Master thesis and internship[BR]- Master's thesis : Development of a preliminary aircraft design optimization framework[BR]- Integration internship

Auteur : Van Den Berghe, Jérome

Promoteur(s) : Terrapon, Vincent

Faculté : Faculté des Sciences appliquées

Diplôme : Master en ingénieur civil en aérospatiale, à finalité spécialisée en "aerospace engineering"

Année académique : 2024-2025

URI/URL : http://hdl.handle.net/2268.2/23387

Université de Liège - Faculté des Sciences Appliquées

Development of a preliminary aircraft design optimization framework

Travail de fin d'études réalisé en vue de l'obtention du grade de master
"Ingénieur Civil en Aérospatiale" par

Jérome Van Den Berghe

Promoteur : Prof. Vincent E. Terrapon

Année académique 2024-2025

Abstract

Traditional aircraft design methods rely on iterative procedures with limited automation and weak coupling between disciplines, resulting in a time-consuming process that can lead to suboptimal designs. Multidisciplinary Design Optimization (MDO) emerges as a powerful alternative to overcome these limitations.

This thesis presents the development of RAPTOR (Rapid Aircraft Preliminary Optimization and Refinement), a modular, open-source framework tailored for early-stage aircraft design. RAPTOR addresses the shortcomings of conventional approaches by integrating a fully multidisciplinary optimization architecture built on top of OpenMDAO, enabling efficient coordination between geometry, aerodynamics, structures, propulsion, stability, weight estimation, and performance evaluation. The framework supports rapid configuration and refinement of both conventional and unconventional aircraft through gradient-based optimization and multi-fidelity analyses.

Validation is carried out through benchmark studies and optimization cases, including a tailless unmanned aerial vehicle (UAV) and a single-seat aerobatic aircraft, illustrating the ability of RAPTOR to handle diverse aircraft configurations.

Acknowledgements

I would like to express my gratitude to my advisor, Prof. Vincent Terrapon, for his attentive guidance and steady support throughout this thesis. His thoughtful remarks created an inspiring environment that made this project truly enriching. I always felt encouraged to explore ideas freely and appreciated being treated as a full member of the research team.

I would also like to sincerely thank my supervisor, Paul Dechamps, for his constant help and availability. His technical knowledge and practical approach really helped me move forward, especially when things got complicated. Our many discussions were always useful and gave me new ideas when I didn't know how to continue.

Finally, I want to thank my family, my close friends, and my girlfriend for always being there for me this year. Their support, kindness, and patience really helped me stay motivated, even during the more stressful times.

Declaration of Artificial Intelligence Usage

This document includes the use of ChatGPT, primarily to rephrase certain passages and enhance the clarity of the language. The AI tool served only as a writing assistant. All ideas and content are entirely the author's own. Its use was in full compliance with the faculty's guidelines on responsible AI usage.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Context and motivations

Over the past few decades, the commercial aviation sector has experienced a steady and significant increase in global demand. This growth has placed considerable pressure on aircraft manufacturers to improve aircraft capabilities and performances, and reduce operational costs, as documented by Airbus [3]. Historically, several key performance metrics have shown marked evolution, among them the thrust-to-weight ratio, as illustrated in Figure 1.1. This parameter is commonly used as a global indicator of aircraft performance. It reflects the aircraft's ability to accelerate and climb, and higher values are generally associated with better overall performance, as highlighted by NASA [61].

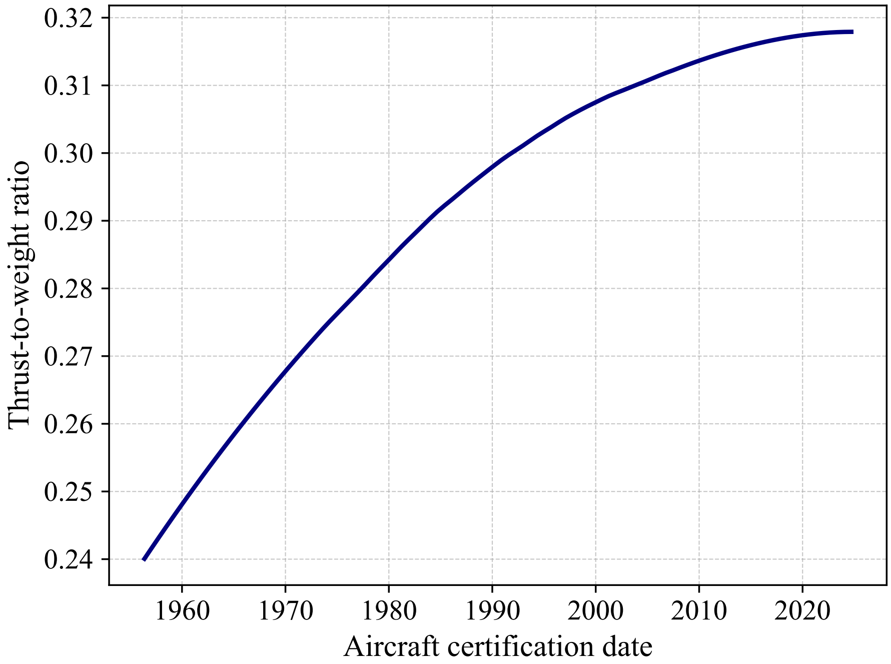


Figure 1.1: Historical trends in thrust-to-weight ratios for aircraft, reproduced from Acar et al. [2].

Although a clear upward trend in thrust-to-weight ratio is observed throughout much of the twentieth century, the data also reveal a noticeable stagnation in more recent years. For newly certified aircraft, the thrust-to-weight ratio has plateaued, indicating that traditional methods for performance improvement may be reaching their limits, as explained by Acar et al. [2]. Although this flattening can be partially attributed to the increasing payload weight, it also underscores the growing difficulty of achieving further gains through conventional ways. In this context, it becomes essential to explore innovative design strategies that can unlock new sources of performance.

In traditional methods, the aircraft design process is typically divided into three distinct stages, as outlined by Grihon [24]. Although aircraft design is inherently multidisciplinary [79], the design process is generally handled by separate engineering teams assigned to individual disciplines, leading to weak coupling between domains and limited cross-disciplinary interaction, as described by Lucio [43]. Another major limitation of this approach is its reliance on iterative trade-off procedures, which are rarely automated and often time-consuming. In the following sections, these limitations are further discussed, and alternative strategies are introduced.

The aircraft design process begins with the conceptual design phase, which is strongly driven by initial specifications and mission requirements. According to Raymer [72], this phase involves high-level decisions that define the overall configuration of the aircraft, such as the propulsion type (jet or propeller) and the empennage layout (conventional or T-tail). A preliminary sketch of the aircraft is developed at this stage using a combination of empirical formulas, statistical methods, and simplified models. These methods are widely documented in references such as Torenbeek [87], Gudmundsson [26], and the DATCOM handbook [18]. The configuration is progressively refined through trade-off studies and iterations on key parameters to satisfy the mission constraints and design targets. While conceptually straightforward, this iterative process can be long and resource-intensive, even at this early stage of development. To assist designers in building the initial aircraft layout, several frameworks have been proposed. For instance, RADE [9], developed at the Georgia Institute of Technology, offers a modular and parametric environment for early-stage structural analysis. CASDAT [29], developed under the supervision of the U.S. Air Force, supports multidisciplinary analysis at the conceptual level.

Once the main configuration is established, the project enters the preliminary design phase. At this stage, higher-fidelity models are introduced to analyze and refine the aircraft's geometry, performance, and systems integration. For aerodynamic simulations, CFD tools such as OpenFOAM [68] or SU2 [83] are frequently used. Geometry modeling and mass estimation rely on precise CAD environments such as OpenVSP [69] or Siemens-NX [81]. The design remains iterative, as engineers evaluate performance metrics, compare them against targets, and adjust design parameters accordingly until requirements are satisfied. As in the conceptual phase, traditional preliminary design is often conducted by separating disciplinary engineering teams, which limits inter-domain coupling and coordinated trade-offs. Several tools are also available to support analysis at this stage. AAA (Advanced Aircraft Analysis) [11], developed by DARcorporation and based on Roskam's foundational texts [75], is widely used for performance evaluation and sizing. CASTOR [71] is another well-established software platform for preliminary aircraft design.

Finally, the detailed design phase concludes the process. This stage involves the refinement of every component, the execution of the certification process, and the construction of a physical prototype, as described by Mattos et al. [51].

In this traditional workflow, a first major limitation lies in the fact that the conceptual design phase typically explores only very simplified configurations, as pointed out by Eisler [15]. In contrast, the preliminary design stage directly shifts to higher-fidelity methods, often without an optimization of the initial configuration. Introducing an optimization step during the conceptual phase could provide a smoother and transition to the subsequent stages, while reducing both development time and cost. As emphasized by Tian et al. [86], early design space exploration in the conceptual phase has a significant impact on the preliminary and detailed design stages, as well as on the overall life-cycle cost and operational effectiveness of the aircraft.

Moreover, as explained above and highlighted by Lucio [43], traditionally, aircraft design processes are organized around separate disciplinary teams, each focusing on specific areas such as aerodynamics, propulsion, structures, or stability. This distinct structure inherently limits the coupling and interaction between disciplines. However, aircraft design is intrinsically multidisciplinary, involving domains that are highly interdependent, as said by Schwinn [79]. As Torenbeek [87] points out, the complex interactions between these disciplines make it nearly impossible to derive a closed-form solution, and coordination between them becomes essential. Without such coordination, decisions made in isolation by one team can degrade overall system performance. As emphasized by Nicolai and Carty [63], this separation often results in local optima rather than a global optimum. For example, minimizing the structural weight may lead to an internal configuration that compromises strength, unless this decision is balanced with structural constraints.

In addition to the disciplinary separation, conventional design methods rely on a sequential, manual, and iterative process. As described by Martins and Ning [50], engineers typically begin with an initial configuration, evaluate its performance, apply trade-offs, and repeat this process until a satisfactory solution is found. While this approach can eventually yield a feasible design, it is inherently time-consuming and heavily dependent on the engineer's intuition and experience. As Olds [67] notes, the outcome of this process is sensitive to the range of parameters chosen and to which ones are kept fixed throughout the iterations. Saves et al. [78] further highlight that realistic aircraft design problems involve a very large number of design variables. Hajela [28] points out that traditional techniques struggle to manage such high-dimensional design spaces, and thus limit the ability to explore the full potential of modern configurations.

To support this sequential design methodology, aircraft performance analysis tools are required. The evolution of the AAA [11] software previously introduced, General Aviation Computer Aided Design (G.A.-CAD) environment, combined analytical calculations with a graphical drafting interface [76]. Another well-known tool is RDS [73], created by Raymer and distributed by the Conceptual Research Corporation, widely used in conceptual design studies.

In contrast to the iterative, manually driven approach, optimization-based methods aim to automate the design process. According to Martins and Ning [50], using such methods moves the iteration process from the engineer to the automated algorithm. While human input is still required to define the problem and constraints, the optimization loop itself is executed automatically. Given the multidisciplinary nature of aircraft design, these methods are in fact in the category of Multidisciplinary Design Optimization. As Sadrehaghighi [77] explains, the increasing complexity of modern engineering systems has intensified interest in MDO techniques. Though organizationally complex, these approaches offer major benefits in reducing labor, human error, and development time [15]. More importantly, as Giesing and Barthelemy [22] argue, MDO is more likely to identify the best global compromise across all disciplines, rather than a local optimum. Applying MDO in early-stage design also improves overall aircraft quality and reduces cost downstream [49], which means that the use of MDO at the end of the conceptual design stage to make a first optimization of the layout before the preliminary design would be very efficient.

Numerous frameworks now support MDO in aircraft design. On the commercial side, ADS (Aircraft Design Software) [70], developed by OAD, offers validated optimization capabilities for a wide range of configurations. ACSYNT [59], originally developed at NASA Ames, remains a modular tool for analyzing and optimizing diverse aircraft types. OPDOT [82], another NASA initiative, is one of the early optimization frameworks incorporating sizing and mission performance. TASOPT [14], from MIT, is based on low-order physics models rather than empirical correlations, making it particularly well-suited for non-conventional aircraft configurations. Piano [41], developed by Lissys Ltd., is tailored for commercial subsonic transport aircraft and is widely used in industry. PrADO (Preliminary Aircraft Design and Optimization) [6]is well adapted for the conceptual design of civil subsonic aircraft. In parallel, more recent open-source initiatives have emerged. SUAVE [84], developed at Stanford University, supports modular, multi-fidelity analysis across a wide array of aircraft architectures. FAST-OAD-GA [44], developed by ISAE-SUPAERO and ONERA, provides a flexible environment for optimization at the conceptual level. NASA's Aviary [60] project aims to provide a unified and extensible MDAO framework tailored to modern aerospace design needs.

To summarize, traditional aircraft design methods suffer from several limitations: they are inherently iterative, non-automated, lack cross-disciplinary coordination, and impose a rigid separation between the conceptual and preliminary design stages. Many of these challenges can be at least partially addressed through the use of multidisciplinary optimization techniques, as demonstrated by the various frameworks reviewed above.

However, despite their potential, these frameworks also present several drawbacks. As highlighted by Hosseini et al. [32], some of them are not open source, requiring paid licenses or being entirely inaccessible. Others are no longer actively maintained, which limits their applicability to current research or industrial needs. In many cases, these frameworks are also restricted to conventional aircraft configurations, with fixed component architectures that limit their use for unconventional or novel concepts. This constraint often results from a lack of modularity in the code architecture. Another common issue is the lack of flexibility: extending or modifying the framework, such as adding a new aerodynamic model or integrating a different fidelity level can be difficult. For instance, in a truly flexible environment, a developer wishing to implement a panel method to evaluate the aerodynamic performance of a wing should be able to do so with minimal difficulty, seamlessly plugging the new model into both the analysis and optimization workflows. This type of extensibility remains limited in many existing tools. Another important limitation observed in many existing frameworks is the lack of support for variable fidelity levels. In this cases, the fidelity is fixed, leaving the user unable to select the appropriate level of accuracy for a given task. This can be problematic: for rapid optimization studies, low-fidelity models are often sufficient and computationally efficient, whereas higher-fidelity models are preferred when more precise results are required. Ideally, a design environment should offer multi-fidelity capabilities, allowing the user to switch between or combine different levels of fidelity depending on the phase of the design process and the objectives of the analysis.

In light of these limitations, there is a clear need for a new aircraft design framework that integrates all the key capabilities discussed above, that are not simultaneously offered by the existing tools. This paper therefore presents the development of the RAPTOR framework (Rapid Aircraft Preliminary Optimization & Refinement), designed to address these gaps through a modular, open-source, and optimization friendly architecture.

1.2 Positioning of the RAPTOR Framework

Regarding the deficiencies identified in existing frameworks, whether related to modularity, fidelity, openness, extensibility, or optimization capability, the development of RAPTOR was conducted with the clear objective of addressing these issues through a robust, flexible, and open-source architecture. Rather than adapting an existing tool, RAPTOR was designed from the ground up following a structured and methodical approach, placing a strong emphasis on modularity and ease of extension. The aim is to provide an environment for early stage aircraft design optimization, while ensuring sufficient usability and adaptability for variable use. To realize these objectives, a number of foundational architectural decisions were made. These decisions concern the general structure of the framework, the separation of responsibilities between modules, the handling of model couplings, and the integration of optimization capabilities. Each of these elements is presented in the following developments.

A fundamental step in any optimization procedure is the selection of an appropriate optimization algorithm. As highlighted by Farag et al. [16], optimization techniques play a key role in enhancing efficiency of the procedure. In the context of aircraft design, this choice is particularly important due to the complexity and scale of the problem. As discussed in more detail in a dedicated section of this report, a wide range of algorithms are available, and summarized by Rhinehart [74]. However, the nature of the problem makes the choice between gradient-free and gradient-based approaches relatively straightforward. As described by Jaeger et al. [35], aircraft design optimization typically involves a very large number of design variables parameters that are adjusted during the optimization to improve the overall design. Moreover, several constraints should be used to formulate the optimization procedure.

In such high-dimensional constrained problems, as highlighted by Kim [38], gradient-based methods are generally far more efficient and scalable than gradient-free alternatives.

Furthermore, aircraft design problems are inherently constrained, as emphasized by Hoburg and Abbeel [31]. Stability, structural integrity, and performance targets all translate into nonlinear constraints that must be satisfied throughout the optimization process. To address such constraints effectively, the SLSQP (Sequential Least Squares Programming) method has been adopted. SLSQP belongs to the family of Sequential Quadratic Programming (SQP) algorithms, which are particularly well suited for constrained nonlinear optimization problems frequently encountered in engineering, as highlighted by Martins [48]. Compared to other gradient-based methods such as BFGS [19], L-BFGS-B [42], conjugate gradient (CG), Newton-CG, or truncated Newton methods [65], SLSQP offers broader capabilities. Most of these alternatives either do not support general constraints or are limited to bound-constrained optimization. This makes them unsuitable for the complex, multidisciplinary problems involved in aircraft design. In contrast, SLSQP can handle general equality and inequality constraints, making it a more robust and appropriate choice for the types of optimization problems tackled in this work. A more detailed justification of this selection is presented in the dedicated optimization section of the report.

In gradient-based optimization, an essential consideration is the method used to compute the gradients of the objective and constraint functions with respect to the design variables, as highlighted by Ghate [21]. Several approaches exist, each offering trade-offs between accuracy, computational efficiency, and implementation complexity. In the development of the RAPTOR framework, particular emphasis has been placed on the use of automatic differentiation, which provides an accurate, efficient, and systematic means of computing exact derivatives, as explained by Grosse [25]. Automatic differentiation is based on the application of the chain rule and can be implemented in two main modes: forward and reverse. For problems such as aircraft design, where the number of input variables (design parameters) typically exceeds the number of outputs (objectives or constraints), reverse mode AD is generally the more efficient choice, as precised by Martins and Ning [50]. This decision is justified in more detail in the dedicated section of this report. While the most accurate method remains the manual (hand) derivation of gradients, Halim and Yeak [30] points out that it is extremely time-consuming, as it requires to develop the analytical expression of all the gradients. At the other end of the spectrum, finite difference methods are simple to implement but suffer from poor numerical accuracy, as precised by Xue et al. [88] and high computational cost, particularly in high-dimensional design spaces. For this reason, finite differencing is only used in specific cases where no other alternative is available.

Once the optimization algorithm and the gradient computation strategy are selected, it becomes essential to ensure proper coupling between the different disciplinary models, both in terms of variable dependencies and derivative propagation. To address this challenge, the RAPTOR framework is built on top of OpenMDAO [23], an open-source platform specifically designed for multidisciplinary optimization developed by NASA. OpenMDAO is particularly well suited for integrating models from different disciplines, such as aerodynamics, weight estimation, stability, and propulsion, by enabling the user to define how outputs from one model serve as inputs to another. For instance, the center of gravity position computed by the weight estimation module is required by the stability analysis module to evaluate the aircraft stability. OpenMDAO provides a natural and scalable interface to manage such interdependencies across models, as precised by Backhaus et al. [5]. Beyond variable connections, OpenMDAO also supports the propagation of derivatives across the entire multidisciplinary system. This is crucial for obtaining accurate gradients of the objective function with respect to the design variables. Using its well-established coupled adjoint method, documented by Martins and Ning [50], OpenMDAO enables efficient and automated computation of total derivatives. In practice, the developer only needs to implement the disciplinary models and supply partial derivatives of outputs with respect to inputs at the local level. OpenMDAO then, as described by Kaiyoom et al. [1], handles the assembly and propagation of these derivatives throughout the coupled system, significantly reducing implementation complexity and minimizing human error. This choice to build RAPTOR on top of OpenMDAO is therefore optimal to perform multidisciplinary analysis and optimization procedures, as the coupling between the different disciplines is straightforward.

Once the overall structure of the optimization procedure and the strategy for coupling the different disciplinary models have been defined, key architectural choices concerning the software framework itself can be made. Among the core capabilities required for RAPTOR, modularity stands out as a primary objective. The goal is firstly to allow the user to consider the disciplines he desires in the optimization framework. To do so, the models are decomposed by disciplines, such as aerodynamics, weight, stability and more. The framework must also be capable of handling both conventional and unconventional configurations. For example, it should be able to analyze and optimize a tailless aircraft, meaning that the optimization workflow must remain functional even in the absence of empennage components. To achieve this, the disciplinary models are organized into modular components, each corresponding to a specific aircraft feature. For instance, aerodynamic performance is evaluated through separate modules for the wing, fuselage, and tail, allowing the user to include or exclude components as needed. In addition, RAPTOR has been designed with multi-fidelity analysis in mind. This means that modularity is implemented not only across physical components (e.g., wing vs. fuselage) but also across modeling approaches. For example, the fuselage weight can be estimated using either an analytical method or a CAD-based approach, each represented by a distinct component within the framework. Another essential requirement is flexibility. From a developer's perspective, it must be straightforward to integrate a new model and use it directly within the analysis or optimization workflow. The separation of components by both aircraft physical components and modeling methods supports this flexibility by making the internal structure extensible and transparent. Moreover, thanks to the integration with OpenMDAO, developers are only required to implement the internal logic of their models and supply partial derivatives of outputs with respect to inputs. OpenMDAO then handles the assembly and propagation of variables and derivatives across the multidisciplinary system automatically, significantly reducing implementation effort. These capabilities enable RAPTOR to handle both conventional and unconventional aircraft configurations, while supporting multiple levels of modeling fidelity. Furthermore, its flexible architecture ensures that the range of aircraft types it can optimize is not fixed, but can evolve over time as new models and disciplines are integrated into the framework.

Figure 1.2 provides a visual summary of this modular structure, showing how models are subdivided both by disciplines, aircraft components and by the level of modeling fidelity.

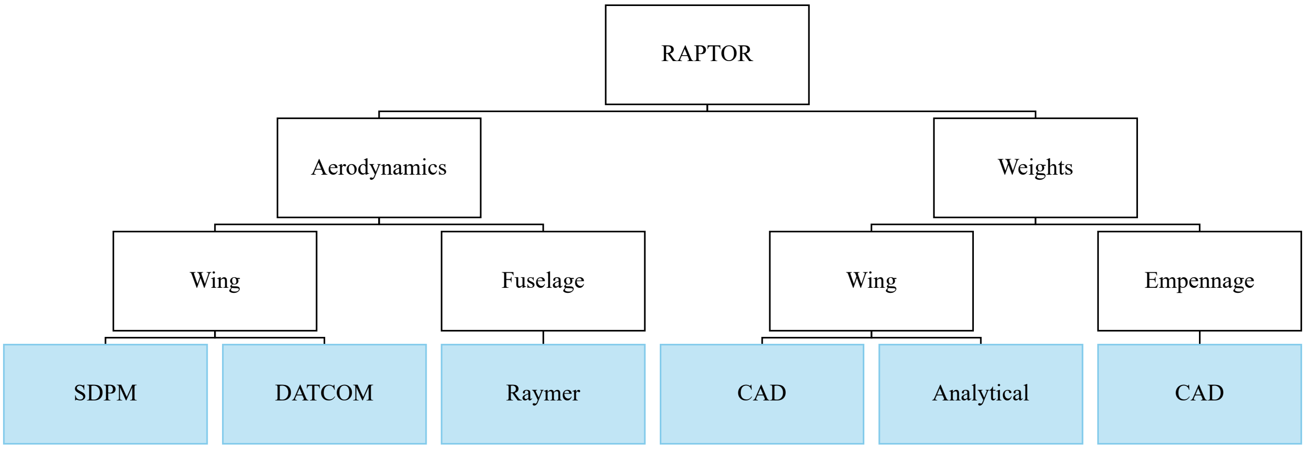


Figure 1.2: Example illustrating the hierarchical modular architecture of the RAPTOR framework. Disciplines are decomposed by aircraft components, each of which supports multiple modeling approaches with varying levels of fidelity.

Once the architectural philosophy of the framework has been established, the next step involves selecting and implementing the appropriate models to assess aircraft performance. These models must not only cover key disciplines such as aerodynamics, structures, weight estimation, stability, and propulsion, but also be compatible with the optimization framework. In particular, to enable the use of automatic differentiation, all models must be written in a fully differentiable manner. To satisfy this requirement, the low-fidelity models are based on analytical or empirical formulations, carefully adapted to ensure continuity and differentiability of the mathematical expressions. To model the geometry, both a CAD option based on OpenVSP [69], and a user-defined model are available. In the aerodynamic domain, the low-fidelity models implemented in RAPTOR draw on well-established sources such as Raymer [72], Torenbeek [87], DATCOM [18], and Roskam [76]. For higher-fidelity aerodynamic analysis, RAPTOR integrates a panel method based on the Source-Doublet Panel Method (SDPM) developed by Crovato [10], and presented by Sanchez Martinez and Dimitriadis [47]. For weight estimation, the low-fidelity model approximates internal structural volumes of each component, while the high-fidelity alternative uses a CAD-based evaluation pipeline by coupling the framework with OpenVSP [58]. Static stability analysis includes both a semi-empirical model for longitudinal stability, based on DATCOM [18], and a higher-fidelity variant also derived from the SDPM aerodynamic model. Lateral stability is evaluated through a model grounded in Roskam's methodology [76]. Structural design modeling involves the construction of the aircraft's flight envelope, identification of critical load cases, and internal wing structure calculations, based on the reference work of Megson [53]. Finally, overall aircraft performance and propeller-based propulsion modeling are handled using methods described by Raymer [72].

The following section presents a detailed review of existing aircraft design frameworks, including both analysis and optimization tools. Each framework is described, and then compared to RAPTOR in order to highlight the specific contributions and innovations introduced by the present work.

1.3 State of the art

Over the years, numerous aircraft design tools have been developed, ranging from industrial-grade commercial software to modern open-source frameworks. This section reviews a selection of these tools and highlights their similarities and differences with the present work, RAPTOR.

One of them is AAA [11], a commercial software developed by DARcorporation in 1990, based on Roskam [75] books. The Advanced Aircraft Analysis framework focuses primarily on the analysis and sizing of conventional fixed-wing aircraft using empirical and semi-empirical methods. While it provides detailed and reliable performance assessments, as well as a user-friendly interface AAA does not include capabilities for design optimization. In contrast, the RAPTOR framework developed in this study is fully open-source and integrates both analysis and optimization within a unified, modular environment. This makes it particularly suited for early-phase design iterations, where trade-offs between disciplines must be explored rapidly. Despite their differences, one key similarity between AAA and RAPTOR lies in their modularity. Both tools adopt an architecture that enables users to analyze a wide variety of configurations, from conventional aircraft to more exotic layouts.

A compact commercial solution aimed at early design studies is ADS [70] (Aircraft Design Software), developed by the OAD company. It is capable of analyzing a broad range of aircraft performance aspects and includes functionalities for performing optimization tasks. The key distinction lies in the fact that RAPTOR is an open-source framework, allowing users to more easily modify or replace performance models to suit their specific needs. Furthermore, RAPTOR is entirely free to use, whereas conducting conceptual aircraft design with commercial software can entail significant costs.

Among the widely used professional solutions, RDS (Raymer Design Software) is a well-established software developed by Daniel P. Raymer in 1992 and distributed by the Conceptual Research Corporation. It is available in both student and professional versions. According to Raymer [73], the professional version supports the entire conceptual design process, including the modeling of the aircraft configuration, performance analysis across several disciplines, and subsequent optimization of the design. It also contains its own CAD modeling system. Like RAPTOR, it features a multidisciplinary optimization approach, allowing different aspects of the aircraft to be considered simultaneously during the design process. The main distinction lies in the fact that RAPTOR is open-source. This gives the user full access to the source code and the ability to modify or extend more easily the framework.

Continuing with commercial solutions, ACSYNT [59] (Aircraft Synthesis) is developed for the conceptual design and performance evaluation of aircraft. It offers capabilities for sizing, mission and performance analysis, as well as multidisciplinary optimization. The tool is highly modular in terms of the types of aircraft it can handle, supporting a wide range of configurations, and is structured to accommodate multiple disciplines such as aerodynamics, propulsion, structures, and mission analysis. The models integrated within ACSYNT are generally well-established and sophisticated, making it a powerful platform for advanced design studies. However, the most significant distinction compared to RAPTOR lies in its closed-source nature. As a proprietary software, ACSYNT is not freely available, and its internal models cannot be easily modified or extended by the user. While the framework does exhibit a certain level of modularity, this is constrained by its commercial architecture and implementation. In addition, due to its technical depth and the complexity of its configuration, a considerable learning effort is often required to use ACSYNT efficiently. In contrast, the RAPTOR framework developed in the present work is designed to be lightweight, open, and more accessible to new users, with an emphasis on flexibility and ease of integration for custom solvers and models.

Developed under the authority of the U.S. Air Force, CASDAT [29] is a conceptual aircraft design and analysis framework. It relies on the integration of existing legacy tools to perform multidisciplinary analyses such as aerodynamics, propulsion, weights, and mission performance. In contrast, RAPTOR is more oriented toward the development and customization of models, offering greater flexibility for modifying or extending the analysis capabilities. Additionally, unlike RAPTOR, CASDAT does not include an integrated optimization process. However, both frameworks share common features, notably a modular architecture and a multidisciplinary design philosophy.

One of the earliest examples of a multidisciplinary design tool is OPDOT [82], developed by NASA in the 1980s to support the preliminary sizing of transport aircraft. Like RAPTOR, it follows a multidisciplinary design optimization approach, where key aircraft characteristics such as geometry, mass, and performance are optimized under various constraints. Although developed several decades earlier, its structure already reflects a modular philosophy, with separate subroutines for each technical discipline. However, there is a significant difference, as RAPTOR is open-source, while OPDOT is not publicly available or designed for broad user accessibility.

Focusing specifically on commuter aircraft, CASTOR [71] is a program developed in the 1990s for the preliminary design of twin-engine, propeller-driven configurations. The goal is to find a configuration that performs optimally across a realistic mission scenario, making CASTOR an early example of integrated aircraft and trajectory design. While RAPTOR shares this multidisciplinary philosophy, there are notable differences. First, RAPTOR is built on OpenMDAO, providing a more flexible and modern architecture. Its modular structure allows users to replace or upgrade discipline models, and to define custom optimization problems with varying levels of fidelity. Furthermore, RAPTOR is open-source and not restricted to conventional layouts or propulsion types, making it more adaptable to novel configurations or applications.

A physics-based approach to system-level optimization is proposed by TASOPT (Transport Aircraft System OPTimization), an aircraft design and optimization tool developed at MIT [14]. Unlike traditional methods that rely heavily on empirical correlations, TASOPT is built on low-order physical models, making it particularly well-suited for evaluating unconventional architectures with greater fidelity. However, due to its close-source nature, modifying or extending the code, such as replacing a disciplinary model or adding a new one can be challenging for users. In contrast, RAPTOR is open-source, and therefore designed to facilitate this kind of modification.

Used extensively for performance and emissions analysis in industry, Piano [41] is a proprietary software tool developed by Lissys Ltd for the preliminary sizing of aircraft. The software is specifically tailored to the analysis of conventional commercial subsonic aircraft, from small business jets to the largest wide-body airliners. In contrast, the RAPTOR framework is designed to accommodate a broader range of configurations, including unconventional and non-standard layouts.

LAMBDA [32] is a modular aircraft design and optimization framework developed at Amirkabir University of Technology. It integrates a wide range of disciplinary modules, from aerodynamics to structural finite element method analysis and cost estimation. The framework emphasizes flexibility, allowing users to substitute or extend built-in methods with custom scripts, and interfaces with high-fidelity tools such as CATIA [12], AVL [13], and Nastran [54]. However, unlike open-source solutions such as RAPTOR, the LAMBDA framework is not publicly released and remains proprietary. Despite featuring an open architecture internally, the code is not distributed as open source, which limits accessibility and collaborative development.

Another early and influential example is the PASS framework (Program for Aircraft Synthesis Studies) [39], developed at Stanford University. Conceived in the late 1980s, PASS was designed as a modular system for conceptual aircraft design, integrating a wide range of disciplines including cost estimation. Its novelty at the time lay in its use of artificial intelligence principles, particularly knowledge-based and quasi-procedural logic, to guide design decisions. The architecture allowed designers to plug in modules with different levels of fidelity, depending on the design phase or objective. While not actively maintained today, PASS laid conceptual groundwork for later frameworks by emphasizing flexibility, AI integration, and modularity.

Among the early modular frameworks, PrADO (Preliminary Aircraft Design and Optimization) [6] was developed at the Technical University of Braunschweig in the early 1990s. Initially tailored for hypersonic vehicle studies, it was later adapted for the conceptual design of civil subsonic aircraft. PrADO offers a comprehensive set of modules, all integrated within a Fortran-based environment. While its modular architecture allows for some flexibility, PrADO is not open source. In contrast, RAPTOR distinguishes itself by being open source.

Another notable example is QCARD (Quick Conceptual Aircraft Research and Development) [34], developed at KTH Royal Institute of Technology. The framework was designed to support real-time interaction through a graphical user interface. QCARD is not open source, while RAPTOR not only provides offers a modern open-source code, enabling easier models customization.

Developed by Boeing, MDOPT (Multidisciplinary Design Optimization) [40] is a high-fidelity MDO framework designed to optimize air vehicles at the conceptual and preliminary design stages. A distinctive feature of MDOPT is its integration of Computational Fluid Dynamics (CFD) simulations within the optimization loop, enabling more accurate aerodynamic analysis compared to low-fidelity methods. The system also includes a graphical user interface (GUI) to facilitate input handling and scenario exploration. However, unlike open-source frameworks such as RAPTOR, the internal structure and source code of MDOPT are not publicly available, and the framework remains proprietary.

PreSTo (Preliminary Sizing Tool) [80], developed by Seeckt and Scholz at the Hamburg University of Applied Sciences, is a conceptual design framework implemented entirely through Microsoft Excel worksheets. It includes modules for geometry definition, sizing, aerodynamics, mass estimation, propulsion, cabin layout, stability and control, landing gear, cost, and performance analysis. The spreadsheet-based structure significantly reduces pre-processing time and enhances accessibility, making it particularly suited for early design studies and educational use. However, due to its simplified formulation and lack of integration with external solvers or simulation tools, it is not well suited for high-fidelity workflows. In contrast, RAPTOR, is open source and supports scalable analysis across fidelity levels and is more suitable for advanced optimization.

VAMPzero [7] is a conceptual aircraft design framework developed by the German Aerospace

Center (DLR). It is built around the CPACS (Common Parametric Aircraft Configuration Schema) standard [57], which serves as a centralized data model for storing, sharing, and communicating aircraft configuration data between various tools and teams. The use of CPACS enables seamless integration in distributed, collaborative environments, supporting workflows that span different levels of modeling fidelity. VAMPzero has been integrated within larger MDAO toolchains, offering compatibility with high- and low-fidelity modules. In contrast to RAPTOR, which is open-source, VAMPzero is defined by a closed-source nature.

RAPID (Robust Aircraft Parametric Interactive Design) [55] is a geometry-centric conceptual design tool developed at Linköping University. It is built around CATIA, which serves as the core geometry engine, enabling detailed parametric modeling of major aircraft components including the fuselage, wings, nacelles, control surfaces, cabin, and windshield. The main goal of RAPID is to provide interactive and intuitive control over geometry during early-phase design. The tool was later integrated into the CADLab framework [56], which extends the environment with aerodynamic analysis and systems simulation modules. While this framework enables detailed geometry handling, it is not open source and depends heavily on commercial software. In contrast, RAPTOR offers an open environment.

RADE (Rapid Airframe Design Environment) [9] is a conceptual design framework developed at the Georgia Institute of Technology. It is intended to address the need for higher-fidelity structural modeling during the early phases of airframe design. RADE structure supports rapid iteration, which makes it particularly suitable for studies involving structural sizing or topology changes. However, the framework is not open source and appears to be tailored to in-house research use at Georgia Tech. In contrast, RAPTOR offers a publicly accessible and extensible environment, with an emphasis on openness.

DELWARX (Distributed Design Optimization of Large Aspect Ratio Wing Aircraft with Rapid Transonic Flutter Analysis in Linux) [36] is a multidisciplinary optimization framework developed at Virginia Polytechnic Institute and State University. Written in Python and targeting the Linux environment, DELWARX focuses specifically on the MDO of transonic transport aircraft with high aspect ratio wings. Its distinctive feature lies in its integration of rapid transonic flutter analysis into the optimization loo. This makes it especially suited for configurations where aeroelastic constraints significantly influence design feasibility. Compared to RAPTOR, which aims to deal with a large number of configurations, and therefore general modularity, DELWARX is tailored to a narrower but more structurally intensive domain and is not distributed as open source.

Beyond proprietary tools, several open-source frameworks have been developed in recent years, offering increased flexibility and transparency for academic and exploratory use.

Among them, SUAVE [84] stands out as a modular environment developed by Stanford University. It is designed to support modular, multi-fidelity analysis of both conventional and unconventional configurations. Like RAPTOR, it is structured around a set of independent modules covering key disciplines, and supports multidisciplinary optimization. However, to the best of our knowledge, SUAVE has not been actively maintained or updated in the past three years. Additionally, in the context of gradient-based optimization, RAPTOR distinguishes itself through its use of automatic differentiation for gradient computation. In contrast, SUAVE relies on finite-difference methods, which are simpler to implement but generally less accurate.

Another open-source software aligned with the conceptual design stage is FAST-OAD-GA [33], developed by ISAE-SUPAERO and ONERA, and built on top of the OpenMDAO platform. Like RAPTOR, it features a modular architecture and supports multidisciplinary design analysis and optimization. The similarities between the two frameworks lie in their open access philosophy and their use of OpenMDAO. However, some distinctions can be noted. FAST-OAD-GA relies on empirical models for the mass estimation, while in RAPTOR, the models developed include both an analytical and a CAD-based method, which are more accurate and offer multiple options. Another difference lies in the intended application domain: according to Lutz et al. [44], FAST-OAD-GA is primarily tailored for General Aviation aircraft, whereas, as previously discussed, RAPTOR is designed to handle both conventional and unconventional configurations.

