UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale

Tesi di Laurea Magistrale

in Ingegneria Aerospaziale

TOWARDS DESIGN SENSITIVITIES FOR AIRCRAFT/ENGINE CONFIGURATIONS

Relatore:

Prof. ERNESTO BENINI

Correlatori:

Prof. A. MARK SAVILL

Dott. TIMOLEON KIPOUROS

Laureando: FABIO LORENZI

Anno Accademico: 2012 - 2013

SOMMARIO

Nell'industria aeronautica l'utilizzo di elementi in grado di incrementare le portanza di un profilo alare è molto diffuso. Questi elementi vengono impiegati per aumentare il coefficiente di portanza e sono particolarmente sfruttati durante decollo ed atterraggio. Queste fasi di volo costituiscono una piccola frazione dell'intera missione, nonostante ciò rivestono un ruolo molto importante nella progettazione dell'intero aeroplano.

I profili multielemento utilizzano queste appendici aerodinamiche per modificare le proprie prestazioni a seconda dei requisiti di missione. La progettazione di questi profili viene spesso sviluppata impiegando algoritmi di ottimizzazione, che comprendono simulazioni con software CFD per poter prevedere le prestazioni di tali componenti. Da alcuni anni a questa parte lo scopo delle ricerche è quello di trovare le configurazioni ottimali che garantiscono elevata portanza e bassa resistenza. Tuttavia problemi legati alla complessità del flusso attorno agli elementi ed ai vincoli geometrici dovuti alla presenza dei meccanismi di movimentazione rendono impegnativi questi studi.

Il lavoro presentato nelle pagine successive mira a realizzare uno studio di ottimizzazione robusta, utilizzando il metodo multi-punto, su un profilo bidimensionale tri-elemento dell'Airbus denominato Test Case A. Si intende individuare le configurazioni che massimizzano il c_l e sono meno sensibili alle variazioni delle condizioni operative, in particolare sono considerate incertezze sull'angolo d'attacco e sull'inclinazione assunta dal flap.

I risultati ottenuti forniscono configurazioni che migliorano il coefficiente di portanza fino all'11% e riducono la sensibilità alle condizioni operative fino al 70%. Questo elaborato dimostra che il metodo multi-punto è una tecnica efficace ed affidabile per realizzare ottimizzazione robusta.

ABSTRACT

High-lift devices are commonly used in aeronautical environment. These elements are deployed in order to increase the lift coefficient of the airfoil and they are particularly exploited during take-off and landing. Even if these phases have a short duration during the flight mission, they play an important role in the design of the whole airplane. The increase in the lift and the reduction of drag is pursued in order to reduce the fuel consumed keeping the payload constant or increase the payload keeping the fuel consumption constant.

In order to design high-lift devices and predict their performance optimization studies, involving Computational Fluid Dynamics (CFD) evaluations, are performed. Finding the optimal configurations that allow high lift and low drag is a goal that has been pursued for a long time, but problems related with the complexity of the flow around the elements and the constraints due to the deployments mechanisms make this researches demanding.

This project aims to conduct a robust optimization, using a multipoint method, that is focused on finding configurations of the Airbus Test Case A 2D airfoil that guarantee high performance, in terms of lift coefficient, and low sensibility on the variations of operating conditions. The previous works realized by MSc students developed multi-objective optimization studies that were focused on finding optimal values for the performance regardless of their robustness. In this work the objective is to find the optimum configurations that are less affected by the uncertainties on the angle of attack and of the deflection of the flap.

As result of the work, configurations, which improve the lift coefficient up to the 11% and a decrease of the sensitivities to the operating conditions up to the 70%, have been found. This project demonstrates that Multi-Point method is a reliable and powerful tool to execute robust optimization.

Keywords:

Multi-Objective Optimization, CFD Analysis, Multi-Element Airfoils, Robust Optimization, High-Lift Devices, Multi-point Optimization

RINGRAZIAMENTI

Vorrei ringraziare il mio relatore, Professor Ernesto Benini, ed i suoi collaboratori Dottor Roberto Biollo e Dottor Domenico Di Cugno che mi hanno permesso di recarmi a Cranfield per realizzare questo lavoro di tesi.

Un ringraziamento va ai correlatori Professor A. Mark Savill e Dottor Timoleon Kipouros ed Airbus UK che mi hanno permesso di lavorare su questo progetto. Inoltre vorrei porgere i miei ringraziamenti a Giuseppe Trapani che mi ha offerto sostegno tecnico e consigli indispensabili per la realizzazione del progetto.

Vorrei ringraziare tutte le persone che ho conosciuto a Cranfield, la loro compagnia e le esperienze vissute assieme hanno allietato il mio soggiorno inglese rendendolo indimenticabile.

Ringrazio tutti i miei amici e i miei compagni di corso di qui a Padova: il loro appoggio e sostegno sono stati indispensabili per concludere con soddisfazione questi 5 anni di carriera universitaria.

Infine vorrei riservare l'ultimo grazie alla mia famiglia. Non solamente ai miei genitori e a mia sorella, ma a tutta la famiglia al completo che nonostante i momenti di difficoltà mi ha sostenuto incondizionatamente sin da quando ho iniziato la mia "carriera" da studente e mi ha permesso di arrivare fin qui.

