POLITECNICO DI TORINO

Master's Degree in AEROSPACE ENGINEERING

Master's Degree Thesis

Fluid-dynamic optimization of tapered bulbs in hydrofoil configuration

Supervisors

Prof.ssa Giuliana MATTIAZZO

Dott. Pietro CASALONE

Candidate

Enrico SCRIBANO

December 2023

Summary

Recent attention in sailing has shifted towards foiling boats, which can rise above the free surface of the water in order to reduce the drag force. This is achieved by placing small wings under the boat, which provide a supporting force that allows the boat to rise. These wings consist of two main surfaces: a vertical surface, which is used to counteract the lateral forces of the sail, and a horizontal surface, which is the source of lift. At the junction of vertical and horizontal surfaces, is made a structural connection. This joint, named Bulb, has a tapered shape in order to reduce its disturbance to the flow.

The purpose of this master's thesis is to establish a process for fluid dynamic optimization of the bulb, to improve the efficiency of the whole wing. To accomplish this, an optimization tool employing genetic algorithms has been developed in MATLAB. The tool explores the design space by evaluating a single objective function through fluid dynamic simulations in STAR-CCM+. To complete the work was explored the possibility of performing a second detailed optimization, only for the best geometries screened by the tool, through adjoint shape optimization, embedded in STAR-CCM+. The thesis was carried out in collaboration with the MOREnergy lab from Politecnico di Torino.

Acknowledgements

Questo percorso giunge finalmente ad una conclusione, portandosi via ciò che sono stato, lasciandomi ciò che sarò. Ritengo doveroso ringraziare chi, in maniera diversa, mi ha aiutato a concludere questa esperienza.

Grazie alla prof.ssa Mattiazzo per avermi permesso di seguire un progetto di tesi su un argomento a me caro e per mantenere vivo il Polito Sailing Team. Ringrazio il Dottor Casalone che mi ha seguito in tutta la realizzazione del lavoro e nella stesura del testo, sempre a disposizione anche da molto lontano. Un grazie anche al dottor Dell'Edera e alla dottoressa Citarella del MORE che mi hanno consigliato in momenti critici.

Non posso non menzionare i miei genitori che hanno contribuito a questo traguardo più di tutti, che mi hanno sempre dato la loro fiducia e hanno messo a mia disposizione tutti i loro mezzi per farmi arrivare qui ora. Avete sempre fatto il massimo per me e non smetterò mai di esservene grato. Mamma, Papà vi voglio bene.

Un grazie a Serena. Condividere l'infanzia è stato un dono meraviglioso e anche ora che siamo cresciuti mi sembra sempre di essere quel bambino che ero un tempo quando sono con te.

Un grazie ai nonni: un punto fisso e centrale nella mia vita, da sempre. Spero di rendervi orgogliosi raggiungendo questo “obbiettivo”. Grazie allo zio che è sempre stato un consigliere speciale.

Grazie al dottor Mastorakis per gli spunti di riflessione e i numerosi consigli condivisi nella stesura di questo lavoro. Un abbraccio agli amici, lontani e vicini.

Infine un pensiero a Dori, ti penso sempre. Mi manchi.

La porta devone essere
correttamente aperte e
richiusa.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The idea behind this thesis work arises due to the lack of publications or guidelines from hydrofoil manufacturers on the design, from a fluid dynamic point of view, of the junctions between lifting surfaces and drifting surfaces. In my years at PoliTo Sailing Team, a student team from Politecnico di Torino that designs and builds sailboats to compete in university regattas, I have been involved in the design and fluid dynamic analysis of different appendages, encountering several times the necessity to understand and realize joint structures between parts with different functions. Since no specific guidelines existed and studies could not be carried out in a very short time due to tight schedules, common sense required that existing geometries should be analyzed and then copied. For this reason, it was thought to use this thesis space to analyze a topic of my interest and that could be useful in future developments for the team.

1.1 Hydrofoil Boats

The concept of foil boats was born at around the same time as the invention of the first airplane. The first hints of it can already be seen in 1869 in England, and in 1898 Italian engineer Enrico Forlanini started developing a series of models of motor boats equipped with small wings. Forlanini obtained several patents for his inventions, which were tested already in 1906 over Lake Maggiore in northern Italy [1].

The development of this technology has been going on for years, expanding even into the passenger transport and military sectors, but its resonance has grown greatly in the sports field through sailing [2]. The integration of hydrofoils into sailing vessels represents a paradigm shift in maritime technology, fostering a novel era characterized by elevated performance and efficiency. Hydrofoils, essentially submerged lifting surfaces, serve as pivotal components in this evolution, facilitating

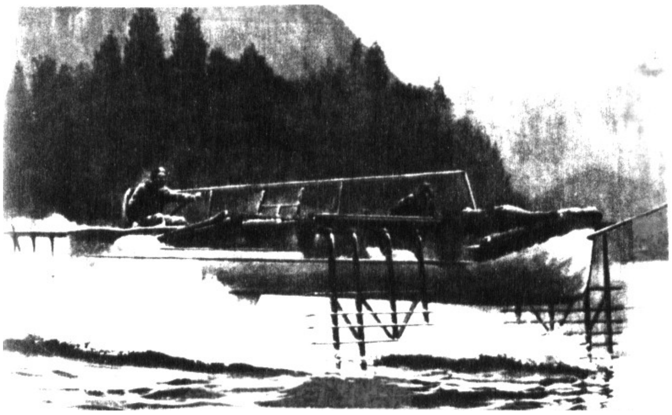


Figure 1.1: Forlanini testing his models over Lake Maggiore, around 1906.

the elevation of sailboats above the water surface to mitigate hydrodynamic drag and enhance overall navigational dynamics. Hydrofoils derive their functionality from the principles of lift and drag mitigation achieved through the manipulation of fluid dynamics. As a sailboat gains velocity, hydrofoils generate lift, causing the hull to rise and essentially "fly" above the waterline. This phenomenon significantly reduces the wetted surface area, minimizing drag and allowing for an augmented ratio of lift to drag, culminating in unprecedented speeds and maneuverability. These hydrofoil-equipped vessels have redefined competitive sailing, particularly evidenced in prestigious events such as the America's Cup, where cutting-edge engineering converges with navigational prowess in a quest for maritime supremacy.

1.1.1 Hydrofoil Configurations

The first distinguishing feature that can be made among the various types of foils is the position of the lifting surface relative to the water level: totally submerged or partially submerged. This positioning affects the use, stability, and possible lift control systems.

Another way of classifying hydrofoils comes from their shape, and they are named after the letters of the alphabet that their front view resembles. The most widely employed configurations are J-shaped and T-shaped foils.

J-shaped hydrofoils adopt a streamlined, single-foil approach that emphasizes lift efficiency and reduced drag. The curvature of the foil, resembling the letter "J," allows for concentrated lift, facilitating higher speeds with minimized hydrodynamic resistance. This configuration is a favorite in the realm of high-performance racing, where every ounce of speed matters. J-foil is a partially submerged configuration.

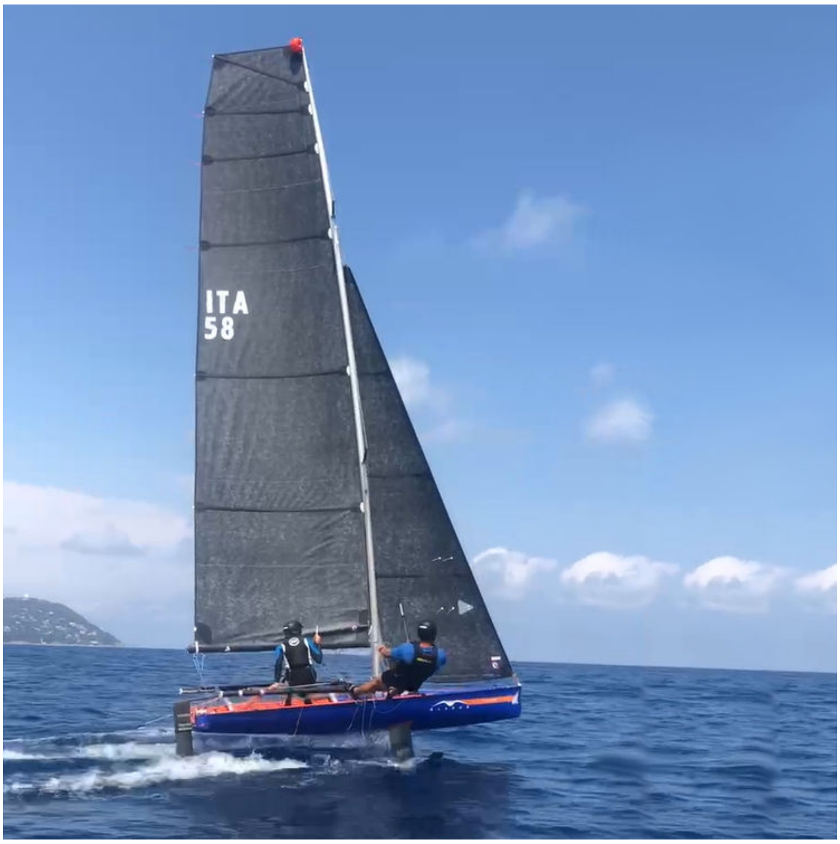


Figure 1.2: Glaros rising above the water surface.

T-shaped hydrofoils represent a versatile synthesis of stability and lift, incorporating a central foil perpendicular to the hull, accompanied by lateral foils on either side. This design strikes a harmonious balance, providing both longitudinal and lateral stability while maintaining lift efficiency. The T-shaped configuration has gained prominence across a spectrum of sailing applications, from cruising to elite racing. Its adaptability makes it an intriguing choice for sailors seeking a middle ground that accommodates various performance demands. Usually, T-shaped foils are a fully submerged configuration. T-shaped foils are used as rudders in most foiling sailboats and also as the main surface in the Moth and Wasp classes.

The majority of hydrofoils require a lift control system. This type of control is necessary to enable navigation in different conditions and to perform specific maneuvers in water or just to rise above the water level. How lift is controlled depends closely on the type of hydrofoil. Generally, partially submerged foils have an active angle of attack control (usually a mechanical system) but also a passive control caused by the surface of the appendage rising out of the free surface of the water. When the lift increases excessively, the foil starts to come out of the

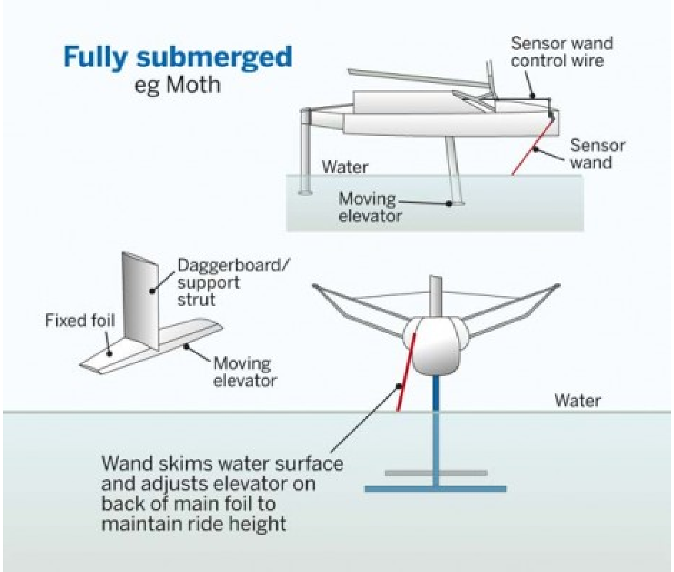

Fully submerged
eg Moth
Sensor wand control wire
Sensor wand
Water
Moving elevator
Daggerboard/ support strut
Fixed foil
Moving elevator
Water
Wand skims water surface and adjusts elevator on back of main foil to maintain ride height

Figure 1.3: A T-shape main foil of a moth and its control system [3].

water. Because of the decrease in density of the fluid in which the structure is submerged, the lift is forced to decrease causing the foil to sink and submerge back into the water. If accurately designed, this procedure allows the lift of the foil to be managed almost autonomously as the boat's speed of travel changes. In contrast, for totally submerged foils, only active controls can be realized. Generally, the method of making the control depends on the position and operation of the foil. Main foils in the moth, wasp or similar categories being positioned in the center of the hull are inaccessible to the sailor. For this reason, a mechanized flap is utilized. The wand, i.e., a rod placed in the bow, is in charge of passing flight height information to a control system that manages the movement of the flap (fig 1.3). In contrast, if a T-foil performs the function of a rudder, its control is managed only by changing its angle of attack, either automatically or handled by the sailor.

This thesis is concerned to study a T-shaped configuration, common to several types of boats, used for the rudder of the PST's skiff Glaros (figure 1.2). Glaros is the team's first foiling skiff, equipped with two J-foils and a T-rudder (figure1.4).

1.1.2 Hydrofoil Junction

Hydrodynamic lifting surfaces must be attached to other components. In T-shaped foils, the lifting surface is attached to the vertical surface which has the functions of a classic drift or rudder. To keep these two surfaces joined, a mechanical interlocking is made, causing one to plug into the other and making a lock through a screw.

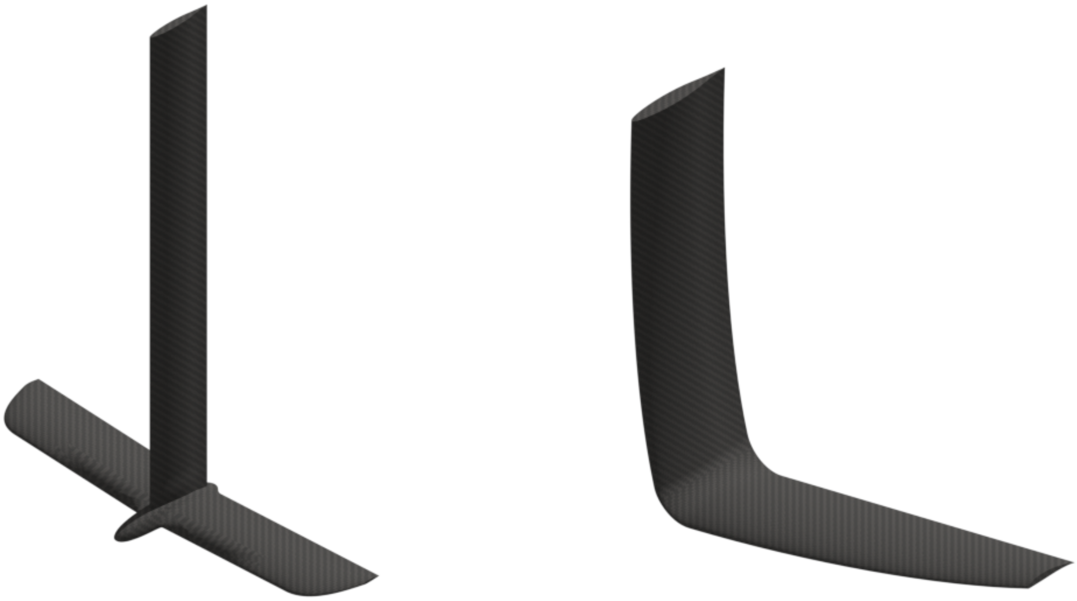


Figure 1.4: Glaros T-shape Rudder and J-foil.

The tightness of this interlock depends on the contact surface between the two foils, so the more lift the horizontal foil generates, the more it will be necessary to increase the surface area of the interlock, leading to an inevitable thickening of the aerodynamic profile of the horizontal foil in that area. This requirement leads to junctions between the lifting surface and another component, adding to the challenge of hydrodynamic lifting surface design.

The shape of the junction affects the pressure distribution and thus cavitation inception and flow separation [4]. If the two coupled surfaces were isolated, two quite different fields of motion would be generated in the area of the coupling. Joining these two areas that are so different from each other imposes strong disturbances to the fluid current around the junction, typically creating recirculation bubbles, strong separations, and possible ventilations (the entrainment of non-condensable gas into the region surrounding the structures) that reduce the efficiency of the surfaces. For these reasons, it is necessary to model the outer shape of the junction in such a way that it is fluid-dynamically the least impacting as possible while serving the identified structural functions. The bulbous shape, often resembling an inverted teardrop, minimizes drag and turbulence by smoothly guiding water flow around the foils.

1.1.3 Bulb existing configurations

There are many geometries in use, as the field is now developing overwhelmingly. Generally, the geometry around the interlocking results in an elongated shape, with a prominent direction toward the direction of motion. These geometries are more elongated for surfaces with tight chords, while they are slightly more blunt in surfaces with bigger chords that consequently generate higher loads.

Usually, the engagement between the two surfaces occurs at the middle of the bulb chord to the trailing edge, but there are also several variations on the topic.

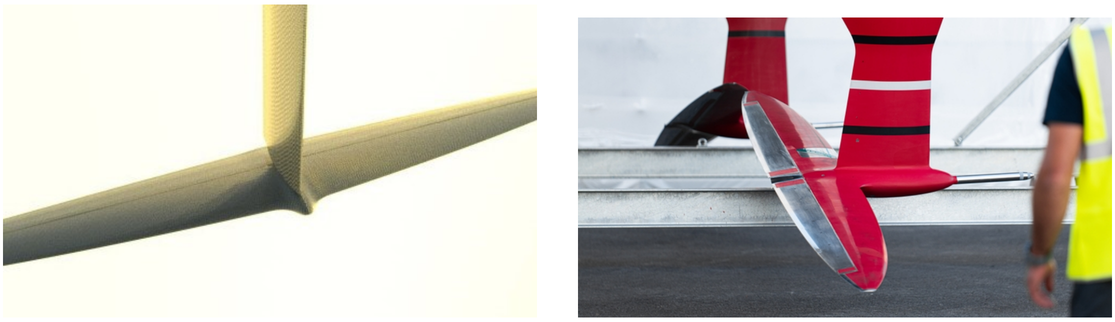


Figure 1.5: Two different joint configurations: a moth main foil (on left) and an Ineos T6's foil (on right).

In figure 1.5 is possible to visualize two different joint configurations: on the left, the graft is made to the back of the bulb; on the right, the vertical surface is embedded outside the lifting surface. Moving the engagement position of the vertical surface back and forward is a classic example of variation in bulb configuration; there are also cases where the horizontal surface is behind the vertical surface. Other possible configurations depend on the thickness and airfoil of the bulb, but these are more difficult to notice with the naked eye and it would be necessary to have sketches to appreciate the variations. Consequently, as all the above configurations change, the geometry in the plan of the leading and trailing edge also varies. These different variants can produce an infinity of possible geometries.

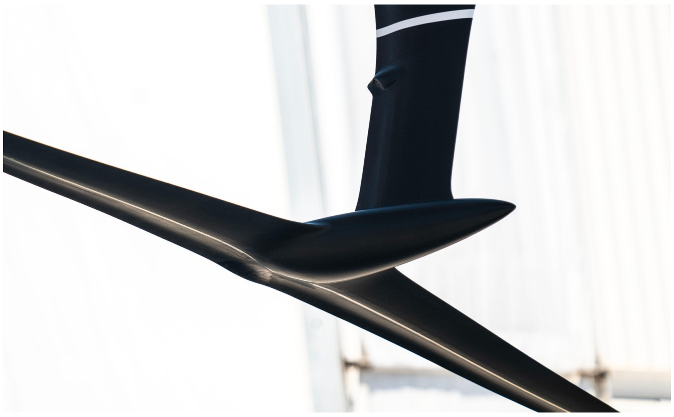


Figure 1.6: Detail of Luna Rossa LEQ12's right foil (2023).

1.2 State of the art of optimization

Optimization in engineering refers to the process of improving a system, process, or project to maximize or minimize certain objectives. In general, the goal of optimization is to find the best possible solution based on specific criteria, considering limitations and constraints. The optimization process typically involves clearly defining objectives, identifying constraints, and finding the optimal combination of variables or parameters. Optimization is a fundamental part of engineering because it enables efficient and cost-effective solutions to a wide range of challenges and problems.

The development of technologies in computing from the viewpoint of algorithms, driven by the development of hardware, has made it possible to handle complex optimization problems that were previously unimaginable. To evaluate certain problems, it is necessary to analyze complex and enormous design spaces, described by a lot of variables. Without an algorithm designed and engineered specifically to perform these tasks, it would be impossible to evaluate every single point in the design space because it would require an excessive amount of processing memory and an extremely long time. The rise of so-called evolutionary algorithms and machine learning has been a driving force for the field of optimization to find powerful, fast, and practical tools for problem-solving.

Fluid dynamic optimization of wings, or generic lifting surfaces, seeks an optimal geometric configuration that improves efficiency, decreases consumption, or improves a wing's performance. As a consequence of the high nonlinearity and complexity of the problems, combined with the absence of an analytical formulation, it is necessary to use optimization algorithms that reduce the number of evaluations to be performed to reach a global minimum or maximum point. Several approaches can be followed to optimize a wing's shape:

- Parametric Design and Iterative CFD Simulations. The optimization can systematically vary parameters defining the wing geometry (sweep angle, span, airfoil, etc) to find the optimal configuration through iterative numerical simulations to evaluate performances and then adjust the geometry for further optimization.

- Numerical Optimization Methods. Numerical optimization algorithms such as evolutionary algorithms or gradient-based algorithms can be used to explore the design space. These methods involve defining an objective function and searching for the wing's geometric configuration that maximizes or minimizes this function but only computing the function for a limited number of configurations.

- Machine Learning-Based Optimization. Machine learning algorithms can learn from CFD simulation results to suggest geometric modifications that enhance aerodynamic performance.

- Wind Tunnel or Tank Experiments: integrating numerical optimization with direct experiments to validate and improve obtained results.

It is common to combine several of these approaches to achieve a comprehensive optimal solution. Developing new high-performing algorithms keeps the field active by expanding the possibility of applying optimization to different objectives or more complex problems.

1.2.1 Optimization of Wings

There are many studies of optimization for airfoils, wings, or tapered geometries (like keel bulbs made for stability reasons). Most of these studies are carried out using numerical simulations coupled with heuristic algorithms or machine learning, which provide for greater freedom of operation and reduced analysis time that would be prohibitive with other techniques. However, the high nonlinearity of the equations that govern the world of fluid dynamics, and consequently fluid dynamics optimization, creates disadvantages in terms of analysis time and computational power: the need to obtain valid (and subsequently validatable) results requires a discretization of the problem with reasonably detailed computational grids, which therefore requires high computational power.

These optimization methods have been tested and used in studies that in some features may approach what is intended in this thesis. In *Design and optimization of rudder bulb based on CFD and genetic algorithm* a genetic algorithm coupled with a CFD solver is used to optimize the propeller of a powerboat bulb, to reduce the energy required to move the boat, resulting in fuel gain. This approach is quite satisfactory to the authors, who achieve a 3.66% increase in the efficiency of the configuration [5]. Genetic algorithms are chosen for their adaptability to complex problems, their ability to handle large search spaces, and their flexible and parallel nature. They are particularly useful when other optimization methods may be limited or inefficient.

Design Space Parameterization

A key aspect to consider when conducting an optimization study is the definition of the design space through a parameterization. Parameterization is a fundamental step of systematically defining geometric and design parameters that can be varied during optimization. These parameters allow a wide range of configurations to be explored to identify the optimal solution that meets certain performance criteria and constraints. Parameterization allows the optimization process to efficiently explore the design space, reducing the complexity of the problem. The use of parameters allows the shape and characteristics of the wing to be adjusted in a controlled manner, providing numerical optimization algorithms or computer-aided design tools the flexibility to explore multiple configurations and find optimal or near-optimal solutions. This approach is particularly useful in aerodynamic design, where complex interactions between geometric shapes and fluid-dynamic flows require careful consideration of parametric variations to achieve optimal performance.

