POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Adjoint Shape Optimization: Cavitation Reduction for Hydrofoils

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria aeronautica

**Maddalena Rossi, 963124**

**Advisor:**
Full Professor Alberto Guardone

**Co-advisors:**
Phd Student Luca Abergo

**Academic year:**
2022-2023

**Abstract:** In this thesis, the problem of cavitation reduction is addressed through shape optimization using the adjoint method. The target is to design an optimized hydrofoil that minimizes the possibility of having cavitation effects under specific flow conditions. Indeed, cavitation is an undesirable phenomenon that occurs when the local pressure on a hydrodynamic surface drops below the vapor pressure of water, generating vapor bubbles that can cause structural damage and reduce aerodynamic efficiency. The SU2 finite volume solver is adopted to simulate the flow around the hydrofoil and compute the cavitation phenomenon. The discrete adjoint method is applied to compute the objective gradient with respect to the design variables, guiding the optimization towards the most suitable shape for cavitation reduction. The sensitivity verification was performed on a 2D profile. The cavitation optimization was carried out on a real 3D hydrofoil of the PoliMi Sailing Team moth.

**Key-words:** cavitation, hydrofoils, optimization, adjoint

## 1. Introduction

Sailboat foils are an innovative technology that is revolutionizing the sailing world. Foils are hydrodynamic appendages that are mounted under the boat's hull and allow it to be lifted out of the water, reducing friction and increasing speed. Sailboat foils represent a real quantum leap in sailing and are set to change the way we think about this sport and about recreational sailing. Since the hull is lifted out of the water, the submerged surface in the water, i.e. that of the foils, is much smaller. Wave resistance in the flight phase no longer exists, so the shape of the submerged appendages becomes more important (from the original concept [26], to the new contribution [42],[45]). These aerodynamic lifting surfaces are highly sensitive to geometric modifications and even slight changes in the shape can have a significant influence on the final design's performance.
In the context of designing foils for sailboats, the phenomenon of cavitation that can occur on the surface of the appendages, due to high sailing speeds, must be taken into account. Cavitation is a physical phenomenon that occurs when the pressure of a fluid reaches or drops below its vapor pressure, leading to the formation of vapor bubbles within the fluid. Avoiding cavitation is crucial, since it will have a detrimental effect on the hydrodynamic performance of the foil. Such physical phenomenon is present in many engineering systems, such as pumps, marine propellers, hydroelectric turbines and pipelines, and can cause structural damage and efficiency losses (e.g. [14] and [10]). In recent decades, cavitation research has been the subject of increasing scientific and technological interest due to its implications in many industrial applications. Indeed, understanding the

dynamics of cavitation bubbles and their interactions with surrounding fluids is essential for improving the energy efficiency of systems and preventing damage to structures.
Cavitation is indeed an important concern for the hydrofoil performance, such as [37], but its numerical prediction remains a difficult problem, [22], [9], [12], [2]. These difficulties explain the reason why cavitation aspects are usually not considered in hydrofoil shape optimization, unless when the objective is precisely to delay the cavitation, such as in [46]. The goal of this thesis is not to simulate the cavitation phenomenon, but rather to search for an optimized shape to prevent its occurrence. Usually, optimization to avoid cavitation is performed using the inverse design method. This method works by prescribing the desired pressure distribution, based on which the profile deformation is obtained. The disadvantage of this method is that it is not trivial to determine the pressure distribution to impose, especially in the three-dimensional case. Instead, in this thesis, the use of adjoint shape optimization is proposed to obtain a profile where the potential cavitation area is minimized. The advantage of adjoint shape optimization is that there is no need to know the pressure distribution beforehand. Usually, optimization to avoid cavitation is performed using the inverse design method. This method works by prescribing the desired pressure distribution, based on which the profile deformation is obtained. The disadvantage of this method is that it is not trivial to determine the pressure distribution to impose, especially in the three-dimensional case [28]. Instead, in this thesis, the use of adjoint shape optimization is proposed to obtain a profile where the potential cavitation area is minimized. The advantage of adjoint shape optimization is that there is no need to know the pressure distribution beforehand. The research project aims to optimize the profile shape of both a 2D and a 3D foil, with the objective of minimizing cavitation. To accomplish this, in this thesis we worked inside the optimization chain implemented in the open source finite volume solver SU2, Stanford University Unstructured software [30]. In particular, to investigate this issue, a new objective function has been coded to estimate an area of possible cavitation.
This research contributes to the existing body of knowledge in hydrofoil design, exploiting a high fidelity optimization method [5] and provide novel insights into the potential of adjoint methods in tackling cavitation-related challenges.
In general, the Automatic Shape Optimization (ASO) has been proved to be a powerful tool able to improve the aerodynamic performance in aeronautical world; it is not been highly investigated in nautical research ([15] and [19]). It is precisely within this context that this research work aims to demonstrate the feasibility of the ASO method in the nautical field. Concerning the aerospace world a large number of scientific papers highlights the sharp and increasing interest of the CFD community and the aerospace industry. In fact, a recent review of the state of the art regarding shape optimization [40] identifies a total of 304 remarkable papers. In the last forty years, researchers have addressed their efforts towards making shape optimization reliable and robust, although many improvements are possible and necessary, especially concerning multi-disciplinary design and multi-objective optimization. In simpler terms, optimization refers to finding the best possible configuration. In the case of gradient-based ASO, the shape that is obtained may not be the absolute best but it is definitely better than the original one. The starting point, the type and number of constraints, and the size of the design space all have a significant impact on the final outcome, as proved in [8].
The optimization process involves multiple interconnected steps, which can be quite complex due to the involvement of various tools, sub-problems, and mathematical aspects that are often still an active field of research. In a study by Shahpar from Rolls Royce [38], seven different modules were identified and described, each with its own clear inputs, processes, and outputs. These seven modules are referred to as the "optimization seven pillars" and include geometry parametrization, mesh generation and deformation, flow solver, optimization algorithms, postprocessing, workflow management, and IT issues. Since these modules are interconnected and affect each other, a consistent and unified strategy is required for the optimization process to be successful. Developers must have a comprehensive understanding of the context, including both mathematical and programming challenges.
In the early 1980s, the community working on Computational Fluid Dynamics (CFD) began to enhance their software by adding the capability to perform sensitivity analysis. This involves calculating the first derivatives of an aerodynamic property that depends on both the geometry and flow, with respect to certain design variables. These DVs are parameters that control functions that describe the body's surface. Initially, Finite Differences was the most straightforward method used to calculate the gradients of interest. This method was introduced by Reneaux and Thibert [33] and did not require modifications to the solver itself, but the computational cost increased with the number of design parameters. A more advanced method, the complex step FD, was proposed by Lyness and Morelli [24], which partially addressed some of the issues associated with Finite Differences. However, the turning point for shape optimization in aerodynamics was the application of optimal control theory to incompressible flow equations, which was introduced by Pironneau [32]. The continuous adjoint method for aerodynamics, which is a shape optimization method, became feasible and attractive only after the famous article by Jameson in 1988 [16]. For the first time, it became possible to calculate the objective sensitivity with respect to design variables, and the cost of this computation scaled with the number of objective functions. The applications progressed from 2D airfoils to 3D wings, and finally to the optimization of entire aircraft. The continuous adjoint approach theory is a natural extension of the linear algebraic duality theory to partial differential equations (PDEs). In this approach, a duality variables vector, called adjoint variables, is added to

the variables space containing the flow state. This requires a deep manipulation of the flow equations and the boundary conditions, and only at the end are the equations discretized to perform the numerical solution. The mathematical complexity of the adjoint and of its corresponding boundary conditions was partially overcome by Shubin and Frank [39], who created a discrete version of Jameson's adjoint method called the "implicit gradient approach", which was later renamed the "discrete adjoint method". In this approach, the control theory is directly applied to the set of discrete flow equations.
Within the context of discrete adjoint, an extension to cavitation has been performed in this work. SU2 is an aerodynamic solver that does not have the capability to simulate the water-air interface. Indeed, SU2 does not have any multiflow interface capturing algorithm, the most common method that could be implemented is the Volume of Fluid (VOF) method ([31]), which is usually exploited for nautical researches. Despite this, SU2 is chosen because it has the discrete adjoint implemented in combination with automatic differentiation which makes easy to code a new objective function and obtain the relative sensitivity. Although it does not have cavitation models, it is possible to calculate the area where cavitation could occur. The objective of this work is precisely to reduce this area maintaining the hydrodynamic performance.
This thesis provides a comprehensive presentation of the optimization chain, the SU2 software, sensitivity computation, and the implementation of the cavitation coefficient (Sec. 2.3). The validity of the simulations and of the newly implemented code is demonstrated through a dedicated chapter (Sec. 3) focusing on flow validation and sensitivity verification. Subsequently, the results of both 2D and 3D cases are presented (Sec. 4). In the 2D investigation, the optimization was performed on a NACA0015 airfoil, while for the 3D optimization, the primary foil of the Moth, designed and manufactured by the PoliMi Sailing Team, was considered.

Figure 1: PoliMi Sailing Team prototype

## 2. Methodology

In this work, the discrete adjoint formulation is implemented using automatic differentiation (AD) to simplify the process. AD was developed based on the understanding that any simulation code, no matter how complex, can be broken down into a series of basic operations with well-known differentiation rules. By applying the chain rule iteratively within the computer program, it becomes possible to compute both the simulation output and its derivative with respect to specific design variables simultaneously.
The Stanford University Unstructured software (SU2) is a tool for solving partial differential equations, multi-physics analysis, and pde-constrained optimization problems on structured and unstructured grids. The software includes a RANS iterative solver for simulating compressible, turbulent flows commonly found in problems in aerospace engineering. A number of convective fluxes discretization schemes have been implemented, such as the Jameson-Schimdt-Turkel (JST) scheme [44] and the upwind Roe scheme [34]. The turbulence can be either modeled by the Spalart-Allmaras(S-A) model [41] or the Menter Shear Stress Transport (SST) Model [25]. The discretization of Navier-Stokes equations is performed using the Finite-volume method on a vertex-based median-dual grid, with several numerical schemes to solve the convective fluxes implemented. The software can also be used for multi-physics problems, including fluid-structure interaction problems and acoustics, as well as for automatic shape optimization. The surface sensitivity is computed using the discrete adjoint and projected into the design space, with the body described using the Free Form Deformation (FFD) method [36]. The mesh around the body is updated, and the optimization algorithm used is gradient-based. One notable

advantage of AD, thanks to its construction, is that it avoids introducing truncation errors typically associated with traditional finite difference methods. In other words, the derivatives calculated using AD are accurate up to the precision of the machine, without any loss of accuracy.