Grazie a tutti.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

AIAA	American Institute of Aeronautics and Astronautics
CAD	Computer Aided Design
CBL	Confluent Boundary Layer
CFD	Computational Fluid Dynamics
DM	Decision Maker
FBSM	Feature-Based Solid Modelling
FCH	First Cell Height
FEA	Finite Element Analysis
FFD	Free-Form Deformation
GARTEUR	Group for Aeronautical Research and Technology in EURope
GUI	Graphical User Interface
H&J	Hooke and Jeeves
HPC	High Performance Computers
LE	Leading Edge
LTM	Long Term Memory
Ma	Mach Number
MCS	Monte Carlo Simulation
MOTS	Multi-Objective Tabu Search
MTM	Medium Term Memory
NURBS	Non Uniform Rational B-Spline
PC	Polynomial Chaos
PDE	Partial Differential Equation
RANS	Reynolds Averaged Navier-Stokes
SA	Spalart Allmaras
SD	Search Diversification
SI	Search Intensification
SSR	Step Size Reduction
SST	Shear Stress Transportation
STM	Short Term Memory
TE	Trailing Edge
TFI	TransFinite Interpolation
TUI	Text User Interface

LIST OF SYMBOLS

α	Angle of Attack
Θ	Deflection Angle (for Slat and Flap)
c	Chord
c_d	Drag Coefficient
c_l	Lift Coefficient
c_p	Pressure Coefficient
$\bar{C}_p$	Canonical Pressure
$\mathbb{R}$	Set of all Real Numbers
Re	Reynolds Number
x	Horizontal Direction
y	Vertical Direction

1 INTRODUCTION

In the last decades particular attention has been paid on the reduction of fuel consumption for economical and environmental reasons. A lot of researches has been developed with the purpose of reducing consumption, analyzing different elements of the aircraft.

Some studies pointed out that small improvements in the lift and drag coefficients during the phases of take-off and landing allow a large increase in the payload that can be applied to wide-body commercial aircrafts [1]. For this reason optimizing the configuration of the airfoil during the take-off is interesting.

Previous thesis projects of MSc students were focused on finding the absolute maximum or minimum for the objective functions they were considering. They did not take into account that the operating conditions of the airfoil can be affected by small uncertainties that may interfere with the performance of the multi-element aerodynamic device. This work aims to find the robust configurations of the Airbus Test Case A airfoil that allow high performance and have low sensitvity to the uncertainties of operating conditions. In order to develop this study the multi-point optimization technique has been employed. An example of a robust optimization using the multi-point method can be given by the work of a PhD student from Cranfield [2]. This previous study has been taken as a model to develop this project.

1.1 Design Process for Aerodynamic Devices using Numerical Optimization

The design process for aerodynamic devices aims to increment the lift coefficient, and reduce the drag one, for a defined airfoil respecting all the constraints applied to the shape. The research for an optimum design is a procedure iterative and complex and requires the supervision of the designer.

The classical approach follows this pattern:

- An initial shape is chosen on the basis of physics, previous experience of the designer and market's requirements.
- The geometry is created from the selected shape employing Computer Aided Design (CAD).
- The geometry is analyzed with engineering tools, such as structural or fluid dynamics analysis in order to quantify the performance.
- Using the results obtained from the analysis the designer decides how to modify the geometry in order to improve the performance of the shape.
- Repeat the previous steps until the desired performance are reached.

This process is time-consuming and the experience of the designer plays a crucial role in the results obtained.

Numerical optimization methods reduce the time necessary for the design and make also the solution independent on the designer's experience and skills. In fact, a large number of iterations can be executed, analysing more solutions and with more accuracy.

Different methods are available in order to perform numerical optimization; each of them is more suitable to be applied to a different type of studies. The performance of the airfoil is influenced by various parameters; for this reason the Multi-Objective method is the one that best represents the real physical problem; whereas the analysis using a Single-Objective approach, even employing weight functions, would be less precise.

In order to apply this method it is necessary to follow some steps:

- Parameterization of the geometry.
- Definition of the design variables that are employed in the optimization problem. They represent the shape of the airfoil and the constraints applied to it.
- Definition of the objective functions that have to be minimized. They represent the performance of the airfoil.

In Figure 1.1 there is a schematic representation of the optimization cycle.