There are several parameterization techniques by which a design space can be represented, and there is no one better than the others in absolute terms; it depends on the type of geometry to be optimized. Some geometries can be represented efficiently by some techniques but not by others. One technique that stands out in the aviation field is CST, which was specially developed by Boeing to easily represent airfoils and wings. An example from which methodological insights can be learned is *Three-Dimensional CST Parameterization Method Applied in Aircraft Aeroelastic Analysis* in which an aeroelastic analysis of a complete reusable reentry aircraft is performed. To model the geometry of the aircraft, the authors chose a simple parameterization that was capable of describing both the wings and the fuselage with a rounded nose [6]. The technique used is the already mentioned CST, which was created specifically to represent complex geometries useful in aviation. The potential of CST is the ability to reduce the number of parameters describing a wing or fuselage compared to classical drawing methods. This potential can be exploited in an optimization to reduce the number of free parameters and decrease the dimension of the design space.

Chapter 2

Geometry Parameterization

The initial stage in generalizing multiple definitions of bulbs involves parameterizing all shared geometric features to represent, with a limited number of parameters, a versatile design space to optimize the desired geometry. To achieve this goal, the choice of CST parameterization was considered the best, as it has been developed specifically for aircraft geometries and has been widely used and tested for various optimization purposes.

2.1 CST Parameterization

The CST parameterization technique is a technique developed in the 2000s by B. Kulfan for Boeing Commercial Airplane Group, to break free from previous mathematical definitions of aircraft surfaces (in the first instance airfoils) since the use of discrete variables as design coordinates can create rather convoluted and angular geometries. Aware of the fact that the type of parameterization can affect the optimum point in the design space, the CST technique has been developed to:

- produce smooth geometries and realistic shapes; is mathematically efficient and numerically stable;
- be mathematically efficient and numerically stable;
- requires relatively few variables to represent a large enough design space to contain optimum aerodynamic shapes for a variety of design conditions and constraints;
- allows specification of design parameters such as leading-edge radius, boat-tail angle, and airfoil closure;
- provides easy control for designing and editing the shape of a curve;

- includes an intuitive geometry algorithm that has an intuitive and geometric interpretation [7].

The CST deals with two main class of shapes: the first includes wings and airfoils while the second includes cross-section-type shapes (aircraft fuselage, channels, and ducts). The mathematical description of class 1 geometries, having a round nose and pointed aft-end, is a continuous but non-analytic function because of the infinite slope at the nose and the corresponding large variations of curvature over the surface. Similarly, in the conventional Cartesian coordinate system, the mathematical definitions of the cross sections of class 2 type of geometries are generally also a continuous but non-analytic function. Consequently, a large number of coordinates are typically required to describe either class 1 or class 2 types of geometries.

2.1.1 Bi-Dimensional CST

For a typical rounded-nose airfoil, in a fixed Cartesian coordinate system, the first and second derivatives are infinite at the nose and large changes in curvature occur all over the airfoil, leading to a non-analytic function with singularities in all derivatives at the nose when mathematically represented. To avoid this problem, CST defines a Shape Function transformation by which the definition of airfoil becomes a well-behaved analytic function with efficient design features. Later the Class Function is introduced to generalize the method and apply it to many two-dimensional geometries and as a section of axisymmetric bodies, like nacelles.

The general form of the mathematical expression for the typical airfoil [8] is

$$\zeta(\psi) = \sqrt{\psi}(1-\psi)\sum_{i=0}^{N} A_i \psi^i + \psi\zeta_T \tag{2.1}$$

where ψ and ζ are the non-dimensional x and z coordinates $\psi = x/c$, $\zeta = z/c$ meanwhile $\psi_T = \Delta Z_{TE}/c$ provides the control of the trailing-edge thickness and the term

$$\sum_{i=0}^{N} A_i \psi^i$$

represent a function that describes the geometry between the nose and the end. This function can be represented by any appropriate analytic, well-behaved mathematical function. The term $\sqrt{\psi}$ is the only term that provides a round trailing edge, but it's also the source of non-analytical features.

At this point, the CST define the **Shape Function** reshaping the equation 2.1, to obtain

$$S(\psi) = \frac{\zeta(\psi) - \psi\zeta_T}{\sqrt{\psi}(1-\psi)} \tag{2.2}$$

The S function can also be viewed as

$$S(\psi) = \sum_{i=0}^{N} A_i \psi^i \tag{2.3}$$

The shape function is a well-behaved equation that can be adapted to represent the trend of the geometry between the nose and the end and can be chosen as simply as possible, respecting the constraints imposed by the leading and trailing edges. In fact, the value of S is related to the design parameters in the following ways:

$$S(0) = \sqrt{2R_{LE}/c} \qquad S(1) = \tan\beta + \Delta Z_{TE}/c \tag{2.4}$$

The value of S in $x/c = 0$, is dependent on the nose radius, and in $x/c = 1$ the function follows the airfoil tail angle, β and the trailing edge thickness. Figure 2.1 shows an example of an airfoil transformation. As can be seen, in transformed coordinates, the function turns out to be a rather straight line with a value at zero and at the end equal to what was mentioned earlier. Also shown, in blue, the constant thickness curve equal to the maximum thickness of the example's airfoil, and in green the constant boat-tail angle line.

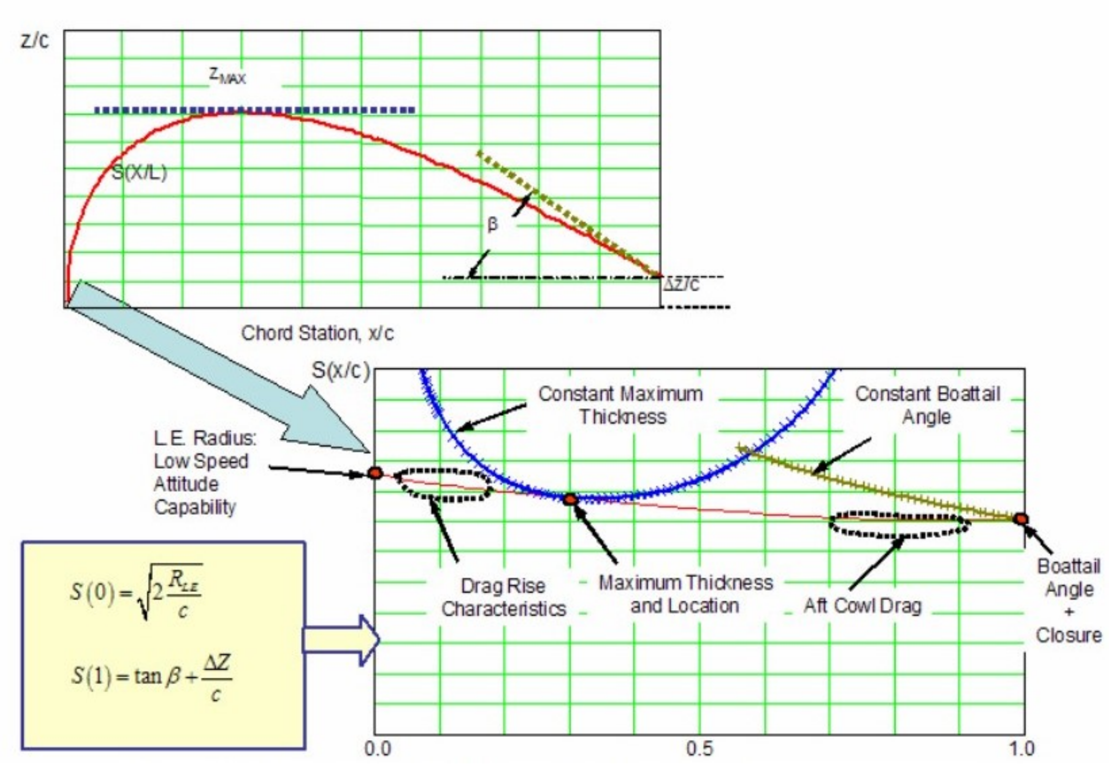

z/c
Z_MAX
S(X/L)
β
ΔZ/c
Chord Station, x/c
S(x/c)
L.E. Radius: Low Speed Attitude Capability
Constant Maximum Thickness
Constant Boattail Angle
Drag Rise Characteristics
Maximum Thickness and Location
Aft Cowl Drag
Boattail Angle + Closure
$S(0)=\sqrt{2\frac{R_{LE}}{c}}$
$S(1)=\tan\beta+\frac{\Delta Z}{c}$
0.0
0.5
1.0

Figure 2.1: Example of an Airfoil Transformation using the shape function [8] .

The term $\sqrt{\psi}(1-\psi)$ can be generalised introducing the **Class Function** C as

$$C_{N2}^{N1} = \psi^{N1}(1-\psi)^{N2} \tag{2.5}$$

By varying the $N1$ and $N2$ coefficients, completely different geometric shapes can be obtained, from a classic shape such as a rectangle or circle to a biconvex airfoil or a typical commercial NACA airfoil. For a round nose airfoil $N1 = 0.5$ and $N2 = 1.0$ [8].

Due to the definition of the class and shape functions, it is now possible to describe an airfoil by them, without resorting to the point coordinates of its points

$$\zeta(\psi) = C_{N2}^{N1}(\psi)S(\psi) + \psi\zeta_T \tag{2.6}$$

The Bernstein Polynomials

The Shape Function in equation 2.3 has not yet been described. It should be a relatively simple function that defines the airfoil between the nose and tail.
The unit shape function $S(\psi) = 1$ can also be decomposed into two different functions, $S_1(\psi) = 1-\psi$ and $S_2 = \psi$. The first one defines a geometry with a round nose and a zero boat-tail angle, meanwhile the last describes an airfoil without nose but with a finite boat-tail angle. Introducing a coefficient KR in the definitions of S_1 and S_2 allows the amplitude of the terms defining the leading and trailing edges to vary. This makes possible to generate families of airfoils with different leading edge radii, trailing edge angle and point of maximum thickness. All these parameters can be controlled by the only coefficient KR. The unit shape function and its decomposition into two different components can be interpreted as the zero-order and first-order Bernstein Polynomials. The general definition of n-th grade's Bernstein Polynomial BP_n is

$$BP_n = \sum_{i=0}^{n} K_{i,n} t^i (1-t)^{n-i} \qquad t \in [0,1] \tag{2.7}$$

in which the factors $K_{i,n}$ are binomial coefficients defined as

$$K_{i,n} = \frac{n!}{i!(n-r)!} \tag{2.8}$$

Bernstein Polynomials have the property of partition of unity, so for every order n, the equation 2.7 is equal to 1.

$$BP_n = \sum_{i=0}^{n} K_{i,n} t^i (1-t)^{n-i} = 1$$

Due also to this property, growing the degree n of Bernstein polynomials, which correspond to the sum of $n+1$ terms, means systematically increasing the

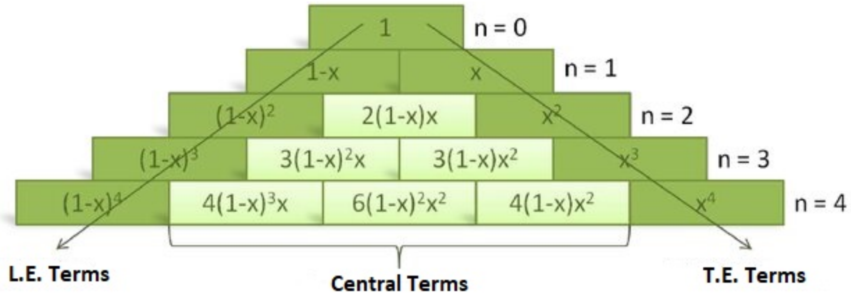


Figure 2.2: The growth of the n-th order of Bernstein Polynomials and their analytical definition.

decomposition of the shape function into various components. Since only the first component of the polynomials affects the L.E. and only the last component affects the T.E., this procedure improves the development of only the central part of the airfoil, leaving the nose and trailing edge unaffected.

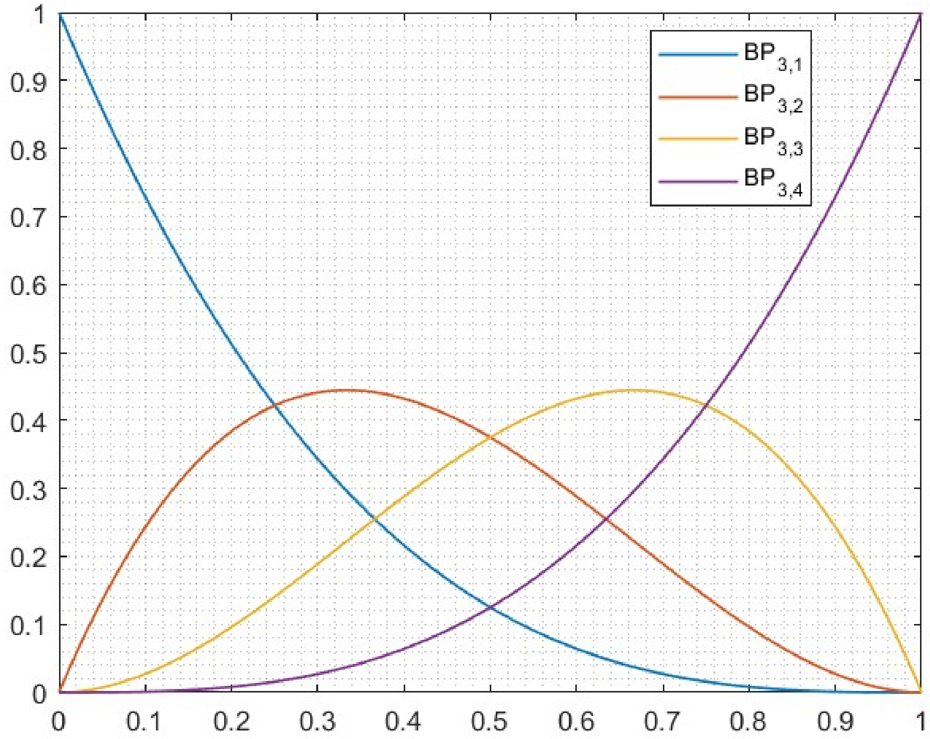


Figure 2.3: Bernstein Polynomials, $n = 3$

2.1.2 Airfoil Definition

The knowledge of the powerful decomposition in shape and class functions makes it possible to represent any airfoil's upper or lower surface using the Bernstein polynomials of selected order n, by describing a group of component shape functions that can be scaled by some factors A. The global shape function of the upper surface is

$$Su(\psi) = \sum_{i=1}^{n} Au_i S_i(\psi) \tag{2.9}$$

that can lead to the definition of the equation of the upper surface

$$\zeta_u(\psi) = C_{N2}^{N1} Su(\psi) + \psi \Delta\zeta_u \tag{2.10}$$

Similarly, the lower surface shape function is

$$Sl(\psi) = \sum_{i=1}^{n} Al_i S_i(\psi) \tag{2.11}$$

and the equation of the lower surface becomes

$$\zeta_l(\psi) = C_{N2}^{N1} Sl(\psi) + \psi \Delta\zeta_l \tag{2.12}$$

where $\Delta\zeta$ is the non-dimensional trailing edge thickness ratio, distinctly between upper surface and lower surface ($\Delta\zeta_u = zu_{TE}/c$).

The multiplicative factors A involved in defining the equations 2.9 and 2.11, which change the behavior of the various components of the shape function, are two arrays of $n+1$ values each that properly define the two surfaces of the airfoil. These coefficients Au_i and Al_i can be determined in a variety of ways, such as by interpolating the contour curves using a least-squares method, and their value depends on the degree of the Bernstein polynomials that are used to define the airfoil. Thus proceeding, to define an airfoil with Bernstein polynomials of the n degree will require only $2n+2$ multiplicative coefficients, $n+1$ for the upper surface and $n+1$ for the lower surface, versus the many pairs of points needed to describe the airfoil with classical representations.

Studies conducted by Kulfan and Bussoletti [8] show how shape functions computed with third-order polynomials interpolate airfoil such as RAE2822 with absolute residuals below 3×10^{-3}, and how the error decreases as the order of the polynomial increases. As for the computation of pressure distributions around the airfoil with CFD methods, satisfactory levels of accuracy can already be achieved with second-order polynomials, and with sixth-order polynomials an almost complete match is achieved. Thus, these results demonstrate how the technique is very convenient in the design phase, minimizing the number of parameters for curves description but retaining the ability to describe almost all existing airfoils and being able to describe new ones that are not commercially available.

2.2 Three-Dimensional CST

Three-dimensional bodies generally can also be represented as distributions of their cross-section along the orthogonal axis. Taking advantage of this property, 3D geometries can be defined by modifying the CST technique to describe the two perpendicular sections and studying the distribution of one along the path described by the other. For a three-dimensional wing, it is possible to define the wing's airfoil using the CST technique shown above and then extrude it along a third dimension by following a curve that describes its wingspan, also defined using the CST technique.

Similar to the shape function of an airfoil, the shape function of a wing is an analytic continuous surface that can thus be described as a double Taylor series in x and y. It can be demonstrated how this Taylor series is equivalent to a first expansion in x whose coefficients are in turn described as an expansion in y [9]. This means that the wing is defined through interpolation of a set of airfoils described by Bernstein polynomials of degree Nx, in accordance with equation 2.10 and 2.12. This interpolation is done through Bernstein polynomials of order Ny that also control the magnitude of each airfoil along the spanwise direction. It can also be seen as placing $Nx + 1$ airfoils, previously chosen, in selected sections of the wing and defining the behavior of the intermediate sections by interpolating the points of neighboring airfoils with a polynomial in y.

The mathematical formulation of what has been explained in the previous lines can be broken down as follows.

The unit shape function in stream-wise direction for Bernstein polynomial of Nx-th grade can be defined as

$$Sx_i(\psi) = Kx_i \psi^i (1-\psi)^{Nx-i} \qquad i = 0 : Nx \tag{2.13}$$

where ψ is the fraction of the local chord

$$\psi = \frac{x - x_{LE}(\eta)}{c(\eta)} \tag{2.14}$$

and Kx_i are the stream-wise binomial coefficients

$$Kx_i = \frac{Nx!}{i!(Nx-i)!} \tag{2.15}$$

At this point is possible to represent the stream-wise upper (similarly, lower) surface shape function at the reference span-wise section η_{ref}, in which η is the non-dimensional y coordinate $\eta = 2y/b$ (b is the span),

$$Su(\psi, \eta_{ref}) = \sum_{i=1}^{Nx} Au_i(\eta_{ref}) Sx_i(\psi) \tag{2.16}$$

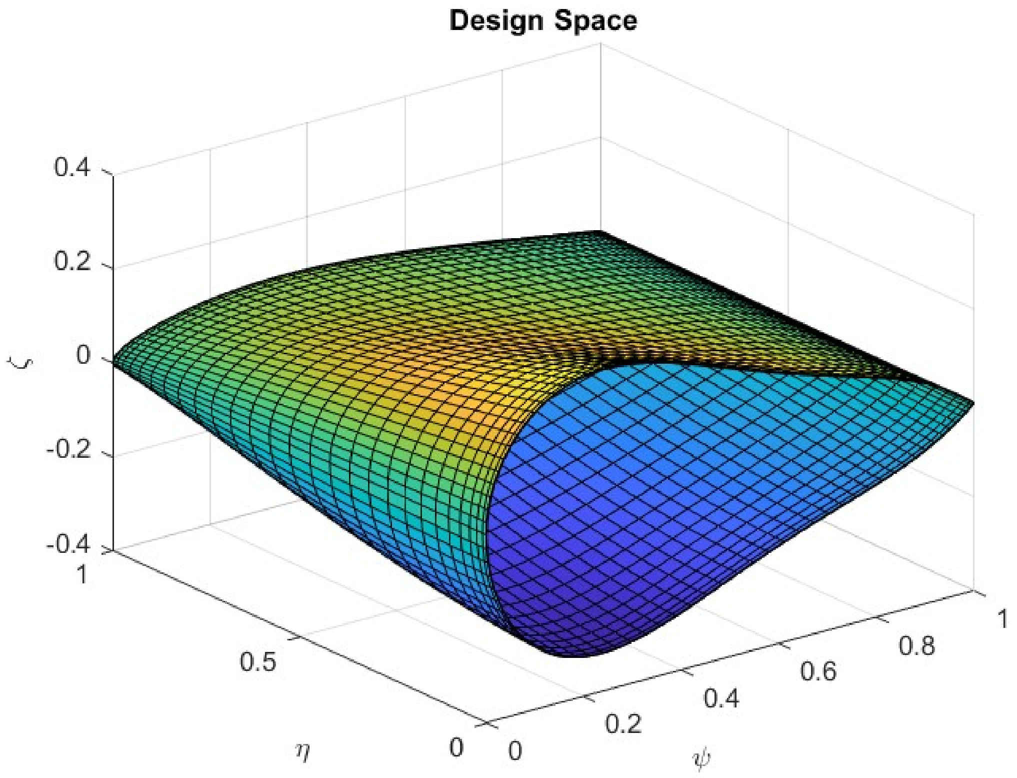


Figure 2.4: Representation, in design space, of a wing designed with CST technique and random $B_{i,j}$ coefficients.

It's now necessary to represent the span-wise variation of each $Au_i(\eta)$ coefficient by Bernstein polynomials of Ny order

$$Au_i(\eta) = \sum_{j=1}^{Ny} Bu_{i,j} Sy_j(\eta) \tag{2.17}$$

where, similar to Sx_i, Sy_j are the span-wise upper (lower) shape functions

$$Sy_j(\eta) = Ky_j \eta^j (1-\eta)^{Ny-j} \qquad j = 0 : Ny \tag{2.18}$$

and, as before

$$Ky_j = \frac{Ny!}{j!(Ny-j)!} \tag{2.19}$$

All these equations lead to the final definition of the wing's upper surface, in non-dimensional coordinates

$$\zeta_u(\psi,\eta) = C_{N2}^{N1}(\psi) \sum_{i}^{Nx} \sum_{j}^{Ny} \Big[Bu_{i,j} Sy_j(\eta) Sx_i(\psi) \Big] + \psi \Big[\zeta_t(\eta) - \tan\Big(\alpha_t(\eta)\Big) \Big] \tag{2.20}$$

in which α_t is the local twist angle.

The equation is similar for the lower surface

$$\zeta_u(\psi,\eta) = C_{N2}^{N1}(\psi)\sum_{i}^{Nx}\sum_{j}^{Ny}\Big[Bu_{i,j}Sy_j(\eta)Sx_i(\psi)\Big] + \psi\Big[\zeta_t(\eta) - \tan\Big(\alpha_t(\eta)\Big)\Big] \tag{2.21}$$

The actual complete wing can now be identified by four equations

$$\begin{cases} x = \psi c_{loc}(\eta) + x_{LE}(\eta) \\ y = \frac{b}{2}\eta \\ z_u = \zeta_u(\psi,\eta)c_{loc}(\eta) \\ z_l = \zeta_l(\psi,\eta)c_{loc}(\eta) \end{cases} \tag{2.22}$$

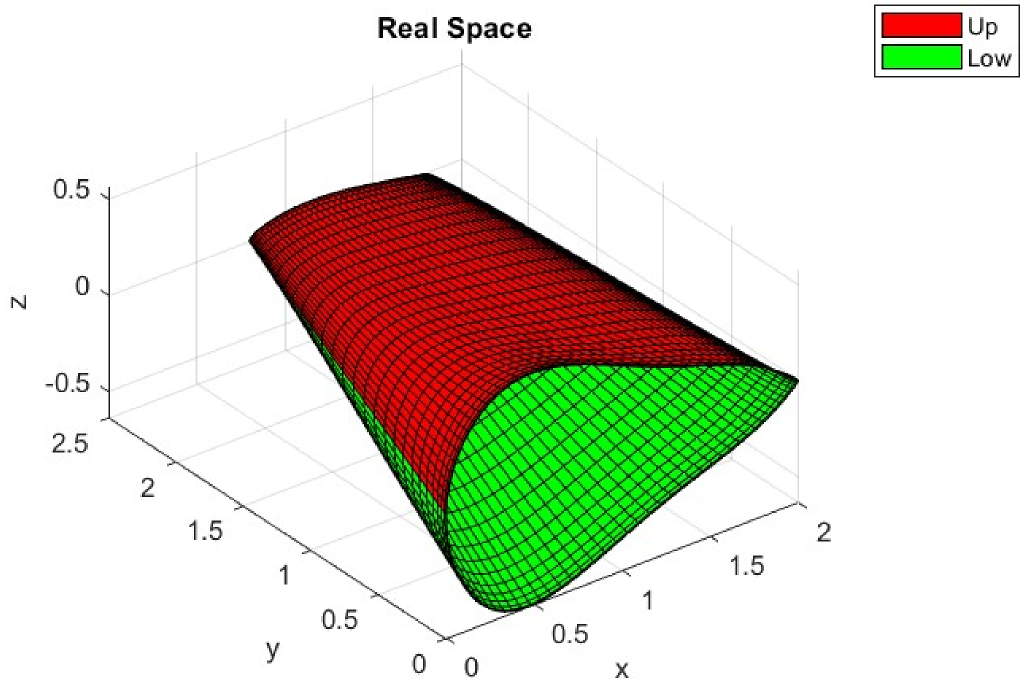


Figure 2.5: Representation, in real space, of a wing designed with CST technique and random $B_{i,j}$ coefficients.

