Università degli Studi di Genova

Scuola Politecnica

DIME

Dipartimento di Ingegneria Meccanica, Energetica,
Gestionale e dei Trasporti

Tesi di Laurea Magistrale

in

Ingegneria Meccanica – Energia e Aeronautica

Surrogate-Based Shape Optimisation for Aircraft Design: Implementation in *CEASIOMpy*

Relatori:
Chiar.mo Prof. Ing. Alessandro Bottaro
Dott. Ing. Jan B. Vos

Correlatore:
Dott. Giacomo Benedetti

Studente:
Marco Ilmi Rushani

Marzo 2026

Abstract

This thesis was developed within the European research project COLOSSUS during a curricular internship at CFS Engineering in collaboration with the University of Genoa. The work focuses on the development and integration of new capabilities into *CEASIOMpy*, an open-source framework for conceptual aircraft design and optimisation jointly maintained by CFS Engineering and Airinnova.

The main goal was to extend the `SMTrain` module to enable surrogate-based geometric optimisation. In early-stage aircraft design, exploring large design spaces across multiple flight conditions is essential, yet the direct use of high-fidelity CFD within iterative optimisation loops is often computationally prohibitive. Surrogate models address this limitation by replacing expensive numerical evaluations with fast approximations trained on a limited set of input–output samples.

The implemented workflow allows the user to: (i) define geometric design variables and bounds through the graphical user interface of *CEASIOMpy*; (ii) generate a Design of Experiments (DoE) via Latin Hypercube Sampling (LHS); (iii) run low-fidelity aerodynamic analyses using AVL to build the training dataset; and (iv) automatically train and validate surrogate models based on Kriging (KRG) or Radial Basis Functions (RBF), enabling rapid performance predictions across the explored design domain.

The framework also supports an optional multi-fidelity strategy, in which higher-fidelity simulations (e.g. Euler or RANS with SU2) can be selectively introduced to refine the surrogate model through adaptive sampling procedures.

The multi-fidelity refinement capability was validated on the ONERA M6 wing benchmark, where the discrepancy between low- and high-fidelity aerodynamic predictions is sufficiently significant to justify surrogate correction.

The geometry optimisation workflow was then applied to the winglet of a UAV developed by Dope Hubs, a student association at the University of Genoa participating in the UAS Challenge 2025. Considering the large dimensionality of the design space and the limited computational resources available, the optimisation study was conducted using a single-fidelity approach based on low-fidelity aerodynamic evaluations with AVL, which is suitable for capturing the dominant aerodynamic trends in low-speed flight conditions.

Overall, this thesis extends *CEASIOMpy* by introducing a flexible and GUI-integrated methodology for surrogate-based shape optimisation, consistent with the existing modular architecture and backward compatible with established workflows. The development represents a step towards more automated multi-fidelity optimisation pipelines for both conventional and unconventional aircraft configurations, aligning with the broader COLOSSUS objectives of advancing digital design environments for sustainable and efficient air transport.

Sommario

La presente tesi è stata sviluppata nell'ambito del progetto di ricerca europeo COLOSSUS durante un tirocinio curriculare presso CFS Engineering in collaborazione con l'Università di Genova. Il lavoro si concentra sullo sviluppo e sull'integrazione di nuove funzionalità all'interno di *CEASIOMpy*, un framework open-source per la progettazione concettuale e l'ottimizzazione di aeromobili sviluppato congiuntamente da CFS Engineering e Airinnova.

L'obiettivo principale è stato quello di estendere il modulo `SMTrain` per consentire l'ottimizzazione geometrica basata su modelli surrogati. Nelle fasi preliminari della progettazione aeronautica è infatti necessario esplorare ampi spazi di progetto e valutare numerose configurazioni operative; tuttavia, l'impiego diretto di simulazioni CFD ad alta fedeltà all'interno di cicli iterativi di ottimizzazione risulta spesso computazionalmente proibitivo. I modelli surrogati permettono di superare questa limitazione sostituendo valutazioni numeriche costose con approssimazioni rapide addestrate su un numero limitato di campioni ingresso–uscita.

Il workflow implementato consente all'utente di: (i) definire le variabili geometriche di progetto e i relativi limiti attraverso l'interfaccia grafica di *CEASIOMpy*; (ii) generare un Design of Experiments (DoE) mediante campionamento Latin Hypercube (LHS); (iii) eseguire analisi aerodinamiche a bassa fedeltà con AVL per costruire il dataset di addestramento; e (iv) addestrare e validare automaticamente modelli surrogati basati su Kriging (KRG) o Radial Basis Functions (RBF), permettendo rapide previsioni delle prestazioni aerodinamiche all'interno dello spazio di progetto esplorato.

Il framework supporta inoltre una strategia opzionale multi-fidelity, nella quale simulazioni a più alta fedeltà (ad esempio modelli Euler o RANS tramite SU2) possono essere introdotte selettivamente per raffinare il modello surrogato attraverso procedure di campionamento adattivo.

La capacità di raffinamento multi-fidelity è stata validata sul caso benchmark dell'ala ONERA M6, nel quale la discrepanza tra le predizioni aerodinamiche a bassa e ad alta fedeltà è sufficientemente significativa da giustificare l'applicazione di tecniche di correzione del modello surrogato.

Successivamente, il workflow di ottimizzazione geometrica è stato applicato al winglet di un UAV sviluppato da Dope Hubs, associazione studentesca dell'Università di Genova partecipante alla competizione UAS Challenge 2025. Considerata l'elevata dimensionalità dello spazio di progetto e le risorse computazionali disponibili, lo studio di ottimizzazione è stato condotto utilizzando un approccio single-fidelity basato esclusivamente su valutazioni aerodinamiche a bassa fedeltà mediante AVL, adeguate a catturare le principali tendenze aerodinamiche nel regime di volo a bassa velocità.

Nel complesso, questa tesi estende *CEASIOMpy* introducendo una metodologia flessibile per l'ottimizzazione geometrica basata su modelli surrogati, integrata nell'interfaccia grafica e coerente con l'architettura modulare esistente del framework, mantenendo al contempo la compatibilità con i workflow già consolidati. Lo sviluppo rappresenta un passo verso pipeline di ottimizzazione multi-fidelity più automatizzate per configurazioni aeronautiche convenzionali e non convenzionali, in linea con gli obiettivi più ampi del progetto COLOSSUS di avanzare gli ambienti digitali per la progettazione di sistemi di trasporto aereo più sostenibili ed efficienti.

The work carried out in this thesis was made in the framework of the COLOSSUS project, funded by the European Union under Grant Agreement no 101097120. The Swiss participation in the COLOSSUS project was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00609.

Contents

List of Acronyms

AGILE	Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts
ANOVA	Analysis of Variance
AVL	Athena Vortex Lattice solver
CFD	Computational Fluid Dynamics
CPACS	Common Parametric Aircraft Configuration Schema
DoE	Design of Experiments
DLR	Deutsches Zentrum für Luft- und Raumfahrt
GUI	Graphical User Interface
HF	High Fidelity
KRG	Kriging surrogate model
LF	Low Fidelity
LHS	Latin Hypercube Sampling
LOO	Leave-One-Out cross validation
ML	Machine Learning
MDO	Multidisciplinary Design Optimisation
RANS	Reynolds-Averaged Navier–Stokes
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SALib	Sensitivity Analysis Library
SMT	Surrogate Modeling Toolbox
SU2	Stanford University Unstructured CFD solver
UAV	Unmanned Aerial Vehicle
VLM	Vortex Lattice Method
XML	Extensible Markup Language

1 Introduction

Aircraft design is a complex and highly interdisciplinary process, in which aerodynamic performance plays a central role in determining efficiency, feasibility, and mission capability. From the earliest stages of development, designers are required to make decisions that strongly influence the final configuration of the aircraft, often in the presence of significant uncertainty and under strict time and resource constraints.

In this context, the conceptual design phase represents a critical stage of the design process, during which the overall aircraft configuration, layout, and primary aerodynamic characteristics are first established. Although the analyses performed at this stage rely on simplified models, the associated design choices have long-lasting consequences on performance and cost, while still being relatively inexpensive to revise compared to later stages.

The increasing availability of computational tools has progressively introduced more physics-based analyses into early-stage design. However, the direct use of high-fidelity Computational Fluid Dynamics (CFD) for large-scale design space exploration and optimisation remains impractical due to its high computational cost, particularly when geometric variations and iterative optimisation procedures are involved. This limitation motivates the adoption of alternative strategies that combine adequate modelling fidelity with computational efficiency.

1.1 Motivation and Context

While the conceptual design phase prioritises flexibility and rapid exploration, aerodynamic performance must still be quantified with sufficient accuracy to support meaningful design decisions. Even at an early stage, designers need to understand how geometric choices affect key aerodynamic quantities such as lift, drag, and overall efficiency, especially when a large number of configurations and design variables are considered simultaneously.

High-fidelity CFD solvers can provide detailed predictions of aerodynamic behaviour, but their computational expense makes them unsuitable as the sole tool for extensive design space exploration and optimisation. Evaluating hundreds or thousands of geometric variants with CFD alone is often infeasible within the time and resource constraints of early-stage aircraft design, and this challenge is exacerbated when gradient-free or population-based optimisation algorithms are employed.

Surrogate-based methodologies address this issue by constructing approximate models of the aerodynamic response from a limited set of high-fidelity simulations. Once trained, these models enable rapid evaluations while significantly reducing the reliance on expensive numerical analyses. Multi-fidelity strategies can further enhance this framework by exploiting the complementary strengths of different analysis tools. Low-fidelity solvers can be used to generate large datasets at minimal computational cost, capturing global trends and sensitivities, whereas high-fidelity CFD simulations may be selectively employed to refine the surrogate predictions in regions where additional accuracy is required.

However, the usefulness of multi-fidelity refinement depends strongly on the aerodynamic regime and on the discrepancy between modelling levels. In certain low-speed conceptual design applications, low-fidelity methods may already capture the dominant aerodynamic trends with sufficient accuracy, reducing the practical benefit of introducing higher-fidelity corrections. Consequently, the choice between single- and multi-fidelity workflows must be guided by a balance between modelling accuracy and computational cost.

CEASIOMpy provides a natural environment for the implementation of such methodologies. As a modular framework for conceptual and preliminary aircraft design, it supports parametric geometry definition, automated aerodynamic analysis, and solver coupling within a unified workflow.

1.2 Conceptual Aircraft Design and Early-Stage Optimisation

Aircraft design is commonly structured into successive phases, typically identified as conceptual, preliminary, and detailed design, as shown in Fig. 1.1 [1]. Among these, the conceptual design phase offers the highest degree of design freedom, as it defines the fundamental configuration of the aircraft, including overall geometry, aerodynamic layout, and high-level performance targets.

As discussed in classical aircraft design methodologies, decisions taken during this phase have a disproportionate impact on the final performance, cost, and feasibility of the aircraft, while remaining relatively inexpensive to modify compared to later stages.

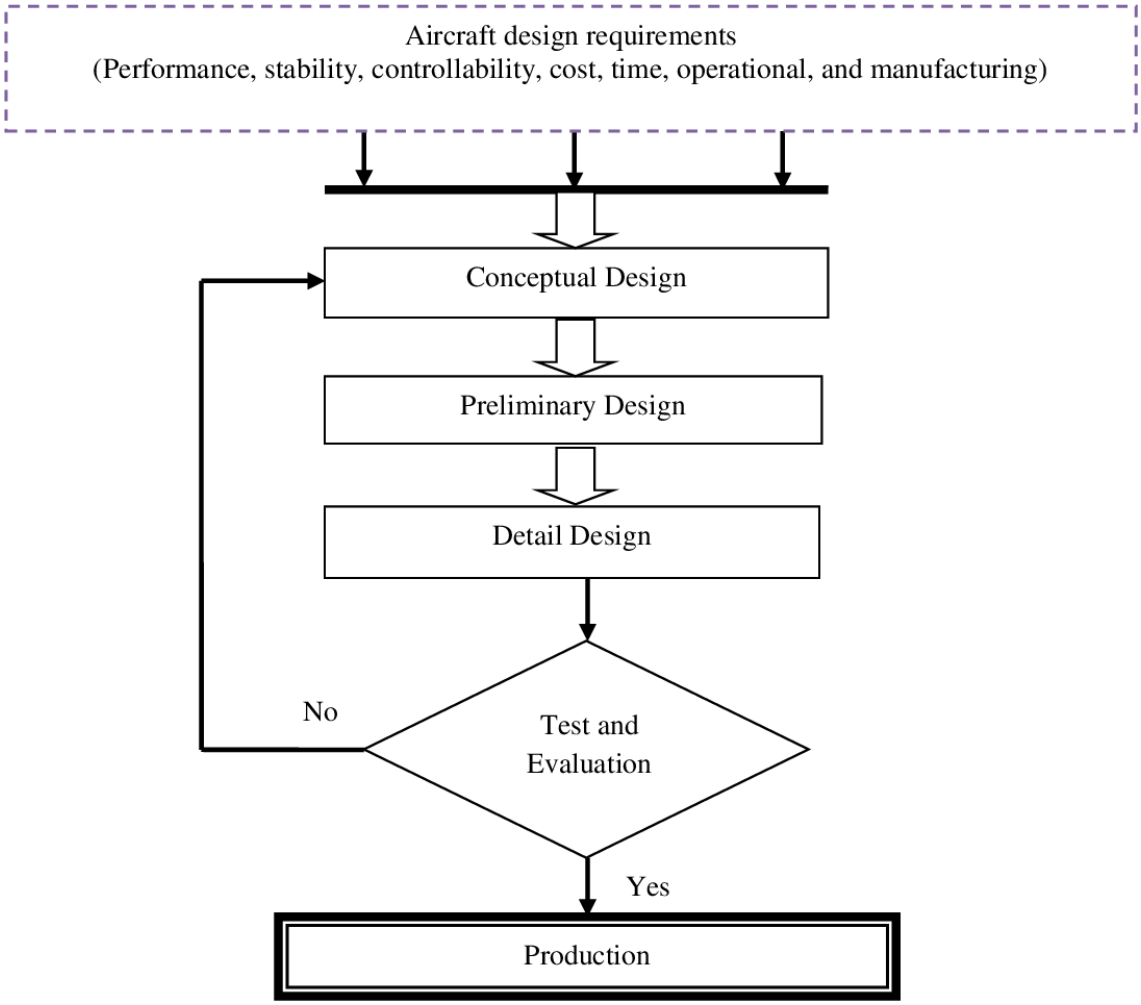


Figure 1.1: Aircraft design workflow from concept definition to detailed engineering.

Rather than pursuing detailed geometric refinement, the goal at this stage is to identify promising design directions and to understand the influence of major parameters on global performance metrics. Consequently, the design process relies on simplified physical models, empirical correlations, and low-fidelity analysis tools that enable rapid evaluation and broad exploration of the configuration space.

In recent years, advances in computational resources and automation have progressively blurred the boundaries between conceptual and preliminary design. Simulation-driven approaches are increasingly adopted at earlier stages, enabling more physics-based analyses without sacrificing efficiency. This evolution has opened the door to optimisation techniques that were traditionally reserved for later design phases, provided that appropriate strategies are employed to control computational cost.

Within this evolving landscape, surrogate-based and multi-fidelity optimisation methods represent a natural extension of the conceptual design philosophy. By combining fast, low-fidelity analyses with selective high-fidelity refinement, these approaches allow designers to retain the exploratory nature of conceptual design while incorporating higher levels of physical accuracy where they are most impactful.

1.3 A reader's guide

This thesis combines theoretical background, software development, and a practical aerospace case study. The document is organised to progressively guide the reader from the fundamental aerodynamic and numerical concepts to the implementation of the surrogate-based optimisation workflow within the *CEASIOMpy* framework, and finally to its application to a real UAV configuration.

Chapter 2 introduces the theoretical background and numerical methods relevant to this work. It discusses the role of Computational Fluid Dynamics in aerodynamic design, presents hierarchical modelling strategies ranging from low-fidelity vortex-lattice methods to high-fidelity Euler and Reynolds–Averaged Navier–Stokes simulations, and introduces machine-learning-based surrogate modelling techniques used for efficient design-space exploration and optimisation.

Chapter 3 presents the *CEASIOMpy* framework and its modular architecture. The chapter describes the CPACS data format used for geometry parametrisation and provides an overview of the main *CEASIOMpy* modules involved in geometry generation, aerodynamic analysis, and data processing.

Chapter 4 describes the surrogate-based optimisation capability implemented during this thesis. It details the developed workflow for geometry exploration, including the generation of design-of-experiments datasets, surrogate model training using RBF, adaptive refinement strategies, and the integration of these capabilities within the *CEASIOMpy* graphical interface. The methodology is validated on the ONERA M6 wing benchmark in order to assess the behaviour of the surrogate modelling framework.

Chapter 5 applies the developed methodology to a practical aerospace design problem involving the optimisation of the winglet geometry of a UAV developed by the Dope Hubs student association for the UAS Challenge 2025 competition. The results of the surrogate-based optimisation are analysed and the resulting aerodynamic trends are discussed.

Finally, Chapter 6 summarises the main contributions of this work, discusses the limitations of the present study, and outlines possible directions for future developments of the surrogate-based optimisation framework.

2 Theoretical Background and Numerical Methods

This chapter presents the theoretical background and numerical methods underlying the aerodynamic analysis and surrogate-based shape optimisation framework developed in this work. The discussion begins with an overview of the role of Computational Fluid Dynamics in aeronautical design and optimisation, highlighting both its predictive capabilities and its computational limitations in iterative design contexts.

Different levels of aerodynamic modelling fidelity are then introduced, ranging from low-fidelity potential-flow methods to high-fidelity Euler and Reynolds-Averaged Navier–Stokes simulations. Particular attention is devoted to the numerical workflow of high-fidelity CFD analyses, including mesh generation, solver strategies, and turbulence modelling, with emphasis on their impact on accuracy, robustness, and computational cost.

Building on this foundation, the chapter motivates the need for hierarchical and multi-fidelity modelling strategies in aerodynamic optimisation. These considerations naturally lead to the introduction of surrogate modelling and machine learning techniques, which are increasingly employed to approximate expensive CFD simulations and enable efficient exploration of large design spaces.

Finally, the general principles of surrogate-based and data-driven optimisation workflows are outlined, providing the theoretical basis for the implementation and numerical experiments presented in the subsequent chapters.

2.1 CFD in Aeronautical Design and Optimisation

Computational Fluid Dynamics (CFD) is a branch of fluid mechanics that employs numerical methods and algorithms to analyse and predict fluid flow behaviour by solving the governing equations of motion under prescribed boundary and initial conditions. In aeronautical engineering, CFD enables the simulation of the interaction between airflow and aircraft surfaces, providing detailed insight into aerodynamic phenomena that are difficult or prohibitively expensive to investigate experimentally [2].

Over the past decades, CFD has evolved from a predominantly academic tool into an indispensable component of both commercial and military aircraft design. Its ability to predict aerodynamic forces and moments, such as lift, drag, and pitching moment, has made it a key enabler for performance assessment and design optimisation [3]. Compared to traditional wind tunnel testing, CFD allows the analysis of a wide range of configurations and operating conditions with reduced cost and increased flexibility, supporting extensive parametric studies and iterative design processes.

The effective application of CFD relies on a structured simulation workflow, which begins with the formulation of the flow problem. This preliminary stage involves defining the objectives of the analysis, the relevant physical phenomena, and the required level of modelling fidelity. Key decisions include the dimensionality of the problem, the choice between steady or unsteady simulations, and the appropriate treatment of viscous and turbulent effects. Based on these considerations, different levels of modelling fidelity can be adopted, ranging from inviscid potential-flow formulations to higher-fidelity Euler and Reynolds-Averaged Navier–Stokes (RANS) equations.