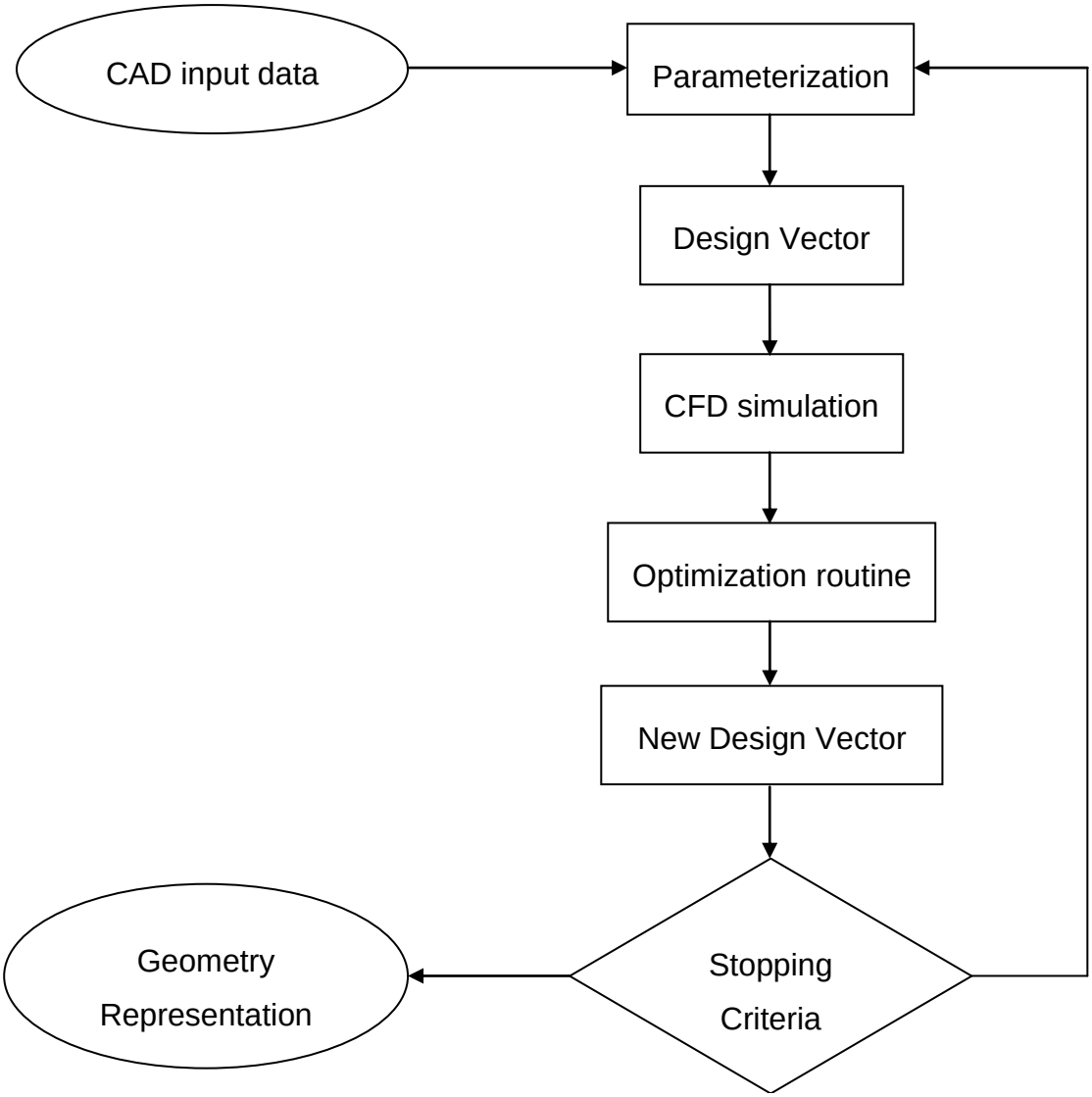

Figure 1.1: Schematic representation of numerical optimization

As it can be seen in the flow chart the generation of the geometry starts the procedure, in this stage the definition of the design variables is performed and they are saved as components of the design vector. In order to decide the number of design variables, it must be taken into account that they should be as few as possible in order to keep the computational cost low, but they must be enough to reproduce the original shape with sufficient quality and smoothness.

After this first part the optimization starts. The flow field is defined and a mesh is built around the airfoil, the CFD simulation is then performed. The results of the CFD analysis are combined together in the objective functions that, together

with the design vector, are given to the optimizer, which creates new design vectors restarting the cycle.

When the optimal configurations for the airfoil are discovered and the design space is completely explored, the stopping criteria is reached and the process ends.

In the end, the design vectors representing the optimal solutions are converted into CAD geometries in order to have a representation of them and test their performance.

1.2 Introduction to High-Lift Devices for Airfoils

Since the second decade of the nineteenth century, just after the first flight performed by the Wright brothers, the concepts at the basis of modifying the flow field with the aim of developing high-lift devices have been known [3]. Nevertheless, there was lack of complete understanding of the phenomenon and a quantitative analysis of it was difficult to carry out at that time

The first appearance of a flap in an official document can be found “Experiment on an Aerofoil Having a Hinged Rear Position”, section of the British RM No.110 published in 1914. This is shown in Figure 1.2.

Figure 1.2: RAF airfoil with a simple flap tested in 1912-1913. Source [3]

It was found out that airfoils composed by n+1 elements have higher performance compared to the n-element ones. For this reason some studies on the increased c_l due to the n-element airfoil were carried out; one example of them is the one conducted by Handley Page (cited in [3]). The Figure 1.3 represents one of the airfoil modified by Page splitting it in n elements. The angle of attack of attack chosen is 42° in order to reach the maximum lift.