2.2.1 The matrix of Bernstein coefficients

Similar to the two-dimensional case for the Bu_i and Bl_i coefficients, the Au_i and Al_i coefficients, which intervene in equation 2.16, represent the $Nx + 1$ Bernstein coefficients that multiply the shape function components describing the profile behavior of the airfoil at the η_{ref} section. Since it is necessary to serially expand, in the direction orthogonal to the profile, the shape functions, Au and Al coefficients are described as in equation 2.17, shown below for convenience, via the shape functions Sy.

$$Au_i(\eta) = \sum_{j=1}^{Ny} Bu_{i,j} Sy_j(\eta)$$

In the definition of Au, and also Al, are included the coefficients $Bu_{i,j}$, which are those that can actually be controlled by optimization. These coefficients can be most easily visualized as the elements of a matrix of size $(Nx + 1) \times (Ny + 1)$, where the elements in the column represent the Bernstein coefficients of the airfoil at the i-th section and the elements in the row the coefficients describing the j-th span-wise section. In this way, it is easy to see how the CST technique greatly reduces the parameters describing the three-dimensional wing by reducing the control of the profile in the design space to the matrix entries only.

2.2.2 Wing CST's parameters

It was shown that the CST technique is very convenient and powerful in the design phase as it is capable of representing a wing through the use of simple analytical functions controlled by a small number of control parameters. Unlike classical methods of representing aerodynamic surfaces, which represent wings as a set of three-coordinate points with rules describing how to connect them, the CST technique can be thought of as a "scalar loft" [9] where all the points are connected accurately by analytical equations.

To summarize, a single-profile tapered wing can be represented via CST by the following parameters:

- $Nx + 1$ coefficients of Bernstein polynomials for each upper and lower surface;
- wing span b;
- root chord c_r;
- taper ratio TR;
- leading edge sweep.

2.3 Bulb Design Parameters

The goal of this parameterization is to describe the main geometric characteristics of a typical tapered bulb of a racing boat rudder by describing it in as much detail as possible but minimizing the free parameters, i.e., the terms of the actual optimization. A design space large enough to contain the optimization optimum must be created, but described by parameters that are easily conceptualized and realizable from a manufacturing point of view. Before exploring the possible dimensions of this space, it is necessary to go on to define the constraints that limit it, to understand the possible maneuvering space and the best tactics for parametric surface definition.

Since the rudder's overall geometry is symmetrical to a plane that passes through the daggerboard chord, the bulb is divided into perfectly mirrored halves. In line with this property, the definition of the bulb and its parameterization is carried out only for the right side of the surfaces, the remaining part is mirrored through the plane of symmetry (x,z). By reducing the surface to be generated, it is possible to use larger orders of the Bernstein polynomials than would be required to generate the entire surface, while maintaining fewer design parameters.

2.3.1 Constrained Parameters

Each bulb, developed in MATLAB, consists of two surfaces: an upper and a lower one. Each of the two surfaces is described through $pts_x \cdot pts_y$ points.

Parameter	*Value*
pts_x	50
pts_y	60

Orders of Bernstein Polynomials

The first parameter to be fixed to use the CST technique, for this purpose, is the orders of the Bernstein polynomials used to describe the bulb surfaces. The idea driving the choice of these two parameters is always the same: to represent the geometry as detailed as possible (considering that it is still an optimization) but using as few parameters as possible. It should be remembered how the choice of two orders Nx and Ny carries the need to have $2 \times (Nx+1) \times (Ny+1)$ free parameters, in the form of coefficients $B_{i,j}$, to be chosen to define both surfaces. As described in the subsection 2.1.1, it is possible to represent an airfoil, in considerable detail, with a BPO3. Using a BPO3 for the stream-wise direction and a BPO2 for the span-wise direction requires $2 \times (3+1) \times (2+1) = 24$ free parameters. However, in the next subsection, some of these will be imposed by setting some constraints on the final sections.

Parameter	*Value*
Nx	3
Ny	2

External Geometry

Since the goal of this thesis is to create an optimization tool for the bulb alone and not for the entire boat's rudder, the most important constrain is enforced by the geometry "outside" the bulb, i.e., the rudder itself. Taking advantage of the CST technique, the bulb has been defined as an independent geometry, exploiting the structural concept of the bulb as the coupling zone of the appendage's lifting components. The lateral interfaces, to the left and right of the bulb, must exactly match the lifting half-wing, while vertical insertion is achieved by interpenetration of the surfaces.

This constraint can be expressed by imposing the ultimate section of the bulb equal to that of the final geometry. Imposing the same section means imposing the tip chord of the bulb equal to the chord of the first section of the wing and the same airfoil. To obtain this goal, the CAD of the rudder of the boat "Glaros" was divided into four parts: the vertical foil, two horizontal half-wings and the bulb. The two half-wings have the same airfoil along their entire wingspan and it is the **RAF38-il** airfoil, with a chord at the beginning equal to $c = 0.19m$. As mentioned, the chord is easily translated into a physical constraint: the chord at the tip c_t of the bulb must be equal to the half-wing first chord. To obtain the same airfoil of the half-wing it is necessary to impose a constant column of the matrix of Bernstein polynomials instead, in this case, the last one that is responsible for the airfoil at the bulb tip. Thus proceeding, as mentioned earlier, we consolidate $2 \cdot (Nx + 1)$ parameters of the Bu and Bl matrices by reducing the free parameters responsible for controlling the surfaces from 24 to 16.

By a simple interpolation between the RAF38-data-by-point profile and the unit shape functions (BPO3), the Bu_i and Bl_i coefficients can be found, which are used to determine it by the CST technique with the equations 2.10, 2.12. These eight parameters are shown, with all the other constrained parameters in table 2.1.

2.3.2 Free Parameters

Fixed the main constraints on the design space, it is necessary to establish the parameters that describe the geometry of the bulb. These parameters will be the

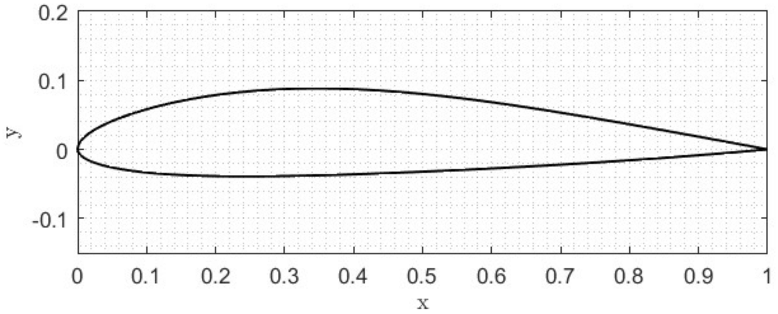


Figure 2.6: RAF38-il Airfoil, used for the Glaros rudder foil.

Parameter	*Value*
Bu_1	0.180
Bu_2	0.278
Bu_3	0.205
Bu_4	0.188
Bl_1	-0.130
Bl_2	-0.085
Bl_3	-0.085
Bl_4	-0.090
c_t	$0.19m$
Nx	3
Ny	2
pts_x	50
pts_y	60

Table 2.1: Constrained parameters in bulb's definition.

subject of the optimization and therefore should be carefully chosen so as not to overwhelm the calculation, but to be able to describe the geometry as completely as possible.
The free parameters defining the bulbs described by the following parameterization can be divided into three categories:

- the coefficients of the matrix of Bernstein polynomials;
- the parameters defining the projected surface;
- the control over the position of the vertical surface of the rudder.

Each of these categories will be analyzed separately below.

Matrix of Bernstein Polynomials

As we have seen, using the degrees of the Bernstein polynomials $Nx = 3$ and $Ny = 2$ to define two surfaces requires the use of two matrices Bu and Bl, both of size $(Nx + 1) \times (Ny + 1)$. In this case, 24 free parameters are generated. However, as 8 of these parameters, two pairs of 4, are required to force the airfoil on the bulb tip, the number is reduced to 16 Bernstein coefficients. Consequently, the Bu and Bl matrices for each bulb will all present the same structure, explained in table 2.2 and table 2.3.

Bu		
$Bu_{1,1}$	$Bu_{1,2}$	0.180
$Bu_{2,1}$	$Bu_{2,2}$	0.278
$Bu_{3,1}$	$Bu_{3,2}$	0.205
$Bu_{4,1}$	$Bu_{4,2}$	0.188

Table 2.2: Bu matrix structure. It represents the coefficients of the Bernstein polynomials for the upper surface of the bulb.

Projected Surface

The geometric parameters that define the projected area (figure 2.7) are designed so that they describe the projection completely but in its essential features. The first are lengths that define the surface area of the bulb projection, while others are more elaborate. Starting with the basic ones, the area is definitely described by a span (half-span since only half a bulb is defined) b and by a chord at the root c_r representing the maximum chord of the bulb, at the plane of symmetry.

Bl		
$Bl_{1,1}$	$Bl_{1,2}$	-0.130
$Bl_{2,1}$	$Bl_{2,2}$	-0.085
$Bl_{3,1}$	$Bl_{3,2}$	-0.085
$Bl_{4,1}$	$Bl_{4,2}$	-0.090

Table 2.3: Bl matrix structure. It represents the coefficients of the Bernstein polynomials for the lower surface of the bulb.

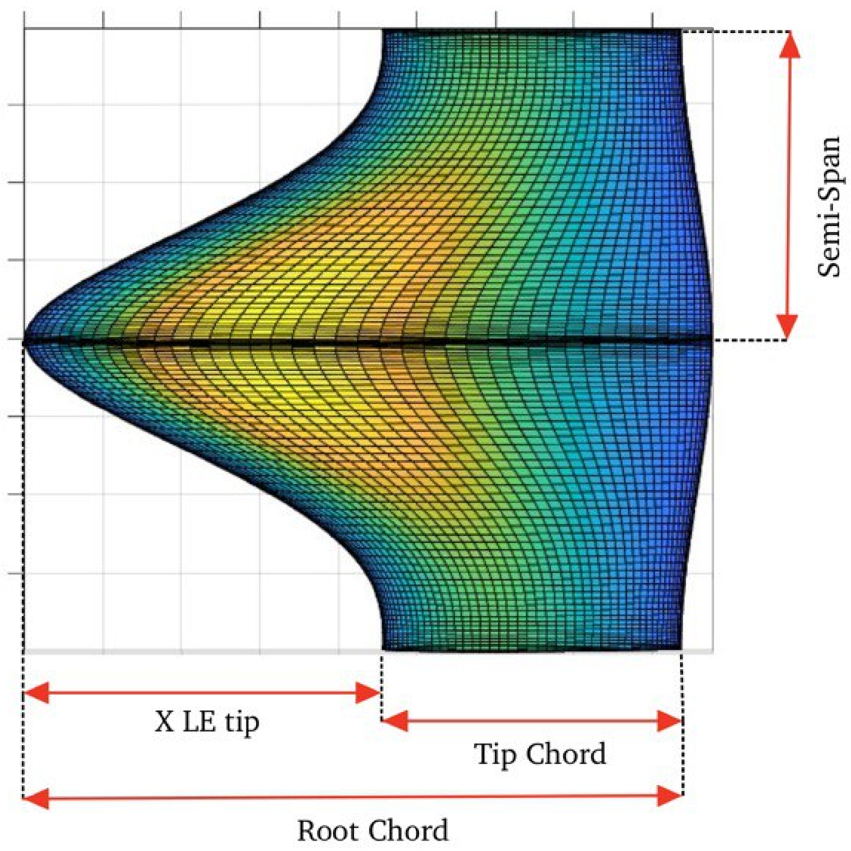


Figure 2.7: Projected Surface of a typical bulb.

Some parameters describing the behavior of local chords in the span direction remain to be defined. To better understand the choices made, it is useful to recall the equation 2.22 that describes the behavior of the variable x.

$$x = \psi c_{loc}(\eta) + x_{LE}(\eta)$$

The x-position of the points describing the bulb surface depends, in addition to the dimensionless variable ψ, on two variables: the local chord c_{loc} and the x_{LE} distance between the trailing edge of the reference chord and the y-axis (passing through the nose of the bulb and orthogonal to the root chord). Both in turn are functions of the dimensionless coordinate η.
One of the possible choices would be to create a variable-parameter function that defines the behavior of the line defining the trailing edge, so as to define both the intermediate chords and the distance x_{LE}. A good result could be obtained by drawing the TE line using the CST technique. However, this would require a minimum of four free parameters (representing the curve at most as a cubic). To allow variability to the line describing the LE, two different parameters were then devised: an $x_{LE}(tip)$ parameter, the distance between the y-axis and the leading edge at the bulb tip, and a flag variable called LE_{flag} that assigns via a discrete value an already written function describing the trailing edge curve.
Acting in this way defines the position of each point on the leading edge by an analytic function selected from a list of possible functions. The trailing edge TE is always computed by spline. This spline requires no additional parameters since it interpolates the trailing edge curve by imposing that the point on the symmetry plane (TE of the chord at the root) is at the point $(C_r; 0)$ and the TE at the tip is at the point $(x_{LE}(tip) + c_t; b/2)$. The slope at the outer points of the spline are imposed equal to zero, forcing continuity of the derivative at the junction with the outer half-wing.

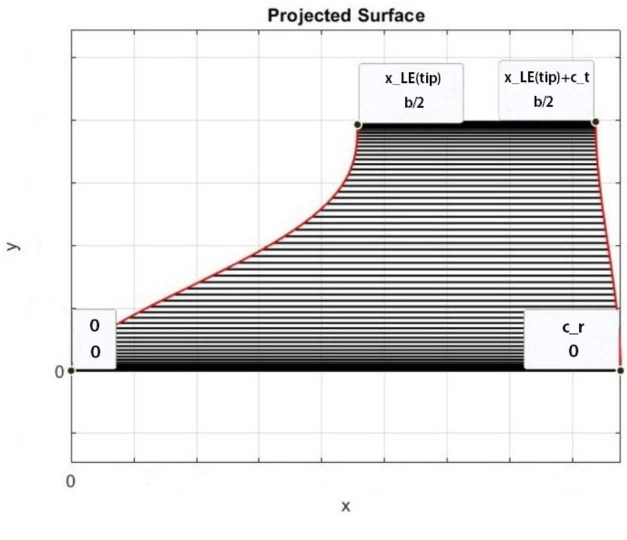


Figure 2.8: Projected surface with reference points.

Having a control over the x-position of the leading edge at the bulb tip gives a control over the coupling point of the half-wing and allows much variability in design. This was felt to be important since there are many rudders on the market for which the wing is shifted backward or forward from the dagger-board insertion position, so by acting this way is also possible to screen this possibility.

Leading Edge Functions

A decision was made to represent the leading edge line using several functions to allow for a variety of curves while using as few variables as possible. The integer optimization variable LE_{flag} is directly linked to a function. The library of functions includes seven options, but can be easily extended in the future. The functions are listed in Appendix A.
Defined the LE and TE trends along the y direction, the local chords of equation 2.22 are simply calculated as

$$c_{loc}(y) = x_{TE}(y) - x_{LE}(y) \tag{2.23}$$

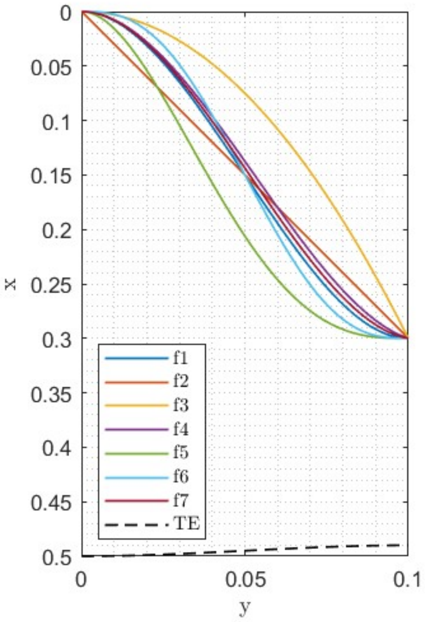


Figure 2.9: Functions describing LE and TE for random parameters: $C_r = 0.5m$, $b = 0.2m$ and $x_{LE}(tip) = 0.3$

Vertical Surface Position

The last parameter necessary for the complete definition of the bulb is the coupling position of the drift surface. This parameter was considered important to include in the optimization because it is able to totally shift the pressure and velocity distribution around the bulb. In addition, what was explained in Chapter 1 about the hydrodynamic function of the bulb places it as an essential variable in view of first optimization. Thus, the parameter that controls the x-position of the vertical surface, x_{vert}, is the distance between the nose of the bulb (point (0,0)) and the first x-point of the daggerboard. Obviously, to maintain symmetry this surface is not moved sideways.

To summarize, the parameterization of the case study is carried out with twenty-one free parameters: twenty describing the bulb and one defining the position of the vertical surface. They are shown in table 2.4.

Name	**Number of variables**
Bu	8
Bl	8
b	1
c_r	1
$x_{LE}(tip)$	1
LE_{flag}	1
x_{vert}	1
Total	21

Table 2.4: Free geometry Parameters.

Chapter 3

Hydrodinamic Modeling

To define a CFD simulation, it is necessary to introduce the concepts behind a numerical simulation by describing the numerical models that allow solving a fluid dynamic problem using computational code. Computational fluid dynamics represents a type of computational science that has its roots in the application of numerical methods to problems of fluid mechanics. Nowadays, CFD is widely used in the fields of aerospace and marine engineering as it is characterized by a high versatility. All Fluid dynamics softwares are organized into three sections by which the simulation flow is defined. The first section is the **Pre-processing** where the geometric parameters, boundary conditions, and discretization of the problem are managed. The second section is the **Solver** which handles the numerical models for solving the governing equations of the fluid dynamics. Last is the **Post-processing** stage, which consists of processing the results obtained.

3.1 Navier-Stokes Equations

The laws that govern the behavior of a generic fluid are the **Navier-Stokes** equations: a system of partial derivative differential equations joined by appropriate initial and boundary conditions. There are three equations: the mass conservation law, the momentum equations (three in total, one for each space direction), and the energy equation. For a Newtonian fluid, neglecting the contribution of volume viscosity (Stokes assumption) and considering Fourier's law for heat flows, the equations are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \tag{3.1}$$

$$\rho\left(\frac{\partial V}{\partial t} + V \cdot \nabla V\right) = -\nabla p + \nabla \cdot \left[\mu(\nabla V + \nabla V^T) - \frac{2}{3}\mu(\nabla \cdot V)I\right] + \rho f \tag{3.2}$$

$$\rho\left(\frac{\partial E}{\partial t}+V\cdot\nabla E\right)=-\nabla\cdot(pV)+\nabla\cdot\left\{\left[\mu(\nabla V+\nabla V^{T})-\frac{2}{3}\mu(\nabla V)I\right]\cdot V\right\} \\ +\rho f\cdot V+\nabla\cdot(k\nabla T)+Q_v \tag{3.3}$$

In which ρ is density, V the velocity vector field, p pressure, E total energy, T temperature, k thermal conductivity, ρf the surface forces and Q_v the volume sources. For the derivation of these equations, reference is made to [10].

These equations constitute a strongly nonlinear system of partial derivative differential equations for which there is no solution since it consists of 5 equations in 7 unknowns, resulting in an ill-posed problem. It should be noted that these equations are evolutionary in form and cannot be solved in closed form except in some simplified cases. So, the Navier-Stokes equations must be solved iteratively by discretizing the control volume into a finite number of elements.

3.2 Turbulent Flows Modeling

The study of this thesis has as its objective the optimization of a complex geometry through an algorithm. Therefore, it is important to control the computational cost of the simulations performed, which are the most time-consuming and costly procedure in the optimization flow. However, the pursuit of fast simulation should not excessively impact the quality of the simulation. A compromise between accuracy and computational cost specific to this type of optimization must therefore be sought. Following this philosophy, methods were chosen to model the fluid dynamics of the problem that favored speed of convergence over high quality.

3.2.1 Type of simulations

Turbulent flows are almost all of the flows that can be encountered in engineering and are all characterized by high instability and three-dimensional nature, which make these types of flows very difficult to analyze. In the past, the primary approach to studying turbulent flows was experimental. Overall parameters such as the time-averaged drag or heat transfer are easy to measure but as the sophistication of engineering devices increases, the levels of detail and accuracy required also increase, as does cost and the expense and difficulty of making measurements. To optimize a design, it is usually necessary to understand the source of the undesired effects; this requires detailed measurements that are costly and time-consuming. Some types of measurements, are almost impossible to make at present. Others cannot be made with the required precision. As a result, numerical methods have an important role [11]. There are several ways to analyze and study turbulent flows numerically, the three main ways being:

- through equations obtained by averaging the equations of motion over time (if the flow is statistically steady), over a coordinate in which the mean flow does not vary. This approach is called one-point closure and leads to a set of partial differential equations called the Reynolds-averaged Navier-Stokes (RANS) equations. These equations do not form a closed set so this method requires the introduction of other approximations;

- through Large Eddy Simulations (LES), that solves for the largest scale motions of the flow, modeling only the small scale motions;

- through Direct Numerical Simulation, where the Navier-Stokes equations are solved for all of the motions of the turbulent flows. This method is the most accurate approach, but impossible to be used for an engineering study due to its very high computational cost (growing with Re_L^3).

From the point of view of accuracy, obviously, DNS are the best simulations since they solve every scale of motion, however, as mentioned above, nowadays they can be used to study small problems at low Reynolds numbers. LES, on the other hand, solves only a part of the eddies: the larger ones. Smaller eddies, which are those that dissipate fluctuations and are not responsible for the transport of mass, momentum, or energy, are modeled and approximated. Taking advantage of this division, LES maintains very good accuracy while reducing the computational cost compared to DNS, which is still high for this study.

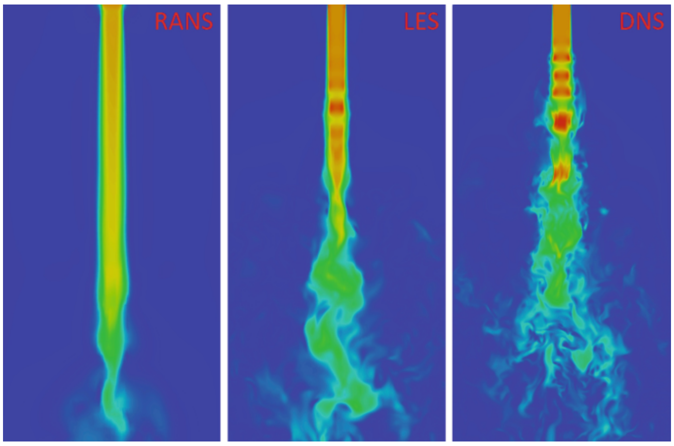


Figure 3.1: Comparison between RANS, LES, and DNS for a jet [12].