Following problem formulation, the CFD workflow typically consists of geometry definition, mesh generation, physics setup, numerical solution, and post-processing. Although these steps are conceptually distinct, they are strongly interdependent, and decisions made in the early stages directly influence the accuracy, robustness, and computational cost of the entire simulation.

Among these stages, the discretisation of the computational domain into a numerical mesh plays a particularly critical role. The mesh determines how accurately geometric features and flow gradients are resolved and directly impacts numerical stability, solution accuracy, and overall computational requirements. In many practical applications, mesh generation represents one of the most time-consuming and expertise-intensive steps of the CFD process.

At the solution stage, modern CFD solvers exploit parallel computing architectures to distribute the computational workload across multiple processors, significantly reducing wall-clock time and enabling the simulation of complex configurations at realistic Reynolds numbers [4]. Advances in

high-performance computing have therefore made large-scale, high-fidelity simulations increasingly accessible in industrial environments.

However, despite these advances in solver efficiency, the benefits of parallel computing do not fully alleviate the costs associated with pre-processing activities, particularly geometry handling and mesh generation. As a result, the overall turnaround time of a CFD analysis is often dominated by meshing considerations rather than by the numerical solution itself.

For this reason, meshing strategies should not be regarded as a purely technical detail, but rather as a central element in the effective and efficient use of CFD within aerodynamic design and optimisation workflows. The role of mesh generation, its associated trade-offs, and its implications for aerodynamic optimisation are therefore discussed in detail in the following subsection.

Advances in numerical algorithms and high-performance computing have progressively enabled the routine use of high-fidelity CFD in aerodynamic design. RANS simulations, in particular, are now widely employed to capture viscous effects, compressibility, and turbulence phenomena that are essential for accurate prediction of transonic flows, shock–boundary layer interactions, and flow separation [5]. Such capabilities are fundamental in the design of modern aircraft components, where small geometric modifications can lead to significant improvements in aerodynamic efficiency.

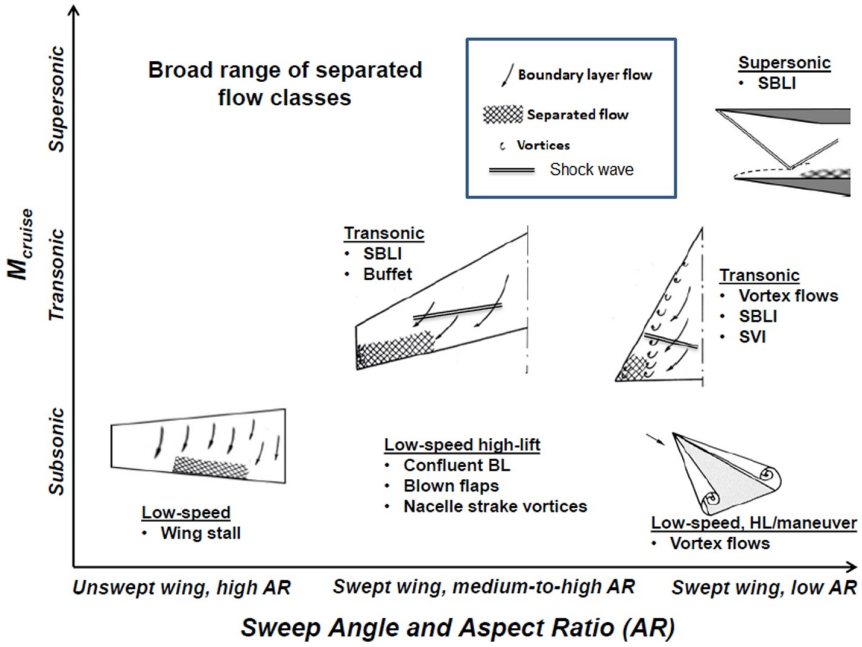


Figure 2.1: Flow separation phenomena relevant to CFD analysis in military aeronautics [5].

Despite these advances, the computational cost associated with high-fidelity CFD remains a major limiting factor, particularly when such simulations are embedded within optimisation loops or large-scale design space explorations. In industrial practice, the adoption of CFD is ultimately governed by a trade-off between its perceived benefits and its overall cost, including not only solution time but also model preparation, mesh generation, and turnaround time [6]. If these costs are allowed to dominate the design cycle, the practical usefulness of CFD is significantly reduced.

This limitation is especially critical during the conceptual and preliminary design phases, where rapid evaluation of a large number of configurations is required. While high-fidelity CFD can provide highly accurate aerodynamic predictions, evaluating hundreds or thousands of geometric variants using CFD alone is often impractical under the time and resource constraints that characterise early-stage design. As a result, the direct use of CFD-based optimisation is generally restricted to later design phases or to a limited number of carefully selected configurations.

Furthermore, as highlighted by Jameson and Fatica [6], in many real-world applications the total cost of a CFD analysis is dominated by pre-processing tasks, such as geometry handling and grid generation, rather than by the numerical solution itself. These setup costs can span weeks or even months for complex configurations, effectively preventing the routine use of CFD in

iterative optimisation workflows. This challenge has historically led to the continued reliance on experimental methods in certain design programmes, despite the potential advantages offered by numerical simulation.

Nevertheless, the potential impact of CFD-driven aerodynamic optimisation is substantial. Even modest improvements in aerodynamic efficiency, such as small increases in lift-to-drag ratio, can translate into significant reductions in fuel consumption and operating costs over the lifetime of an aircraft. Realising these benefits, however, requires moving beyond isolated flow simulations towards systematic aerodynamic shape optimisation and, ultimately, multidisciplinary optimisation frameworks [6].

These considerations motivate the development of hierarchical design strategies in which CFD is employed selectively, rather than exhaustively, as illustrated in Fig. 2.2. Within such frameworks, high-fidelity simulations are used to inform and correct lower-fidelity models, while surrogate-based approaches enable rapid exploration of the design domain. This balance between accuracy and computational efficiency is essential to make CFD-based optimisation viable during the conceptual and preliminary stages of aircraft design.

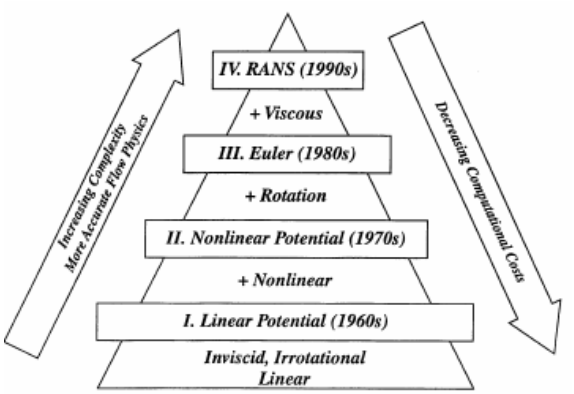


Figure 2.2: Hierarchy of models for industrial flow simulations

2.1.1 Low-Fidelity Aerodynamic Models for Conceptual Design: AVL

Athena Vortex Lattice (AVL) is a widely used low-fidelity aerodynamic analysis tool developed at the Massachusetts Institute of Technology for the aerodynamic and flight-dynamic analysis of rigid aircraft configurations [7]. The software implements an extended vortex lattice formulation combined with potential-flow theory, and is specifically designed for the efficient analysis of lifting configurations.

AVL models lifting surfaces as distributions of bound vortices and enforces flow tangency conditions on the mean camber surface, assuming inviscid, incompressible, and irrotational flow. Under these assumptions, the method provides fast and robust estimates of global aerodynamic quantities such as lift, induced drag, pitching moments, and stability derivatives, which are particularly relevant during early design stages.

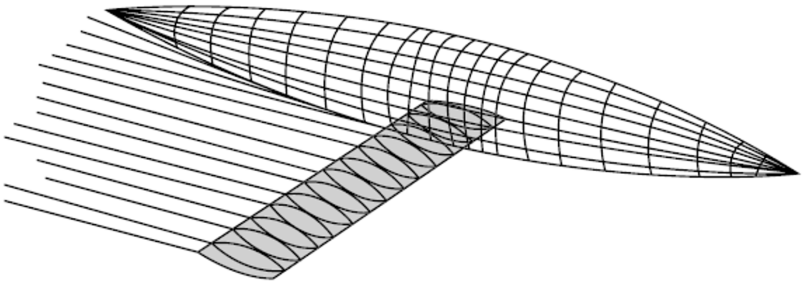


Figure 2.3: Representative paneling for three-dimensional wing vortex lattice.

From a practical standpoint, AVL does not require volumetric mesh generation and operates directly on simplified surface representations of aircraft lifting components, such as wings, tails, and control surfaces. This characteristic, combined with its extremely low computational cost, makes AVL particularly suitable for parametric studies, sensitivity analyses, and preliminary optimisation tasks, where identifying aerodynamic trends and relative performance differences is more important than capturing detailed flow physics.

Despite these advantages, the simplifying assumptions inherent to vortex lattice methods (VLM) limit the predictive capabilities of AVL. Viscous effects, flow separation, shock waves, and compressibility phenomena are not modelled, restricting its applicability to attached-flow conditions and moderate angles of attack. Consequently, AVL results should be interpreted as qualitative or trend-based indicators rather than high-fidelity aerodynamic predictions.

Within the hierarchical modelling framework illustrated in Fig. 2.2, AVL occupies the lowest-fidelity level, enabling rapid exploration of the design space and providing initial aerodynamic estimates that can be progressively refined using higher-fidelity Euler and Reynolds-Averaged Navier–Stokes simulations. This multi-fidelity perspective constitutes a fundamental building block of the surrogate-based optimisation strategy adopted in this work.

2.1.2 Mesh Generation for Aerodynamic CFD Simulations

While low-fidelity methods enable rapid aerodynamic assessments without volumetric discretisation, high-fidelity CFD simulations require an explicit spatial representation of the flow domain through numerical mesh generation.

Mesh generation is a fundamental pre-processing step in Computational Fluid Dynamics, in which the spatial domain of interest is discretised into a finite set of elements over which the governing flow equations are numerically solved [8]. The geometry is decomposed into simple shapes, such as triangles or quadrilaterals in two dimensions, and tetrahedra, prisms, or hexahedra in three dimensions, which are connected through shared faces, edges, or nodes to ensure a continuous representation of the flow domain. The quality and resolution of the resulting mesh play a decisive role in determining both the accuracy of the numerical solution and the computational resources required for the simulation.

From a topological perspective, computational meshes are commonly classified as structured, unstructured, or hybrid. Structured meshes consist of quadrilateral or hexahedral elements with implicit connectivity, offering excellent numerical properties and high accuracy, particularly in boundary-layer regions. However, their generation for complex aircraft geometries is often time-consuming and requires significant manual intervention, as the domain must be decomposed into multiple blocks.

Unstructured meshes, typically composed of triangular or tetrahedral elements with explicitly defined connectivity, provide significantly greater flexibility and automation. These properties make them particularly attractive for complex geometries and for automated simulation workflows. However, purely unstructured meshes may suffer from reduced accuracy in viscous regions due to element skewness and limited control over grid alignment, which can negatively affect Reynolds Averaged Navier–Stokes (RANS) simulations.

To combine the advantages of both approaches, hybrid mesh strategies are widely adopted in industrial practice. In hybrid meshes, the near-wall viscous regions are discretised using prismatic or hexahedral elements to accurately resolve boundary-layer gradients, while the outer flow field is meshed with tetrahedral elements. This approach improves solution accuracy in regions where viscous effects dominate, while keeping the total number of elements and computational cost within practical limits.

Modern CFD solvers often exploit parallel computing architectures to distribute the computational workload across multiple processors, significantly reducing wall-clock time for large-scale simulations. Effective parallelisation places additional constraints on mesh generation, as load balancing and element distribution can influence solver efficiency and scalability, particularly for unstructured and hybrid meshes [9].

Given the importance of viscous effects in aerodynamic design, mesh resolution in the boundary layer is of particular relevance. For wall-resolving RANS approaches, the first grid point normal to the wall must be located within the laminar sublayer, typically corresponding to a non-dimensional wall distance of $y^+ < 1$. Alternatively, when wall-function approaches are employed, larger y^+ values may be acceptable [10].

The non-dimensional wall distance y^+ is defined as

$$y^+ = \frac{u_\tau \, y}{\nu}, \tag{2.1}$$

where y is the distance of the first grid point from the wall, ν is the kinematic viscosity, and $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity, with τ_w denoting the wall shear stress and ρ the fluid density. This parameter represents the local distance from the wall expressed in viscous units and is used to ensure that the near-wall mesh resolution is consistent with the turbulence modelling approach.

Accurate control of boundary-layer mesh parameters is therefore essential to ensure reliable prediction of skin friction, separation, and shock–boundary layer interaction.

Mesh quality assessment represents a critical component of CFD analyses, as the geometric properties of the discretisation directly influence numerical accuracy, stability, and solver convergence. Several metrics are commonly employed to evaluate and optimise mesh quality, particularly in the context of aerodynamic simulations.

One of the most important indicators is the *aspect ratio* (AR), which quantifies the shape quality of mesh elements, especially tetrahedra. It is defined as

$$AR = \frac{l_{\max}}{h_{\min}}, \tag{2.2}$$

where $l_{\max}$ is the maximum edge length of the element and $h_{\min}$ is the minimum height. Lower aspect ratio values are generally associated with improved numerical stability and accuracy.

Orthogonality measures the angular alignment between adjacent mesh elements and is typically evaluated through the angle between the face normal vector $\vec{n}$ and the vector $\vec{s}$ connecting neighbouring cell centres:

$$\cos\theta = \frac{\vec{s} \cdot \vec{n}}{|\vec{s}||\vec{n}|}. \tag{2.3}$$

Higher orthogonality improves discretisation accuracy and enhances solver convergence, particularly for finite-volume formulations.

Another relevant metric is *skewness*, which quantifies the deviation of mesh elements from their ideal geometric shape. A common definition expresses skewness as

$$\text{Skewness} = 90^\circ - \theta_{\min}, \tag{2.4}$$

where $\theta_{\min}$ is the minimum internal angle of the element. Skewness values above approximately 60°–70° are known to cause numerical difficulties and may significantly degrade solution quality.

Additional quality indicators include cell volume consistency and the Jacobian determinant, which ensure correct element mapping and prevent mesh degeneration. Together, these metrics provide a comprehensive assessment of mesh reliability and suitability for high-fidelity CFD

simulations [11].

These metrics are particularly relevant for RANS simulations, where numerical errors introduced by poor mesh quality can significantly affect the prediction of viscous stresses, separation onset, and shock–boundary layer interactions.

It is important to note that increasing mesh density does not necessarily lead to improved solution accuracy beyond a certain threshold. Once the dominant flow features are adequately resolved, further mesh refinement often results in marginal gains in accuracy while causing a disproportionate increase in computational cost [12]. Excessive refinement may therefore reduce the overall efficiency of the simulation without providing meaningful improvements in the predicted aerodynamic quantities.

For this reason, mesh-independence studies are commonly performed in aerodynamic CFD analyses. Such studies aim to verify that key quantities of interest, such as lift, drag, or pressure distributions, remain insensitive to further mesh refinement. Establishing mesh independence is a critical step to ensure that numerical results reflect the underlying physics of the problem rather than artefacts introduced by the discretisation.

The process of grid generation is inherently complex and typically requires dedicated software tools capable of handling complex geometries while maintaining adequate mesh quality. In the present study, mesh generation was performed using the open-source software *GMSH*, which is natively integrated within the *CEASIOMpy* framework. *GMSH* provides automated tools for geometry handling, unstructured volume meshing, and pre- and post-processing, making it well suited for parametric and automated simulation workflows [13]. For RANS simulations, prismatic boundary-layer meshes were generated using *Pentagrow*, which constructs hybrid prismatic–tetrahedral meshes by extruding surface elements normal to solid boundaries [14], as shown in Fig. 2.4.

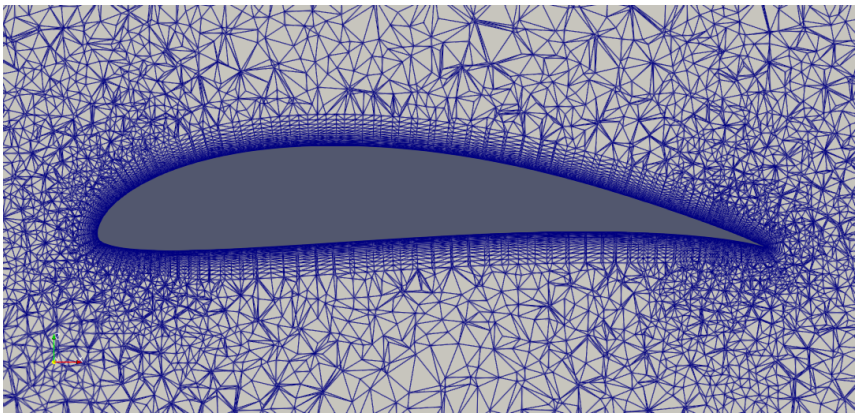


Figure 2.4: Hybrid mesh example created with GMSH and Pentagrow.

This combination enables robust and automated mesh generation while ensuring adequate resolution of near-wall viscous effects.

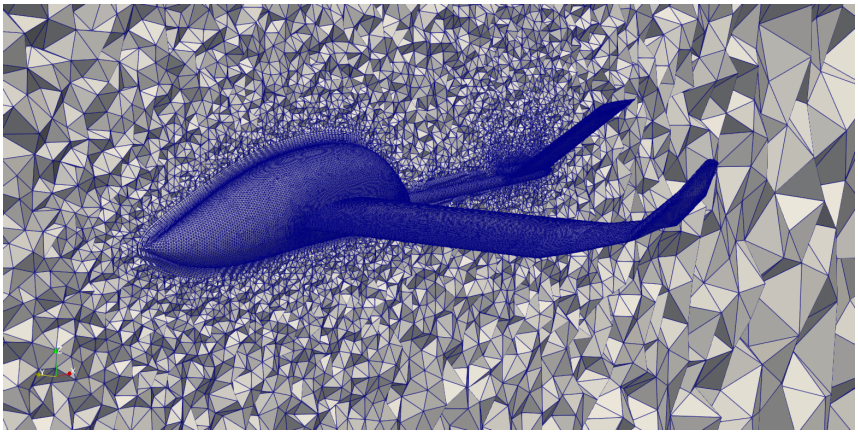


Figure 2.5: RANS mesh example created with GMSH and Pentagrow.

The complexity and cost associated with mesh generation remain one of the primary bottlenecks for the widespread use of high-fidelity CFD in iterative optimisation workflows. Addressing this limitation is essential to enable the effective integration of CFD within surrogate-based and multi-fidelity optimisation frameworks, particularly during the conceptual and preliminary stages of aircraft design.

2.1.3 Mesh-Based Aerodynamic Simulations: Euler and RANS with SU2

Once the computational mesh has been generated, several modelling and numerical choices critically influence the accuracy, robustness, and computational cost of the simulations. These include the selection of the governing equations, turbulence modelling strategy, numerical discretisation schemes, and boundary conditions, all of which must be chosen consistently with the flow regime and the objectives of the analysis.

Inviscid simulations based on the Euler equations neglect viscous effects while retaining compressibility, allowing the prediction of pressure distributions and shock structures at a relatively moderate computational cost.

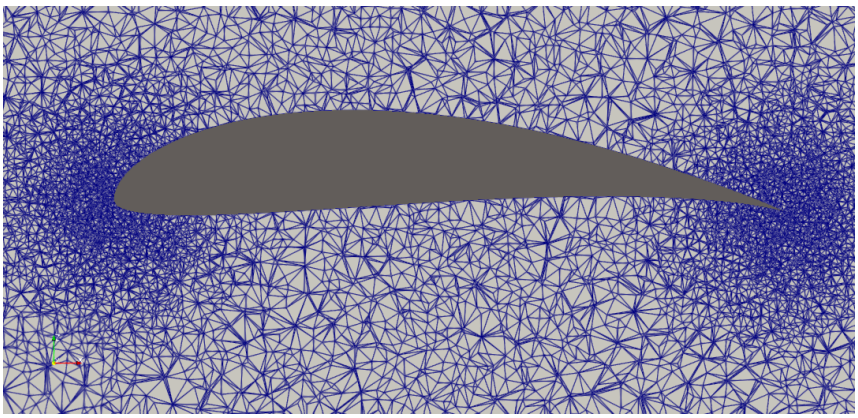


Figure 2.6: Representative wing section of the Euler mesh.