## 2.1. Optimization chain

In general, an optimization problem can be mathematically expressed as the minimization of an objective function J(.), typically representing an aerodynamic coefficient. The optimization problem can be formulated as follows:

$$\begin{aligned} \min_{\alpha} \quad & J(U(\alpha), X(\alpha)) && (1)\\ \text{subject to} \quad & R(U(\alpha), X(\alpha)) = 0 && (2)\\ & && (3) \end{aligned}$$

where $U$ is the state variable. The surface of an object and the flow volume are discretized using a grid called $X(\alpha)$, which is obtained by applying the $X_{vol}(X_{surf}(\alpha))$ transformation. Both, $U$ and $X$, are functions dependent on the design variables $\alpha$.
Note that $R(U)$ might not only include the flow residual but also residuals of other coupled models. Consequently, $U$ might also consist of the variables of these additional equations, for example, in the case of the RANS equations plus a turbulence model, we have:

$$U := \begin{pmatrix} U_f \\ U_t \end{pmatrix}, \quad R(U) = R(U_f, U_t) \tag{4}$$

The Reynolds-Averaged Navier-Stokes (RANS) equations and turbulence models are discretized in SU2 using the Finite-Volume method. This discretization is carried out on a vertex-based median-dual grid. By employing the implicit Euler discretization method, we naturally arrive at a damped Newton method for solving the equation $R(U) = 0$. Consequently, if convergence is achieved, the resulting solution $U^*$ is solely dependent on the residual $R(U)$. This implies that the residual can be a reasonable approximation of the Jacobian matrix $\frac{\partial R}{\partial U}$. The detailed steps of this proof are presented in [3]. This can be illustrated by transforming the solution process of the coupled equations $R(U) = 0$ into a fixed point equation, allowing the computation of feasible flow and turbulent solutions through iterations:

$$U^{n+1} = U^n - P^{-1}(U^n)R(U^n) =: G(U^n) \tag{5}$$

It is natural to assume that G is stationary only at feasible points, i.e.

$$R(U^*) = 0 \iff U^* = G(U^*) \tag{6}$$

The explicit construction of the Jacobian matrix $\frac{\partial R}{\partial U}$, however, is generally a formidable task. To circumvent this problem, a method for solving the adjoint system was proposed by Korivi et al. [18], which resembles the iterative flow solver and permits the use of the same approximative Jacobian. In this work, the approach proposed by [18] is adopted and combined with the efficient evaluation of the occurring gradients using AD techniques.
Since the computational mesh is subject to change, all functions are now considered to additionally depend on $X$. To formally handle the surface and mesh deformation, it is added as a constraint to the original optimization problem by using the equation $M(\alpha) = X$. A similar approach for dealing with the mesh sensitivities was originally proposed by Nielsen and Park. However, in the present case, no assumptions are made about the structure of $M$, except that it is differentiable. Then the optimization problem finally takes the form:

$$\begin{aligned} \min_{\alpha} \quad & J(U(\alpha), X(\alpha)) && (7)\\ \text{subject to} \quad & G(U(\alpha), X(\alpha)) = 0 && (8)\\ & X(\alpha) = M(\alpha)\,. && (9) \end{aligned}$$

We can define the Lagrangian associated to this problem as

$$\begin{aligned} L(\alpha, U, X, \bar{U}, \bar{X}) &= J(U, X) + [G(U, X) - U]^T \bar{U} + [M(\alpha) - X]^T \bar{X} && (10)\\ &= N(U, \bar{U}, X) - U^T \bar{U} + [M(\alpha) - X]^T \bar{X} && (11) \end{aligned}$$

where N is the shifted Lagrangian:

$$N(U, \bar{U}, X) := J(U, X) + G^T(U, X)\bar{U} \tag{12}$$

If $L$ is differentiated with respect to $\alpha$ using the chain rule, we can choose the adjoint variables $\bar{X}$ and $\bar{U}$ in such a way, that the terms $\frac{\partial U}{\partial \alpha}$ and $\frac{\partial X}{\partial \alpha}$ can be eliminated. This leads to the following equation for $\bar{U}$ and $\bar{X}$:

$$\bar{U} = \frac{\partial}{\partial U} N(U, \bar{U}, X) = \frac{\partial}{\partial U} J^T(U, X) + \frac{\partial}{\partial U} G^T(U, X)\bar{U} \tag{13}$$

$$\bar{X} = \frac{\partial}{\partial X} N(U, \bar{U}, X) = \frac{\partial}{\partial X} J^T(U, X) + \frac{\partial}{\partial X} G^T(U, X)\bar{U} \tag{14}$$

Finally, the derivative of the Lagrangian, that is, the total derivative of $J$, reduces to

$$\frac{dL^T}{d\alpha} = \frac{dJ^T}{d\alpha} = \frac{d}{d\alpha} M^T(\alpha)\bar{X} \tag{15}$$

Equation 13 is a fixed-point equation in $\bar{U}$ and can be solved in the style of the flow solver using the iteration

$$\bar{U}^{n+1} = \frac{\partial}{\partial U} N(U^*, \bar{U}^n, X) \tag{16}$$

once we have found a numerical solution $U = U^*$ of equation 5. Since $G$ is a contractive function if the flow solver has reached a certain level of convergence (i.e. $||\frac{\partial G}{\partial U}|| < 1$ in some suitable matrix norm), also $\frac{\partial N}{\partial U}$ will be contractive since

$$\left\| \frac{\partial}{\partial \bar{U}} \left[ \frac{\partial N}{\partial U} \right]^T \right\| = \left\| \left[ \frac{\partial G}{\partial U} \right]^T \right\| = \left\| \frac{\partial G}{\partial U} \right\| < 1 \tag{17}$$

Thus, it directly inherits the convergence properties of the flow solver. Up to now the derivation of the discrete adjoint solver was rather abstract as we did not specify yet how to compute the necessary gradients. But as shown in [4] it turns out that the sensitivity equation 14 and equation 16 can easily be evaluated using Algorithmic Differentiation applied to the underlying routines in the program that compute $G$.
From this assumption derives the practical necessity to obtain a very small final residual during the iterative solution of the flow equations. It is also possible to perform optimization with multiple objective functions, combining them and assigning specific weights determined by the user. In the case of a wing or airfoil, for example, the goal is to simultaneously minimize drag and reduce the moment around a particular axis. Aero-constraints can also be imposed, such as maintaining lift or pitch moment within a certain range. However, the computational cost and time required for sensitivity computation increase linearly. Geometrical constraints, on the contrary, can be introduced without significantly affecting the CPU time, such as specifying a maximum airfoil thickness or ensuring the wing volume does not decrease.
In the case of SU2, the Sequential Least Squares Programming (SLSQP) [17] algorithm drives the optimization process, terminating when the Karush-Kuhn-Tucker (KKT) [21] conditions are satisfied or the maximum number of design loops is reached. The SLSQP treats the computational fluid dynamics solver as a black-box, having no knowledge of the physics involved. It requires inputs such as the value of the objective function $J$, the design variables $\alpha_t$, and the sensitivity $\frac{dJ}{d\alpha}$, and provides an output of a new point in the design space, represented by a new vector of design variables $\alpha_{t+1}$, which is expected to yield an improved value of $J$. Figure 2 illustrates the process within a single design loop, with each component corresponding to a specific mathematical method chosen for its properties. Together, they determine the overall behavior of the optimization chain.

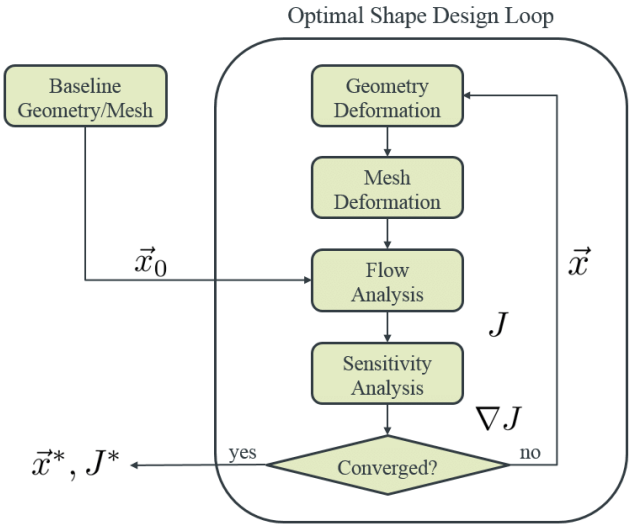



Figure 2: Optimization chain

The entire ASO framework presented in this study is implemented in the open-source multiphysics solver SU2, where the discrete adjoint code was developed by Albring, Saugeman, and Gauge [3]. The choice of SU2 is motivated in Sec.2.2. The software structure combines automatic differentiation (AD) and discrete adjoint, which is conceptually straightforward but more complex from a coding perspective. Once the objective function $J$ is computed, solving the relevant Reynolds-averaged Navier-Stokes (RANS) equations and obtaining surface sensitivity are accomplished using the adjoint theory.

To explore the entire design space, the software must have the capability to deform the body and update the mesh. The proposed ASO approach is CAD-free, meaning the loss of the defining functions of the continuous body, and working directly with the discrete counterpart, which is the grid. To enable body deformation, the wall boundaries need to be mathematically described. The relationship between the object's surface and the design variables is established through a shape parametrization method. In this study, for the 2D case, the free-form deformation method is employed. FFD directly parameterizes the locations of the nodes, eliminating the need for a geometric abstraction of the shape, although it is suitable for handling only small to medium changes in geometry. The basic idea is to enclose the body and mesh within a flexible plastic box, and when the box is deformed, the surface grid is consistently shifted. Mathematically, FFD involves a mapping from $R^3$ to $R^3$ using a tensor product Bernstein polynomial. First, a local coordinate system is set inside the delimited volume, any grid point $X$ inside the control box has coordinates $(s, t, u)$, also called lattice coordinates:

$$X = X_0 + sS + tT + uU \tag{18}$$

The box is divided into sub-control volumes with dimensions $l \times m \times k$. These volumes have vertex coordinates $(i, j, k)$ with respect to a global reference system, represented by the matrix $P_{i,j,k}$. Some or all of these points can be used as design variables in the optimization process. The desired displacement $P_{i,j,k}$ is obtained using the discrete adjoint method and then projected onto the design space. The movement of each point in local coordinates is calculated as follows:

$$x(s,t,u) + \Delta x(s,t,u) = \sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n}[B_{l-1}^{i-1}(s) \cdot B_{m-1}^{j-1}(t) \cdot B_{n-1}^{k-1}(u)] * [P_{i,j,k} + \Delta P_{i,j,k}] \tag{19}$$

where the Bernstein polynomial of degree l-1, also called the blending function here chosen for the clarity of the discussion, is determined as follows:

$$B_{l-1}^{i-1}(s) = \frac{(l-1)!}{(i-1)!(l-1)!} s^{i-1}(1-s)^{l-i} \tag{20}$$

In this work, SU2 does not utilize B-Spline with local support, although it is available. Instead, Bezier's curves with global support are used. The optimization problem involves the positions of the vertices of the sub-control volumes as design variables.

The adjoint computation incorporates a routine to update the fluid grid, which directly affects the sensitivity field. The ability to deform the mesh using mesh deformation techniques allows for handling arbitrary geometry and applying significant displacements to the nodes. This capability greatly impacts the exploration of the design space. It is crucial to maintain the quality of the output grid to prevent the divergence of subsequent

RANS simulations. Additionally, the computational cost, particularly the physical memory consumption during this step, is not negligible.
Strategies for deforming the fluid mesh conforming to the deformation of solid body can be divided into two basic classes: physical analogy or interpolation. The most significant techniques under these two classes are reviewed in [27]. The physical analogy approach describes the fluid mesh deformation according to a physical process that can be modeled using numerical methods. In the interpolation based approaches, an interpolation function is used to transfer prescribed boundary point displacements to the fluid mesh. Alternative ways to approach this category of techniques rely on the concepts of spring analogy or the solutions of partial differential equations. However, one major disadvantage of these physical analogy methods is their reliance on large systems of equations, which significantly increases computational costs. Additionally, these methods necessitate grid connectivity information, leading to greater storage requirements and challenges in parallelization efforts. The spring analogy introduced by Batina [7] and linear elastic equations (ELA) first presented by Baker and Cavallo [6] belong to this category.
The implementation of this method aims to prevent nodes from crossing over element faces during the deformation process. While this approach enhances robustness, it becomes computationally impractical and is unable to handle large displacements. In this case, the grid is treated as an elastic continuum, and a built-in mechanism ensures that negative volume generation is avoided. We can refer to this technique as a Finite Element Method (FEM) model, where the goal is to calculate node displacements by considering the mesh elements as homogeneous and isotropic materials. The constitutive law for this approach can be expressed as:

$$\epsilon_{ij} = D_{ijks}\sigma_{ks} \tag{21}$$

Since the elasticity equations contain material properties, the modulus of elasticity ($E$) and Poisson's ratio ( $\nu$ ), these properties are related to the mesh characteristics. In this work, the "inverse volume" standard approach is utilized to determine the Young modulus. In this approach, a fixed value is assigned to the Poisson's ratio $\nu$, while the Young modulus $E$ is dependent on the cell. Specifically, $E$ is calculated as the reciprocal of the element volume $V_i$, expressed as $E = \frac{1}{V_i}$.
The radial basis function (RBF) interpolation method is a promising technique for interpolation. RBFs have gained recognition as a reliable method for interpolating scattered data. Additionally, RBFs can serve as interpolation functions to transfer known displacements from the structural mesh boundaries to the fluid mesh. This approach yields high-quality meshes while preserving reasonable orthogonality near deforming boundaries. Some advantages of RBF interpolation include: obviating the need for mesh connectivity information, working with a linear system of equations, and the size of the linear system being proportional to the number of boundary nodes rather than all fluid nodes. More detailed and in-depth information on the subject can be found in [1].