RANS remains the best option when the cost cannot be high and a stationary solution is sought. Additionally, by maintaining good computational grid quality, accuracy can also be high.

3.2.2 Reynolds-Averaged Navier-Stokes Equations

To solve the governing equations to represent a fluid statistically, the traditional approach in CFD is to use the simplified Reynolds Avereged-Navier-Stokes (RANS) equations. The RANS equations represent the averaged form of the standard Navier Stokes equations, where the quantities are no longer instantaneous but averaged over a certain period, sufficiently small concerning the phenomena to be followed and sufficiently large concerning turbulence disturbances.

In a statistically steady flow, every variable can be written as the sum of a time-averaged value and a fluctuation about that value:

$$\phi(x_i, t) = \bar{\phi}(x_i) + \phi'(x_i, t) \tag{3.4}$$

where the averaged term is

$$\bar{\phi}(x_i, t) = \lim_{T \to \infty} \frac{1}{T} \int_0^T \phi(x_i, t) dt \tag{3.5}$$

and the term $\phi'(x_i, t)$ is usually orders of magnitude smaller than the average.

From equation 3.5 it follows that $\bar{\phi}' = 0$, so averaging all the linear terms in the equations simply gives the identical term for the averaged quantity. From a quadratic nonlinear term, we get two terms, the product of the average and a covariance:

$$\overline{u_i \phi} = \overline{(\bar{u}_i + u_i')(\bar{\phi} + \phi')} = \bar{u}_i \bar{\phi} + \overline{u_i' \phi'} \tag{3.6}$$

The last term is zero only if the two quantities are uncorrelated, a rare case in turbulent flows. This term leads, in the averaged conservation equations, to the *Reynolds stress tensor* which is composed of terms like $\overline{\rho u_i' u_j'}$, that cannot be represented uniquely in terms of the mean quantities.

Replacing in the N-S equation each variable with its definition as the sum of mean and fluctuating parts, such as:

$$\rho(x_i, t) = \bar{\rho}(x_i) + \rho'(x_i, t)$$

$$p(x_i, t) = \bar{p}(x_i) + p'(x_i, t)$$

it's possible to obtain the averaged continuity and momentum equation. For incompressible flows, without body force

$$\frac{\partial(\rho \bar{u}_i)}{\partial x_i} = 0 \tag{3.7}$$

$$\frac{\partial(\rho \bar{u}_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho \bar{u}_i \bar{u}_j + \rho \overline{u_i u_j}) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \bar{\tau_{i,j}}}{\partial x_j}. \tag{3.8}$$

The presence of the Reynolds stresses in the conservation equations means that the problem is not closed, that is, it contains more variables than equations. Closure requires the use of some approximations, which usually take the form of prescribing the Reynolds stress tensor in terms of the mean quantities [11].

3.2.3 RANS-based Turbulence Models

There are several models for obtaining closure of the RANS equations that vary according to the needs of investigation and available computing power. These models do not provide detailed information on flow structures but only an estimate of the average values of the variables of interest, which can be used for design and optimization purposes. The ability of these equations to produce accurate information about average parameters depends directly on the quality of the models used to approximate the unknown terms. Generally, the preferred approach in modeling the Reynolds tensor consists of introducing the Boussinesq approximation. The basic assumption of Boussinesq is that it is possible to express the Reynolds stresses as a function of the average strain rate tensor through a parameter called turbulent viscosity μ_T

$$-\rho\overline{u_i' u_j'} = \mu_t\left(\frac{\partial\overline{u_i}}{\partial x_j} + \frac{\partial\overline{u_j}}{\partial x_i}\right) - \frac{2}{3}\rho\delta_{i,j}k \tag{3.9}$$

in which $\delta_{i,j}$ is the Kronecker delta [11] . The term k is the kinetic energy and can be defined as

$$k = \frac{1}{2}\overline{u_i' u_j'} \tag{3.10}$$

To solve the RANS equations, it is necessary to define a precise numerical model that expresses the link between the fluid and its turbulent viscosity. Multiple turbulence models express this bond, such as one-equation or two-equation models. The two most employed models in numerical analyses are the two-equation models named $k - \epsilon$ and $k - \omega$.

$k - \epsilon$ model

The $k - \epsilon$ model is a two-equation model that is particularly valid when the flow is totally characterized by a turbulent regime. However, in the case where the chosen computational domain is characterized by the presence of both regimes of motion, this model requires accurate modifications in the mesh size to ensure the same accuracy of results near the wall as away from it. This model is known for its strength and relatively low computational cost and has been shown to be reasonably accurate for a wide range of turbulent flows. The model is composed of the turbulent kinetic energy transport equations k and the turbulent energy dissipation rate ϵ transport equation. Solving the two equations allows us to calculate the turbulent viscosity and close the problem.

The two transport equations are:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon \tag{3.11}$$

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial(\rho u_i \varepsilon)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_{\varepsilon 1}\frac{\varepsilon}{k}P_k - C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \tag{3.12}$$

where P_k is the production of turbulent kinetic energy, σ_k and σ_ϵ the turbulent Prandtl number for k and ϵ and C_ϵ are two model constants [13].

$k - \omega$ **model**

The $k - \omega$ model is also a two-equation model; similarly to the previous one, it is characterized by the turbulent kinetic energy k transport equation, while the second equation is the specific rate of turbulence dissipation ω transport equation. Specific dissipation ω is related to dissipation ϵ by

$$\omega \propto \frac{\epsilon}{k} \tag{3.13}$$

Unlike the previous model, this is suitable for analyzing near-wall flows at $y+$ values below the logarithmic region of the boundary layer. This characteristic makes it unsuitable for analysis of an entire computational domain. The two equations that allow the closure of the problem are the k transport equation and the ω transport equation, respectively:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \beta^* \rho k \omega \tag{3.14}$$

$$\frac{\partial(\rho\omega)}{\partial t} + \frac{\partial(\rho u_i \omega)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \beta\rho\frac{\omega^2}{k} - \beta^* \rho\alpha\frac{\omega k}{D} \tag{3.15}$$

in which β and β^* are two model's constant and D the wall distance.

One reported advantage of the $k - \omega$ model over the $k - \epsilon$ model is its improved performance for boundary layers under adverse pressure gradients. Perhaps the most significant advantage, however, is that it may be applied throughout the boundary layer, including the viscous-dominated region, without further modification [13].

Analyzing the characteristics of the various turbulence models, it was preferred to use the $k - \omega$ model since it is suitable for investigating transition phenomena such as those occurring alongside the chord of a hydrofoil. Using the $k - \epsilon$ model would have neglected the transition effects, assuming a totally turbulent regimen and consequently overestimating the drag.

3.3 Simulation Template

To evaluate each bulb as part of the optimization tool, a fluid dynamic simulation is required. Through the STAR-CMM+ software, a model was set up to serve as a template for simulations of all the bulbs that are to be analyzed. Since this model is to be used for different bulbs but equal wings, it is necessary that each simulation variable is parameterized and each action can be performed by an automatically generated command line input. For reasons of computational cost, it was chosen to perform the simulations in symmetry, without considering possible hydrofoil drift angles, halving the computational domain and speeding up the simulations. The only variable part of the geometry is the bulb, which is the reason why it was chosen to set up a basic simulation without this part of the geometry, leaving in the initial domain only a half-wing and half surface of the vertical drift. Thus, once the bulb has been generated via parameterization, a specially written Java code will select the bulb to be simulated, import it into the ready-made template, and need only to translate the parts already in the domain (joining them with each other at the positions specified by the parameterization), perform the mesh, and start the simulation. The computational domain was parameterized as a function of the chord of the half-wing and the size **target** of the mesh so that all cells are complete and not truncated at the extremes of the domain. Thus, the domain is a parallelepiped of size $(6x5x4)m$ approximately (some variation around these values depends on the mesh size, to avoid cell trimming), whose side lying on the xz plane is the symmetry plane of the simulation.

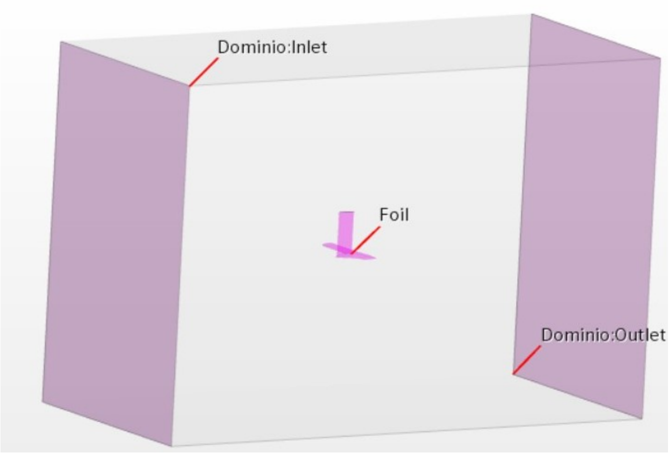


Figure 3.2: View of the fluid domain

3.3.1 Mesh Size

To obtain valuable results it is necessary to have a very good quality and very dense computational grid, but to reduce the simulation time it is necessary not to excessively increase the number of cells. Since it is not possible to establish a correct mesh size at first glance, the computational grid was generated as a function of a parameter *base size* that will be made to vary in the convergence study of the solution.

A trimmed mesh was chosen, with prism layers around the surfaces to be studied. The default characteristics are shown in table 3.1.

Parameter	**Value**	**Unit**
Target size	20.48	% of base size
Minimum size	0.08	% of base size
Prism layer total thickness	0.002	m
Prism layer near wall thickness	3.3×10^{-5}	m
Maximum cell size	20.48	% of base size

Table 3.1: Default Mesh Controls

It is important, to properly use the $k - \omega$ turbulence model, to set the prism layer size to have a dimensionless wall distance $y+ < 1$, in order to keep this region inside the inner layer of the boundary layer.

To allow better mesh discretization around the hydrofoil, a $2cm$ offset was generated around it with cells of dimension 0.64% of base size.

3.3.2 Physics Models and Boundary Conditions

Constant Density
$k - \omega$ Turbulence
All $y+$ Treatment
Reynolds-Averaged Navier-Stokes
Segregated Flow
Steady
3D
Liquid: Water

Table 3.2: Physic Models.

The physical models to represent the domain in question, as mentioned earlier, are based on the solution of RANS with $k - \omega$ turbulence model. The solution researched is a steady-state solution involving water as fluid, with constant density. The equations are solved in a decoupled method (on the software it is possible to choose between a segregated model, the one used, and a coupled where the equations are solved simultaneously). The flow velocity is $3m/s$ and the appendage each time is rotated at an angle of attack of $3deg$ with respect to the chord of the half-wing.

Flow Velocity	$3m/s$
Angle of Attack	$3deg$

Table 3.3: Initial Condition.

Boundary conditions are imposed on the domain boundaries. The flow inlet, on one face of the domain block, is a boundary of type *velocity inlet*, the side on the xz plane is a *symmetry plane* while the others are *pressure outlet*. The surfaces of the hydrofoil are boundaries of type *wall*.

3.3.3 Mesh Convergence Study

The need for fast simulations that integrate smoothly into the optimization tool is a feature that is opposed to the accuracy of the results obtained. However, it is necessary for the simulations to be accurate with a relatively low margin of error. Since an optimal computational mesh size could not be determined at first, it was decided to perform a convergence study for the solution as the mesh size changes. As described in table 3.1, the main cell sizing parameters are given as percentages of the base size, all of which are interrelated except the prism layer, which must meet the conditions imposed by the chosen turbulence model. In the convergence study, several grids were evaluated by varying the value of the base size and analyzing the results obtained. Table 3.4 shows all base size values and results obtained (lift and drag) from the solutions obtained with the different grids. The study was conducted using a bulb generated by CST parameterization.

Base Size	ϕ_D [N]	ϕ_L [N]
0.25	31.9150	441.5289
0.5	32.1566	441.5828
$\sqrt{2}/2$	32.7324	442.4340
1	33.1927	442.4340
$\sqrt{2}$	34.1098	444.5484

Table 3.4: Results of Mesh Convergence Study

The results obtained were analyzed with methods for estimating numerical error in fluid dynamic calculations proposed in *A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies* by Eça and Hoekstra [14].

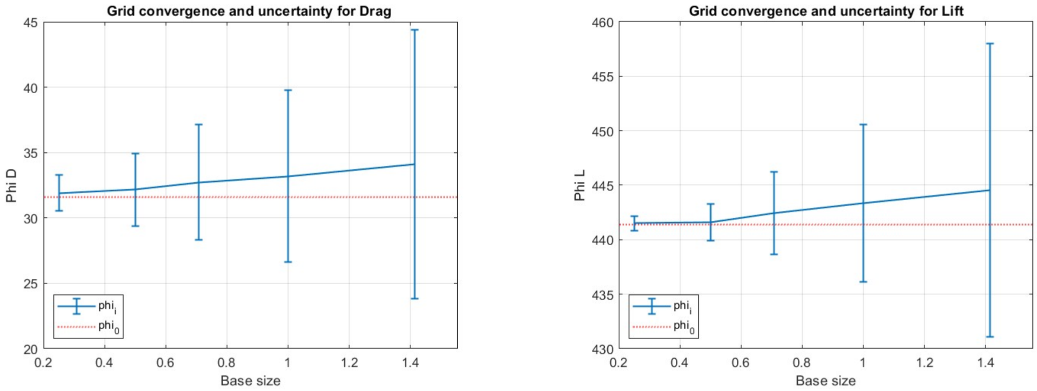


Figure 3.3: Φ_D and Φ_L as a function of base size for the Methods proposed by Eça and Hoekstra[14].

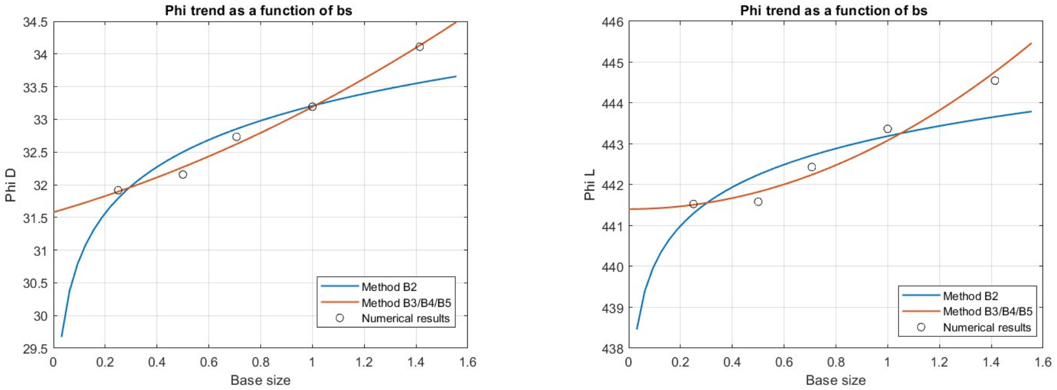


Figure 3.4: Φ_D and Φ_L grid convergence and uncertainty for the Methods proposed by Eça and Hoekstra[14].

Figure 3.3 shows the error estimation values and estimated uncertainties for the proposed methods. In consideration of these data, a value of the base size $bs = 0.8$ is chosen, which allows for small errors and, on the calculator that will be used, allows simulation with computation times less than $120s$ to reach convergence.

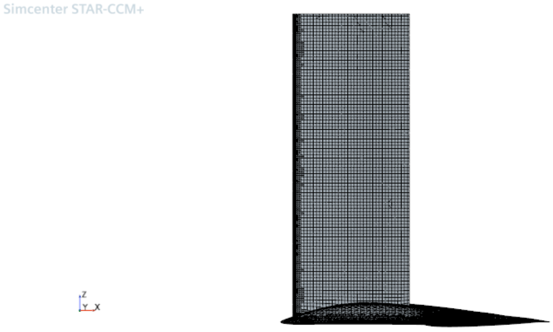


Figure 3.5: Mesh with $bs = 0.8$ on the hydrofoil surfaces.

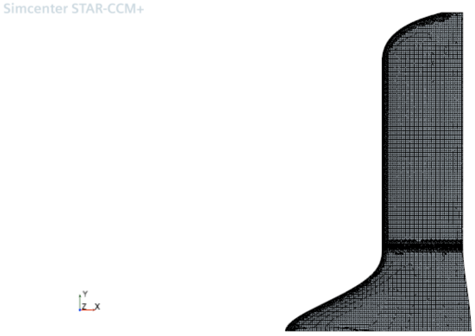


Figure 3.6: Mesh with $bs = 0.8$ on the hydrofoil surfaces.

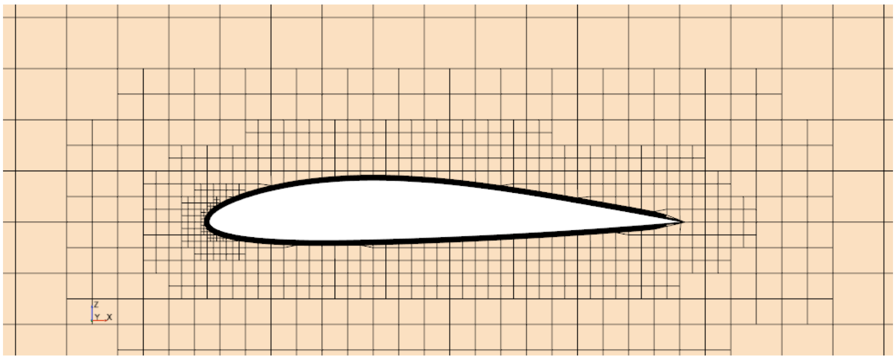


Figure 3.7: Mesh with $bs = 0.8$ section on plane xz: detail of bulb's airfoil.

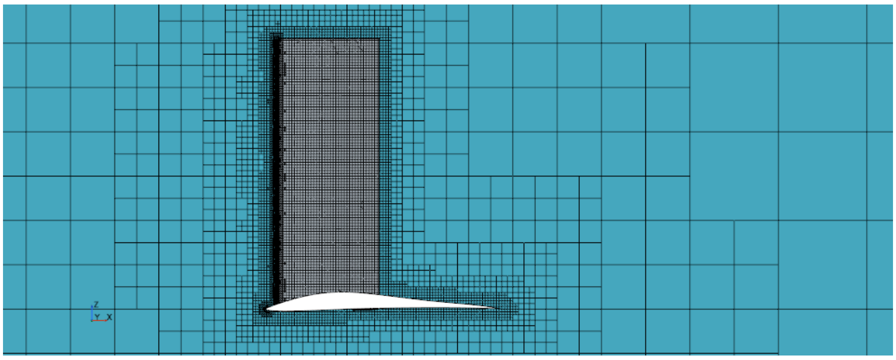


Figure 3.8: Mesh with $bs = 0.8$ section on plane zx.

Chapter 4

Optimization Framework

The choice of the type of algorithm to be used is a key step in the work. Many factors can influence the choice of one family of algorithms over any other. The first fundamental aspect is the objective function: its mathematical description and its behavior (the presence of local minima and maxima, continuity, etc...) are aspects to be taken into account because they influence the reach of the optimum point and the speed of convergence. Another fundamental aspect of this choice is definitely the variables that define the design space: it is important to define how many the variables are and their type, since as the size of the space increases, the possible combinations of parameters among them increase, extending the exploration of possible configurations and their evaluation.

An optimization problem can be presented in the following mathematical form: for a function $f(x)$, defined and continuous in $A \in \mathbb{R}^N$, and a set $\Omega \in A$, $\Omega \neq 0$, identify $\overline{x} \in \Omega$:

$$f(\overline{x}) = \min_{x \in \Omega} f(x). \tag{4.1}$$

In which f is the Objective Function, Ω is the constraints set [15]. An optimization problem can be either minimum or maximum since a maximum can always be traced back to the form

$$\max_{x \in \Omega} f = - \min_{x \in \Omega} (-f).$$

like shown in figure 4.1.

4.1 Objective Function

In mathematical optimization, an Objective Function is a function that maps a configuration of one or more variables into a real number that represents a possible objective of the optimization. The purpose of optimization can be summarized as finding the maximum or minimum of an objective function, so it must be a

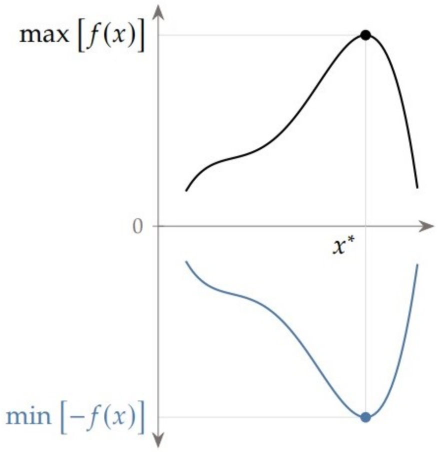


Figure 4.1: Transformation of a maximum problem into a minimum one.

scalar that is computable for a given design variable vector x. Since the objective function is only dependent on the parameters of the optimization and representative of one or more distinctive features of the study, it can be used as a method of evaluating the goodness-of-fit of an individual's parameter configuration. Thus proceeding, it is possible to analyze the design space by assigning an evaluation to each parameter configuration, it determines if one design is better than another [16]. Typical examples of objective functions are cost functions. Cost functions are a subcategory of objective functions of which the minimum is always looked for. A typical problem that requires the use of a cost function may be that of the production of an object as certain parameters change its cost of production. The objective function associates with each parameter (which may be, for example, a size or the use of a certain material) the cost that the final configuration will have. Since the objective of such optimization is to seek the minimum cost, the result of the optimization will be the parameter configuration that is associated with the minimum of the cost function.

In current optimization, there would be many aspects that should be taken into account during optimization: the bulb of an appendage should minimize appendage drag while not affecting lift distribution. Similarly, one of the purposes of the tapered bulb is to delay the stall of the appendage as much as possible, but finding the stall point of each configuration would require an increase in the minimum simulations to evaluate one individual, causing the computational cost to explode. Therefore, because the aim is to go out and investigate both the impact the bulb has on appendage drag and lift, hydrodynamic efficiency is identified as the preferred objective function.

$$E = \frac{L}{D} = \frac{C_L}{C_D} \tag{4.2}$$

Efficiency E is the ratio of lift to drag of the entire appendage (not just the bulb), and is an important evaluation parameter in the field of fluid dynamics because it shows how much the useful force produced by the body prevails over the resistance created. It is the ratio of two forces (so it is dimensionless), or two force coefficients, generated by the interaction between the fluid and the immersed body. By choosing this objective function, increases in lift, decreases in drag and their interaction can be investigated simultaneously while remaining in a single-objective optimization.

Among the approximate formulas of flight mechanics, which can be extended to the field of interest, it is possible to derive a maximum efficiency value for a fixed-wing aircraft. This formula, with appropriate approximations, depends on the aspect ratio of the wing, the zero-lift drag coefficient, and the Oswald factor.

$$E_{max} = \frac{1}{2}\sqrt{\frac{\pi e AR}{C_{D,0}}} \tag{4.3}$$

However, this formula cannot be useful for the case under study because the geometry studied is much more complex and is described by many more parameters than those involved in the equation 4.3, so this optimization cannot be approximated by a simple equation. Thus, the objective function will turn out to be a function that cannot be described analytically, but calculated by solving Navier-Stokes equations in approximate form using the CFD model previously mentioned. Using efficiency E as the objective function, the optimization problem would result in the search for a maximum of the function. As mentioned earlier and reported in [15], a maximum problem can always be traced back to a minimum one; working with algorithms already implemented in MATLAB it turns out how they were developed for finding the minimum of the objective function. Under these assumptions, it is necessary to modify the objective function in question to relate back to a case in which it is minimized. The choice was made to use as the objective function:

$$f = \frac{1}{E} = \frac{D}{L} = \frac{C_D}{C_L}. \tag{4.4}$$

4.2 Optimization Algorithms

In the more general approach, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. The design optimization process requires translating the intent of the optimization into a mathematical statement that can then be solved by an optimization algorithm. Developing this statement has the added benefit that it helps the designer better understand the problem. Being methodical in the formulation of the optimization problem is vital because the optimizer tends to exploit any weaknesses in the formulation or model. An inadequate problem formulation can either cause the optimization to fail or cause it to converge to a mathematical optimum that is undesirable or unrealistic from an engineering point of view [16]. To choose the most appropriate optimization algorithm for solving a given optimization problem, it's necessary to classify the optimization problem and know how its attributes affect the efficacy and suitability of the available optimization algorithms. This is important because no optimization algorithm is efficient or even appropriate for all types of problems. It's possible to classify optimization problems based on two main aspects: the problem formulation and the characteristics of the objective and constraint functions.