Euler simulations are therefore well suited for capturing global aerodynamic trends and assessing inviscid flow phenomena, providing a useful intermediate level of fidelity between potential-flow methods and fully viscous approaches.

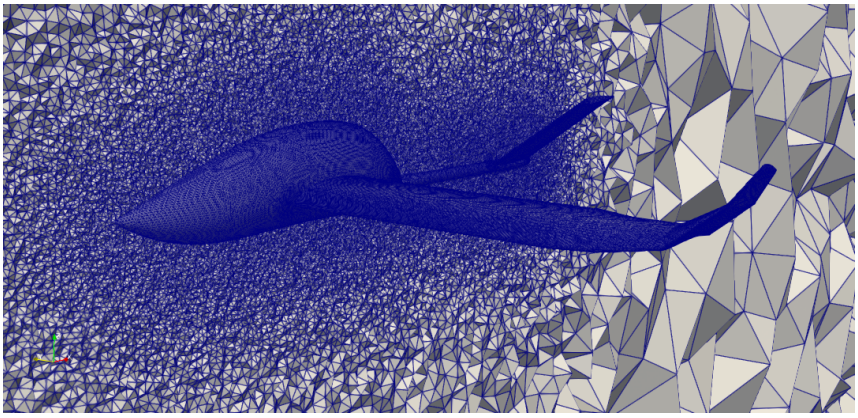


Figure 2.7: Euler mesh example created with GMSH and Pentagrow.

For realistic external aerodynamic flows at high Reynolds numbers, however, viscous effects and turbulence play a dominant role. In such cases, Reynolds-Averaged Navier–Stokes (RANS) simulations are required to model boundary-layer development, flow separation, and shock–boundary layer interactions. While higher-fidelity approaches such as Large Eddy Simulation (LES) or hybrid methods can offer improved physical accuracy, their computational cost remains prohibitive for design-oriented and optimisation workflows. As a result, RANS models represent a practical compromise between modelling fidelity and computational efficiency in industrial aerodynamics.

In the present study, mesh-based simulations were performed using the open-source *SU2* software suite developed at Stanford University. SU2 is a C++-based framework for the numerical solution of partial differential equations and for PDE-constrained optimisation, with a strong focus on aerospace applications [15]. Its finite-volume solvers support compressible Euler and RANS formulations on unstructured and hybrid meshes, making it particularly well suited for automated and parametric simulation workflows.

Euler simulations were conducted on fully unstructured meshes to evaluate inviscid aerodynamic trends at reduced computational cost. RANS simulations, by contrast, were performed on hybrid meshes featuring prismatic layers in the near-wall regions to ensure adequate resolution of boundary-layer gradients.

Turbulence closure for the RANS calculations was provided by the Spalart–Allmaras model, which offers a favourable balance between robustness and efficiency for external aerodynamic flows [16].

All simulations were defined through SU2 configuration files, including the specification of boundary conditions, numerical schemes, turbulence model parameters, and convergence criteria. Solver convergence was monitored by tracking residual histories as well as the stabilisation of integral aerodynamic quantities such as lift and drag coefficients.

Beyond flow solution capabilities, SU2 provides a modular environment that supports mesh adaptation, geometry handling, surface deformation, and post-processing, with solver and model parameters specified through dedicated configuration files, as shown in Fig. 2.8. These features enable detailed analysis of pressure and velocity fields, facilitate solver diagnostics, and support advanced workflows such as gradient-based optimisation.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% SU2 configuration file                                                       %
% Case description: _____________________________________________________     %
% Author: _______________________________________________________________     %
% Institution: __________________________________________________________     %
% Date: __________                                                             %
% File Version 8.4.0 "Harrier"                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ------------ DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION -----------%
%
% Solver type (EULER, NAVIER_STOKES, RANS,
%              INC_EULER, INC_NAVIER_STOKES, INC_RANS,
%              NEMO_EULER, NEMO_NAVIER_STOKES,
%              FEM_EULER, FEM_NAVIER_STOKES, FEM_RANS, FEM_LES,
%              HEAT_EQUATION_FVM, ELASTICITY)
SOLVER= RANS
%
% Specify turbulence model (NONE, SA, SST)
KIND_TURB_MODEL= SA
%
% Specify versions/corrections of the SST model (V2003m, V1994m, VORTICITY, KATO_LAUNDER, UQ, SUSTAINING,
COMPRESSIBILITY-WILCOX, COMPRESSIBILITY-SARKAR, DIMENSIONLESS_LIMIT)
SST_OPTIONS= NONE
%
% Specify versions/corrections of the SA model (NEGATIVE, EDWARDS, WITHFT2, QCR2000, COMPRESSIBILITY, ROTATION,
BCM, EXPERIMENTAL)
SA_OPTIONS= NONE
%
% Transition model (NONE, LM)
KIND_TRANS_MODEL= NONE