## 2.2. Sensitivity Computation

Automatic Differentiation, also referred to as Algorithmic Differentiation, is a valuable tool utilized in SU2 to automatically derive the adjoint equation from the original code. There are two approaches for implementing AD: source code transformation and operator overloading.
In source code transformation, the code is modified by introducing additional statements that calculate the tangent of the original code. This approach is the most efficient but requires significant software modifications. Since SU2 is written in C++, an object-oriented language, the implementation of AD heavily relies on Operator Overloading, which is the only viable option. Although it is easier to implement and extend, this method consumes more virtual memory.
The main concept is to overload every arithmetic operator and function, so that when they are called in the code, both the original operation and the computation of the tangent are simultaneously performed and stored. One advantage of AD is its ability to avoid truncation errors and produce derivatives with an approximation as small as the epsilon machine.
While AD shares similarities with symbolic differentiation, it applies the chain rule to perform calculations, which means it does not generate the full precise expression of the derivative. AD is often introduced based on the idea that complex mathematical code can be viewed as a series of basic functions with at most two independent variables. However, this approach can be inefficient in practice as it requires storing every single operation. A more effective approach is to employ AD at the statement-level, where the information that needs to be stored is independent of the number of operations involved internally.
In a generic function $f$, where $f$: $R^n \to R^m$, it can be represented as a sequence of $l$ statements, $\varphi_i : R^{n_i} \to R$. Each statement represents a local evaluation that can be arbitrarily complex. The table shown in Tab. 3.1 demonstrates how a complex expression is interpreted by the code and broken down into simple arithmetic operations. Here, $f$ takes $n$ inputs (x) and generates $m$ outputs (y). Furthermore, the relation between intermediate variables $v_i$ and $v_j$ is utilized, where the former depends directly on the latter, as represented by $u_i := (v_j)_{j<i}$ for $j$ belonging to the set of indices $i$.

| | | | |
|---|---|---|---|
| $v_i$ | $=$ | $x_i$ | $i = 1...n$ |
| $v_{i+n}$ | $=$ | $\varphi_i(u_i)$ | $i = 1...l$ |
| $y_i$ | $=$ | $v_{n+l-i+1}$ | $i = 1...m$ |

Table 1: Code Interpretation of a generic function

A more advanced representation of the function $f$ can be expressed as follows:

$$f(x) = Qm \cdot \Phi_l \cdot \Phi_{l-1} \cdot \ldots \cdot \Phi_2 \cdot \Phi_1 \cdot P_n^T(x) \tag{22}$$

Here, the function $\Phi_i : V \to V$, where $V := R^n \to R^l$, sets $v_{i+n}$ to $\varphi(v_j)_{j<i}$, while leaving the other $v_j$ unchanged. The final matrix $Q$ extracts the last $m$ elements from a vector of $(n + l)$ components.

### 2.2.1 Forward Mode

By employing the chain rule, it becomes possible to enhance each statement with its corresponding derivative, thereby enabling forward automatic differentiation (AD). The term "forward" originates from the fact that the values $v_i$ are carried forward alongside their derivatives $\dot{v_i}$.

| | | | |
|---|---|---|---|
| $v_{i-n}$ | $=$ | $x_i$ | $i = 1...n$ |
| $\dot{v}_{i-n}$ | $=$ | $\dot{x_i}$ | $i = 1...n$ |
| $v_i$ | $=$ | $\varphi_i(v_j)_{j<i}$ | $i = 1...l$ |
| $\dot{v}_i$ | $=$ | $\sum_{j<i} \frac{\partial \varphi_i(u_i)}{\partial \dot{v}_j}$ | $i = 1...l$ |
| $y_i$ | $=$ | $v_{l-1}$ | $i = m-1...0$ |
| $\dot{y}_i$ | $=$ | $\dot{v}_{l-1}$ | $i = m-1...0$ |

Table 2: Evaluation procedure using forward AD

The chain rule represents the key distinction from symbolic differentiation, wherein $\frac{\partial \varphi_i(u_i)}{\partial v_j}$ would be replaced by an algebraic expression. Combining these expressions would lead to a derivative expression of increasing complexity. Conversely, in forward AD, the memory usage and runtime are predetermined, although some error may be introduced due to floating point value roundoff. It should be noted that with the forward approach, an input variable is chosen, and the first-order derivative of the intermediate steps required to reach the output is computed with respect to that input.

### 2.2.2 Reverse Mode

Reverse mode is particularly useful for problems where a small number of objective functions or a single objective function depend on a large set of variables. A prime example is the Discrete Adjoint method, where we require the sensitivity of $J$ (objective function) with respect to a lengthy vector, $\alpha$. Unlike the forward mode, reverse mode focuses on a single output and calculates the first-order derivative with respect to both intermediate and input variables in a single unified process.
Reverse automatic differentiation (AD), implemented using expression templates, is approximately 2.7-4 times slower than a direct simulation, which puts it on par with hand-written Jacobians in terms of performance. However, it should be noted that reverse AD requires a significant amount of physical memory. Techniques such as local preaccumulation and the use of checkpoints can help mitigate the memory consumption. Initially, for each variable $v_i$ in the equation, we introduce a new variable $\bar{v}_i = \frac{\partial y}{\partial v_i}$, which is referred to as the "adjoint variable." Additionally, for the selected output, an extra variable $\bar{y} = 1$ is introduced. The goal now is to differentiate Equation 22 using the chain rule:

$$y = Q_m A_l A_{l-1} \ldots A_2 A_1 P_n^T \dot{x} \quad where \qquad A_i = \nabla \Phi_i \tag{23}$$

which is the tangent relation rewritten as an evaluation procedure. The Jacobian of the function can be immediately retrieved:

$$\frac{df(x)}{dx} = Q_m A_l A_{l-1} \dots A_2 A_1 P_n^T \tag{24}$$

By transposing the product we obtain the adjoint relation:

$$\bar{x} = P_n A_1^T A_2^T \dots A_{l-1}^T A_l^T Q_m^T \bar{y} = \left(\frac{df}{dx}\right)^T \bar{y} \tag{25}$$

and the identity:

$$\bar{y}\dot{y} = \bar{x}\dot{x}. \tag{26}$$

Upon closer examination of the intermediate matrix multiplication in Equation 25, it becomes apparent that the adjoint relation can be expressed as an evaluation process. This involves computing all matrix-vector products for $i = l, l-1, \dots, 1$, thereby traversing the sequence of elementary functions in reverse order, as presented in Table 2. Since the intermediate values $v_i$ are required, they must be computed beforehand. Additionally, the sequence of the concatenation has to be registered, then it is inverted in the return sweep in order to have the tangent values. The reverse sweep is shown in Table 3.

| | | | |
|---|---|---|---|
| $v_{l-i}$ | $=$ | $y_{m-i}$ | $i = 1...m-1$ |
| $\bar{v}_j$ | $=$ | $\bar{v}_j + \bar{v}_i \frac{\partial \varphi_i(u_i)_{j<i}}{\partial v_j}$ | $i = l...1$ |
| $\bar{x}_i$ | $=$ | $\bar{v}_{l-n}$ | $i = n...1$ |

Table 3: Code Interpretation of a generic function

### 2.2.3 Expression Template

Implementing reverse mode automatic differentiation using expression templates can greatly enhance its attractiveness and computational efficiency. This technique, employed in C++, significantly reduces the virtual memory requirements. Expression templates were initially introduced to expedite the evaluation of mathematical expressions based on operations or arrays. They are now incorporated into various C++ matrix libraries, such as Adept and CoDiPack. The key idea is to modify each operator so that it does not return a value directly; instead, it generates a new object that describes the type of operation to be executed and records the expected input type. This object is known as an "Expression," and all variable types, arithmetic operators, and functions are derived from a common base class. This coding concept relies on static polymorphism, where an object of one type is concealed by another, with the compiler only becoming aware of this during runtime. By avoiding the use of virtual functions, which would significantly slow down the program, the performance is greatly improved.

## 2.3. Cavitation Objective Function

Cavitation is the physical phenomenon according to which the pressure of water decreases until it equals or exceeds the vapor pressure, computed under given conditions of pressure and temperature. Cavitation leads to the formation of air bubbles on the surface of the profile, which strongly impacts the aerodynamic performance of the profile. They can also affect the properties of the material from which the surface is made. The work reported in this thesis is the implementation of a part of SU2 code that checks for cavitation on a profile and optimizes the shape of the profile with the goal of nullifying this damaging phenomenon. The implemented code works by calculating for each grid element the difference between pressure and vapor pressure, in dimensionless terms. In case the following inequality occurs:

$$-C_p - \sigma > 0 \tag{27}$$

the code computes the cavitation coefficient:

$$Coeff_{Cav} = \frac{\sum_{i=1}^{n} A_i | - C_{pi} - \sigma |}{A_{ref}} \tag{28}$$

where $\sigma$ is the cavitation number in the simulating condition, $C_{pi}$ is the pressure value in the i-th element, $A_i$ is the area of the i-th element, $A_{ref}$ is the reference area chosen to dimensionalize. Then the cavitation coefficient

is given by the summation of the product of the pressure difference by the area of the element, for each element in which cavitation occurs (i.e., the inequality given above is true). This is made dimensionless by dividing by a reference area chosen by the user and given in the configuration file. In the case of 2D analysis, the quantity used for adimensionalization is not an area, but rather a length. It is also possible to derive the total cavitation area by summing the areas of the elements in which the phenomenon is evidenced:

$$A_{Cav} = \sum_{i=1}^{n} A_i \tag{29}$$

The cavitation number, $\sigma$, is defined by the user within the configuration file. When a simulation is launched, the code is able to print the cavitation coefficient values in the history file. In the case of shape optimization, the cavitation coefficient itself can be defined as an objective function. Infact it is possible to choose as objective function either the cavitation coefficient either the cavitation area, to be reduced.
The cavitation number is obtained by the following formula:

$$\sigma = \frac{P_{ref} - P_{vap}}{\frac{1}{2}\rho_{ref} \cdot v_{ref}^2} \tag{30}$$

substituting the values of the quantities characteristic of the case being studied. In the equation above, $P_{ref}$ represents the reference pressure at which the simulation is conducted, $\rho_{ref}$ denotes the reference density, and $v_{ref}$ corresponds to the reference velocity. The values employed for these quantities in the simulations of this project are listed in the 2D results section (4.1).
The cavitation number provides a measure of the relative difference between the local pressure and the vapor pressure of the fluid. When the cavitation number is close to or exceeds unity ($\sigma \geq 1$), cavitation is likely to occur. Conversely, when the cavitation number is significantly below unity ($\sigma \ll 1$), cavitation effects are minimal.
The typical values of the cavitation number for profiles can vary depending on the specific application and operating conditions. In general, for hydrofoils, propellers, and other underwater profiles, typical cavitation numbers range from 0.2 to 2.0. These profiles experience varying degrees of cavitation depending on factors such as the flow velocity, pressure distribution, and geometry.
For aircraft wings, which operate in air rather than water, the cavitation numbers are typically much lower. The cavitation number for airfoils can be in the range of $10^{-4}$ to $10^{-2}$ or even lower. This is due to the higher vapor pressure of air compared to water, making cavitation effects less prevalent in aerodynamic applications.