A recent addition to the open-source ecosystem is AVIARY [60], the NASA's latest aircraft design and optimization tool, built on top of the OPENMDAO framework. Like RAPTOR, it supports multidisciplinary design optimization and is designed to be modular and extensible. AVIARY includes core subsystems for geometry, mass, aerodynamics, propulsion, and mission analysis, and allows users to integrate their own external models. One of the key distinctions lies in the optimization philosophy. AVIARY is structured to optimize the aircraft based on a complete mission profile, defined as a sequence of flight phases such as take-off, climb, cruise, and descent. The aircraft is sized and evaluated across this entire mission, with objectives and constraints distributed along the trajectory. In contrast, RAPTOR currently focuses on optimizing the design for one or several specific flight conditions. Another key difference lies in the way gradients are computed. In AVIARY, all model gradients are derived analytically, whereas RAPTOR relies on automatic differentiation. Although analytical gradients can offer high precision, they are significantly more complex and time-consuming to implement. In contrast, adding a new model to RAPTOR only requires ensuring that it is differentiable, allowing automatic differentiation to handle the rest. By comparison, extending AVIARY necessitates that the user not only implement the model but also manually derive and code its gradient expressions.

Focusing on detailed component level optimization, particularly for lifting surfaces, OPENAEROSTRUCT [52] is a lightweight open source tool designed for aerostructural optimization, built on top of NASA's OPENMDAO framework. It combines a Vortex Lattice Method for aerodynamic analysis with a three-dimensional spatial beam model to simulate structural behavior. These analyses are coupled within an optimization loop using OPENMDAO. While such tools are highly effective for optimizing individual components, they are not designed to address the aircraft as a complete system.

Following this literature review and the discussion of how the developed tool differs from existing solutions, the workflow used to build RAPTOR can now be presented.

1.4 Methodology

Following the presentation of the motivations behind the development of the RAPTOR framework and the introduction of its key design principles, this section outlines the methodology adopted to guide its construction and validation.

The process begins by establishing the architectural foundations of the framework. Since aircraft design inherently involves multiple interacting disciplines, such as aerodynamics, structures, weights, and stability, it is essential to define how these domains will be coupled and how data will flow between them. This leads to a global system architecture that ensures consistency and interoperability across all models.

Once this coupling structure is defined, attention turns to the formulation of an appropriate optimization strategy. This involves choosing the type of algorithm best suited to the framework, and ensuring that the architecture supports efficient solution of the optimization problem.

With the optimization approach in place, performance models must be selected and developed for each discipline involved. These models must not only provide physically meaningful outputs, but also be compatible with the requirements of the optimization process, in particular, the ability to compute derivatives. To this end, the majority of models in RAPTOR are designed to work with automatic differentiation tools, which require smooth, continuous formulations.

Once implemented, each model is systematically validated using reference data from the literature to ensure that it provides reliable results. Special attention is also given to verifying the accuracy of computed gradients by comparing them to finite-difference approximations.

After validation, the complete framework is used to solve aircraft optimization problems. This includes reproducing benchmark cases from the literature to confirm that the system behaves as expected, and exploring new, custom configurations to demonstrate its capabilities and flexibility.

This structured development process is summarized in Figure 1.3, which illustrates the logical sequence of tasks involved in building and applying the RAPTOR framework.

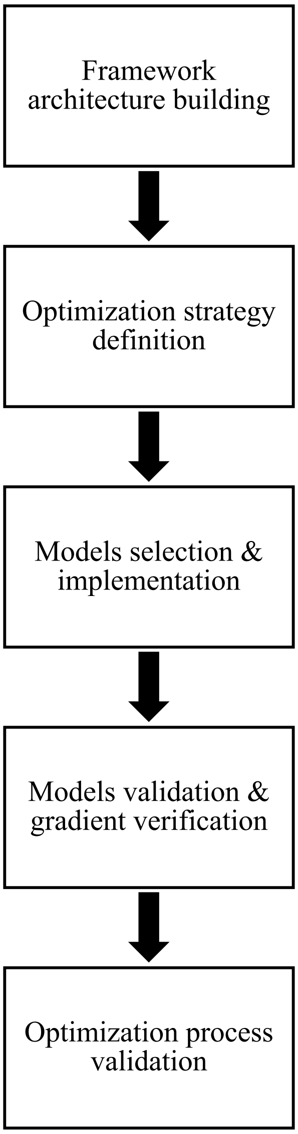

Framework architecture building
Optimization strategy definition
Models selection & implementation
Models validation & gradient verification
Optimization process validation

Figure 1.3: Workflow of the RAPTOR framework development.

Chapter 2

Framework architecture

Following the review of the context, the identification of the main limitations and advantages in existing tools and the positioning of RAPTOR within the current landscape of aircraft design frameworks, it is now possible to provide a detailed description of the developed tool.

This chapter outlines the architectural design strategy adopted for the RAPTOR framework. It begins by describing the key features of the multidisciplinary system under consideration, followed by the modular structure of the framework, including the organization of components, disciplines, and models, and the mechanisms that ensure their interoperability and scalability. The chapter then details the main decisions made regarding the optimization strategy. Finally, it explains the reasons behind building the framework on top of OpenMDAO [23], emphasizing its strengths in managing model coupling and derivative computation in a structured and efficient manner.

2.1 Multidisciplinary system

As discussed in previous sections, aircraft design is inherently a multidisciplinary process. An aircraft must not only be aerodynamically efficient, but also stable, structurally strong, and capable of meeting performance requirements, among other constraints. These disciplines are tightly interconnected and must be considered simultaneously.

Figure 2.1 illustrates a typical example of such coupling. In this case, the geometry module defines the wing's dimensions, which are then used by the aerodynamic model to calculate lift and drag forces. These aerodynamic loads are subsequently transferred to the structural model, which assesses whether the configuration can withstand the expected flight conditions without structural failure.

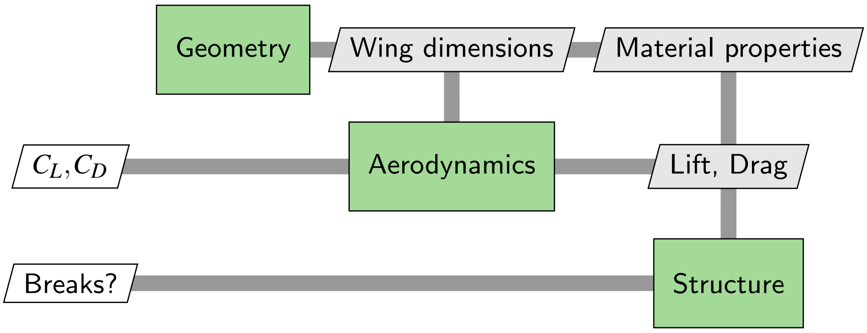


Figure 2.1: XDSM diagram illustration of internal discipline coupling in the RAPTOR framework.

Given the strong interdependence between disciplines, aircraft design must not aim at optimizing a single discipline , but identifying the best possible combination of all of them.

This principle has been fully integrated into the development of RAPTOR. Each model has been carefully developed to allow automatic and efficient coupling with others, ensuring robust interdisciplinary interactions.

In its current state, the framework supports a wide range of disciplinary models. It includes tools to define aircraft geometry, estimate component weights, and evaluate aerodynamic performance. Models are also provided to assess stability and equilibrium, while a structural module allows the computation of load cases, the definition of the flight envelope, and verification of the structure resistance to applied forces. Basic performance estimations are also incorporated.

A detailed description of each of these models is provided in the corresponding section of this report.

With the multidisciplinary system established, the following section focuses on how these disciplines are structured and integrated within the framework.

2.2 Disciplines organization

In this work, the aircraft system is decomposed into several disciplines (such as aerodynamics, weight estimation, and structural analysis). However, before defining the disciplines themselves, their organization within the framework must also be carefully defined.

The discipline organization developed in this section is illustrated in Figure 2.2.

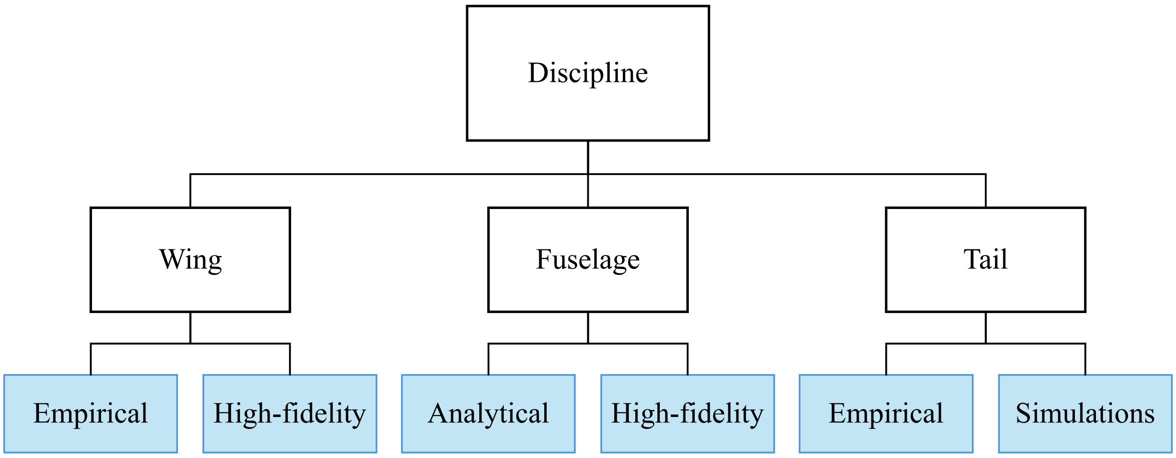


Figure 2.2: Example illustrating the organization of disciplines in RAPTOR, showing their decomposition into aircraft components and associated analysis methods.

As previously discussed, one of the principal objectives in developing RAPTOR is to ensure a high degree of modularity. This means that the framework should be flexible enough to optimize not only conventional aircraft configurations, but also unconventional ones, such as tailless designs or other innovative layouts.

To support this modularity, each aircraft component within a given discipline must be modeled independently. For example, in the aerodynamic discipline, the wing and fuselage can be evaluated separately, and if the aircraft configuration does not include a tail, the corresponding aerodynamic model is simply not included in the analysis. This component distinction approach is applied across all disciplines, allowing the framework to adapt flexibly to various aircraft architectures.

Beyond modularity, the framework is also organized to support multi-fidelity analyses and optimizations. Depending on the objectives, the user should be able to perform rapid, low-fidelity optimizations for early design exploration, as well as higher-fidelity optimizations when more accurate and detailed results are required.

To enable this flexibility, each discipline and aircraft component must offer multiple modeling approaches that can be selected independently. For instance, the aerodynamic model of the wing is not limited to a single method: it can rely on a low-fidelity, empirical formulation for quick estimates, or be replaced by a higher-fidelity approach such as a panel method when greater accuracy is required.

To summarize, for each discipline, the models are differentiated not only by the physical components they apply to, but also by the level of fidelity of the methods they use. This distinction ensures both the modularity of the framework and its ability to support multi-fidelity analysis and optimization.

This organization also promotes extensibility. A future developer can easily integrate a new model without modifying the existing ones. Moreover, since the framework is open-source, users are free to adapt or expand the current models to satisfy their specific needs.

With the multidisciplinary analysis system now clearly established and the internal organization of each discipline defined, the next step is to explore the optimization strategy implemented within RAPTOR. Beyond analyzing aircraft performance, the framework also aims to enhance it through automated optimization.

2.3 Optimization strategy

2.3.1 Theoretical overview

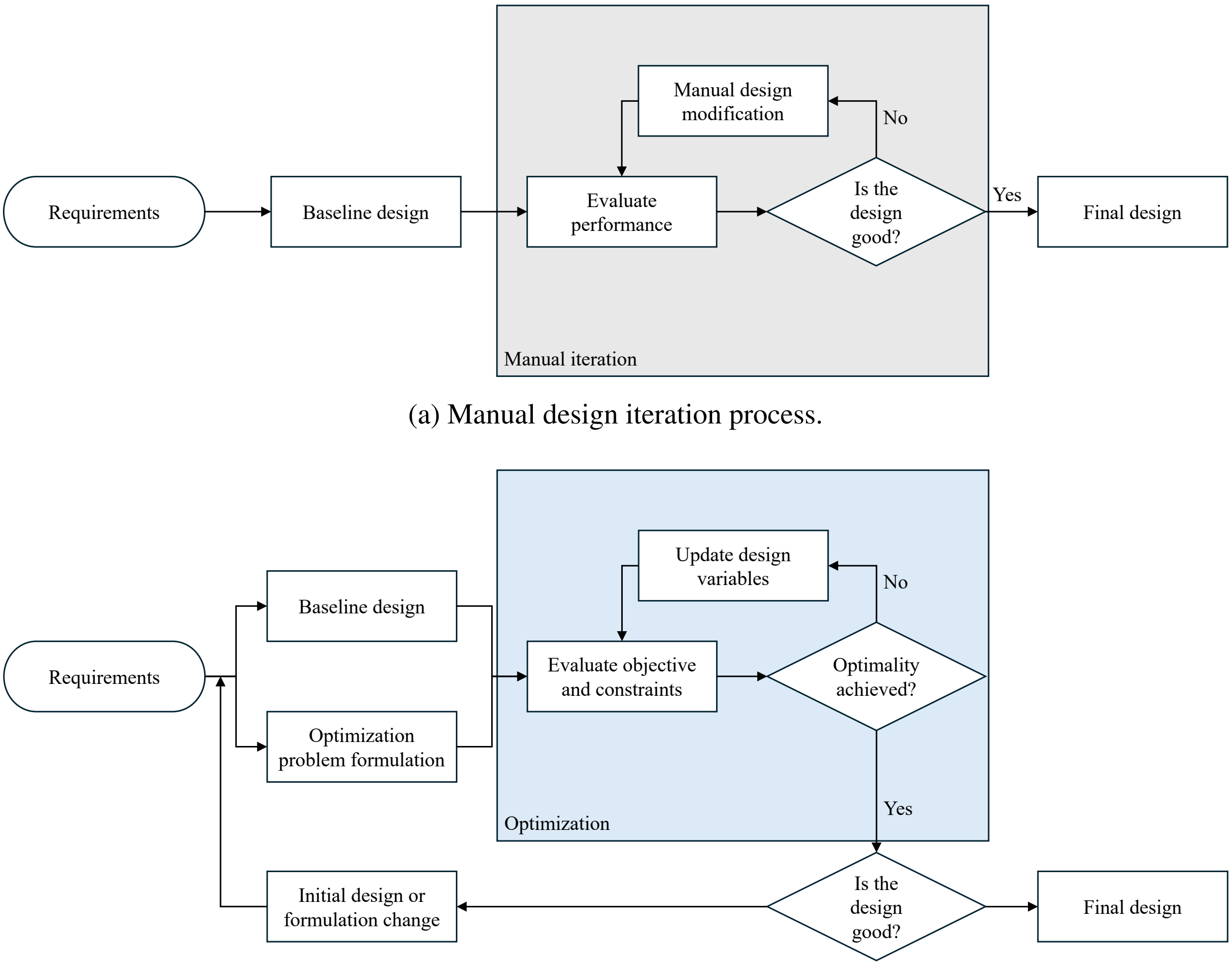
Transitioning from analysis to optimization requires the consideration of several additional aspects. The first step is to understand the logic behind the optimization process and the reasons for adopting this approach. Figure 2.3 compares the traditional aircraft design method with the optimization workflow.

In a traditional design process, the engineers evaluate the performances of a given configuration, adjust the design parameters manually, and repeat this cycle to try to improve the design. This trial-and-error approach is often time-consuming, requires a lot of effort, and does not guarantee that the best possible solution will be found, especially when dealing with complex systems where multiple disciplines interact. Since the coupling between disciplines is not managed automatically, this process can easily lead to solutions that are optimal for one specific discipline (like aerodynamics or weights), but suboptimal when considering the aircraft as a whole. In addition, as the number of variables increases, manually identifying a nearly optimal configuration becomes very difficult.

In contrast, a design optimization framework automates this iterative process by systematically exploring the design space based on predefined criteria. While human input is still required, mainly to define the initial setup, including design variables, constraints, and objectives, the optimization algorithm takes over to find the best feasible solution according to these specifications. In the context of multidisciplinary optimization, this approach has an additional benefit: it automates the coupling between disciplines, which increases the chances of finding a well-balanced solution across all domains, provided the problem is properly formulated.

Based on these considerations, it becomes clear that the optimization approach is particularly well suited to aircraft design. The following paragraphs introduce and explain the key elements of the formulation of an optimization problem, specifically objective function, the design variables and the constraints.

Firstly, the objective function defines the metric to be minimized or maximized. It is typically an output from one of the disciplinary models. In an aircraft aerodynamic optimization, the objective could be the minimization of the fuel burn, or the increase of the lift-to-drag ratio.



(a) Manual design iteration process.

(b) Optimization-based design process.

Figure 2.3: Comparison between manual design iteration and optimization-based approach. Adapted from Martins and Ning. [50].

Then, the design variables represent the degrees of freedom the optimizer can manipulate to improve the design. These variables define the geometry configuration, or the operating conditions of the system. In a practical example, it could be the wing geometry parameters, or the angle of attack of the aircraft.

Finally, the constraints some limits to ensure that the resulting optimal configuration remains feasible. In a practical optimization case, the constraints can be used to ensure the stability and the equilibrium of the aircraft.

Once the decision to perform an automatic optimization process has been made and the problem has been formally defined, the next step is to select an optimization algorithm that is well suited to the nature of the framework.

2.3.2 Optimization approach selection

To carry out an optimization effectively, selecting an appropriate algorithm is a key step. The performance of the process, both in terms of accuracy and computational efficiency, depends heavily on this choice. In general, optimization methods can be grouped into two main families: gradient-based and gradient-free approaches, each with their own advantages and limitations, depending on the nature of the problem.

As highlighted by Martins and Ning [50], gradient-based optimization methods are particularly powerful for solving engineering design problems efficiently. They rely on the availability of accurate derivatives of the objective and constraint functions with respect to the design variables, which enables fast and directed convergence toward a local optimum. However, to take full advantage of these methods, the underlying models must be implemented in a differentiable and numerically consistent way. In addition, the optimization problem must be well formulated, with proper bounds, constraint scaling, and careful choice of the objective function to ensure both convergence and stability.

In contrast, gradient-free methods are more appropriate in contexts where gradients are not accessible, such as when using black-box tools, legacy code, or non-differentiable models. These approaches are also well suited to problems involving discrete design variables, or when handling multiple conflicting objectives. Their ease of use makes them attractive in early design phases. That said, gradient-free methods have important limitations. They generally require a significantly higher number of function evaluations to converge, and they struggle to scale efficiently when the number of design variables grows. Additionally, they tend to provide less precise control over constraint satisfaction. For complex engineering problems involving disciplinary couplings and a high number of continuous design variables, such as aircraft design optimization, these drawbacks become critical.

In the context of RAPTOR, the goal is not to explore an entire design space, but to improve upon an existing baseline configuration. The optimization problems formulated in this framework are typically high-dimensional, multidisciplinary, and involve strict constraints. In such conditions, gradient-based methods are far more appropriate: they offer faster convergence, better scalability, and higher accuracy in satisfying constraints. In that sense, most models implemented in RAPTOR have been designed with gradient consistency in mind. Many use automatic differentiation, enabling straightforward and efficient derivative evaluation. For these reasons, a gradient-based optimization strategy has been adopted in RAPTOR.

The main comparisons between gradient-based and gradient-free optimization methods are summarized in Table 2.1.

Criterion	**Gradient-based methods**	**Gradient-free methods**
Derivatives	Needed	Not needed
Large number of variables	Efficient	Limited
Constraint handling	Robust	Difficult
Use in MDO	Highly suitable	Less appropriate

Table 2.1: Comparison between gradient-based and gradient-free optimization approaches.

2.3.3 Gradient-based algorithm

Once the optimization strategy has been defined as gradient-based, the next step is to select the most appropriate algorithm. Several well-established gradient-based methods exist in the literature and are implemented in standard scientific computing frameworks. However, not all are suited for problems involving constraints, as is almost always the case in aircraft design.

The BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm is a quasi-Newton method. While it is efficient for unconstrained problems and enjoys fast convergence, it does not support nonlinear constraints. As stated by Fletcher [19], BFGS is inherently limited to unconstrained or bound-constrained problems. This disqualifies it for the current framework, where complex constraints are omnipresent. An extension of BFGS, known as L-BFGS-B (Limited-memory BFGS with Bounds), was introduced by Liu et al.[42], and is able to handle large-scale problems with simple box constraints. While L-BFGS-B is better suited to high-dimensional optimization, it remains restricted to bound-constrained formulations. In RAPTOR, this is insufficient, as users may wish to use equality constraints, such as for example, forcing the lift force to be equal to the aircraft weight in cruise conditions.

The Conjugate Gradient (CG) method is tailored for large-scale unconstrained optimization. Although highly efficient in that context, it does not support the inclusion of constraints. The same applies to its second-order extension, Newton-CG, which remains fundamentally suited for unconstrained problems. As clarified in Nocedal et al. [65], these algorithms are not intended for constrained nonlinear processes and thus are not compatible with the requirements of RAPTOR.

Similarly, truncated Newton methods, developed for solving large-scale problems through iterative Hessian vector products, do not accommodate nonlinear constraints. According to Nocedal et al. [65], these methods fall under the category of unconstrained optimization techniques. Their potential to be adapted for constrained cases exists, but it requires additional mechanisms.

For the reasons outlined above, the algorithm adopted in RAPTOR is Sequential Least Squares Programming (SLSQP) , which belongs to the Sequential Quadratic Programming (SQP) family. SQP methods are explicitly designed to solve nonlinear optimization problems with equality and inequality constraints, as underlined by Martins and Ning [50]. They achieve this by solving, at each iteration, a quadratic programming subproblem in which the nonlinear objective function is locally approximated by a quadratic model and the constraints are linearized. The solver computes a descent direction that simultaneously reduces the objective and ensures constraint satisfaction. This makes SLSQP particularly attractive for the multidisciplinary and highly constrained optimization problems encountered in aircraft design. Its ability to handle general constraints, combined with the availability of efficient implementations, makes it an ideal choice for RAPTOR.

The comparison of these gradient-based algorithms is summarized in Table 2.2

Criterion	**BFGS**	**L-BFGS-B**	**CG**	**Newton-CG**	**Truncated Newton**	**SLSQP**
Constraint handling	None	Bounded	None	None	None	General
Scalability	Low	High	Medium	High	High	Medium

Table 2.2: Comparison of gradient-based optimization methods with respect to constraint handling and scalability.

Now that a gradient-based optimization approach has been selected, namely the SLSQP algorithm, the next step is to determine how the framework will compute the partial derivatives of each output functions of the models with respect to the input variables.

2.3.4 Gradients computation

As previously explained, the utilization of gradient-based method is an advantage, precisely if the computation of the gradients is efficient.

As a reminder, the Jacobian matrix for a vector-valued function

$$\boldsymbol{f}(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad \text{with} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

is defined as:

$$J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \tag{2.3.1}$$

In the gradient-based process used in the framework, for each model, the vector f and the vector x in Equation 2.3.1 respectively correspond to the output functions and the input variables To compute these gradients of the Jacobian matrix, several methods exist.

Among them, finite-difference methods are particularly popular due to their ease of implementation. They require no access to the internal structure of the model: as long as the inputs and outputs are known, the gradient can be approximated regardless of whether the model is built from simple, differentiable operations or from complex black-box components. This simplicity, however, comes at the cost of both computational efficiency and numerical accuracy. Finite-difference methods are relatively slow, especially when compared to more efficient methods like automatic differentiation. When dealing with a large number of design variables, the cost of computing the partial derivative of each output variable with respect to the inputs becomes significant, as it typically requires a full model evaluation per direction. For these reasons, finite-difference methods are not ideal for large-scale optimization. Whenever possible, they should be replaced by more accurate and efficient techniques such as analytic derivatives or automatic differentiation.

Another method for computing derivatives is the complex step differentiation. Unlike finite-difference methods. This method provides highly accurate derivatives while avoiding the numerical instabilities associated with finite differences. However, complex step differentiation also has important limitations. First, its computational cost scales linearly with the number of design variables, similar to finite differences, each variable requires a separate function evaluation. Moreover, because it requires full access to the models, it is not applicable in cases where the analysis is performed by external or black-box software components. For these reasons, complex step differentiation is not used in the optimization workflow of RAPTOR.

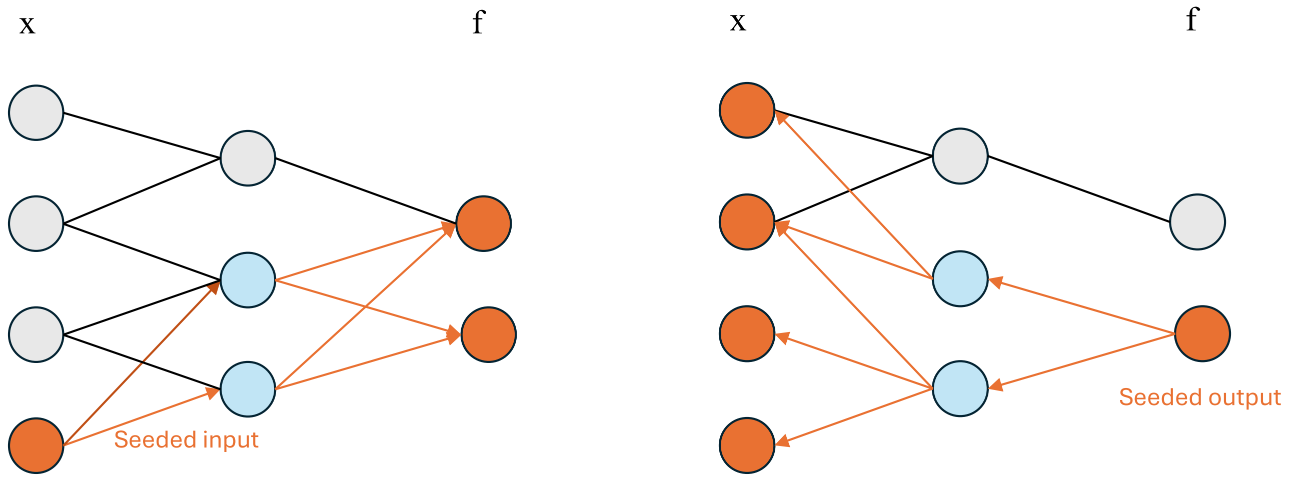
The most accurate approach to compute gradients is symbolic differentiation, also referred to as manual (or hand-computed) differentiation. This technique involves analytically deriving all partial derivatives from the explicit mathematical expressions of the model. As a result, the computed gradients are exact, and the method offers unmatched reliability and numerical precision. However, despite its accuracy, manually deriving and implementing the gradients becomes extremely complex and time-consuming, especially in large codes involving numerous variables and calculations. Additionally, this method is entirely inapplicable when dealing with black-box models, where the internal expressions are inaccessible. In the case of RAPTOR, even if most of the models are analytically derivable, the overall complexity of the framework, and the number of inputs make manual differentiation impractical. For these reasons, symbolic differentiation is not used in the present work.

Among the available methods to compute the Jacobian, automatic differentiation stands out as particularly accurate and efficient. Unlike finite differences, it can provide a high accuracy level. Automatic differentiation works by systematically applying the chain rule to all elementary operations in the computational graph of the function. It can be implemented in two distinct ways: forward mode and reverse mode, whose efficiency depends on the relative sizes of the input and output spaces.

In forward mode, derivative information is propagated from the inputs toward the outputs, by following the structure of the computational graph, as illustrated in Figure 2.4a. For a given input variable v_j, the total derivative of any intermediate or output variable v_i is obtained by recursively applying the chain rule along all computational paths from v_j to v_i.

This propagation involves summing the contributions from all direct predecessors of v_i, denoted by $\mathrm{pred}(i)$, according to:

$$\frac{\mathrm{d}v_i}{\mathrm{d}v_j} = \sum_{k \in \mathrm{pred}(i)} \frac{\partial v_i}{\partial v_k} \cdot \frac{\mathrm{d}v_k}{\mathrm{d}v_j} \tag{2.3.2}$$



(a) Forward mode.

(b) Reverse mode.

Figure 2.4: Illustration of the two main approaches of automatic differentiation: forward mode and reverse mode. Adapted from Martins and Ning [50].

From a matrix perspective, each forward mode pass computes one column of the Jacobian described in Equation 2.3.1. Therefore, the total computational cost is proportional to the number of input variables. Forward mode is thus efficient when the number of inputs is small compared to the number of outputs.

In contrast, reverse mode propagates sensitivities in the opposite direction: from the outputs back to the inputs, as shown in Figure 2.4b. This mode starts by evaluating the entire computational graph forward. Then, the gradients are propagated in reverse.

$$\frac{\mathrm{d}f}{\mathrm{d}v_j} = \sum_{i \in \mathrm{succ}(j)} \frac{\partial v_i}{\partial v_j} \cdot \frac{\mathrm{d}f}{\mathrm{d}v_i} \tag{2.3.3}$$

In this case, one reverse mode pass yields the gradient of one scalar output with respect to all inputs, so one row of the Jacobian. As a result, reverse mode is far more efficient when the number of outputs is small and the number of inputs is large.

In the context of RAPTOR, reverse mode automatic differentiation is the most appropriate strategy for computing gradients. Indeed, aircraft design optimization problems typically involve a high number of design variables compared to the number of objectives and constraints.

As the automatic differentiation is chosen as the mean to compute the gradients, all semi-empirical and analytical models developed in RAPTOR are developed by ensuring differentiability.

However, the situation is different when it comes to high-fidelity weight estimation models. Since these models often rely on external tools, such as CAD software, it becomes difficult to compute gradients automatically. For example, in the case of the weight model that couples with OpenVSP with RAPTOR, the analysis is performed internally by the software, and its source code would need to be modified to enable automatic differentiation. In such cases, gradient computation must rely on finite-difference methods.

In conclusion, automatic differentiation has been chosen as the primary method for gradient computation in the developed tool, due to its ease of implementation, computational efficiency, accuracy, and suitability for large scale problems. However, in cases where higher-fidelity models rely on external software and reverse-mode differentiation cannot be applied, finite-difference methods are used as an alternative.

The comparison of the gradient computation methods presented in this section is summarized in Table 2.3.

Criterion	**Finite difference**	**Automatic differentiation**	**Analytical**	**Complex step**
Accuracy	Low	Very high	Exact	Very high
Scalability	Poor	Excellent	Medium	Poor
Implementation	Very easy	Straightforward	Difficult	Straightforward

Table 2.3: Comparison of gradient computation methods in terms of accuracy, scalability, and implementation complexity.

With the overall architecture of the framework established and the optimization strategy defined, it is now possible to identify and discuss the main implementation challenges encountered during the development of RAPTOR.

2.4 Implementation challenges

2.4.1 Automatic differentiation constraints

A central feature of RAPTOR is the use of automatic differentiation to compute gradients of the models. This technique offers many advantages: it is easy to implement, highly accurate, and computationally efficient. However, it also introduces certain constraints. In particular, all models must be differentiable, meaning they must be built using only mathematical operations that are continuous and smooth.

One common challenge arises with discretization. For example, considering the structural modeling of a wing using the discretization of the section into multiple stringers, it may seem logical to vary the number of stringers to improve the design. However, because the number of booms is a discrete value, it cannot be treated as a continuous design variable. Automatic differentiation cannot propagate gradients through changes in array size. Therefore, in RAPTOR, discretization parameters, such as the number of stringers structural sections, must remain fixed during optimization. Only continuous parameters like the stringers cross-sectional area can be varied.

Another issue arises from non-smooth mathematical operations, such as taking a maximum or minimum value over a set of data. Taking again the structural modeling of a wing section as an example, the component skin is divided into panels to compute shear stresses in each panel. To assess the structural safety, one might look at the maximum stress observed across all panels. However, the `max()` operation is not continuous: a small change in input can cause a sudden change in which panel carries the maximum load, leading to a discontinuity in the model. This breaks the gradient computation. To solve this, in RAPTOR, functions are implemented to replace the maximum operations with smooth approximations. For example, instead of taking the maximum directly, a smooth maximum function can be used, one that closely approximates the true maximum but remains continuous and differentiable. This allows gradients to flow through the model without interruptions.

Discontinuities can also be introduced through conditional logic statements in mathematical models. Take, for example, a simplified aerodynamic model where the performance of a wing is evaluated differently depending on its taper ratio (the ratio between the tip and root chord lengths). Suppose the function f to be minimized is defined as:

$$f(\text{taper}) = \begin{cases} f_1(\text{taper}), & \text{if taper} < 0.4 \\ f_2(\text{taper}), & \text{if taper} \geq 0.4 \end{cases} \tag{2.4.1}$$

This formulation introduces a discontinuity at taper = 0.4, both in the value and in the derivative of f. As a result, the gradient is undefined at that point, which causes optimization algorithms to fail or behave unpredictably. To avoid this, such models must be reformulated using smooth transitions between the zones, for example, by using weighted interpolations, to ensure the entire model remains differentiable.

2.4.2 Solution robustness

A differentiable model must also be robust across the expected range of design variables. For instance, if an aerodynamic model is valid for wing aspect ratios between 5 and 25, what should happen if the user specifies a value of 29. Simply cutting off the function outside this range would again introduce discontinuities. Instead, in RAPTOR, such functions are smoothly extrapolated beyond their nominal validity range. This ensures that the optimizer can still produce a result, even if it explores values slightly beyond the expected limits. Of course, if the input is far outside of physical realism, the output becomes unreliable, and it is the responsibility of the user to define appropriate design bounds. But by handling these edge cases smoothly rather than abruptly, the framework avoids introducing artificial constraints and remains more stable during optimization.

In summary, adapting models for automatic differentiation requires careful attention to mathematical smoothness. While these constraints may initially seem restrictive, they lead to more robust, efficient, and scalable optimization workflows. All the adaptations cited in this section are further investigated in the models development part (Section 3), as practical examples are treated.

2.5 Integration of OpenMDAO in the framework

The decision to build RAPTOR on top of OpenMDAO was driven by the need to efficiently manage multidisciplinary interactions and perform derivative computations. OpenMDAO offers robust tools for model coupling, derivative propagation, and the integration of components within a multidisciplinary system. These capabilities make it ideal to build an aircraft design analysis and optimization environment.

This section highlights the key advantages of using OpenMDAO for the development of such a framework. In particular, it focuses on its ability to facilitate the coupling of the different disciplinary models, to compute the total derivative, and to setup and execute gradient-based optimization workflows.

2.5.1 Discipline coupling

As discussed earlier, aircraft design is a highly multidisciplinary process. A single evaluation of a configuration typically involves geometric computations, aerodynamic evaluations, weight estimation, stability analysis, and structural checks. To enable communication between these disciplines, OpenMDAO offers a flexible and consistent coupling mechanism. Through the use of this framework, the output of one component can be automatically linked as an input to another.

In RAPTOR, thanks to the use of OpenMDAO, this coupling logic is automated and generalized across components and disciplines. Each model has its own input and output variables, and the framework automatically identifies and connects matching variables across the computational graph. It minimizes the implementation efforts and the risk of connection errors.

Another important advantage of OpenMDAO is the modularity it enables. Since it simplifies the coupling between different disciplines, each one can be divided into multiple components and methods, as explained in Section 2.2.

Finally, once the different components are coupled into a multidisciplinary system, the framework is also capable of performing gradient-based optimization. Each model computes the partial derivatives of its outputs with respect to its inputs. However, for optimization, what is really needed are the total derivatives of the objective function and the constraints with respect to the design variables. The next section explains why OpenMDAO is particularly well suited for computing these total derivatives efficiently and accurately.

2.5.2 Total derivatives computation

Once the components are coupled through their input and output variables, the next challenge is to propagate the derivatives accurately across the multidisciplinary system. In OpenMDAO, each component is responsible for computing its own local partial derivatives. From there, OpenMDAO handles the assembly of the full system and the propagation of sensitivities automatically. This allows user to focus on model implementation without needing to manually derive or code derivative interactions between the disciplines. To systematically compute total derivatives in a coupled system, the MAUD (Modular Analysis and Unified Derivatives) architecture reformulates the entire model as a single system of residual equations. This feature allows OpenMDAO to treat derivative propagation in an automatic way.

From a theoretical point of view, the MAUD formalism concatenates all design variables, state (or coupled) variables, and output functions (such as objectives and constraints) into a single global variable vector:

$$u = \left[x_1, \dots, x_m, y_1, \dots, y_n, f_1, \dots, f_p\right]^T \tag{2.5.1}$$

To describe the relationships between these variables, a corresponding residual function is constructed:

$$R(u) = 0 \tag{2.5.2}$$

The global residual function $R(u)$ aggregates the behavior of all components in the model and is structured to include three distinct types of residuals:

1. **Design variable consistency**:

$$R_i(u) = x_i - x_i^*, \quad \text{for } i = 1, \dots, m \tag{2.5.3}$$

This residual ensures that the current value x_i of each design variable matches the input provided by the optimizer, denoted by x_i^*. Here, $i = 1, \dots, m$, where m is the total number of design variables.

2. **State or coupled variable consistency**:

$$R_{m+j}(u) = \begin{cases} y_j - \hat{y}_j(x, y), & \text{if } y_j \text{ is explicitly defined} \\ -G_j(x, y), & \text{if } y_j \text{ is implicitly defined} \end{cases} \tag{2.5.4}$$

This ensures that each state variable y_j matches the value calculated by its component. The hat notation $\hat{y}_j$ indicates an explicit output from a component's model, while $G_j(x, y) = 0$ defines an implicit relation solved internally by the component. These residuals are defined for $j = 1, \dots, n$, where n is the number of coupled or state variables.

3. **Output consistency**:

$$R_{m+n+k}(u) = f_k - \hat{f}_k(x, y), \quad \text{for } k = 1, \dots, p \tag{2.5.5}$$

This residual forces the output value f_k, which is used in the optimization process (as an objective or a constraint), to match the value computed from the design and state variables. Here, $k = 1, \ldots, p$, where p is the total number of output functions included in the optimization problem.