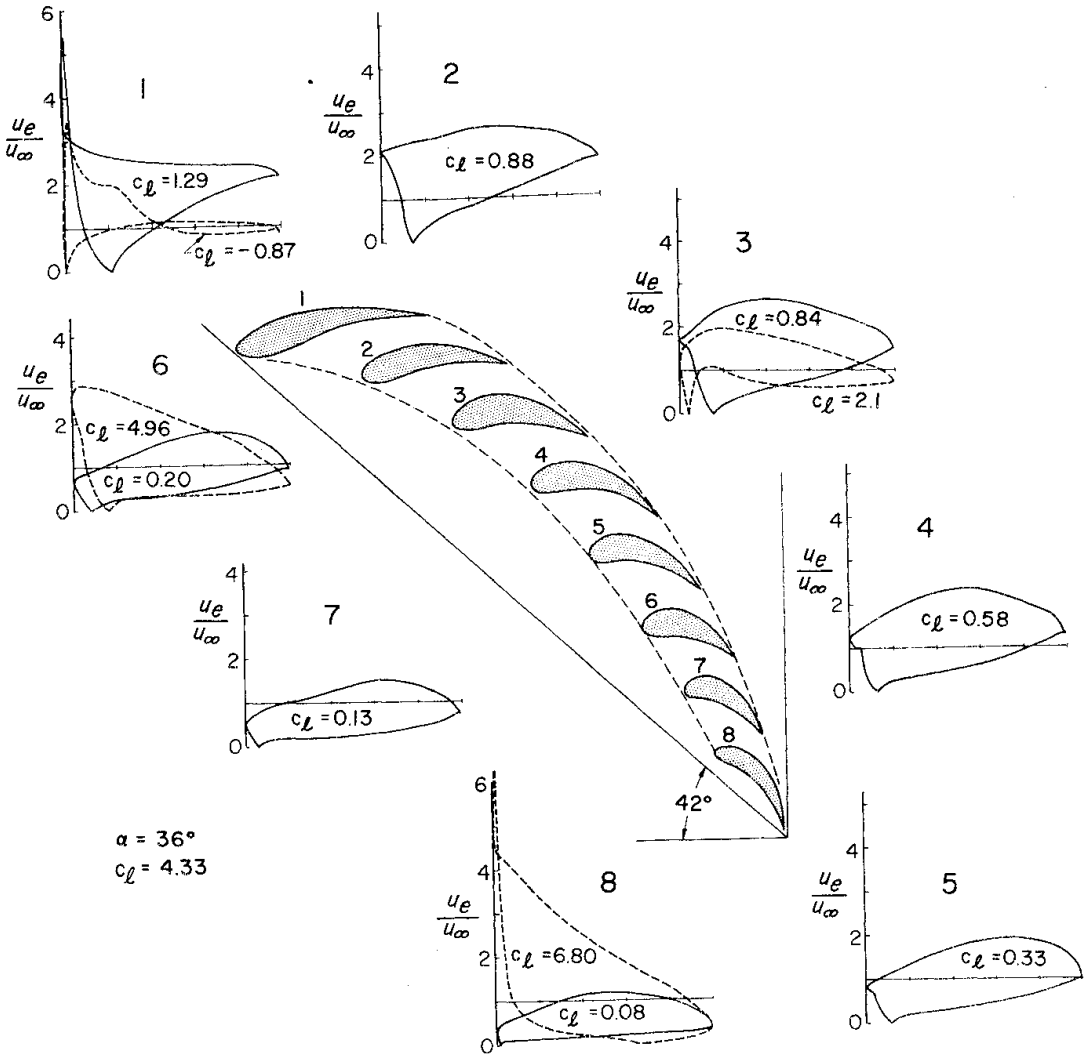

Figure 1.3: RAF 19 modified airfoil. Source [3] p.503

The reasons why flaps are employed are:

- Decreasing the flight velocity in order to perform take-off and landing in a shorter space.
- Decreasing the angle of attack for maximum c_l.
- Controlling the drag with the aim of minimizing it and reducing the noise in the airport proximity.

Many studies have been developed on the high-lift devices; nevertheless, the actual effect of the gaps between the different elements was not understood entirely in the past. As it is suggested by [3], there are five main consequences due to the presence of the gaps between the elements of an airfoil:

- Slat effect.
- Circulation effect.
- Dumping effect.
- Off-the-surface pressure recovery
- Fresh-boundary-layer effect

The ***Slat effect***, shown in Figure 1.4, is generated by the circulation that appears behind the first element. The generated vortex alters the flow field around the leading edge of the second element.

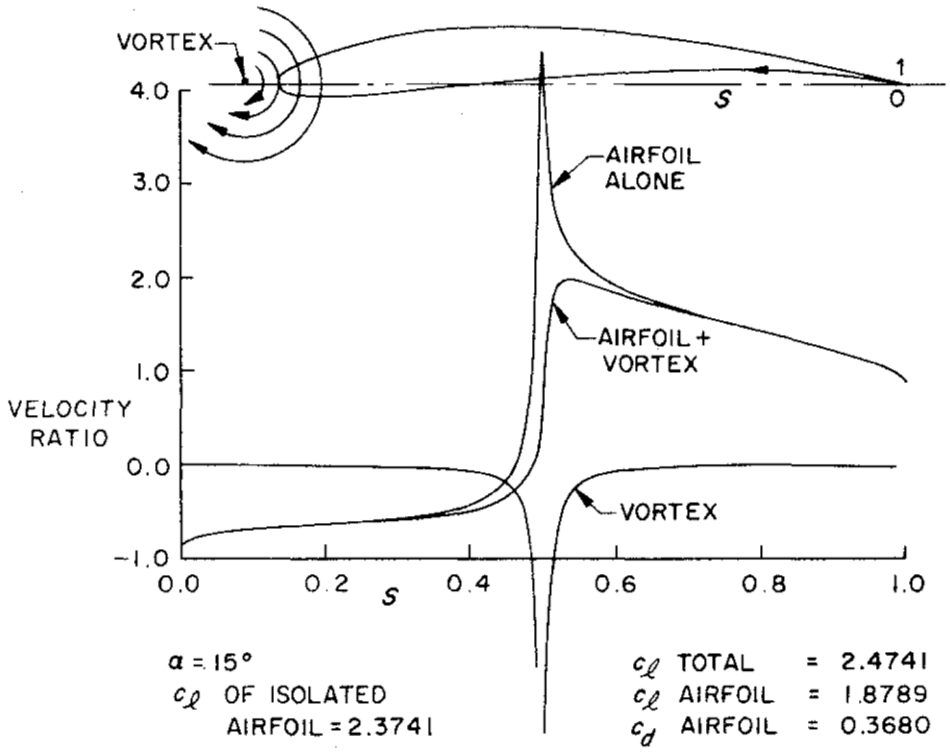

Figure 1.4: Slat effect. Distribution of velocities with and without slat. Source: [3] p.518

In Figure 1.4 the slat is represented with a point vortex. As it can be seen in the image, the slat induces velocities that are counter rotating compared to the ones that the airfoil would produce if it were isolated. This modification of the

flow field implies a reduction of the pressure peaks, moreover, even if the c_l of the second element decreases, the total c_l increases. On the other hand the total c_d increases since the presence of the vortex induces the generation of a component of the force in the tangential direction. The velocity in proximity of the trailing edge is unmodified. The effect of the slat consists in the reduction of the velocity of the flow around the leading edge of the main element of the airfoil. This effect was one of the effects that were not explicitly recognized in the early studies [3].

