads to six vertex-triangle tests and nine edge-to-edge. The nearest of those is the global minimum distance between the two triangles and it is a computationally efficient and exhaustive solution.

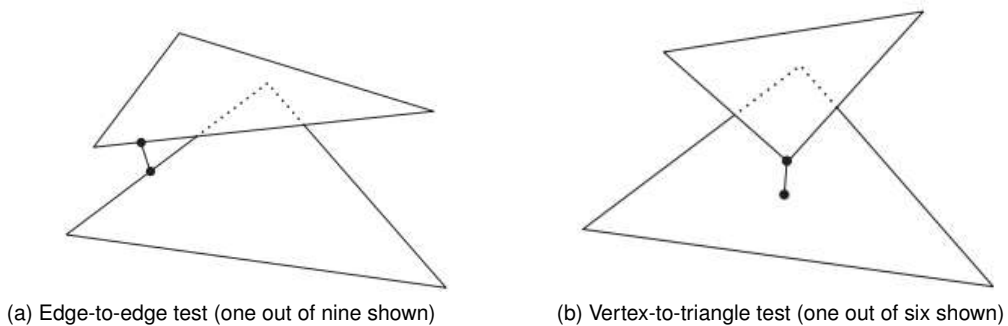


Figure 4.8: Calculate the closest pair of points, [101]

After calculating the different distances, there is a need to aggregate these results into one single

constraint. Tracking each individual distance constraint separately would be computationally infeasible due to the high number of triangle pairs involved, especially in complex geometries with thousands of facets [98]. A widely adopted approach is to use constraint aggregation functions, which combine multiple constraints into one smooth and differentiable expression. Each aggregation have the form [56],

$$\bar{g}(x) \equiv \bar{g}(g(x)) \leq 0, \quad (4.1)$$

where $\bar{g}$ is a scalar, and g is the vector of constraints we want to aggregate. One of the properties we want for the aggregation function is that if any of the original constraints are violated, then $\bar{g} > 0$.

One way to aggregate constraints would be to define the aggregated constraint function as the maximum of all constraints,

$$\bar{g}(x) = \max(g(x)). \quad (4.2)$$

If $\max(g(x)) \leq 0$, then we know that all components of $g(x) \leq 0$. However, the maximum function is not differentiable, making it unsuitable for gradient-based optimization.

The Kreisselmeier–Steinhauser (KS) function is one of the most established and widely adopted methods in gradient-based optimization problems. The following two references present optimization problems that use gradient-based methods, where constraint aggregation is performed using the KS function [102, 103]. It is a differentiable function, which solves the problem explained previously, and it can be defined as

$$\text{KS}[g(x)] = g_{\max}(x) + \frac{1}{\rho} \ln \left(\sum_{j=1}^m e^{\rho(g_j(x) - g_{\max}(x))} \right), \quad (4.3)$$

where ρ is a constant that controls the behaviour of the aggregation. Specifically for spatial integration, $g(x)$ corresponds to the negative of the distance vector, $g(x) = -d_j(x)$, where $d_j(x)$ contains the calculated distances between all pairs of facets from surfaces A and B [98]. Using this relationship, the final form of the aggregated distance constraint for optimization can be written as

$$\text{KS}_{\text{geom}}(x) = \frac{1}{\rho} \ln \left(\sum_{j=1}^J e^{\rho(d_{\max}(x) - d_j(x))} \right) - d_{\min}(x) \leq 0. \quad (4.4)$$

The KS function provides a conservative estimate of the maximum value among its inputs, while maintaining smoothness and retaining useful derivative information from constraints that are close to being active. As $\rho \rightarrow \infty$, $\text{KS}_{\text{geom}}(x) \rightarrow -d_{\min}$. This means that, by adjusting ρ , the spatial tightness of the constraint approximation can be controlled. Nevertheless, caution must be taken since, as ρ increases, the curvature of $\bar{g}$ increases, which can cause poorly conditioned optimization problems.

For the second constraint, the method used by Brelje et al. [98] to calculate the intersection length is the Möller method [104]. The objective of this type of algorithms is to detect collisions by calculating the primitive-primitive intersection. The Möller method determines whether two triangles intersect or not and how the intersection is calculated. Firstly, and as shown in Fig. 4.9, Möller computes the equations of the planes π_1 and π_2 , in which the two triangles T_1 and T_2 lie.

The plane equation π_2 is equivalent to

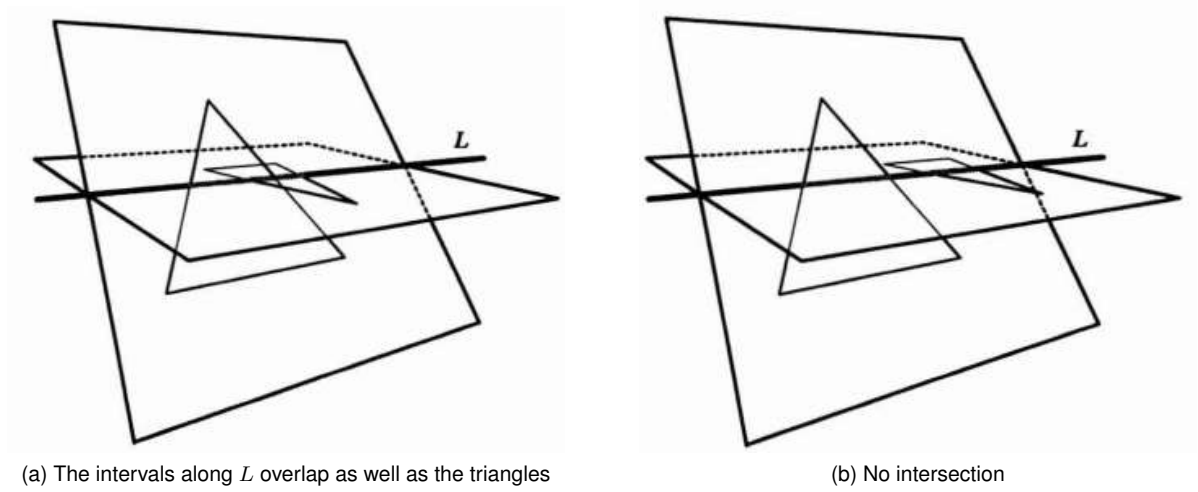


Figure 4.9: Intersected triangles test [104].

$$N_2 \cdot X + d_2 = 0 \quad (4.5)$$

where X represents any point on the plane. In more detail, the parameters are computed as

$$N_2 = (V_1^2 - V_0^2) \times (V_2^2 - V_0^2)$$

$$d_2 = -N_2 \cdot V_0^2.$$

Then, to calculate the distances from the vertices of T_1 into π_2 multiplied by the constant $|N_2|$, Eq.(4.5) is used by replacing the vertices coordinates, taking the form

$$d_{v_i^1} = N_2 \cdot V_i^1 + d_2, \quad i = 0, 1, 2. \quad (4.6)$$

If all the distances are different to zero, meaning that no point is on the plane, and the sign is the same for all the coordinates, then T_1 lies on one side of π_2 and the overlap is rejected. Therefore, if all the distances are something other than zero, meaning that, no point lies on the plane, and the signs are identical for all the coordinates, then T_1 lies on one side of π_2 and the intersection is rejected. The same is done to the other combination of triangle and plane. These two tests will reduce the computational time, as some triangles will automatically be rejected. Note that for a pair to pass in these tests there must be a line of direction $N_1 \times N_2$ that meets both.

By computing the distances using Eq.(4.6), it is established if there are intersected triangles or not. This happens when there are values different to zero. Then, as shown in Fig. 4.10, the Möller method proceeds by projecting the triangle vertices onto the line L , this is computed as

$$p_{v_i^1} = D \cdot (V_i^1 - O). \quad (4.7)$$

To compute the intersection interval along L for T_1 , one considers the signed distances $d_{v_i^1}$ from the vertices of T_1 to the plane π_2 of T_2 . Using geometric similarity, the scalar parameter t_1 representing the point of intersection between the triangle edge and the line is given by

$$t_1 = p_{v_0} + (p_{v_1} - p_{v_0}) \frac{d_{v_0}}{d_{v_0} - d_{v_1}}. \quad (4.8)$$

Similar calculations are performed to determine t_2 , completing the interval for T_2 . If the intersection intervals of both triangles along L overlap, the algorithm concludes that the triangles intersect.

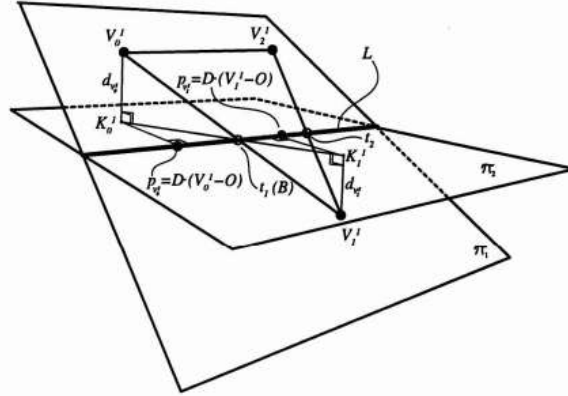


Figure 4.10: Projections of triangle vertices onto the intersection line L [104].

In the case where the result of all $d_{v_i^1}$ equal zero, the triangles are co-planar. This results onto a projection in the axis-aligned plane where the areas of the triangles are maximize. This simplifies the issue into a two-dimensional triangle-triangle intersection test. The algorithm begins by calculating edge intersections of the two triangles. If no intersections are found, a point-in-triangle test [105] is used to determine whether one triangle is entirely contained within the other.

All these geometric computations are of help in forming the basis for the construction of the constraint. They characterize the spatial relationships among the triangles and simplify the problem if feasible, and thus make the constraint logic easier to understand and apply. All this gets used in the triangulated surface constraint that will be implemented into a fuselage optimization problem taking into account an internal surface. Geometric constraints are an essential component in aerodynamic shape optimization, especially for internal parts or strict spatial constraints. In their absence, optimization algorithms will employ the maximum flexibility of the geometry to improve performance at the cost of physical feasibility. Lack of geometric constraints can lead to component interference, unrealistic distortion, or even loss of key internal spaces.

This section outlines the geometric constraints employed in this work to ensure, as much as possible, realistic optimized fuselage geometries when trying to aerodynamically optimize. Two different methods have been employed: the first employs standard volume constraints to ensure internal volumes, and the second employs a more advanced triangulated surface constraint that provides precise control of clearances and avoids geometric intersections between surfaces.

After considering the information in Sec. 4.1, it must be mentioned that pyGeo also has a submod-

ule called DVConstraints which derives nonlinear geometric constraints such as thickness and volume constraints from surface mesh geometry. For the purposes of this study, the triangulated surface constraints method within DVConstraints was used to restrict and create the different optimization problems discussed in Sec. 5.

Although not used in this work, it is important to describe how the volume constraints work to better understand its differences compared to the triangulated surface constraints. The `addVolumeConstraints` allows to set a volume-based constraint to the geometry for optimization problems. It avoids deformations from reducing a specified limit below the internal or external volume of a particular section. This section may be defined using the `PointSelection` method which allows to group together the desired points. In aerodynamic applications, volume constraints are extremely helpful in order to maintain internal space requirement, or passage of flow aerodynamically. By imposing bounds and a resolution grid, the component monitors the bounded volume during the optimization to prevent over-deformation, which could compromise aerodynamic performance or manufacturability. This restriction may be used in an optimization problem to create sufficient internal volume for the fuel tank or a radar while allowing shape adjustment to attain improved aerodynamic efficiency.

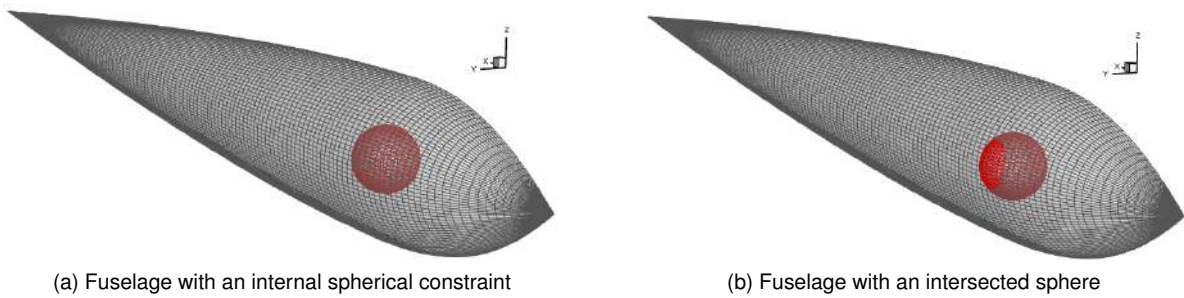


Figure 4.11: Triangulated surface constraint used in the optimization problem

However, the focus was placed on a more advanced approach, the Triangulated Surface Constraint method. In this thesis, the approach used is rather similar to the one in [98] where, instead of imposing volume constraints, this method uses surface integration measures to enforce spatial limitations, which gives more freedom to the optimizer. The Triangulated Surface Constraint method imposes constraints based on two key parameters: the minimum distance between surfaces and the total length of their intersection curves. What makes this approach particularly effective is its ability to use triangulated STL files to represent the bounding geometry. This method is also advantageous because it provides the optimizer with gradient information even in cases where surfaces come into contact, something that volume constraints alone cannot fully manage. In this case, as shown in Fig. 4.11, a sphere was defined as the bounding surface, allowing the optimizer to control clearances accurately while still exploring aerodynamic improvements.

In Chap. 5, it will be shown how these constraints were applied. Several optimization cases will be implemented and analysed to explore the different parametrization approaches discussed in this chapter (Chap. 4). These cases will also illustrate the practical implementation and effects of the triangulated surface constraint, providing insight into its role in shaping the final fuselage in Chap. 6.

Chapter 5

Study Cases Using Different Parametrization Approaches

Following the parametrization approaches and constraints established earlier, several figures will be presented to illustrate the main differences between the various techniques applied. The goal is to evaluate each approach and compare their impact on the fuselage shape, ultimately guiding the selection of the most suitable methods for the final optimization studies.

As outlined, each parametrization approach was tested independently to allow for an isolated assessment of its influence on the fuselage geometry. This study considered a range of scenarios, beginning with simpler applications and progressively introducing additional techniques to build more complex cases as the analysis evolved.

Across all cases, the objective remains the same: minimizing the fuselage drag coefficient. For this purpose, the baseline fuselage, shown in Fig. 4.1, serves as the reference geometry for the aerodynamic optimization. The optimization problem is formulated as

$$\begin{aligned} & \underset{x}{\text{minimize}} && C_D(x) \\ & \text{subject to} && \text{KS}_{\text{geom}} \leq 0 \end{aligned}, \quad (5.1)$$

where x represents the vector of design variables (DVs), including the coordinates of the FFD control points and/or the amplitude of the camber function.

In the initial stages of the study, no geometric constraints are imposed, allowing a clear observation of the effects of each parametrization approach on the fuselage shape. As the study progresses, additional constraints are introduced, in particular, the geometric constraint $\text{KS}_{\text{geom}}(x)$, which aggregates surface deformation limits using the Kreisselmeier–Steinhauser (KS) function. This provides a smooth and conservative approximation of the maximum deviation between triangulated surfaces [98].

All optimization cases were performed in the MACH-Aero framework [72], using ADFlow [53] to evaluate the aerodynamic performance. The gradient-based SLSQP algorithm was used with a convergence tolerance of 10^{-6} and with a maximum of 1000 iterations.