Taken together, these residuals define the full coupled system. The objective is to ensure that all residuals vanish:

$$R(u) = \begin{bmatrix} R_1(u) \\ \vdots \\ R_{m+n+p}(u) \end{bmatrix} = 0 \tag{2.5.6}$$

To compute the total derivative of an output function f with respect to the design variables x, one has to differentiate both the function and the residual system. Using the chain rule:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u}\frac{du}{dx} \tag{2.5.7}$$

However, since u and x are coupled via $R(u,x) = 0$, the differentiation of the residuals is performed:

$$\frac{dR}{dx} = \frac{\partial R}{\partial x} + \frac{\partial R}{\partial u}\frac{du}{dx} = 0 \Rightarrow \frac{du}{dx} = -\left(\frac{\partial R}{\partial u}\right)^{-1}\frac{\partial R}{\partial x} \tag{2.5.8}$$

Substituting into Equation (2.5.7) yields:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \frac{\partial f}{\partial u}\left(\frac{\partial R}{\partial u}\right)^{-1}\frac{\partial R}{\partial x} \tag{2.5.9}$$

To avoid computing the inverse of $\frac{\partial R}{\partial u}$, the adjoint method is applied.The adjoint vector $\psi \in \mathbb{R}^{m+n}$ is defined as the solution of the following linear system:

$$\left(\frac{\partial R}{\partial u}\right)^T \psi = \left(\frac{\partial f}{\partial u}\right)^T \tag{2.5.10}$$

Once ψ is known, the total derivative becomes:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \psi^T\frac{\partial R}{\partial x} \tag{2.5.11}$$

This expression forms the basis of the coupled adjoint method. It is especially advantageous when the number of design variables is large compared to the number of objectives and constraints. In practice, this method allows for the evaluation of all necessary sensitivities with a single backward solve, independent of the number of design variables.

This expression highlights the key strength of the adjoint method: the ability to compute total derivatives with a computational cost that is largely independent of the number of design variables. In large-scale design problems such as aircraft optimization, where hundreds of design variables may influence performance metrics, this method offers significant efficiency gains over forward or finite-difference approaches.

Moreover, one of the powerful features enabled by the MAUD architecture in OpenMDAO is the coupling of both explicit and implicit components. In many aerospace applications, certain physical relationships are not given in closed form but must be solved iteratively, such as aerodynamic trim, thermal equilibrium, or structural deformations under load. OpenMDAO allows such implicit models to be integrated directly into the process. These implicit systems can be coupled with explicit ones, and the derivative propagation remains consistent and efficient across the entire model. This flexibility is essential for building aircraft models that can include feedback loops.

In summary, the use of OpenMDAO and its MAUD derivative propagation allows RAPTOR to scale efficiently with problem size, maintain modularity across disciplines, and perform gradient-based optimization with minimal manual effort. It allows the developer to only develop the models and the derivative of the output functions of each model with respect to its input variables, and the assembly is then automatically performed, thanks to this OpenMDAO coupled-adjoint method.

With this architecture in place, built specifically to support efficient multidisciplinary design, the optimization strategy itself can now be investigated more in details.

2.6 RAPTOR utilization

As explained in the RAPTOR positioning section, the aim of this work is to develop a framework that is able to analyze the performance of aircraft, and perform multidisciplinary optimization to obtain a better configuration.

The process begins with the user defining an initial aircraft configuration. Based on statistical methods and available reference data, the geometry of the layout to be optimized is established. To complete this initial step, RAPTOR offers two input options. The first consists of creating the aircraft model in OpenVSP [69]. Alternatively, the user can manually provide the necessary geometric parameters.

The second step consists in formulating the optimization problem. This involves specifying the performance metric to be improved, defining the constraints that the optimized configuration must satisfy, and selecting the geometric parameters that will act as design variables.

Once the optimization problem is defined, the user selects which disciplines will be included in the analysis, such as aerodynamics, stability, or mission performance. For each selected discipline, an appropriate level of model fidelity can be specified, depending on the desired balance between accuracy and computational cost. These choices are made directly in the test configuration, and the framework automatically integrates the corresponding models into the optimization workflow.

Once the models have been selected, the optimization process is initiated to search for an improved configuration compared to the initial design.

The workflow followed by a RAPTOR user is illustrated in Figure 2.5.

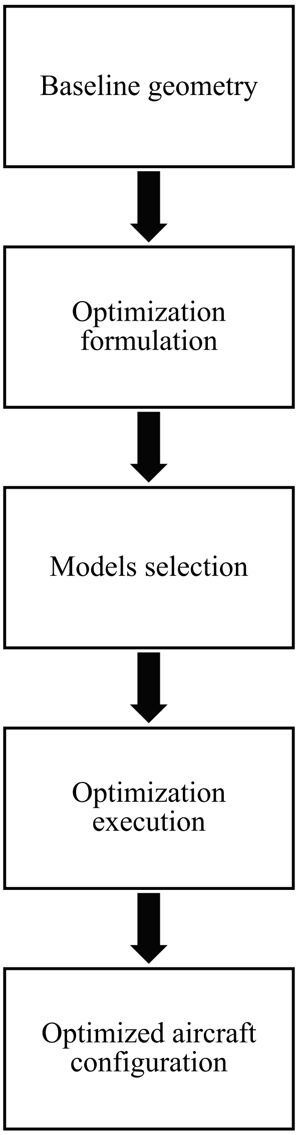


Figure 2.5: Overview of the workflow followed by a RAPTOR user.

Now that the main architectural choices have been made, the optimization strategy has been defined, the necessary model adaptations have been identified, and the typical user workflow has been established, the next step is to develop the performance analysis models, with particular attention to all critical aspects involved.

Chapter 3

Disciplinary modeling in RAPTOR

As discussed in Section 1.1, the aircraft design process involves strong interactions between multiple disciplines. To address this complexity, RAPTOR incorporates a range of analysis models covering the main aspects of aircraft design. For several disciplines, multiple modeling approaches have been implemented to support multi-fidelity optimization workflows. This flexibility enhances the adaptability of the framework to consider various configurations.

This section describes the analysis models integrated into the framework, as well as the modifications required to make them compatible with the optimization process.

3.1 Geometry

In the majority of aircraft design optimization processes, the geometry is the principal discipline that describes the design variables. As such, it is important to accurately model this part of the system.

Within the developed framework, users can define the aircraft geometry using one of two approaches: a CAD-based method or manual parameter specification. The integration of RAPTOR with the CAD software OpenVSP represents a key feature, enabling highly accurate evaluation of geometric quantities. In addition to its precision, OpenVSP offers a user-friendly interface that significantly simplifies the setup of aircraft configurations for analysis and optimization. Compared to manual definition, which can require specifying a large number of geometric parameters, the CAD-based approach notably reduces the time and effort needed to initialize a design.

3.1.1 CAD-based geometry definition

In the CAD-based geometry model, the aircraft configuration is initially defined within OpenVSP by the user. Based on this definition, key geometric parameters, such as surface areas, lengths and volumes are computed, and serve as inputs for the other disciplinary models within the framework.

The overall workflow of the geometry module is illustrated in Figure 3.1. It outlines the sequence of operations starting from the definition of the geometry input variables. These variables are first applied to the CAD model, and then, geometrical analyses are performed through OpenVSP to evaluate specific quantities when needed. Finally, the relevant variables are extracted, to serve as inputs of the other disciplines.

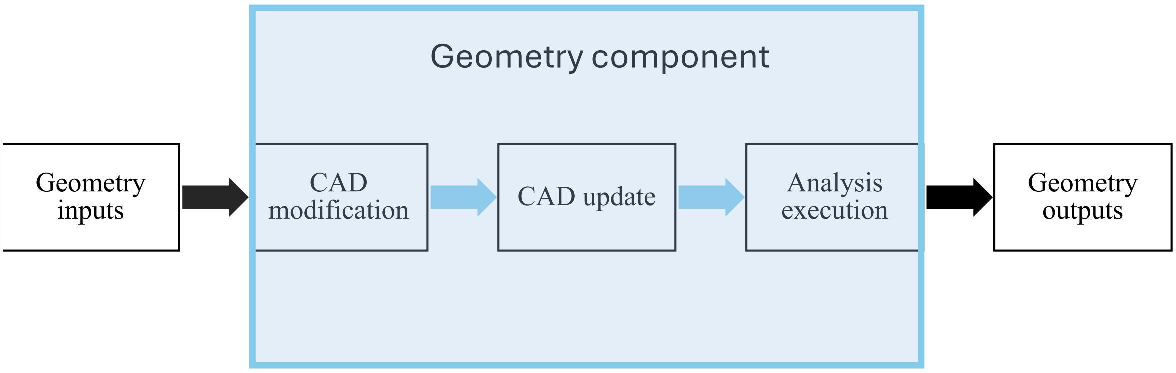


Figure 3.1: Illustration of the CAD-based geometry module workflow.

3.1.2 Manual geometry parameterization

An alternative option for defining the aircraft geometry in RAPTOR is to use a manual parameterization approach. In this configuration, the user directly inputs the key geometric characteristics of the aircraft, such as component locations, chord lengths, spans, and other relevant dimensions. From these inputs, the framework applies analytical geometric relationships to compute the surface areas, volumes, and other quantities needed by the various disciplinary models.

3.2 Weight estimation

In the aircraft design process, weight estimation is a key factor, as it directly influences performance, stability, structural integrity, and other critical aspects. Therefore, accurately computing the weight of each component and determining the aircraft's center of gravity is essential.

As previously mentioned, one of the key strengths of RAPTOR lies in its multi-fidelity and modular architecture. For the estimation of component weights, the user has two main options: a simplified analytical method, and a more advanced CAD-based approach.

3.2.1 Analytical weight estimation

This section presents the analytical weight estimation method developed in the present work. Based on geometrical relations, it approximates the volume of the internal structure of the components. One of the key advantages of this model lies in its differentiability, unlike CAD-based methods. This property enables the use of automatic differentiation to compute the corresponding Jacobian, which is particularly valuable in optimization contexts.

In RAPTOR, methods are implemented to calculate the weights of the main structural elements: the lifting surfaces (main wing, horizontal tail, canard), which share similar internal structural configurations, and the fuselage.

For the lifting surfaces, the internal structure is typically composed of several key elements: stringers, ribs, spars, and skin. An example of such a wing structure, constructed in the NX-Siemens software [81] is illustrated in Figure 3.2.

To evaluate the weight of a lifting surface component, one has therefore to sum the weight of each internal structure feature, which leads to the following expression:

$$m_{\mathrm{st,w}} = \rho \left(\mathrm{Vol}_{\mathrm{stringers,w}} + \mathrm{Vol}_{\mathrm{ribs,w}} + \mathrm{Vol}_{\mathrm{spars,w}} + \mathrm{Vol}_{\mathrm{skin,w}}\right) \tag{3.2.1}$$

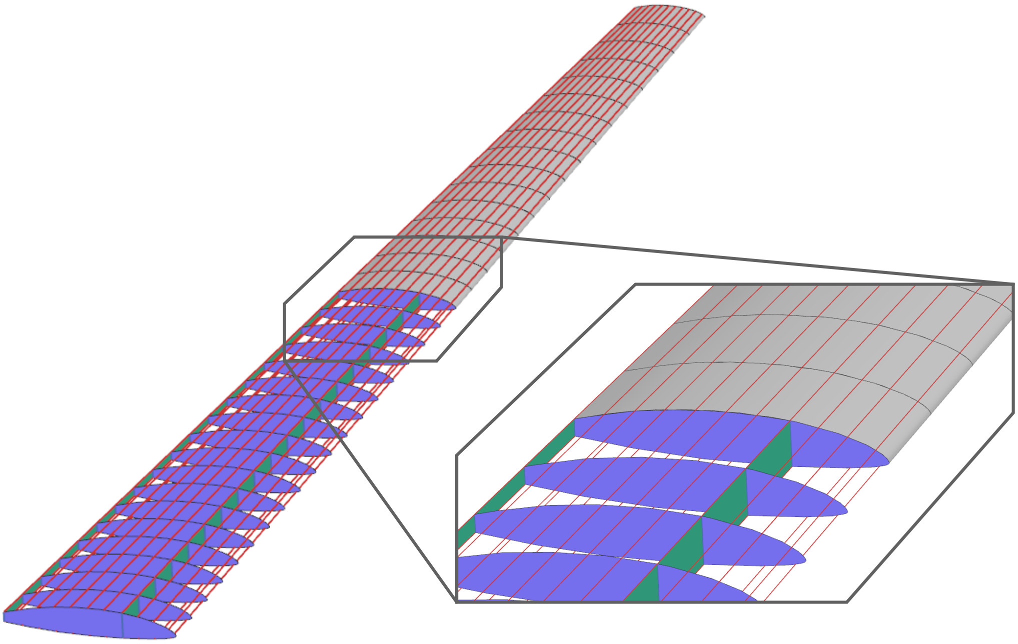


Figure 3.2: Internal structure of a typical half-wing, showing stringers in red, ribs in blue, spars in green, and skin in grey.

Where ρ is the material density, and Vol_i is the volume of the feature i. It has to be noted that Equation 3.2.1 assumes that all the internal structure is composed of the same material.

To compute the volume of material contained in the stringers, the cross-sectional area of each stringer is provided as an input, under the assumption that it remains constant along its entire length. The length of each stringer is then determined by calculating the distance between its root and tip positions. The tip coordinates are derived using key wing parameters such as the dihedral angle, sweep angle, and taper ratio. Once the lengths are obtained, the total volume is computed by summing the individual volumes of all stringers, as expressed in the following equation:

$$\mathrm{Vol}_{\mathrm{stringers,w}} = \sum_{i=1}^{n_{\mathrm{stringers}}} A_{\mathrm{stringer},i} \sqrt{\left(x_i^{\mathrm{tip}} - x_i^{\mathrm{root}}\right)^2 + \left(y_i^{\mathrm{tip}} - y_i^{\mathrm{root}}\right)^2 + \left(z_i^{\mathrm{tip}} - z_i^{\mathrm{root}}\right)^2} \tag{3.2.2}$$

where $A_{\mathrm{stringer},i}$ denotes the cross-sectional area of stringer i, and x, y, and z represent its spatial coordinates, taken respectively at the root and tip of the component. The total number of stringers is $n_{\mathrm{stringers}}$.

To estimate the total volume of the ribs, a uniform thickness is assumed and provided as an input. The total number of ribs is also specified by the user, under the assumption that they are evenly distributed along the span. The area of each rib is calculated using the swept area method: a reference point within the airfoil is chosen, the airfoil contour is interpolated, and the area is approximated by summing triangular surface elements. Based on this, the total rib volume is computed as follows:

$$\mathrm{Vol}_{\mathrm{ribs,w}} = \sum_{i=1}^{n_{\mathrm{ribs}}} A_{\mathrm{rib},i} t_{\mathrm{rib},i} \tag{3.2.3}$$

where $A_{\mathrm{rib},i}$ is the area of rib i, $t_{\mathrm{rib},i}$ its thickness, and n_{ribs} the total number of ribs.

The next step is to compute the volume of the spars. The model takes as inputs the thickness of each spar as well as its chordwise position. Based on this position, the spar height at both the wing root and tip is evaluated by locating the corresponding points on the interpolated airfoil contours. The spar length is then determined using the component span, the local sweep angle at the spar location, and the wing dihedral angle. The total volume of the spars is computed using the following expression:

$$\mathrm{Vol}_{\mathrm{spars,w}} = b_{\mathrm{w}} \sum_{i=1}^{n_{\mathrm{spars}}} \frac{t_{\mathrm{spar},i}\left(h_{\mathrm{spar},i}^{\mathrm{root}} + h_{\mathrm{spar},i}^{\mathrm{tip}}\right)}{2\cos\Lambda_{x_i,\mathrm{w}}} \cos\Gamma_{\mathrm{w}} \tag{3.2.4}$$

where b_{w} is the span of the component, n_{spars} is the number of spars, $t_{\mathrm{spar},i}$ is the thickness of spar i, $h_{\mathrm{spar},i}^{\mathrm{root}}$ and $h_{\mathrm{spar},i}^{\mathrm{tip}}$ are its heights at the root and tip respectively, $\Lambda_{x_i,\mathrm{w}}$ is the local sweep angle at the position of spar i, and Γ_{w} is the wing dihedral angle.

The final volume to be estimated is that of the wing skin. The skin thickness is provided as an input and is assumed to remain constant along the entire span. To compute the skin surface area, the perimeter of the airfoil is calculated using interpolation and numerical integration along the contour (curvilinear length). The total skin volume is then given by:

$$\mathrm{Vol}_{\mathrm{skin,w}} = t_{\mathrm{skin,w}} \frac{\left(P_{\mathrm{root,w}} + P_{\mathrm{tip,w}}\right)}{2} \frac{b_{\mathrm{w}}}{2\cos\Lambda_{c/4,\mathrm{w}}} \cos\Gamma_{\mathrm{w}} \tag{3.2.5}$$

where $t_{\mathrm{skin,w}}$ is the skin thickness, $P_{\mathrm{root,w}}$ and $P_{\mathrm{tip,w}}$ are the airfoil perimeters at the root and tip respectively, b_{w} is the span of the component, $\Lambda_{c/4,\mathrm{w}}$ is the sweep angle at the quarter-chord line, and Γ_{w} is the dihedral angle.

In the current version of the model, the wing is assumed to be trapezoidal and uniformly tapered. However, the method can easily be extended to multi-section wings by discretizing the span into additional airfoil sections.

From all these quantities, the mass of the wing, tails and canard can be evaluated, by injecting them in Equation 3.2.1.

As previously explained, the internal structure of a fuselage is different than the one of a lifting surface. Indeed, it is composed of stringers, frames and a skin. An example of the rear part of the internal structure of a fuselage constructed in NX-Siemens is represented in Figure 3.3.

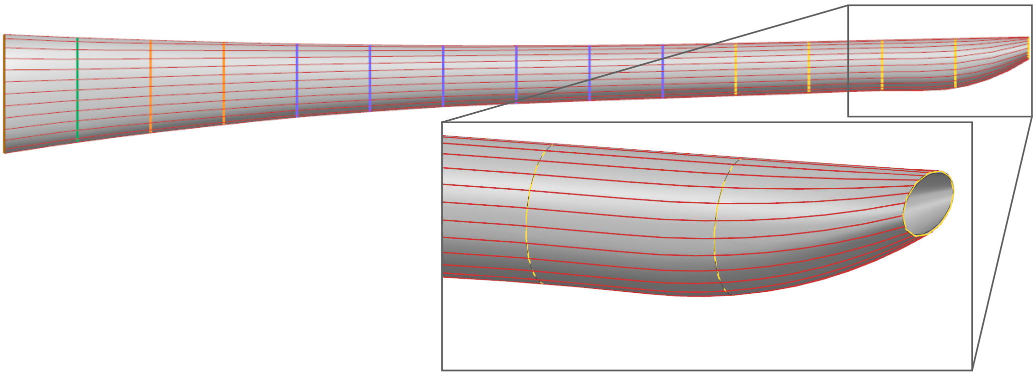


Figure 3.3: Internal structure of a typical rear fuselage, showing skin in grey, stringers in red and frames in other colors.

The total structural mass of the rear fuselage is computed, similarly to the case of the lifting surfaces, as the sum of the masses of its internal structural components:

$$m_{\mathrm{st,fus}} = \rho\left(\mathrm{Vol}_{\mathrm{stringers,fus}} + \mathrm{Vol}_{\mathrm{skin,fus}} + \mathrm{Vol}_{\mathrm{frames,fus}}\right) \tag{3.2.6}$$

where ρ is the material density, and $\text{Vol}_{\text{stringers,fus}}$, $\text{Vol}_{\text{skin,fus}}$, and $\text{Vol}_{\text{frames,fus}}$ are respectively the volumes of the stringers, the skin, and the frames.

Unlike lifting surfaces, assuming a uniformly tapered fuselage would introduce significant inaccuracies, particularly considering that the framework is intended to handle a wide range of configurations. To improve geometric flexibility and accuracy, the fuselage is instead defined by a set of cross-sectional profiles, spaced along its length. Each section is assumed to have an elliptical shape, which approximates the geometry of most aircraft fuselages. In the intervals that link two sections together, the shape is assumed to be uniformly tapered. The corresponding inputs include the vertical and horizontal semi-axes of each section, as well as the distance between consecutive sections. An illustration of the subdivision of the fuselage into sections and intervals is shown in Figure 3.4. This configuration is used as a reference case for validating the weight estimation model.

To compute the total volume of the stringers, the number of stringers and their cross-sectional area are provided as inputs. The fuselage is discretized based on the defined sections, and the coordinates of the stringers at each section are stored in vectors. As with the lifting surfaces, the lengths of the stringers are computed from this discretized geometry, and the total volume is then obtained using Equation 3.2.2.

For the fuselage skin, the thickness $t_{\text{skin,fus}}$ is provided as an input and assumed constant along the entire length. Since the fuselage geometry is defined by a series of cross-sectional profiles, it is discretized into intervals between consecutive sections. The total volume of the skin is then computed by summing, over all intervals, the product of the skin thickness, each interval length and the average perimeter of its two bounding sections. This leads to the following expression:

$$\text{Vol}_{\text{skin,fus}} = t_{\text{skin,fus}} \sum_{i=1}^{n} l_i \frac{P_{\text{sec}_{i-1}} + P_{\text{sec}_i}}{2} \tag{3.2.7}$$

where $t_{\text{skin,fus}}$ is the skin thickness, n is the number of intervals (equal to the number of sections minus one), l_i is the length of interval i, and $P_{\text{sec}_{i-1}}$, P_{sec_i} are the perimeters of the sections at the beginning and end of the interval, respectively.

To compute the total volume of the frames, the model assumes a uniform cross-sectional area $A_{\text{frames,fus}}$, provided as an input. For each interval between two fuselage sections, the user also specifies the number of frames, which are assumed to be evenly distributed along the interval. Given the length of each interval and the number of frames it contains, the model estimates the perimeter at each section and uses it to approximate the local frame perimeter. The total frame volume is then computed as the sum, over all intervals, of the product of the number of frames, the cross-sectional area, and the average perimeter within the interval:

$$\text{Vol}_{\text{frames,fus}} = A_{\text{frames,fus}} \sum_{i=1}^{n} P_i \tag{3.2.8}$$

with P_i the perimeter of frame i.

By substituting the computed volumes of the stringers, skin, and frames into Equation 3.2.6, the structural mass of the fuselage can be evaluated.

With the implementations described above, RAPTOR is capable of computing the structural weights of the wing, tail surfaces, optional canard, and fuselage.

The estimation of each component's center of gravity depends on the geometry option selected by the user. If the CAD-based model is used, the CG positions are directly evaluated using OpenVSP within the geometry module. In contrast, if the manual geometry mode is selected, empirical approximations are employed to estimate the CG locations. For other aircraft components such as the engine or subsystems, both their masses and CG positions must be provided explicitly by the user.

Once the CG and mass of each component are known, either computed or given as input, the overall aircraft center of gravity is calculated by the weight class using the following expression:

$$\mathrm{CG} = \frac{\sum \mathrm{CG}_i \, m_i}{\sum m_i} \tag{3.2.9}$$

where CG is the vector representing the overall center of gravity, CG_i is the center of gravity of component i, and m_i its mass.

With the analytical weight model now implemented, it must be validated against reference data. For this purpose, a CAD model of a wing and a fuselage, including their internal structures, is built using the OpenVSP software. The weights of the internal components are then computed by performing the appropriate structural analyses within the tool.

For the wing, a typical glider configuration is modeled. It corresponds to a trapezoidal, uniformly tapered lifting surface, which aligns with the assumptions made in the analytical model. The main geometric parameters of the wing used for the validation are listed in Table 3.1. Two spars are included in the model, positioned respectively at the quarter-chord and at 80% of the chord length.

Wing parameter	**Symbol**	**Value**
Airfoil type	–	Wortmann FX 62-K-131
Wing span	b_{w}	20.0 m
Wing root chord	$c_{\mathrm{r,w}}$	1.286 m
Wing taper ratio	λ_{w}	0.4
Wing quarter-chord sweep	$\Lambda_{c/4,\mathrm{w}}$	0.0°
Wing dihedral angle	Γ_{w}	0.0°
Material density	ρ_{mat}	1800 kg/m^3
Wing stringer area	$A_{\mathrm{stringers,w}}$	90 mm^2
Wing skin thickness	$t_{\mathrm{skin,w}}$	1.5 mm
Wing rib thickness	$t_{\mathrm{rib,w}}$	1.5 mm
Wing spar thickness	$t_{\mathrm{spar,w}}$	1.5 mm
Number of wing ribs	$n_{\mathrm{ribs,w}}$	70
Number of wing spars	$n_{\mathrm{spars,w}}$	2

Table 3.1: Main geometrical and structural parameters of the wing used for validation.

Using the geometry defined in Table 3.1, the structural weight of the wing is computed with OpenVSP and the analytical model developed in RAPTOR. The results from both methods are compared in Table 3.2. The very small relative difference confirms the accuracy and reliability of the implementation in RAPTOR, despite the simplifying assumptions on which it is based.

The most significant limitation lies in the assumption that the wing is trapezoidal and uniformly tapered, which excludes the possibility of modeling multi-section wings. However, extending the model to support more complex geometries appears to be straightforward and is listed among the future development possibilities discussed later in this report.

To validate the fuselage weight estimation model, a fuselage including its internal structure is created in OpenVSP. The overall shape is defined by four cross-sectional profiles, resulting in three

	Wing structural mass
RAPTOR	172.66 kg
OpenVSP	169.99 kg

Table 3.2: Comparison between the wing structural mass predicted by RAPTOR and OpenVSP, used to validate the structural weight model.

intervals along the fuselage length. This configuration is illustrated in Figure 3.4, where l_1, l_2, and l_3 denote the lengths of the intervals, and sec_1, sec_2, and sec_3 represent the corresponding intermediate cross-sections.

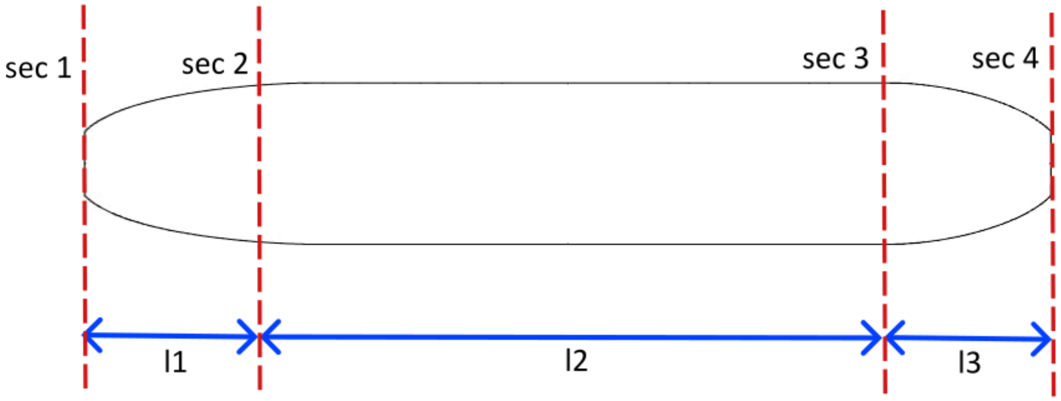


Figure 3.4: Subdivision of the fuselage into sections and intervals.

The key input parameters used to compute the mass of the fuselage illustrated in Figure 3.4 are summarized in Table 3.3.

From there, the structural mass of this fuselage is calculated, and the comparison of the results obtained using RAPTOR and OpenVSP are compared in Table 3.4.

As for the wing component, the fuselage weight estimation model developed in RAPTOR has been validated, and the results confirm that it provides accurate predictions. In the case of the fuselage, the model also offers increased flexibility, as the geometry is defined using multiple sections. This allows for more complex shapes to be represented.

The main limitation of the current implementation lies in the assumption that all fuselage cross-sections are elliptical. While this approximation is realistic for most conventional aircraft, extending the model to support more general shapes could be considered in future developments.

With both analytical models now validated, it can be concluded that they achieve a good level of accuracy, despite the simplifications made, particularly in the geometric assumptions. A key advantage of these models over CAD-based methods is that they are implemented in a way that supports automatic differentiation. This enables the computation of accurate gradients for optimization purposes.

3.2.2 CAD-based weight estimation

The second approach implemented in RAPTOR for computing component weights relies on the coupling with OpenVSP.

Parameter	Symbol	Value
Fuselage material density	ρ_{fus}	2800 kg/m^3
Fuselage stringer area	$A_{\text{stringers,fus}}$	20 mm^2
Fuselage skin thickness	$t_{\text{skin,fus}}$	1.0 mm
Frame cross-sectional area	$A_{\text{frames,fus}}$	20 mm^2
Horizontal semi-axes (per section)	a_i	[0.10, 0.25, 0.25, 0.10] m
Vertical semi-axes (per section)	b_i	[0.10, 0.30, 0.30, 0.10] m
Distances between fuselage sections	l_i	[0.7, 1.8, 0.5] m
Number of fuselage stringers	$n_{\text{stringers,fus}}$	14
Number of frames per segment	$n_{\text{frames},i}$	[2, 6, 2]

Table 3.3: Main geometrical and structural parameters of the fuselage used for validation.

	Fuselage structural mass
RAPTOR	15.70 kg
OpenVSP	15.53 kg

Table 3.4: Comparison between the fuselage structural mass predicted by RAPTOR and OpenVSP, used to validate the structural weight model.

In its first version, the model uses the component volume as input, calculated within the geometry class based on the CAD. The user must also define a lightening ratio, which represents the proportion of internal empty space within the component. The structural mass is then estimated using the following relation:

$$m_{\text{st,i}} = \rho \text{Vol}_i \,(1 - \text{ratio}_i) \tag{3.2.10}$$

Where $m_{\text{st,i}}$ is the structural mass of the component i, Vol_i its volume and ratio_i its lightening ratio, reflecting the ratio of the volume that is empty.

A second, more detailed version directly computes the weight of the internal structures of the aircraft components. To use this method, the user must not only model the outer geometry of each component in the CAD file, but also explicitly define their internal structural elements using the appropriate OpenVSP tools. RAPTOR then executes the structural weight analysis.

While these CAD-based methods offer higher accuracy, they currently rely on finite-difference techniques to compute gradients, as they do not support automatic differentiation in the current version of the framework.

To conclude this section, the user has access to three different weight estimation strategies within RAPTOR, reflecting the framework's multi-fidelity capabilities. The analytical model provides reasonable accuracy while allowing gradient evaluation through automatic differentiation. Alternatively, the user can opt for one of the two CAD-based models, which offer higher fidelity but rely on finite-difference methods for gradient computation.

3.3 Aerodynamics

Once the weight of the aircraft has been estimated, it becomes essential to design an aerodynamic configuration capable of generating sufficient lift to support this weight, while minimizing drag to ensure efficient performance. In addition to lift and drag, the calculation of aerodynamic moments is also critical, as it directly influences the aircraft stability characteristics.

This section presents two different aerodynamic models developed within RAPTOR to compute these key quantities. The first model relies primarily on semi-empirical formulations, while the second uses a panel method. The availability of both models illustrates the multi-fidelity architecture of RAPTOR, allowing users to balance computational cost and accuracy depending on the design stage or application.

Moreover, similarly as for the weight calculations, for a given method, the classes are separated in components, which ensures modularity.

3.3.1 Semi-empirical model

The first aerodynamic model implemented in RAPTOR is semi-empirical, and is primarily based on methods and equations from Raymer [72], DATCOM [18], and Roskam [75]. As will be explained further, to ensure compatibility with automatic differentiation, some equations have been slightly reformulated. The choice of these specific methods is motivated by the fact that only minimal adjustments were necessary to make the model differentiable, thus preserving both accuracy and implementation efficiency.

Starting with the lift model for the wing, the lift coefficient is computed using the following relation:

$$C_{L,\mathrm{w}} = C_{L_{\alpha,\mathrm{w}}} \left(\alpha + i_{\mathrm{w}} - \alpha_{L=0,\mathrm{w}}\right) \tag{3.3.1}$$

where $C_{L_{\alpha,\mathrm{w}}}$ is the lift curve slope of the wing, α is the angle of attack of the fuselage, i_{w} is the wing incidence relative to the fuselage centerline, and $\alpha_{L=0,\mathrm{w}}$ is the wing zero-lift angle of attack.

In this model, $\alpha_{L=0,\mathrm{w}}$ is assumed to be equal to that of the airfoil used, and both i_{w} and α are user-defined inputs.

The lift curve slope of the wing, $C_{L_{\alpha,\mathrm{w}}}$, is computed using the following expression from DATCOM [18]:

$$C_{L_{\alpha,\mathrm{w}}} = \frac{2\pi AR_{\mathrm{w}}}{2 + \sqrt{\frac{AR_{\mathrm{w}}^2\beta^2}{\kappa^2}\left(1 + \frac{\tan^2\Lambda_{c/2,\mathrm{w}}}{\beta^2}\right) + 4}} \tag{3.3.2}$$

where AR_{w} is the wing aspect ratio (provided by the geometry discipline), $\beta = \sqrt{1 - M^2}$ is the compressibility factor based on the Mach number M, $\Lambda_{c/2,\mathrm{w}}$ is the mid-chord sweep angle, and $\kappa = \frac{a_0}{2\pi}$ is the ratio between the airfoil lift curve slope a_0 and 2π.

By injecting these terms into Equation 3.3.1, the wing lift coefficient $C_{L,\mathrm{w}}$ can be calculated.

The model for the horizontal tail lift coefficient follows a similar formulation, with the additional inclusion of the downwash effect:

$$C_{L,\mathrm{ht}} = C_{L_{\alpha,\mathrm{ht}}} \left(\alpha + i_{\mathrm{ht}} - \epsilon - \alpha_{L=0,\mathrm{ht}}\right) \tag{3.3.3}$$

where $C_{L_{\alpha,\mathrm{ht}}}$ is the lift curve slope of the horizontal tail, i_{ht} is the tail incidence, $\alpha_{L=0,\mathrm{ht}}$ its zero-lift angle of attack, and ϵ is the downwash angle at the tail location.

The value of $C_{L_{\alpha,\mathrm{ht}}}$ is computed using the same formula as for the wing (Equation 3.3.2), but using the geometrical and airfoil parameters of the horizontal tail. The incidence i_{ht} and the zero-lift angle of attack are considered user inputs. The downwash angle ϵ is calculated using expressions based on DATCOM [18], which are not detailed here.

Once the lift contributions from the lifting surfaces are calculated, the total lift of the aircraft can be evaluated. To preserve modularity, the model is designed so that if a component is not included by the user, its contribution to the lift is automatically set to zero.

The overall lift coefficient of the aircraft is computed using the following relation:

$$C_L = C_{L_\alpha} (\alpha - \alpha_{L=0}) \tag{3.3.4}$$

where $\alpha_{L=0}$ represents the zero-lift angle of attack of the full aircraft. This angle is assumed to be the one for which the sum of the lift generated by all lifting surfaces is zero.

The lift curve slope of the aircraft, C_{L_α}, is calculated using the following expression:

$$C_{L_\alpha} = K_{\mathrm{WB}} C_{L_{\alpha,\mathrm{w}}} + C_{L_{\alpha,\mathrm{ht}}} \eta_{\mathrm{ht}} \frac{S_{\mathrm{ht}}}{S_{\mathrm{w}}} \left(1 - \frac{d\epsilon}{d\alpha}\right) \tag{3.3.5}$$

In this equation, K_{WB} is a wing–body interference factor, computed using the methods described in DATCOM [18]. The term $\frac{d\epsilon}{d\alpha}$ represents the rate of change of the downwash angle with respect to the angle of attack. The coefficient η_{ht}, corresponding to the aerodynamic efficiency of the horizontal tail, is set to 0.9 as a conservative value. Finally, S_{ht} and S_{w} denote the planform areas of the horizontal tail and the wing, respectively.

The next feature computed inside the aerodynamics models is the wing pitching moment coefficient about its aerodynamic center, $C_{\mathrm{m,w}}$, evaluated using the following DATCOM [18] formula:

$$C_{\mathrm{m,w}} = \left[c_{m,0} \frac{\mathrm{AR_w} \cos^2 \Lambda_{c/4,\mathrm{w}}}{\mathrm{AR_w} + 2\cos \Lambda_{c/4,\mathrm{w}}} + \left(\frac{\Delta c_{m,0}}{\theta_{\mathrm{w}}}\right) \theta_{\mathrm{w}} \right] \left(\frac{(c_{m,0})_M}{(c_{m,0})_{M=0}}\right) \tag{3.3.6}$$

where $c_{m,0}$ represents the pitching moment coefficient of the airfoil, $\Lambda_{c/4,\mathrm{w}}$ is the wing sweep angle at the quarter chord, θ_{w} the twist angle, $\frac{\Delta c_{m,0}}{\theta_{\mathrm{w}}}$ is a systematically negative factor accounting for the twist effect, and $\frac{(c_{m,0})_M}{(c_{m,0})_{M=0}}$ represents the compressibility correction factor. It can also be noted that this model assumes that the wing twist is linear.

In Equation 3.3.6, the values of the factors $\frac{(c_{m,0})_M}{(c_{m,0})_{M=0}}$ and $\frac{\Delta c_{m,0}}{\theta}$ was calculated based on digitized curves. This operation is a key point to ensure that the models are implemented in a way that respects a possible optimization algorithm.

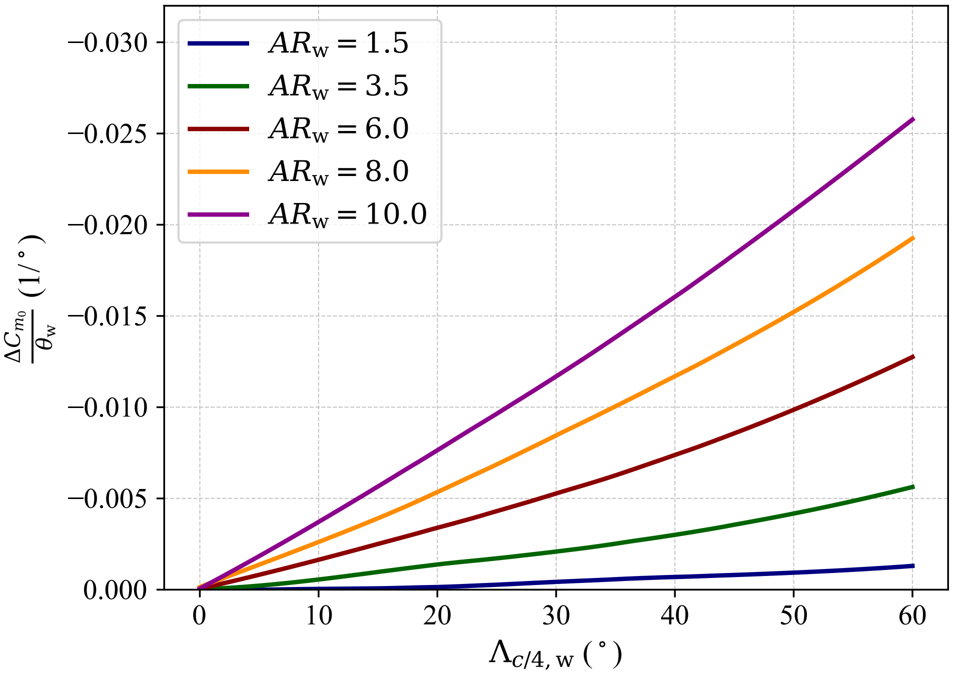


Figure 3.5: Illustration of the pitching moment contribution due to twist for a taper ratio of $\lambda = 0.5$. Reproduced from DATCOM [18].