% ------------------------------ SOLVER CONTROL ------------------------------%
%
% Number of iterations for single-zone problems
ITER= 1
%
% Maximum number of inner iterations
INNER_ITER= 9999
%
% Min value of the residual (log10 of the residual)
CONV_RESIDUAL_MINVAL= -8
%
% Start convergence criteria at iteration number
CONV_STARTITER= 10
```

Figure 2.8: Example of an SU2 configuration file defining solver, turbulence, and numerical settings.

Within the hierarchical modelling framework illustrated in Fig. 2.2, Euler and RANS simulations occupy intermediate and high-fidelity levels, respectively. They provide physically accurate aerodynamic information that complements low-fidelity models such as AVL and serves as the primary data source for the construction of surrogate models. This structured use of mesh-based CFD simulations forms a cornerstone of the multi-fidelity and machine-learning-driven optimisation strategy developed in the subsequent sections.

2.2 Machine Learning in CFD

In recent years, machine learning (ML) techniques have gained significant attention within the fields of computational fluid dynamics and aerodynamic analysis, driven by the need to reduce the computational cost associated with high-fidelity simulations while preserving acceptable levels of accuracy. Advances in data-driven modelling, together with the increased availability of high-performance computing resources, have enabled ML methods to complement traditional physics-based solvers in a wide range of aerospace applications.

Broadly speaking, the application of machine learning in CFD can be classified into three main categories, depending on the role played by data-driven models with respect to physics-based solvers [17, 18].

The first category comprises approaches in which machine learning is used *within* CFD solvers to enhance, accelerate, or partially replace specific modelling components. Typical examples include data-driven turbulence closures, correction of Reynolds stress tensors, wall-model augmentation, and ML-assisted acceleration of iterative solvers. These methods aim to improve predictive accuracy or computational efficiency while retaining the governing equations as the backbone of the simulation. Although promising, such approaches often require large, high-quality training datasets and tight integration with existing solvers, which can limit their robustness and general applicability.

A second category includes machine learning techniques applied *to* CFD data for flow analysis, model reduction, and physical insight extraction. In this context, ML is used to process high-dimensional simulation data in order to identify dominant flow structures, extract reduced-order representations, or discover governing dynamics. Representative examples include proper orthogonal decomposition (POD) combined with regression models, clustering of flow regimes, and data-driven discovery of dynamical systems. These approaches do not replace CFD simulations, but rather augment their interpretation and facilitate understanding of complex flow phenomena.

The third category, which constitutes the focus of this work, involves the use of machine learning techniques *around* CFD solvers to construct surrogate or emulator models that approximate the input–output behaviour of aerodynamic simulations. In this case, ML models are trained to learn a functional relationship between a set of design variables (e.g. geometric parameters and operating conditions) and aerodynamic quantities of interest, such as lift, drag, or moment coefficients, based on a limited number of CFD evaluations.

Once trained, these surrogate models provide fast predictions at a negligible computational cost compared to full CFD simulations, enabling rapid design space exploration, sensitivity analysis, and optimisation. In this sense, surrogate models do not aim to replace CFD solvers, but rather to complement them by enabling a more efficient and targeted use of high-fidelity simulations within iterative design and optimisation workflows.

From a design perspective, ML-based surrogate modelling is particularly well suited for conceptual and preliminary aircraft design, where simplified representations and reduced levels of fidelity are already accepted as part of the design philosophy. Within a hierarchical modelling framework, surrogate models can be positioned between low-fidelity methods (e.g. vortex lattice models) and high-fidelity Euler or RANS simulations, enabling an effective trade-off between accuracy and computational efficiency.

Furthermore, surrogate models provide a natural foundation for *multi-fidelity* optimisation strategies, in which data from solvers of different fidelity levels are combined to improve predictive performance while limiting overall computational cost [19]. Low-fidelity models enable broad exploration of the design space and capture global aerodynamic trends, whereas high-fidelity CFD simulations are selectively employed to correct and refine surrogate predictions in regions of particular interest.

The integration of machine learning within CFD-driven optimisation workflows therefore represents a key enabler for modern aerodynamic design. By reducing the reliance on computationally expensive high-fidelity simulations and enabling rapid evaluation of candidate designs, ML-based surrogate models allow CFD to be effectively embedded within iterative optimisation loops. This capability forms the foundation of the surrogate-based shape optimisation framework developed in the subsequent chapters of this thesis.

2.2.1 Surrogate Modelling for Aerodynamic Design Optimisation

Surrogate modelling can be interpreted within the broader framework of supervised machine learning. Given a dataset of input–output pairs, the objective is to construct a function that approximates an unknown underlying mapping while minimising a suitable error measure.

Let $\mathbf{x} \in \mathbb{R}^d$ denote a vector of d design variables and let $\mathbf{y} = f_{\text{truth}}(\mathbf{x}) \in \mathbb{R}^m$ represent the vector of aerodynamic responses of interest, such as lift coefficient C_L, drag coefficient C_D, moment coefficients, or other relevant performance indicators. In aerodynamic optimisation problems, m may be equal to one (single-objective optimisation) or larger when multiple quantities are predicted simultaneously.

Given a training dataset

$$\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N},$$

the goal of supervised learning is to identify an approximation function $f_{\text{approx}}(\mathbf{x})$ within a chosen hypothesis class, such that a prescribed loss function is minimised.

Formally, the learning problem can be written as

$$\min_{f_{\text{approx}}} \mathcal{L}(\mathbf{Y}, f_{\text{approx}}(\mathbf{X})), \tag{2.5}$$

where $\mathbf{X}$ denotes the matrix of input samples, $\mathbf{Y}$ the corresponding labels, and $\mathcal{L}$ is a loss function measuring the discrepancy between predictions and true values. In regression problems, the most common choice is the root mean squared error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.6}$$

where y_i denotes the reference value obtained from simulations and $\hat{y}_i$ the corresponding surrogate prediction, for a total of N samples. The RMSE quantifies the average magnitude of the prediction error, with lower values indicating higher surrogate accuracy.

In the context of aerodynamic design, the surrogate model therefore seeks to approximate the mapping between geometric design variables and aerodynamic performance quantities while minimising prediction error over the available dataset.

Training, Validation, and Test Sets To assess generalisation capability, the available dataset is typically partitioned into distinct subsets:

- The *training set* is used to fit the surrogate model.
- The *validation set* is used for model selection and hyperparameter tuning.
- The *test set* provides an unbiased estimate of predictive performance on previously unseen data.

The rationale behind this partitioning is to avoid overfitting, a phenomenon whereby a model fits the training data extremely well but performs poorly on new samples. A common split allocates approximately 60–70% of the data to training, 15–25% to validation, and the remainder to testing [20]. However, in aerodynamic optimisation, high-fidelity CFD data are often scarce. In such data-limited regimes, alternative validation strategies are preferred.

Surrogate Techniques for Aerodynamic Optimisation Among the various regression techniques proposed in the literature, Kriging-based models and RBF interpolants are widely adopted in aerodynamic optimisation [21]. Their popularity stems from their flexibility, their strong performance in data-scarce regimes, and their suitability for adaptive sampling strategies.

These methods are therefore introduced in detail in the following subsections, with particular emphasis on their behaviour in high-dimensional aerodynamic design problems and on their integration within the proposed multi-fidelity framework.

2.2.2 Kriging-Based Surrogate Models

KRG, closely related to Gaussian Process Regression (GPR), is a probabilistic surrogate modelling technique originally developed in geostatistics and later widely adopted in engineering design optimisation [22].

KRG models the unknown function $f(\mathbf{x})$ as the sum of a deterministic trend and a stochastic component represented by a Gaussian random process [23, 24].

KRG is particularly well suited for modelling highly nonlinear responses. Depending on the value of the noise parameter, known as the *nugget*, the model can behave either as an interpolator or as an approximator.

As in other surrogate modelling techniques, the KRG prediction is expressed as a linear combination of the available data points:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i(\mathbf{x})\, f(\mathbf{x}_i), \tag{2.7}$$

where the weights $\lambda_i(\mathbf{x})$ determine the relative influence of each sample point on the prediction at location $\mathbf{x}$.

The main difference with respect to other interpolation approaches lies in the computation of the weights. In KRG, they are obtained by minimizing the estimation variance

$$\hat{\sigma}^2 = \mathrm{var}\big(\hat{f}(\mathbf{x}) - f(\mathbf{x})\big), \tag{2.8}$$

with the constraint

$$\sum_{i=1}^{N} \lambda_i = 1. \tag{2.9}$$

This optimization problem leads to a linear system derived from the properties of the covariance between sampled points:

$$C(f(\mathbf{x}_i), f(\mathbf{x}_j))\, \lambda(\mathbf{x}) = c(f(\mathbf{x}_i), f(\mathbf{x})), \tag{2.10}$$

where C denotes the covariance matrix between training points and c is the covariance vector between the prediction point and the training samples. Since the true covariance structure is not known a priori, it is typically modelled through a *semivariogram* function.

The semivariogram is defined as

$$\gamma(f(\mathbf{x}), f(\mathbf{y})) = \frac{1}{2}\, \mathrm{var}\big(f(\mathbf{x}) - f(\mathbf{y})\big), \tag{2.11}$$

and is commonly approximated by analytic models depending only on the separation distance h. A typical example is the spherical model:

$$\gamma(h) = \begin{cases} 0, & \text{for } h = 0, \\ C_0 + C_1 \left(\frac{3}{2}\frac{h}{R} - \frac{1}{2}\left(\frac{h}{R}\right)^3 \right), & \text{for } 0 < h < R, \\ C_0 + C_1, & \text{for } h \geq R, \end{cases} \tag{2.12}$$

where C_0 is the nugget, C_1 the partial sill, and R the range (i.e., the distance for which $\gamma(h)$ reaches approximately 95% of the sill value $C_0 + C_1$).

In practice, the weights $\lambda_i(\mathbf{x})$ are computed by solving the following linear system:

$$\begin{bmatrix} \gamma(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_1,\mathbf{x}_N) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(\mathbf{x}_N,\mathbf{x}_1) & \cdots & \gamma(\mathbf{x}_N,\mathbf{x}_N) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1(\mathbf{x}) \\ \vdots \\ \lambda_N(\mathbf{x}) \\ \Psi \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{x}_1,\mathbf{x}) \\ \vdots \\ \gamma(\mathbf{x}_N,\mathbf{x}) \\ 1 \end{bmatrix}, \tag{2.13}$$

where Ψ is a Lagrange multiplier used to enforce the unbiasedness constraint. This system can be compactly written as

$$\gamma^* \lambda^* = \gamma_0^*, \tag{2.14}$$

where γ^* is the augmented semivariogram matrix and λ^* the vector of unknown weights and multiplier.

Once the weights have been computed, the KRG predictor is obtained as a weighted sum of the sampled responses, yielding both a mean prediction and an associated prediction variance.

One of the main practical advantages of KRG is that, in addition to the mean prediction, it provides an estimate of the local prediction uncertainty through the posterior variance. This property is particularly important in adaptive refinement and multi-fidelity workflows, since it makes it possible to identify the regions of the design space where the surrogate is less confident. In such regions, additional high-fidelity samples can be introduced in order to improve model accuracy in a targeted and data-efficient manner. For this reason, points associated with high predictive variance are commonly used as candidate refinement points in Kriging-based optimisation strategies.

KRG models are exact interpolants, meaning that they reproduce the training data exactly in the absence of noise. They are therefore well suited for deterministic CFD data, where numerical noise is typically negligible. However, the construction of KRG models involves the solution of a dense linear system whose computational cost scales cubically with the number of training samples, which may limit their applicability for very large datasets.

2.2.3 Radial Basis Function Surrogate Models

RBF surrogate models approximate the target function as a linear combination of radially symmetric basis functions centred at the training points. In contrast to KRG, RBF models are deterministic interpolants and do not provide an explicit measure of prediction uncertainty.

An RBF surrogate model is typically expressed as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \, \phi\left(\|\mathbf{x} - \mathbf{x}_i\|\right), \tag{2.15}$$

where each $\phi(\mathbf{x},\mathbf{c}) = \phi(\|\mathbf{x}-\mathbf{c}\|)$ is a real-valued function depending solely on the distance between the evaluation point $\mathbf{x}$ and the center $\mathbf{c} = \mathbf{x}_i$.

The coefficients λ_i are computed by imposing the interpolation condition

$$\Phi\lambda = y, \quad \text{with} \quad \Phi_{i,j} = \phi\left(\|\mathbf{x}_i - \mathbf{x}_j\|\right), \tag{2.16}$$

where Φ is the interpolation matrix, λ is the vector of unknown weights, and y collects the known responses at the training points.

Different functional forms of the radial basis $\phi(\cdot)$ can be employed, such as Gaussian, multiquadric, inverse multiquadric, or polyharmonic spline functions. For instance, the Gaussian RBF is defined as

$$\phi(r) = \exp(-\beta^2 r^2), \tag{2.17}$$

where $r = \|\mathbf{x} - \mathbf{x}_i\|$ denotes the Euclidean distance and β is a shape parameter that controls the spread of the basis function. The influence of the basis function choice on locality, smoothness, and global behaviour is illustrated in Fig. 2.9.

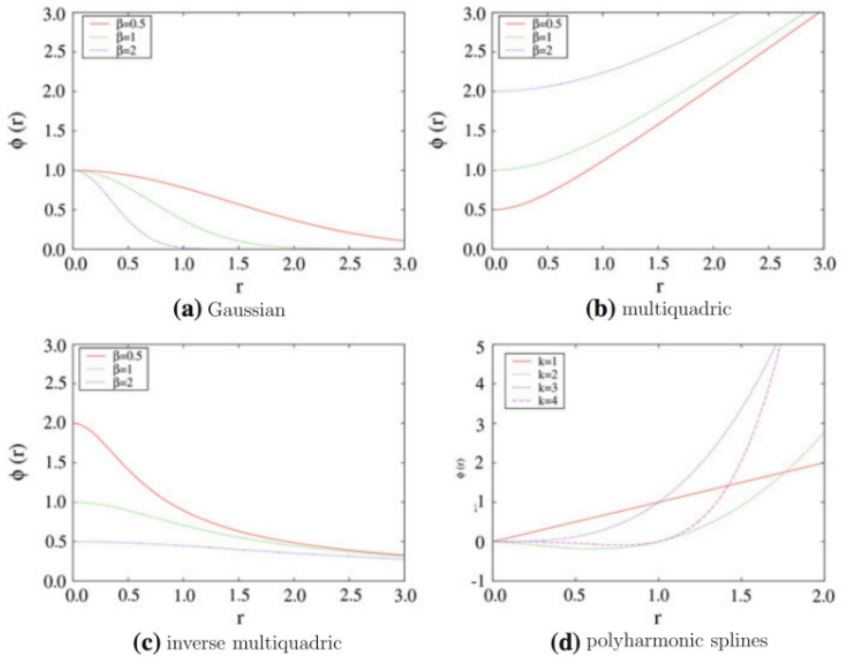


Figure 2.9: Comparison of different radial basis functions commonly employed in RBF surrogate models.

The choice of the basis function and its associated shape parameter significantly influences the smoothness, accuracy, and generalisation capability of the surrogate model.

RBF models are exact interpolants for deterministic data and are generally simpler to implement and train than KRG models. Their computational cost scales more favourably with the number of samples, making them attractive for problems involving moderately large datasets.

In aerodynamic optimisation, RBF surrogates are frequently used due to their robustness, ease of implementation, and good performance in capturing smooth response surfaces. However, the lack of inherent uncertainty quantification can be a limitation in adaptive sampling or multi-fidelity frameworks, where error estimates play an important role.

For this reason, RBF models are often employed alongside KRG models, allowing their respective strengths to be exploited depending on the optimisation problem.

2.2.4 Adaptive Refinement and Model-Specific Sampling Strategies

Beyond providing fast approximations of simulation responses, surrogate models can also support adaptive sampling strategies aimed at improving model accuracy in regions of the design space where the prediction error is expected to be higher.

In the optimisation framework implemented in this work, adaptive refinement strategies are exploited to selectively introduce additional higher-fidelity simulations and progressively improve the surrogate accuracy. The specific refinement strategies adopted for the different surrogate models are described in detail in Chapter 4.

3 CEASIOMpy Framework

CEASIOMpy is a powerful and open-source environment for conceptual aircraft design and optimisation, freely available to the public via GitHub [25]. It has been developed by CFS Engineering [26] in collaboration with Airinnova [27], with the aim of providing a comprehensive and user-friendly platform for the design of both conventional and unconventional aircraft configurations.

The development of *CEASIOMpy* was initiated as part of the AGILE project, funded by the European Commission in 2015, which sought to improve the efficiency and collaboration among teams of experts involved in the aircraft design process. A key objective of the project was to reduce both the time-to-market and the development costs of new aircraft, which remain critical challenges for the aeronautical industry.

One of the core strengths of *CEASIOMpy* is its implementation in Python, a programming language widely adopted in scientific and engineering communities due to its simplicity, versatility, and extensive ecosystem. Its modular architecture allows users to easily customize and extend the framework to accommodate specific design and optimisation requirements, enabling multi-disciplinary analyses and workflows.