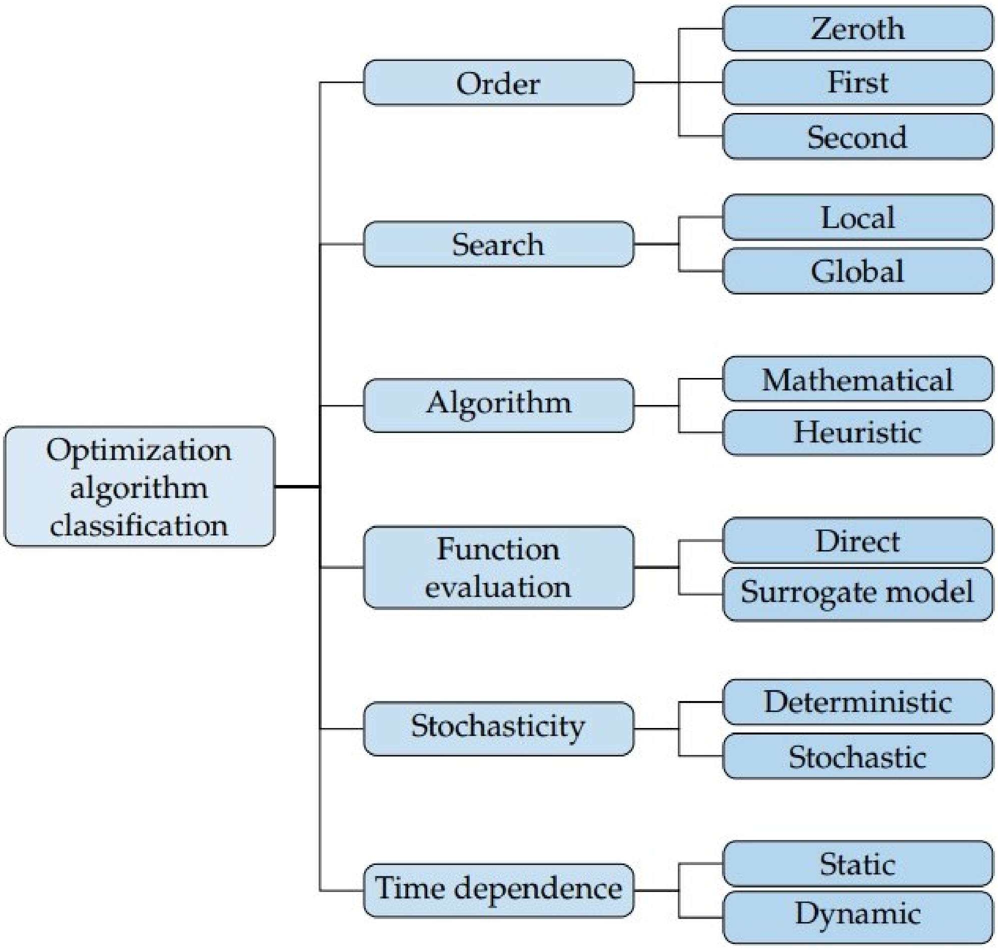


Figure 4.2: Classification of Optimization Algorithm [16].

4.2.1 Algorithm choice

Algorithms for searching for the optimal solution can generally be divided into three macro-categories: Enumerative Techniques, Numerical Techniques, and, Probabilistic Techniques. Enumerative techniques search for the optimal solution at all points in the domain, possibly discretized, of the function. Numerical techniques use a set of necessary and sufficient conditions that must be satisfied by the solution of the optimization problem. These can be classified into direct and indirect. Indirect methods seek the minimum of a function by solving a set of nonlinear functions and iteratively searching for the solution until the gradient of the cost function is zero. Direct methods make sure that it is the gradient that guides the search for the solution. Probabilistic techniques are based on enumerative techniques, but they use information additional information to perform the search and can be viewed as evolutionary processes [17].

For the study of problems of this complexity, with objective functions whose general trend is unknown, it is effective to use probabilistic algorithms, since a more traditional approach would be time prohibitive. By analyzing studies similar to the one proposed in this thesis work, such as "Design and optimization of rudder bulb based on CFD and genetic algorithm" by Xialong W. et al [18], it can be seen that the most widely used algorithms are genetic algorithms, a sub-branch of metaheuristic algorithms. In general, the objective of a heuristic is to produce a solution in a reasonable time frame. This solution may not be the best of all the solutions to this problem, or it may simply approximate the exact solution. But it is still valuable because finding it does not require a prohibitively long time. Examples of metaheuristic algorithms, which are usually inspired by nature, are:

- *Simulated Annealing.* This algorithm is inspired by the annealing process of metals. The atoms in a metal form a crystal lattice structure. If the metal is heated, the atoms move around freely. As the metal cools down, the atoms slow down, and if the cooling is slow enough, they reconfigure into a minimum-energy state [16].
- *Neural Network.* Artificial neural networks are a branch of machine model algorithms, inspired by the mechanism of the human brain.
- *Evolutionary Algorithms.* They are based on the principle of biological evolution, which consists of modifying a population of candidate solutions through mechanisms such as reproduction, mutation, recombination, and selection.

4.3 Genetic Algorithm

In recent years, metaheuristic algorithms have been used to solve real-life complex problems arising from different fields such as economics, engineering, management,

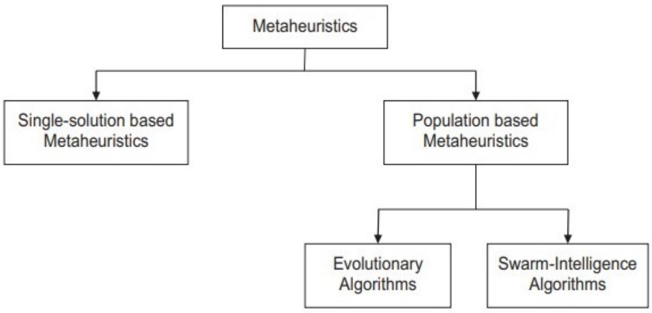

Metaheuristics
Single-solution based Metaheuristics
Population based Metaheuristics
Evolutionary Algorithms
Swarm-Intelligence Algorithms

Figure 4.3: A classification of metaheuristic Algorithms.

and engineering. Population-based metaheuristics utilize multiple candidate solutions during the search process. These metaheuristics maintain the diversity in population and avoid the solutions being stuck in local optima. Some of well-known population-based metaheuristic algorithms are genetic algorithm (GA) [19]. A Genetic Algorithm mimics the evolution theory of survival of the strongest in nature. Following this statement, basic elements of GA are chromosome representation, fitness selection, and other biological-inspired operators. Chromosomes are considered as points in the design space. These are processed using genetic operators by iteratively replacing its population. The fitness function is used to assign a value for all the chromosomes in the population. The biological-inspired operators actively intervene in the algorithm, allowing it to effectively analyze the design space. The three most important are:

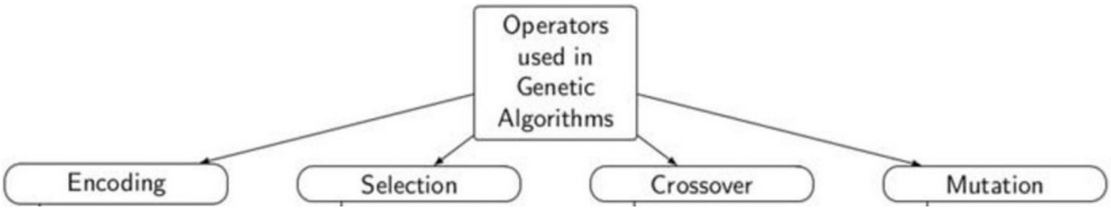


Figure 4.4: Operators used in GA.

- *Selection*: an important step in genetic algorithms that determines whether the particular chromosome value will participate in the reproduction process or not. The selection step is sometimes also known as the reproduction operator because if a gene is positively marked, it will be present in the future generation. The selection techniques are probabilistic operators such as roulette wheel, Boltzmann tournament, or rank.
- *Crossover* : Crossover operators are used to generate offspring by combining the genetic information of two or more parents.

- *Mutation*: Mutation is an operator that maintains the genetic diversity from one population to the next population [20].

As shown in figure 4.5, the rough functioning of a GA is very intuitive. First, the zero generation is created, i.e., a first set of individuals to which a value of the design parameters is assigned. Each individual in this first set is evaluated by the fitness function: the best elements, called "elites," are stored aside since they are the ones that come closest, for this iteration, to the optimal solution. Next, the selection operator chooses which genes will be transferred from the current generation to the next. At this point the crossover and mutation operators intervene, assigning free parameter values to the offspring by overlapping the parent values and randomly introducing a mutation to maintain a value of randomness in the reproduction of the generations. The new generation is thus generated, which in turn is fed back into the cycle to reevaluate its performance and generate the next generation until a generic stop condition is reached. The stop can be the asymptotic attainment of the optimum, a maximum number of generations, or an interval of generations for which the best fitness remains unchanged.

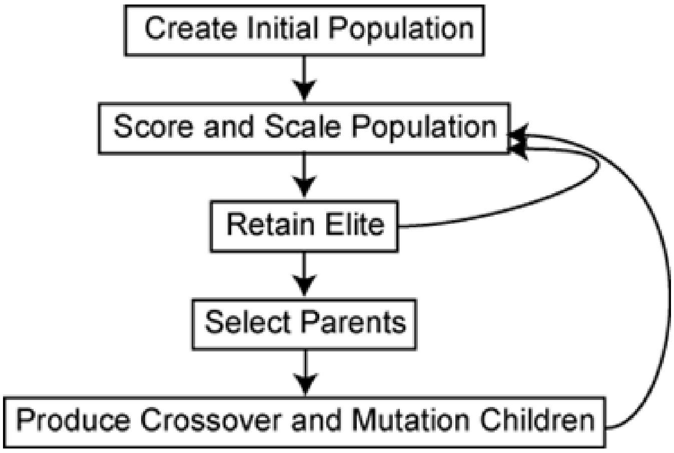

Create Initial Population
Score and Scale Population
Retain Elite
Select Parents
Produce Crossover and Mutation Children

Figure 4.5: Ga Flow Chart: main algorithm steps [21].

4.4 Optimization Flow and Algorithm Structure

Once the algorithm to be used and the fitness function has been chosen, a structured and robust code must be built to evaluate each individual and reach the solution as quickly as possible. The tool should be constructed in such a way as to lighten the procedure as much as possible: the fluid dynamic simulation step is the most computationally and time-consuming one, so simulation should be avoided for all those individuals for which inadequate fitness can be predicted without simulation.

It is also necessary for the algorithm to foresee exceptions for possible errors during the various steps and to know how to handle these appropriately without getting stuck and without discarding possible performing individuals.

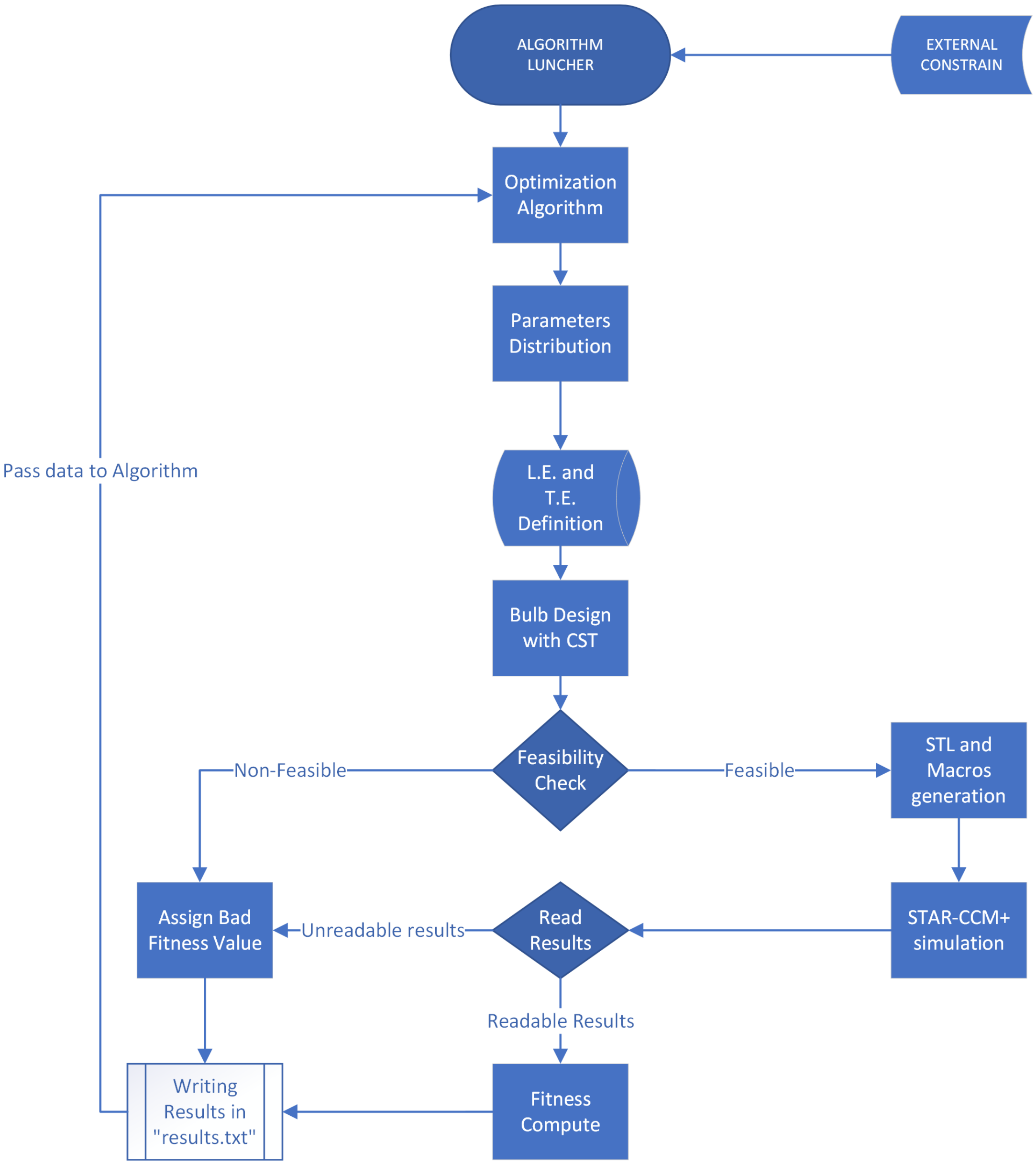


Figure 4.6: Algorithm's workflow

4.4.1 Optimization constraints and options

The first step of the algorithm (in figure 4.6) is the tool launcher in which initially the fixed parameters of the geometry are declared, those shown in table 2.1. In addition, the launcher takes as input the constrain of the problem and the optimization options of MATLAB's ga function. The constraints of this optimization are given through ranges of allowable values for each free parameter. Thus, it was stated, through preliminary optimizations and literature studies, the maximum and minimum values assignable to each parameter by the selection, mutation, and crossover operators at assignment. These constraints are given in table 4.1.

Parameter	**Min**	**Max**	**Unit**
b	0.08	0.4	m
c_r	c_t	0.6	m
x_{LEt}	0	0.5	m
x_{vert}	0	0.31	m
le_{flag}	1	7	–
$Bu_{1,1}$	0	0.6	–
$Bu_{2,1}$	0	0.8	–
$Bu_{3,1}$	-0.3	0.5	–
$Bu_{4,1}$	-0.1	0.8	–
$Bu_{1,2}$	0	0.4	–
$Bu_{2,2}$	0	0.8	–
$Bu_{3,3}$	-0.3	0.5	–
$Bu_{4,2}$	-0.1	0.8	–
$Bl_{1,1}$	-0.5	0	–
$Bl_{2,1}$	-0.6	0.1	–
$Bl_{3,1}$	-0.6	0.1	–
$Bl_{4,1}$	-0.5	0.1	–
$Bl_{1,2}$	-0.5	0	–
$Bl_{2,2}$	-0.6	0.1	–
$Bl_{3,3}$	-0.6	0.1	–
$Bl_{4,2}$	-0.5	0.1	–

Table 4.1: Free parameters constraints.

The ga function is very powerful and allows several options to make the optimization customizable. The options are among the most varied, those used for the study are shown in table 4.2. It is important to note an option that intervenes in parameter n.5, LE_{flag}, which is a discrete value that allows the choice of the leading edge, it should be set as an integer parameter for MATLAB's ga function, via option *par.inticon*.

Option	Value
Pop. Size	200
Max Stall Gen.	30
UseParallel	True
inticon	5
Max Gen	100

Table 4.2: Ga function options.

The options shown in the table are those that were modified for this study. The Pop Size indicates the population size of each generation and was chosen to be 200 elements in each generation, the Max Stall Gen is the maximum number of generations for which optimization goes on without minimum fitness improvement. Using parallel set to "True" allows the calculation to be executed in parallel in multiple processes, so that the computational cost is divided over several processors, reducing the time of optimization. Other options, such as the choice of crossover, selection, and mutation methods of the genetic algorithm were not specified, so the basic ones of the MATLAB function were kept unchanged [22].

Immediately after the launcher, the genetic algorithm intervenes, which, as explained earlier, by selection assigns values to the 21 free parameters, taking into account the constraints specified in 4.1, for each individual and for each generation, cyclically individual by individual. These parameters will be used to construct the geometry.

4.4.2 Design and Feasibility Check

Having chosen the values by selection-crossover-mutation, the surfaces that constitute the bulb can be generated. The geometry is created as specified in the chapter 2, via CST parameterization. To begin the process, the first two columns of free parameters are assigned to the Bu and Bl matrices, while the last column is fixed. Out of the remaining free parameters, four are lengths that identify the dimensions of the bulb or vertical position, while the LE_{flag} parameter is the one that specifies the function that interpolates the points of the Leading and Trailing edges. Through a specially structured function, the parameters c_t, x_{let}, c_r, b and LE_{flag} construct the leading and trailing edges of the bulb. At this point, the geometry simply needs to be composed via a script *crea bulbo* with the CST parameterization. The bulb has been created and is still stored only as point matrices of the top and bottom two surfaces. Before creating the STL file of it, it is useful to check the feasibility of the geometry. There are some configurations for which it is possible to determine in advance that the geometry is not valid, reason why it is useful to discard them before the simulation and save precious computing time for individuals who might perform well. In a separate function, *check feasibility*, four geometric checks are made for the validity of the configuration and the calculation of the z-height at which the vertical surface is plugged into the bulb so that the engagement is completely inside the bulb and does not come out. The four controls are :

1. Interpenetration. The two surfaces, upper and lower, should not interpenetrate each other.

2. Bulb thickness. The maximum thickness of the bulb should generally be decreasing along y. For appropriate sections along y, it is checked that the maximum thickness does not increase.

3. Chord at the root. The trailing edge at the root cannot be forward the tip's trailing edge, due to constructional issues. This control affects the chord at the root; it cannot be smaller than $x_{let} + c_t$ (fixed parameter).

4. Graft of the centerboard. The point of coupling of the vertical surface in x (parameter x_{vert}) cannot be at a point such that it causes the trailing edge of the vertical drift to leave the bulb.

If an individual does not pass at least one of these checks it will be automatically discarded by the algorithm by assigning it a very high fitness value that identifies a code for that error, it will be reported in the results file that shows all its characteristics, and the genetic algorithm will produce a new individual to be tested.

An individual that passes these checks is suitable for simulation, the reason why a geometric mesh of it will be created in STL file, to be exported to the star CCM+ simulation software.

Type of check	**Condition checked**
Interpenetration	$z_u(x, y) \geq z_l(x, y)$
Growing Thickness	$\delta_{max}(y_{i+1}) \leq \delta_{max}(y_i)$
Root Chord	$c_r \geq x_{let} + c_t$
Centerboard Graft	$x_{vert} + c_{vert} \leq c_r$

Table 4.3: Feasibility checks.

4.4.3 CFD Simulation

Once the STL file of the bulb geometry is created, the next step is fluid-dynamic simulation. Once the STL file of the bulb geometry has been created, the next step is fluid dynamic simulation. Before starting the simulation software, it is necessary to write the macro file with the operations to be performed by STAR CCM+. Through the function *write macro* a Java code file is written to have the following actions performed in order:

1. Import the STL file of the current geometry taking care to place the bulb nose in the origin of the three axes;
2. Translate the horizontal wing of the template (chapter 3) so that it grafts to the bulb tip, via the parameters x_{let} and $b/2$;
3. Translate the vertical surface, the drift, to the position identified by parameter x_{vert};
4. Unite the surfaces of the bulb, horizontal wing, and vertical drift together and rotate them at the specific angle of attack for this optimization ($3deg$);
5. Compute the mesh of the geometry with the parameters already set previously;
6. Initialize the solution and run the calculation;
7. Save a file containing the results of the simulation.

Each bulb, and consequently each result file obtained on that geometry, are identifiable by their name, which corresponds to the time the geometry was generated, with the format hh:mm:ss, to avoid possible misunderstanding. Having generated the macro file for the simulation, the tool starts STARCCM+ via command line and executes the macro file corresponding to the current bulb. When the simulation starts, a counter is also set to delete a simulation that does not reach convergence in the set time (150 s). Once the results file is successfully produced the timer stops and the tool tries to read the file. If it is possible to do so, C_L and C_D are detected from the results file, and the fitness, which is the value of the objective function associated with the individual in question, is calculated. Each parameter of the subject, including generation time and calculated fitness are reported in an optimization report file as a text line. The simulation process is finished, the genetic algorithm receives as input the fitness of the bulb and proceeds by selection to generate a new bulb of the current generation. This process continues until the genetic algorithm meets a stopping condition, which could be a maximum number of generations, a maximum computation time, or the attainment of the objective function's minimum.

4.4.4 Error and non-feasibility Management

The tool must work stably without crashing if an error occurs and be able to handle any possible stumbling situation in an optimal way. It is also necessary for the tool to let the genetic algorithm know when geometry is invalid and the shades between the various feasibility checks. For example, a non-feasible subject due to a failed check on the position of the vertical (because it comes out of the bulb) is a less important error than a bulb that has a maximum thickness at the root smaller than the thickness at the tip. It is possible to combine the handling of errors and invalidities through a penalty method. For each possible error or feasibility check, it is necessary to identify an order of relevance that ranks them in ascending order and assign a high fitness value to each of them. This method has several advantages: depending on the assigned fitness it is possible to identify the type of error detected, it saves time by avoiding simulating invalid individuals, and it communicates to the algorithm the degree of error and invalidity that individual has. Table 4.4 lists the possible errors or invalidities of the geometry and their assigned fitness, which also acts as an exit flag from the generation and simulation functions.

Type of unfeasibility	**Assigned Fitness**
Centerboard Graft	30
Root Chord	50
Growing Thikness	100
Interpenetration	200
Type of error	**Assigned Fitness**
Unreadable results	3
Negative results	5
No results file	6
Exceeding time limit	10

Table 4.4: Type of possible errors or unfeasibility and their assigned fitness.