The evolution of the twist-effect term as a function of quarter-chord sweep angle is shown in Figure 3.5, for various aspect ratios and a fixed taper ratio of $\lambda_w = 0.5$. This figure is reproduced from DATCOM [18].

In order to compute this term within the model, three DATCOM charts were digitized, corresponding to taper ratios of $\lambda_w = 0$, 0.5, and 1.0. Since these values are discrete, interpolation was required to obtain results for intermediate taper ratios. A similar interpolation procedure was applied for the aspect ratio, as the curves in Figure 3.5 represent only a limited set of discrete values. This approach ensures smooth variation of the twist effect term for arbitrary combinations of taper and aspect ratios, which is essential in optimization workflows.

The main adaptation made to the semi-empirical model to enable gradient-based optimization concerns the extension of its valid input domain for the aspect ratio. As shown in Figure 3.5, the original data is limited to a maximum aspect ratio of 10, beyond which the twist effect is not defined. To address this, a linear extrapolation strategy was implemented. This allows the model to provide estimates even when the optimization process explores aspect ratio values beyond the original range. However, this extrapolation introduces certain limitations: for very high aspect ratios, the predicted twist effect may no longer reflect realistic aerodynamic behavior.

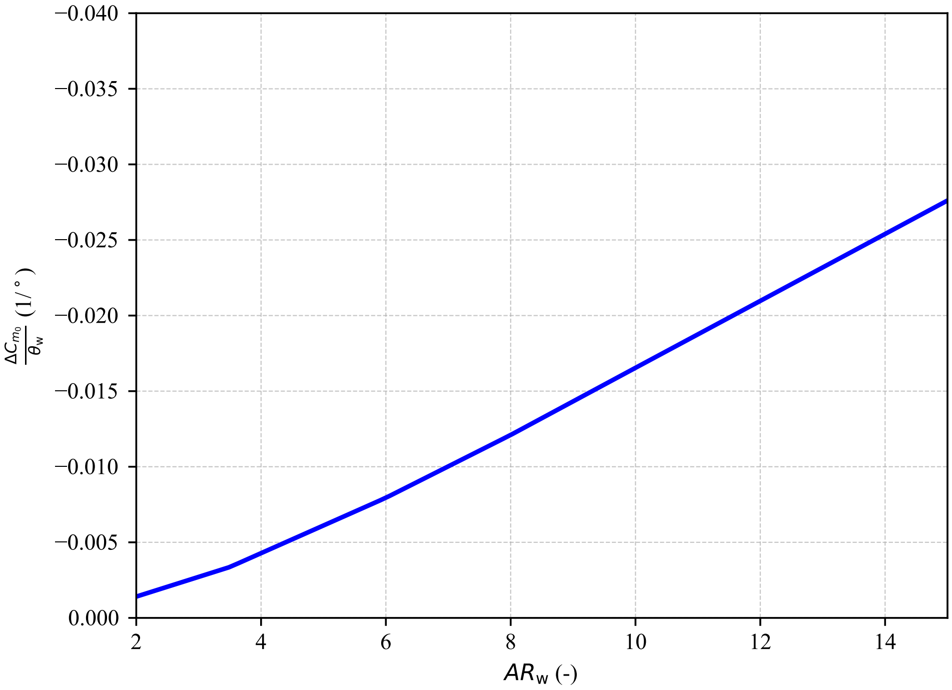


Figure 3.6: Evolution of the twist-effect factor as a function of aspect ratio, for $\lambda_\mathrm{w} = 0.5$ and $\Lambda_{c/4,\mathrm{w}} = 50°$.

The type of extrapolation applied was selected by analyzing the behavior of the twist effect factor $\frac{\Delta c_{m,0}}{\theta_\mathrm{w}}$ for fixed values of sweep angle and taper ratio. The goal was to ensure a smooth continuation of the trend, avoiding abrupt changes in the gradient near the upper limit of the available aspect ratio data (i.e., $AR_\mathrm{w} = 10$). This behavior is illustrated in Figure 3.6, where the quarter-chord sweep angle is set to 50° and the taper ratio to 0.5. The figure confirms that a linear extrapolation is appropriate, as the factor already exhibits a nearly linear trend between $AR_\mathrm{w} = 4$ and $AR_\mathrm{w} = 10$.

The pitching moment coefficient of the horizontal tail about its aerodynamic center is neglected in this model. This simplification is justified by the fact that the airfoil used for the horizontal tail is often symmetric, resulting in a negligible pitching moment contribution.

Once the lift and pitching moment coefficients have been computed, the next step is to address the drag analysis. As with the other disciplines, the implementation is structured in a modular way by dividing the code into separate models for each aircraft component. This architecture allows users to include only the components relevant to their configuration.

The drag analysis is structured into three main components. First, the parasite drag is computed separately for each aircraft element. Then, additional contributions such as miscellaneous drag are included. Finally, for the lifting surfaces, the induced drag is evaluated.

The evaluation of parasite drag is based on the component buildup method described by Raymer [72]. For each component, the zero-lift drag coefficient is calculated using the following relation:

$$C_{D_{parasite,c}} = \frac{C_{f_c}\, FF_c\, Q_c\, S_{\mathrm{wet},c}}{S_\mathrm{ref}} \tag{3.3.7}$$

In this expression, C_{f_c} denotes the flat-plate skin friction coefficient of the component, FF_c is a form factor accounting for pressure drag due to viscous separation, and Q_c represents the interference effects between the component and the rest of the aircraft. $S_\mathrm{wet,c}$ is the wetted surface area of the component, and S_ref is the reference surface area, which is taken to be the wing planform area.

The wetted areas of the components, as well as the wing planform area, are defined as inputs of the model. These quantities are computed within the geometry discipline, either through CAD analysis or analytical formulas, depending on the geometry option selected by the user.

The interference factor Q_c is assigned a constant value for each component, based on the recommendations provided in Raymer [72].

The flat-plate skin friction coefficient C_{f_c} is determined from the Reynolds number of the flow:

$$Re = \frac{VL}{\nu} \tag{3.3.8}$$

where V is the freestream velocity, ν is the kinematic viscosity of air, and L is the characteristic length of the component. For lifting surfaces such as wings, horizontal and vertical tails, or canards, L corresponds to the mean aerodynamic chord. For other components, such as the fuselage, nacelles, pylons, struts, or external stores, it is taken as the total length of the feature.

Based on the Reynolds number, the flat-plate skin friction coefficient of each component can be estimated. This coefficient combines both laminar and turbulent contributions, weighted according to the expected percentage of laminar flow, which depends on the nature of the component.

The laminar skin friction coefficient is given by:

$$C_f = \frac{1.328}{\sqrt{Re}} \tag{3.3.9}$$

For the turbulent portion, the following empirical relation is used:

$$C_f = \frac{0.455}{\left(\log_{10} Re\right)^{2.58} \left(1 + 0.144M^2\right)^{0.65}} \tag{3.3.10}$$

A linear weighting between the laminar and turbulent contributions, based on the assumed laminar flow percentage, yields the final skin friction coefficient for each component.

The form factor FF_c depends on the type of component considered. For surfaces such as the wing, tail, strut, or pylon, it is evaluated using the following expression:

$$FF_c = \left[1 + \frac{0.6}{(x/c)_m}\frac{t}{c} + 100\left(\frac{t}{c}\right)^4\right]\left[1.34M^{0.18}\left(\cos\Lambda_m\right)^{0.28}\right] \tag{3.3.11}$$

where $(x/c)_m$ is the location of maximum thickness relative to the chord, t/c is the thickness-to-chord ratio, and Λ_m is the sweep angle at the location of maximum thickness.

For the fuselage and smooth canopy, the form factor is calculated as:

$$FF_c = 0.9 + \frac{5}{f^{1.5}} + \frac{f}{400} \tag{3.3.12}$$

where f is the fineness ratio, defined as the length-to-diameter ratio of the body:

$$f = \frac{l}{(4/\pi)A_{\max}} \tag{3.3.13}$$

Here, l is the length of the component and $A_{\max}$ its maximum cross-sectional area.

For nacelles and smooth external stores, the following simplified form factor is used:

$$FF_c = 1 + \frac{0.35}{f} \tag{3.3.14}$$

Once the parasite drag of each component has been evaluated, the remaining step to complete the zero-lift drag coefficient C_{D_0} model is to account for the miscellaneous contributions.

In RAPTOR, a dedicated model has been implemented to estimate the miscellaneous drag associated with elements such as the landing gear, cockpit, and engine installations, following the methodology proposed by Raymer [72] and Torenbeek [87].

The next step in the drag analysis consists in evaluating the induced drag. For the wing, it is computed using the classical expression:

$$C_{D_{\mathrm{i,w}}} = \frac{C_{L,\mathrm{w}}^2}{e_\mathrm{w} \pi AR_\mathrm{w}} \tag{3.3.15}$$

In this relation, $C_{L,\mathrm{w}}$ is the wing lift coefficient, previously obtained through the lift model, and AR_w is the aspect ratio of the corresponding surface, determined within the geometry discipline. The parameter e_w denotes the Oswald efficiency factor.

The estimation of the Oswald factor is based on a hybrid approach combining two empirical formulas. For straight wings, the reference provides the following expression:

$$e_\mathrm{w} = 1.78 \left(1 - 0.045\, AR_\mathrm{w}^{0.68}\right) - 0.64 \tag{3.3.16}$$

For swept wings, defined by a leading-edge sweep angle $\Lambda_{\mathrm{LE,w}}$ greater than 30°, the efficiency factor is given by:

$$e_\mathrm{w} = 4.61 \left(1 - 0.045\, AR_\mathrm{w}^{0.68}\right) \left(\cos \Lambda_{\mathrm{LE,w}}\right)^{0.15} - 3.1 \tag{3.3.17}$$

The reference recommends performing a linear interpolation between Equations 3.3.16 and 3.3.17 when the leading-edge sweep angle is below 30°. This feature is mandatory in the context of optimization, as it is crucial to ensure that the Oswald efficiency factor varies continuously with respect to the sweep angle. A discontinuity in its value could significantly disrupt the optimization process by introducing a loss of smoothness in the design space. This approach, similarly to the adaptation applied to the twist effect in the pitching moment model, is essential to ensure the continuity and differentiability of the model, which are key requirements for gradient-based optimization.

The result of the Oswald efficiency factor implementation which aims to ensure smooth variation with respect to the leading-edge sweep angle, is shown in Figure 3.7. It confirms that the gradient of this quantity with respect to $\Lambda_{\mathrm{LE,w}}$ remains continuous and well-behaved, even in the vicinity of 30°, making it suitable for use in gradient-based optimization.

To evaluate the induced drag of the horizontal tail, a similar approach is employed as that used for the wing.

Once both the zero-lift drag and the induced drag of the lifting surfaces have been evaluated, the total drag coefficient of the aircraft can be computed using the following expression:

$$C_D = C_{D_0} + C_{D_i} \tag{3.3.18}$$

In this relation, C_{D_0} and C_{D_i} represent the sum of the respective drag contributions from all the aircraft components.

Now that the semi-empirical aerodynamic model of RAPTOR has been developed, it must be validated.

The first aspect to be assessed is the lift prediction. For this purpose, a higher-fidelity study, using a VLM (Vortex Lattice Method), conducted on a swept and tapered wing, carried out by Mariën [46], is used as a reference.

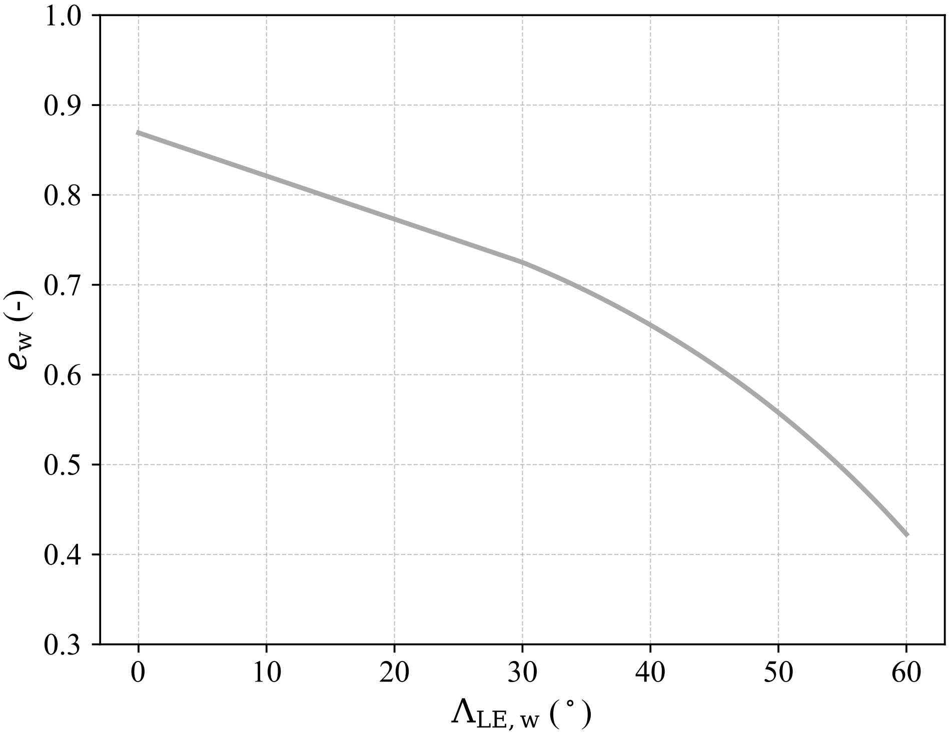


Figure 3.7: Evolution of the Oswald efficiency factor with respect to the leading-edge sweep angle $\Lambda_{\mathrm{LE,w}}$, for an aspect ratio of $AR_{\mathrm{w}} = 6$.

Parameter	**Symbol**	**Value**
Wing airfoil type	–	NACA 0010
Wing span	b_{w}	18.0 m
Wing root chord	$c_{\mathrm{r,w}}$	3.28 m
Wing taper ratio	λ_{w}	0.219
Quarter-chord sweep angle	$\Lambda_{c/4,\mathrm{w}}$	20.0°
Wing dihedral angle	Γ_{w}	0.0°
Wing twist angle	θ_{w}	0.0°

Table 3.5: Main geometrical parameters of the wing used for the lift model validation.

The geometrical parameters used to reproduce the reference case are summarized in Table 3.5. The Mach number is set to 0.2 for this analysis.

The comparison between the lift coefficient predicted by the semi-empirical model of RAPTOR and the reference study is shown in Figure 3.8. The close agreement between the two curves confirms the validity and relevance of the lift model implemented in the present tool.

After validating the lift model, it is essential to demonstrate the reliability of the drag estimation implemented in RAPTOR. To this end, two reference studies from the literature, both based on CFD analyses, are reproduced. These comparisons aim to show that the drag predictions for the wing and fuselage are accurate. This is considered sufficient to validate the overall drag model, since the remaining lifting surfaces are wing-like in nature, and the nacelle drag is modeled using an approach closely aligned with that used for the fuselage.

The validation of the wing drag prediction begins with the study conducted by Elouardi et al. [45], who replicated NASA's CFD simulations on the well-known ONERA M6 wing using RANS (Reynolds-

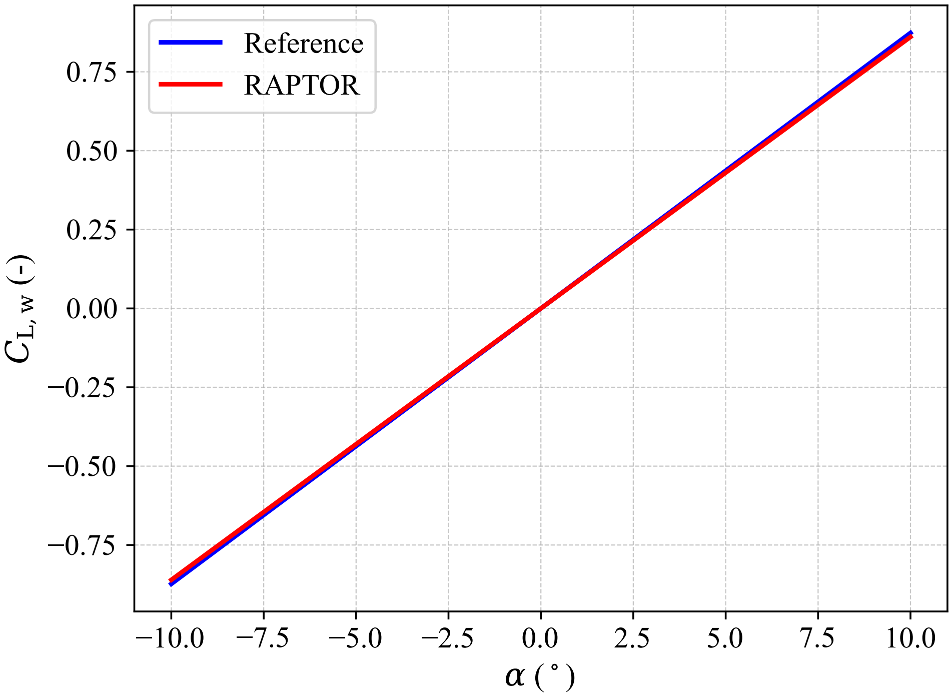


Figure 3.8: Lift coefficient C_L as a function of angle of attack α, compared with Mariën [46].

Averaged Navier-Stockes) turbulence models. Despite being focused on transonic conditions (a Mach number of 0.8395), this study serves as a robust benchmark for assessing the accuracy of the drag model. The angle of attack is set to 3.06°, and the main geometrical characteristics of the ONERA M6 wing are presented in Table 3.6.

Parameter	**Symbol**	**Value**
Wing span	b_{w}	1.196 m
Wing root chord	$c_{\mathrm{r,w}}$	0.8059 m
Wing taper ratio	λ_{w}	0.562
Quarter-chord sweep angle	$\Lambda_{c/4,\mathrm{w}}$	26.7°

Table 3.6: Main geometrical parameters of the ONERA M6 wing used for the drag model validation.

	NASA-CFD	S-A	k-ε Std	k-ε RNG	k-ω Std	k-ω SST	RAPTOR
Drag coefficient $C_{D,\mathrm{w}}$	0.0088	0.0097	0.0145	0.0103	0.0236	0.0096	0.0114

Table 3.7: Comparison of wing drag coefficient $C_{D,\mathrm{w}}$ obtained by Elouardi et al. [45], using various turbulence models and the semi-empirical model of RAPTOR.

The drag coefficient results obtained from the reference study, the original NASA simulation, and the RAPTOR model are summarized in Table 3.7. Depending on the turbulence model used, the prediction from the developed framework shows varying levels of agreement with the reference. Nevertheless, it remains within a reasonable and consistent range, confirming the relevance of the implemented wing drag model.

Concerning the fuselage drag, Nicolosi et al. [64] performed RANS simulations on a fuselage. The key parameters used to replicate this case within the RAPTOR framework are listed in Table 3.8.

Parameter	Symbol	Value
Wing reference area	$S_{\text{ref,w}}$	105 m^2
Fuselage length	l_{fus}	38.0 m
Fuselage max. cross-section area	$A_{\text{max,fus}}$	9.8422 m^2
Cruise Mach number	M	0.62

Table 3.8: Main geometrical and flight parameters of the fuselage used for the drag model validation.

The results obtained for the fuselage drag validation case are summarized in Table 3.9. While a discrepancy is observed between the value predicted by the RAPTOR model and the reference study, both remain of the same order of magnitude. This deviation can be primarily attributed to the difference in modeling fidelity: the reference relies on RANS CFD simulations, whereas RAPTOR uses a semi-empirical approach. Despite this, the result demonstrates that the implemented drag estimation method provides a reasonable and consistent approximation.

	Fuselage parasite drag coefficient
RAPTOR	0.0099
Nicolosi et al. [64]	0.0080

Table 3.9: Comparison between the fuselage parasite drag coefficient predicted by RAPTOR and the CFD-based reference from Nicolosi et al. [64], used to validate the aerodynamic model.

Now that the lift and drag models are validated, it remains to investigate the reliability of the model implemented to predict the wing pitching moment around its aerodynamic center.

To do so, a swept, tapered and twisted wing is considered, and a VSPAERO [69] simulation is conducted, using the Vortex Lattice Method as the higher-fidelity model. The geometrical parameters of the considered wing are summarized in Table 3.10.

Parameter	Symbol	Value
Airfoil type	–	EPPLER 328
Wing span	b_{w}	18.0 m
Wing root chord	$c_{\text{r,w}}$	4.0 m
Wing taper ratio	λ_{w}	0.25
Wing quarter-chord sweep angle	$\Lambda_{c/4,\text{w}}$	10.0°
Wing dihedral angle	Γ_{w}	0.0°
Wing twist angle	θ_{w}	–3.0°

Table 3.10: Main geometrical parameters of the wing used for the pitching moment model validation.

Since the moment coefficient is computed about the aerodynamic center, several pre-processing simulations are carried out to determine its exact location. This is done by adjusting the reference point for the moment calculation until the derivative of the pitching moment coefficient C_m with respect to the angle of attack becomes zero. Through this process, the aerodynamic center is found to be located at an x-coordinate of 1.605 m.

The comparison between the pitching moment coefficients obtained from RAPTOR and VSPAERO is presented in Table 3.11. The observed difference remains within an acceptable range, especially

	Wing pitching moment coefficient $C_{m,\mathrm{w}}$
RAPTOR	–0.00944
VSPAERO	–0.01160
Relative difference	18.62%

Table 3.11: Comparison between the wing pitching moment coefficient $C_{m,\mathrm{w}}$ predicted by RAPTOR and VSPAERO [69], used to validate the pitching moment model.

when considering the difference in modeling fidelity between the two tools. While VSPAERO relies on a higher-fidelity vortex lattice method, RAPTOR uses a semi-empirical approach that offers reduced computational efforts. This confirms the ability of the developed model to provide a reasonably accurate estimation of the moment coefficient.

3.3.2 Source-Doublet Panel Method model

The second aerodynamic model integrated into RAPTOR is based on the Source-Doublet Panel Method (SDPM), developed by Crovato [10]. The choice of this software is motivated by its relevance to the aircraft configurations analyzed, as well as its continuous maintenance and improvement. Its inclusion within the framework further highlights the multi-fidelity capabilities available to RAPTOR users. In this case, as with the higher-fidelity tools used for weight estimation, the external software coupled with RAPTOR has already been validated. Consequently, this validation is not reproduced in the present report.

The main inputs required by the class include the wing geometry, flow conditions (such as angle of attack and Mach number), airfoil shape, and lifting surface characteristics. Additionally, the reference point for the moment calculation must be specified. Based on these inputs, the objective is to compute the lift coefficient C_L, the lift curve slope C_{L_α}, the pitching moment coefficient C_m, and its derivative with respect to the angle of attack C_{m_α}.

To enable this functionality, the first step involves automating the mesh generation process from the provided geometric inputs. This is achieved through the integration of Gmsh [20] within the RAPTOR framework. To reduce computational cost, the symmetry capabilities of the SDPM solver are used, and only half of the lifting surface is modeled.

Once the mesh is generated, the relevant configuration parameters, the mesh itself, the reference point for moment coefficient computation, the reference surface area and chord length, as well as the freestream flow conditions, are used as inputs to the SDPM-based model, which is then used to carry out the aerodynamic panel method calculations.

From there, the aerodynamic lift and pitching moment coefficients of the lifting surface, as well as their derivative with respect to the angle of attack are obtained.

The overall workflow of this aerodynamic ccomponent for lifting surfaces is illustrated in Figure 3.9.

The two aerodynamic models presented in this section reflect the multi-fidelity capabilities offered by the RAPTOR framework. The semi-empirical approach provides a lower level of fidelity compared to the SDPM-based panel method, particularly in the prediction of lift and pitching moment. However, in the current version of RAPTOR, only the semi-empirical model supports automatic differentiation for computing aerodynamic gradients. In contrast, the SDPM-based model relies on finite difference approximations.

An important feature for future development is to enable gradient computation via automatic

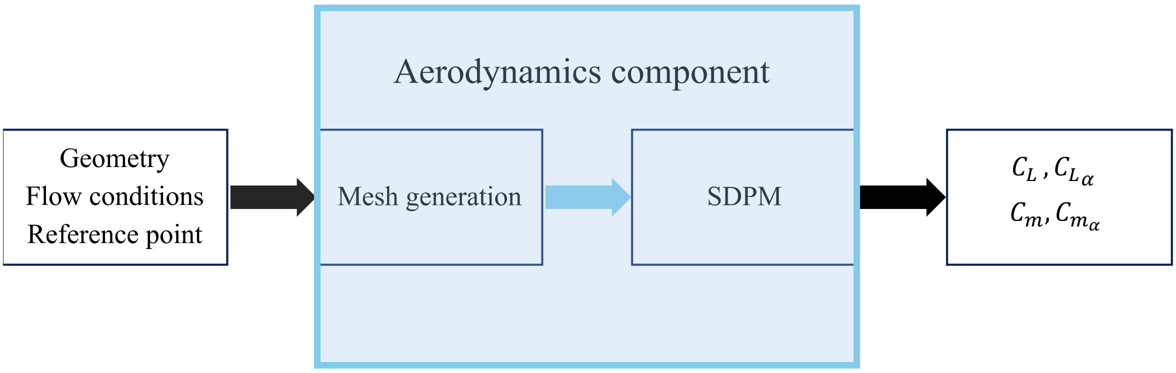


Figure 3.9: Workflow of the SDPM-based aerodynamic component implemented in RAPTOR.

differentiation in the SDPM-based class. While SDPM is already capable of providing analytical gradients with respect to the spatial coordinates (x, y, z) at each mesh node, obtaining gradients with respect to high-level geometric parameters (span, root chord, taper ratio) requires an additional layer of differentiation. Specifically, a mesh deformation algorithm must be implemented to relate changes in global parameters to the displacement of mesh nodes. This enhancement is technically feasible and has been identified as a key objective in the future development of RAPTOR.

3.4 Static stability

In aircraft design, ensuring stability is a fundamental requirement and is often treated as a constraint within the optimization process. Regardless of how efficient the aerodynamic or propulsive performance may be, an aircraft that lacks stability cannot be considered viable for flight.

To evaluate the stability of the aircraft, only the static stability analysis has been implemented in the current version of RAPTOR. The integration of dynamic stability assessment is planned as part of future developments of the software.

In line with the multi-fidelity approach adopted by RAPTOR, two modeling options are available for assessing longitudinal static stability: a semi-empirical model and a higher-fidelity model based on the SDPM panel method. Additionally, a semi-empirical model has been implemented to estimate static lateral stability characteristics.

3.4.1 Semi-empirical model

In the lower-fidelity model, the methodology proposed by Raymer [72] is employed, while the higher-fidelity approach relies on the SDPM software for a more detailed aerodynamic analysis.

To ensure longitudinal static stability, the derivative of the pitching moment coefficient with respect to the angle of attack must be negative, i.e., $C_{m_\alpha} < 0$. Based on the aircraft external loading diagram shown in Figure 3.10, and assuming that the pitching moment of the horizontal tail about its own aerodynamic center is negligible, the aircraft moment coefficient about the center of gravity is given by:

$$C_{m,\text{cg}} = C_{L,\text{w}} \frac{x_{\text{cg}} - x_{\text{ac,w}}}{\bar{c}} + C_{m,\text{w}} + C_{m,\text{fus}} - \frac{S_{\text{ht}}}{S_{\text{w}}} C_{L_{\text{ht}}} \frac{x_{\text{ac, ht}} - x_{\text{cg}}}{\bar{c}} \tag{3.4.1}$$

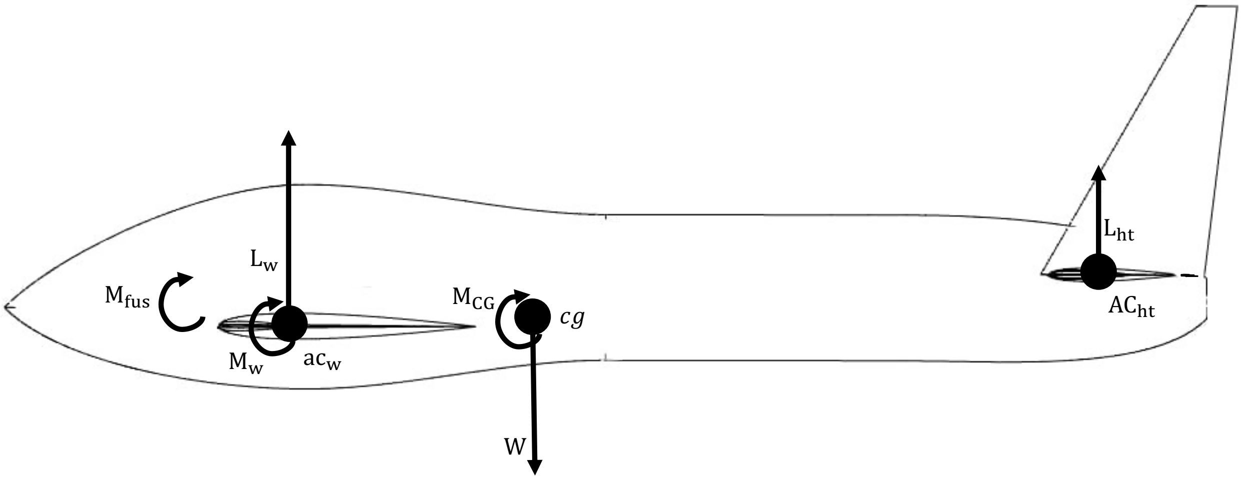


Figure 3.10: Schematic representation of aerodynamic loads and reference points used in stability analysis.

Here, $x_{\text{ac,w}}$ and $x_{\text{ac, ht}}$ denote the aerodynamic centers of the wing and horizontal tail, respectively, and are assumed to be located at the quarter-chord of their respective mean aerodynamic chords. x_{cg} is the aircraft center of gravity position with respect to the its nose.

By differentiating Equation 3.4.1 with respect to the fuselage angle of attack α, and using the expressions for the lift coefficients of the wing and tail (see Equations 3.3.1 and 3.3.3), the expression for the moment coefficient derivative becomes:

$$C_{m_\alpha} = C_{L_{\alpha,\text{w}}} \frac{x_{\text{cg}} - x_{\text{ac,w}}}{\bar{c}} + C_{m_{\alpha,\text{fus}}} - \eta_{\text{ht}} \left(1 - \frac{d\epsilon}{d\alpha}\right) \frac{S_{\text{ht}}}{S_{\text{w}}} C_{L_{\alpha,\text{ht}}} \frac{x_{\text{ac,ht}} - x_{\text{cg}}}{\bar{c}} \tag{3.4.2}$$

The wing and horizontal tail lift curve slopes are determined within the aerodynamics discipline, while their respective planform areas, S_{w} and S_{ht}, are provided by the geometric definition of the aircraft. The horizontal tail efficiency factor, η_{ht}, is set to 0.9, a conservative value commonly adopted in preliminary design. The downwash gradient with respect to the angle of attack, $\frac{d\epsilon}{d\alpha}$, is estimated using the methodology proposed in the DATCOM handbook [18].

The fuselage pitching moment derivative with respect to the angle of attack, $C_{m_{\alpha,\text{fus}}}$, is computed using the empirical formula proposed by Raymer [72]:

$$C_{m_{\alpha,\text{fus}}} = \frac{K_{\text{fus}} W_{\text{fus}} L_{\text{fus}}}{\bar{c} S_{\text{w}}} \tag{3.4.3}$$

Here, K_{fus} is an empirical pitching moment factor obtained by digitizing the corresponding curve from Raymer's reference, and W_{fus} and L_{fus} denote the fuselage width and length, respectively.

The neutral point corresponds to the position of the center of gravity for which the pitching moment derivative with respect to the angle of attack vanishes $C_{m_\alpha} = 0$. Solving for x_{cg} under this condition yields the position of the neutral point x_{np}:

$$x_{\text{np}} = \frac{C_{L_{\alpha,\text{w}}} x_{\text{ac,w}} - C_{m_{\alpha,\text{fus}}} + \eta_{\text{ht}} \left(1 - \frac{d\epsilon}{d\alpha}\right) \frac{S_{\text{ht}}}{S_{\text{w}}} C_{L_{\alpha,\text{ht}}} x_{\text{ac,ht}}}{C_{L_{\alpha,\text{w}}} + \eta_{\text{ht}} \left(1 - \frac{d\epsilon}{d\alpha}\right) \frac{S_{\text{ht}}}{S_{\text{w}}} C_{L_{\alpha,\text{ht}}}} \tag{3.4.4}$$

Longitudinal (or pitch) stability is thus evaluated by computing the static margin, which represents the distance between the aircraft's center of gravity and its neutral point, expressed as a fraction of the wing's mean aerodynamic chord. It provides a direct measure of the stability level: the larger the margin, the more statically stable the aircraft. The static margin is defined as:

$$SM = \frac{x_{\text{np}} - x_{\text{cg}}}{\bar{c}} \tag{3.4.5}$$

The main outputs of the model are therefore the static margin and the derivative of the aircraft pitching moment coefficient with respect to the angle of attack, respectively given by Equation 3.4.5 and 3.4.2.

Now that the longitudinal stability model is established, it must be validated. This is done by computing the relevant stability derivative using a Vortex Lattice Method with VSPAERO [69] and comparing the result with the one obtained from RAPTOR. Since VLM does not account for the fuselage, the analysis is carried out on a configuration composed solely of the wing and horizontal tail. The corresponding geometry is summarized in Table 3.12. The wing and horizontal tail configuration used in the VLM computation is illustrated in Figure 3.11.

Parameter	**Symbol**	**Value**
Wing span	b_{w}	26.6 ft
Wing root chord	$c_{\text{r,w}}$	6.3 ft
Wing tip chord	$c_{\text{t,w}}$	2.52 ft
Wing sweep angle (c/4)	$\Lambda_{c/4,\text{w}}$	0°
Horizontal tail span	b_{ht}	10.1 ft
Horizontal tail root chord	$c_{\text{r,ht}}$	3.6 ft
Horizontal tail tip chord	$c_{\text{t,ht}}$	1.4 ft
Horizontal tail sweep angle (c/4)	$\Lambda_{c/4,\text{ht}}$	12°
Wing leading edge position	–	0.0 ft
Horizontal tail leading edge position	–	12.665 ft
Reference point location from wing LE	–	0.873 ft

Table 3.12: Main geometrical parameters of the configuration used for longitudinal stability validation.

The analysis is conducted at Mach 0.15 and an altitude of 8000 ft. The results are presented in Table 3.13. Given the difference in model fidelity, the discrepancy observed remains well within acceptable limits.

Stability derivative	**VSPAERO**	**RAPTOR**
Pitch stability derivative C_{m_α} (1/rad)	−1.17	–0.86

Table 3.13: Comparison of longitudinal pitch stability derivative between VSPAERO[69] and RAPTOR.

Once longitudinal (pitch) stability has been ensured, the aircraft must also demonstrate stability in yaw. This is assessed by examining the derivative of the yawing moment coefficient with respect to the sideslip angle, denoted C_{n_β}.

Lateral static stability is traditionally evaluated by summing the contributions from three main components:

$$C_{n_\beta} = \left(C_{n_\beta}\right)_{\text{w}} + \left(C_{n_\beta}\right)_{\text{fus}} + \left(C_{n_\beta}\right)_{\text{vt}} \tag{3.4.6}$$

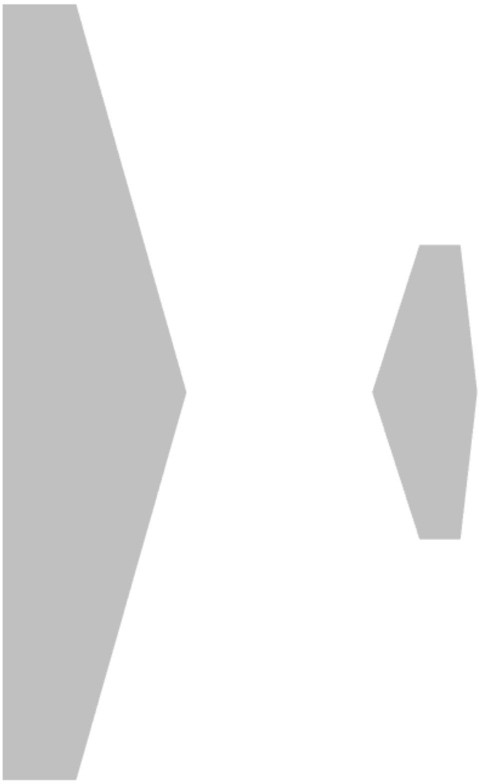


Figure 3.11: Wing and horizontal tail configuration used in the VLM analysis.

where $\left(C_{n_\beta}\right)_{\text{w}}$, $\left(C_{n_\beta}\right)_{\text{fus}}$, and $\left(C_{n_\beta}\right)_{\text{vt}}$ refer to the wing, fuselage, and vertical tail contributions, respectively.

Following Roskam [75], the wing contribution to directional stability is generally negligible:

$$\left(C_{n_\beta}\right)_{\text{w}} \approx 0.0 \tag{3.4.7}$$

The fuselage contribution is estimated using the empirical formula proposed by Raymer [72]:

$$\left(C_{n_\beta}\right)_{\text{fus}} = -1.3\,\frac{Vol_{\text{fus}}}{b_{\text{w}} S_{\text{w}}} \frac{D_{\text{fus}}}{W_{\text{fus}}} \tag{3.4.8}$$

where Vol_{fus} is the total fuselage volume, D_{fus} is the fuselage depth, W_{fus} its width, b_{w} the wing span, and S_{w} the wing reference area.