The fuselage operating condition represented a flight at a speed of 34 m/s at Standard Sea Level, which is a typical cruise speed for ARX, with zero angle of attack (flow aligned with X-axis).

The different deformation approaches discussed in Chap. 4 are evaluated and their effectiveness compared. The strategies individual considered are normal-based deformations (Fig. 5.2a), directional deformations along a specific axis Y (Fig. 5.2b), Z (Fig. 5.2c) or both (Fig. 5.2d), and cambered fuselage deformations (Figs. 5.3a, 5.3b and 5.3c). The combined camber and normal vectors (Fig. 5.3d) or Y-/Z-directions (Fig. 5.3e) are also studied. Lastly, the effect of volume constraints in the optimal fuselage shape is also studied for both internal and protruding objects (Fig. 5.4). The detailed discussion of all results is included in Sec. 5.5.

5.1 Selected FFD Points

The first set of studies included the normal-based, and the (Y- and Z-) direction-based deformation parametrization approaches. Each approach required specific considerations to ensure the stability of the optimization process, particularly to prevent the generation of invalid geometries, such as negative volumes, which would cause the mesh to fail and compromise the optimization. The resulting optimized fuselage shapes are illustrated in Fig. 5.2.

For the normal-based deformations, Fig. 5.2a, it was necessary to carefully select which regions of the FFD box could be parametrized. Points located near the symmetry plane ($y = 0$) were excluded, as their displacements could easily produce negative CFD mesh volumes and, subsequently, make the optimization process to fail. Additionally, the points near the edges of the fuselage, specifically at the nose and tail, were also frozen to prevent geometric inconsistencies at the edges, avoiding similar problems. Furthermore, the freedom given to the FFD points was kept relatively small to avoid, again, the risk of generating invalid mesh volumes. These limitations directly impacted the optimization results, as seen in the corresponding figures and data summarized in Tab. 5.1.

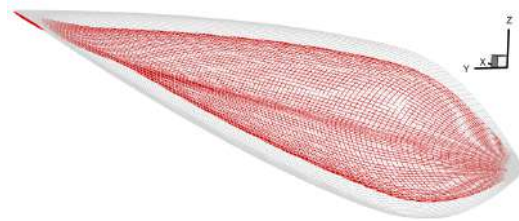


Figure 5.1: Irregularities encountered using the directional deformation method

In contrast, the directional deformation approach, which deforms the geometry along the Y- and Z-axis, was generally easier to control. As expected, the freedom given to the FFD points could be increased without immediately risking mesh failures in these cases. However, this approach tended to excessively deform certain regions of the fuselage, particularly near the tail, creating irregular geometries, almost like a twist or warp in these regions, as observed in Fig. 5.1. Similar to the normal-based strategy, the points at the fuselage leading and trailing edges were kept fixed to avoid negative mesh volumes, which limited the flexibility of the surrounding regions, thus aggravating the localized deformations

observed.

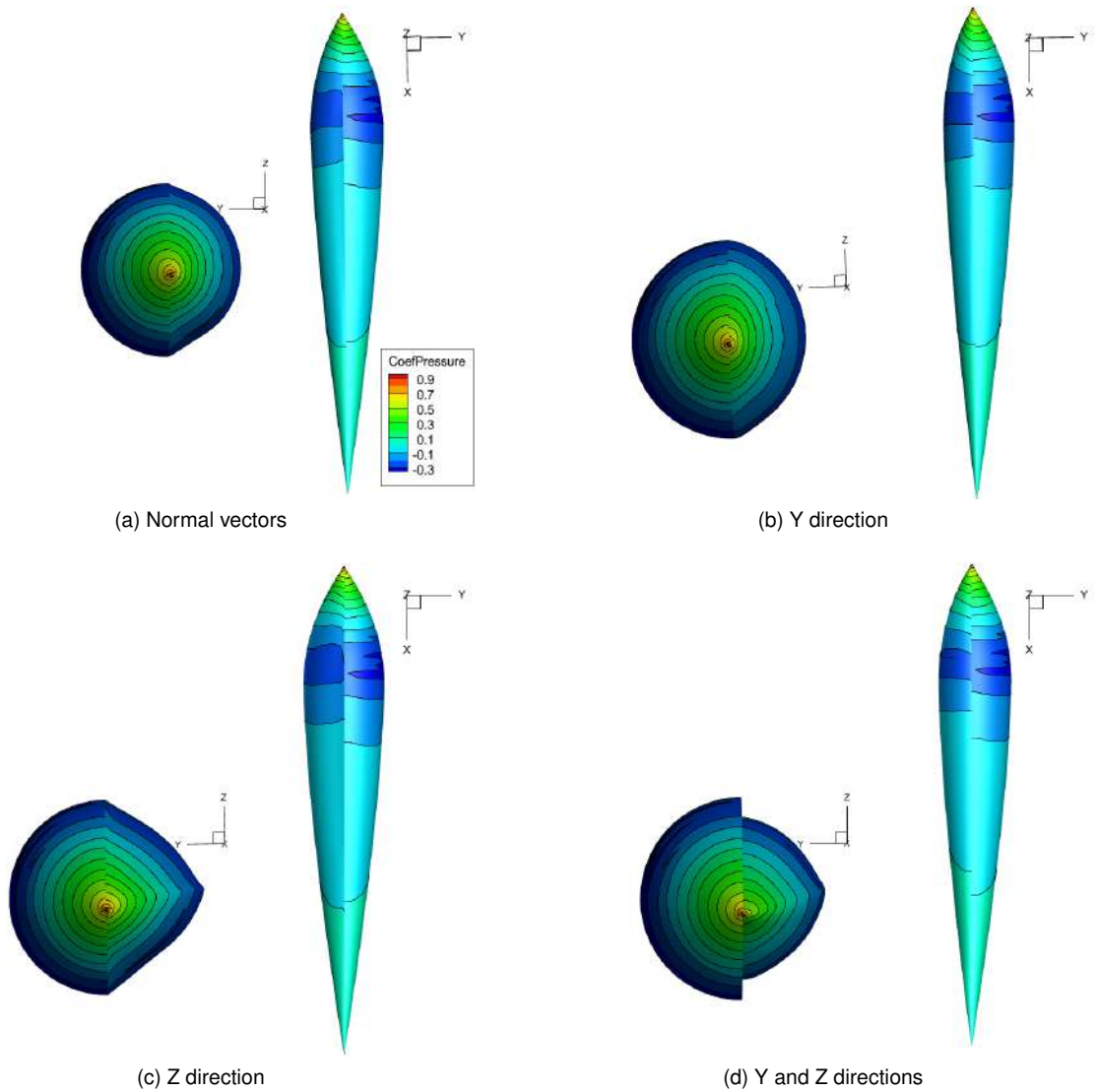
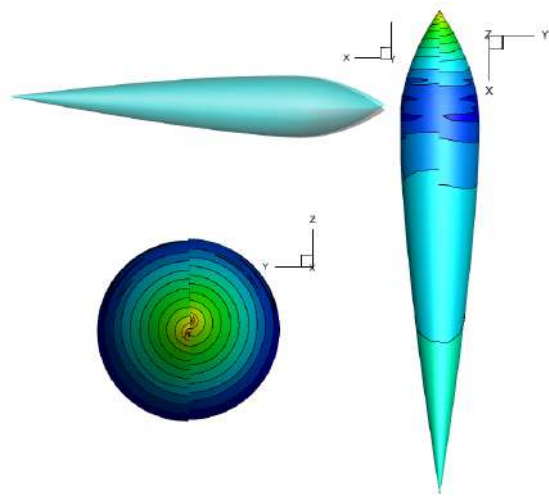


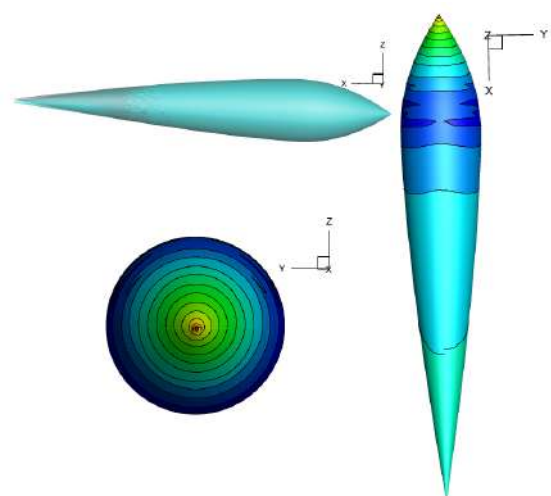
Figure 5.2: Comparison of resulting optimal shapes using different parametrization approaches (half-left is baseline, half-right is optimum).

5.2 Camber Function

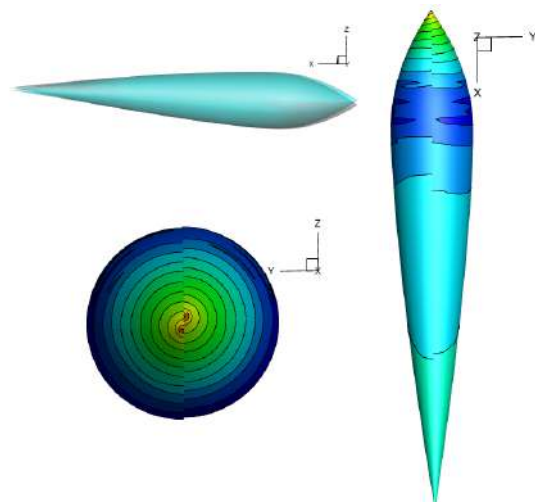
The second set of studies was conducted to identify the benefits of adding camber to the fuselage. The precautions when setting up these cases were primarily related to avoid creating mismatches in the intersection regions with components such as wings or tails. To address this issue, freedom was given only at specific longitudinal (X-axis) intervals that excluded intersection regions to prevent the aforementioned problems. Three different cases were tested, applying the camber function only in the nose region, tail region, or both, which produced the optimal shapes shown in Fig. 5.3.



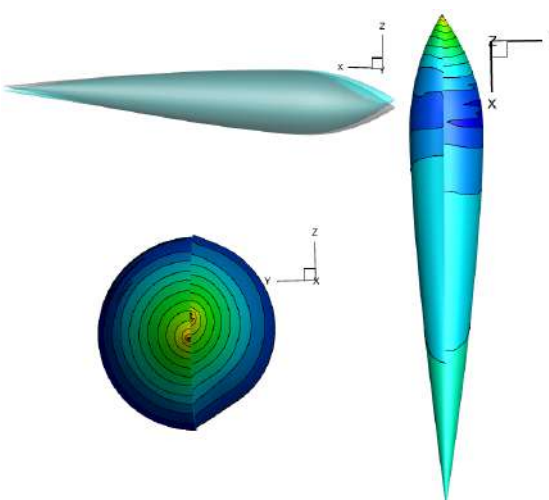
(a) Restricted to nose.



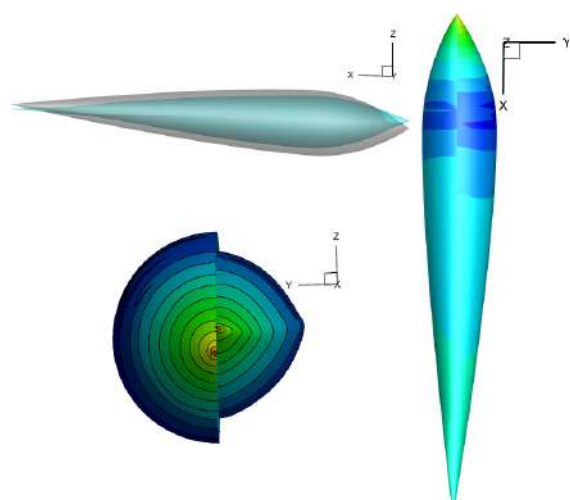
(b) Restricted to tail.



(c) Restricted to nose and tail.



(d) Camber combined with normal vectors.



(e) Camber combined with Y and Z directions.

Figure 5.3: Optimal shapes with camber deformation approach (half-left is baseline, half-right is optimum).

5.3 Combined Deformation Strategies

The combination of the camber function with the normal- or direction-based strategy was also studied. The resulting fuselage shapes are shown in Figs. 5.3d and 5.3e, respectively. These two optimization cases differ significantly in both deformation strategy and resulting geometry. In Fig. 5.3d deformations along the normal directions of the FFD box were employed, resulting in a smooth and rounded fuselage shape that remains close to the baseline geometry. This is partly due to the constraint that points along the $y = 0$ axis were fixed, limiting symmetry-plane movement and preventing sharp geometric changes. In contrast, Fig. 5.3e involved displacements along the global Y and Z axes, with no restrictions on the $y = 0$ axis, allowing more aggressive shape modifications. As a result, the fuselage in this last optimization case appears more abruptly deformed and significantly more shrunk, particularly around the midsection. These differences highlight how the choice of deformation direction and axis constraints can strongly influence the smoothness and overall geometry of the optimized fuselage.

5.4 Volume Constraints

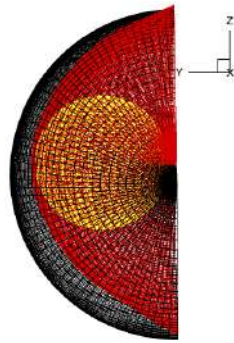
To complete the study, the Triangulated Surface Constraint was tested with the same different shape deformation strategies. The same 3D object, a sphere, was used in these cases to represent any given payload. Two different object locations were experimented: the object inside the baseline fuselage and partially outside, protruding the baseline fuselage shape.

The cross-section of the resulting optimal fuselage shapes obtained with the different deformation strategies, passing through the object, are illustrated in Fig. 5.4. The solutions to the problems with the object inside the baseline fuselage include: combined camber and normal-based deformations (Fig. 5.4a), combined camber and deformations along the Y- and Z-axes (Fig. 5.4b), and deformations only along normal-vectors of the FFD box (Fig. 5.4c). In addition, the cases with the object initially protruding the baseline fuselage are shown in Figs. 5.4d and 5.4e, using normal-vectors, and combined with z-axis deformation, respectively.

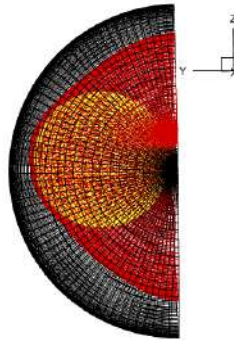
As attested in Fig. 5.4, each of the presented cases made the optimizer manipulate the FFD control points in different ways to ensure that the fuselage could envelop the payload (sphere).

The main differences encountered between the cases shown in Figs. 5.4d and 5.4e arise from the variation in the selected points. While deforming the shape using the normal vectors of the FFD box points, it was necessary to exclude points near the plane of symmetry ($y = 0$) to prevent mesh issues and negative volumes, which caused the differences. To mitigate this problem, it was given freedom to these points along the Z axis in the second case, which visibly improved the design, as the fuselage was able to shrink near the symmetry plane, resulting in a smoother and more effective deformation.

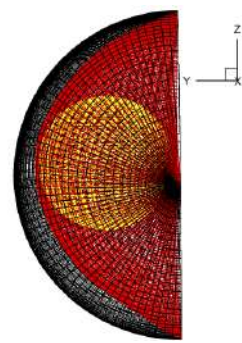
Finally, the differences between parameterizations with and without the triangulated surface constraint can be observed in Fig. 5.5, where two different cases are illustrated: combined camber and normal-based deformations (Fig. 5.5a) and camber with deformations along the Y and Z axes (Fig. 5.5b), both showing optimized fuselages taking (red) or not (black) into consideration the triangulated surface



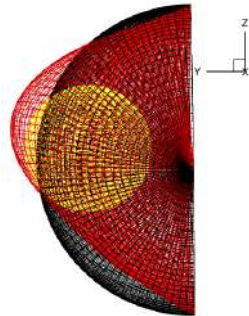
(a) Camber + normal vectors deformations with sphere inside fuselage.