4.4.5 Results Collection

The operation of the optimization tool made for this study has been described. To summarize, the tool uses a genetic algorithm in Matlab to search for the optimal configuration of the bulb. The tool first assigns values to the design variables following the rules of the genetic algorithm, defines the geometry characterized by the chosen parameters, and evaluates its feasibility: if the individual is not feasible it will be discarded by assigning a bad value to the objective function, if it is valid it will be simulated through STAR CCM+ and its results will be read from the software report and passed to the genetic algorithm that will take it into account to generate the next individuals and subsequent generations. For each individual generated, the parameter values and their fitness are saved in a text file that will be used to evaluate the actual performance and operation of the tool. For each individual generated, the parameter values and their fitness are saved in a text file that will be used to evaluate the actual performance and operation of the tool.

4.5 Algorithm Results

The tool was run on the *Legion* cluster of the Politecnico di Torino and was stopped at the 100th generation. Below are the results obtained and the outputs generated by the tool.

4.5.1 Death Rate

Death Rate, or Mortality, is defined as the percentage of individuals that were found to be unfeasible compared to the total number of individuals generated. The death rate is extremely high in the first generations and falls in the first 10 generations. It can be seen that there is a slight increase toward the latter

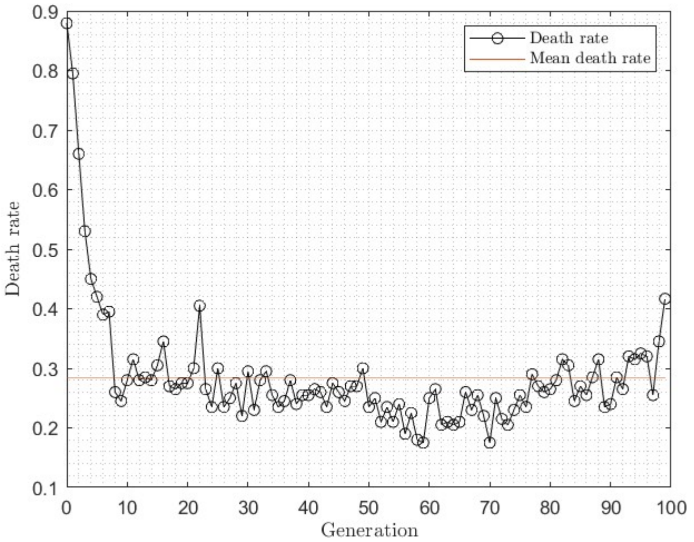


Figure 4.7: Death rate across generations.

generations, a symptom that the algorithm was still exploring areas of the design space that were not optimal 4.7.

4.5.2 Fitness

Below are shown the minimum, which are the best, and average fitness across generation 4.8. In figure 4.9 are shown fitness of all simulated elements is shown, compared with the best ones in each generation.

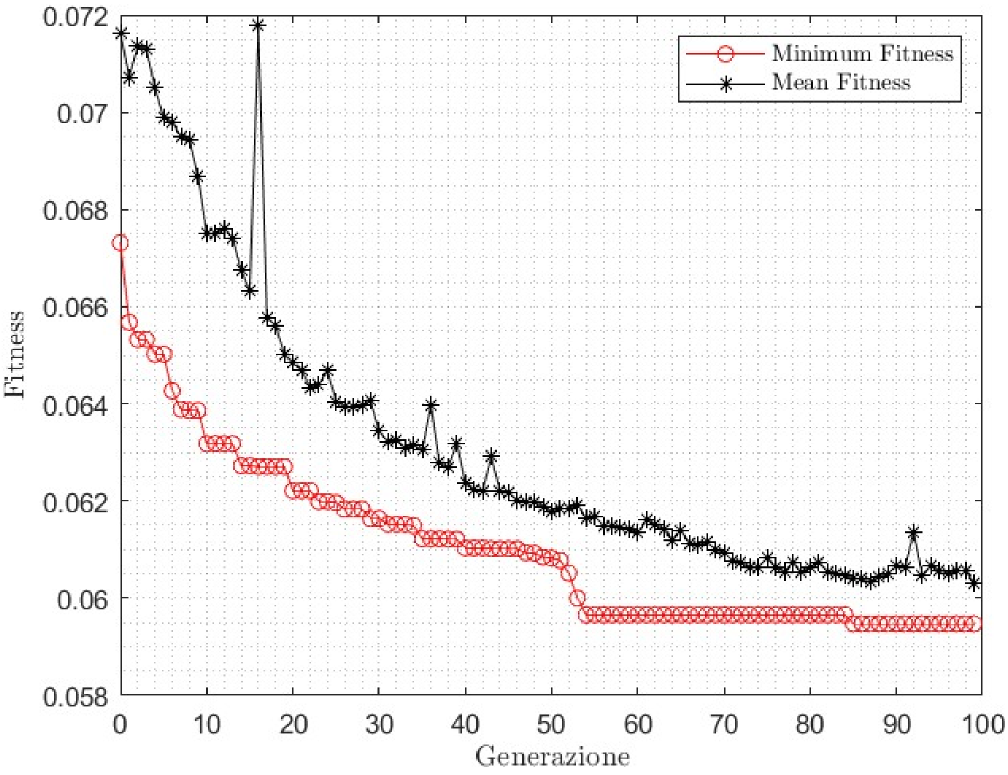


Figure 4.8: Minimum and mean fitness across generation.

4.5.3 Parameters of best-performing individuals across generations

Below are the trends of the design parameters of the best-performing individuals in each generation.

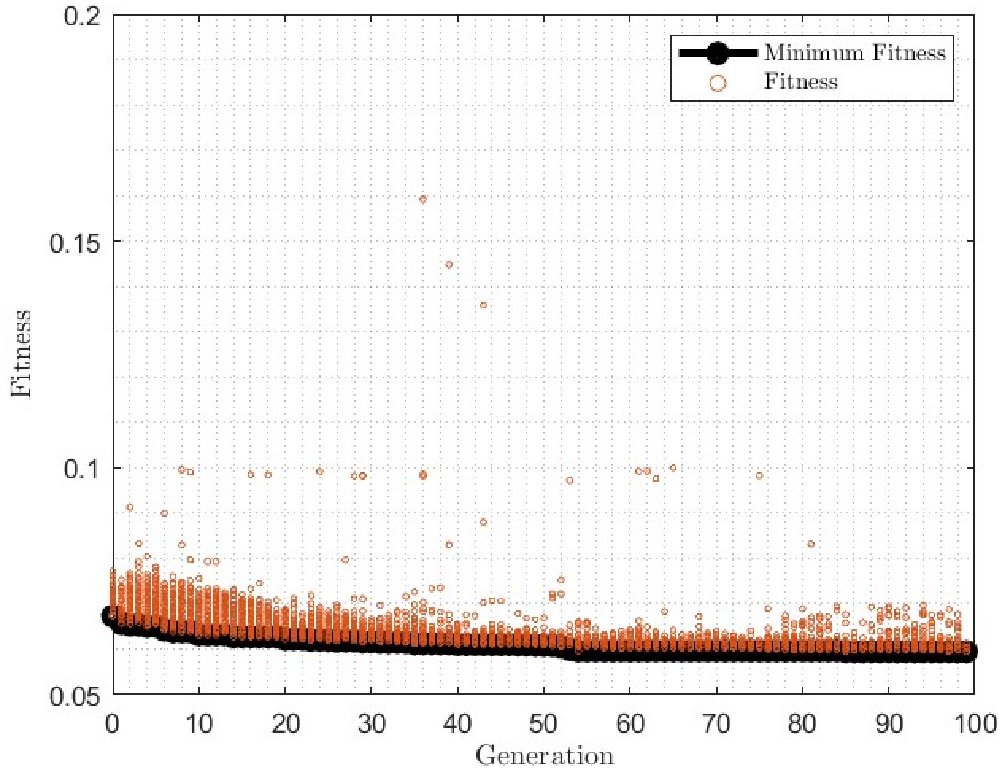


Figure 4.9: Fitness of each simulated bulb compared to best fitness of each generation.

4.5.4 Best Individual

The optimization found as best individual the bulb shown in Figure 4.17, characterized by the parameters in table 4.5.

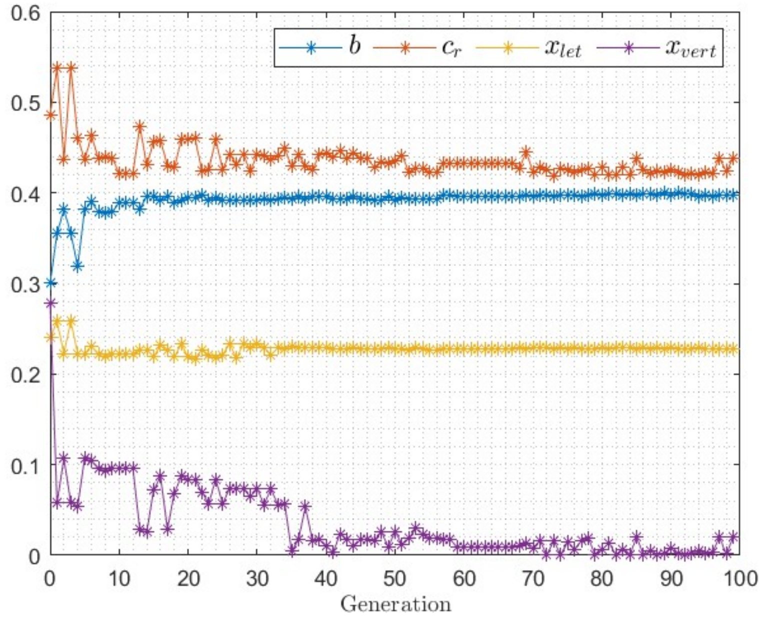


Figure 4.10: Length design parameters of a best scoring individual across generations.

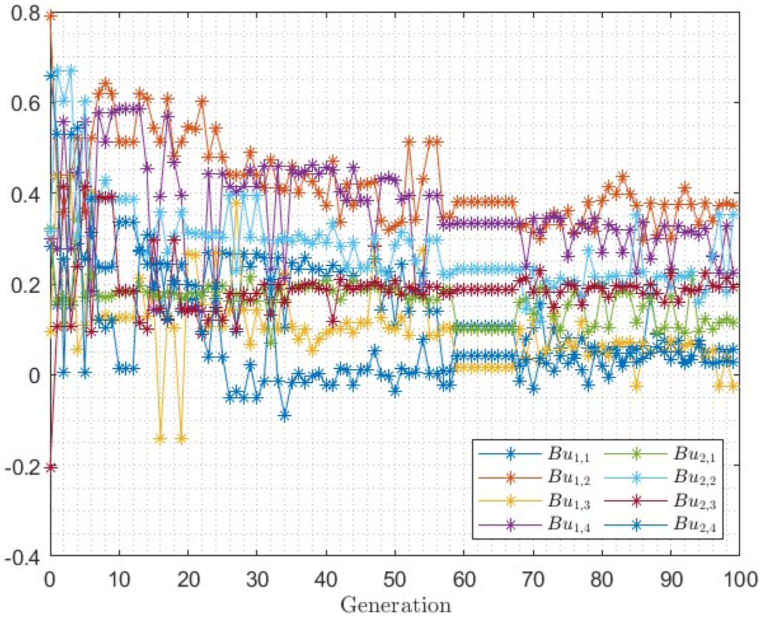


Figure 4.11: Coefficients of Bu matrix of a best scoring individual across generations.

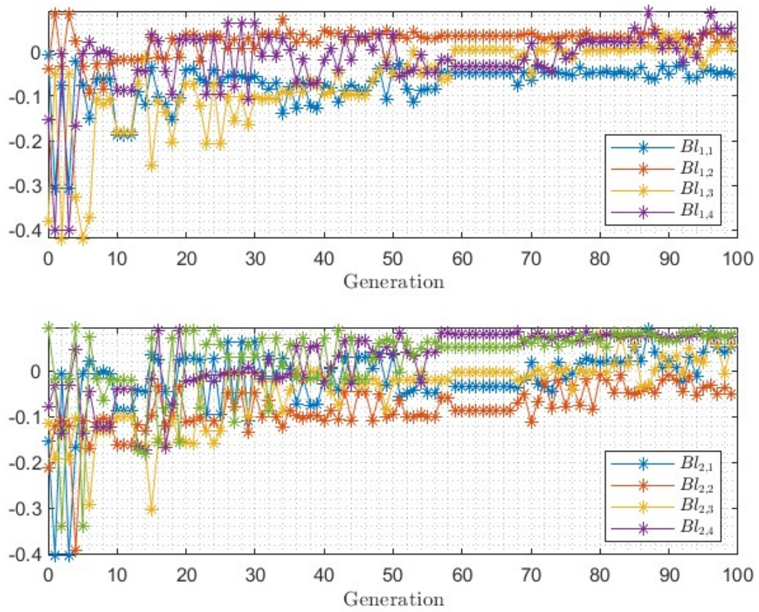


Figure 4.12: Coefficients of Bl matrix of best scoring individual across generations.

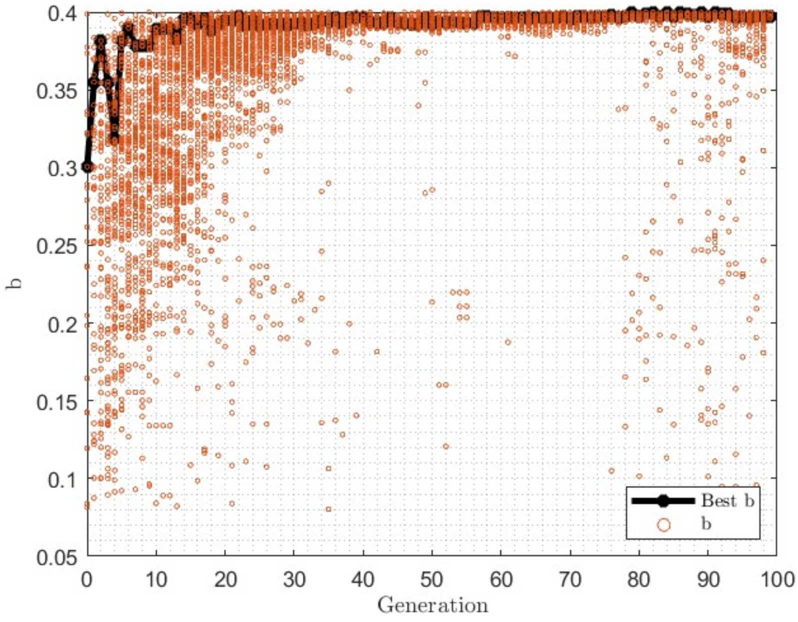


Figure 4.13: b of each simulated bulb compared to b of best individual of each generation.

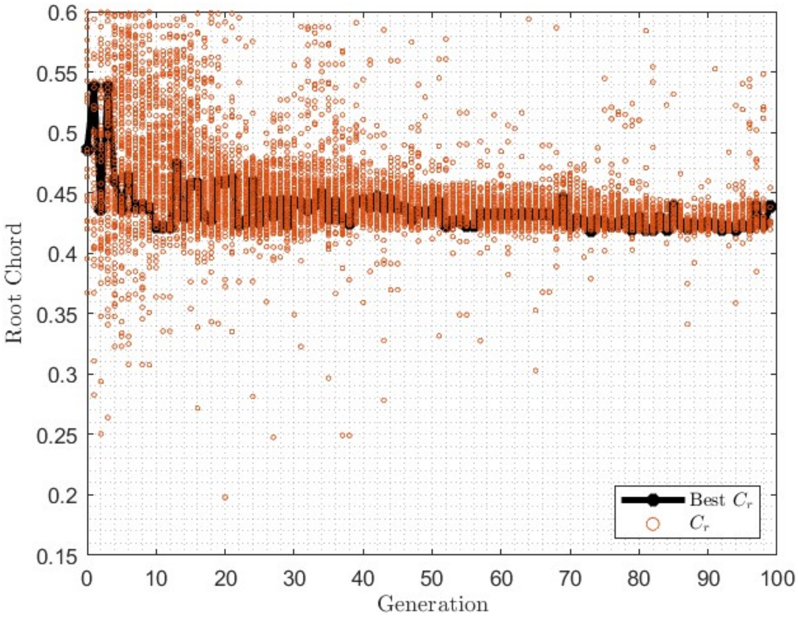


Figure 4.14: Root Chord of each simulated bulb compared to root chord of best individual of each generation.

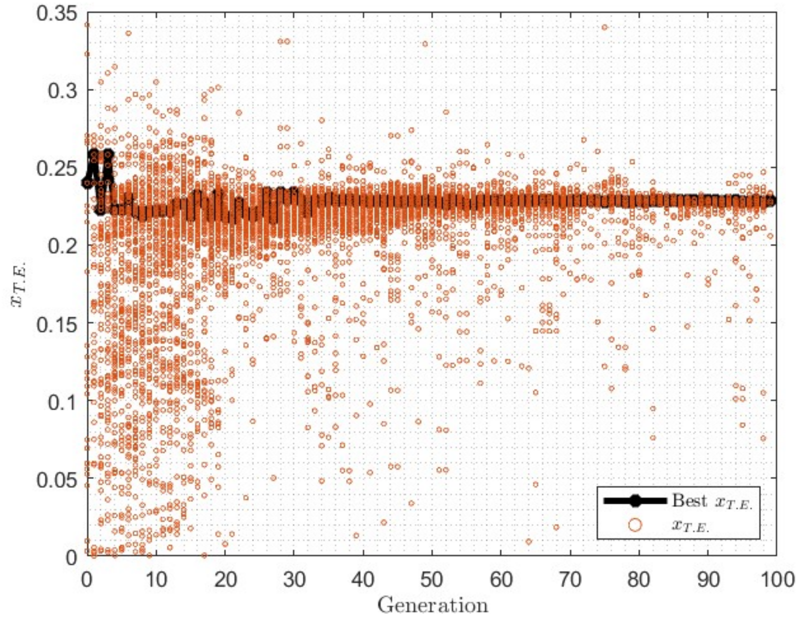


Figure 4.15: x_{TE} of each simulated bulb compared to x_{TE} of best individual of each generation.

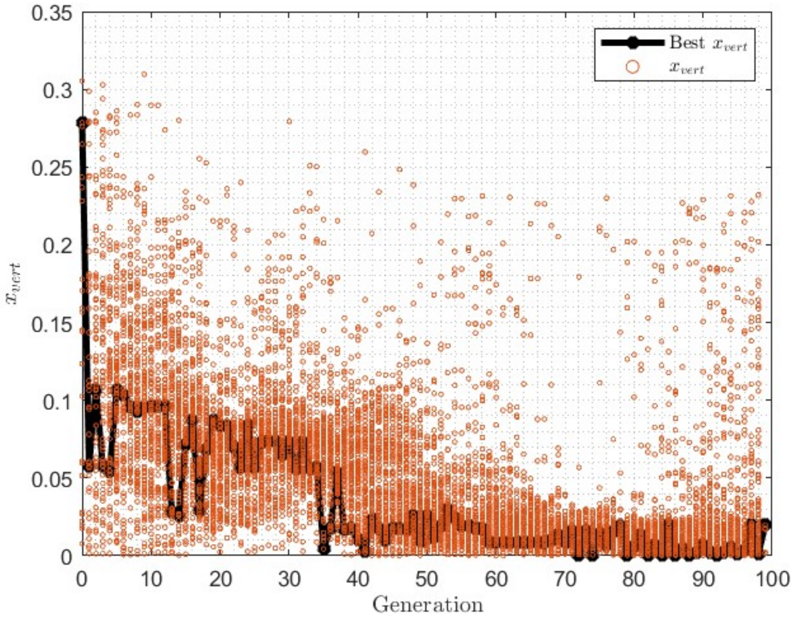


Figure 4.16: x_{vert} of each simulated bulb compared to x_{vert} of best individual of each generation.

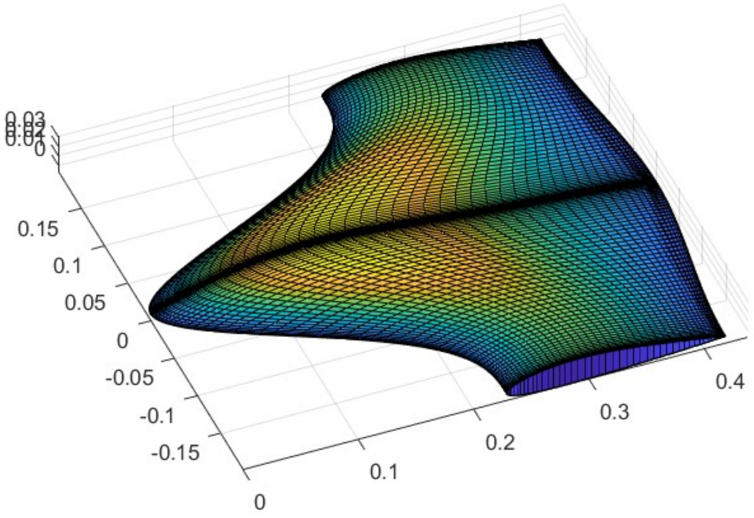


Figure 4.17: Best Individual.

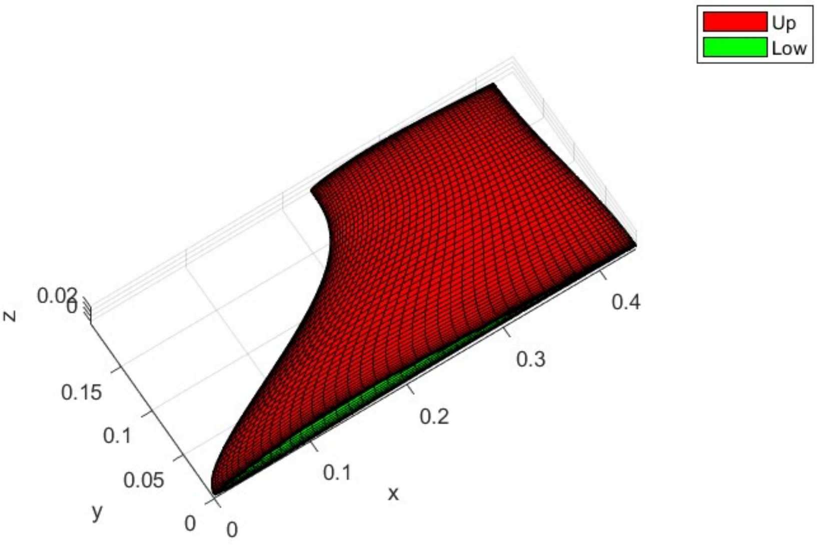


Figure 4.18: Best Individual: half surfaces generated with CST.

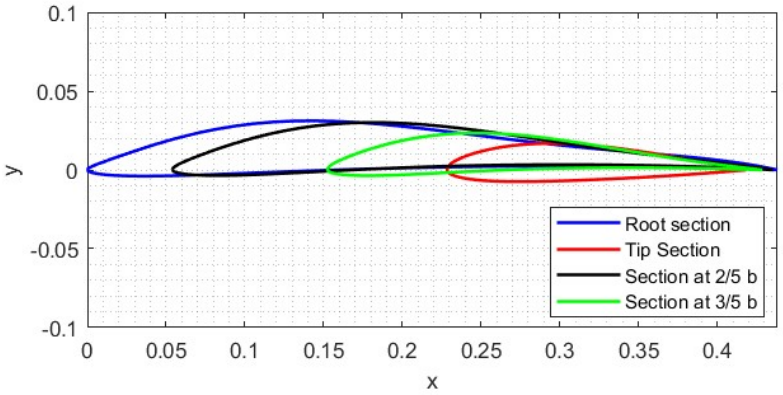


Figure 4.19: Sections of the Optimized Bulb at different spans.