The final and most critical contribution to the yaw stability derivative comes from the vertical tail. This component is evaluated using the formulation proposed by Roskam [75]:

$$\left(C_{n_\beta}\right)_{vt} = -C_{Y_\beta} \frac{l_{\text{vt}} \cos\alpha + z_{\text{vt}} \sin\alpha}{b_{\text{w}}} \tag{3.4.9}$$

where l_{vt} and z_{vt} are the longitudinal and vertical distances between the aerodynamic center of the vertical tail and the aircraft center of gravity, and C_{Y_β} is the side force derivative with respect to the sideslip angle. This latter term is computed as follows:

$$C_{Y_\beta} = -k_V\, C_{L_{\alpha,\text{vt}}} \left(1 + \frac{d\sigma}{d\beta}\right) \eta_{\text{vt}} \frac{S_{\text{vt}}}{S_{\text{w}}} \tag{3.4.10}$$

Here, k_V is an empirical correction factor obtained from interpolated reference curves, $C_{L_{\alpha,\mathrm{vt}}}$ is the lift curve slope of the vertical tail (computed similarly to that of the wing, as in Equation 3.3.2, but using an effective aspect ratio), and $\left(1 + \frac{d\sigma}{d\beta}\right)\eta_{\mathrm{vt}}$ is a correction term dependent on the configuration geometry, also derived from Roskam's relations.

A key modification in this model lies in the computation of the effective aspect ratio of the vertical tail. This is expressed as:

$$AR_{\mathrm{vt,eff}} = \left[\frac{AR_{\mathrm{vt,f}}}{AR_{\mathrm{vt}}}\right] AR_{\mathrm{vt}} \left[1 + K_{\mathrm{VH}} \left(\frac{AR_{\mathrm{vt,hf}}}{AR_{\mathrm{vt}}} - 1\right)\right] \tag{3.4.11}$$

where AR_{vt} is the actual aspect ratio of the vertical tail, K_{VH} is a correction factor obtained from digitized curves, and the terms $\frac{AR_{\mathrm{vt,f}}}{AR_{\mathrm{vt}}}$ and $\frac{AR_{\mathrm{vt,hf}}}{AR_{\mathrm{vt}}}$ represent correction ratios accounting for the fuselage and horizontal tail intersection effects, respectively.

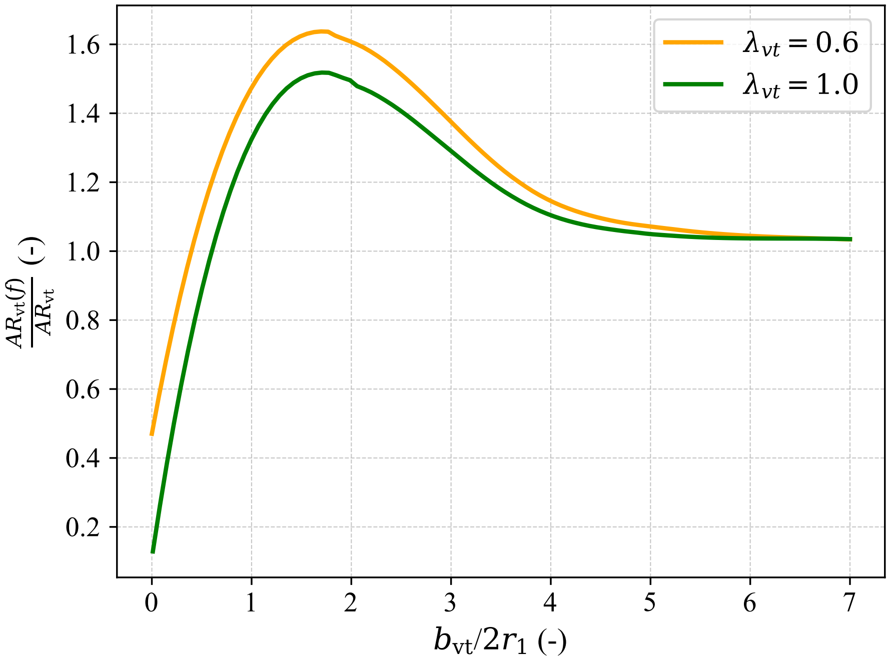


Figure 3.12: Fuselage effect in the computation of the effective aspect ratio of the vertical tail. Reproduced from Roskam [75].

To ensure compatibility with the optimization framework, particular attention must be paid to the interpolation of empirical correction terms. For example, the value of $\frac{AR_{\mathrm{vt,f}}}{AR_{\mathrm{vt}}}$ is extracted from Figure 3.12, which has been digitized from the original reference. In this context, r_1 denotes the fuselage depth in the region of the vertical tail, and b_{vt} represents the span of the vertical tail.

Since the reference figure only provides data for taper ratios ranging from 0.6 to 1.0, a linear interpolation is applied within this interval. To accommodate configurations with taper ratios below 0.6, where no data is available, a linear extrapolation is employed. This approach extends the model's applicability and ensures smooth behavior during gradient evaluations in the optimization process. Without such handling, the absence of defined values in these regions could lead to discontinuities or unphysical behavior when computing derivatives near the lower bound, ultimately causing numerical instabilities.

The validation of this model is presented in the performance Section 3.6, since it relies on the same test case.

3.4.2 Source-Doublet Panel Method model

As previously discussed, two models are available for assessing pitch stability. The second, higher-fidelity approach relies on the Source-Doublet Panel Method (SDPM [10]).

In this context, the static margin can be directly computed using the following relation:

$$SM = -\frac{C_{m_\alpha}}{C_{L_\alpha}} \tag{3.4.12}$$

To apply this formulation within the SDPM-based model, a multi-body simulation is performed in the aerodynamics module, considering both the wing and the horizontal tail as lifting surfaces. The model requires several inputs: the aircraft center of gravity location (used as the reference point for computing C_{m_α}), the geometry of each lifting surface, and the respective leading-edge positions. Based on these inputs, the aerodynamic derivatives are computed, and the static margin is then evaluated using Equation 3.4.12.

3.5 Propulsion

In the current version of RAPTOR, propulsion is not the central focus of the optimization process. However, in order to enable consistent performance evaluation and support validation of test cases, a simplified semi-empirical propulsion module has been implemented. This module provides basic estimations of thrust and propulsive efficiency for propeller-driven aircraft, based on typical input parameters such as flight conditions and engine characteristics.

3.5.1 Semi-empirical model

The first key parameter evaluated in the simplified propulsion model is the design advance ratio of the propeller, defined as:

$$\mathcal{J}_{\text{design}} = \frac{V_{\text{design}}}{nD} \tag{3.5.1}$$

where V_{design} is the design flight velocity, n is the rotational speed of the propeller, and D is the propeller diameter.

Next, the power coefficient c_P is computed using the following expression:

$$c_P = \frac{\mathcal{P}}{\rho n^3 D^5} \tag{3.5.2}$$

where $\mathcal{P}$ is the shaft power delivered to the propeller and ρ is the air density.

Using these two parameters, the design propulsive efficiency $\eta_{p_{\text{design}}}$ is estimated from a reference chart provided in Raymer [72], which relates η_p to the power coefficient c_P and the advance ratio $\mathcal{J}$ for a typical three-bladed propeller. If the propeller under consideration differs from the reference case in terms of blade number or material, appropriate correction factors are applied.

To allow evaluation under off-design conditions, a correction to the propulsive efficiency is made based on the ratio $\mathcal{J}/\mathcal{J}_{\text{design}}$, using another chart from the same reference. This chart provides the efficiency ratio $\eta_p/\eta_{p_{\text{design}}}$ as a function of $\mathcal{J}/\mathcal{J}_{\text{design}}$, allowing the model to adapt to varying flight conditions.

Finally, the resulting thrust is calculated using the following relation:

$$\mathcal{T} = \frac{\eta_p \mathcal{P}}{V} \tag{3.5.3}$$

where V is the actual flight speed at which the thrust is to be estimated.

The validation of this model is presented in the following section, since it relies on the same test case used for the performance analysis.

3.6 Performances

In aircraft design, performance evaluation plays a central role, as several key metrics, such as range, endurance, cruise speed, or rate of climb, are either subject to strict requirements or must be optimized to meet specific operational goals. These performance parameters often act as constraints or objectives within the design process, guiding the sizing and configuration of the aircraft. A precise and reliable assessment of performance is therefore essential to ensure that the final design meets both regulatory standards and mission needs.

In its current version, RAPTOR is capable of evaluating several key performance metrics of the aircraft, including the cruise range, the rate of climb, the takeoff distance, and the stall speed. A natural direction for future development is to extend this list by incorporating additional features.

3.6.1 Semi-empirical model

The model developed in this work to estimate the cruise range is based on the Breguet range equation for propeller-driven aircraft:

$$\text{Range} = 550 \frac{\eta_p}{C_{\text{bhp}}} \left(\frac{L}{D} \right) \ln \left(\frac{W_i}{W_f} \right) \tag{3.6.1}$$

In this equation, η_p is the propulsive efficiency, and C_{bhp} is the brake horsepower specific fuel consumption. Both quantities can either be provided directly by the user or retrieved from the propulsion discipline. W_i and W_f represent the aircraft weight at the beginning and end of the cruise segment, respectively. The term $\frac{L}{D}$ is the lift-to-drag ratio, evaluated under cruise conditions. It is an output of the aerodynamics discipline.

In RAPTOR, the rate of climb is computed using the classical expression from Raymer [72]:

$$V_V = V_{\text{climb}} \frac{T - D}{W_{\text{climb}}} \tag{3.6.2}$$

Here, V_{climb} is the aircraft horizontal true airspeed during the climb phase, T is the available thrust (provided either directly by the user or computed via the propulsion module), and W_{climb} is the aircraft weight during climb, estimated as the gross takeoff weight minus the fuel burned during the takeoff segment. The drag force D is an output of the aerodynamics discipline.

The takeoff distance estimation in RAPTOR is based on the use of a takeoff parameter (TOP), defined as:

$$\text{TOP} = \frac{W/S_{\text{w}}}{\sigma \, C_{L_{\text{TO}}} \, (T/W)} \tag{3.6.3}$$

where σ is the relative air density (the ratio between the current air density and the standard sea-level density), $C_{L_{\text{TO}}}$ is the lift coefficient during takeoff, and T/W is the thrust-to-weight ratio.

According to Raymer [72], the takeoff speed is approximately 1.1 times the stall speed. Therefore, the effective takeoff lift coefficient is taken as the maximum lift coefficient divided by 1.21. Using Equation 3.6.3 and digitized reference curves from Raymer [72], the actual takeoff distance is estimated.

The last performance parameter computed in the current version of RAPTOR is the stall velocity. To calculate it, a simple equilibrium between the lift and weight is performed, resulting in the following relation:

$$V_{stall} = \sqrt{\frac{2W}{\rho S_{\mathrm{w}} C_{L_{\max}}}} \tag{3.6.4}$$

Then, a simple formula from Anderson [4] is used to compute $C_{L_{\max}}$:

$$C_{L_{\max}} = 0.95 \frac{c_{l_{\max,\mathrm{root}}} + c_{l_{\max,\mathrm{tip}}}}{2} \tag{3.6.5}$$

Now that the performance model has been selected, developed, and adapted, it must be validated. For this purpose, the single-seat aerobatic aircraft presented in Raymer book [72], and also analyzed by Thu et al. [85], is used as a reference case. The stability results generated by RAPTOR are directly compared with those reported in the study by Thu et al. [85], which investigates the same aircraft configuration. The analysis part presented in that paper serves as a validation reference. The main geometric parameters used for this comparison are summarized in Table 3.14.

Parameter	**Value**
Wing root chord $c_{\mathrm{r,w}}$	6.25 ft
Wing tip chord $c_{\mathrm{t,w}}$	2.53 ft
Wing span b_{w}	26.60 ft
Wing taper ratio λ_{w}	0.40
Wing aspect ratio AR_{w}	6.10
Horizontal tail root chord $c_{\mathrm{r,ht}}$	3.60 ft
Horizontal tail tip chord $c_{\mathrm{t,ht}}$	1.40 ft
Horizontal tail span b_{ht}	10.10 ft
Horizontal tail aspect ratio AR_{ht}	4.00
Vertical tail root chord $c_{\mathrm{r,vt}}$	4.00 ft
Vertical tail tip chord $c_{\mathrm{t,vt}}$	1.60 ft
Vertical tail span b_{vt}	4.10 ft
Vertical tail aspect ratio AR_{vt}	1.50

Table 3.14: Main geometrical parameters of the aircraft used for stability model validation.

To validate the lateral stability model, as well as the propulsion and performance estimations, the SSA aircraft case is analyzed under representative cruise conditions: 8000 ft altitude, Mach 0.15, and an angle of attack of 3°. The wing uses a NACA 63-2015 airfoil at the root and a NACA 63-012 at the tip, while both horizontal and vertical tails are equipped with NACA 0012 profiles. The propulsion system is modeled using the LYCOMING O-320-A2B engine, delivering 150 HP at a nominal rotational speed of 2700 RPM, with a specific fuel consumption of $C_{\mathrm{bhp}} = 0.5\,\mathrm{lb/(hp \cdot hr)}$.

A comparison between the values obtained in RAPTOR and those reported in Thu et al.[85] is provided in Table 3.15.

Metric	Reference	RAPTOR
Performance metrics		
Cruise range	207.9 nm	210.90 nm
Rate of climb	1500.00 ft/min	1591.64 ft/min
Takeoff distance	1000.0 ft	1056.0 ft
Stall speed	84.39 ft/s	84.88 ft/s
Lateral stability derivative		
Yaw stability derivative C_{n_β}	0.0717 1/rad	0.0478 1/rad

Table 3.15: Comparison of aerodynamic and performance metrics for the SSA aircraft between values from Thu et al.[85] and the RAPTOR framework.

The results of this validation indicate that the values obtained in both the present and reference studies are very close, with minor discrepancies, certainly due to the differences in the modeling approaches. The model implemented in RAPTOR is therefore validated.

3.7 Structure

In aircraft design, significant attention is typically devoted to optimizing aerodynamic performance in order to meet mission requirements and ensure sufficient stability. Additionally, minimizing the overall mass of the aircraft contributes to improving efficiency and reducing material and operational costs. However, beyond aerodynamic and weight considerations, structural integrity remains a critical and non-negotiable aspect of the design process. The internal structure must be capable of withstanding the various loads encountered throughout the flight envelope, ensuring the safety and reliability of the aircraft. As such, structural constraints must be systematically integrated into the design framework.

In the current version of RAPTOR, structural analysis focuses primarily on the wing, which is not only a key load-bearing component but also the most geometrically complex to model. In a similar way, a model is developed for the fuselage structural calculations.

The structural analysis begins with the construction of the flight envelope, which defines the range of critical flight conditions. This envelope is derived from geometric inputs and results provided by other disciplines (aerodynamics, propulsion, weights). From this, key load cases are identified.

Subsequently, a static equilibrium is established to determine the structural loads acting on the wing, such as bending moments and shear forces. Based on these loads, the model estimates the required structural dimensions, including the minimum cross-sectional area of the stringers, as well as the thicknesses of the skin and spars.

These structural quantities are then used as constraints in the optimization process to ensure that the aircraft remains structurally sound under all relevant operating conditions.

The structural model developed here is primarily based on the reference textbook by Megson [53]. This choice is motivated by the model's inherent compatibility with gradient-based optimization. Its formulation requires only minimal adaptation to enable automatic differentiation, making it particularly well suited for integration within the present optimization framework.

3.7.1 Analytical model

The flight envelope diagram consists of two main components. The first is the manoeuvre envelope, which defines the allowable range of load factors as a function of the equivalent airspeed during manoeuvre. The second is the gust envelope, which accounts for the additional load factors induced by atmospheric gusts. Both envelopes are constructed for a specified flight altitude, where air density and gust intensity are considered constant.

To initiate the evaluation of the flight envelope, the design cruise velocity must first be determined. This is done by considering the aircraft in cruise conditions, with the propulsion system operating at its maximum available power. Under these conditions, the thrust exactly balances the aerodynamic drag force:

$$T = D \tag{3.7.1}$$

By substituting the expressions for thrust and drag, the following equilibrium is obtained:

$$\frac{\eta \mathcal{P}_{\text{max}}}{V_C} = \frac{1}{2}\rho V_C^2 S_{\text{w}} C_D \tag{3.7.2}$$

where η is the engine efficiency, either provided by the user or calculated through the propulsion module, $\mathcal{P}_{\text{max}}$ is the maximum engine power, and V_C is the design cruise speed.

Rearranging the equation and substituting the drag coefficient C_D by its typical form, which includes the zero-lift drag coefficient C_{D_0}, obtained from the aerodynamic module, and the induced drag, primarily attributed to the wing, leads to the expression for the design cruise velocity:

$$V_C^3 = \frac{2\eta \mathcal{P}_{\text{max}}}{\rho S_{\text{w}} \left(C_{D_0} + \frac{C_{L,\text{w}}^2}{e_{\text{w}} \pi AR_{\text{w}}} \right)} \tag{3.7.3}$$

Following the definition of the cruise speed, the design dive speed V_D is derived in accordance with FAR 23.335 [17], leading to:

$$V_D = 1.25 V_C \tag{3.7.4}$$

The construction of the flight envelope is then based on the methodology presented in Megson [53], which serves as the main reference for the structural considerations developed in this section.

The manoeuvre envelope defines the range of permissible load factors the aircraft may encounter during manoeuvres, as a function of equivalent airspeed. The stall boundary for the flaps-up configuration is determined by the lift limit, expressed as:

$$n = \frac{L}{W} = \frac{\rho_0 V_e^2 S_{\text{w}} C_{L_{\text{max}}}}{2W} \tag{3.7.5}$$

where n is the load factor, ρ_0 the sea-level air density, V_e the equivalent airspeed, S_{w} the wing surface, $C_{L_{\text{max}}}$ the maximum lift coefficient, and W the aircraft weight.

The design manoeuvre speed V_A is then defined as the intersection between the stall curve (Equation 3.7.5) and the maximum allowable load factor n_{max}. It represents the highest speed at which full control deflection can be applied without exceeding structural limits.

For speeds above V_A, the positive limit load factor is constant and equal to $n_{\max}$. On the negative side of the envelope, the same logic is applied: at low speeds, the negative load factor follows the opposite of the positive stall curve. Once it reaches the minimum allowable load factor $n_{\min}$, it remains constant up to higher speeds. The final portion of the curve is defined by a linear interpolation from V_C, where $n = -1$, to V_D, where the negative load factor is assumed to reduce to zero.

The gust envelope accounts for structural loads induced by atmospheric turbulence, represented as sudden vertical gusts of velocity U. These gusts generate an additional aerodynamic load, resulting in a variation of the load factor given by:

$$\Delta n = \frac{\rho_0 V_e F S_{\mathrm{w}} C_{L_\alpha} U_e}{2W} \tag{3.7.6}$$

where C_{L_α} is the lift curve slope, obtained from the aerodynamic module, and U_e is the equivalent gust velocity, typically retrieved by interpolating regulatory tables for a given altitude. The term F is the gust alleviation factor, defined as:

$$F = \frac{0.88\mu}{5.3 + \mu} \quad \text{with} \quad \mu = \frac{2W}{C_{L_\alpha} \rho_0 \bar{c} g S_{\mathrm{w}}} \tag{3.7.7}$$

Here, $\bar{c}$ is the wing mean aerodynamic chord, g the gravitational acceleration, and μ the mass ratio of the aircraft.

From this, the total load factor due to gusts is given by:

$$n_g = 1 \pm \Delta n \tag{3.7.8}$$

To assess whether the aircraft is susceptible to stalling under gust conditions, the gust penetration speed V_B is determined. It corresponds to the intersection between the gust load factor curve (Equation 3.7.8) and the stall curve (Equation 3.7.5). If the aircraft flies below V_B, it will stall before reaching the full gust load.

Based on these developments, two key load factors are extracted. The limit load factor $n_{\text{limit}} = n_g(V_C)$, which defines the maximum expected operational load. The ultimate load factor, defined as $1.5 n_{\text{limit}}$, representing the structural load threshold above which permanent deformation may occur.

Figure 3.13 illustrates a representative flight envelope computed using RAPTOR. It displays both the manoeuvre and gust envelopes, along with the corresponding critical points used for structural sizing, and the previously defined characteristic velocities.

Based on the critical load factors and corresponding velocities identified in the flight envelope analysis, it is possible to determine the aerodynamic forces acting on the wing, which is the principal component of the structural model developed in RAPTOR.

The first assumption made is that, at each critical point, the lift generated by the aircraft is equal to the product of the load factor and the aircraft weight, resulting in the following equilibrium:

$$L = nW \tag{3.7.9}$$

By expressing the lift force using the classical aerodynamic relation, one obtains the following expression, which links the angle of attack to known or previously computed quantities:

$$\frac{1}{2}\rho V^2 S_{\mathrm{w}} C_{L_\alpha} (\alpha - \alpha_{L=0}) = nW \tag{3.7.10}$$

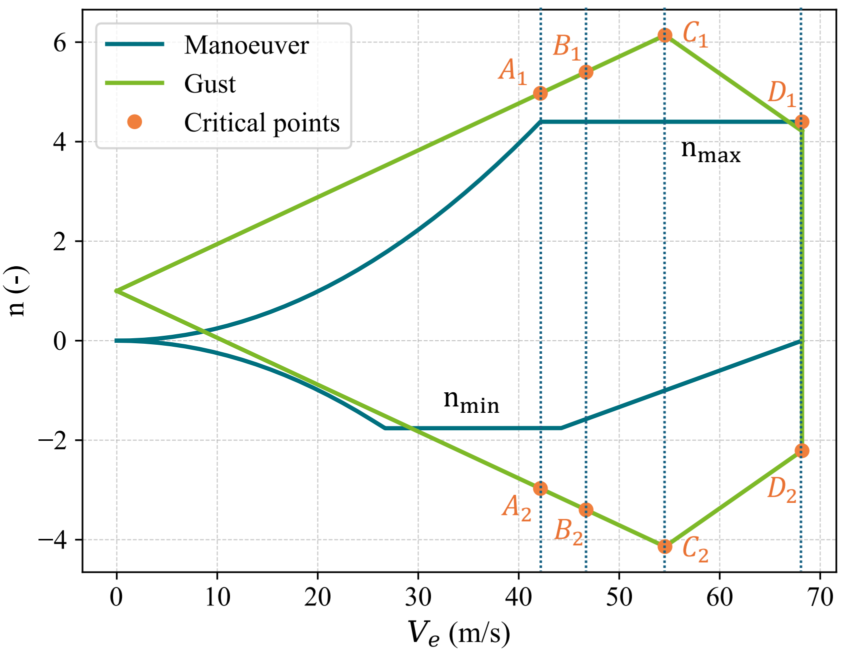


Figure 3.13: Example of a flight envelope computed with RAPTOR.

Here, V is the critical velocity under consideration, C_{L_α} is the aircraft lift curve slope, and $\alpha_{L=0}$ is the zero-lift angle of attack, both provided by the aerodynamics module. This relation allows the determination of the fuselage angle of attack α at each critical condition.

Once the angle of attack is known, the corresponding wing lift coefficient $C_{L,\mathrm{w}}$ can be evaluated using the wing formulation (see Equation 3.3.2).

The drag coefficient on the wing is then calculated using the classical parabolic drag model:

$$C_{D,\mathrm{w}} = C_{D_{0,\mathrm{w}}} + \frac{C_{L,\mathrm{w}}^2}{e_\mathrm{w} \pi AR_\mathrm{w}} \tag{3.7.11}$$

All required parameters, such as the zero-lift drag coefficient $C_{D_{0,\mathrm{w}}}$, the Oswald efficiency factor e_w, and the wing aspect ratio AR_w, are obtained either from the geometry, the aerodynamics discipline, or from previous computations.

Finally, for each critical flight condition, the aerodynamic forces acting on the wing are given by:

$$L_\mathrm{w} = \frac{1}{2}\rho V^2 S_\mathrm{w} C_{L,\mathrm{w}} \quad , \quad D_\mathrm{w} = \frac{1}{2}\rho V^2 S_\mathrm{w} C_{D,\mathrm{w}} \tag{3.7.12}$$

These lift and drag values serve as inputs for the subsequent structural load analysis.

Once the aerodynamic forces acting on the wing have been computed for each critical point of the flight envelope, and the structural weight of the wing is known, the half-wing structural model can be established. This model accounts for the distribution of external loads and internal reactions specific to the semi-span configuration. The applied loads and reference points used in the analysis are illustrated schematically in Figure 3.14.

The structural loads expressions are therefore obtained, by computing the equilibrium equations, and presented in Equation 3.7.13.

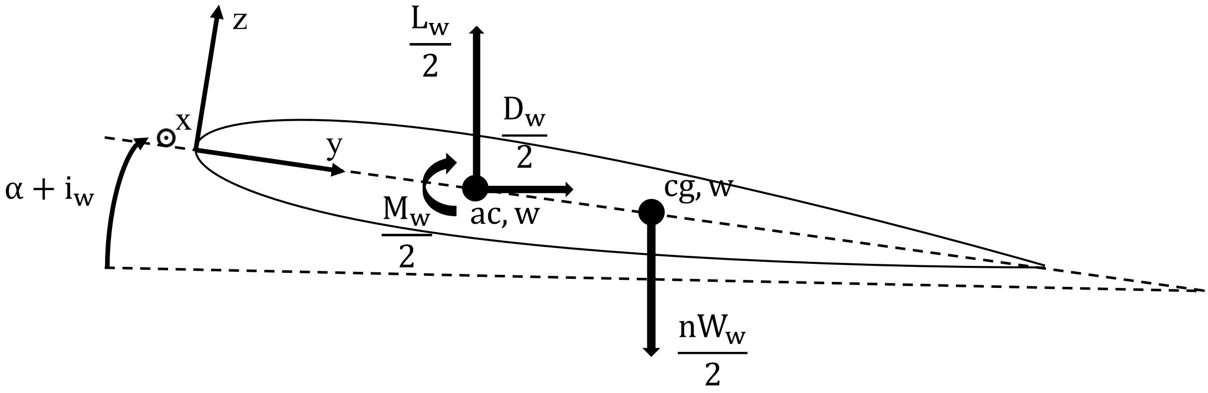


Figure 3.14: Representation of external loads applied on the half-wing.

$$\begin{cases} T_y = \frac{1}{2} D_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) + \frac{1}{2} n W_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) - \frac{1}{2} L_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) \\ T_z = \frac{1}{2} L_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) + \frac{1}{2} D_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) - \frac{1}{2} n W_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) \\ M_x = -\frac{1}{2} n W_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) y_{\mathrm{cg,w}} + \frac{1}{2} D_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) y_{\mathrm{ac,w}} - \frac{1}{2} M_{\mathrm{w}} \\ \qquad + \frac{1}{2} L_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) y_{\mathrm{ac,w}} - \frac{1}{2} n W_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) z_{\mathrm{cg,w}} \\ \qquad + \frac{1}{2} L_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) z_{\mathrm{ac,w}} - \frac{1}{2} D_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) z_{\mathrm{ac,w}} \\ M_y = -\frac{1}{2} D_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) x_{\mathrm{ac,w}} - \frac{1}{2} L_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) x_{\mathrm{ac,w}} + \frac{1}{2} n W_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) x_{\mathrm{cg,w}} \\ M_z = -\frac{1}{2} L_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) x_{\mathrm{ac,w}} + \frac{1}{2} D_{\mathrm{w}} \cos(\alpha + i_{\mathrm{w}}) x_{\mathrm{ac,w}} + \frac{1}{2} n W_{\mathrm{w}} \sin(\alpha + i_{\mathrm{w}}) x_{\mathrm{cg,w}} \end{cases} \tag{3.7.13}$$

In RAPTOR, additional contributions to the external loads, such as the presence of wing-mounted stores or engines generating thrust, can be accounted for within the model.

Based on the structural loads acting on the wing, the necessary size of its internal components can be determined to ensure they withstand the applied stresses.

In conventional aircraft design, the internal wing structure is typically composed of ribs, spars, skin, and stringers. Structurally, a wing section is divided into multiple closed cells by the spars, with each cell bounded by stringers and skin panels. Most classical designs include between one and five or six cells. The RAPTOR framework allows the user to define a variable number of cells, offering considerable flexibility to represent different structural architectures and adapt to diverse aircraft concepts.

The objective of this structural module is to compute, from the applied loads, the minimum required cross-sectional area of the stringers, along with the minimum thicknesses of the spars and skin. These structural outputs are intended to be used as constraints within the overall optimization process. By doing so, the framework ensures that any configuration remains structurally viable, preventing convergence toward designs that would lead to failure under the computed load conditions.

The first step in the structural analysis consists of the geometric modeling of the wing cross-section. To that end, the user must provide the airfoil geometry, the number of stringers, and their coordinates along the airfoil surface. Once the airfoil is discretized, the stringers positions on the section can be obtained by multiplying the coordinated by the chord length. To facilitate a uniform distribution of stringers, RAPTOR includes built-in functions that assist in positioning them at equal arc lengths along the airfoil contour.

It is assumed that the maximum structural loads are experienced at the wing root, which is therefore selected as the reference cross-section for analysis. An example of such a discretized wing section is shown in Figure 3.15. In this configuration, two structural cells are explicitly modeled. A third, rearward region corresponds to the control surface, which does not carry significant bending loads and is therefore excluded from the structural discretization. The skin panels are represented in blue, and the stringers in orange. The centroid of the section is also highlighted, and it is the reference for the stringers coordinates.

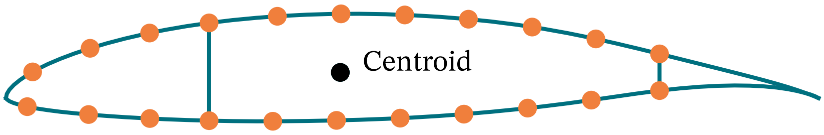


Figure 3.15: Representation of the wing root cross-section, including stringers, spars, skin panels and the centroid.

The structural formulation developed below relies on a set of simplifying assumptions and an idealized representation of the wing box. In this model, the stringers (represented as orange dots in Figure 3.15) are assumed to carry only direct (normal) stress, while the skin panels are modeled as carrying shear stress exclusively. Furthermore, the shear stress in each skin panel is assumed to be uniformly distributed between two adjacent stringers. To ensure that these assumptions remain valid, the spacing between stringers must be sufficiently small and evenly distributed along the airfoil contour.

With the coordinates of all stringers now defined, the next step is to compute their minimum required cross-sectional area. Since stringers are responsible for resisting direct stresses, one must first evaluate the normal stress acting on each boom. The axial force carried by the i-th stringer is expressed as:

$$A_{\text{stringer,w},i}\,\sigma_{xx}^{i} = \frac{\left(M_Y I_{zz} + M_Z I_{yz}\right) z^i - \left(M_Z I_{yy} + M_Y I_{yz}\right) y^i}{I_{yy} I_{zz} - I_{yz}^2} \tag{3.7.14}$$

where y^i and z^i are the coordinates of the i-th stringer in the wing cross-section, $A_{\text{stringer,w},i}$ is its cross-sectional area, and σ_{xx}^{i} is the corresponding direct stress. The moments of inertia are computed per unit area, using the following expressions:

$$I_{yy} = \sum_i (z^i)^2, \quad I_{zz} = \sum_i (y^i)^2, \quad I_{yz} = \sum_i y^i z^i \tag{3.7.15}$$

Once the axial loads $A_{\text{stringer,w},i}\,\sigma_{xx}^{i}$ are known for each stringer, the minimum required cross-sectional area is determined from the most critical loading condition:

$$A_{\text{stringers,w,min}} = \frac{\max\left(\left|A_{\text{stringer,w},i}\,\sigma_{xx}^{i}\right|\right)}{\sigma_{\max}} \tag{3.7.16}$$

Here, $\sigma_{\max}$ is the allowable direct stress in the material, provided by the user and typically reduced by a safety factor of 1.5 to ensure structural integrity under conservative design assumptions.

An important adaptation of the structural model must be introduced at this stage. In Equation 3.7.16, the minimum required cross-sectional area of the stringers is defined as the maximum value of the axial force-to-stress ratio among all stringers. However, the use of a strict `max` operator poses a significant problem in the context of gradient-based optimization. This function is non-differentiable and introduces discontinuities when the maximum value switches from one stringer to another during the optimization process. Additionally, it leads to vanishing gradients with respect to the positions and sizes of all other stringers, except the one currently carrying the maximum load.

To address this issue, a smooth approximation of the maximum function is employed. This continuous and differentiable formulation preserves the optimization flow while maintaining a close representation of the original maximum. The approximation used is derived from Boyd et al. [8], and is expressed as:

$$\max(x_1, x_2, \ldots, x_n) \approx \frac{1}{k} \log\left(\sum_{i=1}^{n} e^{kx_i}\right) \tag{3.7.17}$$

where k is a tunable sharpness parameter: a higher value yields a closer approximation to the true maximum. This smooth formulation ensures differentiability with respect to all inputs and is well-suited in this case.

Once the minimum required cross-sectional area of the stringers has been determined, the next step is to compute the minimum thickness of the skin panels. This requires the evaluation of the shear flow $q(s)$ along each panel, which is composed of two main contributions:

$$q(s) = q_0(s) + q(0) \tag{3.7.18}$$

Here, $q_0(s)$ represents the open-section shear flow, and the term $q(0)$ is a corrective term, introduced to satisfy the compatibility condition of the closed-section structure and ensure equilibrium of the internal moment. Together, these components provide the total shear flow required for sizing the skin.

To compute the open shear flows within each closed cell of the wing section, it is necessary to introduce a virtual cut in one of the skin panels. This cut enforces a boundary condition where the open shear flow $q_0(s)$ is assumed to vanish locally.

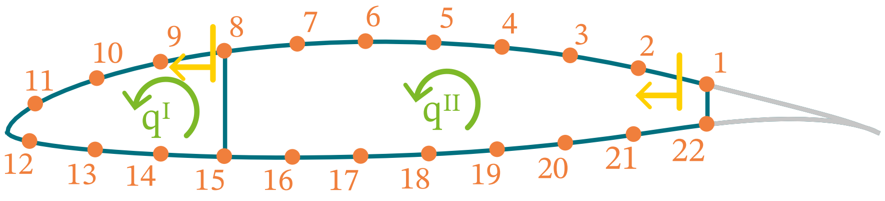


Figure 3.16: Detailed structural representation of the wing section with corrected shear flows and cut locations.

An example of a discretized wing section with numbered stringers, corrected shear flows, and the locations of the virtual cuts (highlighted in yellow) is presented in Figure 3.16. Based on this configuration, the open shear flows are computed using the following expression:

$$q_0(s) = -\frac{I_{yy}\,T_z^{\text{web}} - I_{yz}\,T_y^{\text{web}}}{I_{yy}\,I_{zz} - I_{yz}^2}\left[\sum_{i:\,s_i \le s} y^i\,A_{\text{stringer,w},i}\right] + \frac{I_{zz}\,T_y^{\text{web}} - I_{yz}\,T_z^{\text{web}}}{I_{yy}\,I_{zz} - I_{yz}^2}\left[\sum_{i:\,s_i \le s} z^i\,A_{\text{stringer,w},i}\right] \tag{3.7.19}$$

In this formulation, the only unknown quantities are the effective shear forces transmitted by the web, T_y^{web} and T_z^{web}. These are corrected to account for the redistribution of internal forces resulting from the axial stresses carried by the stringers, particularly when their positions vary along the span. Assuming that the direct stresses align with the orientation of the stringers (i.e., follow the boom direction), the expressions for the corrected shear forces become:

$$T_y^{\text{web}} = T_y - \sum_i A_{\text{stringer,w},i}\,\sigma_{xx}^i\,\frac{dy^i}{dx}, \tag{3.7.20}$$

$$T_z^{\text{web}} = T_z - \sum_i A_{\text{stringer,w},i}\,\sigma_{xx}^i\,\frac{dz^i}{dx}, \tag{3.7.21}$$

where $\frac{dy^i}{dx}$ and $\frac{dz^i}{dx}$ represent the spatial variation of the stringers positions along the spanwise direction. These terms allow the model to capture the influence of planform tapering or geometric changes on the shear load distribution within the wing structure.

Once the open shear flow has been computed, the next step is to determine the corrected shear flow $q(0)$, in order to recover the total shear distribution along the wing panels, as expressed in Equation 3.7.18.

To compute this corrective term, a linear system of equations must be solved. The first equation in this system ensures the equilibrium of moments around a reference point. For convenience, this reference is chosen as the midpoint of the front spar, which is also used to recenter the structural loads model. The resulting moment balance equation is given by:

$$M_x = \sum_i \int_{\text{wall}_i} q_0\,p\,ds + \sum_c 2A_h^c\,q^c(0) + \sum_j y^j\,A_{\text{stringer,w},j}\,\sigma_{xx}^j\,\frac{dz^j}{dx} - \sum_j z^j\,A_{\text{stringer,w},j}\,\sigma_{xx}^j\,\frac{dy^j}{dx} \tag{3.7.22}$$

In this formulation, M_x refers to the pitching moment computed in Equation 3.7.13, with the midpoint of the front spar taken as the reference point. The term $\sum_i \int_{\text{wall}_i} q_0\,p\,ds$ corresponds to the contribution of the open shear flow and is evaluated using a model that estimates the swept area of each skin panel segment. The term A_h^c denotes the area of cell c, computed via geometric approximation methods. The unknowns in this equation are the corrected (constant) shear flows $q^c(0)$ associated with each closed cell.

To close the system of equations and ensure structural compatibility, the twist rate θ_x in each closed cell is computed and enforced to be identical across adjacent cells. The twist rate in a given cell i is expressed as:

The twist angle θ_x^i in cell i is calculated using the following expression:

$$\theta_x^i = \frac{1}{2A_h^i\mu_{\text{REF}}}\left[-q^{i-1}(0)\,\vec{l}_{i-1}^i + q^i(0)\,\vec{l}_i^i + \oint_{\text{cell }i}\frac{q_0}{t_{\text{skin,w}}}\frac{\mu}{\mu_{\text{REF}}}\,ds - q^{i+1}(0)\,\vec{l}_{i+1}^i\right] \tag{3.7.23}$$

In this equation, A_h^i represents the area of the closed cell i, while μ and μ_{REF} are the local and reference shear moduli, assumed uniform here. The term $t_{\text{skin,w}}$ denotes the local skin thickness of the wing, also assumed to be constant. The quantities $q^i(0)$ are the constant shear flows in the corresponding cells. The vectors $\vec{l}_j^i$ represent the effective lengths of wall segments contributing to cell i, shared with adjacent segments j.

By enforcing $\theta_x^i = \theta_x^{i+1}$ for all adjacent cells, a set of compatibility equations is added to the system, enabling the unique resolution of all unknown corrected shear flows $q^i(0)$.

Once the shear flow distribution $q^i(s)$ has been computed for each skin and spar panel, the minimum required skin thickness can be determined using the following expression:

$$t_{\text{skin,w,min}} = \frac{\max\left(|q^i(s)|\right)}{\tau_{\max}} \tag{3.7.24}$$

Here, $\tau_{\max}$ denotes the maximum allowable shear stress for the material, which includes a safety factor.