# 3. Validation and verification

In this study, a flow solution validation process for two-dimensional simulations has been conducted using the NACA 0015 airfoil profile. The selection of this particular profile was motivated by the opportunity to compare the results with those presented in [29]. The article provides both experimental and computational data, allowing for a comprehensive evaluation.
On the symmetrical NACA 0015 profile, the angle of attack of 8 degrees has been investigated. Upon analysis, it can be observed that the error at an 8 degree angle of attack was relatively small (Fig. 3) and therefore, it was decided to continue the simulations and optimizations at this angle. The boundary conditions used for the simulations align with those specified in the reference table provided in the results section (Tab. 5). In fact, the graph shows that the three lines, which indicate the variation of the pressure coefficient on the NACA 0015 airfoil, are almost coincident. The three lines directly compare the data obtained in the paper [29], the experimental data, and the results obtained in this study. By adhering to these established boundary conditions, consistency and comparability with the referenced article were ensured.

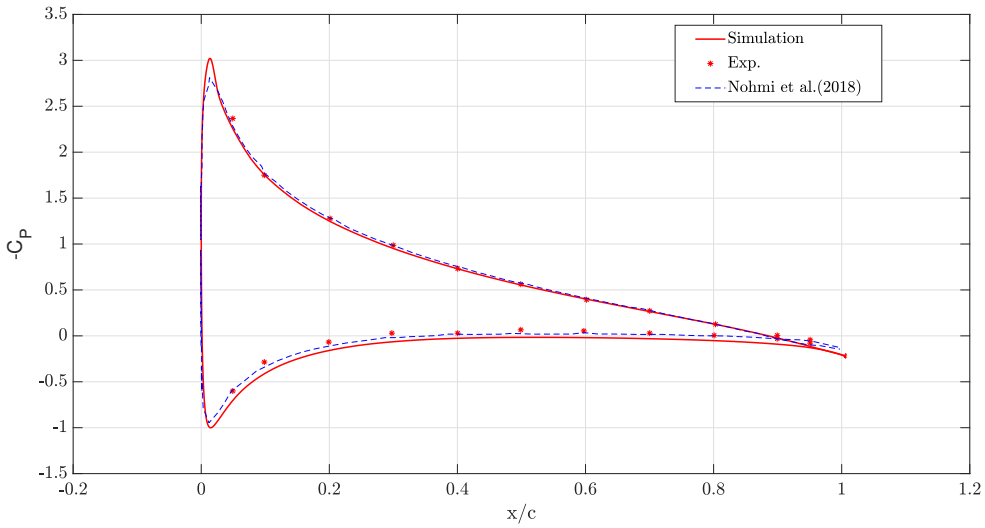



Figure 3: AoA 8

For the direct simulations aimed at code validation, the incompressible RANS solver was employed. The one-equation model of Spalart-Allmaras [41] was adopted to simulate the viscous nature of water. Several numerical methods, including JST and FDS, were initially tested, but eventually, the choice converged on the JST method for all simulations. This selection was based on its superior performance and suitability for our purposes. Several different $2^{nd}$ and $4^{th}$ order artificial dissipation coefficients have been tested. In the end, the choice came down to the respective values of 0.5 and 0.02.
To simplify the simulation setup, an O-grid shape was chosen for conducting the simulations around the NACA 0015 airfoil. Pointwise ®(http://www.pointwise.com/) software was used to build the mesh. The cells close to the airfoil are structured rectangles, the total height of this region has to contain the entire boundary layer. Meanwhile, the rest of the grid is unstructured. The farfield outline was divided by 190 points, with a ray extending up to 30 chords. The boundary layer region was discretized using the T-Rex mode available in Pointwise. The selection of the first height cell aimed to achieve a $y+$ value of approximately 1; the boundary conditions used for computing the Reynolds number are detailed in section 4. To ensure a $y+$ value close to unity, the height of the first cell, $\Delta s$, of the structured grid starting from the foil contour was calculated. $\Delta s$ should be equal to $3 \times 10^{-6} m$.
The mesh independence analysis focused specifically on the nearfield region surrounding the airfoil. Several refinements were explored by adjusting the number of points along the airfoil profile, the growth rate of the mesh, and the first cell height. A summary of the investigated cases is provided in the table below.

| **N** | $N_{profile}$ | **growth rate** | **f.c.height** |
|---|---|---|---|
| 20395 | 100 | 1.25 | $1.5 \cdot \Delta s$ |
| 32785 | 200 | 1.2 | $\Delta s$ |
| 39223 | 200 | 1.15 | $0.5 \cdot \Delta s$ |
| 45759 | 250 | 1.15 | $0.5 \cdot \Delta s$ |
| 52239 | 300 | 1.15 | $0.5 \cdot \Delta s$ |

Table 4: NACA 0015 mesh indipendence

Convergence of the solutions was assessed by considering the lift coefficient($C_l$). Figure 4 presents the relationship between $C_l$ and the number of grid points. The graph clearly indicates a significant reduction in error when the number of points exceeds or equals $3.9 \times 10^4$. Therefore, it was possible to use the first grid among those with the same $C_l$, which corresponds to 39223 elements.

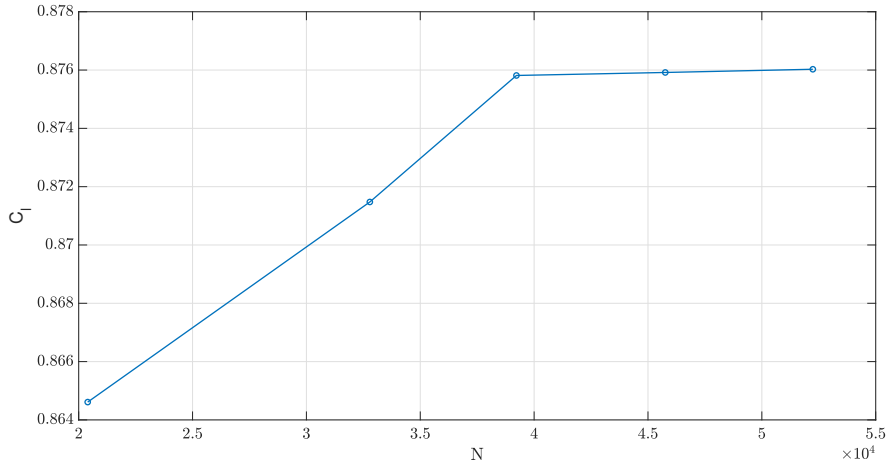



Figure 4: Grid convergence

The research focuses on finding the optimal shape using gradient-based algorithms. To achieve this, it is necessary to determine the sensitivity of the objective function with respect to the design variables. Generally, the objective function $J$ can be comprised of $n_f$ individual functions denoted as $J_k$, each of which must be defined and differentiated with respect to the design parameters. The relationship between the objective functions, and the mesh can be clarified by the following expression:

$$J_k(\alpha) = J_k(U(\alpha), X(\alpha)) \quad \text{for } k \in \{1, \ldots, n_f\} \tag{31}$$

Initially, the objective was to calculate the partial derivatives $\frac{\partial J_k(\alpha)}{\partial \alpha_i}$, which would involve $n_f \times n_\alpha$ derivatives. However, more recent codes calculate the total derivative $\frac{dJ_k}{dX}$ for k $\in \{1...n_f\}$. These codes incorporate both the direct influence of node positions and the indirect influence caused by changes in the flow field to achieve steady-state convergence. The Finite Difference method is a straightforward and traditional approach for obtaining the gradient of the objective function. This method does not require any modifications to the solver. To calculate the object function gradient, the discrete flow solution needs to be computed for the given foil described by the design variables $\alpha$, as well as for perturbed values $(\alpha + \delta\alpha)$ and $(\alpha - \delta\alpha)$. Typically, a second-order finite difference scheme is employed.
If the geometry of the body is defined using an FFD box, as in this work, a perturbation in the direction $m$ of a quantity $\delta\alpha$ is represented by shifting a designated vertex, which serves as a design variable. This leads to two mesh deformations, $X(\alpha + \delta\alpha_m)$ and $X(\alpha - \delta\alpha_m)$, that need to be performed, and two flow solutions are computed on the morphed grids, satisfying the following conditions:

$$R(U(\alpha - \delta\alpha), X(\alpha - \delta\alpha)) = 0 \quad R(U(\alpha + \delta\alpha), X(\alpha + \delta\alpha)) = 0 \tag{32}$$

However, this method becomes impractical when dealing with a large number of design variables because computing the matrix $\frac{dJ_\alpha}{d\alpha}$ incurs a cost equivalent to $2 \times n_\alpha$ times the cost of a single flow solution. The verification of the adjoint solver is performed by comparing the obtained gradients with a second-order centered finite difference method:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \tag{33}$$

where h represents the step size. One critical aspect to consider is how to determine the appropriate step size. The choice of step size significantly impacts the accuracy of the computed gradient. If the step size is too small, rounding errors become significant. Conversely, if it is too large, the truncation of higher-order terms in the Taylor expansion is no longer valid.
In the current study the verification is carried out considering the NACA 0015. The surface is parameterized using a two-dimensional FFD box, with 14 design variables. The box containing the profile and the DVs are riported in Figure 5, where it is also shown the direction of the possible displacements. The shape can be modified changing the thickness and camber of the airfoil. Before reaching this size of the box, other dimensions were tested, starting from larger sizes and gradually reducing the box, so that the edges were much closer to the profile. A reasonable value for the step size for finite differences was found to be $10^{-5}$. Very good agreement is found between the sensitivities calculated with finite differences and with the discrete adjoint method. A direct comparison between the two methods is observable in Figure 6, where the cavitation sensitivity is plotted with respect to the design variables.