Parameter	**Value**
b	0.3971
c_r	0.4382
x_{LEt}	0.2284
x_{vert}	0.0197
le_{flag}	4
$Bu_{1,1}$	0.056954
$Bu_{2,1}$	0.372630
$Bu_{3,1}$	-0.026904
$Bu_{4,1}$	0.223240
$Bu_{1,2}$	0.1146500
$Bu_{2,2}$	0.3525400
$Bu_{3,3}$	0.1933200
$Bu_{4,2}$	0.02685500
$Bl_{1,1}$	-0.049439
$Bl_{2,1}$	0.019649
$Bl_{3,1}$	0.0070783
$Bl_{4,1}$	0.053188
$Bl_{1,2}$	-0.049327
$Bl_{2,2}$	0.057067
$Bl_{3,3}$	0.075935
$Bl_{4,2}$	0.079870
Fitness	0.05947

Table 4.5: Parameters of the best scoring bulb.

Chapter 5

Adjoint Optimization

Optimization through the genetic algorithm was carried out with low-fidelity CFD simulations, which means that they would guarantee results affected by not too high error but would reach convergence in the shortest possible time. The errors were evaluated through the mesh convergence study. The created model was parameterized through interpolations of functions, describing span-wise and chord-wise sections of limited degree, to keep the number of free parameters of the optimization limited, allowing the geometry a restricted number of possible configurations. The union of all these possible sources of displacement relative to a hypothetical optimal solution opens the space for the integration, after obtaining results from the optimization tool, of a second detailed optimization. The second optimization, with a model affected by minor errors, can be applied only to the result of the tool with a genetic algorithm, so that high-fidelity simulations can be performed and the geometry of the bulb can be modified in all its facets, not only through a limited number of design parameters.

The use of a second optimization method, one that is already developed and tested, may allow to highlighting critical aspects of the design that were not appropriately evaluated in the tool or that were discarded when choosing parameters. Given these assumptions and considerations, the Adjoint Solver already implemented in the commercial software STAR CCM+ was chosen as the optimizer. This solver has been developed and tested for years and is an efficient means to predict the influence of many input parameters on some engineering quantities of interest in a simulation. In other words, it provides the sensitivity of objectives (cost functions) with respect to input parameters such as design variables or values of boundary conditions.

5.1 Discrete Adjoint Approach

The goal of aerodynamic design optimization is the minimization of an objective function that is a nonlinear function of a set of discrete flow variables. For example, efficiency E may be expressed as $E(U)$ where U is the set of all flow variables at discrete grid points arising from an approximate solution of the Euler equations, and E is a scalar function that approximates the appropriate weighted integral of pressure over the surface of a wing [23]. In design optimization, the question of interest shifts to what perturbations of the objective function are due to the perturbation of the geometry, and thus how the flow field is changed. If u is the perturbation in the flow field, the linearized perturbation in Efficiency is

$$g^T u = \frac{\partial L}{\partial U} u \tag{5.1}$$

and the goal is to evaluate $g^T u$, where u satisfies the appropriate linearized flow equations.

Given a set of design variables, α, which control the geometry and a set of flow variables at discrete grid points, U, the aim is to minimize a scalar objective function $J(U, \alpha)$. This minimization is subject to the constraint that the discrete flow equations and boundary conditions need to be all satisfied. These statements can be expressed as

$$N(U, \alpha) = N(U, X(\alpha)) = 0 \tag{5.2}$$

where X is the vector of grid point coordinates which depends on α. Using an appropriate technique, the grid can be deformed, as smoothly as possible, following the surface sensitivity to the field function. For a single design variable, the solution can be linearized about a primal solution U_0, to get

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U}\frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha} \tag{5.3}$$

which need to be subject to the constraint that the flow sensitivity $dU/d\alpha$ satisfies the linearized flow equation

$$\frac{\partial N}{\partial U}\frac{dU}{d\alpha} + \frac{\partial N}{\partial \alpha} = 0 \tag{5.4}$$

Defining

$$u = \frac{dU}{d\alpha}, \qquad A = \frac{\partial N}{\partial U}, \qquad g^T = \frac{\partial J}{\partial U}, \qquad f = -\frac{\partial N}{\partial \alpha}, \tag{5.5}$$

the equation 5.3 can be rewritten in its standard form

$$\frac{dJ}{d\alpha} = g^T u + \frac{\partial J}{\partial \alpha} \tag{5.6}$$

The direct sensitivity of the objective function to perturbation in the design variables can be evaluated by direct approach, solving $Au = f$ [23].

An alternative description can be obtained using Lagrangian multipliers associated. The adjoint variables are Lagrangian multipliers, called λ, which are introduced in an augmented objective function

$$I(u, \alpha) = J(U, \alpha) - \lambda^T N(U, \alpha) \tag{5.7}$$

Considering the perturbation dU and $d\alpha$ gives

$$dI = \Big(\frac{\partial J}{\partial U} - \lambda^T \frac{\partial N}{\partial U}\Big) + \Big(\frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial N}{\partial \alpha}\Big) d\alpha \tag{5.8}$$

If λ^T satisfies the adjoint equation

$$\frac{\partial J}{\partial U} - \lambda^T \frac{\partial N}{\partial U} = 0 \tag{5.9}$$

that can lead to

$$dI = \Big(\frac{\partial J}{\partial \alpha} - \lambda^T \frac{\partial N}{\partial \alpha}\Big) d\alpha \tag{5.10}$$

which can be used to compute the sensitivity $dI/d\alpha$ [23].

5.2 Model Setup

The Adjoint method is useful to predict the influence of parameters or different boundary conditions on an objective function. The adjoint simulation needs the computation of the field solution, named *Primal Solution*, and after that is possible to study the sensitivity of the cost function. Once the sensitivity is obtained, the deformation of the mesh is carried out, leading to a change in geometries. At this point, it is necessary to reevaluate the flow solution around the new shapes and evaluate the effects of this change. This process can go on, following a loop, until a goal is reached.

To perform this optimization flow, it is necessary to create a new model on STAR-CCM+. The model is generally similar to the one previously used for the simulation phase of the tool but differs in some crucial points.

5.2.1 Mesh

The mesh needs to be slightly more detailed than the tool's template, especially in the area of interest, so a rectangular volumetric control is inserted to encompass the bulb, part of the wing at the graft, and the immediate surroundings (fig. 5.1). The offset around the surface remains unchanged.

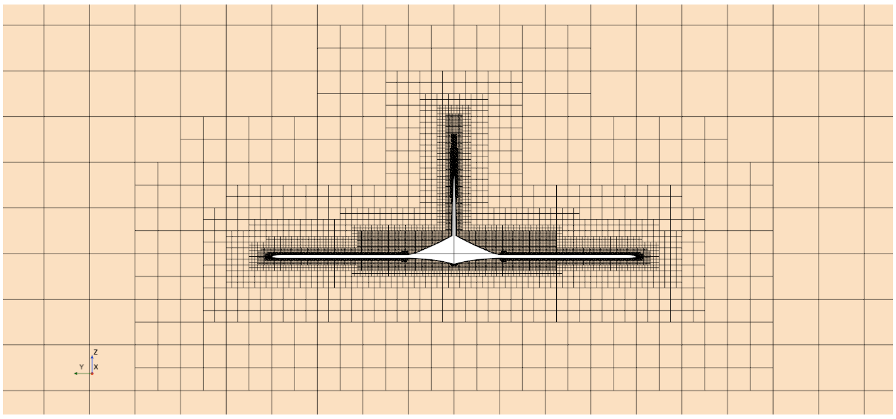


Figure 5.1: Mesh section: yz plane.

5.2.2 Physics Models and Boundaries Conditions

To use the Adjoint solver in the software, some physical models must be changed: equations can no longer be segregated but the *Coupled Flow* model must be selected. Models that activate the Adjoint solvers must also be added.

Physics Models
Adjoint
Adjoint Mesh Deformation
Constant Density
Coupled Flow
Mesh Deformation
Steady
Surface Sensitivity
Reynolds-Averaged Navier-Stokes
Three Dimensional
$k - \omega$ Turbulence

Table 5.1: Main Physics Models.

As a natural extension of the tool optimization, no initial conditions are changed from previous simulations: the rudder has an angle of attack of $3deg$ and the flow velocity is $3m/s$.

The boundary conditions of the domain remain unchanged to keep the same conditions as in previous simulations as much as possible: all sides of the fluid domain are velocity inlets, except for the outlet, which is a pressure outlet.

Flow Velocity	$3m/s$
Angle of Attack	$3deg$

Table 5.2: Initial Condition.

5.2.3 Adjoint Solver

The Adjoint equations are solved with an iterative defect-correction algorithm. As reported in the official *Simcenter STAR-CCM+ Guide* [24], usually no changes to the default settings are required if the convergence is satisfied, they have been selected as best practice.

Adjoint Solution Algorithm	GMRES-DR
Krylov Space dimension	50
Courant Number	500

Table 5.3: Adjoint Solver Setup.

Adjoint Cost Function

Parameter sensitivity should be calculated concerning a cost function that corresponds to the engineering objective that is examined during the adjoint flow analysis. Cost functions are differentiated to form the right-hand side of the adjoint flow equations. The adjoint solver solves the system of differential equations with regard to the cost function specified [24]. Using this second optimization as a normal continuation of the study with the genetic algorithm tool, again the efficiency of the entire carrier wing is chosen as the objective function. This efficiency is calculated by report as the ratio of the lift C_L and drag C_D coefficients of the hydrofoil. In this case, it was decided to maximize the efficiency E, since there is no limitation on the type of optimization (maximum or minimum) unlike the previous case where the reciprocal of E was minimized.

5.2.4 Analysis Workflow

To perform a simulation with adjoint optimization, a precise iterative scheme must be followed, as shown in figure 5.2. The physical model, initial, and boundary conditions are chosen. The mesh must be defined and the primal solution calculated. Once convergence is reached in the primal solution, a loop must be built in which the adjoint and stationary solution are solved iteratively: in the software used, it is possible to do this using the **Simulation Operations** node 5.3. For this

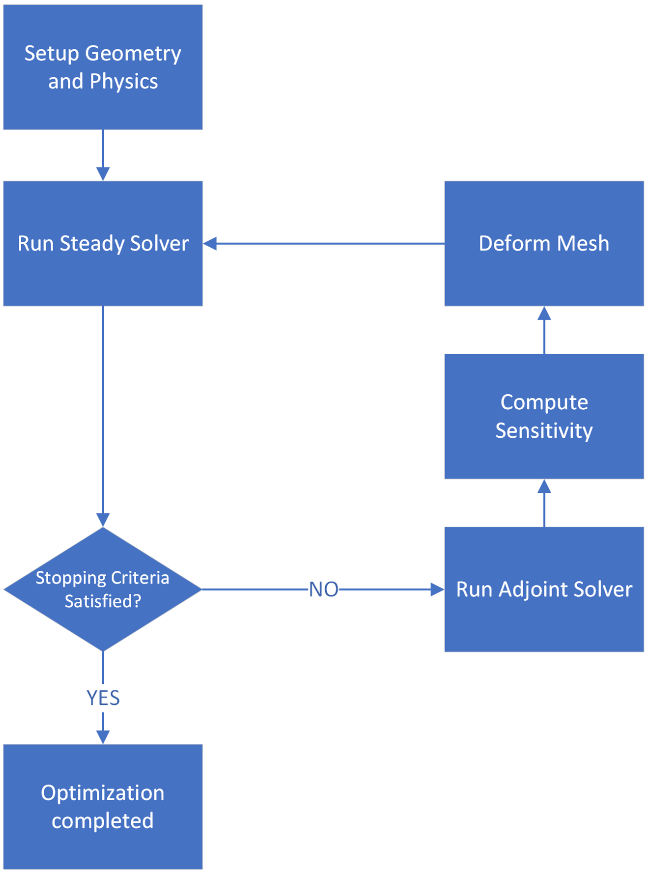
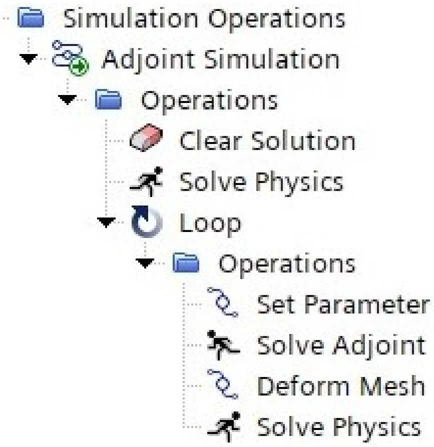


Figure 5.2: Simulation workflow.

optimization, it was chosen to perform 10 cycles of geometry optimization: to perform this, a counter was created. If the counter is lower than 10 the optimization shall continue. As long as this criterion is not fulfilled the analysis is carried out in this way:

1. updating the counter;
2. solving the adjoint equations and computing sensitivity;
3. mesh and geometry deformation;
4. solving the flow equations.

When the counter has reached value 10 the simulation ends.



- Simulation Operations
- Adjoint Simulation
  - Operations
    - Clear Solution
    - Solve Physics
    - Loop
      - Operations
        - Set Parameter
        - Solve Adjoint
        - Deform Mesh
        - Solve Physics

Figure 5.3: Simulation Operation node created to perform the optimization.

5.2.5 Mesh Deformation

To deform the mesh and modify the geometry directly within the software, control points must be inserted near the surfaces that need to be optimized. The software uses these points to deform the mesh by following field gradients. The function chosen to change the position of the control points is a simple first-order function:

$$x = x_0 + \alpha \frac{dE}{dX} \tag{5.11}$$

where x_0 is the position of the point at the previous simulation step, α is a multiplicative coefficient that defines the displacement rate by chasing the gradient of the cost function. STAR-CCM+ stores all this parameters in a function named *CumulativeMorpherDisplacement*. The function directly changes the volume of cells, inevitably causing a drastic drop in mesh qualities, even in some cases leading to a simulation crash due to the presence of negative volumes or self-intersecting cells.

5.3 Sensitivity Study to Displacement Variables

As previously mentioned, the morphing of the mesh is achieved by moving some control points. These control points displace following the gradient of the objective function and impact the volume of the computational cells based on the distance between the volume and control point. The location and number of points to be used to achieve the absolute optimum are not known, and neither is the α step to be used in the equation 5.11. For this reason, it was decided to do a quick sensitivity study of the optimization of these parameters. Using a random bulb generated by the optimization tool, preliminary optimization simulations were performed to evaluate the impact of optimization parameters on the attainable optimum point. Variable parameters are:

- the number of control points;
- the location of the control points;
- the α coefficient.

5.3.1 Number and Location of Control Points

In this software it is possible to arrange the control points individually, aligned, inside a box, or directly on the surface of the geometry to be analyzed. For the case at hand, it was considered not useful to arrange the points in line to control the displacement of the mesh. The possible arrangements of points chosen were:

- arranged around the bulb inside a parallelepiped of sides equal to the opening, chord at the root and maximum thickness of the bulb;
- directly on the surface of the bulb.

The number of points and their location are two variables that cannot be considered alone but are closely interrelated. Therefore, the two geometric configurations chosen but with varying numbers of points were evaluated for both geometries.

Simulations were run and the results were processed to seek which configuration is the best.

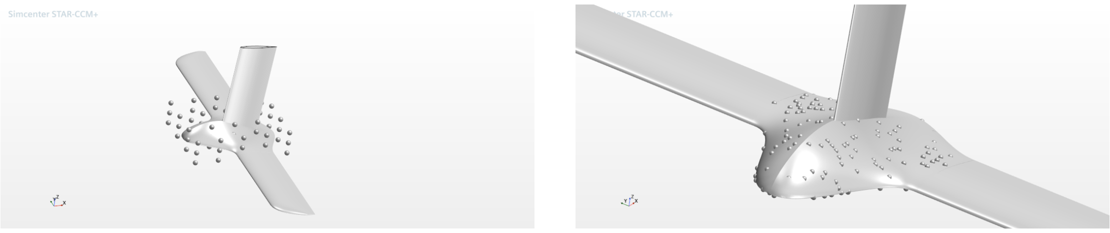


Figure 5.4: Two different control point configurations: box shape and directly on the bulb surface.

From the results of this first part of the study, shown in table 5.4, it can be seen that the optimization does not lead to impressive results: the best configuration leads to an increase in efficiency of less than 1 percent. Some simulations, those with a higher number of points, are not successful because, due to excessive mesh deformation, some interpenetration problems occur between cells. Results are also represented in figure 5.6 and figure 5.5

The best configuration turns out to be the one with 200 points arranged directly on the surface of the bulb.

Shape	n. points	Initial E	Final E	% change
Box	8	15,1227	15,2019	0,5237
Box	20	15,1227	15,2050	0,5442
Box	60	15,1227	15,2018	0,5231
Box	100	15,1227	15,1963	0,4688
Box	200	15,1227	-	-
Part	8	15,1227	15,1630	0,2665
Part	20	15,1227	15,2004	0,5138
Part	60	15,1227	15,2314	0,7188
Part	100	15,1227	15,2318	0,7214
Part	200	15,1227	15,2621	0,9218
Part	1000	15,1227	-	-
Part	2000	15,1227	-	-

Table 5.4: Result for different shapes and number of points.

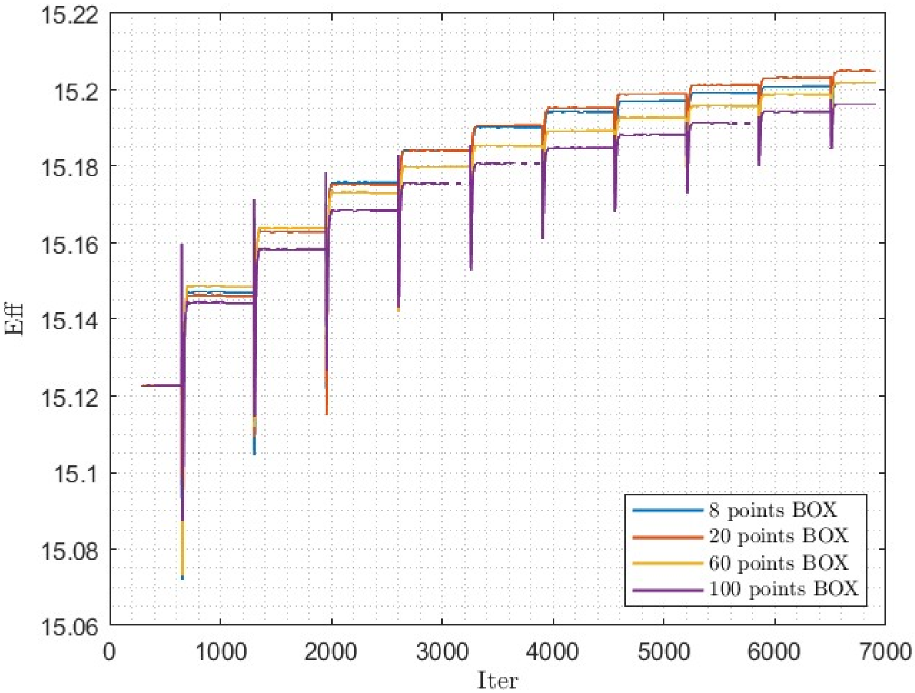


Figure 5.5: Efficiency increasing in various optimization cycles: Box configuration.

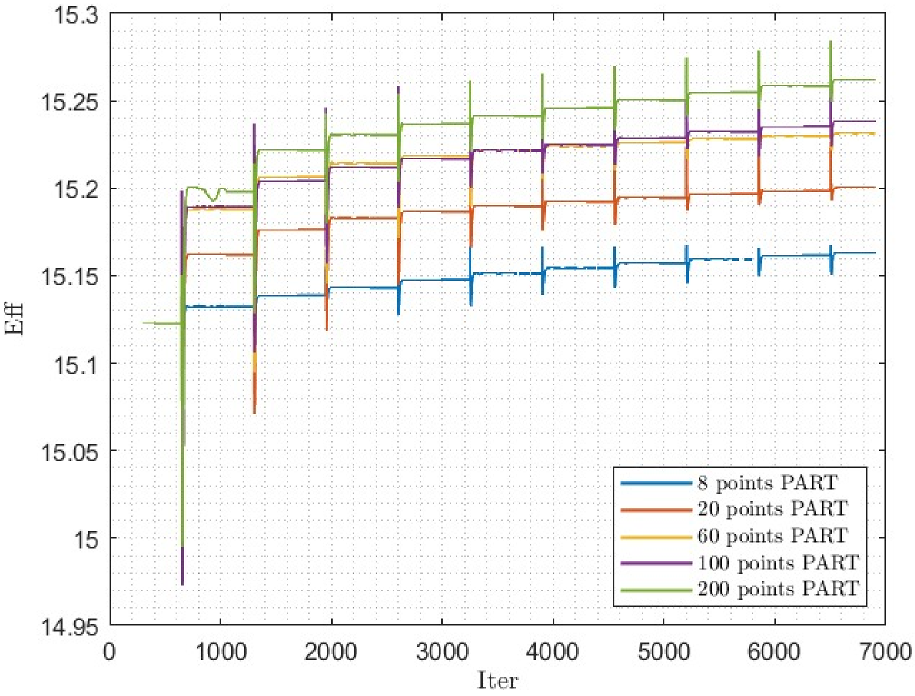


Figure 5.6: Efficiency increasing in various optimization cycles: Part configuration.

5.3.2 Displacement rate

The impact of the α multiplicative coefficient intervening in the REF equation was also evaluated. This coefficient can be viewed as the fraction of the gradient by which a control point must be moved in the identified direction to increase the objective function. The value recommended by the guide is $\alpha = 1 \times 10^{-4}$ [24]. Different values of α were evaluated for the 20-point Box configuration. Results are shown in table 5.5 and in figure 5.7

α	**Initial E**	**Final E**	**% change**
5×10^{-4}	15,1227	15.2218	0,6553
1×10^{-4}	15,1227	15,2050	0,5442
1×10^{-3}	15,1227	15,2406	0,7796
5×10^{-3}	15,1227	-	-

Table 5.5: Results for different α value.

The final efficiency increases as the displacement rate increases, however, if too

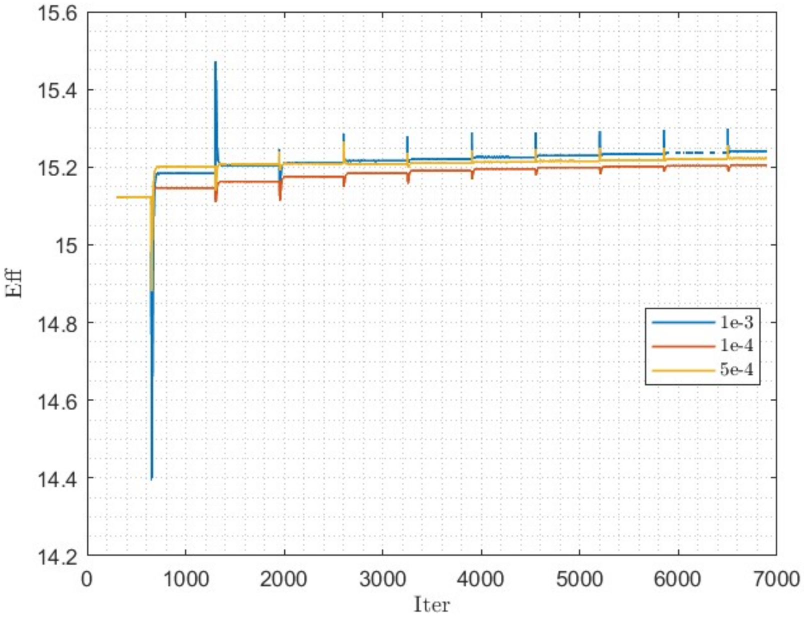


Figure 5.7: Efficiency increasing in various optimization cycles for different α values.

high, problems of excessive mesh deformation arise.

5.3.3 Additional Assessments

There are many variables to take into account, and it is not possible in this study to analyze all of them. Some other simulations were carried out to assess their possible impact on the optimization.

Increasing Cycles of optimization

The two best configurations of the number of points for the two different spatial arrangements box and part were also evaluated with a longer optimization, for 50 deformation cycles.