As in the case of the minimum stringer cross-sectional area, the use of a strict maximum function in Equation 3.7.24 introduces non-differentiability issues within the optimization process. To preserve continuity and ensure compatibility with gradient-based optimization, the maximum is approximated using the smooth formulation presented in Equation 3.7.17.

Finally, as previously mentioned, the structural wing model developed in this work is intended for analysis and optimization purposes. In the context of an optimization process, its role is to serve as a constraint mechanism that ensures structural feasibility. Specifically, the model verifies that the internal structural components, such as stringers, spars, and skin panels, meet strength requirements at each critical loading condition identified in the flight envelope.

The same methodology is applied to the fuselage. While the structural loads model for the fuselage is still under development, the structural analysis module is already operational. The fuselage can be viewed as a simplified wing structure, since it typically consists of a single-cell section.

To validate the model developed in this section, the rear fuselage test case presented by Noels [66] is reproduced. As RAPTOR does not yet include an internal computation of fuselage loads, the input data consist of the geometric parameters and structural loads directly taken from the reference.

The rear fuselage has a circular cross-section, defined by four stations (AA, BB, CC) with respective diameters of 1.28 m, 0.73 m, and 0.10 m. The overall geometry is illustrated in Figure 3.17.

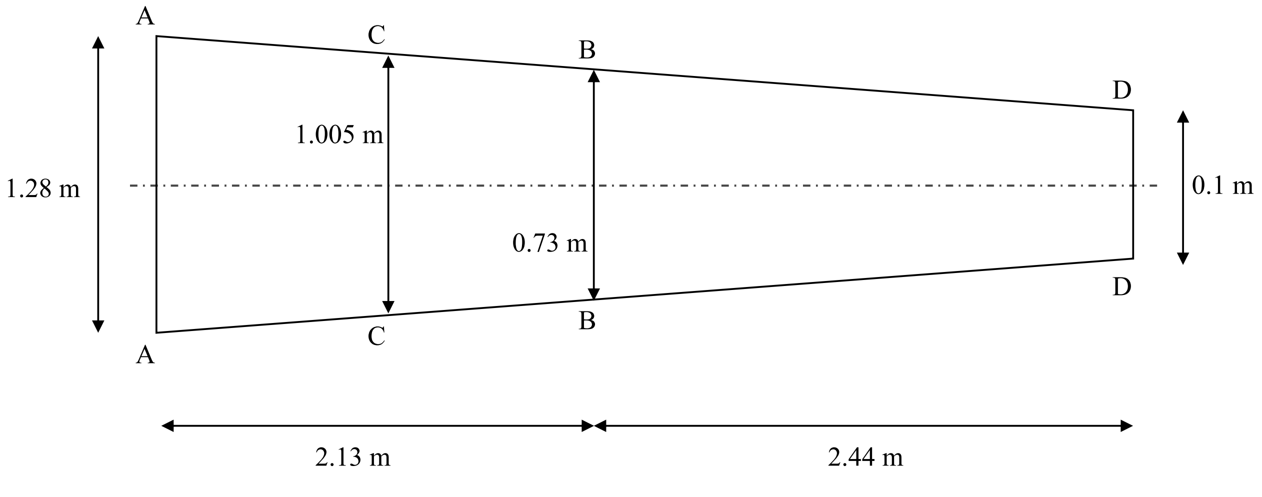


Figure 3.17: Rear fuselage geometry used for the structural model validation. Reproduced from Noels [66]. (not at scale).

The material used for the internal structure has allowable stress limits of $\sigma_{\max} = 155$ MPa for normal stress and $\tau_{\max} = 97$ MPa for shear stress. A representation of the cross-sectional layout is provided in Figure 3.18.

As previously described, the current implementation of the fuselage model in RAPTOR takes as input the structural loads for each flight condition and fuselage section. From these, the minimum required skin thickness is computed. For the validation case, the critical section for direct stress is

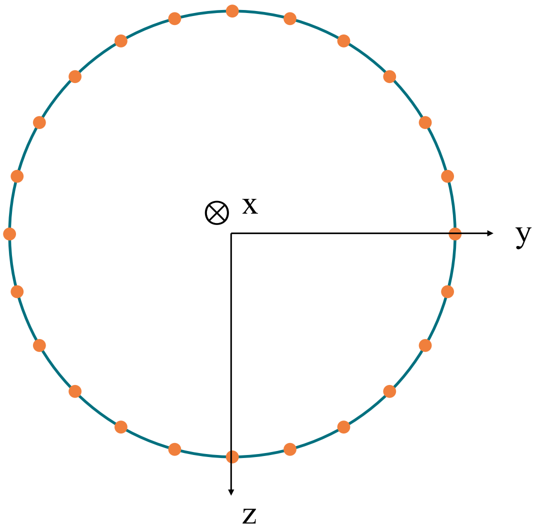


Figure 3.18: Cross-sectional view of the fuselage showing the stringers in orange, the skin panels in blue and the reference axis for the calculations. Adapted from Noels [66].

station AA, subjected to moments $M_y = 42387$ Nm and $M_z = 54680$ Nm. The most critical point for shear stress calculations (skin thickness) is station BB, with loads $M_x = -40533$ Nm, $T_y = -15000$ N, and $T_z = 12000$ N. The comparison between the values obtained in the reference and those predicted by RAPTOR is shown in Table3.16.

Quantity	**Reference**	**RAPTOR**
Minimum skin thickness (section BB)	0.67 mm	0.66 mm
Required stringers area (section AA)	57.7 mm^2	57.4 mm^2

Table 3.16: Comparison of structural model results between the reference case and the RAPTOR model.

As shown, the agreement between the two sets of results is very close, which validates the accuracy and reliability of the developed structural model.

Chapter 4

Applications and optimization results

4.1 Optimization of a tailless unmanned aerial vehicle

4.1.1 Problem definition

Tailless UAVs are a category of aircraft designed to minimize structural complexity by eliminating traditional tail components. This approach offers advantages such as reduced weight, improved flight efficiency, and lower manufacturing costs, making mass production more feasible. However, the design of tailless configurations involves a delicate balance between achieving high aerodynamic performance and maintaining sufficient static stability, which makes the use of computer-aided design tools essential. In this section, a reference tailless UAV optimization case presented by Kim et al. [37] is reproduced to demonstrate the capabilities of the developed optimization tool.

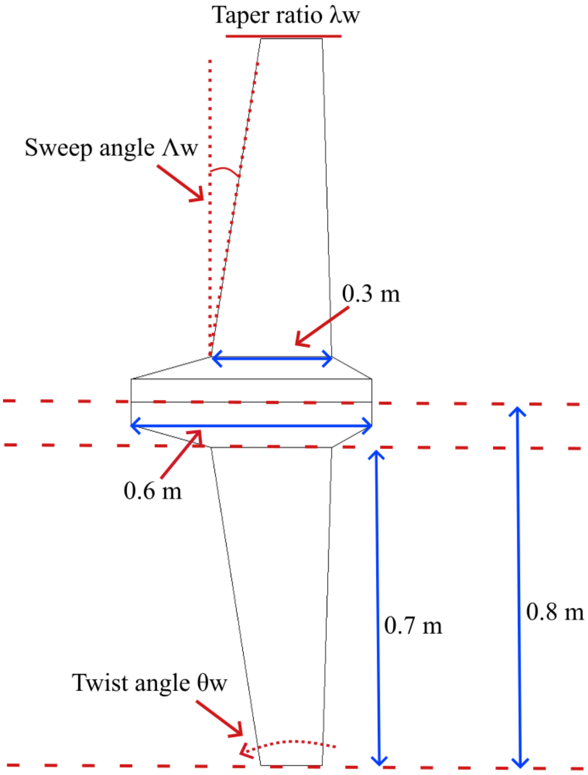


Figure 4.1: UAV geometry and fixed parameters, reproduced from Kim et al. [37].

The UAV is optimized for cruise conditions corresponding to a speed of 12 m/s, an angle of attack of 3 °, and an altitude below 100 meters. The fixed geometrical characteristics are shown in Figure 4.1.

An Eppler 328 airfoil is used at the wing root, transitioning to a NACA 0012 at the wing tip.

4.1.2 Optimization formulation

The optimization process focuses on the wing design, with the taper ratio (λ_{w}), sweep angle (Λ_{w}), and twist angle (θ_{w}) defined as the design variables, as detailed in Table 4.1.

Variable	**Lower bound**	**Upper bound**
Wing taper ratio λ_{w}	0.30	0.60
Wing sweep angle Λ_{w}	20°	60°
Wing twist angle θ_{w}	–10°	0°

Table 4.1: Design variables and bounds for the tailless UAV optimization problem.

The objective is to minimize the mass margin, expressed as $m_{\mathrm{margin}} = \dfrac{mg - L}{g}$, which effectively seeks to maximize lift relative to weight. The constraints imposed on the problem primarily ensure the aerodynamic stability of the UAV, including requirements on the static margin (SM), the pitch stability derivative (C_{m_α}), and the pitching moment coefficient (C_m). Table 4.2 summarizes the objective and the constraints that drive the process.

Quantity	**Relation**	**Target**
Objective		
Mass margin $m_{\mathrm{margin}} = \dfrac{mg - L}{g}$	→	Minimize
Constraints		
Pitching moment coefficient C_m	$\leq$	10^{-4}
Pitch stability derivative C_{m_α}	$<$	0
Static margin SM	$\geq$	0.05
Lift margin $mg - L$	$\leq$	0.0 N

Table 4.2: Objective and constraints for the tailless UAV optimization problem.

4.1.3 Modeling approach

Once the optimization procedure has been introduced and formulated, the next step is to define the analysis models. In some cases, entirely new models were developed to match with the application. These operations highlight the modularity and flexibility of the RAPTOR framework, which allows for efficient integration and customization of disciplinary models.

The process is organized into distinct but interconnected disciplines, as illustrated in Figure 4.2.

First, the weight module computes the total mass and center of gravity of the UAV. The aerodynamics module then determines the total lift generated. Finally, the stability module evaluates the pitching moment about the center of gravity of the UAV, its derivative with respect to the angle of attack, and the static margin.

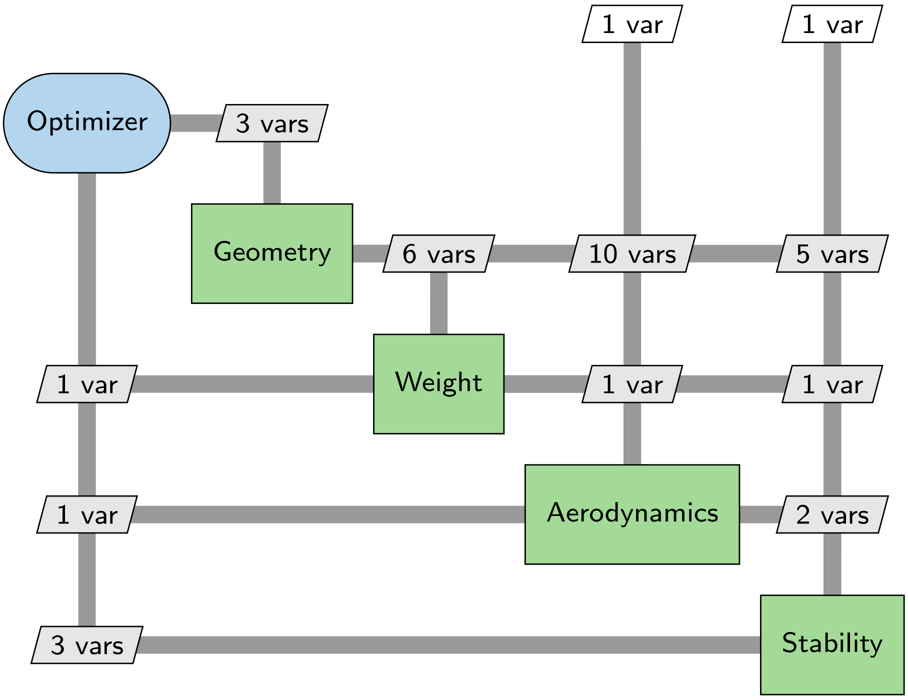


Figure 4.2: XDSM diagram representing the structure of the UAV optimization process.

The initial phase of this test case relies on semi-empirical methods implemented within RAPTOR.

In the reference, the tailless aircraft structural mass is calculated using Equation 3.2.10, where $\rho = 30$ kg/m^3 is the material density, Vol is the structural volume computed with OpenVSP, and ratio = 0.2 is the lightening ratio applied to account for structural weight reduction. In RAPTOR, the same expression (Equation 3.2.10) is used to estimate the structural mass. However, the volume is first approximated using geometric relations based on the airfoil surface areas and the main dimensions of the UAV.

Concerning the center of gravity, the benchmark study relies on a functionality of OpenVSP to estimate its position. In the developed tool, the CG is approximated by averaging the locations of two distinct parts: the main wing and the central section. The wing center of gravity is assumed to lie at 35% of its mean aerodynamic chord, while for the central part, it is taken at 40% of its maximum length.

To determine the overall mass and center of gravity position of the UAV, the characteristics of the onboard equipment summarized in Table 4.3 are incorporated. The total mass is obtained by adding the structural and equipment masses, and the CG is computed according to Equation 3.2.9.

In the aerodynamics module, the reference approach computes the cruise lift using a Vortex Lattice Method implemented in VSPAERO [69]. The lift curve slope, the pitching moment coefficient, and its derivative with respect to the angle of attack are also extracted from the simulation results. In the present method, the aerodynamic quantities are instead estimated using a semi-empirical model based on the formula provided by DATCOM [18], related in Section 3.3.1.

The final discipline addressed in this optimization case is the longitudinal stability. In the reference work, both the pitching moment coefficient around the center of gravity and its derivative with respect to the angle of attack are obtained from VLM simulations. In RAPTOR, the pitching moment coefficient is defined by the following relation:

Equipment	Mass (g)	x-position (m)	y-position (m)
Battery	200	0.35	0.00
Motor	50	0.50	0.00
ESC	20	0.45	0.00
Propeller	30	0.50	0.00
Servo 1	10	0.05	0.45
Servo 2	10	0.05	–0.45
Servo 3	10	0.05	0.73
Servo 4	10	0.05	–0.73
Receiver	10	0.20	0.00

Table 4.3: Mass and center of gravity location of the electrical equipment in the chordwise (x) and spanwise (y) directions, from Kim et al. [37].

$$C_m = C_{m,\mathrm{w}} + C_{m_\alpha}\,\alpha, \tag{4.1.1}$$

where $C_{m,\mathrm{w}}$ is the pitching moment coefficient of the main wing around its aerodynamic center, evaluated in the aerodynamics module. In Equation 4.1.1, C_{m_α} represents the variation of the pitching moment coefficient with respect to the angle of attack. It is computed using the developments presented by Raymer [72], adapted to the tailless aircraft:

$$C_{m_\alpha} = C_{L\alpha}\,\frac{x_{\mathrm{cg}} - x_{\mathrm{ac}}}{\bar{c}}, \tag{4.1.2}$$

where x_{cg} denotes the longitudinal position of the center of gravity, obtained from the weight module, x_{ac} is the aerodynamic center of the UAV, estimated by spanwise averaging the aerodynamic centers of the different sections, and $\bar{c}$ is the mean aerodynamic chord, determined by a similar averaging process.

From these quantities, in the Kim et al.'s study and in the present framework, the static margin is deduced:

$$\mathrm{SM} = -\frac{C_{m_\alpha}}{C_{L\alpha}}, \tag{4.1.3}$$

4.1.4 Optimization results and analysis

Once the optimization process is formulated and the analysis models are chosen, the optimization procedure can be launched. Table 4.4 summarizes the outcomes of the optimization process, both for the present study and the reference one.

A first observation is the strong similarity between the optimal values of the design variables obtained with RAPTOR and those from the literature results. In both cases, the taper ratio significantly increases, moving from the lower bound to the upper range of the design space. Regarding the sweep angle, the difference between the two optima is remarkably small, with a relative deviation of approximately 0.35%, which is impressive considering the differences between the models used. For the twist angle, a discrepancy can be observed between the two results, although it remains relatively limited.

Metric	Baseline RAPTOR	Optimum RAPTOR	Optimum Reference
Design variables			
Wing taper ratio λ_w	0.30	0.60	0.51
Wing sweep angle Λ_w	45.0°	28.90°	29.0°
Wing twist angle θ_w	–4.2°	–3.78°	–4.6°
Constraints			
$mg - L$	–1.46 N	–2.37 N	–2.06 N
Pitch stability derivative C_{m_α}	–1.121	–0.224	–0.198
Static margin SM	0.239	0.050	0.050
Pitching moment coefficient C_m	–0.027	0.0001	-2.66×10^{-6}
Objective			
Mass margin m_margin	–0.149 kg	–0.242 kg	–0.210 kg
Other outputs			
UAV surface area S	0.378 m^2	0.441 m^2	0.422 m^2
Lift coefficient C_L	0.246	0.235	0.224

Table 4.4: Comparison of UAV optimization results obtain with RAPTOR the in Kim et al.[37] reference study.

Concerning the constraints, the forces margin (mg - L) condition is satisfied both at the initial and final stages of the optimization. The aim is to achieve a negative value, which becomes increasingly negative as the optimization progresses. Comparing the final results between the developed tool and the reference work, the mass margin values are relatively close.

The evolution of the pitching moment derivative with respect to the angle of attack shows a notable increase over the course of the optimization. Within the semi-empirical model used in RAPTOR, and according to Equation 4.1.2, this trend is primarily due to the reduction in the distance between the center of gravity and the overall aerodynamic center. When comparing the optimum values obtained in the present and in the benchmark study, the results remain fairly close, with a relative difference of approximately 11.6%.

Initially, the static margin is significantly higher than the minimum constraint threshold. In both the in-house optimization framework and the litterature study, this margin progressively decreases throughout the optimization, eventually reaching the lowest admissible value. This behavior is consistent with the evolution of the pitching moment derivative, which shows a substantial reduction in absolute value.

Finally, regarding the pitching moment coefficient, the constraint imposes a value lower than or equal to 0.0001. During the optimization, this coefficient progressively increases. As described by Equation 4.1.1, this trend is linked to the rise in C_{m_α} previously discussed. In RAPTOR, the optimization converges precisely to the limit value, while in the reference study, the final result is slightly lower.

The objective in this case is to minimize the mass margin previously defined. It can first be observed that this goal is achieved when moving from the baseline to the optimized design. When comparing the final results with the litterature, the values remain fairly close, with a relative error of approximately 13.3%. Considering the difference in model fidelity, this level of agreement can be regarded as acceptable.

Regarding the wing surface area, a slightly lower value is obtained in the reference work. This difference can be attributed to the higher taper ratio in the RAPTOR optimum, which leads to an increased surface area.

At the level of the lift coefficient, a relative difference of 4.68 % is observed at the optimum, which remains well within acceptable limits.

Now that the optimum configuration obtained with RAPTOR has been compared to the reference study, attention can be turned to the evolution of the objective function and design variables throughout the optimization process.

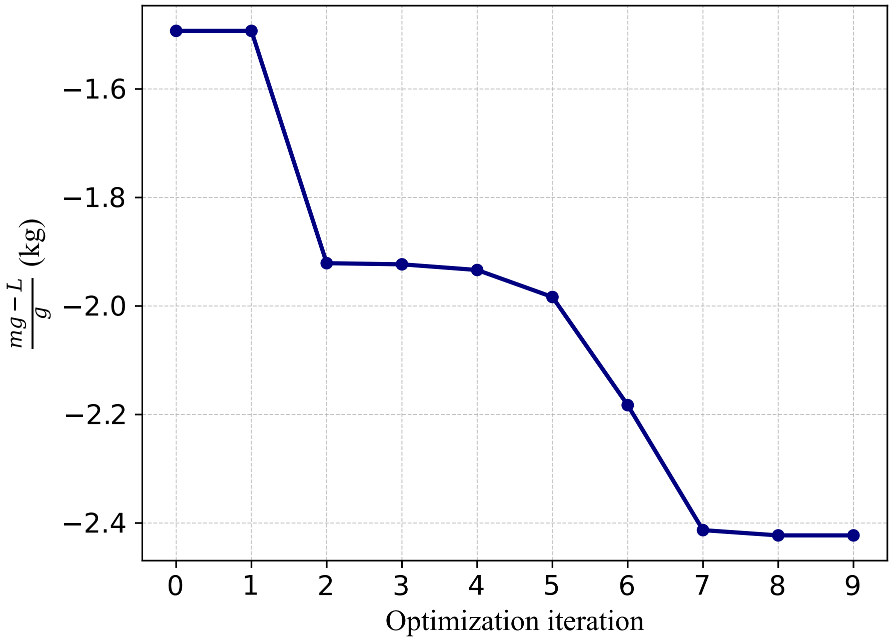


Figure 4.3: Evolution of the objective function with respect to the optimization iterations.

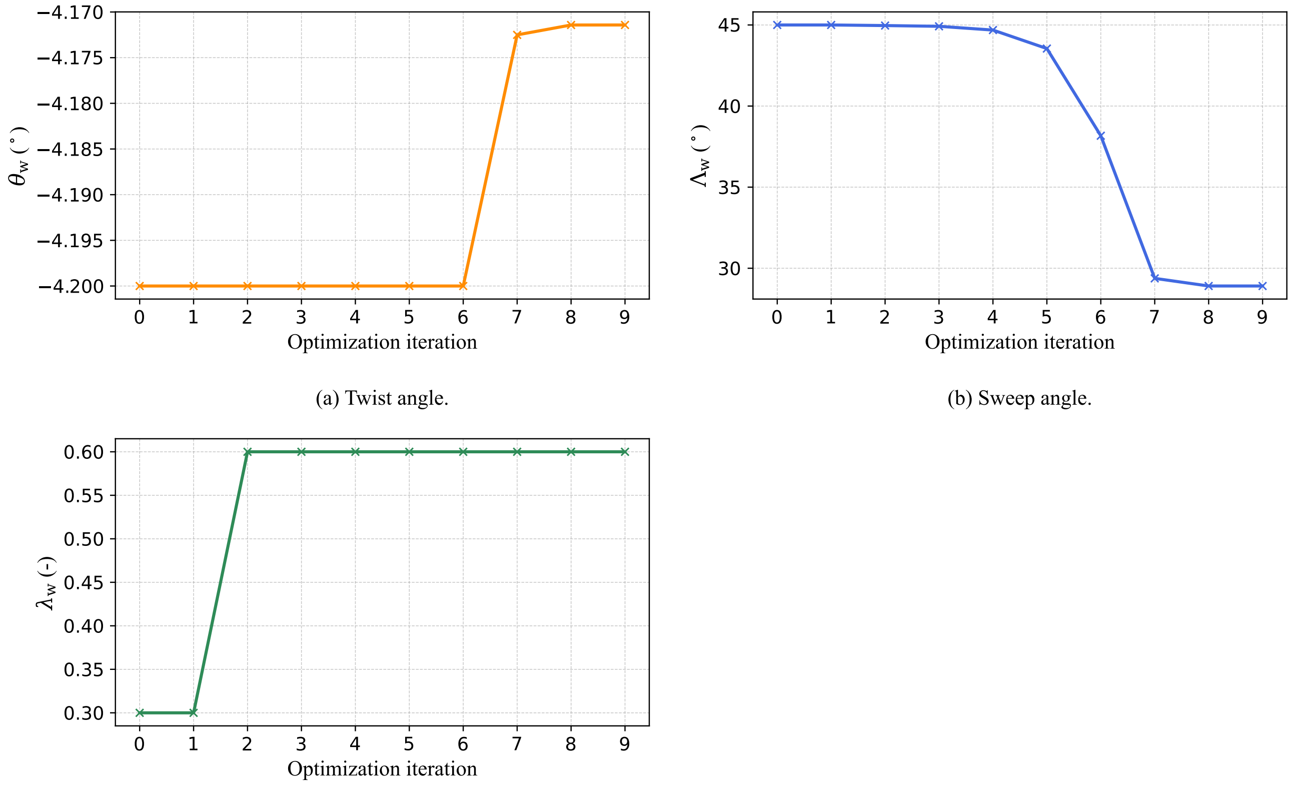
Figure 4.3 shows the evolution of the objective function over the course of the optimization iterations.

A first observation, when compared to the reference study, is the significantly lower number of iterations required to reach convergence: while the literature case involved approximately 100 iterations, the present study converged in just 10. This difference can be partly explained by the relative complexity of the models used in each study, but also, by the algorithm selected for the optimization. In RAPTOR, a gradient-based approach is adopted, with automatic differentiation, which offers both high accuracy and computational efficiency. In contrast, the tool used in the study by Kim et al. [38] relies on a differential evolution method, which is a gradient-free optimization algorithm.

Regarding the curve pattern, a sharp decrease occurs between the second and third iterations, followed by a plateau, and then a gradual decline until convergence to the optimum. This behavior is driven by the evolution of the design variables over the course of the optimization.

The evolution of the taper ratio during the optimization is shown in Figure 4.4(c). Between the first and second iterations, its value abruptly shifts from the lower bound to the upper limit of the design space. When comparing with the evolution of the objective function (Figure 4.3) and the other design variables (Figures 4.4(a) and 4.4(b)), it becomes evident that the drop in the objective function at the beginning of the process is primarily due to this sudden change in taper ratio.

From a physical standpoint, increasing the taper ratio leads to a larger wing surface area, which lowers the lift curve slope and thus the lift coefficient. In addition, the increased volume results in a higher structural mass. Nevertheless, the lift force grows proportionally with the wing surface, and the optimization results indicate that the gain in lift due to the surface increase outweighs the negative effects of the mass increase and the slight reduction in lift coefficient. As the objective is to minimize mg



(a) Twist angle.

(b) Sweep angle.

(c) Taper ratio.

Figure 4.4: Evolution of the taper ratio over the optimization iterations.

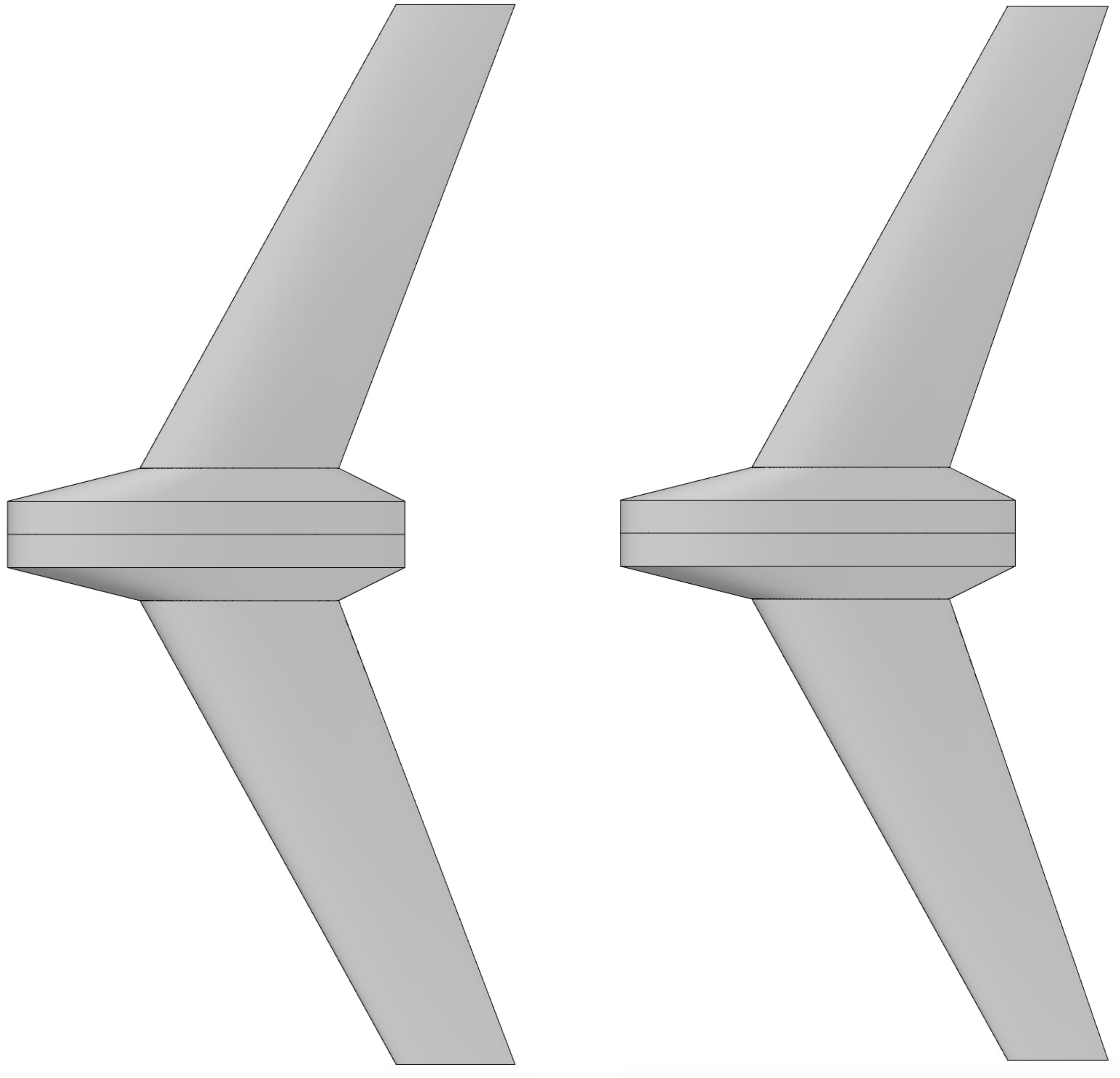
- L, maximizing the surface area through a higher taper ratio proves beneficial, since the lift increases more importantly than the associated mass growth.

The evolution of the sweep angle throughout the optimization process is illustrated in Figure 4.4(b). The sweep remains relatively constant up to iteration 6, after which it decreases over two iterations before stabilizing. When comparing with the behavior of the objective function (Figure 4.3), it becomes clear that this reduction in sweep angle coincides with the second significant drop in the objective. This correlation can be confirmed by examining the taper ratio evolution (Figure 4.4(c)), which remains almost unchanged over the same interval. Moreover, as will be discussed later, the twist angle has no influence on the objective function within the semi-empirical model. These observations confirm that the second major improvement in the objective function is driven by the reduction in sweep angle.

From a physical perspective, this behavior is consistent with the objective of minimizing mg - L. Reducing the sweep angle for a given span and fixed chord lengths results in a lower structural volume, thereby decreasing the mass.

The evolution of the twist angle throughout the optimization is presented in Figure 4.4(a). At the beginning of the process, it remains constant, before a significant increase between iterations 7 and 9, followed by a plateau until the end of the process.

Within the semi-empirical models, the twist angle has no impact on the estimated mass or the lift force, and therefore does not directly influence the objective function. Instead, it only affects the wing pitching moment about its aerodynamic center, $C_{m,w}$, as defined in Equation 3.3.6. Considering the constraints listed in Table 4.2, the variation of the twist angle ensures that the pitching moment coefficient remains within its prescribed limits, as it is the only constraint on which it can have a mathematical effect. As shown in Table 4.4, the value of C_{m_α} increases significantly over the course of the optimization. Since the cruise angle of attack remains constant, regarding Equation 4.1.1, the only quantity that can be adjusted to ensure compliance with the pitching moment constraint is the main wing pitching moment about its aerodynamic center, $C_{m,w}$. Additionally, from Equation 3.3.6, one can deduce that the more the twist is negative, the higher the value of the wing pitching moment. Consequently, a less negative twist decreases the pitching moment of the wing, and therefore of the entire aircraft with respect to its center of gravity. For these reasons, during the optimization, the twist angle must increase from $-4.2°$ to approximately $-3.8°$ to satisfy the pitching moment constraint at the optimum.



(a) RAPTOR.

(b) Reference.

Figure 4.5: Top view comparison of the optimized UAV configurations obtained with RAPTOR and in Kim et al.[37].

The optimized configurations of the UAV in RAPTOR and the reference study are represented in Figure 4.5.

4.1.5 Design space exploration

To complete the analysis of this optimization process, it is necessary to explore the design space. For this purpose, the initial values of the three design variables are modified, and the resulting evolution of both the variables and the objective function throughout the optimization is evaluated.

	Baseline $\lambda_w = 0.3$	Baseline $\lambda_w = 0.4$	Baseline $\lambda_w = 0.5$
Baseline design variables			
Wing taper ratio λ_w	0.30	0.40	0.50
Wing sweep angle Λ_w	45.0°	45.0°	45.0°
Wing twist angle θ_w	–4.2°	–4.2°	–4.2°
Baseline objective			
Mass margin m_{margin}	–0.149 kg	–0.168 kg	–0.183 kg
Optimum design variables			
Wing taper ratio λ_w	0.60	0.60	0.60
Wing sweep angle Λ_w	28.90°	28.90°	28.90°
Wing twist angle θ_w	–3.784°	–3.784°	–3.784°
Optimum objective			
Mass margin m_{margin}	–0.242 kg	–0.242 kg	–0.242 kg

Table 4.5: Impact of initial wing taper ratio on optimization results using RAPTOR.

Table 4.5 summarizes the results obtained by varying the initial taper ratio of the main wing, while keeping the baseline sweep and twist angles constant. As expected, the only difference between the baseline configurations lies in the initial value of the objective function, which directly reflects the change in taper ratio.

It is also observed that, regardless of the initial taper, the optimization systematically converges to the same optimum for the sweep angle, twist angle, and taper ratio, leading to an identical final objective value. These results confirm the previous analysis, which indicated that increasing the taper ratio consistently reduces the objective function when using semi-empirical models. Consequently, in terms of taper ratio, the optimum is always reached at the upper bound of the design space.

	Baseline Λ_w = 10°	Baseline Λ_w = 45°	Baseline Λ_w = 60°
Baseline design variables			
Wing taper ratio λ_w	0.30	0.30	0.30
Wing sweep angle Λ_w	10.0°	45.0°	60.0°
Wing twist angle θ_w	–4.2°	–4.2°	–4.2°
Baseline objective			
Mass margin m_{margin}	–0.208 kg	–0.149 kg	–0.062 kg
Optimum design variables			
Wing taper ratio λ_w	0.60	0.60	0.60
Wing sweep angle Λ_w	28.90°	28.90°	28.90°
Wing twist angle θ_w	–3.784°	–3.784°	–3.784°
Optimum objective			
Mass margin m_{margin}	–0.242 kg	–0.242 kg	–0.242 kg

Table 4.6: Impact of initial wing sweep angle on optimization results using RAPTOR.

Table 4.6 summarizes the results obtained by varying the initial sweep angle. Similarly to the taper ratio exploration, only the baseline value of the objective function is affected by changes in the initial sweep. This observation reinforces the conclusion that the final optimum represents the true global best within the entire design space, rather than a simple local minimum. Indeed, by exploring the design space across the two variables that influence the objective function, the optimization consistently converges to the same solution, confirming the robustness of the result.

	Baseline $\theta_w = -2.0°$	Baseline $\theta_w = -4.2°$	Baseline $\theta_w = -9.0°$
Baseline design variables			
Wing taper ratio λ_w	0.30	0.30	0.30
Wing sweep angle Λ_w	45.0°	45.0°	45.0°
Wing twist angle θ_w	–2.0°	–4.2°	–9.0°
Baseline objective			
Mass margin m_{margin}	–0.149 kg	–0.149 kg	–0.149 kg
Optimum design variables			
Wing taper ratio λ_w	0.60	0.60	0.60
Wing sweep angle Λ_w	28.90°	28.90°	28.90°
Wing twist angle θ_w	–2.0°	–3.784°	–3.784°
Optimum objective			
Mass margin m_{margin}	–0.242 kg	–0.242 kg	–0.242 kg

Table 4.7: Impact of initial wing twist angle on optimization results using RAPTOR.

Table 4.7 presents the results obtained by varying the initial twist angle. In all three configurations, the optimization converges to the same final objective value, even though the twist angle does not necessarily reach the same value at the end of the process. This confirms that the twist angle has no direct influence on the objective function when using semi-empirical models.

At the optimum, the final twist angle remains equal to its initial value when starting at −2.000°, while for the two other cases, it converges to −3.784°. As previously explained, this adjustment is related to the pitching moment constraint. When the initial twist is lower than −3.784°, it must increase during the optimization to satisfy the constraint. Conversely, if the initial value is already higher, no modification is needed as the constraint is respected throughout the process.

This design space exploration further confirms that the results obtained correspond to the global optimum within the design space. In addition, the strong agreement with the reference case validates the optimization process implemented in RAPTOR. The rapid convergence also highlights the reliability of the architectural and algorithmic choices made in the development of the framework.

The next step will be to reproduce this benchmark test using higher-fidelity models.

4.1.6 Optimization using higher fidelity models

To demonstrate the modularity of RAPTOR and its ability to accommodate multi-fidelity optimizations, the UAV test case described previously is reproduced using higher-fidelity models.

For the geometry input, the UAV configuration is built using the model based on OpenVSP [69]. Then, the structural mass is computed using Equation 3.2.10, where the volume estimation is performed by the coupling with OpenVSP. For the aerodynamic and stability evaluations, the semi-empirical models are replaced by the Source-Doublet Panel Method (SDPM [10]) model, described in Sections 3.3.2 and 3.4.2. These models are easily adapted to the application, illustrating the flexibility of RAPTOR.

	Baseline RAPTOR	**Optimum RAPTOR**	**Optimum Reference**
Design variables			
Wing taper ratio λ_w	0.30	0.60	0.51
Wing sweep angle Λ_w	45.0°	29.6°	29.0°
Wing twist angle θ_w	-4.2°	0.0°	–4.6°
Constraints			
$mg - L$	–2.943 N	–5.944 N	–2.06 N
Pitch stability derivative C_{m_α}	-1.002	–0.205	–0.198
Static margin SM	0.26	0.050	0.050
Pitching moment coefficient C_m	-0.0479	–0.0155	–0.00000266
Objective			
Mass margin m_margin	–0.30 kg	–0.61 kg	–0.21 kg
Other outputs			
UAV surface area S	0.378 m^2	0.441 m^2	0.422 m^2
Lift coefficient C_L	0.268	0.319	0.224

Table 4.8: Comparison of the UAV optimization results between RAPTOR (higher-fidelity models) and Kim et al.[37].

The results of the optimization process are presented in Table 4.8.