The ***Circulation effect*** is usually caused by the presence of an element behind the main one. In order to explain it, the same schematization employed for the slat can be applied. In this second case there will be a vortex placed in the proximity of the trailing edge (Figure 1.5).

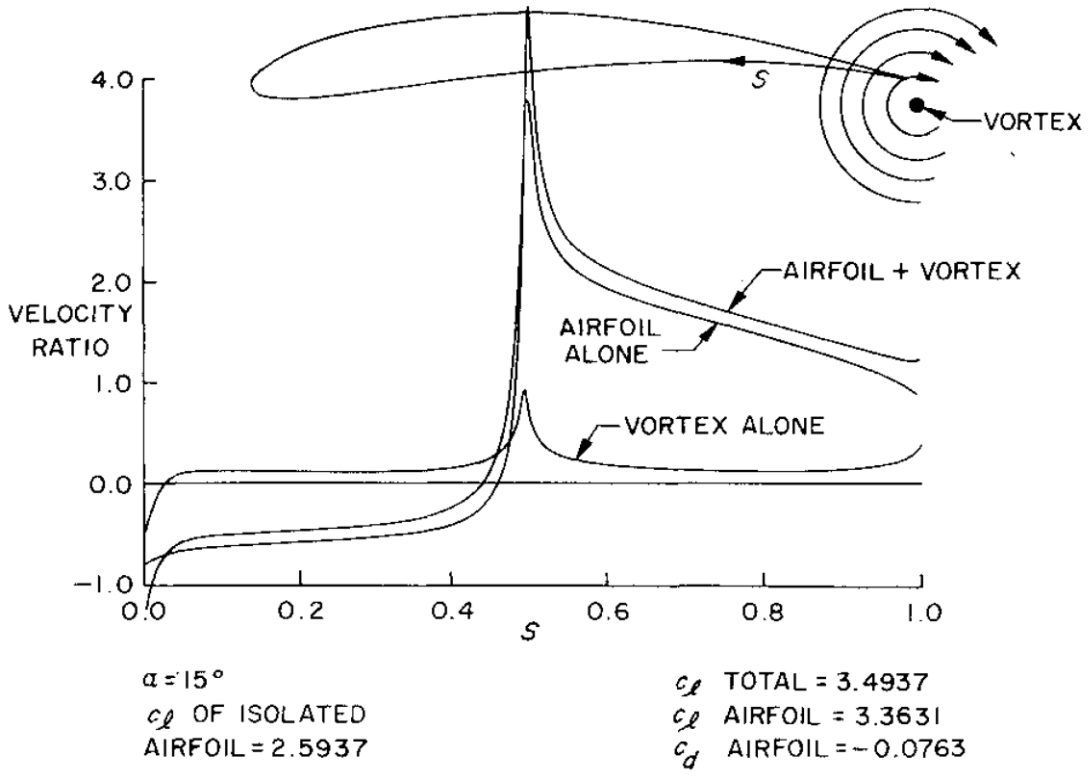

Figure 1.5: Circulation effect. Distribution of velocities with and without flap. Source [3], p. 520

The vortex creates a distribution of velocities that has the same direction of the isolated airfoil's one. Therefore, In order to satisfy the Kutta condition the total

circulation increases and, as a consequence, the c_l grows as well. The presence of a vortex or of an obstacle like a flap produces the same increment in circulation that results in the increase of the c_l. Furthermore, this change in circulation in the main airfoil reduces the angle of attack and the velocities on the rear element. This changes cause a reduction of lift coefficient in the flap.

The ***Dumping effect*** has a close interaction with the Circulation effect. The tangential velocity at the rear part of the main element can increase a lot due to the flap influence on the flow. This flow leaves the main element and goes to a region characterized by a velocity higher than the free-stream one. This effect reduces the probability of boundary layer separation since the pressure recovery is decreased. The theoretical study developed for a two dimensional three-element airfoil, in [3], shows that there is an increase of 46% of the c_l in comparison with a the single element one, maintaining the same separation behaviour. The real, three-dimensional flow is influenced by compressibility and many more parameters, therefore this high increment of lift coefficient cannot be reached. However, the influence of the Dumping effect on the performance of the airfoil is not negligible during the design of high-lift devices.

The ***Off-the-surface pressure recovery*** is a particular effect that is usually developed in multi-element airfoils. It consists in the movement of the fluid towards a region of high pressure without touching any wall. The wakes of the front elements, in fact, develop in the nose of the rear ones where the pressure is higher. The theory of these wakes has not been well developed and some analytical approach has been applied in order to study them [3]. First, it has to be known that there are two different types of wake: the first kind separates from the boundary layer, whereas in the second case the boundary layers of the two consecutive elements merge together forming a unique thicker boundary layer. The wake can develop following the first or the second case depending on the position of the elements; if the gap between two elements is sufficiently big the wake will develop following the first case, if this gap is small the second pattern will be applied. This second case is usually referred as Confluent Boundary Layer (CBL) flow.