Shape	n. points	n. cycles	Initial E	Final E	% change
Box	20	50	15,1227	15,2231	0,6308
Part	200	50	15,1227	15,3360	1,3774
Box	20	10	15,1227	15,2050	0,5442
Part	200	10	15,1227	15,2621	0,9218

Table 5.6: Comparison between 20 and 50 cycles optimization for two different configurations.

By increasing the number of cycles, under certain conditions of position and number of points, the final efficiency increases. However, the simulation time becomes much longer, making this test rather slow by requiring a large number of computational iterations.

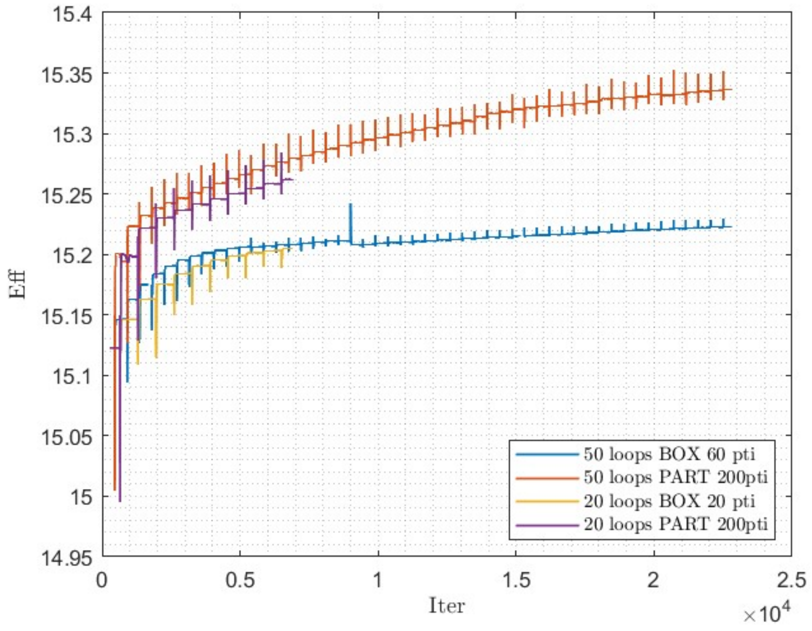


Figure 5.8: Efficiency increasing for different shape configurations and different number of cycles.

Minimize drag coefficient

To conclude, two additional simulations were also performed. In the second to last simulation, the objective function was changed to test if changing it would result in greater optimizations. The optimization objective was switched by trying to minimize the drag coefficient C_D. This simulation also did not yield high results, leading to a decrease in efficiency E.

Shape	n. points	Goal	Initial C_D	Final C_D	% change
Part	60	Minimize C_D	0.0284	0.0283	<0.14

Table 5.7: Results of optimization with the goal of minimizing drag coefficient.

Optimization of the entire wing

The second optimization, to maximize efficiency, was performed by distributing the points over the entire horizontal wing but with the possibility of deforming (thus optimizing) the entire hydrofoil. As could be expected this simulation brought greater improvements in final efficiency since the modified surface area is much larger than the bulb alone.

Shape	n. points	Goal	Initial E	Final E	% change
All Wing	2000	Maximize E	15.1227	16.2493	7.4497

Table 5.8: Results of optimization with the goal of minimizing drag coefficient.

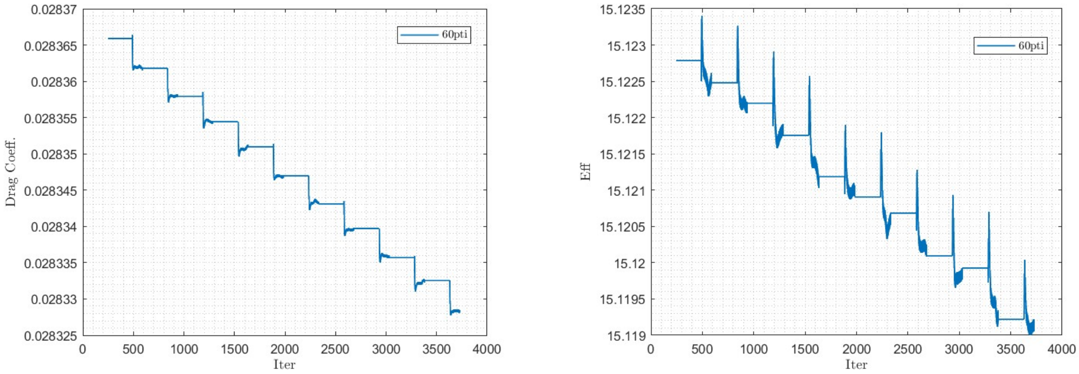


Figure 5.9: Drag coefficient and Efficiency in an optimization to decrease the drag coefficient.

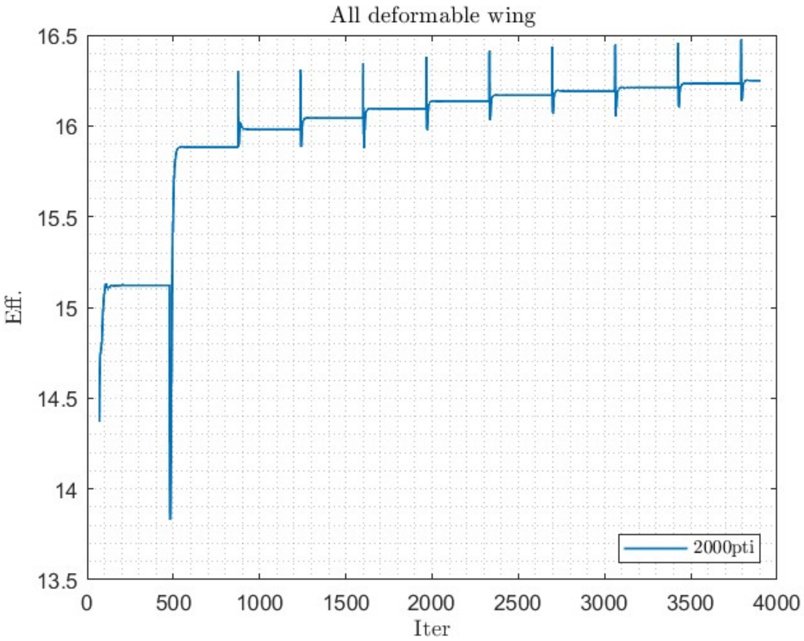


Figure 5.10: Efficiency trend in the simulation with the whole deformable wing.

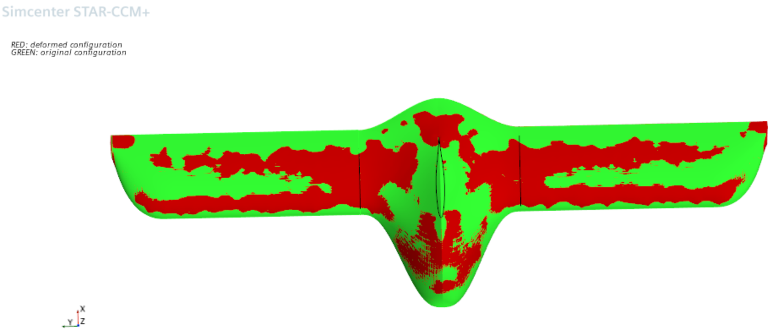


Figure 5.11: Top view of the deformed and original shapes of an all-optimized wing. In red is the optimized configuration.

5.4 Results

The simulations performed provide some insight into the adjoint optimization approach. This technique is not optimal for the case at hand. The final efficiency values turn out to be less than 1% better than the initial value. Certainly, the explanation for this behavior can be justified by the fact that the goal is to optimize the impact of a portion of the surface, the bulb, with much less area than the whole wing. The geometry of such a limited surface area has little impact on the total efficiency; actually from the optimization of the full horizontal wing (table 5.8) the efficiency improves greatly.

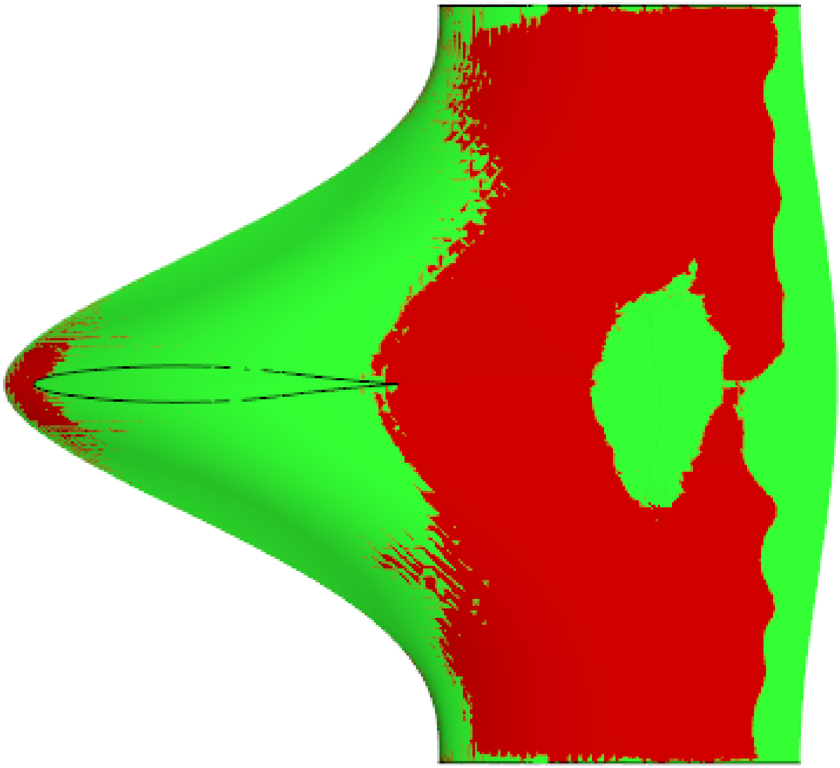


Figure 5.12: Top view of the deformed and original shapes of the bulb. In red is the optimized configuration.

Figures 5.13 and 5.12 show the final configurations of the bulb obtained from the tool in the previous chapter: the original surface (in green) and the optimized surface (in red) are compared. The optimization led, in general, to a change in the airfoil of the bulb in the tail area and by thickening the lower surface at the trailing edge. The final and initial efficiency values are shown in table 5.9.

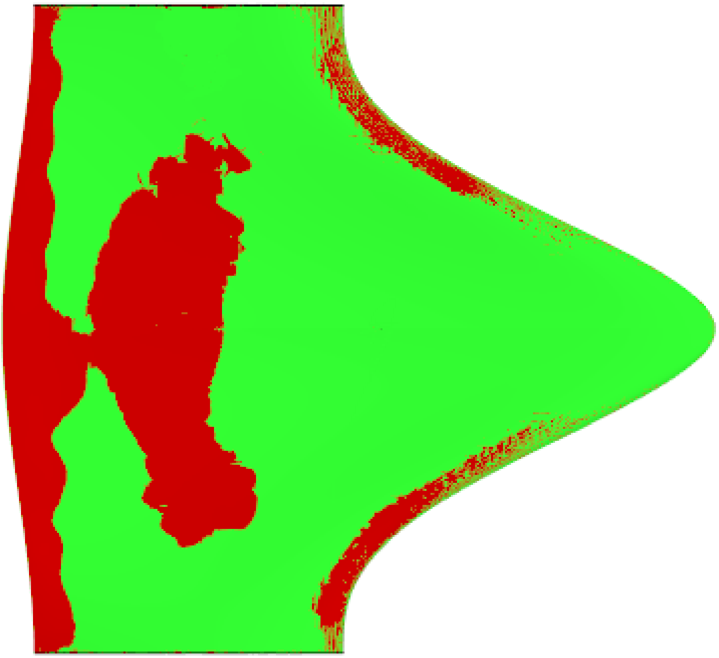


Figure 5.13: Bottom view of the deformed and original shapes of the bulb. In red is the optimized configuration.

Shape	**n. points**	**Initial E**	**Final E**	**% change**
Part	200	16.6403	16.7165	0.4581

Table 5.9: Adjoint optimization of the best scoring bulb 4.5.

Chapter 6

Conclusions

The objective of this thesis was to realize an optimization process to be integrated into the design and construction of the outer surface of the junction between the two surfaces of a T-shaped hydrofoil in small racing boats such as skiffs or moths. To achieve this goal, a tool was built in MATLAB that, through a genetic algorithm, explored a parameterized design space through the Class-Shape-Transformation technique. Using this parameterization technique, 21 parameters, deemed essential to the search for an optimum point, were selected to generate possible configurations of the bulb to be studied. The tool was implemented by seeking a compromise between simulation time (and thus its cost in terms of software license) and accuracy since they are in strong opposition. The algorithm, once defined, is capable of cyclically assigning a value to each free parameter within a range of possible values for that parameter and designing the corresponding bulb. Once the feasibility and validity of the bulb surfaces have been assessed, it is plugged into the reference wing of the hydrofoil to be optimized and its total efficiency will be evaluated through CFD analysis in STAR-CCM+. Through the mechanism of the genetic algorithm and the logic behind it, a winning geometry was selected, that is, the one characterized by the highest efficiency E. Once this optimization was completed, the possibility of performing another optimization through the native Adjoint solver of the CFD software used was evaluated: several optimization configurations were evaluated, and once the optimal one was chosen, the best bulb obtained with the tool was attempted to be optimized.

6.1 Summary of Results

The results obtained should not be interpreted in terms of the value achieved but by observing their trends across the various generations in the algorithm. In fact, the value history and its trend allow us to derive some information that could be generalized in the design of different bulbs, at the interlocking of wings with geometries different from the one used as a reference in this study. The most significant parameters in the design context are those in figure 4.10, which are the actual dimensions of the bulb plant. From the trend of these parameters over the generations (only the best individual for that generation is highlighted in the graph), some useful information can be deduced.

The graft position of the vertical surface along the bulb chord tends to be located the closest to the leading edge, thus being offset from the horizontal wing. This type of configuration with unaligned lifting surface and drift is quite common; an example is shown in figure 1.5.

The root chord moves toward a value about twice the size of the parameter x_{LEtip}, which would be the x position of the leading edge at the tip. This again may be related to the stagging of the perpendicular surfaces, which reduces the interaction zone and limits the fluid re-circulation zones.

The span of the bulb, as early as in the first generations, tends to move toward a maximum value, almost to the limit of the constraint given in the range definition, increasing wetted surface area.

Concerning the optimization study through adjoint, despite finding a suitable configuration for the case study, and spreading the points most suitably, the results do not lead to significant improvements. This type of approach may prove more useful in optimizing a larger surface area, as obtained for the entire horizontal wing in table 5.8.

6.2 Critical issues and possible Improvement

Some aspects show possible critical issues or some details that could be improved in the future to achieve better results. To allow a fair rate of exploration of the design space by the genetic algorithm, the parameters describing the bulb were restricted, especially regarding the degree of the Bernstein polynomials describing the airfoils and span-wise sections. By choosing degree $Nx = 3$ and $Ny = 2$ an attempt was made to avoid disproportionate growth in the numbers of parameters describing the surfaces, which turn out to be $2 \cdot (Nx + 1) \cdot (Ny + 1)$ for an integer bulb. However, a limitation on possible geometries is imposed, limiting the variety of airfoils and especially span-wise curves.

Another parameter that could be improved is the one describing the shape of the leading edge. In this study, the leading edge is described through the fundamental points of the bulb plan, and these two are joined through one of the seven functions reported in the appendix. If there were a chance to expand the design space, it might be useful to release this parameter from the analytic functions reported and instead parameterize the curve of the leading edge in a different way: one possible idea might be to describe it through a third-grade line using a CST technique, which as seen is convenient for this type of problem.

One correction that is considered can be made concerns the b span of the bulb. It is evident from the results how the best-performing individuals have as large a span as possible, at the limits of the imposed constraint of $0.4m$, as well as how the best element presents $b = 0.3971m$. This behavior may result from the effect that increasing the span of the bulb increases the aspect ratio of the whole wing and consequently a benefit in efficiency is derived. To investigate whether this is indeed the cause of the span behavior, an artifice for fixing the aspect ratio of the wing could be introduced into the tool. For each individual produced with b less than the expected value from the design aspect ratio, a wing log long $(AR \cdot C_t - b)$ could be added to have the desired aspect ratio but with a reduced bulb's size.

Appendix A

Leading Edge Functions

This appendix shows the functions that were chosen to represent the curves describing the leading edge of the bulbs. These functions were derived analytically and depend on the free parameters b bulb span, and x_{LEtip}, distance x of the leading edge at the tip. For some functions, the k value of the slope at the tip of the bulb is given, while where not specified $k = 0$ is meant.
All curves, except for the linear function(equation A2), exhibit a null slope at $x = 0$, on the axis of symmetry of the bulb, coincident with the root chord.

All functions are evaluated in the interval $[0; b/2]$ by the tool to calculate local bulb chords, defined as the distance between the trailing edge and the leading edge.

1.
$$f_1(x) = spline(yy, [0; xx; 0]) \tag{A.1}$$

2.
$$f_2(x) = \frac{x_{LE}(tip)}{b/2} \cdot x \tag{A.2}$$

3.
$$f_3(x) = \left(\frac{x_{LE}(tip)}{(b/2)^2}\right) \cdot x^2 \tag{A.3}$$

4.
$$f_4(x) = \Big(100k - 2000x_{LE}(tip)\Big) \cdot x^3 + \Big(300x_{LE}(tip) - 10k\Big) \cdot x^2 \qquad k = 1 \tag{A.4}$$

5.
$$f_5(x) = \frac{3x_{LE}(tip)}{(b/2)^4} \cdot x^4 - \frac{8x_{LE}(tip)}{(b/2)^3} \cdot x^3 + \frac{6x_{LE}(tip)}{(b/2)^2} \cdot x^2 \tag{A.5}$$

6.
$$f_6(x) = \frac{6x_{LE}(tip)}{(b/2)^5} \cdot x^5 - \frac{15x_{LE}(tip)}{(b/2)^4} \cdot x^4 + \frac{10x_{LE}(tip)}{(b/2)^3} \cdot x^3 \tag{A.6}$$

7.

$$f_7(x) = \frac{k - 2x_{LE}(tip)}{(b/2)^3} \cdot x^3 + \frac{1}{b/2}\left(\frac{3x_{LE}(tip)}{b/2} - k\right) \cdot x^2 \qquad k = 0.5 \qquad \text{(A.7)}$$

Bibliography

[1] «Italian Hydroplane of Curious Type.» In: *Popular Mechanics* (Dec. 1991), p. 927 (cit. on p. 1).

[2] A J Acosta. «Hydrofoils and Hydrofoil Craft». In: *Annual Review of Fluid Mechanics* (1973), pp. 161–184. DOI: 10.1146/annurev.fl.05.010173.001113 (cit. on p. 1).

[3] M. Sheahan. «The foiling phenomenon – how sailing boats got up on foils to go ever-faster.» In: *Yachting World* (2015). URL: https://www.yachtingworld.com/special-reports/the-foiling-phenomenon-66269 (cit. on p. 4).

[4] Y. Liao, A. Yildirim, J. R.R.A. Martins, and Y. L. Young. «RANS-based optimization of a T-shaped hydrofoil considering junction design». In: *Ocean Engineering* 262 (2022), p. 112051. DOI: https://doi.org/10.1016/j.oceaneng.2022.112051 (cit. on p. 5).

[5] Xiaolong W., Zaojian Z., Tao F., Li X., and Weidong L. «Design and optimization of rudder bulb based on CFD and genetic algorithm». In: *Ocean Engineering* 265 (2022). ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.112601 (cit. on p. 9).

[6] Sobel K., M. Su H., Gong, and G. Liangxian. «Three-Dimensional CST Parameterization Method Applied in Aircraft Aeroelastic Analysis». In: *International Journal of Aerospace Engineering* (2017). DOI: https://doi.org/10.1155/2017/1874729 (cit. on p. 9).

[7] B. M. Kulfan. «Universal Parametric Geometry Representation Method». In: *Journal of Aircraft* 45 (Feb. 2008), pp. 142–158 (cit. on p. 11).

[8] B. M. Kulfan and J. E. Bussoletti. «Fundamental Parametric Geometry Representations for Aircraft Component Shapes». In: *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference: The Modeling and Simulation Frontier for Multidisciplinary Design Optimization* AIAA Paper 2006-6948 (2006) (cit. on pp. 11–13, 15).

[9] B. M. Kulfan. «Universal Parametric Geometry Representation Method: CST». In: *45th AIAA Aerospace Science Meeting and Exhibit* AIAA Paper 2007-0062 (2007) (cit. on pp. 16, 19).

[10] R. Arina. *Fondamenti di Aerodinamica.* Torino: Levrotto&Bella, 2015 (cit. on p. 30).

[11] M. Peric J. H. Ferziger. *Computational Methods for Fluid Dynamics.* Third. Berlin: Springer, 2002 (cit. on pp. 30, 32, 33).

[12] S. Rodriguez. «LES and DNS Turbulence Modeling». In: *Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques.* Cham: Springer International Publishing, 2019, pp. 197–223. DOI: 10.1007/978-3-030-28691-0_5 (cit. on p. 31).

[13] Wilcox D.C. «Turbulence Modeling for CFD». In: DCW Industries, Inc, 2006. ISBN: 978-1-928729-08-2 (cit. on p. 34).

[14] L. Eça and M. Hoekstra. «A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies». In: *Journal of Computational Physics* 262 (2014), pp. 104–130. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2014.01.006 (cit. on pp. 37, 38).

[15] I. Capuzzo Dolcetta, F. Lanzara, and A. Siconolfi. «Appunti di Lezioni di Ottimizzazione». In: *Dipartimento di Matematica, Guido Castelnuovo, Sapienza Università di Roma* (2010) (cit. on pp. 41, 43).

[16] J. R. R. A. Martins and A. Ning. *Engineering Design Optimization.* Cambridge University Press, 2022. ISBN: 9781108833417 (cit. on pp. 42, 44, 45).

[17] M. Salmeri. *Progettazione Automatica di Circuiti e Sistemi Elettronici.* Vol. 3. Università degli studi di Roma Tor Vergata (cit. on p. 45).

[18] W. Xiaolong, Z. Zaojian, F. Tao, X. Li, and L. Weidong. «Design and optimization of rudder bulb based on CFD and genetic algorithm». In: *Ocean Engineering* 265 (2022), p. 112601. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.112601. URL: https://www.sciencedirect.com/science/article/pii/S0029801822018844 (cit. on p. 45).

[19] T. Mathew. *Genetic Algorithm.* 2021. URL: https://1library.net/document/q7xwxrdy-genetic-algorithm-assistant-professor-department-engineering-institute-technology.html (cit. on p. 46).

[20] S. Katoch, S.S. Chauhan, and V. Kumar. «A review on genetic algorithm: past, present, and future». In: *Multimed Tools Appl* 80 (2021). DOI: 10.1007/s11042-020-10139-6 (cit. on p. 47).

[21] *What is the Genetic Algorithm?* MathWorks: MATlab Documentation. 2023. URL: https://it.mathworks.com/help/gads/what-is-the-genetic-algorithm.html (cit. on p. 47).

[22] *Genetic Algorithm Options*. MathWorks: MATlab Documentation. 2022. URL: https://it.mathworks.com/help/gads/genetic-algorithm-options.html?searchHighlight=population%20options&s_tid=srchtitle_support_results_1_population%2520options (cit. on p. 50).

[23] Pierce N.A. Giles M.B. «An Introduction to the Adjoint Approach to Design». In: *Flow, Turbulence and Combustion* 65 (2000), pp. 393–415. DOI: doi.org/10.1023/A:1011430410075 (cit. on pp. 64, 65).

[24] *Adjoint Solver Reference*. Simcenter STARCCM+ Guide. URL: file:///C:/Program%20Files/Siemens/17.04.008-R8/STAR-CCM+17.04.008-R8/doc/en/online/index.html#page/STARCCMP%2FGUID-7B87A2C7-840A-4785-B0BA-B0D8005E8A6F.html%23 (cit. on pp. 67, 72).