When compared to the reference solution, the objective function is lower with RAPTOR, indicating that the tool successfully identified a more favorable configuration under the given constraints. Moreover, at the end of the process, all the specified constraints are respected.

Regarding the design variables, the optimized sweep angle and taper ratio obtained with RAPTOR are very close to those found in the reference study. However, for the twist angle, a significant difference is observed: while the reference study yields a twist near the center of the design range, RAPTOR converges to a solution with zero twist.

The differences both in terms of objective function and final twist angle are highly correlated, as the reference configuration converges toward a negative wing twist. This reduces the lift generated by the main wing, leading to a less favorable value of the objective.

In terms of constraints, the static margin at the optimum matches exactly between the two studies, and the value of C_{m_α} is nearly identical. On the other hand, a difference is observed for the moment coefficient constraint.

As discussed above, the two optimal configurations obtained by RAPTOR and the reference study present some differences. To investigate the origin of these discrepancies, the optimum of Kim et al. [37] is analyzed using the RAPTOR models. The main results of this analysis are summarized in Table 4.9.

	C_m	C_{m_α}	C_L	SM	Margin Objective
RAPTOR	0.0022	-0.220	0.247	0.05	-0.29
Reference	$-0.266 \cdot 10^{-5}$	-0.198	0.224	0.05	-0.21

Table 4.9: Comparison of the analysis of the optimum of Kim et al. [37] study, using RAPTOR models.

In terms of lift coefficient, and consequently the margin objective, only minor differences are observed between the two approaches. Similarly, the values of the static margin and the derivative of the moment coefficient with respect to the angle of attack C_{m_α} are nearly identical. However, the main discrepancy lies in the moment coefficient C_m. This deviation has a direct impact on the optimization outcome, as when considering the optimum configuration of the reference study in RAPTOR, the pitching moment constraint is not respected.

In summary, the reproduction of the benchmark test using higher-fidelity models leads to a part of the results that closely align with those of the reference study. Nevertheless, a difference is observed in the final optimal configurations, primarily attributed to discrepancies in the predicted pitching moment coefficient. These variations can be explained by the use of different aerodynamic solvers, namely SDPM in RAPTOR and VSPAERO in the reference study.

4.1.7 Conclusion

This section focused on replicating the optimization of a tailless UAV. The first step involved selecting suitable analysis models, adapting or developing new ones when necessary. Thanks to the modular and extensible architecture of RAPTOR, these modifications were implemented with relative ease.

The optimization process was then executed, and the results were compared to those of the reference study. The strong agreement between the two sets of results confirms the validity and reliability of the optimization approach implemented in RAPTOR.

Subsequently, the evolution of both the objective function and the design variables was analyzed throughout the optimization iterations, providing insight into the convergence behavior and solution characteristics.

To further assess the robustness of the framework, additional optimization runs were conducted from varying initial conditions. The consistent convergence toward the same optimum demonstrates the stability and effectiveness of the process, suggesting that the identified solution represents a global optimum within the defined design space.

Finally, higher-fidelity models were integrated into the framework to evaluate aerodynamic forces, stability characteristics, and weight estimations more accurately. Based on it, the optimization of the UAV was replicated, demonstrating the multi fidelity options available in RAPTOR.

4.2 Optimization of a single-seat aerobatic aircraft

4.2.1 Problem definition

This section presents the analysis and optimization of a single-seat aerobatic aircraft configuration, aiming to reproduce the case study conducted by Thu et al. [85], itself based on a classical exercise proposed by Raymer [72].

The primary objective of this optimization task is to reduce the aircraft's drag coefficient during cruise conditions, thereby enhancing its aerodynamic efficiency. The mission profile considered includes a standard flight sequence: take-off, climb to a cruising altitude of 8000 ft, steady cruise, descent, and landing. This flight segmentation is illustrated in Figure 4.6.

The aircraft is based on a conventional low-wing, unswept configuration. The initial geometric parameters for the wing include an aspect ratio of 6 and a taper ratio of 0.4. The selected airfoils are NACA 63-2012 at the wing root and NACA 63-2015 at the tip. Both the horizontal and vertical stabilizers are initially modeled with aspect ratios of 4, taper ratios of 0.4, and employ the symmetric NACA 0012 airfoil.

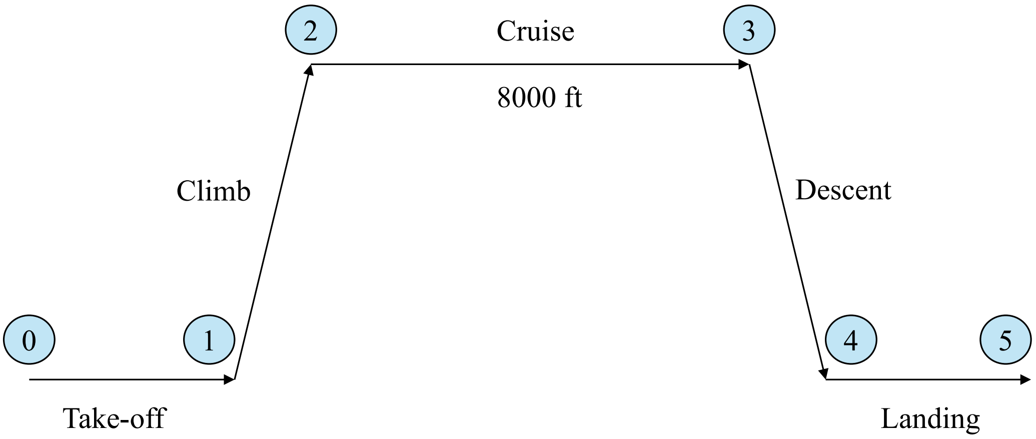


Figure 4.6: Typical mission profile of the optimized aerobatic aircraft. Reproduced from Thu et al. [85].

The propulsion system is defined by the LYCOMING O-320-A2B engine, a widely used 150 HP powerplant in light aircraft. It operates at a nominal rotational speed of 2700 RPM and has a specific fuel consumption at cruise of $C_{\text{bhp}} = 0.5$ lb/hp·hr.

4.2.2 Optimization formulation

The main objective is to minimize the aircraft's total drag coefficient during cruise conditions. To achieve this, several geometric parameters of the aircraft are allowed to vary during the optimization process. These include key characteristics of the wing, as well as those of the horizontal and vertical tail surfaces. The optimization problem is also subject to constraints. Regarding the aircrfat performances, constraints include the required cruise range, minimum stall speed, maximum takeoff distance, and minimum rate of climb. In addition, the design has to guarantee longitudinal and lateral stability.

The design variables and associated bounds are summarized in Table 4.10, and the constraints are listed in Table 4.11.

Design variable	**Lower bound**	**Upper bound**
Wing root chord $c_{\text{r,w}}$ (ft)	2.40	6.25
Wing tip chord $c_{\text{t,w}}$ (ft)	2.40	3.19
Wing span b_{w} (ft)	23.00	30.80
Vertical tail root chord $c_{\text{r,vt}}$ (ft)	1.60	4.92
Vertical tail tip chord $c_{\text{t,vt}}$ (ft)	1.10	2.45
Vertical tail span b_{vt} (ft)	3.20	4.90
Horizontal tail root chord $c_{\text{r,ht}}$ (ft)	1.32	3.60
Horizontal tail tip chord $c_{\text{t,ht}}$ (ft)	0.88	1.58
Horizontal tail span b_{ht} (ft)	8.79	10.10
Wing loading W/S_{w} (lb/ft^2)	8.5	23.0

Table 4.10: Design variables and bounds for the aerobatic aircraft optimization problem.

4.2.3 Modeling approach

In terms of weight estimation, the same empirical formulas as the model of the reference, which are proposed in Raymer [72] are adopted. The extensibility of the RAPTOR framework allows to develop and include these models very easily.

Constraint	Relation	Target value
Cruise range	≥	280 nm
Stall speed	≤	84.39 ft/s
Takeoff distance	≤	1000 ft
Rate of climb	≥	1500 ft/min
Pitch moment slope C_{m_α}	≥	–1.50
Pitch moment slope C_{m_α}	≤	–0.16
Yaw moment slope C_{n_β}	≥	0.05
Yaw moment slope C_{n_β}	≤	0.10

Table 4.11: Optimization constraints applied to the aerobatic aircraft configuration.

For the drag prediction, both the present study and the reference rely on the Raymer book [72] methodology. Notably, the reference study estimates the lift coefficient by directly balancing lift and weight during cruise, and this approach is also selected in the present work.

The stability modeling approaches are slightly different between studies. Static stability is computed using Raymer [72] formulations in the reference. Concerning RAPTOR, elements from Roskam [75], in addition to Raymer's approximations are involved. These models are described in Section 3.4.

The performance and propulsion aspects, including cruise range estimation, rate of climb, takeoff distance, stall velocity and propeller thrust, are all modeled using classical empirical equations from Raymer developed in Section 3.6, similarly to the original study.

Regarding the optimization strategy, the reference study also adopts a gradient-based approach, specifically employing Sequential Quadratic Programming (SQP) methods. However, while Thu et al. [85] do not detail the method used to compute the derivatives.

With the overall setup now established, the next step is to perform the optimization in order to obtain an enhanced aircraft configuration.

4.2.4 Optimization results and analysis

In this section, the analysis results obtained in the present study are first compared to those of the reference case. Following this comparison, the optimization process is carried out and discussed. The process is conducted under cruise conditions, defined by an altitude of 8000 ft and a Mach number of 0.15.

The analysis results are presented in Table 4.12. The comparison reveals a strong agreement between the two sets of results, indicating that the models and methods implemented in RAPTOR reproduce the reference behavior for the analysis part. Small differences are observed in terms of stability derivatives, which should be partially due to the differences in the models used.

With the analysis results now established, attention is turned to the optimization outcomes. Table 4.13 summarizes the results obtained from both frameworks.

One of the first inconsistencies that becomes apparent when analyzing the reference study lies in the definition of the upper bounds for the wing chord lengths. In the reference, the root and tip chords of the wing are limited to maximum values of 6.25 ft and 2.53 ft, respectively. While the authors provide the optimal wing span value (26.82 ft), they do not report the corresponding optimal chord lengths. Instead, they give the optimal value of the wing surface area, which is 141.18 ft^2. Nevertheless, using the maximum allowable chord values from the reference (6.25 ft and 2.53 ft), the largest wing area that

Metric	RAPTOR	Thu et al.
Design variables		
Wing root chord $c_{\mathrm{r,w}}$	6.25 ft	6.25 ft
Wing tip chord $c_{\mathrm{t,w}}$	2.53 ft	2.53 ft
Vertical tail root chord $c_{\mathrm{r,vt}}$	4.00 ft	4.00 ft
Vertical tail tip chord $c_{\mathrm{t,vt}}$	1.60 ft	1.60 ft
Horizontal tail root chord $C_{\mathrm{r,ht}}$	3.60 ft	3.60 ft
Horizontal tail tip chord $C_{\mathrm{t,ht}}$	1.40 ft	1.40 ft
Wing span b_{w}	26.60 ft	26.60 ft
Horizontal tail span b_{ht}	10.10 ft	10.10 ft
Vertical tail span b_{vt}	4.10 ft	4.10 ft
Constraints		
Cruise range	210.90 nm	207.90 nm
Rate of climb	1591.64 ft/min	1500.00 ft/min
Takeoff distance	1056.0 ft	1000.0 ft
Stall speed	84.88 ft/s	84.39 ft/s
Pitch moment slope C_{m_α}	–0.7595 1/rad	–0.58 1/rad
Yaw moment slope C_{n_β}	0.0478 1/rad	0.0717 1/rad
Other results		
Empty weight	881.0 lb	883.0 lb
Wing surface area S_{w}	116.74 ft^2	116.74 ft^2
Wing taper ratio	0.40	0.40
Objective		
Drag coefficient (cruise) C_D	0.0335	0.0336

Table 4.12: Analysis results before optimization: comparison between the values obtained with the RAPTOR framework and those reported by Thu et al. [85].

can be achieved within the specified design space is:

$$S_{\mathrm{w,max}} = \frac{b_{\mathrm{w}}}{2}\left(c_{\mathrm{r,w,max}} + c_{\mathrm{t,w,max}}\right) = \frac{26.82}{2}(6.25 + 2.53) = 117.74\ \mathrm{ft}^2 \tag{4.2.1}$$

This contradiction suggests either an inconsistency in the published bounds or an omission in reporting the actual chord values at the optimum. To ensure a fair and meaningful comparison, the present study has therefore chosen to relax the chord bounds to 8.0 ft at the wing root, and 4.0 ft at the tip, enabling the optimizer to reach a wing area consistent with that of the reference.

Once the inconsistency regarding the wing chord bounds is addressed, the comparison of the two optimal configurations can proceed.

Although the optimal root and tip chord lengths of the wing are not explicitly reported in the reference study, the authors do provide the corresponding wing taper ratio, which is 0.5102. This value is very close to the taper ratio obtained in the present study, which is 0.496, indicating a high level of agreement on this geometric parameter.

The resulting wing planform surface areas are also very similar. In RAPTOR, this similarity is partially attributed to the relaxed bounds on the chord lengths, which were necessary to ensure consistency with the surface area reported in the reference.

Metric	**Baseline RAPTOR**	**Optimum RAPTOR**	**Optimum Thu et al.**
Design variables			
Wing root chord $c_{\mathrm{r,w}}$	6.25 ft	8.00 ft	–
Wing tip chord $c_{\mathrm{t,w}}$	2.53 ft	3.97 ft	–
Vertical tail root chord $c_{\mathrm{r,vt}}$	4.00 ft	1.99 ft	1.72 ft
Vertical tail tip chord $c_{\mathrm{t,vt}}$	1.60 ft	1.10 ft	1.13 ft
Horizontal tail root chord $C_{\mathrm{r,ht}}$	3.60 ft	1.32 ft	1.32 ft
Horizontal tail tip chord $C_{\mathrm{t,ht}}$	1.40 ft	0.88 ft	0.88 ft
Wing span b_{w}	26.60 ft	23.54 ft	26.82 ft
Horizontal tail span b_{ht}	10.10 ft	8.79 ft	8.79 ft
Vertical tail span b_{vt}	4.10 ft	4.90 ft	4.80 ft
Constraints			
Cruise range	210.90 nm	228.0 nm	228.0 nm
Rate of climb	1591.64 ft/min	1528.01 ft/min	2487.19 ft/min
Takeoff distance	1056.0 ft	857.8 ft	500.0 ft
Stall speed	84.88 ft/s	77.30 ft/s	66.22 ft/s
Pitch moment slope C_{m_α}	–0.7595 1/rad	–0.2566 1/rad	–0.3786 1/rad
Yaw moment slope C_{n_β}	0.0478 1/rad	0.0500 1/rad	0.0600 1/rad
Other results			
Empty weight	881.0 lb	841.7 lb	843.4 lb
Wing surface area S_{w}	116.74 ft^2	140.83 ft^2	141.18 ft^2
Wing taper ratio	0.40	0.496	0.5102
Objective			
Drag coefficient (cruise) C_D	0.0335	0.02998	0.02757

Table 4.13: Optimization results: comparison between the initial RAPTOR analysis, the optimum configuration from RAPTOR, and the results from Thu et al. [85].

For the vertical tail geometry, both the root and tip chord lengths converge to very similar values in the two studies. While these variables are initially positioned near the upper bound of the design space in the baseline configuration, they shift toward the lower part of the allowable range by the end of the optimization process in both the reference and the present study. For the vertical tail span, the initial value of 4.10 ft, located near the center of the design space, is increased during the optimization. It reaches 4.80 ft in the reference study and 4.90 ft in RAPTOR, positioning it near the upper bound in both cases. This again reflects a shared trajectory in design space exploration.

A similar pattern is observed in the horizontal tail chord lengths. Both the root and tip chords exhibit identical values at the optimum in RAPTOR and in the reference. Once again, these parameters move from the upper region of the design space in the baseline to the lower bound after optimization, reinforcing the trend of minimizing surface dimensions. Regarding the horizontal tail span, the initial value is 10.10 ft in both cases. At the optimum, this value decreases to 8.79 ft in both studies, indicating perfect agreement.

Finally, the wing span exhibits different trends between the two studies. Starting from an initial value of 26.6 ft, RAPTOR reduces the span to 23.54 ft, very close to the lower bound of 23 ft, whereas the reference study increases it slightly to 26.82 ft. This suggests that while in RAPTOR, the wing span tends to converge to the lower part of the design space, the reference study optimum remains closer to the center of the allowable range.

Overall, these comparisons confirm that despite slight differences, the two studies converge on remarkably similar configurations for most design variables.

By examining the wing span and chord lengths at the beginning and end of the optimization process, it is possible to deduce the evolution of the wing aspect ratio in both studies. In the baseline configuration, the aspect ratio is approximately 6 in both RAPTOR and the reference study. After optimization, this value decreases to 3.93 in the present work and to 5.09 in the reference. In a first time, this trend may seem counterintuitive. When the objective is to minimize the drag coefficient, one might expect the process to converge toward an increase of the aspect ratio. In that sense, the induced drag coefficient would be decreased.

An increase in aspect ratio would mean an increase of the wing span, with respect to the chord lengths. In this case, the opposite direction is followed

To better understand this behavior, a series of optimizations were performed, and constraints were progressively relaxed one at a time. It was found that the wing span only reached its upper bound when the cruise range constraint was removed or made less restrictive. As the constraint on minimum cruise range was progressively relaxed, the optimized wing span increased accordingly. This observation clearly suggests that the cruise range constraint plays a critical role in limiting the span.

To explain this phenomenon, recall the Breguet range equation for propeller-driven aircraft (Equation 3.6.1), which shows that the cruise range is directly proportional to the lift-to-drag ratio. In cruise conditions, lift equals weight. Moreover, developing the wing surface formula, one has $S_\mathrm{w} = \frac{b_\mathrm{w}(c_{\mathrm{r,w}}+c_{\mathrm{t,w}})}{2}$. Injecting these in the lift-to-drag ratio, and simplifying the relation, one obtains:

$$\left(\frac{L}{D}\right)_{\textbf{cruise}} = \frac{1}{\dfrac{q\,C_{D_0}\left(c_{\mathrm{r,w}} + c_{\mathrm{t,w}}\right) b_\mathrm{w}}{2W} + \dfrac{W}{q\,e_\mathrm{w}\,\pi b_\mathrm{w}^2}} \tag{4.2.2}$$

Where q is the dynamic pressure, C_{D_0} is the total zero-lift drag coefficient, and W is the aircraft weight. This expression highlights two opposing effects in the denominator. The first term, associated with zero-lift drag, increases linearly with the wing span b_w, while the second term, related to induced drag, decreases with the square of b_w. In the present case, as the span increases, the rise in zero-lift drag dominates the reduction in induced drag. This leads to an overall decrease in the lift-to-drag ratio, and consequently to a shorter cruise range, which explains why the optimization does not favor maximizing the wing span under the given constraint.

With the span evolution now clarified, the behavior of the chord lengths can be analyzed. In theory, increasing the aspect ratio would require a reduction in chord lengths. However, in this case, the optimal values for both the root and tip chords lie at their upper bounds.

This trend is directly linked to the definition of the objective function, which targets the minimization of the total drag coefficient. Since the reference area used in the drag coefficient calculation is the wing surface, increasing this area reduces the overall drag coefficient. Therefore, the optimizer tends to maximize the wing surface by increasing chord lengths. Nonetheless, this surface growth is constrained by the limited ability to increase the span, as previously explained.

A comparison of the objective function value can now be made between the two works. The reference study reports an optimal drag coefficient of $C_D = 0.02757$, while the RAPTOR framework yields a slightly higher value of $C_D = 0.02998$. Although these results are not identical, they remain reasonably close. One factor that may explain this discrepancy is the wing span. As shown earlier, the reference study converges toward a larger wing span than RAPTOR, which leads to a lower drag coefficient. Another important factor influencing C_D is the chord distribution. Unfortunately, the reference study does not provide the optimal values of the root and tip chord lengths, making a direct comparison on this point impossible. However, given that their reported values for wing span, planform surface area, and taper ratio are very close to those obtained in the present study, it is reasonable to assume that their chord lengths are also similar to those of RAPTOR. This would reinforce the idea that the small difference in drag coefficient is primarily due to the slightly larger span in the reference configuration.

Now, the evolution of the constraint functions over the optimization procedure can be discussed.

At the level of the range in cruise conditions, the final values found perfectly match between the reference and the present study.

With respect to the takeoff distance, both studies report values that satisfy the imposed constraint. However, a significant discrepancy exists between the two results: the reference study reports an optimized takeoff distance of 500.0 ft, whereas the present study, using RAPTOR, yields a value of 857.8 ft. In the present work, the takeoff distance is computed using a model adapted from the developments in Raymer [72], as detailed in Section 3.6. This model is based on a takeoff parameter (TOP), defined in Equation 3.6.3, which depends on several key quantities: aircraft weight, wing planform surface area, altitude, takeoff lift coefficient, and thrust-to-weight ratio. At the optimum, the aircraft weight and wing surface area are very similar in both studies, eliminating them as sources of discrepancy. Regarding the takeoff lift coefficient, the reference study indicates that it is scaled from the maximum lift coefficient, yet this value should remain constant throughout the optimization, as the airfoil geometry is not modified. The only remaining influential parameter is the available thrust. However, the reference study does not specify how thrust is calculated, nor does it report its value at the optimum. Given this lack of information and the significant impact of thrust on the TOP, it is reasonable to conclude that this difference in takeoff distance most likely stems from differing assumptions or models related to the available thrust.

A noticeable divergence between the present study and the reference concerns the rate of climb at the optimum configuration. In RAPTOR, the rate of climb converges to 1528.01 ft/min, a value that remains close to the lower bound of the design space (1500 ft/min). In contrast, the reference study reports a significant increase from 1500 ft/min to 2487.19 ft/min in its optimal solution. This discrepancy can be partially attributed to differences in drag performance. In RAPTOR, the optimized drag coefficient is $C_D = 0.0299$, whereas the reference study reports a lower value of $C_D = 0.0276$. Given the classical rate of climb expression (Equation 3.6.2), which shows an inverse relationship between drag and climb performance, this difference in drag likely plays a substantial role in the observed gap. As for the other parameters affecting the rate of climb, namely the aircraft weight during the climb segment and the available thrust, no major differences are expected regarding the weight, which is comparable in both studies. However, once again, the reference study does not provide the value of the available thrust at the optimum configuration, which prevents a detailed assessment of its impact. Nonetheless, the magnitude of the difference observed in the rate of climb suggests that thrust likely plays a significant role in this discrepancy.

Regarding the stall speed, noticeable differences arise between the results obtained with RAPTOR and those reported in the reference study. The maximum allowable stall speed was set to 84.39 ft/s. At the optimum, the value computed in RAPTOR is 77.30 ft/s, while the reference study reports a significantly lower value of 66.22 ft/s. Although both results respect the constraint, the value from the reference study appears unexpectedly low when examined in more detail. As a reminder, the stall speed is computed using Equation 3.6.4. In both studies, the flight gross weight at the optimum is identical. Additionally, in the simplified models employed in the reference study, the maximum lift coefficient $C_{L_{\max}}$ is determined solely by the airfoils used at the wing root and tip, and potentially the wing sweep angle, none of which are modified during optimization. Therefore, the only varying parameter in the stall speed formula is the wing planform surface area S_{w}. Recomputing the stall speed using the geometric characteristics described in the reference study, one finds a value that differs by approximately 17% from the reported 66.22 ft/s. This suggests a possible inconsistency in the stall speed calculation or a missing detail in the modeling assumptions of the reference.

Regarding the pitch moment slope ($C_{m\alpha}$), the final values obtained in RAPTOR and in the study by Thu et al. [85] differ slightly, but remain within the same order of magnitude.

For the yaw moment slope ($C_{n\beta}$), the values at the end of the process are very close: 0.06 in the reference study and 0.05 in the present work.

To summarize the comparison of the final constraint values presented in Table 4.13, it can be observed that all constraints are satisfied in both the present study and the reference by Thu et al. [85]. Several results show a strong agreement between the two studies, particularly for the cruise range, as well as the pitch and yaw moment slopes. The most significant discrepancies are found in the rate of climb and the takeoff distance.

4.2.5 Conclusion

This test case aimed to reproduce the optimization of a single-seat aerobatic aircraft, as originally investigated by Thu et al. [85].

The first step of the process consisted in selecting and implementing appropriate models for this highly multidisciplinary problem. Thanks to the modular and extensible architecture of the RAPTOR framework, integrating the necessary models and adapting a few others was straightforward.

Subsequently, a comparative analysis was carried out between the results obtained using RAPTOR and those reported in the reference study. This analysis revealed a strong agreement.

In the optimization phase, many of the resulting design variables converged toward similar values in both studies, and several shared trends were identified. These evolutions were examined and explained through analytical developments. Certain differences were also noted and investigated. In some cases, potential inconsistencies in the reference study were identified and discussed.

Overall, this case study highlights the ability of RAPTOR to successfully replicate a complex multidisciplinary aircraft optimization problem.

4.3 Optimization of a customized demonstration aircraft

4.3.1 Problem definition

The last test case consists in the optimization of an aircraft, for which the geometry has largely been inspired from the single-seat aerobatic aircraft previously considered. The external layout of this initial configuration (without its propulsion system) is represented in Figure 4.7.

To define the overall shape and scale of the aircraft configuration, the study conducted by Nguyen et al. [62] served as the primary reference. However, the selection of design variables and constraints was adapted and tailored to meet the specific objectives of the present work.

The wing is modeled with no sweep, twist, or dihedral, and it is aligned with the fuselage centerline, with zero initial incidence. Regarding the tail surfaces, the horizontal tail incorporates a quarter-chord sweep of 12 °, while the vertical tail is swept at 5 °. The tail airfoils are set to NACA 0012, and a NACA 63-2012 is used for the wing.

4.3.2 Optimization formulation

In this case, the objective is to minimize the aircraft drag coefficient under cruise conditions. The optimization is subject to several constraints, ensuring that the final configuration is stable, in

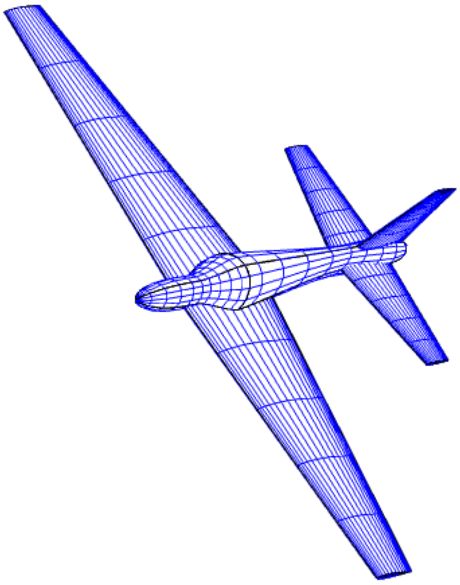


Figure 4.7: Representation of the initial aircraft geometry that aims to be optimized in this test case.

longitudinal equilibrium, and that the internal wing structure is capable of withstanding the external loads applied on it.

These design variables and their bounds are listed in Table 4.14, while the objective function and constraints are detailed in Table 4.15.

Variable	**Lower bound**	**Upper bound**
Wing span b_{w} (m)	8.5	11.6
Wing root chord $c_{\mathrm{r,w}}$ (m)	1.10	1.30
Wing tip chord $c_{\mathrm{t,w}}$ (m)	0.40	0.95
Wing skin thickness $t_{\mathrm{skin,w}}$ (m)	0.0002	0.001
Wing stringers area $A_{\mathrm{stringers,w}}$ (m^2)	$5 \cdot 10^{-6}$	$5 \cdot 10^{-5}$
Wing leading-edge position $X_{\mathrm{LE,w}}$ (m)	0.90	1.65
Horizontal tail span b_{ht} (m)	1.92	2.88
Horizontal tail root chord $c_{\mathrm{r,ht}}$ (m)	0.64	0.80
Horizontal tail tip chord $c_{\mathrm{t,ht}}$ (m)	0.30	0.60
Horizontal tail incidence i_{ht} (°)	–5.0	2.0
Vertical tail span b_{vt} (m)	1.00	1.50
Vertical tail root chord $c_{\mathrm{r,vt}}$ (m)	0.40	1.00
Vertical tail tip chord $c_{\mathrm{t,vt}}$ (m)	0.20	0.746
Angle of attack α (°)	–1.0	10.0

Table 4.14: Design variables and their bounds for the aircraft drag coefficient minimization problem.

4.3.3 Modeling approach

Several disciplinary models are involved in this optimization process.

Quantity	Relation	Target
Objective		
Aircraft drag coefficient C_D	→	Minimize
Constraints		
Static margin	≥	0.0
Wing skin panels strength margin	≥	0.0
Wing stringers strength margin	≥	0.0
Lift margin $\frac{L-W}{L}$	∈	[–0.05, 0.05]
Pitching moment coefficient C_m	∈	[–0.01, 0.01]

Table 4.15: Objective and constraints used in the optimization problem.

The geometry and its associated evaluations are managed through the CAD-based coupling with OpenVSP, as described in Section 3.1.

For the estimation of component weights, the wing mass is computed using the internal structure model implemented in RAPTOR, described in Section 3.2.1. The weights of the fuselage, horizontal tail, and vertical tail are evaluated using the CAD-based volume method introduced in Equation 3.2.10. Their material lightening ratio is set to 0.95 to approximately match the reference configuration described by Raymer [72]. The total weight of the aircraft and the position of its center of gravity are then estimated. To account for the remaining systems, such as the engine, landing gear, fuel system, flight controls, and electrical systems, a lumped mass approach is adopted. These components are represented using reference values inspired by the example of a comparable aircraft presented in Raymer [72], slightly adjusted downward to better highlight the influence of variations in the computed components. The total mass of these fixed systems is approximated at 314 kg, with their center of gravity positioned 1.15 m from the aircraft nose.

The aerodynamic computations rely on the empirical model presented in Section 3.3.1, applied to the wing, tails, fuselage, and the overall aircraft. To account for the drag contributions of the landing gear, cockpit, and engine, drag coefficient values from Raymer's single-seat aerobatic aircraft [72] are incorporated. These are normalized by the wing reference area to ensure consistency.

For stability and trim analysis, the low-fidelity static model introduced in Section 3.4 is used.

Finally, the structural evaluation relies on the model presented in Section 3.7. In this test case, only the wing structure is taken into account. The flight envelope is defined by setting the minimum and maximum load factors to –2.0 and 4.0, respectively. The cruise altitude is fixed at 2438 m. To determine the design cruise velocity, a constant engine power of 125,000 W and a propulsive efficiency of 0.8 are assumed. For the structural analysis, the wing cross-section is discretized into 18 stringers and 2 cells, bounded by spars located at 25 and 80 % of the chord. The third cell, corresponding to the control surface, is excluded from the analysis. The material used is Carbon Fiber Reinforced Polymer (CFRP), and a safety factor of 1.5 is applied in the structural limits. The outputs of this model are the skin panels and stringers strength margin. They are defined as the difference between the actual stringers area and panel thickness, and the minimum required values to withstand the external loads.

4.3.4 Optimization results and analysis

With the problem formulation established and the analysis models defined, the optimization process can then be carried out. Table 4.16 summarizes the evolution of the objective function, the constraints, and the design variables from the initial baseline to the final optimized configuration.

Metric	**Baseline**	**Optimum**
Design variables		
Wing span b_{w} (m)	6.4	11.6
Wing root chord $c_{\text{r,w}}$ (m)	1.28	1.30
Wing tip chord $c_{\text{t,w}}$ (m)	0.80	0.95
Vertical tail span b_{vt} (m)	1.25	1.00
Vertical tail root chord $c_{\text{r,vt}}$ (m)	1.22	0.72
Vertical tail tip chord $c_{\text{t,vt}}$ (m)	0.49	0.25
Horizontal tail span b_{ht} (m)	2.41	2.75
Horizontal tail root chord $c_{\text{r,ht}}$ (m)	0.68	0.77
Horizontal tail tip chord $c_{\text{t,ht}}$ (m)	0.50	0.38
Wing skin thickness $t_{\text{skin,w}}$ (m)	0.00050	0.00039
Wing stringers area $A_{\text{stringers,w}}$ (m^2)	1.0×10^{-5}	3.54×10^{-5}
Angle of attack α (°)	3.50	2.30
Wing leading-edge position $X_{\text{LE,w}}$ (m)	1.53	1.38
Horizontal tail incidence i_{ht} (°)	0.00	–2.12
Constraints		
Static margin SM	0.276	0.088
Wing skin panels strength margin	2.59×10^{-4}	0.0
Wing stringers strength margin	-2.60×10^{-6}	0.0
Lift margin $\frac{L-W}{L}$	–0.447	–0.050
Pitching moment coefficient C_m	–0.105	0.010
Objective		
Drag coefficient C_D	0.0401	0.0237

Table 4.16: Summary of the optimization results for the aircraft configuration using RAPTOR.

The first notable outcome is the significant reduction of the drag coefficient throughout the optimization process, clearly demonstrating the effectiveness of the optimization. The ratio between the optimum and the baseline value is close to 0.5. This strong improvement is explained by the evolution of the design variables, discussed in the next developments.

Regarding the constraints, all are satisfied at the optimal configuration. The aircraft remains stable, with a positive static margin. It is also in equilibrium, as the pitching moment coefficient with respect to the center of gravity falls within the required tolerance, and the lift matches the weight. The internal structure is also robust enough to withstand the loads from the critical flight conditions.

Focusing on the design variables, the first observation is made about the wing geometry. The wing span, as well as the root and tip chords, all converge toward their upper bounds. This results in an increased aspect ratio, evolving from 6.15 to 10.31, which aligns with aerodynamic principles: a higher aspect ratio reduces induced drag. However, the aspect ratio is not pushed to its maximum value, which would imply minimizing the chord lengths. This compromise is explained by the fact that the wing planform area serves as the reference area for the drag coefficient contributions of other components. Consequently, the optimizer tends to maximize the wing surface area.

For the wing internal structure, the skin thickness decreases while the stringers area increases. These two design variables influence both the structural integrity and the overall weight of the aircraft. Their evolution therefore reflects the way the optimizer adapts the structure to respect the constraints, while minimizing the objective.

At the level of the vertical tail geometry, both the span and chord lengths converge toward the lower bounds of the design space. This trend can be directly linked to drag reduction: the parasite drag contribution of the vertical tail decreases with a reduction in its mean aerodynamic chord and wetted surface area. This explains the observed evolution of the vertical tail parameters during the optimization.

Regarding the horizontal tail, its geometric evolution is straightforward: the span and the root chord reach their upper bound while both the tip chord length is minimized. This leads to a higher aspect ratio, which reduces the induced drag contribution of this surface. The fact that the root chord length is not minimized here can result from the stabilization of the aircraft.

A noticeable decrease in the angle of attack is observed, which results in a reduction in lift and, consequently, a lower induced drag coefficient.

The position of the wing leading edge also shifts, from 1.53 m to 1.30 m from the aircraft's nose. This adjustment contributes to achieving the required equilibrium and longitudinal stability.

In a similar way, the horizontal tail incidence relative to the fuselage centerline is slightly reduced. This modification not only supports pitch stability but also reduces the lift generated by the horizontal tail, thereby lowering its induced drag contribution.

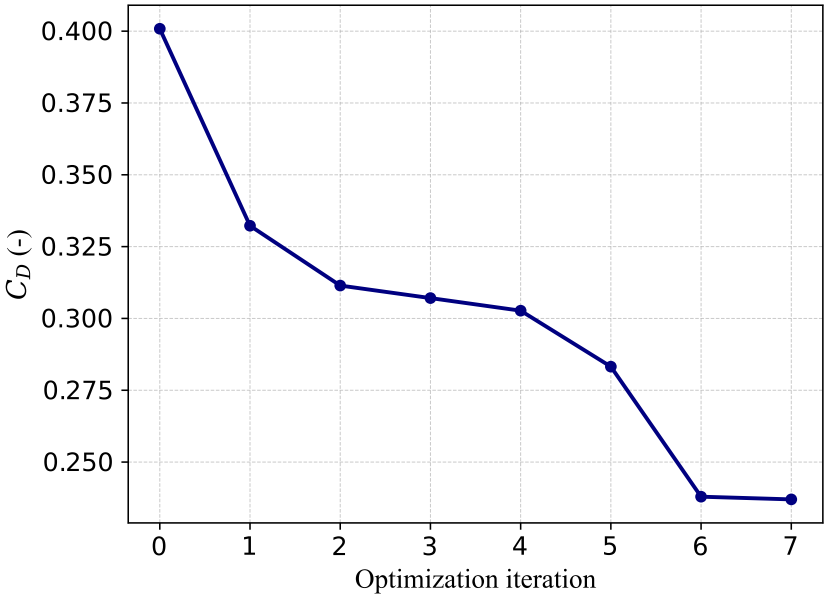
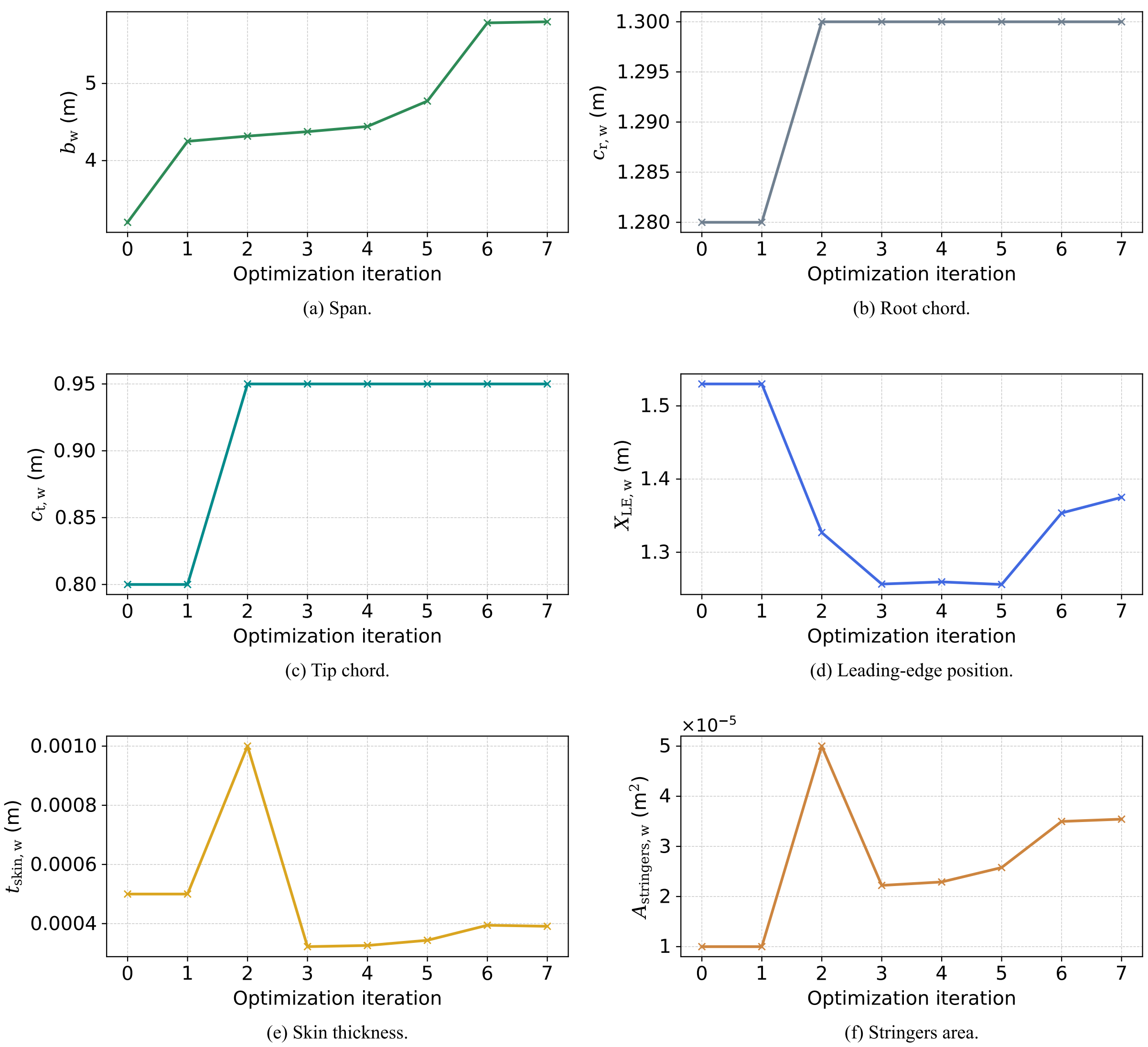


Figure 4.8: Evolution of the objective function C_D throughout the optimization process.