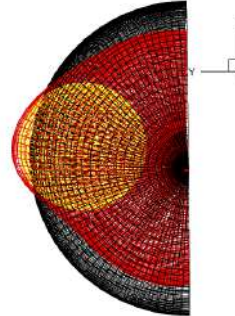
(b) Camber + (Y and Z) deformations with sphere inside fuselage.



(c) Parametrization using normals of FFD points with sphere inside fuselage.

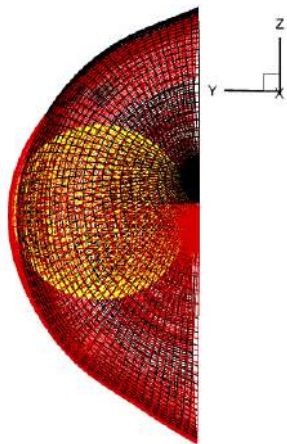


(d) Parametrization using normals of FFD points.

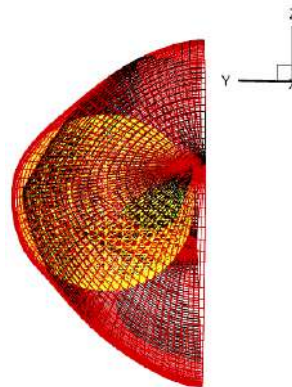


(e) Parametrization along Z-axis, combined with normal vectors of FFD points.

Figure 5.4: Optimal shapes using the triangulated surface constraint.



(a) Camber + normal vectors deformations.



(b) Camber + (Y and Z) deformations.

Figure 5.5: Comparison of optimal shapes with (red) or without (black) triangulated surface constraint.

constraint. It is clear that, despite using the same parameter values, the resulting shapes differ significantly. It is also worth noting that, for the optimized fuselages (black), the optimal design variable values reached their imposed lower bounds, which suggests that the optimizer could have further reduced the fuselage size in the absence of these.

5.5 Discussion of Results

The drag reduction obtained for each of the different deformation strategies studied in this Chap. 5 is listed in Tab. 5.1.

Table 5.1: Optimization deformation strategy studies and corresponding drag coefficient reduction.

Parametrization strategy	Constraints	Opt. shape	Min C_D	ΔC_D
(baseline)			0.0189	ref
Normals	None	Fig.5.2a	0.0161	-16.6%
Y-axis	None	Fig.5.2b	0.0155	-21.6%
Z-axis	None	Fig.5.2c	0.0168	-11.7%
(Y, Z)-axis	None	Fig.5.2d	0.0114	-40.4%
Nose Camber	None	Fig.5.3a	0.0185	-2.1%
Tail Camber	None	Fig.5.3b	0.0185	-2.1%
(Tail, Nose) Camber	None	Fig.5.3c	0.0184	-2.3%
Camber + Normals	None	Fig.5.3d	0.0161	-17.1%
Camber + (Y, Z)-axis	None	Fig.5.3e	0.0133	-41.6%
Camber + Normals	TS (IS)	Fig.5.4a	0.0162	-16.9%
Camber + (Y, Z)-axis	TS (IS)	Fig.5.4b	0.0135	-39.8%
Normals	TS (IS)	Fig.5.4c	0.0162	-16.4%
Normals	TS (IntS)	Fig.5.4d	0.0176	-7.0%
Normals + Z-axis	TS (IntS)	Fig.5.4e	0.0155	-21.3%

TS = Triangulated Surface, IS = Internal Sphere, IntS = Intersected Sphere.

Through the analysis of optimal shapes presented in Figs. 5.2, 5.3, 5.4 and 5.5, and the aerodynamic performance gains summarized in Tab. 5.1, it is possible to assess about the effectiveness of the different shape parametrization approaches proposed. Notice, however, that these optimal shapes were obtained from problems that differed slightly in terms of design variable bounds to overcome mesh morphing fails, as such, the comparisons might not extrapolate directly to other fuselage geometries.

In the cases with the deformation parametrization along the normal vectors of the FFD box points, the optimized shapes exhibited fewer irregularities and a more stable CFD mesh morphing, leading to a nearly 17% drag reduction for either the unconstrained and internal volume constrained cases, but less an expressive 7% reduction in the protruding payload case. However, this approach revealed to be relatively limited and not well adapted to arbitrary fuselage shapes.

Looking at the approaches using deformations along the Y- and Z- directions, they exhibited a much greater drag reduction compared to the normal direction approach. The Y-direction only approach, corresponding to the lateral fuselage deformations led to considerably better shapes compared to the Z-direction only approach, and, unsurprisingly given the greater deformation freedom, the combination of both Y- and Z-direction deformations led to an impressive 40% drag reduction.

When considering the deformation using the camber function alone, little impact in the aerodynamic performance was obtained, limited to about 2% reduction in drag, regardless of the regions manipulated (nose, tail or both). However, tuning the fuselage camber has been shown to improve the airflow going through the fuselage to the wing [96, 97], as should not be discarded in more complex cases.

The combination of the camber deformation with the simultaneous Y- and Z- direction deformation resulted in the best overall unconstrained fuselage shape, which exhibited a massive 41.6% drag reduction compared to the baseline. However, some cautions regarding the optimized shape must be taken into consideration since some irregularities were created during the optimization. This combined strategy also produced the best aerodynamic shape for the constrained case with internal payload, with almost 40% drag reduction.

In future studies, to mitigate the referred problems seen in Fig. 5.1, it might be beneficial to make some modifications near the tail region of the fuselage, such as adding a volume constraint might prevent the formation of torsion or warping. Another way to prevent these irregularities could be to incorporate a parametrization strategy, such as those discussed in this study, which, when combined with the Point Selection method, might allow for greater flexibility and deformation control in the affected region.

Referring to the protruding payload cases, only the deformation strategy along the normal vectors was considered to prevent irregular deformations in the fuselage while integrating the triangulated surface constraint with the sphere intersected in the main body. However, in this study, and when comparing the cases represented by Figs. 5.4d and 5.4e, the normal direction deformation only case led to a 7% drag reduction while the combination with the Z-direction deformation increase the reduction significantly to over 21%, mostly due to the impact of the manipulation of the FFD points near the fuselage vertical symmetry XZ-plane.

Several parametrization approaches for fuselage deformation in aerodynamic shape optimization were proposed, tested and compared. This allowed an understanding of the characteristics of each strategy and how they can be applied in more complex design problems.

Drag reduction ranging from 2.1% to 41.6% were observed, depending on the case study considered. Despite the directional parametrization leading to greater drag reduction, the parametrization along the normal vectors of the FFD box points should not be excluded when opting for a deformation strategy. This approach combined with a deformation along the vertical Z-axis have produced good results as well, with the added benefit of reducing the surface irregularities if the DV bounds are properly set.

This work highlighted the importance of selecting appropriate parametrization approaches based on both aerodynamic performance and payload constraints (internal to the fuselage or partially protruding). The triangulated surface constraint proved to be a powerful tool in constraining an aerodynamic problem, offering multiple applications to handle the payload positioning and volume required.

Chapter 6

Fuselage Shape Optimization on Real Case Scenarios

The ARX is a fourth-generation unmanned aerial vehicle (UAV) under development by TEKEVER. It has been designed with long-range mission capability and the ability to cover large areas of operation, as well as support varied sophisticated payloads. These payloads include RADAR, eletro-optical or infra-red cameras, as well as satellite communication (SATCOM) modules, thus rendering the ARX highly versatile to both military and civilian applications.



Figure 6.1: Tekever ARX design.

In this chapter, the tools introduced in Chapters 4 and 5 are going to be used in order to set and solve different aerodynamic shape optimization problems based on real-case scenarios. Taking into consideration the ARX. The test cases presented here are meant to show how shape deformation and parametrization methods can be applied to optimize a more complex geometry. These examples involve deviations from the baseline fuselage shape, illustrated in Fig. 6.1, because of the addition of specific

payloads. The objective will be to model and analyse the resulting increase in fuselage shape and volume to accommodate the payloads presented, while maintaining aerodynamic efficiency, comparing two different sources of drag. In addition, due to mesh simplification of the fuselage, the optimization problems also consider the nacelle geometry, giving further insight into the likely changes to the design required to fit the design payload.

The following sections begin by presenting the different optimization problems that will be addressed, along with the motivation behind each. This is followed by the mesh refinement process, and the reasoning for the final mesh selection used in the optimization cases. Specifically, two payload integration scenarios are considered: first, the integration of a spherical SATCOM antenna mounted on top of the fuselage; and second, the integration of a recovery parachute system modelled as an inclined cuboid volume as a more complex and asymmetrical payload, where issues like corners and flat surfaces will be investigated.

Note that, in this chapter, the absolute drag values will not be disclosed due to their sensitivity and confidentiality in the context of proprietary information of TEKEVER. Percentage differences relative to the original absolute values will instead be used. Percentage results will not influence the comprehension of the chapter since the conclusions are based on relative performance rather than specific numerical values.

6.1 Problem Definition

For the purposes of this study, a simplified fuselage geometry was adopted, as shown in Fig. 6.2a, to give the optimizer maximum freedom to explore shape variations without being constrained by the complex parametrisation of the original design. This geometry is a slight simplification of the original model shown in Fig. 6.1, with the main difference being that the canopy has been merged into the fuselage body. Although this configuration does not represent the actual separation between these components in the original design, it serves as a useful test case for analysing how the optimizer modifies and parametrises a more streamlined version. The insights gained from this study can later support the development of more advanced parametrisations, particularly for improving the transition and blending between the canopy and fuselage in future designs.

For the first optimization case, the focus will be on one specific payload arrangement by representing a SATCOM antenna as a sphere positioned on top of the fuselage. The sphere was designed and exported as an STL file using PyVista, an open-source Python library for 3D plotting and mesh analysis based on the work of Sullivan and Kaszynski [106], having a radius of 20cm. The goal is to apply the deformation and parametrization techniques explained in the previous chapters to form the fuselage shape around this constraint.

The optimization problem is posed as a minimization of the drag coefficient, C_D , under the constraint that the SATCOM sphere must be fully enclosed in the deformed geometry. This will be achieved by using the Point Selection method Sec. 5.1 that enables shape deformation of the FFD box shown in Fig. 6.2b.

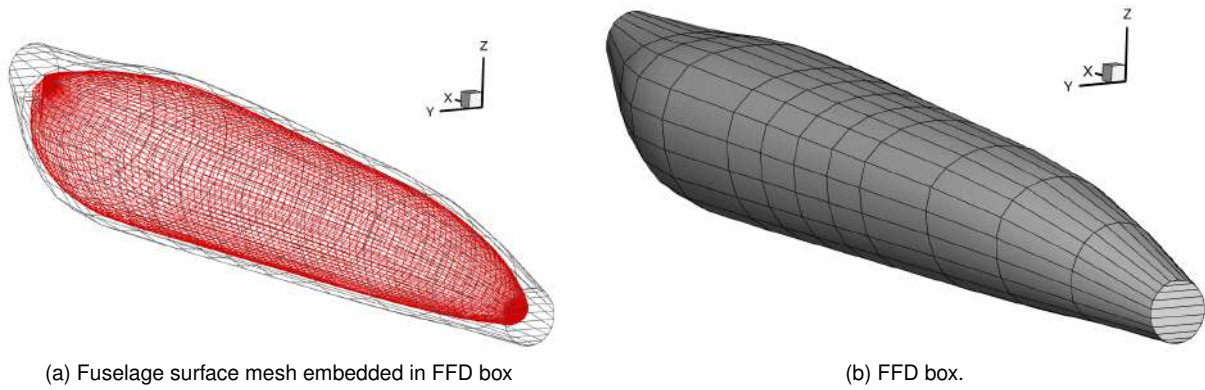


Figure 6.2: Simplified TEKEVER ARX fuselage.

It is important to remember that minimizing the drag coefficient does not necessarily imply minimizing the fuselage volume or surface area. In fact, as a strategy for making the aerodynamics better, the optimizer may actually increase the cross-section of the fuselage in localized regions, especially around the SATCOM sphere. By smoothing out the airflow and promoting more efficient pressure recovery, these local expansions can provide a net drag reduction, despite at the cost of an increase in wetted surface.

To enable effective deformation of the fuselage geometry, two types of design variables were added to the optimization problems that contributed to deformations along the axes y and z . This way, the FFD control points were free to move within the boundaries established. Furthermore, increasing the complexity to the optimizations, the triangulated surface constraint method was applied, using, for the first case, a SATCOM sphere as a reference surface. While the fuselage surface deforms during the optimization, the sphere remains fixed, guiding the optimizer to generate a shape that fully encloses the payload while delivering optimal aerodynamic performance.

In addition to the initial configuration, a second payload setup is examined in this chapter, following the same general optimization methodology. This case introduces an inclined cuboid surface, asymmetrically positioned with respect to the centreline of the fuselage, thereby adding greater geometric complexity to the optimization problem. As with the previous case, the objective is to reduce the drag coefficient while ensuring full enclosure of the payload through controlled surface deformation. Although the number of design variables depends on the resolution of the FFD box, the same parametrization approach is employed—namely, deformation of the Free-Form Deformation box nodes along the Y and Z directions, with the payload geometry acting as a constraint. Visual guidance will be provided later in this work to clarify the geometrical setup and the associated optimization approach.

6.2 Mesh Refinement

To ensure the accuracy of the employed aerodynamic solutions within the optimization process, a mesh refinement study was performed. The objective was to quantify mesh sensitivity and make sure that the computed drag coefficient converges towards the exact value as the mesh becomes finer.