The ***Fresh-boundary-layer effect*** is the last one to be described. It characterizes well designed multi-element airfoils, which have a new boundary layer upon each element. The airfoils designed in this way delay the separation of the flow due to the adverse pressure gradient in the thin boundary layers. In order to explain this effect Stratford's equation, presented in [3], is shown:

$$\bar{C}_p \sqrt{x \frac{d\bar{C}_p}{dx}} = (10^{-6} Re)^{1/10} S \qquad \textbf{(1-1)}$$

The parameters used in Equation (1-1) are: $\bar{C}_p$ that is the Canonical Pressure Distribution, x is the distance of the flow from the origin, Re the Reynolds number and S is a constant; detailed information about these variables can be found in [3]. The part of the equation that is on the right side of the equal symbol is almost constant, therefore $\frac{d\bar{C}_p}{dx}$ can be increased keeping the same safety against the separation just reducing x [3].This simple equation shows that breaking the airfoil in different elements it is possible to keep a thin boundary layer and sustain larger pressure gradients.

Designing a multi-element airfoil is complex; it is not easy to find the optimal gaps between different parts because there are many parameters to take into account. Nevertheless many studies were developed; they discovered that the gap between the different elements must be large enough to avoid the merging of wake and boundary layer in order to avoid an early separation [3].

1.3 Introduction to Test Case A

The Test Case A is the two-dimensional configuration obtained from the three-dimensional wing of the Airbus family of aircrafts. The section at 59% of the wing span is used and normalized (as shown in [4], p.49).

This particular airfoil has been studied by GARTEUR (Group for Aeronautical Research and Technology in EURope) [4]. Different programmes were developed in order to carry out research on the flow around this airfoil; the

AD(AG08) analyzed the performance of the airfoil in the wind tunnel, whereas the AD(AG13) and AD(AG25) meant to validate the CFD analysis of the flow [5]. The results obtained by the last two programmes were compared with the experimental data of the AD(AG08) in order to ensure the validation of the computations.

Airbus UK provided the geometry that has been employed for this work. The configuration for this given geometry is the take-off one and it is defined as “Datum Configuration” or “Datum Geometry” from now on. The representation of the datum configuration is shown in Figure 1.6.

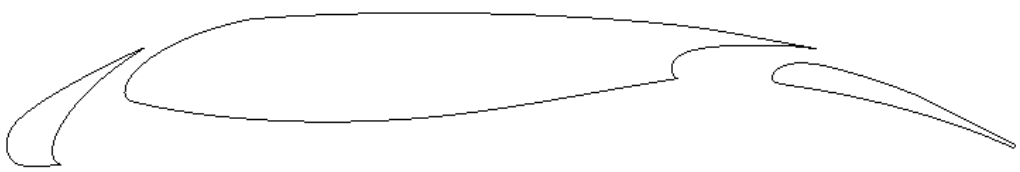

Figure 1.6: Airbus Test Case A, in take-off configuration

The dimensions of the airfoil and the relative position of the elements are not shown due to confidentiality issues.

1.4 Research Scope and Outline

1.4.1 Scope

The take-off phase is a fundamental phase of the flight mission. As previously mentioned, it represents just a very short part of the whole mission; nevertheless the improvement of the performance in this crucial moment can bring to high improvements in terms of payload (as outlined Meredith, 1993, cited in [1], p 1).

In this field multi-objective optimization has an important role, since it can lead to significant improvement in the performance. This tool has been employed for different cases in aerodynamics studies, but the intense application of it to high-lift devices is relatively recent. This is due to the fact that the optimization

process requires the development of a complex and computational demanding design procedure, which needs the application of state-of-the-art tools for each step.

Previous MSc projects were focused on finding configurations of the multi-element airfoil that maximizes the c_l and minimize the c_d [6] or maximize the ratio of c_l upon c_d and minimize the noise production at landing [7] or at take-off [8]. Those works aimed to find configurations that guarantee the maximum performance. The objective of this project is, instead, finding robust configurations that allow high values of performance and, at the same time, have low sensitivity to the uncertainty on the operating conditions. For this study, uncertainties on the angle of attack and on the deflection of the flap are considered and the robust study is developed using the multi-point method following the example of [2].

1.4.2 Outline

In order to develop the research for the optimal configurations of the Test Case A, two different studies have been executed, both of them followed the pattern presented below:

- First of all, the datum configuration was reproduced and a CFD analysis on it has been carried out; the results have been validated comparing them with the experimental data from [4].
- Second, the model developed for the datum configuration is employed in the optimization cycle; this loop includes the C++ code to execute the parameterization of the geometry, the generation of the mesh with ANSYS ICEM CFD, the CFD evaluation carried out with ANSYS FLUENT and the optimizer, i.e. Multi-Objective Tabu Search (MOTS).
- In the end, the post process is executed and the optimum configurations are analyzed and compared with the datum configuration. Moreover, the results obtained from the two different studies have been put side by side.

Chapters from 2 to 6 describe parameterization, mesh generation, CFD evaluation and set up for the MOTS; chapter 7 shows and analyzes the results; finally, chapter 8 contains conclusion and recommendations for future works.

2 GEOMETRY PARAMETERIZATION

2.1 Introduction

When an optimization process, that has the aim of finding the best shape for a certain product, is carried out, a parameterization of the geometry is necessary. This tool must generate as output a file containing the new geometry shape that can be read by the mesh generator allowing the prosecution of the optimization research. It must be automatic, flexible and robust; in fact, has to modify the geometry's shape in order to create in every loop new different forms that respect the constraints and are suitable for the generation of the grid around them. For these reason the creation of a parameterization device is a challenging part during the construction of the optimization process.

The parameterization uses the design variables as instrument to control the new shapes' generation. The number of these variables has to be enough to describe properly each shape, but, at the same time, it has to be kept as low as possible in order to maintain an acceptable computational cost. The optimization process becomes more time consuming if the number of design variables rises; in fact, the design space is explored varying the values of the different design variables and the more they are the more it is the time required to perform the exploration.