The evolution of the objective function, namely the aircraft drag coefficient during the optimization is represented in Figure 4.8. The plot reveals a steady and continuous decrease in drag coefficient until the optimum, in a smooth way. In only seven iterations, the optimum is found, highlighting the efficiency of the optimization process. This smooth and rapid convergence also suggests that the design space is well-defined and that the selected design variables have a strong and direct influence on the drag coefficient of the aircraft.

The pattern followed by the of the wing design variables is shown in Figure 4.9. The wing span exhibits a continuous increase throughout the optimization process. The root and tip chords rapidly grow and reach the upper bounds of their respective design spaces. The leading-edge position initially decreases before slightly increasing again to satisfy the imposed constraints. As for the structural parameters, both the skin thickness and the stringers area show an early peak before gradually converging toward their final optimized values.



(a) Span.

(b) Root chord.

(c) Tip chord.

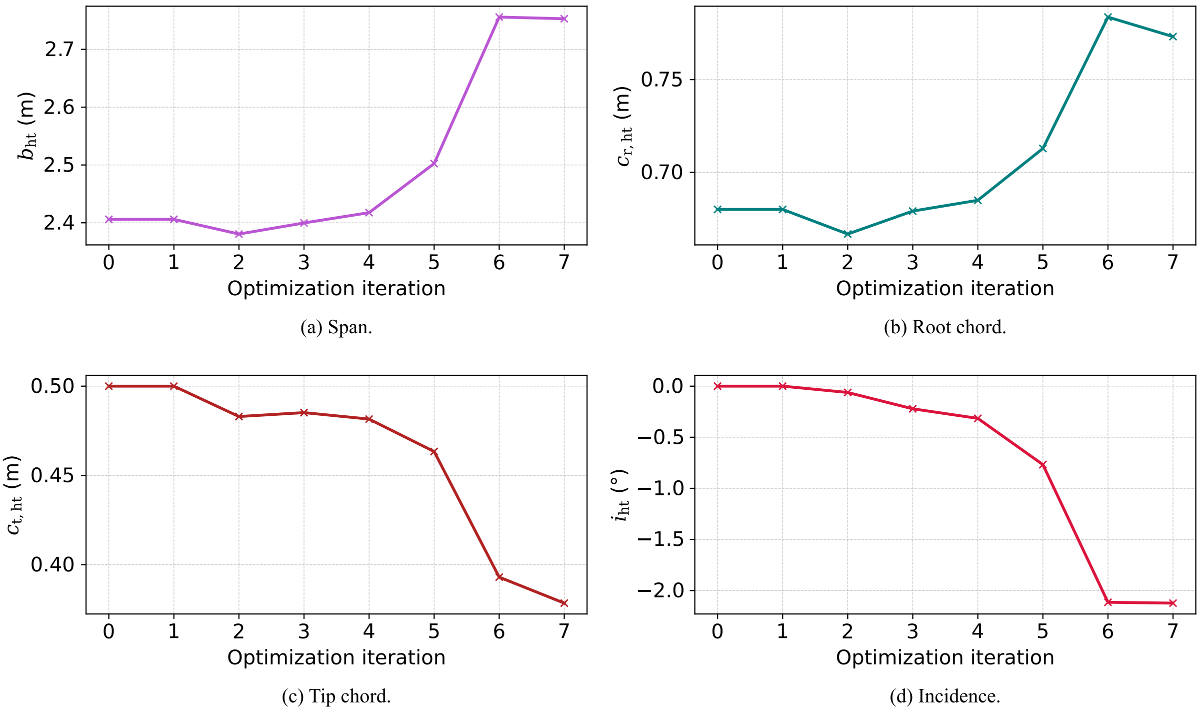
(d) Leading-edge position.

(e) Skin thickness.

(f) Stringers area.

Figure 4.9: Evolution of the wing design variables throughout the optimization process. Subplots correspond to: (a) b_{w}, (b) $c_{\mathrm{r,w}}$, (c) $c_{\mathrm{t,w}}$, (d) $X_{\mathrm{LE,w}}$, (e) $t_{\mathrm{skin,w}}$, and (f) $A_{\mathrm{stringers,w}}$.

The evolution of the horizontal tail design variables is presented in Figure 4.10. The span remains constant during the first four iterations, then gradually increases to reach the upper bound of the design space. A similar trend is observed for the root chord length. In contrast, both the tip chord and the incidence angle steadily decrease throughout the optimization process. As previously discussed, these changes generally contribute to a reduction in the induced drag coefficient of the horizontal tail, while also likely supporting the overall stability of the aircraft.



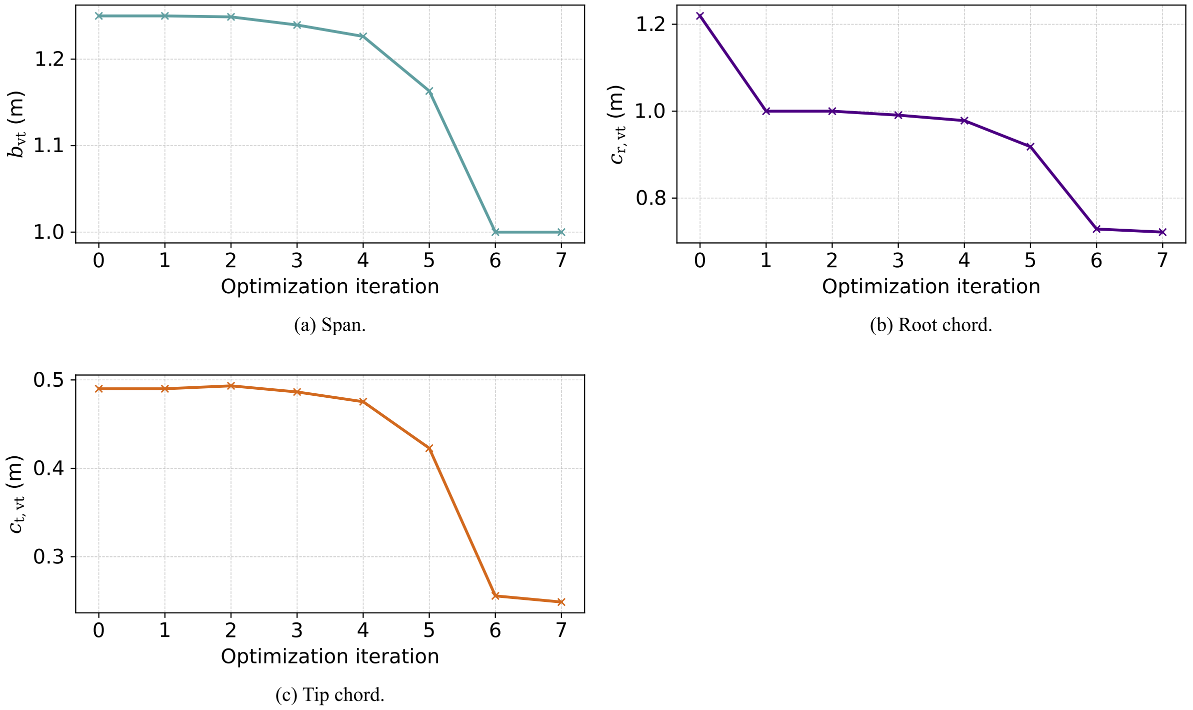
(a) Span.

(b) Root chord.

(c) Tip chord.

(d) Incidence.

Figure 4.10: Evolution of the horizontal tail design variables throughout the optimization process. Subplots correspond to: (a) b_{ht}, (b) $c_{\mathrm{r,ht}}$, (c) $c_{\mathrm{t,ht}}$, and (d) i_{ht}.



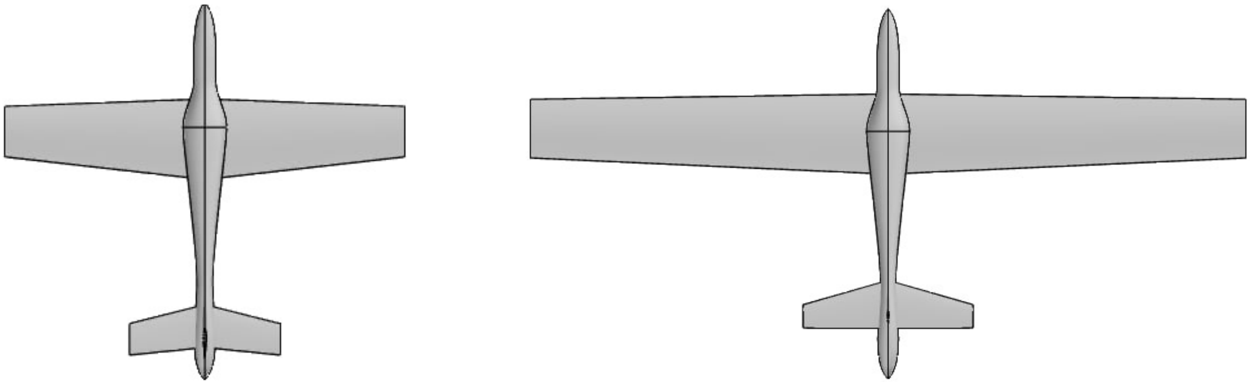
(a) Span.

(b) Root chord.

(c) Tip chord.

Figure 4.11: Evolution of the vertical tail design variables throughout the optimization process. Subplots correspond to: (a) b_{vt}, (b) $c_{\mathrm{r,vt}}$, and (c) $c_{\mathrm{t,vt}}$.

The pattern followed by the vertical tail span, root chord, and tip chord during the optimization is illustrated in Figure 4.11. All three variables exhibit a continuous decrease throughout the process. This trend reflects a reduction of the vertical tail wetted surface area, which leads to a decrease in its parasite drag contribution. Since this surface contributes to the total drag of the aircraft, the observed evolution is both expected and coherent with the objective of minimizing the overall drag coefficient.



(a) Baseline configuration.

(b) Optimized configuration.

Figure 4.12: Top view comparison between the baseline and optimized aircraft configurations. The optimized layout shows a longer wing span and modified tail geometries.

Finally, the top view comparison between the baseline aircraft and the optimized one is represented in Figure 4.12.

4.3.5 Conclusion

This test case focused on the aerodynamic optimization of a light general aviation aircraft, with the objective of minimizing its drag coefficient.

In a first stage, all necessary models were selected to represent the different disciplines involved, namely geometry generation, mass estimation, aerodynamic performance, static equilibrium and stability, and structural analysis.

The optimization was then performed using a set of design variables related to the wing, horizontal tail, and vertical tail geometries, as well as the flight conditions. Their evolution, along with the progression of the objective function, was then analyzed. In this case, the drag coefficient was significantly reduced.

This study confirms the ability of RAPTOR to carry out robust multidisciplinary and multi-fidelity optimization. The rapid convergence of the procedure also highlights the efficiency of the framework's overall architecture.

Chapter 5

Conclusion

5.1 Summary of the work

The objective of this thesis was to develop a preliminary aircraft design optimization framework, named RAPTOR. To this end, the software architecture was built upon OpenMDAO[23], a platform particularly well-suited for the integration and coupling of the various disciplines involved in aircraft design. A gradient-based optimization approach was selected to efficiently handle high-dimensional design spaces, which are common in this field. Special attention was given to ensuring modularity, flexibility, and validity, while also enabling the integration of multi-fidelity modeling capabilities.

To ensure modularity, the models were developed with a clear separation between disciplines, physical components, and analysis methods. This design choice allows the framework to handle a wide range of aircraft configurations, from conventional layouts to more unconventional or exotic designs.

In terms of flexibility, it was considered essential to ensure that future developers of RAPTOR can easily modify or integrate new models into the framework. To facilitate this, automatic differentiation was chosen as the method for computing optimization gradients, due to its ease of implementation and computational efficiency. As a result, a developer only needs to implement their model in a differentiable manner for it to be directly usable in the optimization workflow.

Following these key architectural choices, the analysis models were developed and validated. They were also adapted to ensure the robustness of the solutions, even in the presence of unconventional inputs provided by the framework users. Additional modifications were made to guarantee differentiability, enabling the use of automatic differentiation for their integration into optimization workflows. This part of the work also focused on implementing a multi-fidelity approach. Low-fidelity models were first developed and validated. To enhance accuracy, higher-fidelity alternatives were then introduced by coupling the framework with external tools. In particular, OpenVSP [69] was used for CAD-based modeling, geometry calculations and weight estimations, and SDPM [10] for panel method computations in aerodynamics and stability fields.

The capabilities of the framework and the validity of the optimization process were subsequently demonstrated through several test cases. First, the UAV optimization case study presented by Kim et al.[38] was reproduced using the low-fidelity models developed in this work. The results showed strong agreement with the reference data, thereby validating the core components of the framework. The same case was then re-evaluated using the higher-fidelity models available in RAPTOR, revealing both similarities and certain discrepancies, which were analyzed and discussed. In addition, another reference aircraft case from the literature was tested, again yielding comparable trends alongside some differences. Finally, a demonstrative test case was formulated to illustrate the framework's ability to perform multidisciplinary and multi-fidelity optimization. This example combined several key disciplines, such as aerodynamics, weight estimation, and internal structural calculations, and employed models of varying fidelity across different components, reflecting the flexible and modular nature of the RAPTOR framework. These results demonstrate that RAPTOR provides a solid foundation for aircraft design optimization, combining modularity, flexibility, and extensibility. It is well-suited to support future developments, whether for the integration of higher-fidelity models, the extension to new disciplines, or the application to a wider range of aircraft configurations.

5.2 Future work

Throughout this study, several opportunities for future improvement have been identified.

First, regarding the flow regime in which the optimized aircraft operates, the current models are primarily suited for subsonic conditions, particularly in the aerodynamic domain. Extending the framework to include transonic aerodynamic models would represent a significant enhancement.

Although the multidisciplinary nature of aircraft design is already well captured in this work, with key disciplines covered, the inclusion of additional areas, such as aeroelasticity, dynamic stability, or more detailed performance modeling would further strengthen the framework's ability to handle complex and realistic design scenarios.

As previously discussed, one of the key features of the framework is its use of automatic differentiation to compute gradients efficiently. However, in the current implementation, the aerodynamic and stability models based on SDPM [10] still rely on finite difference approximations, which are less accurate and computationally more expensive. This limitation stems from the fact that, while gradients can be computed with respect to mesh node positions, they are not directly available with respect to high-level geometric parameters such as span or chord. To overcome this issue, implementing a mesh deformation strategy would enable the propagation of geometric variations through the mesh, thereby allowing the aerodynamic model to remain differentiable with respect to design parameters. In this context, the pyGeo framework [27] has been identified as a suitable solution. Future work will focus on the use its functionalities to compute the derivatives of the surface mesh points with respect to geometric design variables.

Additionally, the current SDPM-based model supports only lifting surfaces, such as wings and tails, but does not account for bodies like the fuselage. Extending the model to handle non-lifting components would represent another valuable enhancement to the aerodynamic modeling capabilities of the framework.

Finally, the range of multi-fidelity options could be expanded. While multiple fidelity levels are already available in several disciplines, other areas such as structural analysis currently rely on a single fidelity level. Enhancing this aspect could involve the implementation or coupling of Finite Element Methods with RAPTOR, although such an integration would likely require substantial development effort. Similarly, propulsion modeling was not a focus in the current version of the framework. However, the inclusion of more accurate models, such as Blade Element Momentum Theory for propeller-driven aircraft would provide a more realistic representation of propulsion effects and improve the consistency of performance evaluations in relevant configurations.

Bibliography

[1] M. A. Saja Abdul-Kaiyoom, Anil Yildirim, and Joaquim R. R. A. Martins. "Robust Schur-Complement Solvers and Large-Scale Multidisciplinary Design Optimization". In: Proceedings of the 34th Congress of the International Council of the Aeronautical Sciences (ICAS). Florence, Italy, 2024.

[2] Huseyin Acar et al. Historical Trends and Future Projections of Key Performance Parameters in Aircraft Design. Technical Report. Accessed May 2025. Department of Aerospace Engineering, University of Michigan, 2024.

[3] Airbus. Global Market Forecast 2024–2043. Airbus, 2024. Available at: `https://www.airbus.com/en/aircraft/market/global-market-forecast`. Accessed May 2025.

[4] J. D. Anderson. Fundamentals of Aerodynamics. 4th International Edition. McGraw-Hill, 2006.

[5] Thomas Backhaus et al. "Advancing Modularity and Framework Integration Level for Scalable High-Fidelity MDO". In: AIAA AVIATION Forum. San Diego, CA, 2023. DOI: `10.2514/6.2023-3315`.

[6] A. Bardenhagen, H. Kossira, and W. Heinze. "Interdisciplinary Design of Modern Hypersonic Waveriders Using the Integrated Program PrADO-HY". In: Proceedings of the 19th International Congress of the Aeronautical Sciences. Anaheim, CA, USA, 1994, pp. 1053–1063. URL: `https://www.icas.org/ICAS_ARCHIVE/ICAS1994/data/papers/ICAS9444.PDF`.

[7] D. Böhnke, B. Nagel, and V. Gollnick. "An Approach to Multi-Fidelity in Conceptual Aircraft Design in Distributed Design Environments". In: Proceedings of the 2011 Aerospace Conference. Big Sky, MT, USA: IEEE, 2011, pp. 1–10. DOI: `10.1109/AERO.2011.5747391`.

[8] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Available at: `https://web.stanford.edu/~boyd/cvxbook/`. Cambridge University Press, 2004. ISBN: 9780521833783.

[9] J. A. Corman et al. "Rapid Airframe Design Environment (RADE): A Parametric, Modular, and Multidisciplinary Framework for Conceptual Phase Airframe Design". In: Proceedings of the 2018 AIAA Modeling and Simulation Technologies Conference. AIAA Paper 2018-1753. Reston, VA, USA, 2018.

[10] Adrien Crovato. SDPM – Source and Doublet Panel Method: Theory Manual and Quick Reference Guide. Technical Report. Version 1.2.0. Available at: `https://gitlab.uliege.be/am-dept/sdpm`. Accessed May 2025. Department of Aerospace & Mechanical Engineering, University of Liège, 2024.

[11] DARcorporation. Advanced Aircraft Analysis (AAA) Software. DARcorporation, 2024. Available at: `https://www.darcorp.com/advanced-aircraft-analysis-software/`. Accessed May 12, 2025.

[12] Dassault Systèmes. CATIA. Version V5-6R2024. Dassault Systèmes, 2024. Available at: `https://www.3ds.com/products-services/catia/`. Accessed May 2025.

[13] Mark Drela. AVL – Athena Vortex Lattice. Massachusetts Institute of Technology, 2001. Available at: http://web.mit.edu/drela/Public/web/avl/. Accessed May 2025.

[14] Mark Drela. TASOPT: A Transport Aircraft System Optimization Framework. Technical Report. Version 1.0. Accessed May 12, 2025. Massachusetts Institute of Technology, 2017.

[15] Cheryl Eisler. "Multidisciplinary Optimization of Conceptual Aircraft Design". Available at: https://www.researchgate.net/publication/316691398. Accessed May 2025. PhD thesis. Defence Research and Development Canada, 2003.

[16] Abdulrahman Abdullah Farag et al. "Exploring Optimization Algorithms: A Review of Methods and Applications". In: Journal of Artificial Intelligence and Metaheuristics (JAIM) 7.2 (2024), pp. 8–17. DOI: 10.54216/JAIM.070201.

[17] Federal Aviation Administration. FAR – Part 23: Airworthiness Standards for Normal, Utility, Acrobatic, and Commuter Aircraft. Federal Aviation Administration, 2025. Available at: https://www.risingup.com/fars/info/23-index.shtml. Accessed May 2025.

[18] R. D. Finck. USAF Stability and Control DATCOM. Technical Report AFWAL-TR-83-3048. Wright-Patterson Air Force Base, OH: McDonnell Douglas Corporation, 1978.

[19] Roger Fletcher. Practical Methods of Optimization. 2nd. New York: John Wiley & Sons, 1987. ISBN: 978-0-471-91547-8.

[20] C. Geuzaine and J.-F. Remacle. "Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities". In: International Journal for Numerical Methods in Engineering 79.11 (2009), pp. 1309–1331. DOI: 10.1002/nme.2579.

[21] Devendra Ghate. Gradient Calculation in MDO. 2023. Available at: https://devendra-ghate.github.io/courses/mdo/notes/gradientCalc.html. Accessed May 25, 2025.

[22] J. P. Giesing and J.-F. M. Barthelemy. A Summary of Industry MDO Applications and Needs. Summary White Paper AIAA 98-4737. Available at: http://adg.stanford.edu/aa222/giesingsummary_paper.pdf. Accessed May 2025. AIAA, 1998.

[23] Justin S. Gray et al. "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization". In: Structural and Multidisciplinary Optimization 59.4 (2019), pp. 1075–1104. DOI: 10.1007/s00158-019-02211-z.

[24] Stéphane Grihon, Lars Krog, and David Bassir. "Numerical Optimization Applied to Structure Sizing at AIRBUS: A Multi-Step Process". In: International Journal for Simulation and Multidisciplinary Design Optimization 3 (2011), pp. 432–443. DOI: 10.1051/ijsmdo/2009020.

[25] Roger Grosse. Lecture 10: Automatic Differentiation. CSC321: Introduction to Neural Networks and Machine Learning, University of Toronto, 2018. Available at: https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/slides/lec10.pdf. Accessed May 25, 2025.

[26] Snorri Gudmundsson. General Aviation Aircraft Design: Applied Methods and Procedures. Second Edition. Academic Press, 2021. ISBN: 978-0-12-818465-3.

[27] Hannah M. Hajdik et al. "pyGeo: A geometry package for multidisciplinary design optimization". In: Journal of Open Source Software 8.87 (2023), p. 5319. DOI: 10.21105/joss.05319.

[28] P. Hajela. Optimal Design in Multidisciplinary Systems. Course Outline. AIAA Professional Study Series, 2000.

[29] Mark A. Hale, Dimitri N. Mavris, and Dennis L. Carter. "The Implementation of a Conceptual Aerospace Systems Design and Analysis Toolkit". In: Proceedings of the World Aviation Congress (SAE/AIAA 1999-01-5639). Society of Automotive Engineers. San Francisco, CA, USA, 1999.

[30] Mohamad Azrin Syafiq Halim and Yeak Su Hoe. "Introduction to Automatic Differentiation and Neural Differentiation Equation". In: Proceedings of the Scientific Mathematics Journal (Proc. Sci. Math.) 24 (2024), pp. 81–89. URL: `https://science.utm.my/procscimath`.

[31] Warren Hoburg and Pieter Abbeel. "Geometric Programming for Aircraft Design Optimization". In: AIAA Journal (2012). Accessed May 2025. URL: `https://hoburg.github.io/gpkit/`.

[32] Saeed Hosseini, Mohammad Ali Vaziry-Zanjany, and Hamid Reza Ovesy. "A Framework for Aircraft Conceptual Design and Multidisciplinary Optimization". In: Aerospace 11.4 (2024), p. 273. DOI: `10.3390/aerospace11040273`.

[33] ISAE-SUPAERO Aircraft Design Team. FAST-OAD-GA: An Open-Source Aircraft Design Framework. 2024. Available at: `https://github.com/supaero-aircraft-design/FAST-GA`. Accessed May 12, 2025.

[34] A. T. Isikveren. "Multidisciplinary Design Optimisation and Conceptual Aircraft Design". Available at: `https://kth.diva-portal.org/smash/record.jsf?pid=diva2:9126`. Accessed May 2025. PhD thesis. Stockholm, Sweden: KTH Royal Institute of Technology, 2002.

[35] Laure Jaeger et al. "Aircraft Multidisciplinary Design Optimization Under Both Model and Design Variables Uncertainty". In: Journal of Aircraft 50.2 (2013). Available on HAL, pp. 528–538. DOI: `10.2514/1.C031914`.

[36] K. H. Khan et al. "Distributed Design Optimization of Large Aspect Ratio Wing Aircraft with Rapid Transonic Flutter Analysis in Linux". In: Proceedings of the AIAA SciTech 2021 Forum. Virtual Event, USA, 2021.

[37] Ju-Hoe Kim and Takeshi Tsuchiya. "OpenVSP Based Aerodynamic Design Optimization Tool Building Method and Its Application to Tailless UAV". In: Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences (ICAS). 2022.

[38] Jiha Kim. Optimization Theory for Machine Learning – Iterative Methods: Gradient-Free vs Gradient-Based. 2023. Available at: `https://jiha-kim.github.io/series/optimization-theory-for-machine-learning/2-iterative-methods-gradient-free-vs-based/`. Accessed May 25, 2025.

[39] Ilan M. Kroo and M. Takai. "A Quasi-Procedural, Knowledge-Based System for Aircraft Design". In: Proceedings of the AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting. AIAA Paper 88-4502. Atlanta, GA, USA, 1988. DOI: `10.2514/6.1988-4502`.

[40] S. LeDoux et al. "MDOPT—A Multidisciplinary Design Optimization System Using Higher Order Analysis Codes". In: Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. AIAA Paper 2004-4460. Albany, NY, USA, 2004.

[41] Lissys Ltd. Piano – Aircraft Performance Program. 2024. Available at: `https://www.lissys.uk/index2.html`. Accessed May 12, 2025.

[42] Dong C. Liu and Jorge Nocedal. "On the Limited Memory Method for Large Scale Optimization". In: Mathematical Programming B 45.3 (1989), pp. 503–528. DOI: `10.1007/BF01589116`.

[43] Lourenço Manuel Martins Lúcio. "Multidisciplinary Aircraft Design and Trajectory Control Optimization". Master's thesis. Lisbon, Portugal: Instituto Superior Técnico, University of Lisbon, 2019. URL: `https://fenix.tecnico.ulisboa.pt/downloadFile/1689241049735946/Thesis_LourencoLucio.pdf`.

[44] Florent Lutz et al. “FAST-OAD-GA: An Open-Source Extension for Overall Aircraft Design of General Aviation Aircraft”. In: Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences (ICAS). ICAS Paper No. ICAS-2022-7.1.1. Stockholm, Sweden, 2022. URL: https://hal.science/hal-03828230v1.

[45] Rabii El Maani, Hocine Imine, and Mohamed El Hadi Latreche. “Study of the Turbulence Models over an Aircraft Wing”. In: Incertitudes et Fiabilité des Systèmes Multiphysiques (2018). DOI: 10.21494/ISTE.OP.2018.0306.

[46] Floris Mariën. “Software Testing: VSPAERO”. MA thesis. Hamburg University of Applied Sciences, 2021. DOI: 10.15488/11559. URL: https://nbn-resolving.org/urn:nbn:de:gbv:18302-aero2021-07-16.018.

[47] Mariano Sánchez Martínez and Grigorios Dimitriadis. “Subsonic source and doublet panel methods”. In: Journal of Fluids and Structures 113 (2022), p. 103624. DOI: 10.1016/j.jfluidstructs.2022.103624.

[48] Joaquim R. R. A. Martins. “CFD-based Aircraft Design Optimization”. In: Keynote Presentation, AIAA Aviation Forum (2019). Available at https://www.researchgate.net/publication/336547383_CFD-based_Aircraft_Design_Optimization.

[49] Joaquim R. R. A. Martins and Andrew B. Lambe. “Multidisciplinary Design Optimization: A Survey of Architectures”. In: AIAA Journal 51.9 (2013), pp. 2049–2075. DOI: 10.2514/1.J051895.

[50] Joaquim R. R. A. Martins and Andrew Ning. Engineering Design Optimization. Cambridge University Press, 2021. ISBN: 9781108833417. URL: https://mdobook.github.io.

[51] Bento Silva de Mattos and Paulo Eduardo Cypriano da Silva Magalhães. “Aircraft Design Phases”. In: Conceptual Design of Green Transport Airplanes. Ed. by Bento Silva de Mattos. Bentham Science Publishers, 2018, pp. 203–241. DOI: 10.2174/9781681083278118030006. URL: https://www.researchgate.net/publication/323698903_Aircraft_Design_Phases.

[52] MDO Lab, University of Michigan. OpenAeroStruct: Open-Source Tool for Aero-Structural Optimization. 2024. Available at: https://github.com/mdolab/OpenAeroStruct. Accessed May 12, 2025.

[53] T. H. G. Megson. Aircraft Structures for Engineering Students. Elsevier Aerospace Engineering Series. Elsevier Science, 2007.

[54] MSC Software Corporation. MSC Nastran – Finite Element Analysis Software. Version 2024, https://www.mscsoftware.com/product/msc-nastran. 2024.

[55] R.C. Munjulury, P. Berry, and P. Krus. “RAPID—Robust Aircraft Parametric Interactive Design: A Knowledge Based Aircraft Conceptual Design Tool”. In: Proceedings of the 4th CEAS Conference. Linköping, Sweden, 2013, pp. 255–262. URL: http://liu.diva-portal.org/smash/record.jsf?pid=diva2:687478.

[56] R.C. Munjulury et al. “A Knowledge-Based Integrated Aircraft Conceptual Design Framework”. In: CEAS Aeronautical Journal 7 (2016), pp. 95–105.

[57] B. Nagel et al. “Communication in Aircraft Design: Can We Establish a Common Language?” In: Proceedings of the 28th ICAS Congress. Brisbane, Australia, 2012. URL: https://www.icas.org/ICAS_ARCHIVE/ICAS2012/ABSTRACTS/201.

[58] NASA. OpenVSP API Documentation. 2024. Available at: https://openvsp.org/api_docs/latest/. Accessed May 15, 2025.

[59] NASA Ames Research Center and Virginia Tech CAD Laboratory. “ACSYNT: Aircraft SYNThesis Tool for Conceptual Aircraft Design”. In: NASA Spinoff (1996).

[60] NASA Aviary Team. Aviary: Open-Source Aircraft Design and Optimization Framework. 2024. Available at: https://www.nasa.gov/aeronautics/aviary-software-overview/. Accessed May 12, 2025.

[61] NASA Glenn Research Center. Thrust to Weight Ratio – Beginner's Guide to Aeronautics. 2023. Available at: https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/thrust-to-weight-ratio/. Accessed May 2025.

[62] Nhu Van Nguyen et al. "Multidisciplinary Configuration Design Optimization for Advanced Very Light Aircraft". In: Proceedings of the Korean Society for Aeronautical and Space Sciences Conference. Seoul, Korea, 2011, pp. 18–23. URL: https://www.researchgate.net/publication/297758923.

[63] L. M. Nicolai and A. Carty. Role of the Aerodynamicist in a Concurrent Multi-disciplinary Design Process. Tech. rep. Lockheed Martin Skunk Works, 1999.

[64] Fabrizio Nicolosi et al. "Aerodynamic Design and Analysis of an Innovative Regional Turboprop Configuration". In: ICAS 2018 – 31st Congress of the International Council of the Aeronautical Sciences. 2018. URL: https://www.icas.org/ICAS_ARCHIVE/ICAS2018/data/papers/ICAS2018_0496_paper.pdf.

[65] Jorge Nocedal and Stephen J. Wright. Numerical Optimization. 2nd. Springer Science & Business Media, 2006. ISBN: 978-0-387-30303-1.

[66] Ludovic Noels. Aircraft Structures – Design Example. http://www.ltas-cm3.ulg.ac.be/. Lecture notes based on "Aircraft Structures for Engineering Students" by T. H. G. Megson, Elsevier, 2003. 2013.

[67] J. R. Olds. "The Suitability of Selected Multidisciplinary Design and Optimization Techniques to Conceptual Aerospace Vehicle Design". In: 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization. AIAA Paper 92-4791. Cleveland, OH, 1992, pp. 820–843.

[68] OpenCFD Ltd. OpenFOAM – The Open Source CFD Toolbox. 2024. Available at: https://www.openfoam.com/. Version accessed May 2025.

[69] OpenVSP Development Team. OpenVSP Official Website. 2024. Available at: http://openvsp.org/. Accessed May 12, 2025.

[70] Optimal Aircraft Design. Aircraft Design Software (ADS). 2024. Available at: https://www.pca2000.com/en/aircraft-design-software.php. Accessed May 12, 2025.

[71] Rajkumar Pant. "Optimization of Regional Aircraft Configurations from Passengers' Perspective". In: Proceedings of the 20th Congress of the International Council of the Aeronautical Sciences (ICAS). ICAS Paper No. ICAS-96-7.7.1. Sorrento, Italy, 1996.

[72] Daniel P. Raymer. Aircraft Design: A Conceptual Approach. Sixth. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 2018.

[73] Daniel P. Raymer. "Conceptual Design Modeling in the RDS-Professional Aircraft Design Software". In: 49th AIAA Aerospace Sciences Meeting. AIAA Paper 2011-161. American Institute of Aeronautics and Astronautics. Orlando, FL, USA, 2011.

[74] R. Russell Rhinehart. "Optimization Algorithm Selection for Process Applications". In: InTech Magazine (2019). URL: https://www.isa.org/intech-home/2019/november-december/features/optimization-algorithm-selection-for-process-appli.

[75] Jan Roskam. Airplane Design Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics. Lawrence, KS: DARcorporation, 2000.

[76] Jan Roskam and Willem A. Anemaat. "General Aviation Aircraft Design Methodology in a PC Environment". In: Proceedings of the AIAA Aircraft Design, Systems and Operations Conference. AIAA Paper 96-5520. 1996.

[77] Ideen Sadrehaghighi. Aircraft Multidisciplinary Design & Optimization (MDO). Tech. rep. CFD Open Series, 2023. DOI: 10.13140/RG.2.2.24725.76003.

[78] Paul Saves et al. “Bayesian optimization for mixed variables using an adaptive dimension reduction process: applications to aircraft design”. In: AIAA SciTech 2022 Forum. Also available at: https://www.researchgate.net/publication/388272854. Accessed May 2025. San Diego, United States, 2022. DOI: 10.2514/6.2022-0082.

[79] Thomas Schwinn. “Development of a Multidisciplinary Process Chain for the Preliminary Design of Aircraft Structures”. In: Proceedings of the 4th CEAS Air & Space Conference. Linköping, Sweden, 2013. URL: https://elib.dlr.de/86040/1/2013_06_10_schwinn_NWC2013_development_of_a_multidisciplinary_process_chain_for_the_preliminary_design_of_aircaft_structures_paper.pdf.

[80] K. Seeckt and D. Scholz. “Application of the Aircraft Preliminary Sizing Tool PreSTo to Kerosene and Liquid Hydrogen Fueled Regional Freighter Aircraft”. In: Proceedings of the Deutscher Luft- und Raumfahrtkongress 2010: Tagungsband – Ausgewählte Manuskripte. Available at: https://perma.cc/K94E-BW3Z. Hamburg, Germany, 2010.

[81] Siemens Digital Industries Software. Siemens NX – Product Engineering Software. 2024. Available at: https://plm.sw.siemens.com/en-US/nx/. Accessed May 2025.

[82] Steven M. Sliwa and P. Douglas Arbuckle. OPDOT: A Computer Program for the Optimum Preliminary Design of a Transport Airplane. Tech. rep. NASA-TM-81857. NASA Technical Memorandum 81857. Hampton, Virginia: NASA Langley Research Center, 1980.

[83] SU2 Developers. SU2: The Open-Source CFD Code. 2024. Available at: https://github.com/su2code/SU2. Accessed May 2025.

[84] SUAVE Development Team. SUAVE: An Open-Source Environment for Aircraft Conceptual Design. 2024. Available at: https://suave.stanford.edu/documentation.html. Accessed May 12, 2025.

[85] Zin Win Thu, Aung Myo Thu, and Phyo Wai Thaw. “Design and Optimization of Aircraft Configuration for Minimum Drag”. In: International Journal of Aviation, Aeronautics, and Aerospace 5.5 (2018). URL: https://commons.erau.edu/ijaaa/vol5/iss5/10.

[86] Yuan Tian, Hong Liu, and Jian Huang. “Design Space Exploration in Aircraft Conceptual Design Phase Based on System-of-Systems Simulation”. In: International Journal of Aeronautical and Space Sciences 16.4 (2015), pp. 624–635. DOI: 10.5139/IJASS.2015.16.4.624.

[87] E. Torenbeek. Synthesis of Subsonic Airplane Design. Delft University Press, 1982.

[88] Nansi Xue et al. “Optimization of a Single Lithium-Ion Battery Cell with a Gradient-Based Algorithm”. In: Journal of The Electrochemical Society 160.8 (2013), A1071–A1078. DOI: 10.1149/2.036308jes.