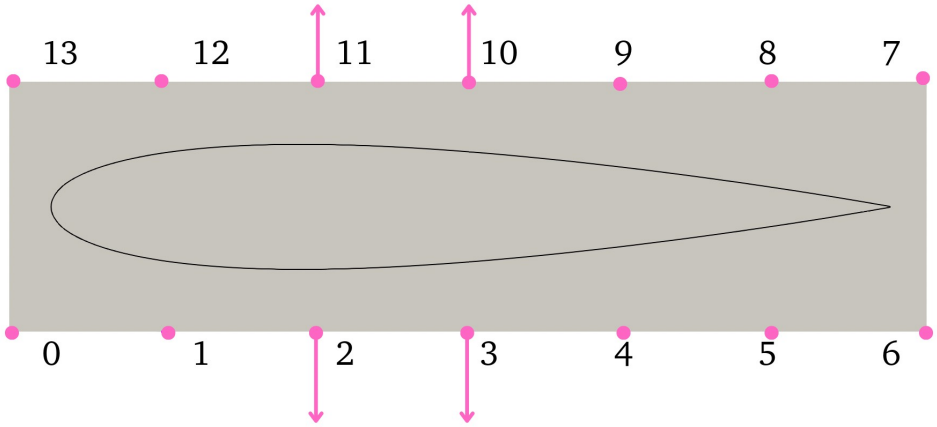



Figure 5: 2D box and design variables

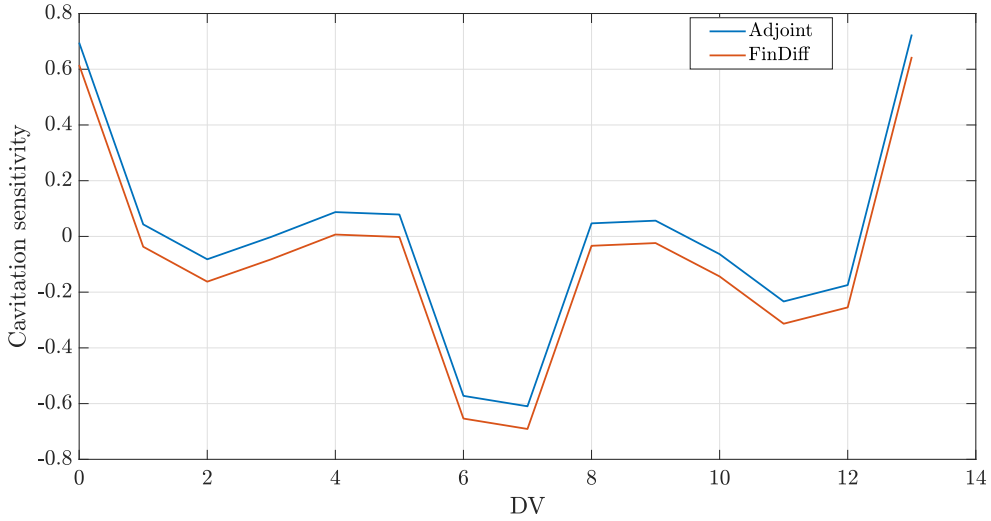



Figure 6: Cavitation sensitivity computed with the adjoint method and with the finite difference

The graph below (Figure 7) illustrates the trends of cavitation and drag sensitivities, normalized to the same magnitude. From this comparison, it is evident that the decrease in cavitation does not align with the decrease in drag.

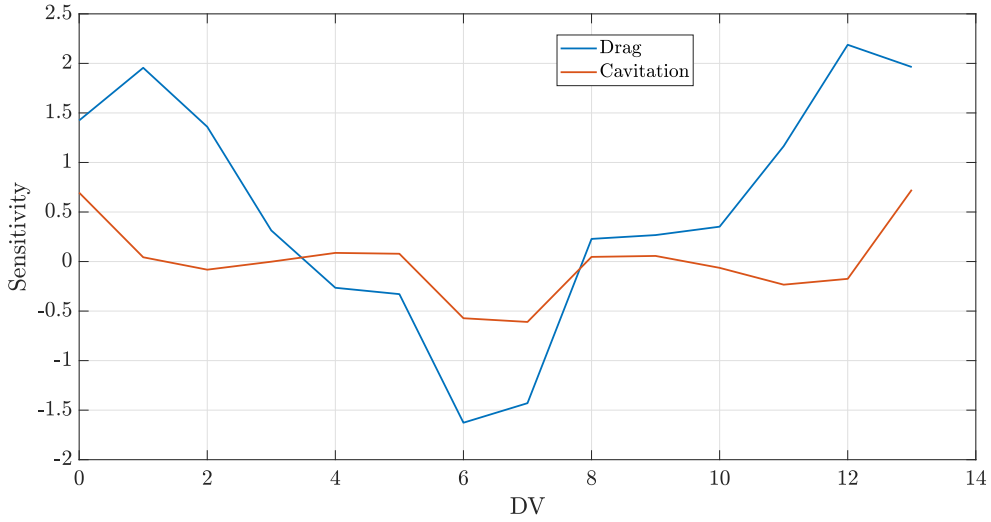



Figure 7: Sensitivity of drag and cavitation

# 4. Results

This section presents the results of the cavitation optimization, which is divided into two parts: the first part focuses on the 2D case using the NACA 0015 airfoil, while the second part discusses the 3D Moth foil.

## 4.1. NACA 0015 Optimization

Starting with the two-dimensional profile, the chosen free stream conditions are provided in Table 5. The target of the optimization is to reduce the possible area of cavitation. The optimizations are carried out with a fixed value of $C_l$, which was determined through prior direct simulations. The angle of attack is let free to vary in order to more easily respect the constraint about the lift. The free stream velocity was chosen to be equal to that in the paper [29], as mentioned above. The density and dynamic viscosity are typical values for water. The Reynolds number was calculated considering the chord length of the profile, which is equal to 1 m, as the reference length.

| | |
|---|---|
| **Velocity** | $8\ m/s$ |
| **Temperature** | 20°C |
| **Density** | $998.2\ kg/m^3$ |
| **Viscosity** | $8.9 \times 10^{-4} kg/(m \cdot s)$ |
| **Reynolds** | $8.97 \times 10^6$ |
| $C_l$ **constraint** | 0.876 |

Table 5: Free stream conditions

Given the boundary conditions mentioned above, the cavitation number $\sigma$ is found to be 1.937. This value is obtained using Equation 30, knowing that the reference pressure is computed as $P_{ref} = \rho_{ref} v_{ref}^2 = 63884.8Pa$. Considering the water temperature to be 20°C, the vapor pressure value $P_{vap} = 2000Pa$ is extracted from the tabulated data. By comparing the value of $\sigma$ with the pressure coefficient plot in Figure 3, it confirms the presence of cavitation on the upper surface of the profile, specifically in the leading edge region. Here, the magnitude of the pressure coefficient is greater than the cavitation number. The same result is visible in Figure 11, which depicts the cavitation area $A_{Cav}$ before optimization. The cavitation area includes elements on the profile that satisfy the inequality stated in Equation 27.
The body is parameterized with a single FFD box and has 14 design variables that can be moved in the y direction (Figure 5). The mesh deformation in this two dimensional study follows the linear elastic equation ELA. The history of optimization is reported in Figure 8. It can be noticed that, after 9 design loops, cavitation

decreases to zero. Additionally, the trend of the cavitation area is also shown in the same graph, demonstrating a clear decrease. The cavitation area is expressed as a percentage relative to a reference area $A_{ref}$. The chosen reference area has a value of unity.

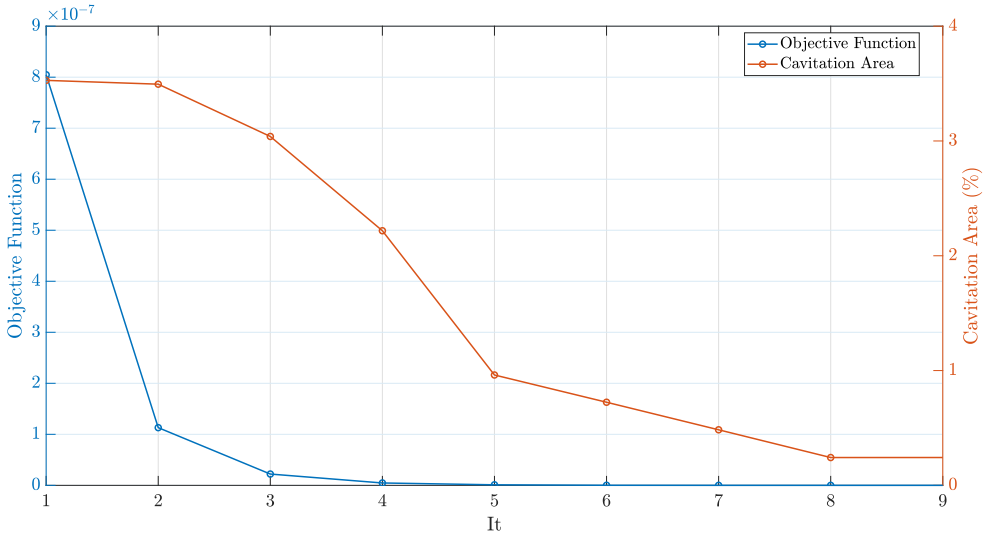



Figure 8: Objective function and cavitational area wrt design loops

Figure 9 depicts the evolution of the airfoil shape after 9 design loops of optimization. The obtained profile is asymmetric, with a downward-leading edge and an upward-trailing edge. The modification of the leading edge is a characteristic feature of profiles aimed at increasing the pressure ahead of the flow. The increase in curvature on the leading edge generates a lower peak in the pressure coefficient. The last part of the profile is deflected upright, to compensate for the change in the leading edge, as well as to satisfy the constraint on the lift. Graph 10 compares the pressure coefficient distribution of the NACA 0015 airfoil with that of the optimized airfoil. It highlights that the magnitude of the pressure coefficient decreases in the leading-edge region, eventually becoming lower than the cavitation number. This result is further confirmed by the comparison of Figures 12 and 13. The first image represents the NACA 0015 profile, while the second image represents the optimized profile. We can observe that the peak value of the coefficient of pressure ($C_p$) has decreased, and simultaneously, the area of minimum $C_p$ along the profile's upper surface has increased. In the given operating conditions, the previously existing cavitation area on the NACA 0015 profile has now been completely eliminated through shape modification. This remarkable achievement signifies that the air bubble, which used to occupy the dark blue-colored region along the profile, has been effectively flattened due to the optimized profile shape.

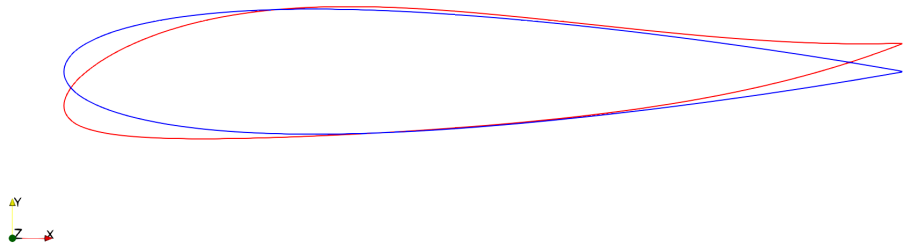


Figure 9: Comparison between NACA0015 (blu) and the optimized profile (red)