Different optimization researches require different types of parameterization; there is a wide range of techniques that can be selected; in order to perform it in the most effective way possible, the most suitable method must be chosen. For this reason in the following section a brief introduction to different parameterization approaches is presented. Then, the one chosen for this particular case is described.

2.2 Geometry Parameterization Approaches

When a geometry, that will change shape, has to be studied and it is necessary to build a mesh around it, there are two possible ways to manage the deformations and create the grid.

The first is creating a parametric mesh around the initial geometry that will deform in order to reproduce the changes in shape required by the optimizer. This strategy works properly only for small changes in shape, but it has the advantage that it does not require a grid generator inside the optimization loop, the mesh is created at the beginning and afterwards it is just deformed.

The second method consists in the creation of a completely new mesh for every new shape of the geometry. In this case the geometry, not the mesh, is parameterized. This approach allows the research in a wider design space, making possible large changes in shape. On the other hand, it requires the grid generation inside the optimization cycle, making it more complex and time consuming.

The parameterization of the geometry can be performed in different ways depending on the aim of the research; in this section the main methods are described.

2.2.1 Basis Vector Approach

This first method, suggested by Picket et al. 1973 (quoted in [9], p.879), in order to represent the change in geometry, employs the Equation (2-1):

$$\bar{R} = \bar{r} + \sum_{n} \bar{v}_n \, \bar{U}_n \qquad \textbf{(2-1)}$$

In (2-1): $\bar{R}$ represent the new shape generated, $\bar{r}$ the datum geometry and $\bar{U}_n$ is the nth design vector that represents the nth shape proposed for the geometry. Using this method a set of design variables, that are parameterized at each change in shape, is obtained. For this reason a new mesh can be created using Equation (2-1) without the need of a generator.

2.2.2 Domain Element Approach

This second approach creates a connection between some points of the grid and an element of the domain. This element is called *macroelement* and it controls the deformation of the shape. One example of this approach is shown in Figure 2.1:

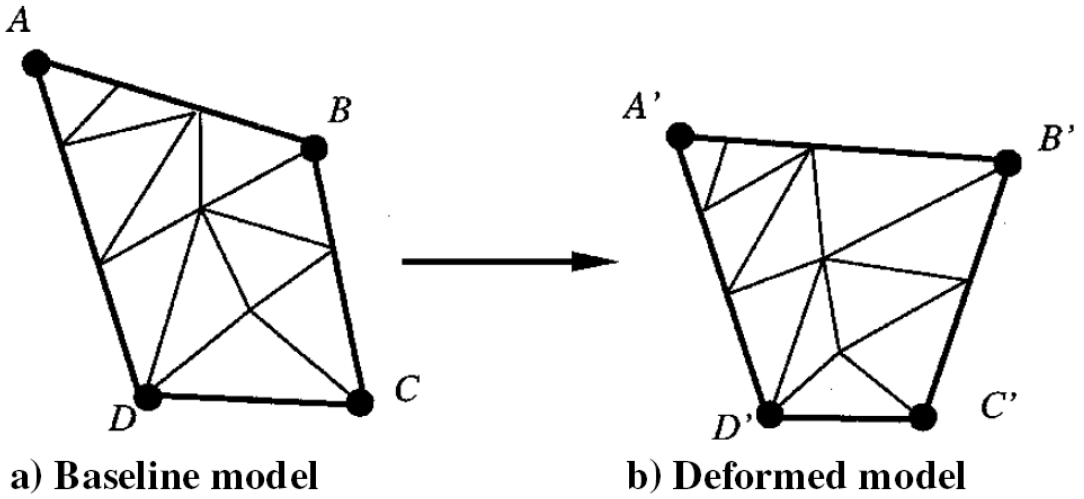

Figure 2.1: Example of deformation using the Domain Element Approach. Source [9], p. 879

The domain element in this particular case is composed by a quadrangle, the change in shape is given by the movement of the four vertex of the domain, the grid inside deforms together with the domain without changing the number of nodes. An inverse mapping between the grid and the domain makes this deformation possible. Likewise the method presented before also this one does not require the mesh generation since the grid is simply deformed. Due to the fact that the Domain Element Approach is efficient and easy to implement it is available in some commercial software.

2.2.3 Partial Differential Equation Approach

This approach was suggested in 1995 by Bloor and Wilson (cited in [9], p.879) in order to perform the parameterization of the surfaces of an aircraft. This method describes the surfaces in terms of solution of an elliptic partial differential equation (PDE). Therefore, the parameterization procedure is converted in a boundary-value problem. Employing this technique, even to describe complex geometries, just few parameters are necessary since the definition of the surface is realized using points that are located around its edge, not inside the whole surface. For this reason this approach is particularly suitable for applications that have a high computational demand, as numerical optimization.

2.2.4 Discrete Approach

This is probably the easiest method to parameterize. In order to control the shape it uses the coordinates of the points that define the geometry. In this way very complex shapes can be generated. On the other hand, in order to describe complex geometries many points are needed, increasing the computational cost required for the optimization. This approach has also a second disadvantage; it is, in fact, difficult to maintain a smooth geometry after the change in shape. For this reason, unrealistic geometries can be created, leading to problems in the prosecution of the optimization. In Figure 2.2 an unsmooth geometry modified with the Discrete Approach is shown

Figure 2.2: Geometry defined with the Discrete Approach. Source [10] p. 61

2.2.5 Polynomial and Spline Approaches

This method decreases the number of design variables necessary to describe the geometry; the shape is not defined by the set of points that composes it, but it is identified by polynomial or spline representations. In Figure 2.3, a geometry created employing this approach is presented.