Three structured meshes of increased resolution, shown in Fig. 6.3, were created by refining both surface and volume elements uniformly, with special attention given to the near-wall region around the fuselage to preserve boundary layer resolution. During the generation of the different meshes, the non-dimensional wall distance, y^+ . For the selected mesh was observed while the CFD analysis were being done and it was observed that this value was always smaller than 1 unity. For each mesh, CFD analy-

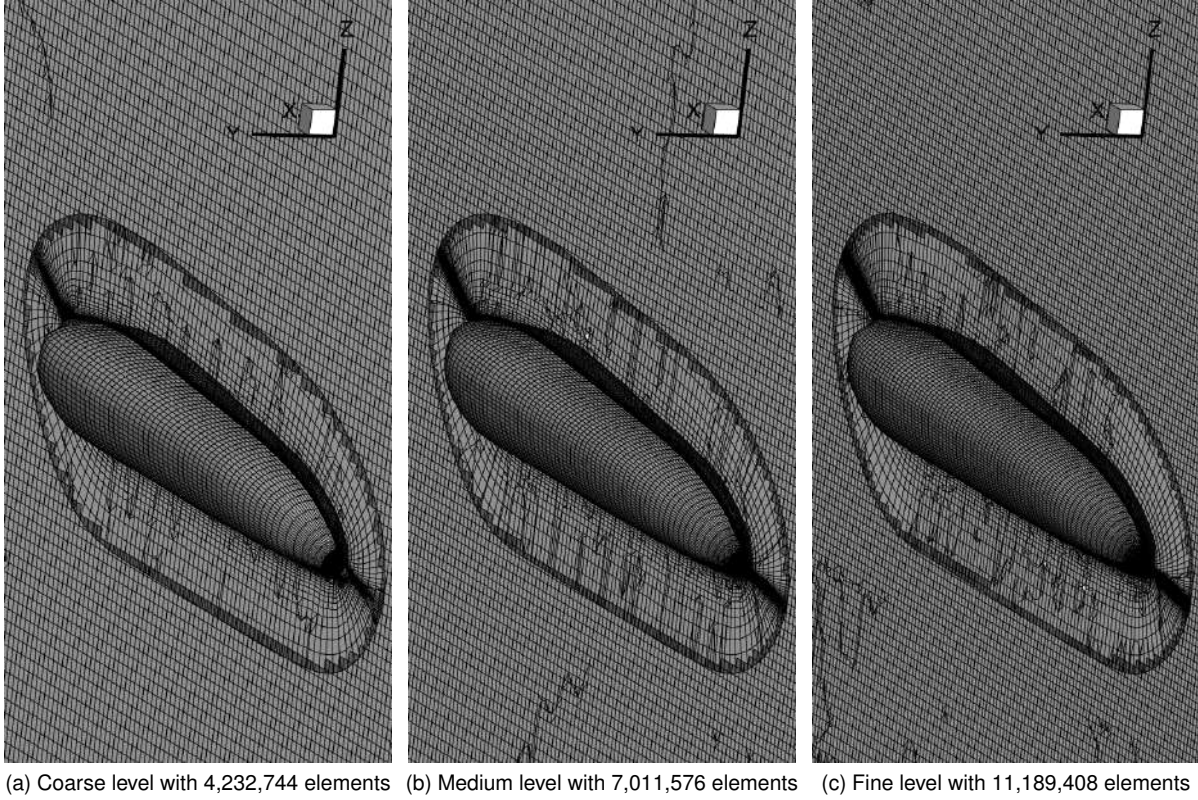


Figure 6.3: Simplified TEKEVER ARX fuselage domain discretization using three levels of mesh refinement.

sis were performed using the same parameters, and the corresponding drag coefficient was recorded. These results are illustrated in Fig. 6.4, where computed drag counts are plotted for the different mesh cell sizes.

Using these results, a Richardson extrapolation was applied to estimate the asymptotic (grid-independent) value of the drag coefficient, C_D^{ext} , which represents the value that would be obtained with an infinitely fine mesh. This method provides a practical means to verify the convergence of the mesh in CFD studies and has been successfully applied in various aerodynamic simulations. Some examples of such applications can be found in [107, 108], where Richardson extrapolation was used to assess mesh sensitivity and ensure the reliability of aerodynamic and aeroacoustic results in wind turbine simulations. The convergence trend is illustrated in Fig. 6.4, showing the expected monotonic behaviour.

To select the mesh for the optimization phase, two main criteria were considered: the computational time required to perform the CFD simulations, and the accuracy of the resulting drag coefficient when compared to a reference value obtained through Richardson extrapolation. The mesh refinement study using the three grids of increasing resolution are summarized in Table 6.1.

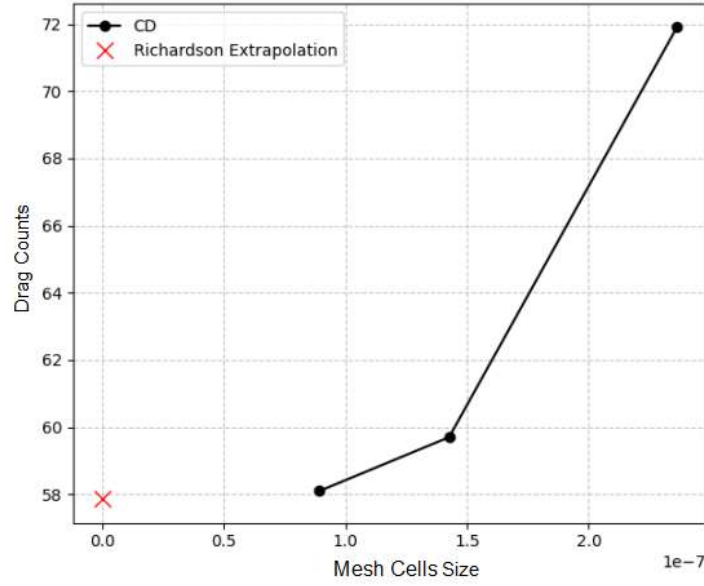


Figure 6.4: Drag coefficient convergence study using Richardson's extrapolation

The selected mesh for the optimization process corresponds to the medium-resolution grid, which produced a drag coefficient approximately 3% higher than the value estimated through Richardson extrapolation. The extrapolated drag coefficient, calculated using the Richardson method was used, and the relative error was computed as

$$\text{Relative Error} = \left| \frac{C_D - C_D^{\text{ext}}}{C_D^{\text{ext}}} \right| \times 100\% \quad (6.1)$$

For the chosen mesh, it resulted in a 3.29% error, within the admissible limit of 5% for aerodynamic optimization. The lower error of 0.52% for the finer mesh was at the expense of effectively double the computational time to simulate; over 8000 seconds compared to some 4300 seconds for the medium mesh, both timed over 80 CPUs. Since all optimization problems were executed with just 60 or 40 CPUs, due to resource limitations, simulation time for the fine mesh would have been much larger. Conversely, although the coarse mesh required only half the time of the medium-resolution case, its relative error of around 24% exceeded the acceptable limit and was therefore easily discarded. Thus, the medium-resolution mesh was a reasonable trade-off between accuracy and computational effectiveness.

Table 6.1: Drag coefficient error in mesh refinement study.

Mesh ID	# CPUs	# Mesh Cells	Run Time (s)	ΔC_D in %
Coarse	80	4,232,744	2,463	+24.39
Medium	80	7,011,576	4,323	+3.29
Fine	80	11,189,408	8,028	+0.52
RE	—	—	—	Ref

RE – Richardson Extrapolated drag coefficient.

Relative error calculated against the extrapolated value of C_D .

6.3 Baseline Analysis

Before addressing the optimization challenges, it was important to first understand the aerodynamic behaviour of the baseline ARX fuselage. In this case, the baseline analysis corresponds to the initial iteration of the optimization process, before any shape deformations were applied. Note, that no payloads were considered at this stage, and the fuselage geometry was evaluated without the sphere nor the cube. From the CFD solution obtained, the distribution of the pressure (C_p) and skin-friction (C_f) coefficients on the fuselage surface were analysed and are shown, respectively, in Figs. 6.5 and 6.6.

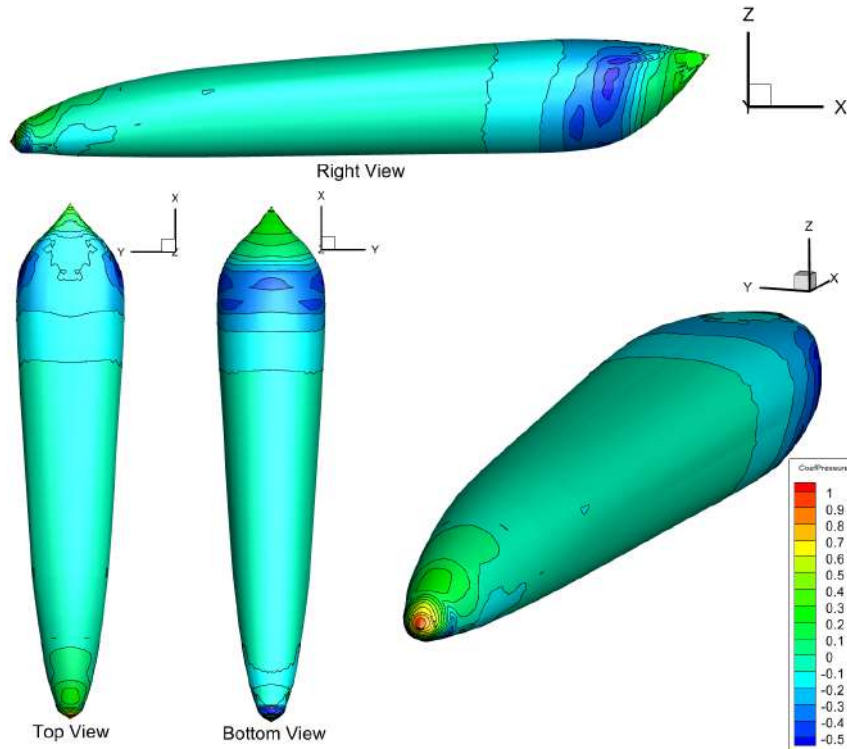


Figure 6.5: Pressure coefficient distribution on the baseline TEKEVER ARX fuselage.

As expected for this class of UAVs, after analysing the pressure distribution on the baseline fuselage, a high-pressure region was identified near the nose, followed by a progressive decrease along the forward fuselage, indicating flow acceleration. Across the mid-section, the pressure remained relatively stable, with moderate negative values corresponding to attached and relatively undisturbed flow. It is observed that the pressure continued to decrease smoothly along the fuselage. Near the tail, a slight increase in pressure was detected, which may be considered an adverse pressure gradient, indicating a risk of flow separation. This action underscores the importance of aft-body shaping in optimization, particularly with the addition of volumes that correspond to the payloads. This confirms that slight geometric improvement can greatly mitigate the impact of adverse pressure gradients and reduce form drag.

As addressed throughout this thesis, the main sources of drag for the presented problems are pressure drag and skin-friction drag, so, it is also important to analyse Fig. 6.6, which illustrates the distribution of this second type of drag. As expected, the skin friction coefficient, near the nose is very close to

zero due to proximity of the stagnation point where tangential flow is negligible. As the boundary layer develops along the mid-fuselage, a relatively uniform distribution of skin friction is observed, indicating attached and stable flow. Towards the aft, in regions where the curvature increases, the highest values of skin-friction are reached, corresponding to areas with greater velocities. Finally, near the trailing edge of the fuselage, a dark blue area is observed, where the skin-friction takes negative values indicating possible separations in the flow. This supports the interpretation previously made when describing the pressure distribution at the rear of the fuselage.

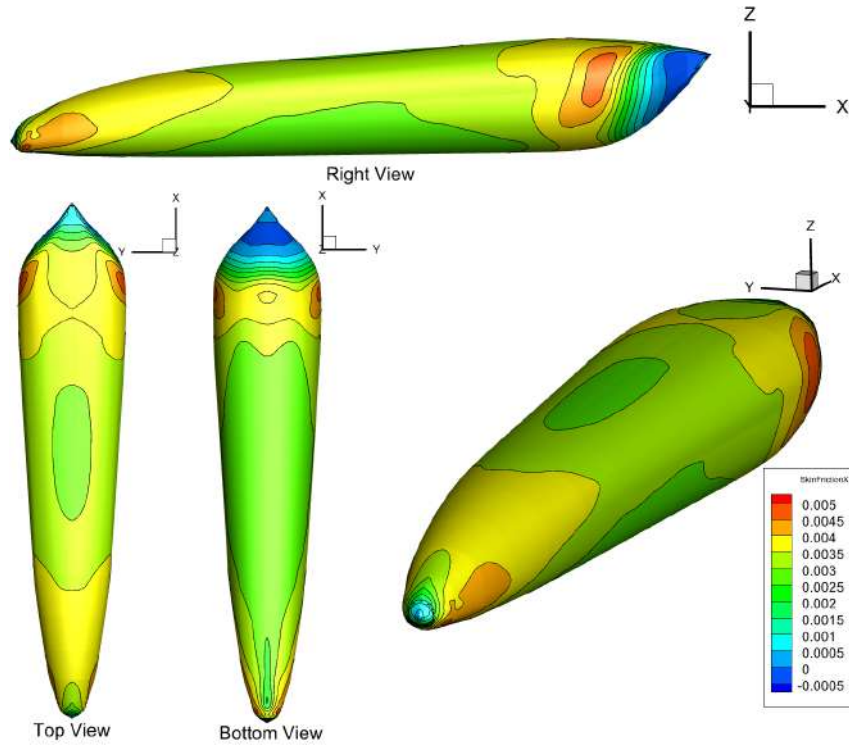


Figure 6.6: Skin friction coefficient distribution on the baseline TEKEVER ARX fuselage.

The drag coefficient (C_D) of the baseline configuration was recorded, providing a reference point for evaluating the optimization gains. It is important to note, however, that since the SATCOM sphere was not considered in the baseline geometry, the overall drag coefficient of the final optimized fuselage is expected to increase. However, as it will be shown later, in some cases, the optimizer was able to compensate for the added drag by smoothing the aft region around the enclosed payload, effectively mitigating part of the aerodynamic penalty.

The following two sections present the development and evaluation of two distinct aerodynamic shape optimization cases. Each case begins with an overview of the specific configuration under study, followed by a detailed discussion of the challenges and irregularities encountered during the problem formulation phase. By progressively increasing the complexity of the problems, the issues encountered along the way were addressed and overcome. These insights proved essential in defining the final parametrization of each optimization problem. A complete problem description is then provided, including the objective function, imposed constraints, and selected design variables. Finally, for each case, an aerodynamic assessment of the optimized fuselage is conducted through the analysis of pressure and

skin-friction coefficient distributions (C_p and C_f), offering insight into the flow behaviour and aerodynamic performance achieved through the applied deformation approaches.

6.4 Fuselage Shape Optimization with a Protruded Spherical Payload

While the initial goal was to apply the parametrization methods and optimization approaches developed in the earlier study examples directly to the ARX fuselage, several modifications had to be made in order to create an efficient final result. Certain modifications were implemented in the deformation methods, mesh refinement, and application of the constraints, tailored to the particular difficulties of the ARX shape and payload distribution. Furthermore, since the FFD box was different from the previous Chapter 5, the point selection needed to be different, as well as the bounds established for the different parametrization approaches. These changes allowed the optimization process to accommodate the integration of the SATCOM antenna more easily, without compromising aerodynamic efficiency. Following, the major outcomes of the optimization are described, including the improvements achieved and the significant modifications to the approach. The first optimization case is shown in Fig. 6.7.

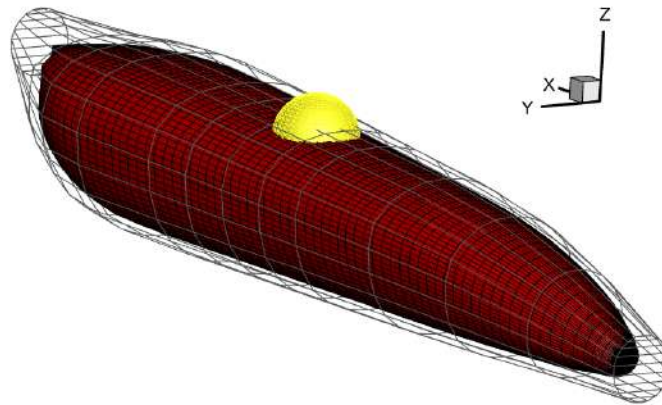


Figure 6.7: First TEKEVER ARX fuselage optimization problem with an integrated sphere.

Although the SATCOM antenna, idealized as a sphere, was symmetrical about the $y = 0$ plane, a symmetric mesh approach was initially employed with the aim of reducing computational cost by modeling half of the fuselage. A number of mesh failures were, however, encountered during the optimization. It is believed that these were mostly due to the types of deformations that were required in an attempt to create room for the payload. As explained in Sec.4.5, the triangulated surface constraint measures distances and intersections along the entire surface. Simulating only half of the fuselage alone limited the optimizer's ability to correctly read and impose the constraint, especially for large deformations. Therefore, a number of cases yielded invalid meshes, which made the optimization stall. In order to handle these problems, the symmetric mesh strategy was discarded in favour of a full-fuselage mesh to provide uniform constraint enforcement and robustness of the optimization process.