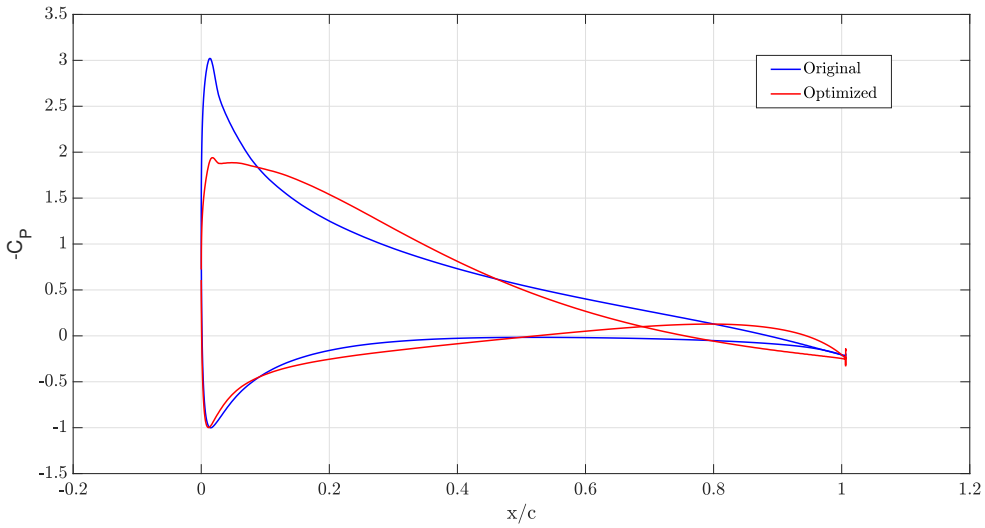



Figure 10: Pressure coefficient of NACA0015 and the optimized profile

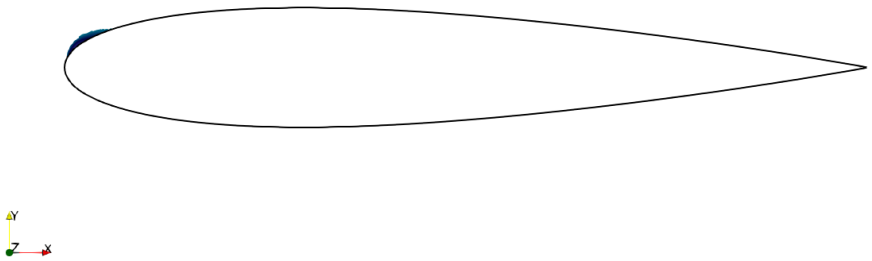


Figure 11: Cavitation area on NACA0015

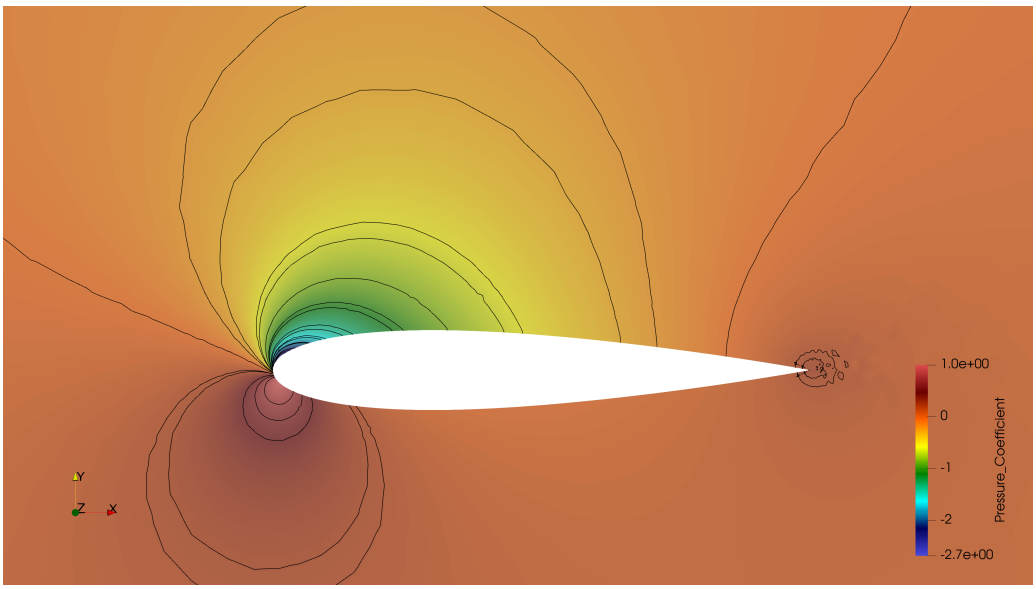



Figure 12: Pressure contours on the NACA 0015

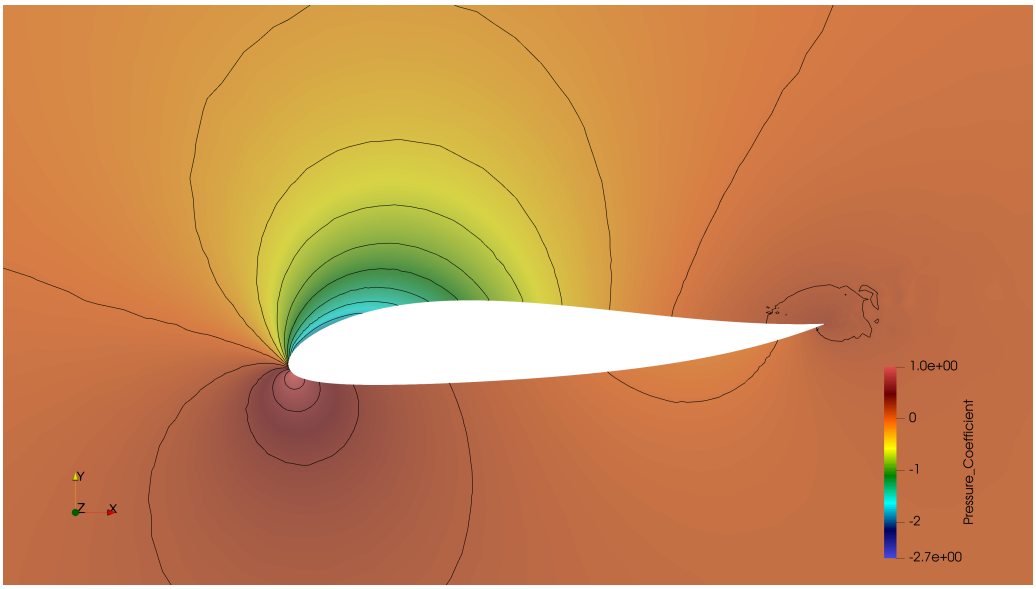



Figure 13: Pressure contours on the optimized profile

It can be noticed that the reduction in cavitation area generates an increase of drag. This can be expected since the two sensitivities are different and have different local minima. Actually, we are not able to simulate cavitation, so if we had a cavitation model, the actual drag would likely be higher. Therefore, reducing cavitation could also mean reducing drag. It cannot be concluded that reducing cavitation automatically leads to an increase in drag, but it may depend on the specific case and further studies are needed. For example, in the next section, it will be shown that the drag has not increased.
Leaving the angle of attack unconstrained aids in the optimization process, but there is a significant modification of 5 in the angle of attack, which is a substantial change.

| | **1 loop** | **9 loop** |
|---|---|---|
| **Cavitation Area** | 3.52788% | 0.24247% |
| **AoA** | 7.25413° | 12.3547° |
| $C_d$ | 0.011729 | 0.012295 |

Table 6: 2D results

The elimination of the cavitation area is of significant importance as it indicates a successful mitigation of the detrimental effects caused by cavitation. The modified profile shape reduces the occurrence of low-pressure regions that could trigger cavitation.

## 4.2. Hydrofoil Optimization

Regarding the 3D case, the main foil of the Moth sailboat designed by the PoliMi Sailing Team was optimized. The target of optimization is the cavitation area, keeping constant the lift of the baseline geometry. The main foil is equipped with a central bulb, which serves as the junction point with the vertical surface. In the analyzed case, only half of the foil was considered, taking advantage of its symmetry to reduce computational time.
The foil was encapsulated within a free-form deformation box, as shown below in Figure 14. The vertices highlighted in black represent the control points with freedom of displacement in the z-direction. The DVs are 126, 7 in the chord direction, 9 in the span direction. Since the design variables are only allowed to move along the z-axis, neither the chord length nor the span length can change. The bulb is not part of the optimization. First-order continuity properties are applied at the intersection.
Similar to the 2D case, the boundary conditions used for this optimization were applied, as the Moth sailboat is assumed to operate at a speed of approximately 15 knots in lake waters during the summer. As a result, the cavitation number remains unchanged at 1.937. Only the fixed value of $C_l$ is different, in this case equal to 1.41.
The grid surrounding the foil was created using Pointwise and is an unstructured grid in the shape of a hemisphere. The region around the wing was structured using the T-Rex mode in 3D.
RANS equations are solved with SA turbulence model, JST is used to calculate the fluxes. $2^{nd}$ and $4^{th}$ order artificial dissipation coefficients are 0.5 and 0.02. It is required to have at least six order reduction of the relevant residuals and the finals lower than $10^{(-13)}$.
As mesh deformation method in this optimization test case RBF has been used. RBF has been settlend using Wendland C2 as baseline functions are selected, using a number of control points equal to 15% of total number of surface elements.

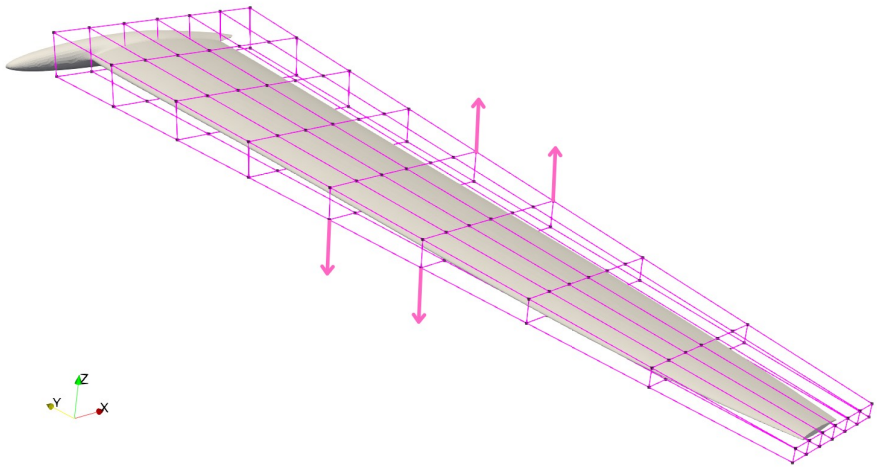



Figure 14: 3D box

In Figure 16, the surface area affected by cavitation is highlighted. It was visualized by examining the pressure coefficient, using a threshold value of $\sigma = 1.937$. The reference area used to compute the $A_{Cav}$ corresponds to 0.0287 $m^2$.
Due to limited computational resources, only 5 design loops were completed. The optimality conditions are not fullfilled, but we are stuck in a local minimum. We can consider the obtained shape as optimized since the cavitation coefficient is reduced of the 3%. As can be seen from Table 7, a decrease in cavitation area suggests that the modifications made to the profile have effectively mitigated the detrimental effects of cavitation.
In this three-dimensional case, an improvement in fluid dynamics performance was also achieved, as the drag coefficient decreased from the first to the fifth design loop. The decrease in the drag coefficient observed during

the cavitation optimization process is a positive outcome, despite not being the objective of the optimization. In this case, it can be observed that the angle of attack has changed much less compared to the 2D case.

| | **1 loop** | **5 loop** |
|---|---|---|
| **Cavitation Area** | 7.11743% | 4.58159% |
| **AoA** | 7.91938° | 9.76929° |
| $C_d$ | 0.052297 | 0.051707 |

Table 7: Design loops

Figure 15 compares the original foil with the optimized foil, while Figures 16 and 17 display their respective cavitation areas. The original foil is colored yellow, and its cavitation area is depicted in blue, extending throughout the foil span, particularly at the leading edge. In contrast, the optimized foil is colored aqua green, and its cavitation area is a small red area observed near the junction with the bulb.
For further clarity, the pressure coefficient distributions are shown from a top view (Figures 18, 19). It is noticeable that in the optimized foil the region with the lowest $C_p$ is narrower compared to the original case and close to the interface with the bulb. However, this is nearby the area that is not optimized. This suggests the possibility of optimizing the junction region between the bulb and the foil in the future.
The residual presence of cavitation could be due to the premature termination of the optimization; otherwise, it would have decreased further.