CEASIOMpy provides a wide range of tools for the preliminary design and optimisation of aircraft components, including wings, fuselage, and propulsion systems. The environment also supports a graphical user interface (GUI) for intuitive interaction, as well as automated workflows for complex multi-disciplinary studies.

A web-based version of *CEASIOMpy* has also been developed, providing browser-based access to the framework and lowering the barrier to entry for new users.

Overall, *CEASIOMpy* represents a significant advancement in conceptual aircraft design, combining an open-source and collaborative framework with powerful analytical capabilities and a user-friendly interface.

3.1 CPACS files

All geometrical input data in *CEASIOMpy* are based on the open-standard CPACS (Common Parametric Aircraft Configuration Schema), developed by the German Aerospace Center (DLR) [28]. CPACS is an XML-based data exchange format used in the aerospace industry to store and share information related to aircraft design and configuration. It is widely adopted by aircraft manufacturers, suppliers, research institutes, and regulatory bodies, providing a common and standardised representation of aircraft data.

One of the main advantages of CPACS lies in its ability to consistently describe geometry, structural layout, aerodynamic properties, and other key characteristics of an aircraft configuration. This standardisation enables efficient information exchange between different tools and organisations, significantly improving collaboration during the aircraft design and development process.

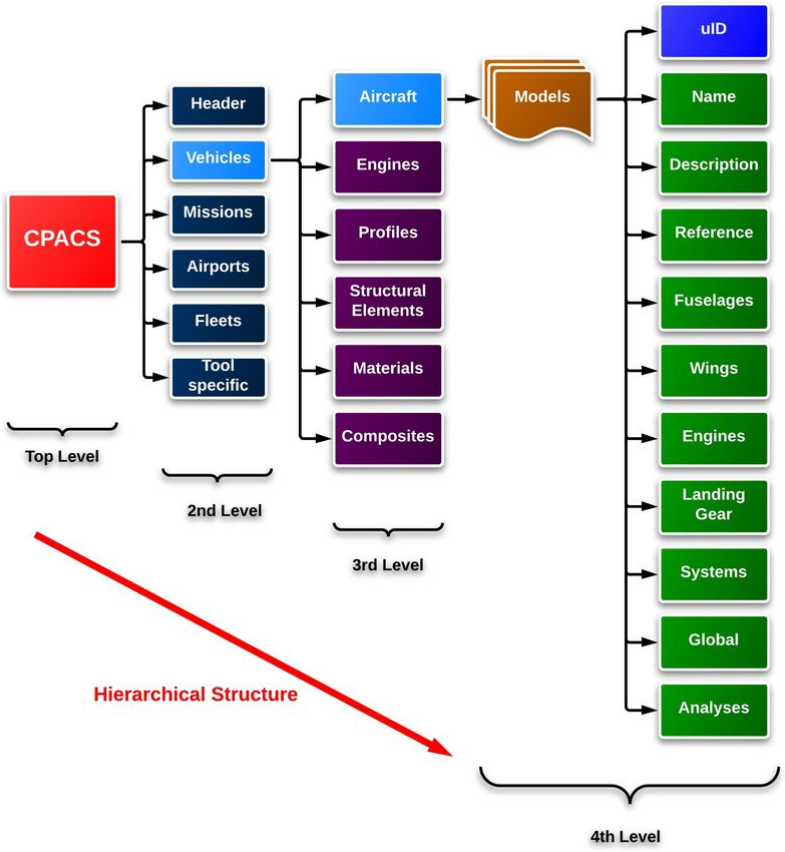


Figure 3.1: CPACS XML file hierarchical structure.

Within CPACS, the aircraft geometry is organised hierarchically. Starting from the root of the CPACS file, the chain follows `CPACS` → `VEHICLES` → `AIRCRAFT` → `MODELS` (see Fig. 3.1). At this level, the individual *bodies* of the aircraft—such as fuselage, wings, nacelles, and propeller disks—are defined. Each body is further represented as a set of sections (see Fig. 3.2), with parameters associated to each section specifying its geometry, positioning, and other relevant properties. This approach allows for fine-grained parametric control over every part of the aircraft, enabling the user to modify, add, or remove sections according to the needs of the design study.

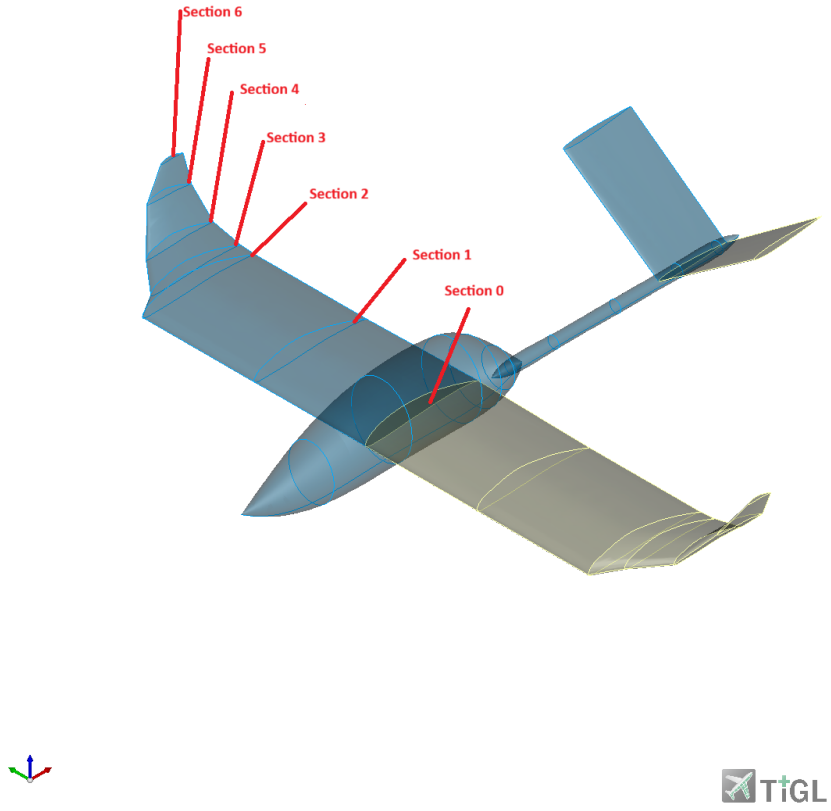


Figure 3.2: Wing Sections.

Through this representation, CPACS provides a flexible and consistent way to manage complex aircraft configurations, supporting rapid design iterations and facilitating multi-disciplinary analyses. The structured hierarchy of bodies and sections ensures that modifications at the component level propagate correctly throughout the aircraft model, preserving both geometric and parametric consistency.

3.2 *CEASIOMpy* Modules

CEASIOMpy is structured around a modular architecture, where individual modules can be combined to build customised design and optimisation workflows. Each module addresses a specific aspect of the aircraft design process, allowing for flexibility, scalability, and multi-disciplinary integration. The currently available modules can be grouped according to their primary purpose, as described below.

Geometry and Mesh Modules Geometry and mesh generation form the foundation of the *CEASIOMpy* workflow, as all analyses rely on a consistent parametric description of the aircraft. This group includes the `CPACSCreator` and `CPACSUpdater` modules, which enable the creation and modification of aircraft configurations using the CPACS standard. These tools allow users to parametrically define and update aircraft components such as wings, fuselage, nacelles, and control surfaces.

Mesh generation is handled by the `CPACS2GMSH` module, which converts CPACS geometries into computational meshes using the GMSH library. The resulting meshes can then be used for aerodynamic and structural analyses, ensuring a seamless transition between geometry definition and simulation.

Aerodynamics Modules The aerodynamics modules provide tools for estimating aerodynamic performance at different levels of fidelity. Low-fidelity aerodynamic analyses are performed using the

`PyAVL` module, which interfaces with AVL to compute aerodynamic coefficients efficiently, making it particularly suitable for early-stage design and parametric studies.

Higher-fidelity analyses are enabled through the `SU2Run` module, which integrates the SU2 CFD solver for detailed flow simulations.

Meta Modules Meta modules are designed to operate on top of simulation data and other *CEASIOMpy* modules, enabling advanced data-driven workflows. The `SMTrain` module belongs to this category and is used for surrogate model training. It facilitates the construction of surrogate models based on simulation data, supporting tasks such as design space exploration, optimisation, and multi-fidelity modelling.

Mission Analysis Modules Mission analysis modules aim to evaluate aircraft performance over prescribed flight conditions and trajectories. The `StaticStability` module provides tools to assess longitudinal and lateral static stability characteristics. The `DynamicStability` module, while currently under development, is intended to extend these capabilities to dynamic stability analysis, enabling a more comprehensive evaluation of aircraft handling qualities.

Structural Modules Structural analysis within *CEASIOMpy* is addressed by the `AeroFrame` module, which provides preliminary structural assessments by coupling aerodynamic loads with simplified structural models. This allows for early-stage evaluations of load paths and structural feasibility, which are essential during conceptual design.

Data Analysis and Storage Modules To support data management and post-processing, *CEASIOMpy* includes modules dedicated to data analysis and storage. The `Database` module is intended to store and organise simulation results and design data, facilitating traceability and comparison across different design iterations. Although still under development, this functionality is crucial for large-scale parametric and optimisation studies.

Overall, the modular structure of *CEASIOMpy* enables the construction of flexible and reusable workflows, where information is exchanged consistently between modules through the CPACS data format. This design philosophy allows engineers and researchers to tailor the framework to specific design objectives while maintaining interoperability between different analysis disciplines.

4 Surrogate-Based Shape Optimisation in *CEASIOMpy*

This chapter describes the surrogate-based optimisation capability developed and integrated within the *CEASIOMpy* framework. The focus is on the implemented workflow and on the design choices made to enable efficient *geometry exploration* and progressive model refinement.

It is important to emphasise that the workflow implemented in this thesis supports both single-level and two-level modelling strategies. In single-level mode, surrogate models are trained exclusively on low-fidelity aerodynamic data generated with AVL. In two-level mode, a limited number of higher-fidelity simulations performed with SU2 can be introduced to refine the surrogate through adaptive sampling.

In practice, the adoption of multi-fidelity refinement depends on the expected discrepancy between modelling levels and on the available computational resources. In some aerodynamic regimes, particularly at low Mach numbers and moderate angles of attack, low-fidelity vortex-lattice methods may already capture the dominant aerodynamic trends with reasonable accuracy. In such cases, introducing a limited number of high-fidelity simulations may not significantly improve the predictive capability of the surrogate model.

4.1 Overview of the Implemented Workflow

The workflow is configured through the *CEASIOMpy* graphical interface, implemented in Streamlit. The GUI acts as the orchestration layer between CPACS parametrisation, solver execution, surrogate training, and optional refinement.

The main workflow page (Fig. 4.1) allows the user to: (i) select the CPACS file, (ii) choose the simulation purpose, (iii) define the fidelity level, (iv) configure surrogate settings, and (v) launch the automated optimisation process.

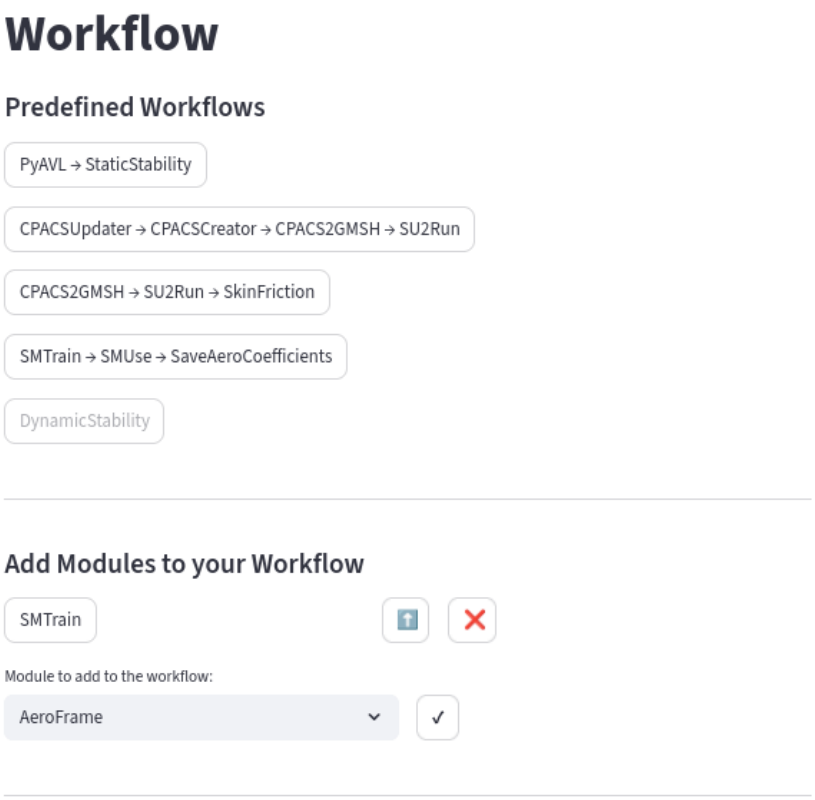


Figure 4.1: Workflow page of CEASIOMpy GUI.

At runtime, the module reads: (i) the selected aeromap UID (flight condition set), (ii) the fidelity level (*One Level* or *Two Level*), (iii) the train/validation/test split ratio, (iv) the objective (e.g. C_L, C_D, C_L/C_D), (v) the RMSE target (used as a stopping criterion during refinement), (vi) the simulation purpose (*Flight Condition Exploration* or *Geometry Exploration*), (vii) whether to run new simulations or load an existing dataset, and (viii) the selected surrogate model(s): KRG, RBF, or both.

Two main branches exist:

- **Flight condition exploration:** design variables are aeromap parameters (e.g. altitude, Mach, angle of attack, etc.). A LHS is created in that space, AVL is run for the sampled conditions, and the surrogate is trained.
- **Geometry exploration (focus of this thesis):** design variables are geometric parameters selected in the GUI for specific wing(s) and section(s). An LHS is created in the space of those geometric parameters; for each sample, a new CPACS is generated by updating the corresponding XML entries; AVL is executed to produce the low-fidelity dataset; the surrogate is trained and optionally refined.

The flight condition settings are defined through the aeromap configuration panel (Fig. 4.2). The user specifies altitude and velocity, from which the Mach number and dynamic pressure are derived. These settings determine the aerodynamic operating point for both AVL and SU2 simulations.

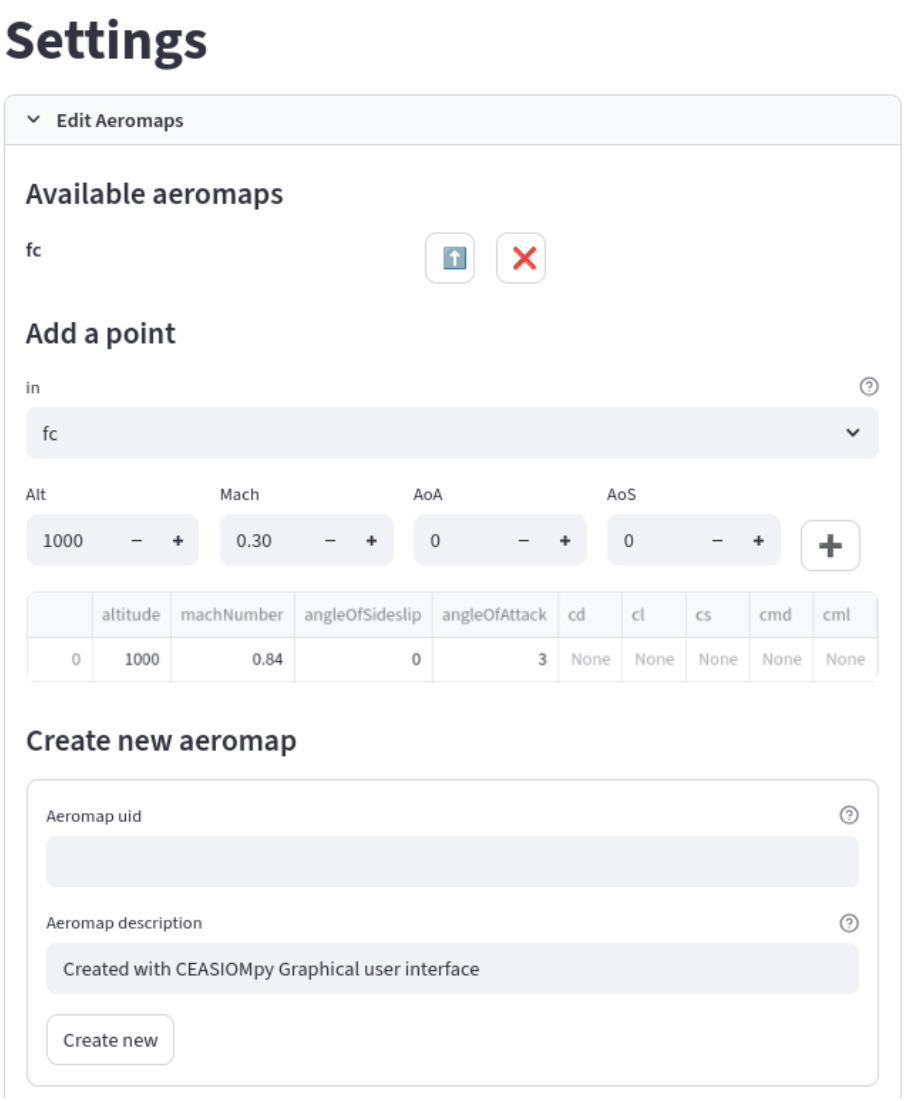


Figure 4.2: Aeromap configuration in the GUI.

The remainder of this chapter focuses on the *Geometry Exploration* path, since it is the main extension implemented during this work.

4.2 Geometry Exploration: Dataset Generation from CPACS

4.2.1 Selection of geometric variables and bounds

The geometric parametrisation is defined interactively in the SMTrain interface (Fig. 4.3), where the user selects the wing(s), section(s), and parameters to be modified, together with their admissible bounds.

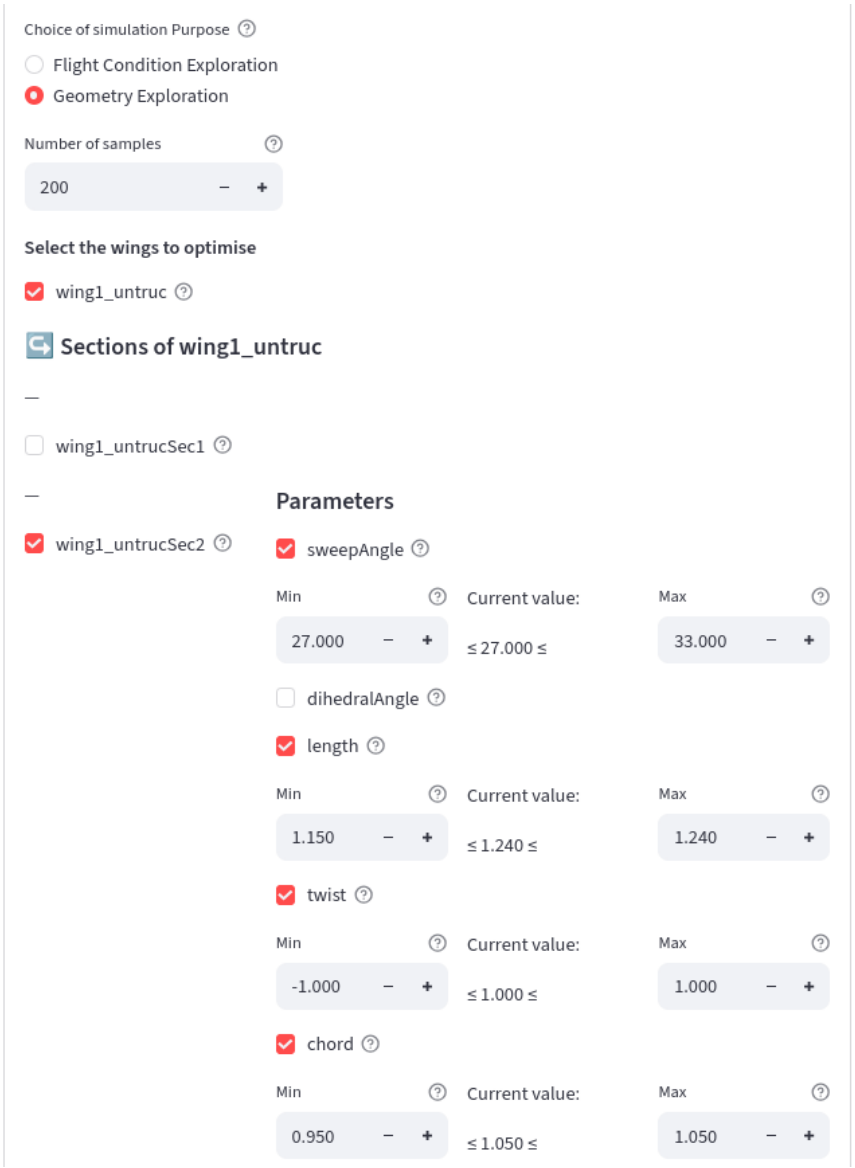


Figure 4.3: Geometric parameter selection in SMTrain.

Each parameter is internally represented with a unique name encoding parameter, section UID, and wing UID according to the convention:

`{param}_of_{sectionUID}_of_{wingUID}`.

This naming convention ensures traceability from numerical design variables to their corresponding CPACS locations.

4.2.2 Sampling and CPACS generation

A space-filling LHS is generated within the user-defined bounds to obtain a set of N geometric configurations. For each sampled configuration:

1. the baseline CPACS is copied to a new XML file,
2. the selected geometric parameters are updated via their CPACS XPath (section transformation nodes and, when required, wing positioning nodes),
3. the resulting CPACS file is stored and used as the input geometry for the aerodynamic analysis.

This procedure produces a list of consistent CPACS geometries that can be evaluated in batch, while keeping a one-to-one association between sampled design vectors and CPACS files.

4.3 First-Level Training: Low-Fidelity Surrogate Construction

4.3.1 Low-fidelity solver and dataset assembly

Low-fidelity aerodynamic evaluations are generated using AVL (VLM). For each sampled CPACS geometry, AVL computes the aerodynamic quantities required to build the training set. The results are assembled into a tabular dataset, where each row corresponds to one evaluated configuration and columns contain the selected geometric parameters plus the objective.

Before training:

- unsuccessful/invalid samples are filtered out (e.g. objective equal to zero),
- constant columns are removed to avoid ill-conditioned training,
- the inputs are normalised using a standard score normalisation ($x \leftarrow (x - \mu)/\sigma$), and
- the dataset is split into training and validation/test subsets according to the user-defined ratio.

4.3.2 Training of KRG and RBF models

The module supports training:

- **KRG:** a Gaussian-process-based surrogate, which provides both mean prediction and an estimate of local uncertainty (posterior variance).
- **RBF:** a deterministic interpolant based on radial basis functions. In contrast to KRG, it does not provide intrinsic uncertainty estimates.

The surrogate configuration is defined in the GUI through the dedicated settings panel (Fig. 4.4), where the user selects the surrogate type (KRG, RBF, or both), the train/validation/test split ratio, and other model-specific parameters.

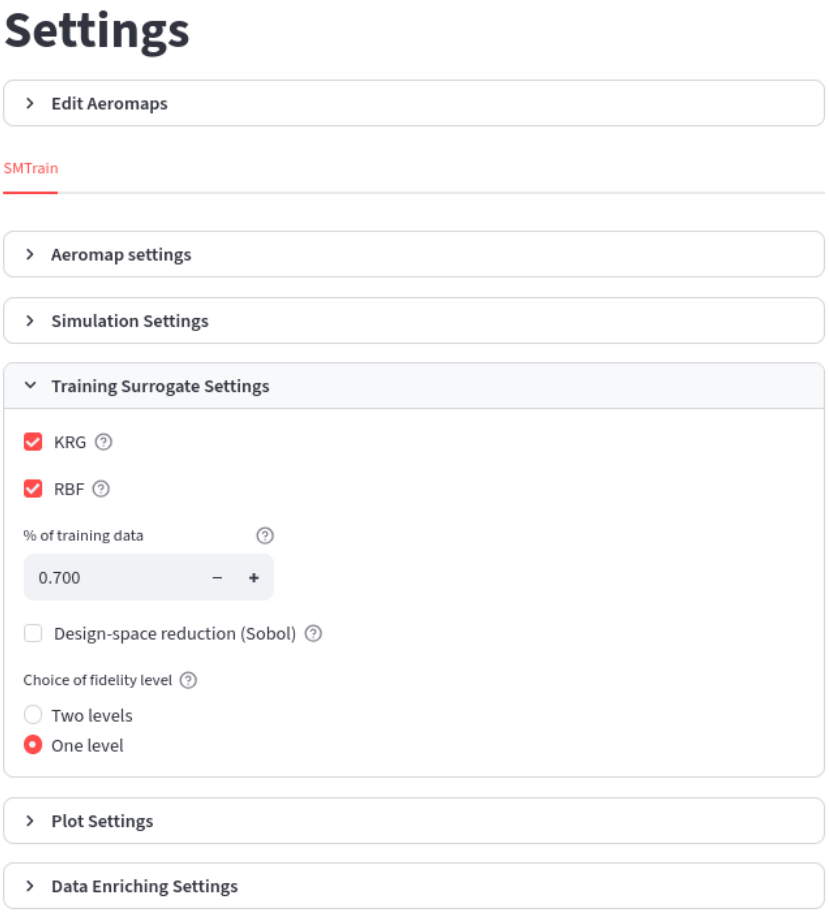


Figure 4.4: Surrogate model settings.

When both models are selected, they are trained independently on the same normalised dataset. Model performance is assessed on the test set using the RMSE metric, and validation plots can be generated and stored if requested by the user.

4.3.3 Design Space Screening via Sobol Sensitivity Analysis

In addition to surrogate training, the workflow optionally includes a variance-based global sensitivity analysis aimed at reducing the dimensionality of the design space prior to optimisation [29]. This feature can be activated by the user through the GUI.

Variance-based sensitivity analysis Let $\hat{f}(\mathbf{x})$ denote the trained surrogate model, with $\mathbf{x} = (x_1, \ldots, x_d)$ representing the d design variables. Assuming independent inputs, the model output variance can be decomposed according to the functional Analysis of Variance (ANOVA) representation [30]:

$$\mathrm{Var}(\hat{f}) = \sum_{i=1}^{d} V_i + \sum_{i<j} V_{ij} + \cdots + V_{1\ldots d}, \tag{4.1}$$

where:

- $V_i = \mathrm{Var}_{x_i}\left(\mathbb{E}[\hat{f} \mid x_i]\right)$ represents the contribution of parameter x_i alone,
- V_{ij} and higher-order terms represent interaction effects.

The *first-order Sobol index* is defined as

$$S_i = \frac{V_i}{\mathrm{Var}(\hat{f})}, \tag{4.2}$$

and quantifies the isolated contribution of x_i.

The *total-effect index* is defined as

$$S_{T,i} = 1 - \frac{\mathrm{Var}_{\mathbf{x}_{\sim i}}\left(\mathbb{E}[\hat{f} \mid \mathbf{x}_{\sim i}]\right)}{\mathrm{Var}(\hat{f})}, \tag{4.3}$$

where $\mathbf{x}_{\sim i}$ denotes all variables except x_i. The total-effect index measures the overall influence of x_i, including all interaction terms.

In surrogate-based optimisation, the total-effect index $S_{T,i}$ is used as screening criterion, since it captures the full contribution of each design variable.

Saltelli sampling implementation Sobol indices are estimated using the Saltelli sampling scheme, as implemented in the `SALib` library. This approach is based on quasi-random low-discrepancy sequences (Sobol sequences) and provides an efficient Monte Carlo estimator of both first-order and total-effect indices.

Given d parameters and a base sample size N, the Saltelli method requires $N(2d + 2)$ model evaluations. In the present workflow, these evaluations are performed on the surrogate model $\hat{f}$ rather than on the CFD solver, making the sensitivity analysis computationally negligible.

Screening strategy The implemented screening procedure follows these steps:

1. Train the surrogate model on the current design space.
2. Generate Saltelli samples in the normalised parameter space.
3. Evaluate the surrogate at all Saltelli sample points.
4. Estimate total-effect indices $S_{T,i}$.
5. Identify parameters satisfying
$$S_{T,i} < 0.05 \cdot \max_j S_{T,j}.$$
6. If such parameters exist, remove them from the design space, regenerate the DoE, and retrain the surrogate in the reduced-dimensional space.

The threshold corresponds to eliminating parameters whose total contribution to output variance is less than 5% of the most influential variable.

Practical implications This screening stage improves sampling density, surrogate stability, and optimisation efficiency, especially in moderately high-dimensional problems. If no negligible parameters are detected, the workflow proceeds directly to artefact saving (single-level mode) or to the refinement stage (two-level mode).

4.3.4 Saving artefacts and best configurations

At the end of the first-level training, the workflow stores:

- the trained surrogate model(s),
- the parameter order used during training,
- the normalisation parameters (μ, σ) required for consistent inference,
- a CSV file containing the best configuration among the evaluated DoE points, and

- a surrogate-based optimum found by global optimisation on the surrogate model.

The surrogate-based optimum is obtained by running a global optimiser (differential evolution) directly on the surrogate in the *normalised space*, with bounds derived from the explored dataset. The resulting optimal design vector is then de-normalised and written to: (i) a CSV summary (parameters and predicted objective) and (ii) a CPACS file where the geometry is updated consistently with the parameter values.

4.4 Two-Level Mode: Progressive Refinement with Higher-Fidelity Samples

The fidelity configuration is controlled via the GUI (Fig. 4.5). When the *One Level* option is selected, only AVL simulations are executed. When *Two Level* is activated, the workflow automatically integrates SU2 (Euler or RANS) for higher-fidelity evaluations.

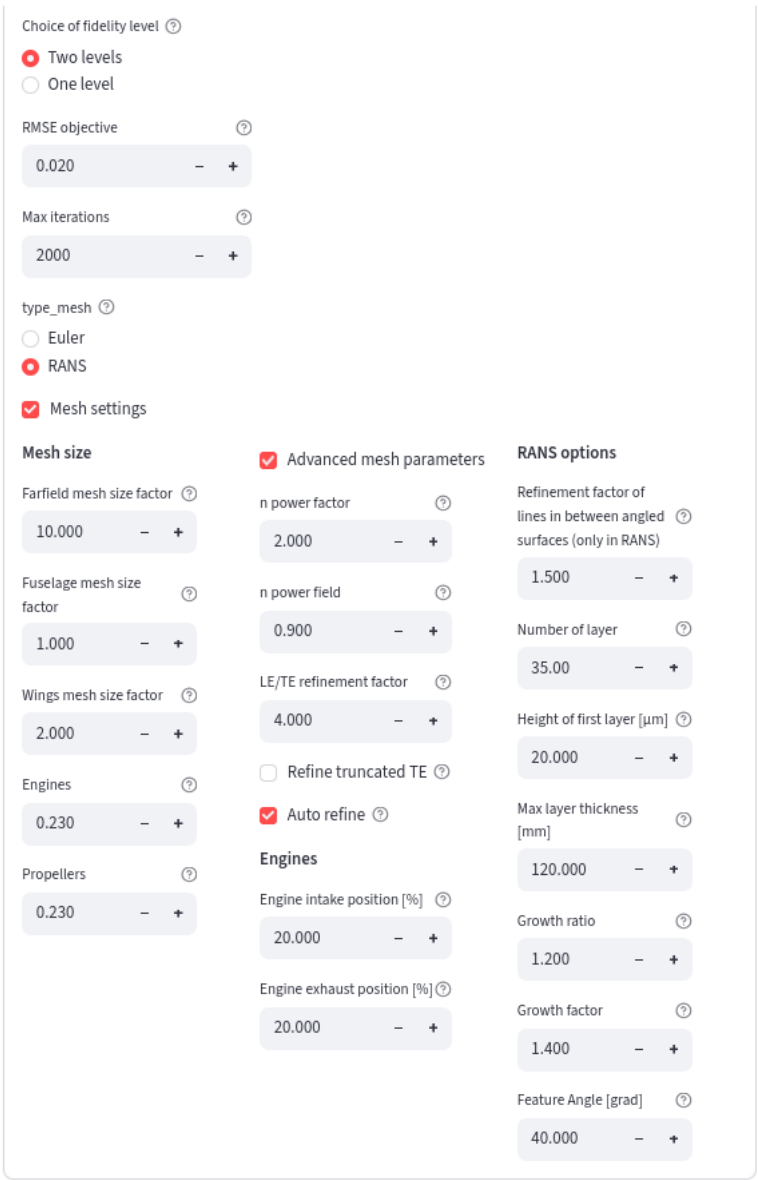


Figure 4.5: Fidelity level selection in SMTrain.

4.4.1 When "multi-fidelity" applies in this thesis