The results obtained in Chapter 5 suggested that deforming the FFD box along the normals of its

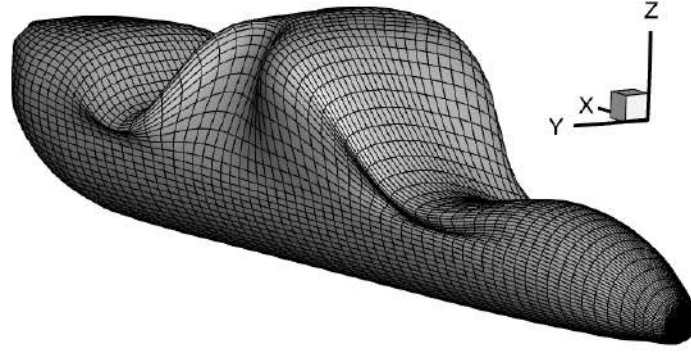


Figure 6.8: Example of a failed mesh in TEKEVER ARX fuselage optimization.

nodes would be the most stable parametrization technique. However, due to the increased complexity of the ARX geometry, and consequently, of its FFD box, this approach proved difficult to apply. In more complex geometries, the local surface curvature strongly influences the orientation of the FFD control point normals. As a result, small perturbations along these directions can lead to unpredictable or highly localized deformations, making it difficult for the optimizer to maintain control over the global shape of the fuselage. In this case, the deformations induced by moving the control points along the normals were unplanned, and the optimizer struggled to control and stabilize the fuselage geometry. This instability, combined with an improper scaling between the different parametrization techniques, the objective function, and the triangulated surface constraint, led to results similar to those illustrated in Fig. 6.8. This was particularly important to understand that deformations along the axes Y and Z would be much more effective for this case. Compared to normal-based deformations, axis-based control point displacements allowed for more consistent and predictable shape changes across the fuselage, providing the optimizer with better control over the node displacements. Additionally, to avoid irregularities like the ones encountered in Sec. 5.1, the point selection used to choose the nodes to be parametrized was crucial as well as the bounds applied, since too large displacements could cause, again, failed mesh deformations.

To obtain the first successful optimization result, the deformation approach was based primarily on displacements along the Y and Z axes. In addition, the scaling applied to the triangulated surface constraint was reduced, which helped to smooth the deformations performed by the optimizer in adapting the fuselage shape to enclose the SATCOM sphere. At this point, the main goal was to be able to run an optimization problem without mesh failures, allowing the optimizer to properly control the fuselage shape to reduce the drag coefficient while correctly integrating the payload.

For the first optimization case (case 1), as mentioned, the goal was to efficiently enclose the SATCOM sphere, while minimizing the drag coefficient, C_D . The setup for this problem is summarised in Tab.6.2, which includes the design variables, their bounds, and the scaling factors applied to ensure stable convergence. A total of 120 design variables were defined, split evenly between FFD control points displacements along the Y and Z axes. To constrain the shape around the payload, the triangulated surface constraint was applied, using a spatial tolerance $\rho = 500$, along with a scaling factor of 0.1 to balance the influence of the constraint in the optimization process. The objective function was scaled by 500 for numerical robustness. To focus the deformations in the region of interest, the point selection

Table 6.2: ARX fuselage aerodynamic shape optimization parameters - Case 1

Variable	Description	Quantity
Minimise		
C_D	Drag at fixed flight condition	—
by varying:		
x_{FFDy}	FFD control points in Y direction	60
x_{FFDz}	FFD control points in Z direction	60
subject to:		
$x_{FFDy} \in [-0.1, 0.1]$	Bounds for Y displacements	—
$x_{FFDz} \in [-0.1, 0.2]$	Bounds for Z displacements	—
$KS_{geom} \leq 0$	Aggregated minimum distance (spatial feasibility Condition 1)	1
$L = 0$	Non-intersection (spatial feasibility Condition 2)	1
at condition:		
Geometry	Baseline ARX with SATCOM sphere	—
FFD point selection	Nodes activated near payload region	—

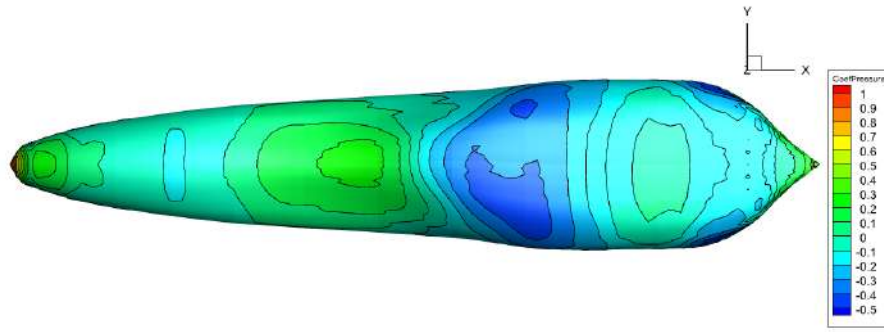
method was used, allowing only the FFD control points near the SATCOM sphere to move. The baseline geometry is shown in Fig.6.7, and the optimal shape shown in Fig. 6.9a confirms that the payload was successfully integrated while maintaining aerodynamic continuity of the fuselage. This case served as a controlled test of the methodology without introducing large-scale shape changes.

In the second optimization case (case 2), represented in Fig. 6.9b, the objective was the same as in the first one: to integrate the payload into the fuselage with optimal aerodynamic performance. The major difference between the two cases lies in the scaling factors applied in the design variables and the objective function. Specifically, a scale factor of 1.0 was employed for all the design variables, and the objective function was scaled by 100. This configuration was used to test how different scaling values would impact the behaviour of the optimizer, particularly in relation to the convergence time and the quality of the obtained solution.

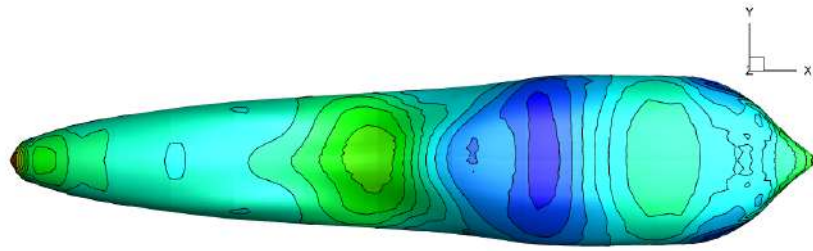
Regarding the aerodynamic performance of both optimization cases, the output geometries are similar and share the same issue: the optimized fuselage is not symmetric. This becomes evident when observing the fuselage top view of both cases in Fig. 6.9, where it can be seen that the pressure distribution differs between the left and right sides. These results are most likely a consequence of the convergence parameters used during the optimization. This issue could potentially be resolved by applying stricter convergence criteria.

In general, the pressure along both fuselages is distributed as expected. Near the nose, the high-pressure region remains unchanged, since no parametrisation was applied in that area. Then, in both cases, a drop in pressure is observed, indicating an increase in flow velocity as the fuselage tapers. Just before the fuselage begins to enclose the SATCOM sphere, where the flow encounters a sudden increase in cross-sectional area, there is a corresponding rise in pressure. This behaviour is expected, and the smoothness of the transition suggests that the flow remains attached, which is a positive outcome in terms of aerodynamic performance.

In addition to the expected rise in pressure where the fuselage begins to enclose the SATCOM, it is



(a) Case 1.



(b) Case 2.

Figure 6.9: Pressure coefficient fuselage top view comparison of the two optimized shapes highlighting asymmetry.

also observed that, over the top section of the payload, the pressure decreases again. This indicates an acceleration of the flow over the upper surface. Following this, towards the aft part of the fuselage, the pressure increases once again. This behaviour suggests the presence of an adverse pressure gradient that could lead to flow separation, particularly near the base of the fuselage. Although flow detachment is not clearly confirmed from the figure, the trend reinforces the importance of aft-body shaping to ensure pressure recovery and maintain attached flow.

In terms of performance, both optimization cases led to geometries that successfully integrated the sphere. These optimizations were completed in approximately 178 hours for case one and 244 for the second case using in each case 40 CPUs. These two optimization results highlight how the scaling of the design variables and objective function can influence the computational cost, as these were the main differences between the two cases. Probably, in the first case, the scaling was better suited for this type of problem, which helped the optimizer to deform the fuselage more efficiently, resulting in fewer iterations overall.

The effect of the different scaling parameters on the resulting fuselage shapes can be clearly observed in Fig. 6.10, where the variation in the slope used to enclose the sphere is noticeable between the two cases. As it is shown in Tab. 6.5, when comparing these cases with the baseline fuselage, in terms of drag coefficient, a reduction of approximately 2.7% in Case 1 and 3.6% in Case 2 were achieved. This showed how effective the optimizer is in adapting the fuselage geometry to integrate the payload while improving aerodynamic performance. As expected, the fuselage volume increased when compared to the baseline, in both cases, as the goal was to integrate the entire sphere into the

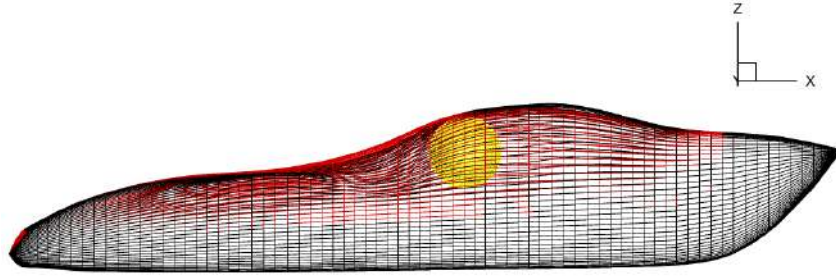


Figure 6.10: Overlay of optimized fuselage shapes for Case 1 (red) and Case 2 (black), both integrating a protruded spherical payload

fuselage, while restricting deformations in certain regions. Such constraints inevitably led to local expansions of the geometry, resulting in an increase in wetted surface and, consequently, in skin friction drag. Nonetheless, the optimizer still managed to reduce the total drag in both cases. According to the CFD simulations, this result was primarily driven by a reduction in pressure drag, while the viscous drag, and as explained, suffered a slight increase.

Overall, while symmetry issues remain and some pressure recovery regions could still be improved, the optimizer was able to produce viable aerodynamic shapes that managed to maintain favourable pressure distributions and reduce overall drag, demonstrating the effectiveness of the methodology.

Following the first optimization results of the ARX fuselage with an integrated payload, it became evident that the optimizer had to undergo a high number of CFD and solver iterations in order to converge. In an effort to address this and simplify the problem to be easily set up, a new method was created by invoking symmetry through the plane $y = 0$. This was done by positioning the FFD nodes in symmetric pairs along the fuselage, in a way that the displacements along the Y and Z axes were imposed only on one side and then copied to the other. This not only reduced the effective number of degrees of freedom, but also prevented mesh irregularities and convergence instabilities like the ones explained in the beginning of this section, since the entire fuselage was still being considered to calculate the distances between the surface triangles. Furthermore, for nodes positioned on the symmetry plane, only displacements along the Z axis were considered to guarantee that the optimizer would converge to a symmetric fuselage.

The parameters used for this third case (case 3) are summarized in Tab. 6.3, including updated bounds of $[-0.2, 0.2]$ for the Z axis and $[-0.1, 0.1]$ for the Y axis. All design variables were scaled by 1.0. The triangulated surface constraint was again applied, this time with a larger spatial tolerance of $\rho = 750$, to promote better spatial fitting between the deformed fuselage and the SATCOM sphere. Overall, this setup was designed to improve convergence behaviour and explore if symmetry and improved enforcement of spatial constraint could lead to more efficient fuselage shapes.

A detailed comparison between the baseline and optimized fuselages for case 3 are presented through the pressure and skin-friction coefficient distributions are shown in Figs. 6.11 and 6.12, with the baseline fuselage on the left side and the optimized geometry on the right side in each figure. The different views are divided along the symmetry plane at $y = 0$, providing a clear comparison of the

Table 6.3: ARX fuselage aerodynamic aerodynamic shape optimization parameters - Case 3 (Symmetry-Enforced)

Variable	Description	Quantity
Minimize		
C_D	Drag at fixed flight condition	—
by varying:		
shape_vertical_func	FFD vertical deformations (Y, Z)	30
shape_horizontal_func	FFD horizontal deformations	24
subject to:		
$x_{FFDy} \in [-0.1, 0.1]$	Bounds for Y displacements	—
$x_{FFDz} \in [-0.2, 0.2]$	Bounds for Z displacements	—
$K S_{geom} \leq 0$	Aggregated minimum distance (spatial feasibility Condition 1)	1
$L = 0$	Non-intersection(spatial feasibility Condition 2)	1
at condition:		
Symmetry	at ($y = 0$)	—
Point selection	near payload	—

two coefficients between the two stages of the optimization. The following analysis highlights the key differences between the two configurations and examines the aerodynamic implications of the applied surface deformations.

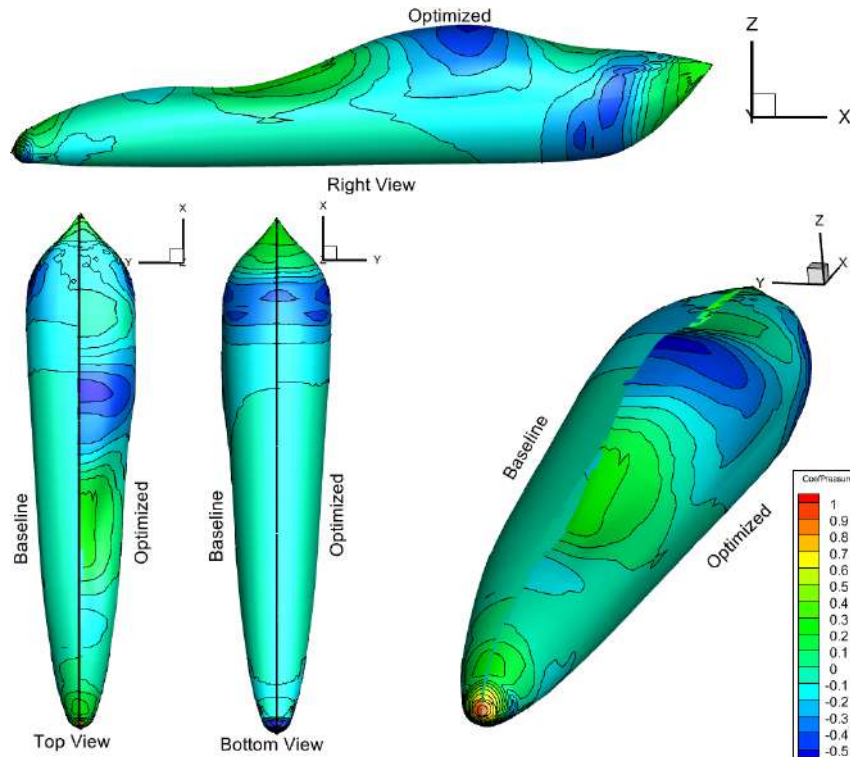


Figure 6.11: Pressure coefficient distribution on the optimized TEKEVER ARX fuselage (case 3).