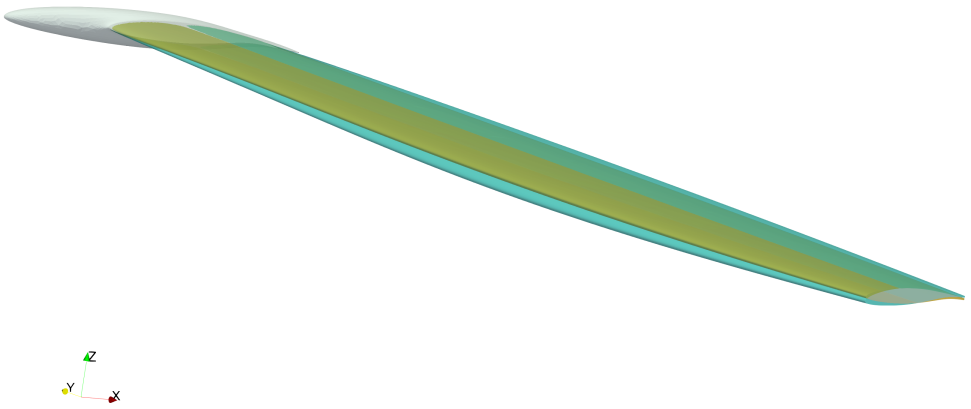



Figure 15: The original foil (yellow) and the optimized foil (aqua green)

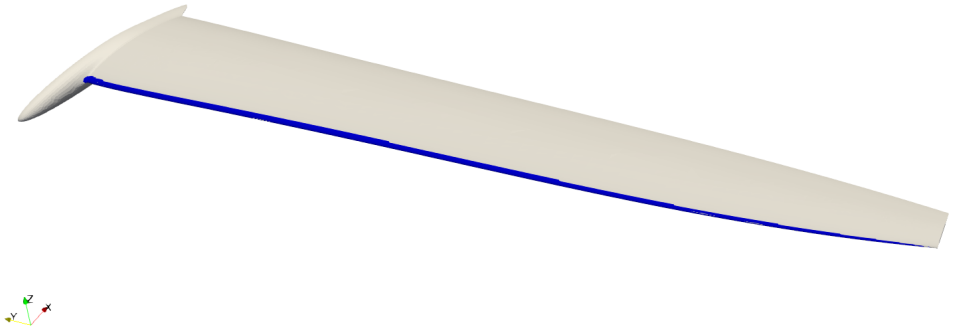


Figure 16: Cavitation area on the baseline foil

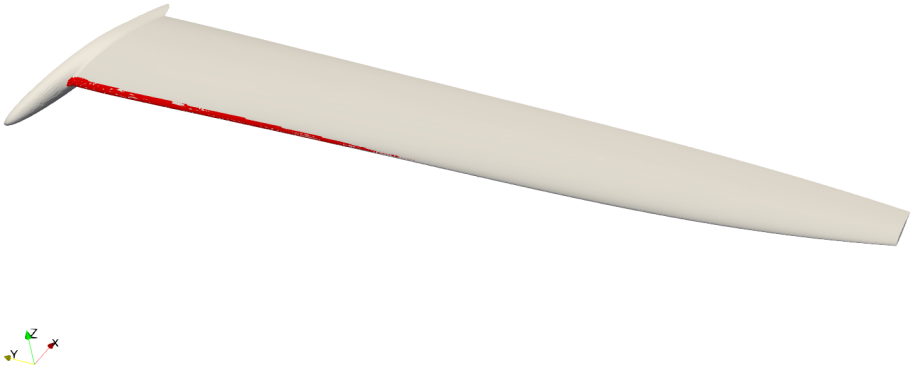


Figure 17: Cavitational area on the optimized foil

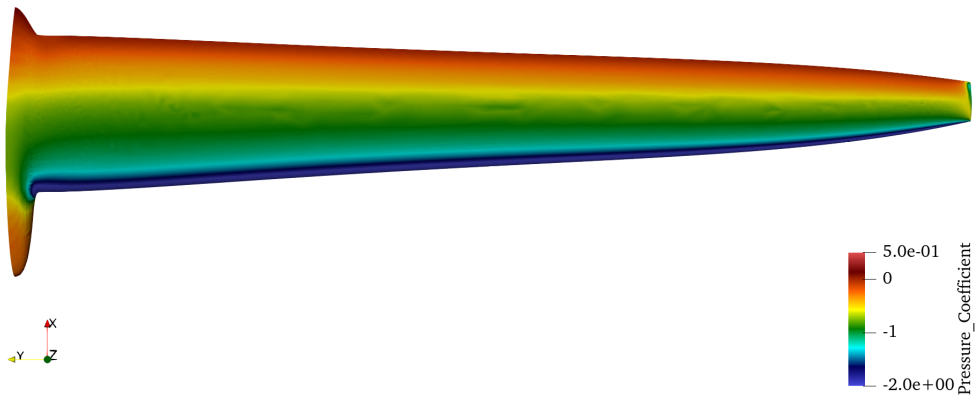



Figure 18: Pressure coefficient of the original foil

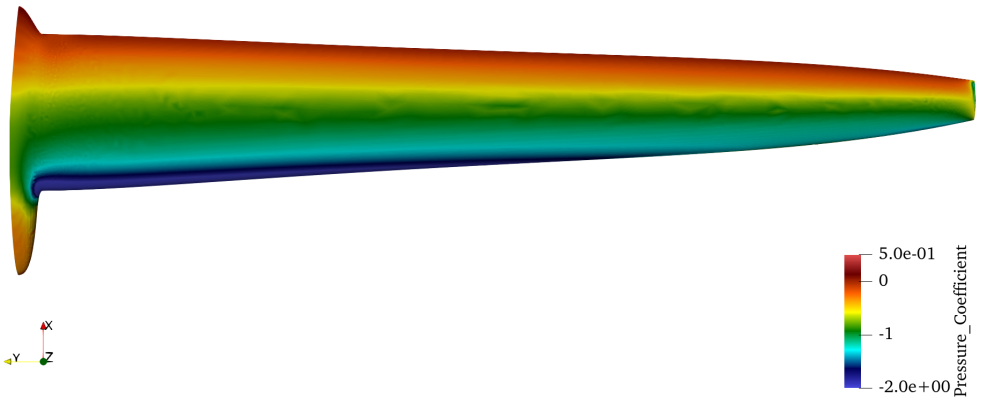



Figure 19: Pressure coefficient of the optimized foil

Moreover, the wing has been discretized into sections, with each section corresponding to a specific location along the span of the foil. The 2D contours of these sections have been provided, allowing for a convenient visual assessment of the shape variations throughout the wing span. The sections were selected at four key positions: 25%, 50%, 75%, and 98% of the wing span, excluding the bulb region. Among these sections, the most notable change in shape is observed at the 75% span location. In this area, the wing exhibits significant morphing, with noticeable alterations in its contour. On the other hand, the sections closer to the root of the wing display relatively minor morphing, while those towards the wingtip primarily exhibit increased twist rather than substantial shape changes. This section-wise analysis provides valuable insights into the morphing characteristics of the wing, highlighting the localized variations in shape and twist along its span.

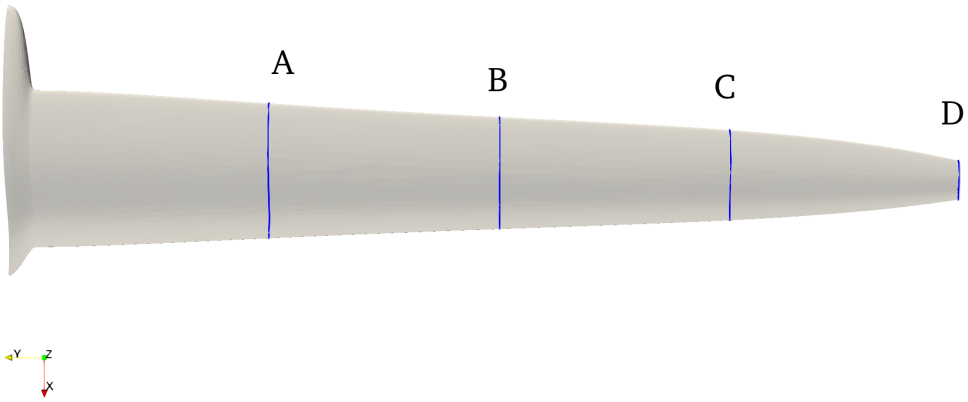
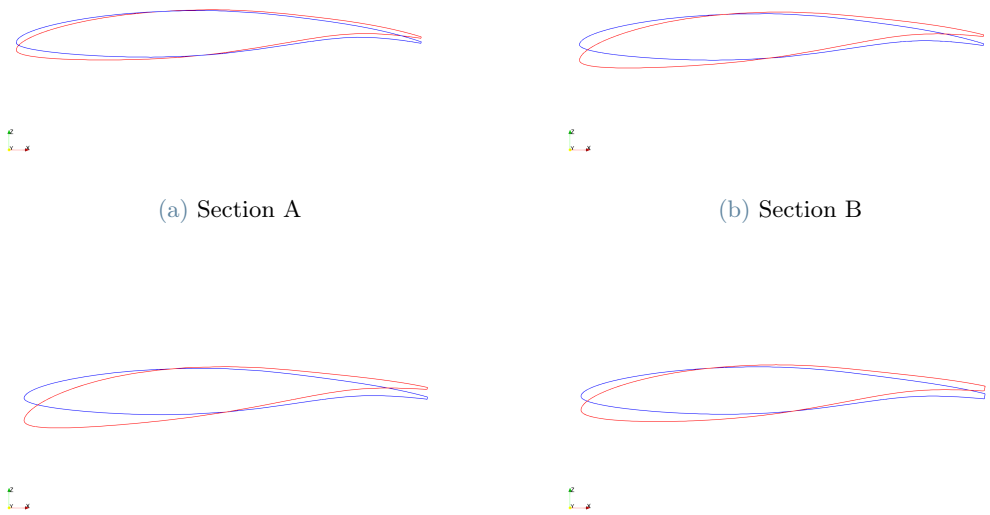



Figure 20: Sections



(a) Section A

(b) Section B

(c) Section C

(d) Section D

Figure 21: Sections of the original foil (blue) and optimized foil (red)

## 5. Conclusions

This thesis is focused on the application of adjoint shape optimization techniques for reducing cavitation on hydrofoils. Indeed, usually in this field, the inverse design method had been predominantly used to search for the optimal shape that avoids cavitation. The use of adjoint shape optimization offers the advantage of not requiring prior knowledge of the pressure distribution, unlike inverse design. The objective of this research project was to optimize the profile shape of a 2D and a 3D foil, with the goal of minimizing cavitation. To achieve this, the optimization framework implemented in the open-source finite volume solver SU2 was exploited. In order to address this objective, a new objective function was developed to estimate the potential cavitation area.
A successful result has been achieved in both the two-dimensional and three-dimensional studies. By utilizing sensitivity information obtained from the adjoint solver, it was possible to systematically modify the hydrofoil's shape to mitigate cavitation-related issues. In the 2D case, a profile with virtually no cavitation area was achieved, accompanied by a significant change in the angle of attack. In the 3D case, after only 5 design iterations, a reduction in the cavitation area was observed, resulting in a decrease in the drag coefficient as a side effect. In this case, the variation in the angle of attack was only 2°. The 3D hydrofoil studied in this case is the main foil dseigned and produced for a Moth by the PoliMi Sailing Team. The successful elimination of the cavitation area demonstrates the effectiveness and potential of the adjoint shape optimization method in addressing cavitation-related challenges.
It should be noted that the optimal solution highly depends on the specific design requirements and operating conditions. The choice of design variables, objective functions, and constraints must be carefully tailored to the hydrofoil's intended application and operational environment. Future studies can explore additional design variables and multi-objective optimizations to achieve a more comprehensive and robust hydrofoil design. Overall, the successful implementation of adjoint shape optimization for cavitation reduction presents a promising avenue for further research and potential real-world applications. In the future, for instance, the optimization could be repeated while keeping the angle of attack fixed. It is hoped that the outcomes of this study will contribute to the advancement of hydrofoil design and optimization techniques, ultimately benefiting the efficiency, sustainability, and performance of hydrofoil-based systems in various industries.