The two-level mode is enabled only when the user selects *Two Level*. In that case, the workflow starts from a low-fidelity surrogate trained on AVL data and progressively improves it by adding a limited number of higher-fidelity samples computed with SU2 (Euler or RANS, depending on the setup). When *One Level* is selected, there is no higher-fidelity dataset and the process remains a purely low-fidelity surrogate training.

Therefore, general definitions of multi-fidelity methods that assume an existing, separate high-fidelity model are only partially applicable here: in this thesis, the high-fidelity level is *optional* and is introduced *selectively* to refine the surrogate when requested by the user and when computational resources allow.

4.4.2 Model fusion for Kriging (SMT multi-fidelity)

For KRG-based refinement, the implementation uses the multi-fidelity capabilities available in the *SMT* library [31]. The refinement loop proceeds as follows:

1. Start from the first-level (AVL) training set and the trained KRG model.
2. Identify candidate refinement locations using the predictive variance of the KRG model.
3. For the selected point(s), generate updated CPACS geometries and run SU2 to obtain higher-fidelity objective values.
4. Augment the higher-fidelity dataset and retrain a two-level surrogate that combines low- and high-fidelity information.
5. Repeat until the target accuracy is reached or a maximum number of iterations is exhausted.

A key practical requirement in the implemented two-level KRG workflow is *consistency between fidelity levels*.

In the SMT implementation, the multi-fidelity Kriging model follows an autoregressive formulation in which the high-fidelity response is expressed as a scaled version of the low-fidelity prediction plus a correction term:

$$f_H(\mathbf{x}) = \rho\, f_L(\mathbf{x}) + \delta(\mathbf{x}),$$

where ρ is a scaling coefficient and $\delta(\mathbf{x})$ represents a Gaussian-process correction trained on the high-fidelity data.

In particular, higher-fidelity evaluations are generated from candidate points derived from the sampled/normalised input space, allowing coherent pairing between levels when building the two-level training sets.

4.4.3 Adaptive refinement for RBF via Leave-One-Out error

RBF models do not provide a native uncertainty estimate. For this reason, the refinement strategy implemented for RBF relies on a data-driven error indicator based on Leave-One-Out (LOO) cross-validation.

The LOO error provides a practical proxy for the local prediction accuracy of the surrogate and allows identifying regions where the model interpolation is less reliable.

Given the current training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, the LOO error at point i is computed by re-fitting the surrogate without that point and measuring the absolute prediction error:

$$e_i^{\mathrm{LOO}} = \left| y_i - \hat{f}^{(-i)}(\mathbf{x}_i) \right|.$$

The refinement procedure then:

1. computes e_i^{LOO} over all training samples,
2. identifies a subset of worst-predicted points (largest LOO errors),

3. generates new candidate samples by applying small local perturbations around those points, while avoiding duplicates across iterations,

4. evaluates these new samples with SU2 to obtain higher-fidelity objective values, and

5. retrains the model including the additional higher-fidelity information.

This approach makes refinement possible also for deterministic surrogates without uncertainty quantification, at the price of additional surrogate re-fits during the LOO computation.

4.4.4 Stopping criterion and practical considerations

Refinement is driven by a user-defined RMSE target, interpreted as the desired surrogate accuracy. As discussed in Section 2.2.1, the RMSE is used here as a global indicator of predictive performance and as the stopping criterion for the refinement loop.

In practice, the refinement process is terminated when the updated surrogate model reaches an RMSE lower than the user-defined *RMSE objective*. If the target is not achieved after a refinement iteration, new candidate points are generated and additional high-fidelity simulations are performed. The surrogate is therefore progressively enriched until the desired accuracy level is reached or the maximum number of refinement iterations is exhausted.

It is worth emphasising that the two-level mode is designed to be *data-efficient*: the low-fidelity dataset provides broad coverage of the design space, while SU2 is called only for a limited number of configurations selected by the refinement criteria. This balance is crucial to make geometry exploration feasible within conceptual and preliminary design constraints.

The effectiveness of this progressive refinement strategy is analysed in the validation chapter, where different multi-fidelity refinement scenarios are compared in terms of accuracy improvement and number of additional high-fidelity simulations required.

4.5 Loading Existing Datasets and Continuing Refinement

In addition to running new low-fidelity simulations, the module supports loading an existing geometry-exploration dataset (AVL results). In this case, the workflow:

- imports the stored results,
- performs normalisation and train/validation/test split,
- trains the selected surrogate model(s), and
- optionally applies the two-level refinement procedure (SU2) starting from the loaded dataset.

This option enables reusing previous computational campaigns and extending them with additional refinement without restarting the full DoE process.

4.6 Results and Interactive Exploration

After execution, the GUI provides a dedicated Results page that supports post-processing and design exploration.

The interface highlights: (i) the best configuration among the evaluated DoE samples, (ii) the surrogate-predicted optimum within the admissible bounds.

In addition, a global sensitivity analysis based on Sobol indices is available (Fig. 4.6), allowing identification of the most influential geometric variables.

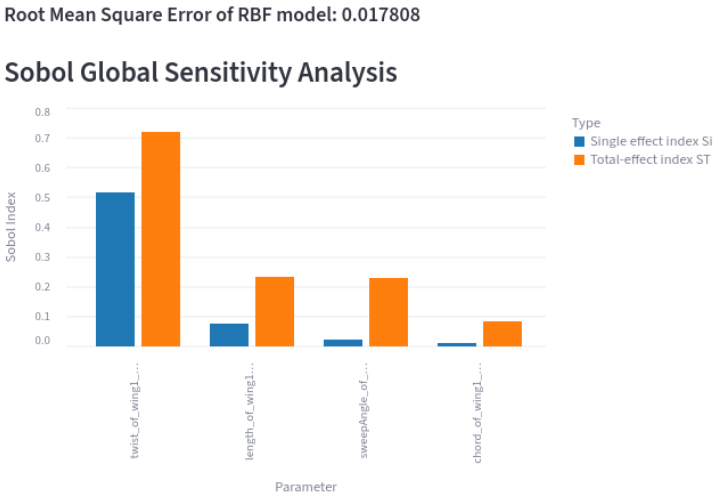


Figure 4.6: Results page: Sobol analysis.

The surrogate response surface can be visualised through two-dimensional parameter slices (Fig. 4.7), facilitating qualitative interpretation of nonlinear trends.

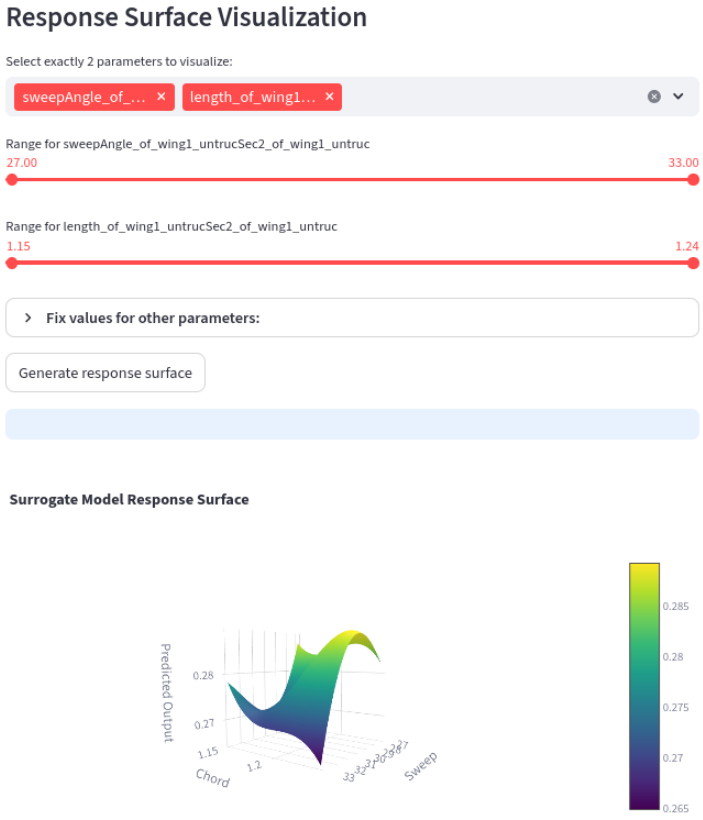


Figure 4.7: Response surface visualisation.

Two interactive modes are available: manual parameter exploration (Fig. 4.8) and target-based optimisation (Fig. 4.9), where classical optimisers (e.g. L-BFGS-B, Powell, COBYLA, Nelder–Mead) operate directly on the surrogate model.

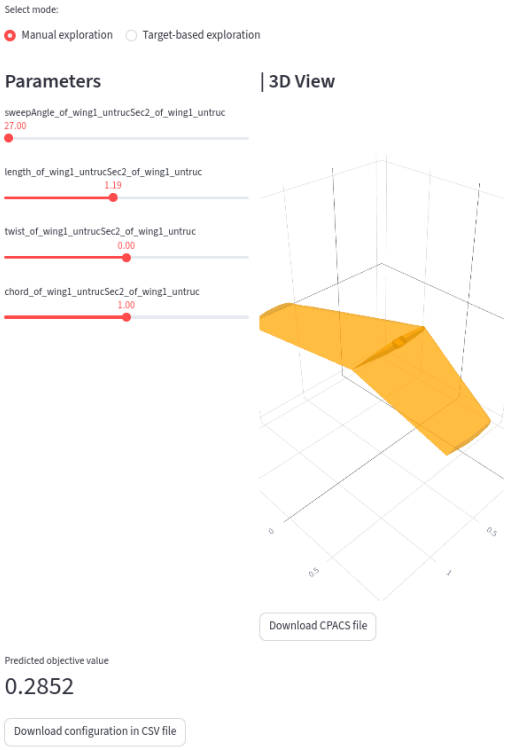


Figure 4.8: Manual surrogate exploration mode.

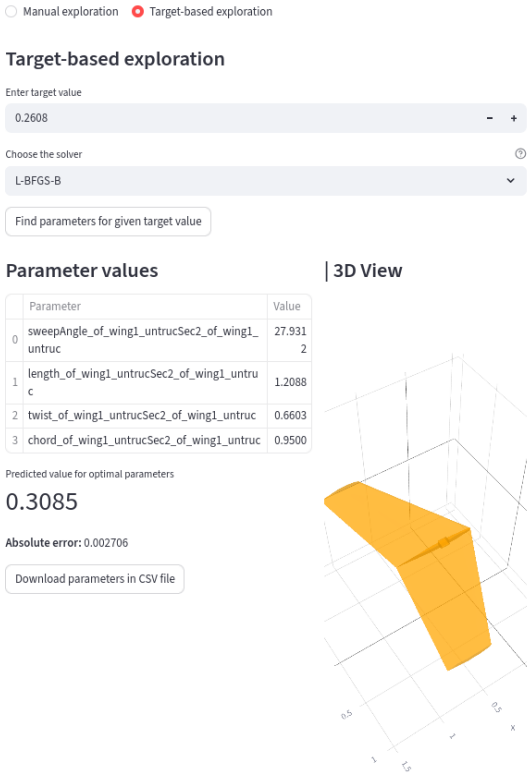


Figure 4.9: Target-based surrogate optimisation mode.

4.7 Summary

To summarise the implemented capability in algorithmic form, the workflow executed by `SMTrain` (and subsequently exploited in `SMUse`) is described in Algorithm 1.

The algorithm illustrates how the framework combines large low-cost AVL campaigns, used to capture the global aerodynamic trends of the design space, with selective SU2 evaluations that provide local corrections through high-fidelity simulations. In this way, the workflow establishes a practical and automated bridge between low- and high-fidelity modelling, delivering a GUI-driven environment for efficient shape exploration and surrogate-based optimisation.

Algorithm 1: Surrogate-Based Geometry Exploration with Two-Level Refinement

Input: Baseline CPACS; GUI settings; $\text{RMSE}_{\text{target}}$; $n_{\max}$
Output: Surrogate model; optimal configurations

Parse GUI settings
Generate LHS samples within bounds
for *each sample* **do**
    Update CPACS; run AVL; store valid outputs
end
Normalise and split dataset (train/val/test)
Train surrogate (KRG/RBF); compute RMSE and Sobol S_T
if $\exists\, p : S_T(p) < 0.05 \max_j S_T(j)$ **then**
    Remove negligible parameter(s); regenerate LHS
    go to sampling step
end
if *one-level* **then**
    Save model
else
    $n \leftarrow 0$
    while $\text{RMSE} > \text{RMSE}_{\text{target}} \wedge n < n_{\max}$ **do**
        Select refinement points
        if *KRG* **then**
            max variance
        else
            max LOO error
        end
        Run SU2; augment dataset; retrain; update RMSE
        $n \leftarrow n + 1$
    end
    Save model
end
Export results and enable exploration

4.8 Validation of the Surrogate-Based Workflow on the ONERA M6 Wing

Before applying the surrogate-based optimisation workflow to the UAV configuration considered later in this thesis, a preliminary validation study was performed on a simplified benchmark case. The purpose of this step is twofold: first, to verify the behaviour of the implemented surrogate-modelling workflow on a well-understood aerodynamic configuration; second, to evaluate the discrepancy between low-fidelity and high-fidelity aerodynamic models under different flow regimes.

For this purpose, the ONERA M6 wing was selected as validation case. This geometry is widely used in aerodynamic research due to the availability of experimental data and its well-documented aerodynamic characteristics, making it a suitable reference configuration for methodological validation studies.

4.8.1 Numerical Setup

Aerodynamic simulations were performed using both the AVL vortex-lattice solver and the SU2 CFD solver in order to compare low- and high-fidelity aerodynamic predictions.

AVL Discretisation The AVL simulations were performed using standard discretisation settings commonly adopted for preliminary aerodynamic analyses. The main vortex lattice parameters are summarised in Table 4.1.

Parameter	Value
Spanwise distribution	Cosine
Number of chordwise vortices	20
Number of spanwise vortices	50

Table 4.1: AVL discretisation settings used for low-fidelity simulations.

Mesh Generation Strategy High-fidelity simulations were performed using meshes generated with the `CPACS2GMSH` module integrated within *CEASIOMpy*.

An unstructured mesh generation strategy was adopted, with local refinement applied to aerodynamically sensitive regions, including the leading and trailing edges of the wing.

The computational domain size is controlled through the parameter `farfield_factor`, defined as a multiplicative factor applied to the largest characteristic dimension of the aircraft in order to determine the radius of the farfield boundary. In the present simulations, this parameter was selected so as to obtain a farfield radius of approximately 17 m, corresponding to about 25 times the wing chord. This choice ensures that the outer boundary is sufficiently far from the lifting surface to minimise potential boundary effects on the computed flow field.

Surface and volumetric discretisation are governed by the `mesh_size` parameters:

- `farfield`: scaling factor proportional to the largest surface element size, defining the cell size in the farfield region.
- `wings factor`: scaling factor proportional to the local curvature radius of wing surfaces, controlling surface mesh resolution.

Additional refinement controls include:

- `refine_factor`: refinement factor applied in the vicinity of leading and trailing edges.
- `auto_refine`: boolean flag enabling automatic refinement of geometrical features small relative to the prescribed mesh size.

Boundary Layer Discretisation Boundary-layer resolution was achieved through the generation of prismatic layers normal to the wall surfaces. The main parameters controlling this discretisation are:

- `number_layer`: number of prismatic layers generated normal to the wall.
- `height_first_layer`: height of the first prismatic cell adjacent to the wall, directly influencing the resulting y^+ value.
- `max_thickness_layer`: maximum allowable total thickness of the prismatic layer region.
- `growth_ratio`: ratio between wall-normal edge lengths of consecutive prismatic layers.

Parameter	**Value**
Mesh type	RANS
Farfield factor	10.0
Fuselage mesh size factor	1.0
Wing mesh size factor	2
Refine factor (LE/TE)	4
Number of prismatic layers	35
First layer height [m]	2.0×10^{-5}
Maximum layer thickness [mm]	120
Growth ratio (prismatic layers)	1.2
Growth factor (tetrahedra)	1.4
Automatic refinement	True

Table 4.2: Mesh generation parameters adopted for the RANS simulations using the `CPACS2GMSH` module.

The resulting mesh (Figure 4.10) contains approximately 1.3×10^6 elements.

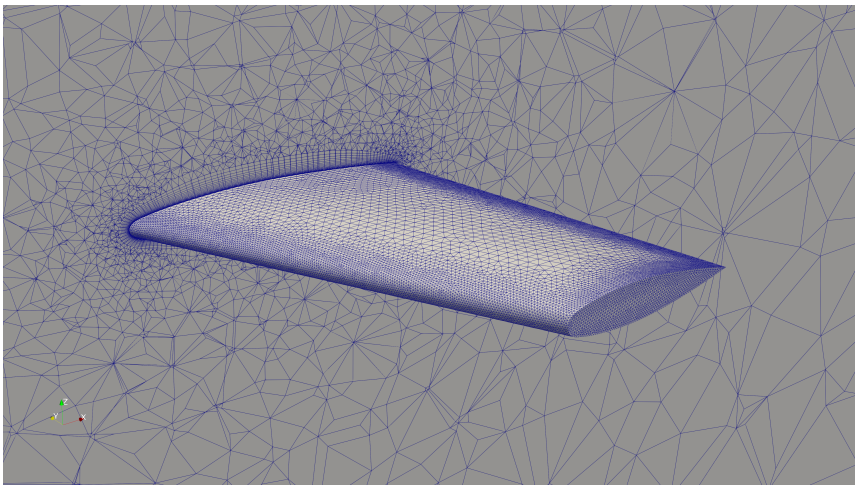


Figure 4.10: RANS mesh for the ONERA M6 wing.

SU2 Solver Settings All high-fidelity simulations were performed using the SU2 solver with a steady-state RANS formulation.

The main solver parameters adopted in SU2 are summarised below:

- **Maximum number of iterations**: 2000, ensuring sufficient iterations for residual reduction and stabilisation of aerodynamic coefficients.
- **CFL number**: 1.0.
- **CFL adaptation**: enabled to accelerate convergence, with adaptation factors of 0.5 (down) and 1.5 (up), and minimum/maximum bounds of 0.5 and 100.0.
- **Multigrid levels**: 3, used to accelerate convergence by damping low-frequency errors.
- **Turbulence model**: k–ω SST.

4.8.2 Low- and High-Fidelity Aerodynamic Comparison

In order to assess the discrepancy between the two modelling levels adopted in this work, a reference aerodynamic evaluation was performed using both AVL and SU2.

All simulations were carried out at the design flight condition of the ONERA M6 wing, corresponding to $M = 0.84$ and $\alpha = 3°$.

Solver	C_L	C_D
AVL	0.2726	0.00591
SU2	0.2807	0.02036

Table 4.3: Comparison between low-fidelity (AVL) and high-fidelity (SU2) aerodynamic predictions for the ONERA M6 wing at the design flight condition.

The lift coefficients predicted by the two solvers are relatively close, indicating that the vortex-lattice method captures the main aerodynamic trends of the configuration under the considered conditions.

A significantly larger discrepancy is observed for the drag coefficient. This behaviour is expected, since AVL only estimates the induced drag component, whereas the RANS simulations performed with SU2 also account for viscous and compressibility effects.

These observations are particularly relevant for the definition of the optimisation objective adopted in this work. Since the low-fidelity model provides a reasonable approximation of the lift coefficient while the drag prediction is strongly affected by modelling limitations, the optimisation problem is formulated using the lift coefficient C_L as the objective function.

4.8.3 Surrogate modelling validation

The validation study follows the same workflow introduced in the previous sections: (i) generation of a low-fidelity DoE, (ii) surrogate model training, (iii) optimisation using the surrogate model, and (iv) optional multi-fidelity refinement through targeted high-fidelity simulations.

Only a limited number of design variables was considered, so as to keep the surrogate model compact and the overall validation procedure computationally affordable.

Since, in this scenario, the geometry is defined by only two sections, any modification applied to the second section produces a global and approximately linear variation of the wing planform or local incidence distribution. For this reason, relatively narrow parameter ranges were selected in order to preserve the benchmark character of the configuration while still generating a sufficiently informative dataset for surrogate training.

The selected design variables were the sweep angle, tip chord scaling factor, section length, and tip twist. The twist range was deliberately kept small, since, with only two sections, the corresponding geometric variation affects the spanwise incidence distribution almost linearly along the entire wing.

Parameter	Baseline value	Selected range
Sweep angle [°]	30	[27, 33]
Twist [°]	0	[-1, 1]
Chord scale [-]	1.0	[0.95, 1.05]
Length [m]	1.196	[1.15, 1.24]

Table 4.4: Selected design variables and admissible ranges for the ONERA M6 validation case.

An initial low-fidelity DoE was generated according to the sampling strategy introduced in the previous sections. A total of 200 geometries were sampled within the bounds reported in Table 4.4 and analysed using AVL, thereby providing the dataset used to train the first-level surrogate.

Low-fidelity surrogate construction The surrogate model was trained using a RBF approximation. The optimal hyperparameters identified during the training phase were a scaling parameter $d_0 = 11.761$, a polynomial degree equal to 0, and a regularisation parameter $\text{reg} = 1.08 \times 10^{-4}$. The resulting model achieved a root mean square error of RMSE = 0.021962 on the test set, indicating a satisfactory prediction accuracy over the explored design space.

Once trained, the surrogate model was used within the optimisation loop as a fast aerodynamic predictor to evaluate candidate configurations during the search for improved designs. The total execution time of the low-fidelity workflow, including the generation of the DoE, the AVL aerodynamic simulations, surrogate model training, and the optimisation stage, was approximately 400 seconds.

Low-fidelity optimisation results The optimisation process was carried out using the Powell optimisation algorithm in order to identify the configuration maximising the lift coefficient C_L within the admissible design space.

The surrogate model predicted a maximum lift coefficient of $C_L = 0.326$. In order to verify the validity of this prediction, the corresponding geometry was evaluated using the AVL solver. The aerodynamic simulation returned a lift coefficient of $C_L = 0.33085$, confirming the accuracy of the surrogate prediction and validating the optimisation result.

The optimisation process produced a configuration characterised by moderate modifications of the wing geometry. In particular, the surrogate-based optimisation tends to increase the spanwise lifting capability of the wing by slightly increasing the wing length, introducing a positive twist, and reducing the sweep angle. These geometric changes contribute to enhancing the effective lift generation along the span while maintaining a configuration close to the original ONERA M6 geometry.