As observed in the pressure distribution shown in Fig. 6.11, the optimized fuselage exhibits a completely different pressure distribution when compared to the baseline fuselage. Near the nose, the pressure is still relatively high, as no shape changes in this region were allowed. As flow progresses, the

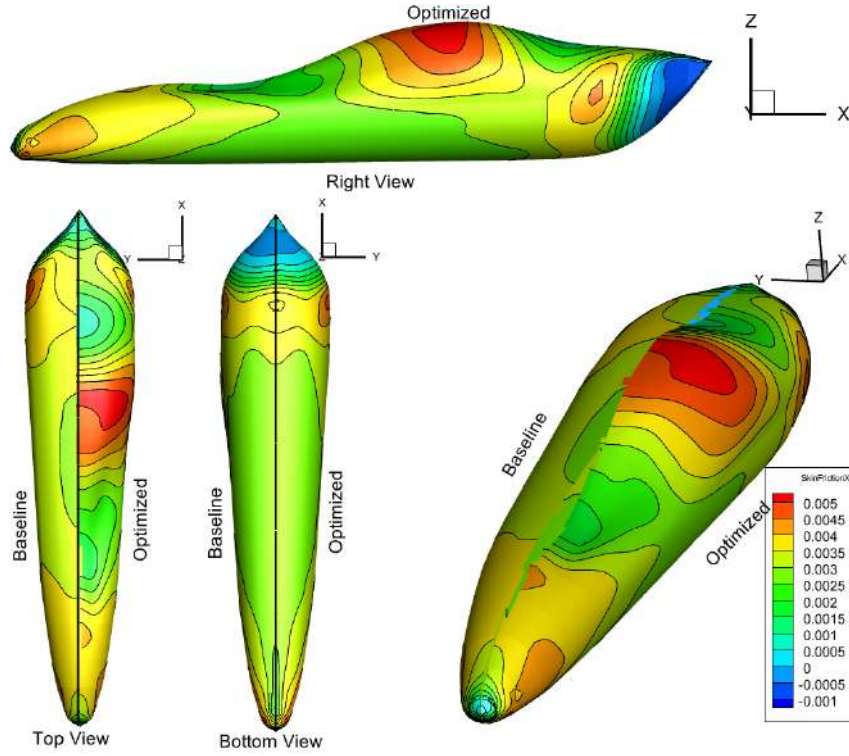


Figure 6.12: Skin friction coefficient distribution on the optimized TEKEVER ARX fuselage (case 3).

fuselage begins to widen, causing the flow to decelerate. This results in a local increase in pressure and, consequently, in the pressure coefficient C_p . Near the aft of the fuselage, and once the sphere is fully enclosed, the geometry transitions into a region of decreasing cross-sectional area. This leads to an acceleration of the flow. This behaviour is reflected in the C_p distribution, where a lower pressure region (dark blue) emerges. Near the trailing edge, the geometry appears to provide a relatively smooth pressure recovery, helping to stabilise the flow and reduce potential flow separation.

The skin friction coefficient distribution shown in Fig. 6.12 complements the pressure data by highlighting the skin-friction patterns. Near the nose, skin friction is low, consistent with the stagnation zone where tangential velocity is minimal. Along the mid-fuselage, skin friction increases steadily, reflecting an attached boundary layer. The highest values are concentrated over the top of the fuselage, particularly around regions of increased curvature where flow speeds are greatest. Towards the tail, C_f gradually decreases, and isolated blue regions, indicating negative or near-zero skin friction, emerge, which may indicate possible flow separations, supporting what was previously discussed.

Overall, the C_p and C_f distributions give a good indication of how the flow behaves around the optimized fuselage. According to Tab. 6.5, this final case resulted in a 4.6% reduction in overall drag. In this optimization case (case 3), pressure drag was reduced by around 10.3%, while skin-friction drag increased by approximately 2.3%. This means that the optimizer was able to reduce the total drag by essentially decreasing pressure drag. On the other hand, the increase in skin-friction drag was due to the larger volume and surface area of the fuselage required to enclose the entire sphere, which naturally leads to higher shear forces along the surface.

6.5 Fuselage Shape Optimization with a Protruded Box Payload

To better understand how the shape of a surface payload might influence the optimization, a new case (Case 4) was explored. For this case, a simple box with dimensions $300 \times 300 \times 168$ mm was created using PyVista, with an inclination of 20° along the X axis. This inclination caused the corners of the box to be raised and lowered relative to one another, introducing geometric irregularities that could challenge the optimizer. The initial setup is shown in Fig. 6.13.

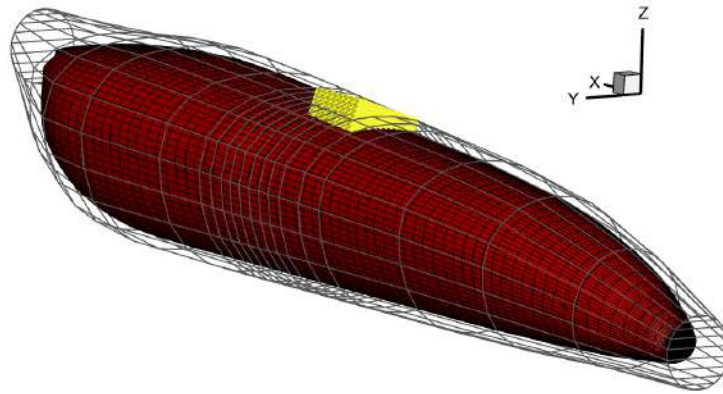
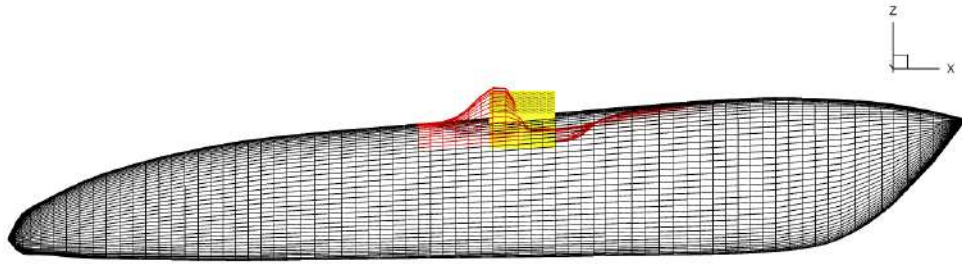


Figure 6.13: Second TEKEVER ARX fuselage optimization problem with an integrated box.

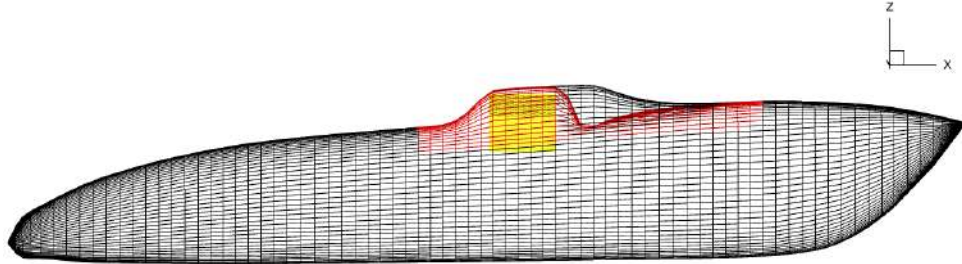
During this study, several considerations had to be made. Since the box was not symmetric with respect to the $y = 0$ plane, there was a risk of obtaining unfeasible geometric solutions due to the lack of symmetry constraints. To address this issue, the same method developed for the previous problem (Case 3) was applied here as well. With this approach, a symmetric result was expected.

The objective of this case was to understand how the optimizer would incorporate the inclined box into the fuselage in a way that maximized aerodynamic performance. While a smooth geometric integration was desired, it was also necessary to control where this integration would begin. To achieve this, instead of using the point selection method to allow movement in specific regions, a new FFD box was created, divided into multiple sections. The nodes near the front of the box were kept fixed, preventing the optimizer from modifying the geometry in that region. As shown in Fig. 6.13, the number of sections in this FFD box, near the payload region, was increased, which also led to a higher number of control points. This provided greater flexibility in shaping the fuselage around the box, allowing for a more refined integration into the inclined geometry.

As expected, the integration of this new payload raised several problems, as illustrated in Fig. 6.14. Initially, the integration was a direct application of what was previously done for the sphere. The optimization began with the baseline fuselage and it would be expected that the optimizer would progressively deform the fuselage to enclose the entire box in a smooth way. However, as observed in Fig. 6.14a, the optimizer became trapped in the intersecting region, preventing further progress, as previously discussed in Sec. 4.5. To overcome this problem, a modified baseline fuselage was generated to ensure the entire box was enclosed. This was achieved by applying a vertical translation to selected FFD nodes along the Z axis, effectively lifting the FFD box and, consequently, the fuselage. The resulting geometry served as the new initial shape for the optimization problem, as shown in Fig. 6.14b.



(a) Comparison between the baseline fuselage (Black) and the optimized result (Red)



(b) Comparison between a previously deformed fuselage (Black) and the optimized result (Red)

Figure 6.14: Failed optimizations using different baseline fuselages

This way, the optimizer was able to converge; however, it did not fully satisfy the imposed constraints, and some of the design variables reached their assigned upper bounds. This could indicate that the optimizer was operating at the limits of the feasible design space and that the problem formulation may require additional flexibility to modify the design variables' values to a more feasible solution. In Fig. 6.14b, the final fuselage shape (in red) is shown, presenting a geometry that is not smooth in the aft region, as it would be expected. Although the box was fully enclosed, the optimizer attempted to conform too closely to its shape, which led to an abrupt transition and a loss of smoothness at the rear of the box.

To conclude this case study and address the problems previously encountered, several parameter adjustments were made as summarized in Tab. 6.4. As in the previous problem, the modified fuselage, already enclosing the entire inclined box, was used as the initial shape for the optimization problem. To increase the flexibility of the optimizer and prevent the values of the design variable from reaching the upper boundaries, the upper limits of the applied design variables were increased. This allowed for broader range of geometric deformations and improved the optimizer ability to adapt the fuselage shape around the box.

Furthermore, the spatial fitness value of the triangulated surface, ρ , was reduced by a factor of five. As mentioned in Sec. 4.5, this parameter controls how close the optimizer is required to match the reference surface, in this case, the box. Based on the work done by Brelje et al. [98], reducing ρ helps relax this constraint, giving the optimizer more freedom to prioritize smoother shapes over exact surface matching. The idea was to make the integration of the inclined box easier by avoiding sharp transitions and giving the optimizer more flexibility to generate a continuous fuselage without being forced to fit the

Table 6.4: ARX fuselage aerodynamic shape optimization parameters - Case 4 (Symmetry-Enforced)

Variable	Description	Quantity
Minimize		
C_D	Drag at fixed flight condition	—
by varying:		
shape_vertical_func	FFD vertical deformations (Y, Z)	27
shape_horizontal_func	FFD horizontal deformations	18
subject to:		
$x_{FFDy} \in [-0.1, 0.1]$	Bounds for Y displacements	—
$x_{FFDz} \in [-0.3, 0.3]$	Bounds for Z displacements	—
$K S_{geom} \leq 0$	Aggregated minimum distance (spatial feasibility Condition 1)	1
$L = 0$	Non-intersection(spatial feasibility Condition 2)	1
at condition:		
Symmetry	at ($y = 0$)	—
Point selection	near payload	—

triangulated surface too tightly. This change was expected to improve the final result by reducing the abrupt parametrization changes, seen in the previous case, in Fig. 6.14b. The goal for the optimized fuselage was to enclose the entire box while ending in a smooth way, avoiding large pressure coefficient differences and helping to keep the flow attached to the fuselage surface.

However, although the optimizer was able to converge under these conditions, it did not satisfy all the constraints due to the presence of sharp corners and edges in the original box design. These abrupt features limited the optimizer's ability to generate a smooth and continuous surface around the reference geometry. As a result, further modifications to the box were necessary and so, all sharp edges and corners were smoothed. This adjustment helped reduce conflicting constraints and improved the optimizer's ability to find feasible solutions that better integrated the box within the overall fuselage shape.

A detailed comparison between the baseline (left-side) and the optimized fuselage (right-side) for the last case (Case 4) is shown in Figs. 6.15 and 6.17. The same methodology is used to demonstrate the distribution of pressure and friction on the fuselages, where a symmetry plane at $y = 0$ is used to divide the different views.

Table 6.5: Comparison of drag coefficient variations and computational time.

Configuration	Run Time (h)	(ΔC_D)	(ΔC_{D_p})	(ΔC_{D_v})
Baseline	—	Ref	Ref	Ref
Case 1	178	-2.7%	-7.5%	+3.3%
Case 2	244	-3.6%	-9.6%	+3.7%
Case 3	118	-4.6%	-10.3%	+2.3%
Case 4	121	+24.7%	+42.8%	+2.5%

As expected, the resulting fuselage is much more similar to the baseline compared to the other three cases previously presented in Sec. 6.4. This is due to the more restricted displacements applied to the FFD box. However, because the box had to be fully enclosed while avoiding large deformations in

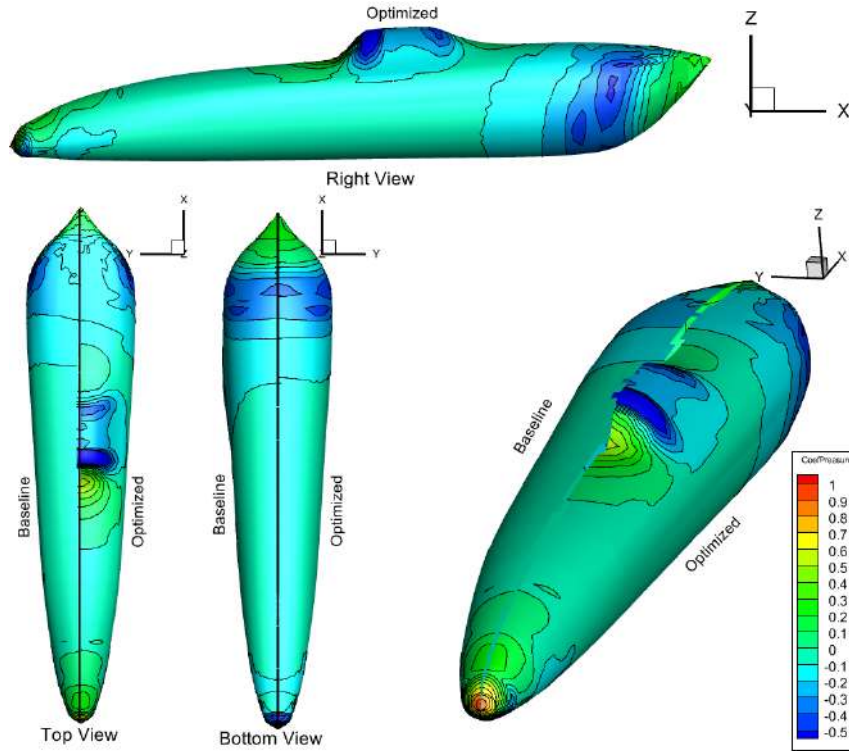


Figure 6.15: Pressure coefficient distribution on the optimized TEKEVER ARX fuselage (case 4).

the front section of the fuselage, the geometry transitions sharply to enclose the parachute box. This leads to a nearly perpendicular slope in that region, which creates a high-pressure area. Once the box is enclosed, the fuselage experiences a pronounced pressure drop, accelerating the flow around this section. To help maintain flow attachment, a smooth and gradual slope is introduced in the aft part of the fuselage, following the cross-sectional enlargement.