## References

[1] L. Abergo, M. Morelli, and A. Guardone. Aerodynamic shape optimization based on discrete adjoint and RBF. *Journal of Computational Physics*, 477:111951, mar 2023. doi: 10.251.

[2] D. T. Akcabay, E. J. Chae, Y. L. Young, A. Ducoin, and J. A. Astolfi. Cavity induced vibration of flexible hydrofoils. *Journal of Fluids and Structures*, 49:463–484, aug 2014. doi: 10.1016/j.jfluidstructs.2014.05.007.

[3] T. A. Albring, M. Sagebaum, and N. R. Gauger. *Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2*. doi: 10.2514/6.2016-3518.

[4] T. A. Albring, M. Sagebaum, and N. R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. jun 2015. doi: 10.2514/6.2015-3240.

[5] D. Anevlavi and K. Belibassakis. An adjoint optimization prediction method for partially cavitating hydrofoils. *Journal of Marine Science and Engineering*, 9(9):976, sep 2021. doi: 10.3390/jmse9090976.

[6] T. Baker and P. Cavallo. Dynamic adaptation for deforming tetrahedral meshes. aug 1999. doi: 10.2514/6.1999-3253.

[7] J. T. Batina. Unsteady euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8): 1381–1388, aug 1990. doi: 10.2514/3.25229.

[8] O. Chernukhin and D. W. Zingg. Multimodality and global optimization in aerodynamic design. *AIAA Journal*, 51(6):1342–1354, jun 2013. doi: 10.2514/1.j051835.

[9] O. Coutier-Delgosha, F. Deniset, J. A. Astolfi, and J.-B. Leroux. Numerical prediction of cavitating flow on a two-dimensional symmetrical hydrofoil and comparison to experiments. *Journal of Fluids Engineering*, 129(3):279–292, aug 2006. doi: 10.1115/1.2427079.

[10] H. Ding, F. C. Visser, Y. Jiang, and M. Furmanczyk. Demonstration and validation of a 3d CFD simulation tool predicting pump performance and cavitation for industrial applications. *Journal of Fluids Engineering*, 133(1), jan 2011. doi: 10.1115/1.4003196.

[11] R. Djeddi and K. Ekici. Discrete adjoint recursive technique for aerodynamic design optimization. jul 2021. doi: 10.2514/6.2021-3035.

[12] A. Ducoin, F. Deniset, J. A. Astolfi, and J.-F. Sigrist. Numerical and experimental investigation of hydrodynamic characteristics of deformable hydrofoils. *Journal of Ship Research*, 53(04):214–226, dec 2009. doi: 10.5957/jsr.2009.53.4.214.

[13] N. Garg, G. Kenway, P. Lyu, J. Martins, and Y. L. Young. High-fidelity hydrodynamic shape optimization of a 3-d hydrofoil. *Journal of Ship Research*, 59:209–226, 12 2015. doi: 10.5957/JOSR.59.4.150046.

[14] P. R. Gogate and A. M. Kabadi. A review of applications of cavitation in biochemical engineering/biotechnology. *Biochemical Engineering Journal*, 44(1):60–72, apr 2009. doi: 10.1016/j.bej.2008.10.006.

[15] P. He, G. Filip, J. R. Martins, and K. J. Maki. Design optimization for self-propulsion of a bulk carrier hull using a discrete adjoint method. *Computers &amp Fluids*, 192:104259, oct 2019. doi: 10.1016/j.compfluid.2019.104259.

[16] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, sep 1988. doi: 10.1007/bf01061285.

[17] T. Johansen, T. Fossen, and S. Berge. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Transactions on Control Systems Technology*, 12(1):211–216, jan 2004. doi: 10.1109/tcst.2003.821952.

[18] V. M. Korivi and A. C. Taylor. An incremental strategy for calculating consistent discrete cfd sensitivity derivatives. *National Aeronautics and Space Administration, Langley Research Center*, 1992.

[19] H. U. Koyuncuoglu and P. He. Simultaneous wing shape and actuator parameter optimization using the adjoint method. *Aerospace Science and Technology*, 130:107876, nov 2022. doi: 10.1016/j.ast.2022.107876.

[20] J. Kröger, N. Kühl, and T. Rung. Adjoint volume-of-fluid approaches for the hydrodynamic optimisation of ships. *Ship Technology Research*, 65:47–68, 01 2018. doi: 10.1080/09377255.2017.1411001.

[21] H. Kuhn and A. Tucker. Non linear programming in: Proceedings of second berkeley symposium on mathematical statistics and probability. *University of California Press, Berkeley, Calif.*, pages 481–492, 1951.

[22] J.-B. Leroux, O. Coutier-Delgosha, and J. A. Astolfi. A joint experimental and numerical study of mechanisms associated to instability of partial cavitation on two-dimensional hydrofoil. *Physics of Fluids*, 17(5): 052101, may 2005. doi: 10.1063/1.1865692.

[23] Y. Liao, S. He, J. R. R. A. Martins, and Y. L. Young. Hydrostructural optimization of generic composite hydrofoils. jan 2020. doi: 10.2514/6.2020-0164.

[24] J. N. Lyness and C. B. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, jun 1967. doi: 10.1137/0704019.

[25] F. Menter. Zonal two equation kw turbulence models for aerodynamic flows. *23rd fluid dynamics,plasmadynamics, and lasers conference*, 1993.

[26] J. H. Michell. The wave resistance of a ship. *Phil.Mag*, 5(45):106–123, 1898.

[27] S. MM and K. RP. Mesh deformation approaches – a survey. *Journal of Physical Mathematics*, 7(2), 2016. doi: 10.4172/2090-0902.1000181.

[28] J. Nahon, M. Zangeneh, M. Nohmi, H. Watanabe, and A. Goto. A robust inverse design solver for controlling the potential aggressiveness of cavitating flow on hydrofoil cascades. *International Journal for Numerical Methods in Fluids*, 93(7):2291–2310, apr 2021. doi: 10.1002/fld.4974.

[29] M. Nohmi, T. Tsuneda, B. An, and T. Suzuki. Cavitation CFD prediction for NACA0015 hydrofoil flow considering boundary layer characteristics. pages 185–190, 2018. doi: 10.1115/1.861851_ch36.

[30] F. Palacios, J. Alonso, K. Duraisamy, M. Colonno, J. Hicken, A. Aranake, A. Campos, S. Copeland, T. Economon, A. Lonkar, T. Lukaczyk, and T. Taylor. Stanford university unstructured : An open-source integrated computational environment for multi-physics simulation and design. jan 2013. doi: 10.2514/6.2013-287.

[31] F. Palacios, J. J. Alonso, and A. Jameson. Design of free-surface interfaces using RANS equations. jun 2013. doi: 10.2514/6.2013-3210.

[32] O. Pironneau. On optimum profiles in stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, jun 1973. doi: 10.1017/s002211207300145x.

[33] J. Reneaux and J.-J. Thibert. The use of numerical optimization for airfoil design. aug 1985. doi: 10.2514/6.1985-5026.

[34] P. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journalof computational physics*, page pp. 357–372, 1981.

[35] M. Sacher, M. Durand, É. Berrini, F. Hauville, R. Duvigneau, O. L. Maître, and J.-A. Astolfi. Flexible hydrofoil optimization for the 35th america's cup with constrained EGO method. *Ocean Engineering*, 157: 62–72, jun 2018. doi: 10.1016/j.oceaneng.2018.03.047.

[36] J. Samareh. Aerodynamic shape optimization based on free-form deformation. aug 2004. doi: 10.2514/6.2004-4630.

[37] M. Sedlar, B. Ji, T. Kratky, T. Rebok, and R. Huzlik. Numerical and experimental investigation of three-dimensional cavitating flow around the straight NACA2412 hydrofoil. *Ocean Engineering*, 123:357–382, sep 2016. doi: 10.1016/j.oceaneng.2016.07.030.

[38] S. Shahpar. Challenges to overcome for routine usage of automatic optimisation in the propulsion industry. *The Aeronautical Journal*, 115(1172):615–625, oct 2011. doi: 10.1017/s0001924000006308.

[39] G. Shubin. Obtaining "cheap" optimization gradients from computational aerodynamics codes. *Applied Mathematics and Statistics Technical ReportAMS, Boeing Computer Service*, 1991.

[40] S. Skinner and H. Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, 62:933–962, jan 2018. doi: 10.1016/j.asoc.2017.09.030.

[41] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. jan 1992. doi: 10.2514/6.1992-439.

[42] L. Sretenskii. *Theory of wave motions in a fluid*. Moscow Izdatel Nauka, 1977.

[43] H. Sun. Numerical study of hydrofoil geometry effect on cavitating flow. *Journal of Mechanical Science and Technology*, 26(8):2535–2545, aug 2012. doi: 10.1007/s12206-012-0633-y.

[44] R. Swanson and E. Turkel. On central-difference and upwind schemes. *Journal of Computational Physics*, 101(2):292–306, aug 1992. doi: 10.1016/0021-9991(92)90007-l.

[45] M. d. P. C. Tsugukiyo Hirayama. New experimental trials for obtaining the added wave resistance f sailing yacht in heeled and yawed condition. *J. Kansai Soc. N. A.*, 1998.

[46] Q. Wei, H. xun Chen, and R. Zhang. Numerical research on the performances of slot hydrofoil. *Journal of Hydrodynamics*, 27(1):105–111, feb 2015. doi: 10.1016/s1001-6058(15)60462-0.

[47] B. Y. Zhou, N. R. Gauger, H. Yao, S.-H. Peng, and L. Davidson. Adjoint-based broadband noise minimization using stochastic noise generation. In *25th AIAA/CEAS Aeroacoustics Conference*. American Institute of Aeronautics and Astronautics, may 2019. doi: 10.2514/6.2019-2697.

# Abstract in lingua italiana

In questa tesi, il problema della riduzione della cavitazione viene affrontato attraverso l'ottimizzazione della forma utilizzando il metodo dell'aggiunto. L'obiettivo è progettare un'ala idrodinamica ottimizzata che minimizzi la possibilità di avere effetti di cavitazione in specifiche condizioni di flusso. Infatti, la cavitazione è un fenomeno indesiderato che si verifica quando la pressione locale su una superficie idrodinamica scende al di sotto della pressione di vapore dell'acqua, generando bolle di vapore che possono causare danni strutturali e ridurre l'efficienza aerodinamica. Il solver a volume finito SU2 viene adottato per simulare il flusso attorno all'ala idrodinamica e calcolare il fenomeno di cavitazione. Il metodo dell'aggiunto discreto viene applicato per calcolare il gradiente della funzione obiettivo rispetto alle variabili di progettazione, guidando l'ottimizzazione verso la forma più adatta per la riduzione della cavitazione. La verifica di sensitività è stata eseguita su un profilo bidimensionale. L'ottimizzazione della cavitazione è stata condotta su un'ala idrodinamica tridimensionale progettata per il moth del PoliMi Sailing Team.

**Parole chiave:** cavitazione, foil, ottimizzazione, metodo dell'aggiunto