The geometric parameters of the optimised configuration are summarised in Table 4.5, together with the baseline values of the reference configuration in order to facilitate a direct comparison.

Parameter	Baseline configuration	Optimised configuration
Sweep angle [°]	30.0	28.2
Twist [°]	0.00	0.93
Chord scale [-]	1.00	1.04
Length [m]	1.196	1.24
Predicted C_L [-]	0.273	0.326

Table 4.5: Comparison between the baseline ONERA M6 geometry and the configuration obtained through low-fidelity surrogate optimisation.

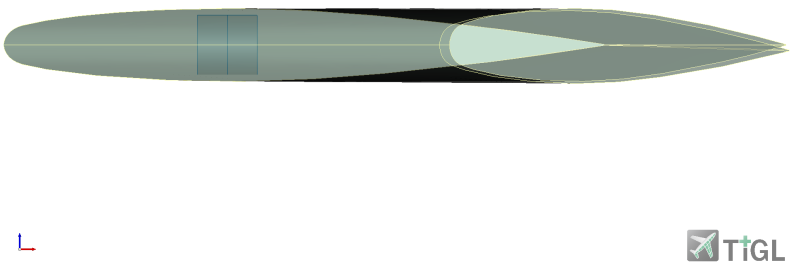


Figure 4.11: Overlay comparison of the baseline and optimised ONERA M6 geometries in lateral view.

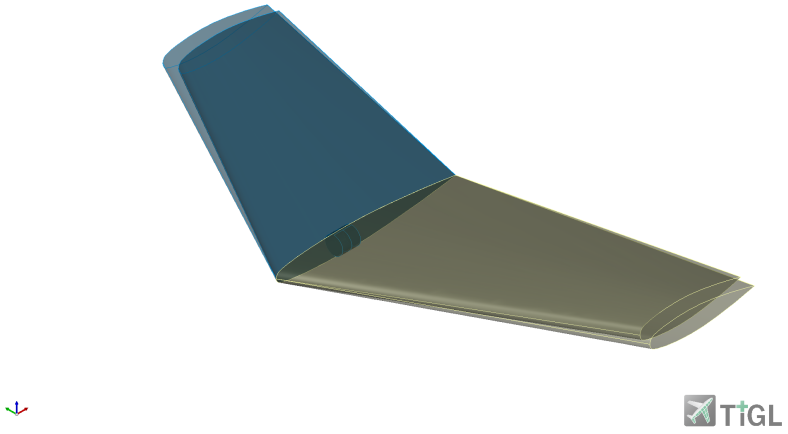


Figure 4.12: Isometric view of the superposed baseline and optimised ONERA M6 geometries generated with the `CPACSCreator` module.

Figures 4.11 and 4.12 present a direct visual comparison between the baseline and optimised configurations generated using the `CPACSCreator` module. By superposing the two geometries, the modifications introduced by the optimisation procedure become immediately visible. Although the geometric variations remain relatively small, the optimisation successfully identifies a region of the design space associated with increased lift performance.

Multi-fidelity refinement The surrogate model can be further improved by introducing a limited number of high-fidelity aerodynamic evaluations, leading to the construction of a multi-fidelity surrogate model. This refinement step is optional and can be activated when a higher level of predictive accuracy is required.

The process starts from the low-fidelity surrogate trained on the AVL dataset. To initiate the refinement, the user specifies a target prediction error expressed through a desired $\mathrm{RMSE}_{\mathrm{objective}}$. New candidate configurations are then selected according to the refinement strategy described in Subsection 4.4.3 and evaluated using high-fidelity CFD simulations performed with SU2, following the numerical setup presented in Subsection 4.8.1. The resulting aerodynamic data are incorporated into the training database and the surrogate model is retrained.

After each refinement step, the prediction error of the updated surrogate is evaluated. The process terminates when the resulting RMSE falls below the prescribed $\mathrm{RMSE}_{\mathrm{objective}}$. Otherwise, additional sampling points are generated and further high-fidelity simulations are performed.

To avoid excessively long refinement campaigns, the procedure also includes a maximum number of refinement iterations $n_{\max}$, which acts as a safeguard stopping criterion. In the present implementation, $n_{\max}$ is set by default equal to the size of the training dataset, as reported in Algorithm 1. The refinement cycle therefore continues until either the target prediction accuracy is reached or the maximum number of iterations is exceeded.

In order to evaluate the effectiveness of the proposed refinement strategy, a first multi-fidelity experiment was conducted starting from the low-fidelity surrogate model described in the previous section. The initial RBF surrogate trained on the AVL dataset achieved a prediction error of RMSE = 0.021962 on the test set.

A moderate accuracy improvement was then targeted by setting the refinement threshold to

$$\mathrm{RMSE}_{\mathrm{target}} = 0.018.$$

This value was intentionally chosen close to the original error level in order to require a meaningful improvement in predictive accuracy while keeping the number of additional high-fidelity simulations limited. In practical surrogate modelling applications, such moderate error reductions are often preferred since they provide a good compromise between computational cost and model accuracy.

In the present case, the first refinement batch consisted of 7 candidate configurations. After incorporating the corresponding high-fidelity data into the training database and retraining the surrogate model, the prediction error was reduced to

$$\mathrm{RMSE} = 0.017808.$$

Since the stopping criterion was already satisfied after this first refinement step, no additional high-fidelity simulations were required and the refinement process terminated automatically.

Figure 4.13 shows the distribution of the high-fidelity refinement samples within the explored design space together with the original low-fidelity dataset.

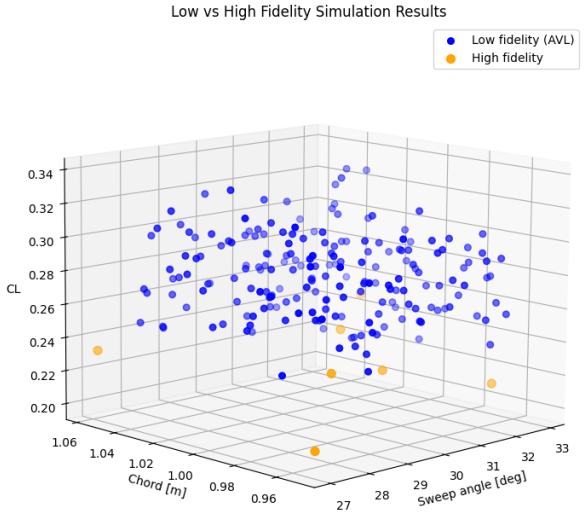


Figure 4.13: Distribution of low-fidelity AVL samples and high-fidelity refinement points for the first multi-fidelity refinement case.

It can be observed that the refinement points are primarily located in regions associated with relatively low values of the lift coefficient. This behaviour is a direct consequence of the LOO-based refinement strategy, which selects points where the surrogate prediction error is largest rather than regions corresponding to optimal aerodynamic performance.

The refined multi-fidelity surrogate was subsequently used to identify the configuration associated with the maximum predicted lift coefficient within the explored design space.

Table 4.6 summarises the optimal geometric parameters predicted by the low-fidelity and the first multi-fidelity surrogate models, while the corresponding geometries are visually compared in Figs. 4.14 and 4.15.

Parameter	Low Fidelity	First multi-fidelity
Sweep angle [°]	28.2	27.0
Twist [°]	0.93	1.00
Chord scale [-]	1.04	1.04
Length [m]	1.24	1.24
Predicted C_L [-]	0.326	0.346

Table 4.6: Optimal geometric parameters predicted by the low-fidelity surrogate model and by the first multi-fidelity surrogate.

Although the overall design trends remain consistent, the inclusion of high-fidelity data leads to noticeable adjustments in several geometric parameters.

In particular, the multi-fidelity model predicts a slightly lower sweep angle and a higher positive twist, both of which contribute to increasing the effective lift generation along the span. At the same time, the optimal chord scaling and spanwise length are reduced compared to the low-fidelity solution.

The inclusion of high-fidelity data in these poorly predicted regions improves the global smoothness and reduces interpolation artefacts, which in turn slightly modifies the location of the predicted aerodynamic optimum.

This behaviour is typical of global surrogate models such as RBF, where the addition of new samples affects the entire response surface rather than only the local region around the sampled point.

As a result, the multi-fidelity surrogate provides a more reliable estimate of the optimal configuration while preserving the overall trends identified by the low-fidelity dataset.

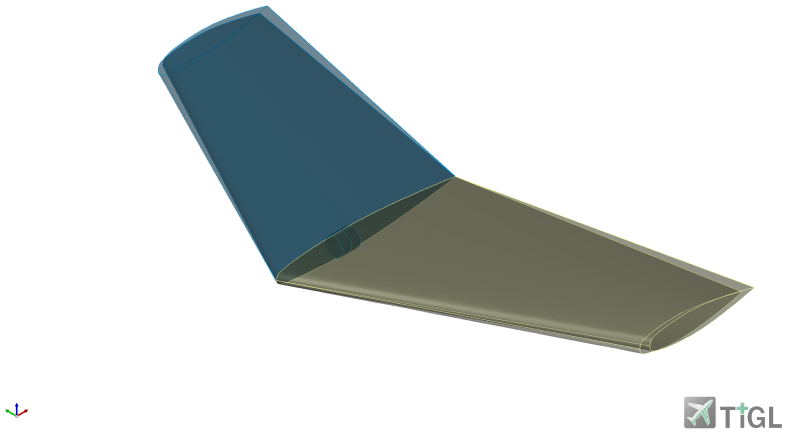


Figure 4.14: Isometric comparison of the optimal wing geometries predicted by the low-fidelity and the refined multi-fidelity surrogate models.

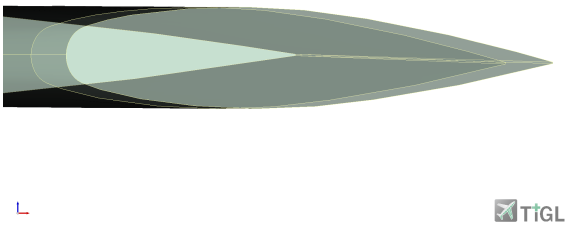


Figure 4.15: Lateral comparison of the airfoil sections associated with the optimal configurations predicted by the low-fidelity and multi-fidelity surrogate models.

In order to further investigate the behaviour of the proposed refinement strategy, a second experiment was performed by imposing a significantly lower target prediction error, namely

$$\mathrm{RMSE}_{\mathrm{target}} = 0.015.$$

The target RMSE was reached after the introduction of 42 high-fidelity refinement points. Each SU2 simulation required on average approximately 140 minutes when executed on 12 CPU cores. As a consequence, the total computational time required to reach the desired surrogate accuracy was on the order of four days.

This result highlights an important practical aspect of the multi-fidelity refinement strategy. While the method is capable of achieving very high surrogate accuracy through the progressive introduction of high-fidelity data, the computational cost associated with the refinement process can become significant when very strict error targets are imposed.

Figure 4.16 shows the distribution of the high-fidelity refinement samples within the design space for this second scenario.

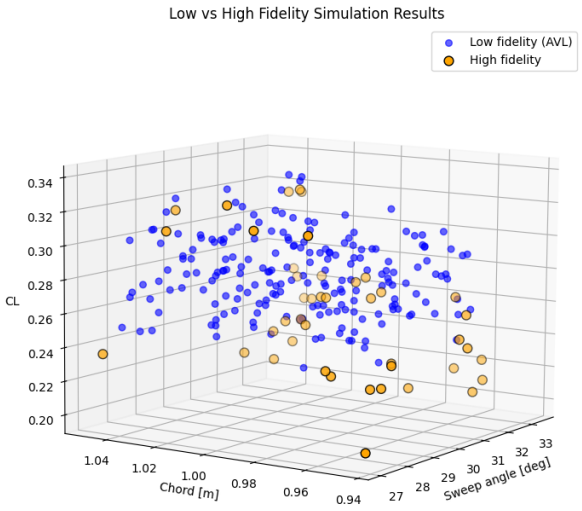
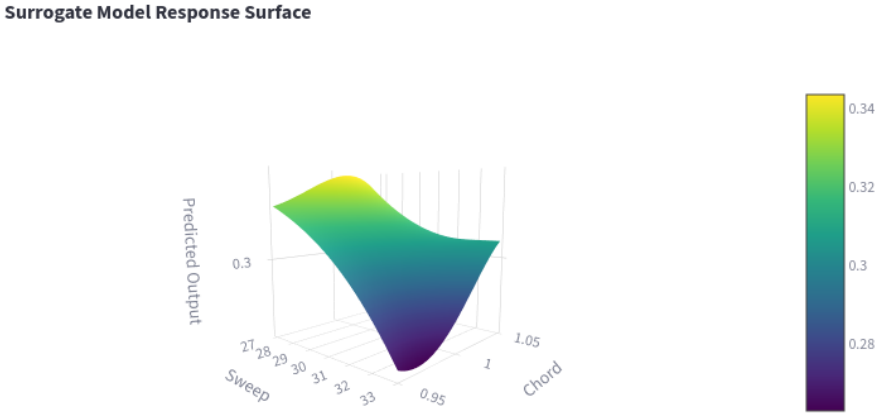
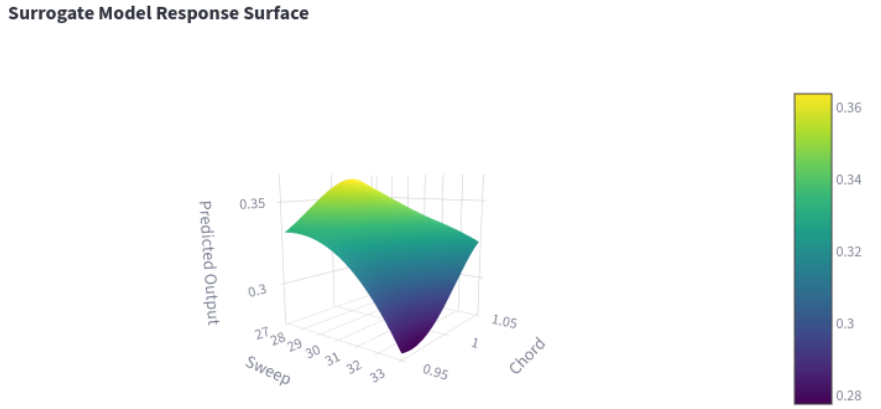


Figure 4.16: Distribution of the high-fidelity refinement samples in the design space for the second multi-fidelity refinement scenario.

In order to further analyse the impact of the refinement process on the surrogate model behaviour, the response surfaces obtained for the two multi-fidelity scenarios are compared in Fig. 4.17 (left/right).



(a) First refinement case ($\text{RMSE}_{\text{target}} = 0.018$)



(b) Second refinement case ($\text{RMSE}_{\text{objective}} = 0.015$)

Figure 4.17: Comparison between surrogate response surfaces for the two multi-fidelity refinement scenarios.

A comparison of the two response surfaces highlights the effect of the number of high-fidelity correction points on the multi-fidelity surrogate model. Both models exhibit a smooth and physically consistent representation of the aerodynamic response, confirming that even a limited number of high-fidelity samples is sufficient to regularise the surrogate surface.

The first refinement stage primarily improves local consistency by reducing interpolation artefacts in regions of high prediction error. Increasing the number of high-fidelity points from 7 to 42, however, leads to a different effect: a noticeable shift in the predicted response levels. While the overall structure of the design space remains unchanged, the second refinement scenario predicts systematically higher values of the lift coefficient. This indicates that the low-fidelity information embedded in the initial dataset tends to underestimate C_L, and that the progressive inclusion of high-fidelity data effectively corrects this bias.

From an aerodynamic perspective, this behaviour is consistent with the limitations of vortex-lattice methods, which may underpredict lift due to the simplified modelling of three-dimensional flow effects and load redistribution. The multi-fidelity refinement therefore acts as a correction mechanism, preserving the global structure of the response surface while improving the quantitative accuracy of the predicted aerodynamic performance.

The configuration associated with the maximum predicted lift coefficient in the second refinement case corresponds to $C_L = 0.363$. This value was subsequently evaluated using AVL and validated against SU2. The AVL evaluation returned $C_L = 0.352$, confirming the residual underestimation

typical of low-fidelity models, while the high-fidelity SU2 simulation yielded $C_L = 0.368$. The multi-fidelity prediction is therefore significantly closer to the RANS result than to the AVL evaluation, confirming the effectiveness of the refinement strategy in recovering high-fidelity aerodynamic behaviour.

Effect of refinement accuracy targets The two refinement scenarios highlight the practical behaviour of the proposed multi-fidelity workflow. When a moderate accuracy improvement is required, the surrogate model can be effectively refined with a very limited number of high-fidelity simulations. In the first experiment, only seven additional CFD evaluations were sufficient to reduce the prediction error below the prescribed threshold, confirming the data-efficiency of the refinement strategy. In this configuration, the large low-fidelity dataset captures the global aerodynamic trends, while a small number of targeted high-fidelity simulations corrects the surrogate response.

Conversely, imposing a significantly lower prediction error requires a substantially larger number of high-fidelity evaluations. As observed in the second experiment, achieving a stricter accuracy target required forty-two additional CFD simulations, leading to a considerable increase in the overall computational cost.

Table 4.7 summarises the main characteristics of the different surrogate modelling strategies investigated in this study.

Model	RMSE	HF simulations	Computational time
Low-fidelity surrogate	0.021962	0	$\sim$400 s
Multi-fidelity (Case 1)	0.017808	7	$\sim$16 h
Multi-fidelity (Case 2)	0.014316	42	$\sim$4 days

Table 4.7: Accuracy–cost trade-off for the investigated surrogate modelling strategies.

These results illustrate the fundamental trade-off that characterises multi-fidelity surrogate modelling. While the introduction of high-fidelity simulations enables a progressive improvement in prediction accuracy, the marginal benefit of additional refinement decreases as the surrogate model becomes more accurate. For practical aerodynamic design applications, moderate accuracy targets therefore often represent the most efficient compromise between predictive reliability and computational cost.

5 Winglet optimisation for the UAV configuration

After validating the surrogate-based workflow on the ONERA M6 benchmark case, the methodology was applied to a practical design problem involving the winglet of a UAV developed by the Dope Hubs student association at the University of Genoa for the UAS Challenge 2025 competition.

The objective of this study is threefold. First, to demonstrate the capability of the implemented surrogate-based workflow to efficiently explore a complex geometric design space. Second, to identify winglet configurations that improve the aerodynamic lift characteristics while satisfying the geometric constraints imposed by the UAS Challenge regulations. Finally, to assess the applicability of the *CEASIOMpy* framework to a practical UAV configuration, thereby validating its use for unconventional aircraft design studies.

5.1 Geometry parametrisation and design variables

The winglet geometry was parametrised by applying geometric transformations to the last four wing sections (Sections 3–6 shown in Fig. 3.2).

For each section, a subset of geometric parameters was selected as design variables, including sweep angle, dihedral angle, twist, and local chord. In total, 15 design variables were considered in the optimisation problem, leading to a relatively high-dimensional design space compared to the validation case discussed in the previous section.

Due to this high dimensionality, the systematic use of high-fidelity CFD simulations would require a prohibitively large computational effort. For this reason, and considering the limited computational resources available, the optimisation study was conducted using a single-fidelity approach based exclusively on low-fidelity aerodynamic evaluations performed with the AVL vortex-lattice solver.

This modelling choice is also consistent with the aerodynamic regime of the considered UAV configuration. The aircraft operates at relatively low Mach numbers ($M = 0.06$) and moderate angles of attack ($\alpha = 3°$), conditions under which vortex-lattice methods are generally able to capture the dominant aerodynamic trends associated with lifting surfaces.

The admissible ranges assigned to each parameter are reported in Table 5.1.

Section	Sweep [°]	Dihedral [°]	Twist [°]	Chord [m]
3	58.0 [10.0–70.0]	3.6 [2.5–8.0]	0.7 [-0.3–2.7]	0.4 [0.34–0.5]
4	38.8 [20.0–50.0]	12.4 [8.0–25]	1.2 [-0.8–2.2]	0.3 [0.26–0.34]
5	33.7 [20.0–45.0]	50.0 [25.0–60.0]	6.0 [-3.0–7.0]	0.2 [0.14–0.26]
6	47.0 [40.0–70.0]	65.9 [60.0–85.0]	2.5 [-3.0–3.5]	–

Table 5.1: Winglet geometric parameters and corresponding design ranges.

The choice of the parameter ranges plays a critical role in surrogate-based optimisation. If excessively large ranges are assigned, the optimisation process may converge towards geometries that are aerodynamically favourable according to the surrogate model but unrealistic from a structural or regulatory perspective. Conversely, overly restrictive bounds may prevent the exploration of potentially beneficial configurations.

For this reason, the selected bounds were defined so as to preserve a realistic geometric envelope while still allowing sufficient design flexibility for the surrogate model to capture meaningful aerodynamic trends. In particular, parameters directly affecting the overall winglet length were intentionally kept fixed in order to remain consistent with the competition constraints. At the same time, wider ranges were allowed for the dihedral angles of the winglet sections, since these parameters control the effective curvature and vertical development of the winglet, which are known to have a significant influence on the aerodynamic behaviour of wingtip devices.

5.2 DoE and surrogate model training

In order to explore the high-dimensional design space defined by the selected geometric parameters, a DoE was generated using a LHS strategy. This approach ensures a space-filling distribution of samples while avoiding clustering effects that could reduce the quality of the surrogate model.

For each sampled design vector, a new CPACS geometry was automatically generated by updating the corresponding geometric parameters in the XML structure. Each sampled configuration therefore corresponds to a specific aircraft geometry defined by a unique CPACS file.

Resulting geometries were analysed using the AVL vortex-lattice solver to compute the lift coefficient C_L. A static stability assessment was performed through the `StaticStability` module integrated in CEASIOMpy. Only configurations satisfying the stability requirement were retained for surrogate model training.

This procedure resulted in a dataset composed of pairs consisting of a vector of geometric parameters and the corresponding aerodynamic response.

All input variables were subsequently standardised through z-score normalisation to ensure comparable scaling among the geometric parameters and to improve the numerical conditioning of the regression problem.

The dataset was then divided into training and validation/test subsets using a 70/20/10 split, allowing the predictive performance of the surrogate model to be evaluated on previously unseen data.