Nevertheless, as shown in Tab.6.5, a flow separation likely occurred due to the significant increase in pressure drag. Specifically, the fuselage experienced an increase of nearly 43% compared to the baseline. Figure 6.16 illustrates the region where this separation takes place, clearly showing where the streamlines reverse direction, and providing valuable insight into the aerodynamic behaviour of the optimized geometry.

The skin friction distribution of the optimized fuselage is compared with the baseline in Fig. 6.17. As expected, since the fuselage did not undergo any deformations near the nose, the C_f distribution in this region remains identical to that of the baseline. As the flow encounters the wall enclosing the box, the velocity decreases, and consequently, so does the skin friction. This is visible in the light blue/green region, corresponding to the area where the fuselage cross section begins to expand. Supporting the previous analysis, a second light blue/green region appears where the cross section starts to decrease. In this region however, the velocity does not show a similar reduction, which may indicate a possible flow separation supporting the affirmations previously made.

Overall, several conclusions can be drawn from this study. Firstly, it must be noted that, contrary the trends observed in the previous cases, the optimizer was not able to minimize the total drag coefficient (C_D). This may be attributed to the restrictions applied, as well as the bounds used for the different

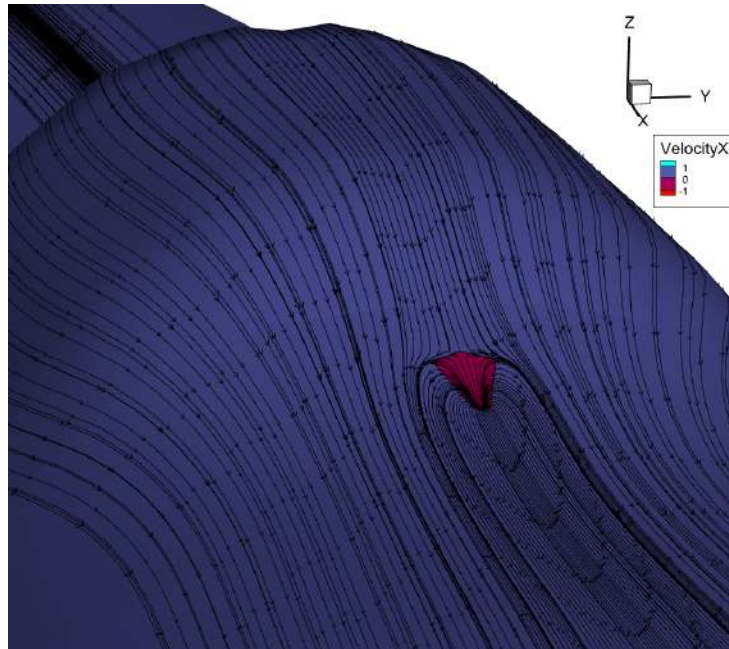


Figure 6.16: TEKEVER ARX fuselage optimization problem with an integrated box (case 4)

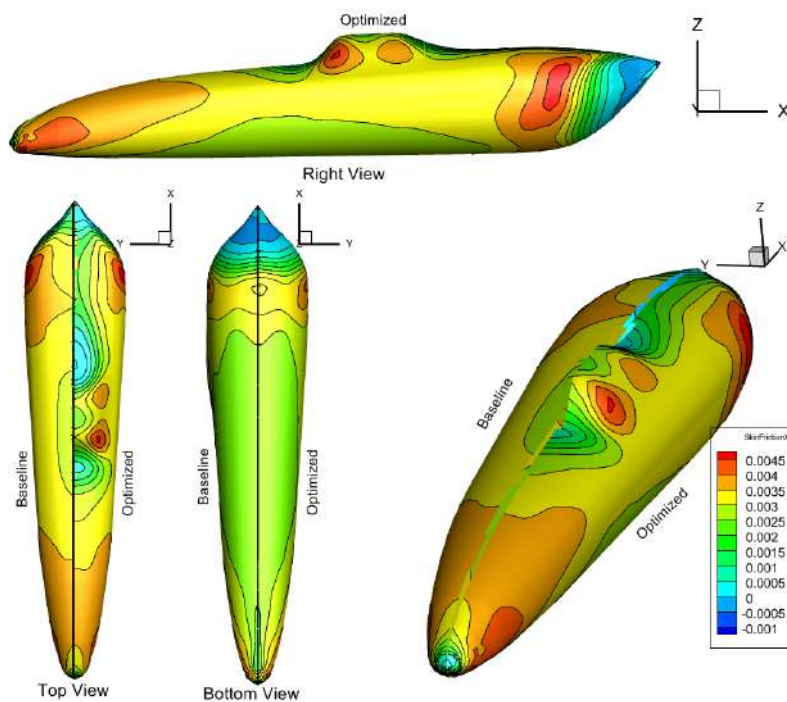


Figure 6.17: Skin friction coefficient distribution on the optimized TEKEVER ARX fuselage (case 4).

design variables, which limited the optimizer's flexibility. In Case 4, an increase of approximately 25% in total drag was observed, as shown in Tab. 6.5. While the skin friction drag increased by only 2.5%, remaining within the range of the increases seen in the earlier cases, the pressure drag component rose around 43%. As discussed previously, this significant rise in pressure drag is likely due to the nearly perpendicular slope introduced to enclose the box, which created adverse pressure gradients and led to flow separation. This highlights how the smoothness of the fuselage streamlines contribute to flow attachment and, consequently, improve the aerodynamic characteristics of a given geometry.

Chapter 7

Conclusions

7.1 Achievements

This thesis used the MACH-Aero framework to perform aerodynamic fuselage shape optimization with payload integration. The objective was to minimize the overall drag while operating under geometric constraints and different design variables. The Free-Form Deformation (FFD) method was used to parametrize the geometry and control the shape during the optimization process. This thesis was structured in a way that helped building both technical understanding and practical skills in applying different parametrization approaches. These methods were first implemented and tested on a simplified geometry to evaluate their behaviour and performance in controlled conditions. After understanding the different methods and how the implementation of these could be carried out, two more complex and realistic scenarios were defined. These final cases combined several of the previously studied approaches and were analysed to draw conclusions on their effectiveness in real-case applications.

During the thesis, several deformation approaches were explored, including deformations along the axes Y and Z , and along the normal vectors of the FFD nodes. It was created a method to add camber as a design variable and the point selection strategy was used to define regions of possible parametrizations for the optimization problems. These were combined with the triangulated surface constraint method to ensure that internal payloads could be enclosed without generating infeasible fuselage geometry or compromising flow quality. These approaches were benchmarked using a basic fuselage to establish the limitations and advantages of each, and were used as the foundation to address more realistic cases.

The final optimization cases (Cases 1, 2 and 3) demonstrated how the integration of these tools and approaches could lead to aerodynamic improvements. In the spheric SATCOM integration case, a total drag reduction of 4.6% was achieved, primarily due to a 10.3% decrease in pressure drag. This gain was slightly offset by an increase of approximately 2.3% in skin-friction drag, caused by the additional surface area required to house the payload. On the other hand, Case 4 presented a different outcome. Despite following the same optimization framework, the stricter design constraints and reduced geometric flexibility led to a total drag increase of approximately 25%. While the increase in skin-friction drag

remained within the same range at 2.5%, the pressure drag rose significantly, by about 43%, highlighting the adverse effects of the abrupt fuselage expansion required to enclose the payload.

It is important to note that, compared to the first studies, these final optimization problems were subject to stricter constraints. Since the geometry aimed to represent a real-case scenario, changes to the fuselage could potentially affect the positioning of internal components or structural elements. These restrictions limited the design room available to the optimizer but also made the problem more realistic. Despite this, the results showed that even under such limitations, several aerodynamic gains can still be achieved with proper definition of the problem. This reinforces the potential of the adopted optimization framework to support and help improving future geometries, taking into account different constraints and design variables.

Overall, this work provided valuable insights regarding aerodynamic shape optimization of fuselages. It contributed to a deeper understanding on how to parametrize different geometries, taking into account the several risks and irregularities that might be obtained if the problem is not properly defined. It also provided knowledge in constraint enforcement, particularly regarding the triangulated surface constraint method, and how these approaches can be combined to produce the best possible solution.

7.2 Future Work

Future studies could include several payloads in the optimization process to represent more complex design challenges. To increase the complexity and explore design flexibility, the optimizer could be given freedom to move the payloads internally, which may provide valuable information regarding the placement of the different payloads within the fuselage. Additionally, to complement the study on TEKEVER ARX, it would be beneficial to use a more detailed fuselage geometry, without the simplifications or smoothing applied in this thesis. Although this would likely add more complexity and challenges to the optimization problem, having a clearer problem definition would bring the research closer to practical conditions. This would allow for comparisons between the software results and real flight data collected by the company.

Additionally, future work could explore how the fuselage interacts with other aircraft components, like wing and tail, by allowing these surfaces to intersect with the fuselage and including their effects in the optimization. An addition of a propeller is also a desired valuable option, since it modifies the flow around that aircraft.

Bibliography

- [1] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7, February 2019. doi: 10.1109/ACCESS.2019.2909530.
- [2] E. A. Kay Wackwitz, Zahra Lotfi, August 2023. URL <https://droneii.com/wp-content/uploads/2023/08/the-drone-industry-barometer-2023.pdf>. Accessed on December 14, 2024.
- [3] Tekever. Tekever AR3 - unmanned aerial vehicle, 2024. URL <https://www.tekever.com/models/ar3/>. Accessed: 2024-02-26.
- [4] B. N. K Karthik Reddy, SK. Kamil Basha and K. Abhilash. The game-changing role of drones in agriculture. *Agri Express*, 02, May 2024. ISSN 2584 - 2498.
- [5] Ö. Ö. Kanat. The significance of unmanned aerial vehicles (UAVs) in strategic contexts. *Anadolu Strateji Dergisi*, 5(2):75–87, 2023.
- [6] S. Islam. Drones on the rise: Exploring the current and future potential of uavs. *arXiv preprint arXiv:2304.13702*, 2023.
- [7] Zion Market Research. Unmanned aerial vehicle (uav) market report, 2024. URL <https://www.zionmarketresearch.com/report/unmanned-aerial-vehicle-uav-market>. Accessed: 2024-02-26.
- [8] Tekever. A comprehensive guide to uavs. URL <https://www.tekever.com/>. Accessed on March 16, 2025.
- [9] MDO Lab, University of Michigan. MACH-aero framework. <https://github.com/mdolab>, 2023. Accessed: May 2025.
- [10] L. K. Loftin. *Quest for performance: The evolution of modern aircraft*. Number 468. Scientific and Technical Information Branch, National Aeronautics and Space Administration, 1985.
- [11] General Atomics Aeronautical Systems. Gray eagle extended range, n.d. URL <https://www.ga-asi.com/remotely-piloted-aircraft/gray-eagle-extended-range>. Accessed: 2024-05-13.

- [12] TopWar.ru. Mq-9 reaper, 2022. URL <https://en.topwar.ru/196920-mq-9-reaper-zhnec-v-nebe-ukrainy.html>. Accessed: 2024-05-13.
- [13] Northrop Grumman. B-2 stealth bomber, n.d. URL <https://www.northropgrumman.com/what-we-do/air/b-2-stealth-bomber>. Accessed: 2024-05-13.
- [14] Airman Magazine. Airframe: The f-117 nighthawk, 2021. URL <https://www.airmanmagazine.af.mil/Features/Display/Article/2604078/airframe-the-f-117-nighthawk/>. Accessed: 2024-05-13.
- [15] McDonnell Douglas f-4 phantom ii. https://tfumux.fandom.com/wiki/McDonnell_Douglas_F-4_Phantom_II, 2024. Transformers Universe MUX Wiki, accessed on 13 May 2024.
- [16] European Airshows. Bristol blenheim first flight, 2023. URL <https://www.europeanairshows.co.uk/aviation-anniversaries/april/bristol-blenheim-first-flight>. Accessed: 2024-05-13.
- [17] L. K. Loftin. *Quest for performance: The evolution of modern aircraft*. Number 468. Scientific and Technical Information Branch, National Aeronautics and Space, 1985.
- [18] J. R. Hansen. *The wind and beyond: a documentary journey into the history of aerodynamics in America*, volume 4409. National Aeronautics and Space Administration, NASA History Office, Office, 2003.
- [19] M. McKinney. Summary of drag clean-up tests in NASA langley full-scale tunnel. In *Kansas Univ. Proc. of the NASA, Ind., Univ., Gen. Aviation Drag Reduction Workshop*, January 1975.
- [20] Army Test and Evaluation Command, Aberdeen Proving Ground, MD. Aerodynamic heating. Technical Report AD0718560, Defense Technical Information Center (DTIC), July 1967. URL <https://apps.dtic.mil/sti/tr/pdf/AD0718560.pdf>. Accessed on March 10, 2025.
- [21] D. P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 2 edition, 2012. ISBN 0-930403-51-7.
- [22] S. Gudmundsson. *General aviation aircraft design: Applied Methods and Procedures*. Butterworth-Heinemann, 2nd edition, April 2022. 978-0128184653.
- [23] R. T. Whitcomb. Technical report, Langley Aeronautical Laboratory Langley Field, Va., October 1953. URL <https://ntrs.nasa.gov/api/citations/20050019402/downloads/20050019402.pdf>.
- [24] D. Ing. S. F. Hoerner. *Fluid-dynamic drag theoretical, experimental and statistical information*. Published by the Author, 1965.
- [25] S. M. Hitzel. Challenges & needs for the understanding of combat aircraft aerodynamics. No. *STO-MP-AVT-307-19*, 2019. Accessed on March 17, 2025.

- [26] J. ter Harmsel. Stealth on aircraft; a literature review. November 2023.
- [27] R. Austin. *Unmanned aircraft systems: UAVS design, development and deployment*. Aerospace series. John Wiley & Sons Ltd, 2011. ISBN 978-0-470-05819-0.
- [28] J. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill series in aeronautical and aerospace engineering. McGraw-Hill Education, 6 edition, 2017. ISBN 978-1-259-12991-9.
- [29] J. D. Anderson. *Aircraft Performance and design*. McGraw-Hill, 2010. ISBN ISBN-13: 978-0-07-070245-5.
- [30] Y. Fan and W. Li. Review of far-field drag decomposition methods for aircraft design. *Journal of Aircraft*, 56(1):11–21, 2019. doi: 10.2514/1.C034781.
- [31] M. Jahanmiri. Aircraft drag reduction: An overview. Research Report 2011:02, Chalmers University of Technology, Göteborg, Sweden, 2011.
- [32] S. Gudmundsson. Chapter 16 - aircraft drag analysis. In S. Gudmundsson, editor, *General Aviation Aircraft Design*, pages 657–752. Butterworth-Heinemann, second edition edition, 2022. ISBN 978-0-12-818465-3. doi: 10.1016/B978-0-12-818465-3.00016-1.
- [33] P. K. Kundu, I. M. Cohen, and D. R. Dowling, editors. *Fluid Mechanics*. Academic Press, fifth edition edition, 2012. ISBN 978-0-12-382100-3. doi: 10.1016/C2009-0-63410-3.
- [34] F. Goetten, M. Havermann, C. Braun, F. Gómez, and C. Bil. *On the Applicability of Empirical Drag Estimation Methods for Unmanned Air Vehicle Design*. 06 2018. doi: 10.2514/6.2018-3192. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2018-3192>.
- [35] M. Aksugur and G. Inalhan. Design methodology of a hybrid propulsion driven electric powered miniature tailsitter unmanned aerial vehicle. *Journal of Intelligent and Robotic Systems*, 57:505–529, 09 2010. doi: 10.1007/s10846-009-9368-0.
- [36] B. W. McCormick. *Aerodynamics, Aeronautics, and Flight Mechanics*. John Wiley & Sons, New York, 2nd edition, 1994. ISBN 978-0471575061.
- [37] L. Kubendran, H. McMahon, and J. Hubbartt. Interference drag in a simulated wing-fuselage juncture. Technical Report NASA-CR-3811, NASA, July 1984.
- [38] FOSSGP Team. Flow analysis of a transonic aircraft configuration, n.d. URL https://fossgrp.com/foss_applications/flow-analysis-of-a-transonic-aircraft-configuration/. Accessed: 2025-04-17.
- [39] T. A. Talay. *Introduction to the Aerodynamics of Flight*. Scientific and Technical Information Office, National Aeronautics and Space Administration, Washington, D.C., 1975.
- [40] E. Houghton, P. Carpenter, S. H. Collicott, and D. T. Valentine. Chapter 8: [viscous flow and boundary layers]. In *Aerodynamics for Engineering Students*. Butterworth-Heinemann, Oxford, UK, 6 edition, 2012. ISBN 978-0-08-096632-8.