Since surrogate model performance strongly depends on the density of sampling within the design space, a preliminary dataset-size analysis was conducted to assess the influence of the number of training samples on the predictive capability of the model. To this end, surrogate models were trained using progressively larger subsets of the available database containing 1000, 3000, and 5000 samples.

The resulting prediction errors are reported in Table 5.2.

Number of samples	RMSE
1000	0.038291
3000	0.031280
5000	0.027561

Table 5.2: Influence of the dataset size on the predictive accuracy of the RBF surrogate model.

The results highlight the strong dependence of surrogate accuracy on the sampling density within the design space. Increasing the number of training samples leads to a consistent reduction of the prediction error, indicating that a denser coverage of the design space allows the surrogate model to better capture the nonlinear aerodynamic relationships between the geometric parameters and the lift coefficient.

In the present case, the aerodynamic response depends on fifteen geometric design variables, resulting in a relatively complex and highly nonlinear design space. Consequently, smaller datasets provide only a sparse coverage of the parameter domain, limiting the ability of the surrogate model to accurately approximate the underlying input–output relationships. As the number of samples increases, the distribution of points becomes progressively more space-filling, enabling the RBF model to provide a more accurate approximation of the global aerodynamic response surface.

It is also worth noting that the improvement in predictive accuracy becomes progressively smaller as the dataset size increases. This behaviour reflects the diminishing returns typically observed in surrogate modelling problems once the main features of the response surface have been captured.

In the present study, the dataset size was limited to 5000 samples due to computational resources available for this study. Although individual AVL simulations are relatively inexpensive, the automated generation of CPACS geometries and the evaluation of large design-of-experiments campaigns eventually saturated the available computational resources of the workstation used for the simulations. For this reason, the sampling campaign was not extended beyond this size.

Based on this analysis, the surrogate model used for the optimisation study was generated using the complete dataset of 5000 samples.

Once trained, the surrogate model was therefore used as a fast aerodynamic predictor within the optimisation loop, enabling the exploration of candidate winglet configurations without requiring additional AVL simulations.

5.3 Optimisation results

The optimisation of the winglet geometry was performed using the surrogate model as an objective function evaluator. The objective of the optimisation was to maximise the lift coefficient C_L of the aircraft configuration.

A Powell optimisation algorithm was employed to search for the design vector corresponding to the maximum predicted lift coefficient within the admissible parameter bounds.

The optimisation process first identified a configuration characterised by a predicted lift coefficient of $C_L = 1.36$. When evaluated with the AVL solver, this configuration (shown in Fig. 5.1) yielded a lift coefficient of $C_L = 1.40$, confirming that the surrogate model correctly identified a region of the design domain associated with improved aerodynamic performance.

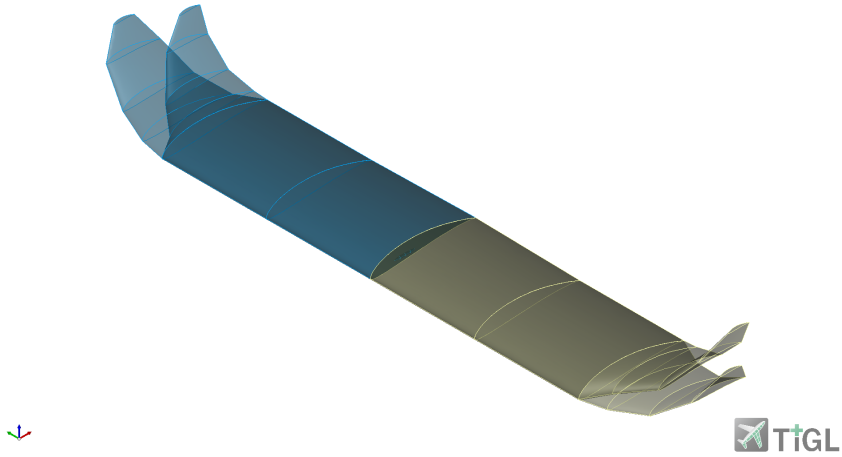


Figure 5.1: Isometric comparison between the baseline winglet geometry and the unconstrained optimised configuration predicted by the surrogate model.

However, the configuration corresponding to this unconstrained optimum resulted in a winglet geometry that increased the overall span beyond the baseline wing length. Although aerodynamically favourable, such a configuration may not always be desirable in practical design scenarios where geometric constraints on the overall aircraft dimensions must be respected.

Interestingly, the parameters associated with this unconstrained optimum were located close to the boundaries of the admissible parameter ranges. In particular, several geometric variables tended towards extreme values corresponding to lower sweep angles, higher twist, and increased chord lengths. This behaviour indicates that the surrogate model successfully captured the underlying aerodynamic trends relating the geometric design variables to the lift coefficient.

These results therefore provide indirect evidence that the surrogate model learned the correct input–output relationships, even though the resulting optimum corresponds to an unconstrained configuration.

If maintaining the original wing span is desired, the surrogate model can be explored interactively through the graphical interface of *CEASIOMpy* (as shown in Fig. 4.8). By manually adjusting the geometric parameters while monitoring the surrogate predictions, configurations satisfying the desired geometric constraints can be identified without requiring additional aerodynamic simulations.

Using this manual exploration capability, a configuration preserving the overall wing length while still improving the aerodynamic performance was identified.

The resulting constrained configuration achieves a lift coefficient of $C_L = 1.28$, corresponding to a noticeable improvement over the baseline configuration, which has a lift coefficient of $C_L = 1.21$.

The geometric parameters of the constrained optimal configuration are summarised in Table 5.3.

Parameter	Value
sweepAngle_of_WingMainSec3_of_WingMain	23.19
dihedralAngle_of_WingMainSec3_of_WingMain	6.01
twist_of_WingMainSec3_of_WingMain	2.70
chord_of_WingMainSec3_of_WingMain	0.45
sweepAngle_of_WingMainSec4_of_WingMain	45.03
dihedralAngle_of_WingMainSec4_of_WingMain	20.34
twist_of_WingMainSec4_of_WingMain	1.50
chord_of_WingMainSec4_of_WingMain	0.34
sweepAngle_of_WingMainSec5_of_WingMain	31.52
dihedralAngle_of_WingMainSec5_of_WingMain	50.09
twist_of_WingMainSec5_of_WingMain	6.40
chord_of_WingMainSec5_of_WingMain	0.26
sweepAngle_of_WingMainSec6_of_WingMain	70.00
dihedralAngle_of_WingMainSec6_of_WingMain	75.06
twist_of_WingMainSec6_of_WingMain	2.7

Table 5.3: Geometric parameters of the optimised winglet configuration satisfying the UAS Challenge constraints.

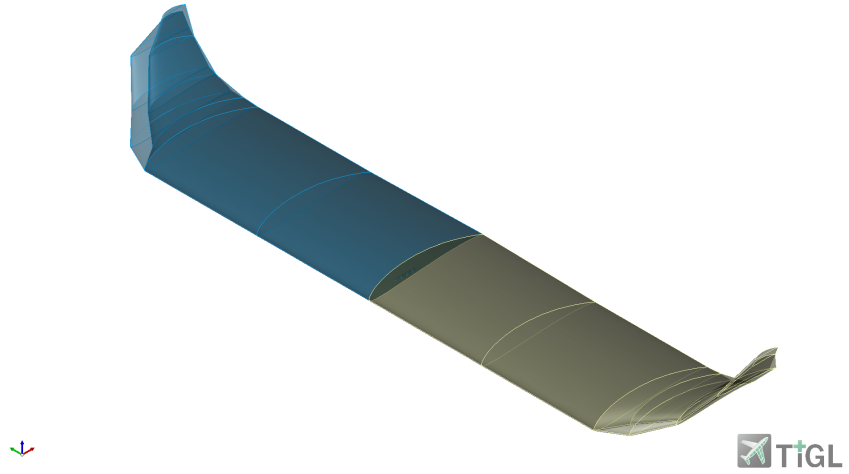


Figure 5.2: Isometric view comparison between the baseline and optimised winglet geometries.

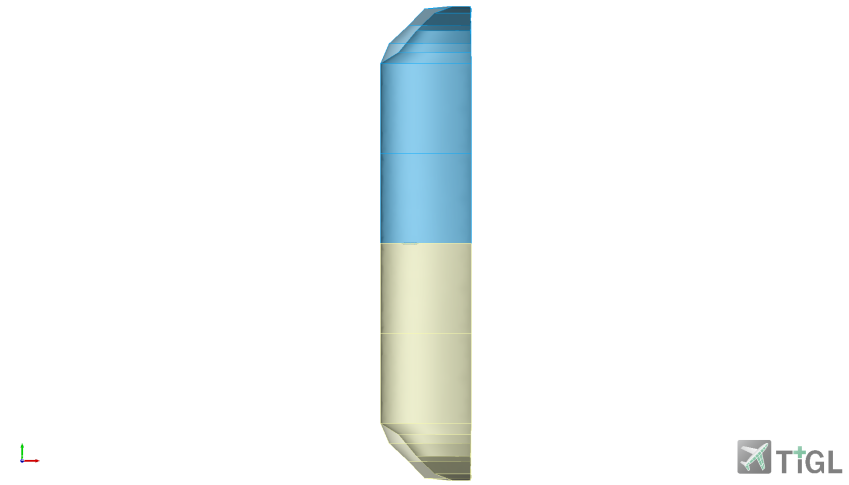


Figure 5.3: Top view comparison between the baseline and optimised winglet geometries.

Figure 5.4: Front view comparison between the baseline and optimised winglet geometries.

5.4 Interpretation of optimisation results and modelling limitations

The optimisation results obtained in the previous sections should be interpreted in light of the modelling assumptions adopted in this study.

Under the low-speed flight conditions considered for the UAV configuration, vortex-lattice methods are generally able to capture the dominant aerodynamic trends associated with lifting surfaces. In particular, they provide reliable estimates of lift generation and spanwise load distribution for attached-flow regimes. For this reason, AVL constitutes an appropriate tool for large-scale design-space exploration, where thousands of aerodynamic evaluations are required.

However, vortex-lattice formulations neglect viscous effects, boundary-layer development, flow separation, and detailed vortex dynamics that may influence the aerodynamic behaviour of wingtip devices. In particular, the prediction of induced drag and the detailed structure of the wingtip vortex may differ from the results that would be obtained using higher-fidelity CFD simulations.

Consequently, the optimisation results presented in this work should be interpreted primarily as the identification of promising geometric trends within the design space rather than as definitive aerodynamic performance predictions. The surrogate model successfully captures the global relationship between the winglet geometric parameters and the lift coefficient, enabling efficient exploration of a high-dimensional design space and the identification of configurations associated with improved aerodynamic performance.

In realistic aircraft design workflows, configurations identified through surrogate-based optimisation would typically undergo further validation using higher-fidelity CFD simulations or experimental testing. Such analyses would allow a more detailed assessment of viscous effects, vortex structure, and potential aerodynamic interactions between the wing and the winglet.

Despite these limitations, the present study demonstrates that surrogate-based optimisation integrated within the *CEASIOMpy* framework provides a powerful tool for early-stage aerodynamic design. By combining large low-cost datasets generated with AVL and data-driven surrogate models, the workflow enables efficient exploration of complex geometric design spaces while maintaining computational costs compatible with conceptual aircraft design studies.

6 Conclusions and Perspectives

This thesis presented the development and integration of a surrogate-based aerodynamic shape optimisation capability within the *CEASIOMpy* framework. The objective of the work was to enable efficient exploration of high-dimensional geometric design spaces during the conceptual aircraft design phase, where rapid evaluation of large numbers of configurations is required while maintaining manageable computational costs.

The proposed workflow combines automated geometry parametrisation through CPACS, aerodynamic evaluations performed at different fidelity levels (ranging from low-fidelity vortex-lattice simulations using AVL to higher-fidelity CFD analyses with SU2), and machine-learning-based surrogate modelling techniques. By approximating the aerodynamic response across the design space, the surrogate model enables efficient optimisation procedures that would otherwise require a prohibitively large number of aerodynamic simulations.

6.1 Main contributions

The main contributions of this thesis can be summarised as follows:

- Implementation of a surrogate-based geometry exploration capability within the *CEASIOMpy* framework, extending the existing `SMTrain` module to support geometric design variables defined through the CPACS parametrisation.
- Development of an automated workflow integrating CPACS-based parametric geometry generation, aerodynamic evaluation at different fidelity levels, surrogate model training, and optimisation within a unified computational environment.
- Implementation of adaptive surrogate refinement strategies compatible with both Kriging and Radial Basis Function models, enabling progressive improvement of surrogate accuracy through selective high-fidelity simulations.
- Validation of the proposed workflow on the ONERA M6 benchmark configuration, demonstrating the capability of the framework to capture aerodynamic trends and improve surrogate predictions through multi-fidelity refinement.
- Application of the methodology to the optimisation of a UAV winglet configuration, illustrating the ability of the framework to explore complex high-dimensional design spaces and identify geometries associated with improved aerodynamic performance.

Overall, the developed capability represents a significant extension of the *CEASIOMpy* ecosystem, providing a practical surrogate-assisted optimisation tool suitable for conceptual aircraft design studies.

6.2 Limitations and Future Work

Despite the promising results obtained in this work, some limitations should be acknowledged.

The aerodynamic optimisation of the UAV configuration relied on low-fidelity aerodynamic evaluations performed with the AVL vortex-lattice solver. While this modelling approach is appropriate for rapid design-space exploration and for capturing global aerodynamic trends at low Mach numbers and moderate angles of attack, it does not account for viscous effects, flow separation, or detailed vortex dynamics.

For this reason, the optimisation results should be interpreted primarily as the identification of promising geometric trends rather than as definitive aerodynamic performance predictions. In practical aircraft design workflows, the configurations identified through surrogate-based optimisation would typically undergo further verification using higher-fidelity CFD simulations or experimental testing.

Another limitation concerns the absence of explicit constraint handling within the optimisation loop. In the present implementation, geometric and regulatory constraints were handled manually

during the exploration of the surrogate response surface. Integrating constraint-aware optimisation strategies directly within the workflow would allow such limitations to be enforced automatically during the optimisation process, improving both the robustness and the level of automation of the framework and enabling its application to more realistic preliminary aircraft design problems.

A further potential extension concerns the adoption of multi-objective and multidisciplinary optimisation strategies. The optimisation studies performed in this work focused on a single aerodynamic objective, namely the maximisation of the lift coefficient. In realistic aircraft design scenarios, however, multiple interacting disciplines must be considered simultaneously, including aerodynamic performance, structural feasibility, stability, and mission requirements.

Overall, this work represents a first step towards more automated and data-driven optimisation workflows for conceptual aircraft design. By combining surrogate modelling, hierarchical aerodynamic simulations, and modular software integration, the proposed methodology contributes to the ongoing evolution of digital aircraft design frameworks aimed at improving design efficiency, flexibility, and sustainability.

References

[1] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Washington, DC, 2nd edition, 1992.

[2] AVI-8. The role of computational fluid dynamics (cfd) in modern aircraft design, 2024.

[3] Edward N. Tinoco. Cfd uncertainty and validation for commercial aircraft applications. Technical report, Boeing Commercial Airplanes, Seattle, WA, USA, 2004. AIAA-2004-2122.

[4] Anders Ytterström. *Parallel Computing for Applications in Aeronautical CFD*. PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2001.

[5] Arthur Rizzi and James M. Luckring. Historical development and use of cfd for separated flow simulations relevant to military aircraft. *Aerospace Science and Technology*, 117:106940, 2021.

[6] Antony Jameson and Massimiliano Fatica. Using computational fluid dynamics for aerodynamics. Technical report, Stanford University, 2006. White paper, National Research Council Workshop.

[7] Mark Drela. *Athena Vortex Lattice (AVL) User Primer*, 2014. Available online: https://web.mit.edu/drela/Public/web/avl/.

[8] Ideen Sadrehaghighi. Mesh generation in cfd. Technical report, GEOLAB, 2020.

[9] N. Gourdain et al. High performance parallel computing of flows in complex geometries - part 1: Methods. Technical Report TR_CFD_09_117, CERFACS, Toulouse, France, 2009.

[10] Marsha J. Berger and Michael J. Aftosmis. An ode-based wall model for turbulent flow simulations. In *AIAA SciTech Forum*, 2017.

[11] Matheo López-Pachón and Jordi Marcé-Nogué. The crucial role of meshing in computational fluid dynamics simulations for organic geometries in paleobiology. *Methods in Ecology and Evolution*, 16(10):p. 2170 ss., 2025.

[12] IdealSimulations. Cfd mesh guide, 2023. Available at: `https://www.idealsimulations.com/guides/cfd-mesh-guide/`.

[13] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):p. 1309 ss., 2009.

[14] SUMO Engineering. *Sumo Pentagrow Manual*. User manual for Pentagrow hybrid mesh generation tool.

[15] SU2 Foundation. Su2: An open-source suite of multiphysics simulation and design software, 2026.

[16] Pratikkumar Raje. Spalart-allmaras turbulence model for compressible flows. Technical report, Indian Institute of Technology Bombay, 2015. Technical report, IIT Bombay.

[17] Steven L. Brunton et al. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:p. 477 ss., 2020.

[18] Ricardo Vinuesa and Steven L. Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2:p. 358 ss., 2022.

[19] Rémi Lam, Douglas Allaire, and Karen Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, jan 2015. American Institute of Aeronautics and Astronautics.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[21] Praveen Chandrashekarappa and Regis Duvigneau. Radial basis functions and kriging metamodels for aerodynamic optimization. Research Report RR-6151, INRIA, 2007. inria-00137602v1.

[22] Xiu Yang, Guzel D. Tartakovsky, and Alexandre M. Tartakovsky. Physics-informed kriging: A physics-informed gaussian process regression method for data-model convergence. Technical Report PNNL-28876, Pacific Northwest National Laboratory, Richland, WA, USA, 2019. Submitted to Journal of Computational Physics.

[23] Donald R. Jones et al. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):p. 455 ss., 1998.

[24] Alexander I. J. Forrester and Andy J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):p. 1 ss., 2009.

[25] CFS Engineering. Ceasiompy: Open-source conceptual aircraft design environment. `https://github.com/cfsengineering/CEASIOMpy`.

[26] CFS Engineering. Cfs engineering: Computational fluid & structures engineering. https://cfse.ch.

[27] Airinnova AB. Airinnova: Computational technology solutions for cutting-edge aircraft design. https://airinnova.se.

[28] DLR System Dynamics and Control. Cpacs: Common parametric aircraft configuration schema. https://dlr-sl.github.io/cpacs-website/.

[29] Ilya M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1(4):p. 407 ss., 1993.

[30] Andrea Saltelli et al. Variance based sensitivity analysis of model output. *Computer Physics Communications*, 181(2):p. 259 ss., 2010.

[31] SMT Developers. Smt: Surrogate modeling toolbox – documentation. https://smt.readthedocs.io/en/latest/.