- [41] J. H. Ferziger, M. Perić, and R. L. Street. *Computational methods for fluid dynamics*. springer, 4th edition, 2019. ISBN 978-3-319-99691-2.
- [42] J. Vos, A. Rizzi, D. Darracq, and E. Hirschel. Navier–stokes solvers in european aircraft design. *Progress in Aerospace Sciences*, 38(8):601–697, 2002. ISSN 0376-0421. doi: 10.1016/S0376-0421(02)00050-7.
- [43] P. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow*, 21(3):252–263, 2000. doi: 10.1016/S0142-727X(00)00007-2.
- [44] T. large eddies of turbulent motion. Analysis of a scale-similarity model of the motion of large eddies in turbulent flows. *Journal of Mathematical Analysis and Applications*, September 2000. doi: :10.1006/jmaa.2001.7688.
- [45] Y. Zhiyin. Large-eddy simulation: Past, present and the future. *Chinese Journal of Aeronautics*, 28(1):11–24, 2015. ISSN 1000-9361. doi: 10.1016/j.cja.2014.12.007.
- [46] D. Drikakis, M. Hahn, A. Mosedale, and B. Thornber. Large eddy simulation using high-resolution and high-order methods. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1899):2985–2997, July 2009. doi: 10.1098/rsta.2008.0312.
- [47] G. Alfonsi. Reynolds-averaged navier-stokes equations for turbulence modeling. *Applied Mechanics Reviews - APPL MECH REV*, 62, 07 2009. doi: 10.1115/1.3124648.
- [48] C. Hirsch. *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*, volume 1. Elsevier, 2nd edition, 2007. ISBN 978-0-7506-6594-0.
- [49] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc, 3rd edition, November 2006. ISBN 978-1-928729-08-2 (1-928729-08-8).
- [50] N. A. C. Sidik, S. N. A. Yusof, Y. Asako, S. Mohamed, and A. Aziz. A short review on RANS turbulence models. *CFD Letters*, 12:83–96, 11 2020. doi: 10.37934/cfdl.12.11.8396.
- [51] N. Lang and T.-H. Shih. A critical comparison of two-equation turbulence models. Technical report, Institute for Computational Mechanics in Propulsion and Center for Modeling of Turbulence and Transition Lewis Research Center, September 1991.
- [52] A. Aabid, L. Kharulaman, and S. Khan. Analysis of flows and prediction of CH10 airfoil for unmanned arial vehicle wing design. *Advances in Aircraft and Spacecraft Science*, 8:87–109, 06 2021. doi: 10.12989/aas.2021.8.2.087.
- [53] C. A. Mader, G. K. W. Kenway, A. Yildirim, and J. R. R. A. Martins. ADflow—an open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 2020. doi: 10.2514/1.1010796.
- [54] P. Kyratsis. Computational design and digital manufacturing applications. *International Journal of Modern Manufacturing Technologies*, XII(1), 2020. ISSN 2067–3604.

- [55] K.-H. Lee and G.-J. Park. Robust optimization considering tolerances of design variables. *Computers & Structures*, 79(1):77–86, 2001. ISSN 0045-7949. doi: 10.1016/S0045-7949(00)00117-6.
- [56] J. R. Martins and A. Ning. *Engineering design optimization*. Cambridge University Press, 2021. ISBN 9781108833417.
- [57] R. M. Rizk-Allah and A. E. Hassanien. A comprehensive survey on the sine–cosine optimization algorithm. *Artificial Intelligence Review*, 56:1–58, 10 2022. doi: 10.1007/s10462-022-10277-3.
- [58] G. C. Pflug. *Optimization of stochastic models: the interface between simulation and optimization*, volume 373. Springer Science & Business Media, 2012. ISBN 13:978-1-4612-8631-8. doi: 10.1007/978-1-4613-1449-3.
- [59] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011. ISBN 978-1-4614-0237-4.
- [60] P. Krokhmal, M. Zabaranin, and S. Uryasev. Modeling and optimization of risk. *Surveys in Operations Research and Management Science*, 16(2):49–66, 2011. ISSN 1876-7354. doi: 10.1016/j.sorms.2010.08.001.
- [61] C. Permatasari, W. Sutopo, M. Hisjam, et al. Aircraft maintenance manpower shift planning with multiple aircraft maintenance licenced. In *IOP Conference Series: Materials Science and Engineering*, volume 495, page 012023. IOP Publishing, 2019. doi: 10.1088/1757-899X/495/1/012023.
- [62] H. Nakayama, M. Arakawa, and K. Washino. Optimization for black-box objective functions. In *Optimization and optimal control*, pages 185–210. World Scientific, September 2003. ISBN 978-981-238-597-0. doi: 10.1142/9789812775368_0013.
- [63] J. M. Reumschüssel, J. G. R. von Saldern, Y. Li, C. Paschereit, and A. Orchini. Gradient-free optimization in thermoacoustics: Application to a low-order model. *Journal of Engineering for Gas Turbines and Power*, 144, 08 2021. doi: 10.1115/1.4052087.
- [64] C. Thoulon, G. Roge, and O. Pironneau. Gradient-based aero-stealth optimization of a simplified aircraft. *Fluids*, 9(8), 2024. ISSN 2311-5521. doi: 10.3390/fluids9080174.
- [65] Z. Lyu, Z. Xu, and J. R. R. A. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics, Chengdu, Sichuan, China*, volume 11, page 585. Citeseer, 2014.
- [66] S. Koziel and X.-S. Yang, editors. *Computational Optimization, Methods and Algorithms*, volume 356 of *Studies in Computational Intelligence*. Springer, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-20859-1.
- [67] S. Roy and W. Crossley. Hybrid multi-objective combinatorial optimization technique with improved compatibility between ga and gradient-based local search. In *12th AIAA Aviation Technology*,

Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, September 2012. doi: 10.2514/6.2012-5568.

- [68] S. Khosravi and D. W. Zingg. A numerical optimization study on winglets. In *15th AIAA/ISSMO multidisciplinary analysis and optimization conference*, Atlanta, GA, USA, June 2014. doi: 10.2514/6.2014-2173.
- [69] Y. Ma, X. Gao, C. Liu, and J. Li. Improved SQP and SLSQP algorithms for feasible path-based process optimisation. *Computers & Chemical Engineering*, 188:108751, 2024. doi: 10.1016/j.compchemeng.2024.108751.
- [70] M. Widhalm, A. Ronzheimer, and M. Hepperle. Comparison between gradient-free and adjoint based aerodynamic optimization of a flying wing transport aircraft in the preliminary design. In *25th AIAA applied aerodynamics conference*, volume 1, Miami, Florida, USA, 06 2007. doi: 10.2514/6.2007-4060.
- [71] J. E. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010. ISSN 0045-7930. doi: 10.1016/j.compfluid.2009.09.013.
- [72] G. K. Kenway, C. A. Mader, P. He, and J. R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019. ISSN 0376-0421. doi: 10.1016/j.paerosci.2019.05.002.
- [73] A. Jameson. Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*, February 2003. URL <http://aero-comlab.stanford.edu/Papers/jameson.vki03.pdf>.
- [74] G. Corrado, G. Ntourmas, M. Sferza, N. Traiforos, A. Arteiro, L. Brown, D. Chronopoulos, F. Daoud, F. Glock, J. Ninic, E. Ozcan, J. Reinoso, G. Schuhmacher, and T. Turner. Recent progress, challenges and outlook for multidisciplinary structural optimization of aircraft and aerial vehicles. *Progress in Aerospace Sciences*, 135:100861, 2022. ISSN 0376-0421. doi: 10.1016/j.paerosci.2022.100861.
- [75] R. da Silva Gameiro. Aerodynamic design of a mame uav wing using high-fidelity numerical tools. Master’s thesis, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal, Nov. 2023.
- [76] J. Reid and P. He. *Free-Form Deformation (FFD)*. DAfoam, February 2021. URL <https://dafoam.github.io/docs/FFD/main.pdf>. Accessed on May 25, 2025.
- [77] N. R. Secco, G. K. W. Kenway, P. He, C. Mader, and J. R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4):1151–1168, 2021. doi: 10.2514/1.J059491.

- [78] E. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins. pyoptspase: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. doi: 10.21105/joss.02564.
- [79] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 3rd edition, 2015. ISBN 9780080999951.
- [80] MDO Lab. *pyHyp Documentation*. MDO Lab, University of Michigan, USA, 2022. URL <https://mdolab-pyhyp.readthedocs-hosted.com>. Accessed: 2025-04-20.
- [81] L. Eça, F. S. Pereira, and G. Vaz. Viscous flow simulations at high reynolds numbers without wall functions: Is $y^+ \simeq 1$ enough for the near-wall cells? *Computers & Fluids*, 170:157–175, 2018. ISSN 0045-7930. doi: 10.1016/j.compfluid.2018.04.035.
- [82] MDO Lab, University of Michigan. Overset theory — MACH-aero tutorials. https://mdolab-mach-aero.readthedocs-hosted.com/en/latest/machAeroTutorials/overset_theory.html, 2024. Accessed: 2025-05-05.
- [83] G. K. W. Kenway, N. Secco, J. R. R. A. Martins, A. Mishra, and K. Duraisamy. An efficient parallel overset method for aerodynamic shape optimization. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Grapevine, TX, 2017. doi: 10.2514/6.2017-0357.
- [84] Y. Lee and J. Baeder. Implicit hole cutting - a new approach to overset grid connectivity. Orlando, Florida, USA, 06 2003. ISBN 978-1-62410-086-4. doi: 10.2514/6.2003-4128.
- [85] J. A. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001. doi: 10.2514/2.1391.
- [86] A. Hahn. Vehicle sketch pad: A parametric geometry modeler for conceptual aircraft design. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida, USA, January 2010. doi: 10.2514/6.2010-657.
- [87] D. Rodriguez and P. Sturdza. *A Rapid Geometry Engine for Preliminary Aircraft Design*. American Institute of Aeronautics and Astronautics, Reno, Nevada, USA, January 2006. doi: 10.2514/6.2006-929.
- [88] J. A. Samareh. Aerodynamic shape optimization based on free-form deformation. *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 6: 3672–3683, 08 2004. doi: 10.2514/6.2004-4630.
- [89] M. Wentrup. An adjoint based optimization chain for complex helicopter fuselage parts using a free form deformation or cad based parameterization method. In *41st European Rotorcraft Forum*, Munich, Germany, September 2015.
- [90] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, September 2010. AIAA 2010-9231.

- [91] N. R. Secco, J. P. Jasa, G. K. W. Kenway, and J. R. R. A. Martins. Component-based geometry manipulation for aerodynamic shape optimization with overset meshes. *AIAA Journal*, 56(9):3667–3679, 2018. doi: 10.2514/1.J056550.
- [92] E. Ekici and M. P. Juniper. Adjoint based shape optimization for thermoacoustic stability of combustors using free form deformation. In *Turbo Expo: Power for Land, Sea, and Air*, volume 87943, page V03AT04A058, London, England, United Kingdom, June 2024. American Society of Mechanical Engineers.
- [93] H. Zhao, J. T. Hwang, and J.-S. Chen. Open-source shape optimization for isogeometric shells using fenics and OpenMDAO, 2024. URL <https://arxiv.org/abs/2410.02225>.
- [94] L. A. Schwarz. Non-rigid registration using free-form deformations. Master’s thesis, Technical University of Munich (TUM), Germany, May 2007.
- [95] H. M. Hajdik, A. Yildirim, E. Wu, B. J. Brelje, S. Seraj, M. Mangano, J. L. Anibal, E. Jonsson, E. J. Adler, C. A. Mader, G. K. w. Kenway, and J. R. R. A. Martins. pygeo: A geometry package for multidisciplinary design optimization. *Journal of Open Source Software*, 8(87):5319, 2023. doi: 10.21105/joss.05319.
- [96] M. Z. bin Md Shah, M. R. bin Abu Bakar, and B. Basuno. The aerodynamics analysis on cambered fuselage model. In *Advances in Mechanical, Materials and Manufacturing Engineering*, volume 660 of *Applied Mechanics and Materials*, pages 492–497. Trans Tech Publications Ltd, 11 2014. doi: 10.4028/www.scientific.net/AMM.660.492.
- [97] K. D. Harris. An experimental investigation of the subsonic drag and pitching moment characteristics of slender cambered bodies with pointed noses and tails. Technical report, College of Aeronautics, Cranfield University, September 1958.
- [98] B. J. Brelje, J. L. Anibal, A. Yildirim, C. A. Mader, and J. R. R. A. Martins. Flexible formulation of spatial integration constraints in aerodynamic shape optimization. *AIAA Journal*, 58(6):2571–2580, 2020. doi: 10.2514/1.J058366.
- [99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, second edition, 2006. ISBN 978-0387303031.
- [100] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40(21):3979–4002, 1997. doi: /10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.
- [101] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann, San Francisco, CA, 2004. ISBN 978-1558607323.
- [102] G. K. W. Kenway and J. R. R. A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014. doi: 10.2514/1.C032150.

- [103] G. Kennedy and J. Martins. A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elements in Analysis and Design*, 87: 56–73, 09 2014. doi: 10.1016/j.finel.2014.04.011.
- [104] T. Möller. A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 2(2):25–30, 1997. doi: 10.1080/10867651.1997.10487472.
- [105] E. Haines. Point in polygon strategies. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 24–46. Academic Press, 1994. ISBN 978-0123361554.
- [106] C. B. Sullivan and A. Kaszynski. PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, 4(37):1450, may 2019. doi: 10.21105/joss.01450.
- [107] S. Shubham, N. Wright, and A. Ianakiev. Application of Richardson extrapolation method to aerodynamic and aeroacoustic characteristics of low Reynolds number vertical axis wind turbines. In *28th AIAA/CEAS Aeroacoustics 2022 Conference*, Southampton, UK, June 2022. doi: 10.2514/6.2022-3022.
- [108] A. Meana-Fernández, J. M. F. Oro, K. M. A. Díaz, M. Galdo-Vega, and S. V.-S. and. Application of Richardson extrapolation method to the CFD simulation of vertical-axis wind turbines and analysis of the flow field. *Engineering Applications of Computational Fluid Mechanics*, 13(1):359–376, 2019. doi: 10.1080/19942060.2019.1596160.