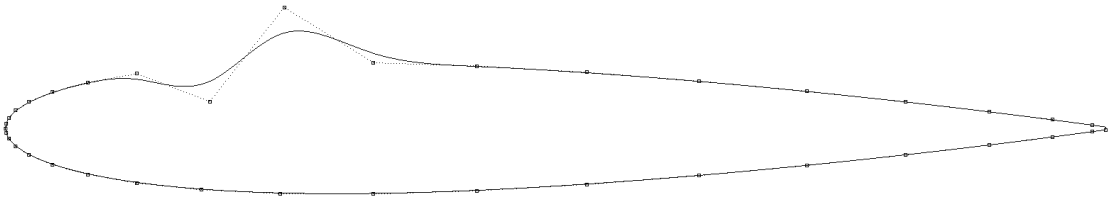

Figure 2.3: Geometry defined with the Spline Approach. Source [10], p. 62

In order to define a curve employing the polynomial representation it is necessary to use an equation like (2-2).

$$\bar{R}_g(u) = \sum_{i=0}^{n-1} \bar{c}_i u^i \qquad \textbf{(2-2)}$$

In this formula, n represents the design variables number, u is a parametric coordinate along the curve, $\bar{c}_i$ are vectors defined by the design variables.

An engineer from Renault, the French cars brand, defined one of the first approaches for the polynomial curves and surfaces. This approach is called *Bezier* representation and it takes the name from that engineer; mathematically the polynomial curves are defined by the equations shown in Equation (2-3)

$$\bar{R}_g(u) = \sum_{i=1}^{n} \bar{P}_i B_{i,p}(u) \qquad \textbf{(2-3)}$$

In (2-3) $\bar{P}_i$ are the coordinates of the n control points of the curve that are usually the design variables of the problem and $B_{i,p}(u)$ are the Bernstein polynomials of degree p. The curves defined in this way are contained within a convex polygon, whose vertices are the control points. This property is important because this polygon makes easier the implementation of the geometrical constraints for the shape during the optimization process.

Nevertheless there are some limitations related to this method. First of all, the order of the curves is equal to $n - 1$, where n has been defined previously and it is the number of control points; this causes some constraints in the flexibility of this technique. Furthermore, since the *Bezier* curve is an entity with a global nature, the change of position of a single control point modify the whole shape; for this reason it is complex to reproduce local deformations.

In order to overcome the issues related with the *Bezier* representation, a new approach has been introduced. This method consists in the composition of several *Bezier* segments of low order forming together the whole curve. This technique is called *B-Spline* and it is mathematically defined by equation (2-4).

$$\bar{R}_g(u) = \sum_{i=1}^{n} \bar{P}_i N_{i,p}(u) \quad \textbf{(2-4)}$$

In this formula all the parameters have been defined previously, except for $N_{i,p}$ that is the B-Spline basis function of p-th degree. This method gives a good representation of complex curves allowing local modifications of the shape. The only drawback related with it is the impossibility of representing conical sections with accuracy.

Another method has been developed in order to solve this last problem of conical sections representation. It is named NURBS and it is a generalization of the *B-Spline*. NURBS guarantees a good representation for many different curves, including conical sections; moreover it allows more degrees of freedom because of its definition, shown in the Equation (2-5).

$$\bar{R}(u) = \frac{\sum_{i=1}^{n} N_{i.p}(u) W_i \bar{P}_i}{\sum_{i=1}^{n} N_{i.p}(u) W_i} \quad \textbf{(2-5)}$$

Where W_i are the weights, whereas all the other parameters have already been defined. When all the weights are equal, the B-Spline method is obtained. Since NURBS is flexible and reliable, it is often employed in order to describe and model shapes with CAD software.

In general, Polynomial and Spline approaches are effective in order to represent 2D and simple 3D models [9]. More information about these methods and their characteristics can be found in [11].

2.2.6 CAD-Based Approach

CAD software can be employed for both geometry creation and modification, leading to an interesting saving in terms of development time. However, it is hard to implement, these programs are still not enough reliable to perform both the tasks. There can be, in fact, some imperfections, like transition cracks, free edges or small gaps, that are not visible and do not affect the CAD visualization. Nevertheless, these imperfections cause failure during the mesh generation, as

creation of not closed mesh or grids with leakages, which are not serviceable for the optimization process.

An alternative has been proposed, the FBSM. This approach allows the creation of complex geometries, both two and three dimensional, in a relatively simple way. It employs Boolean operations like union or intersection of simple shapes in order to create the intricate ones. This technique makes easier and faster the modification of geometries; however many tests have to be performed on it in order to verify its reliability inside an optimization loop.

2.2.7 Analytical Approach

This technique consists in the introduction of *shape functions,* which are analytical functions added to the base geometry in order to create the modified shapes. The analytical functions employed are a composition of previous designs for airfoils. Each *shape function* is associated to a coefficient, which is a design variable that establishes the contribution of each function to the final shape. This approach was introduced in 1978 by Hicks and Hanne (cited in [9], p.881).

2.2.8 Free-Form Deformation Approach

This method was presented in 1986 by Sederberg and Parry (cited in [12], p. 6). It allows modifying the shape of an object regardless of its representation. The Free-Form Deformation (FFD) technique employs soft object animation algorithms that are able to modify images and models [13]. The geometry maintains its overall shape, but can undergo through a series of different deformations, as bending, twisting compression or expansion. In order to perform this modification of the geometry, the space around the object is modified using a parametric, usually three-dimensional spline, approach. The drawback of this technique consists in the lack of physical meaning of the design variables, and the results obtained with it can be complex to analyze in case of intricate geometries. Nevertheless, this approach is employed in many optimization processes due to its versatility to manage curves and surfaces of any formulation and degree.

2.3 Geometry Parameterization Strategy

The geometry optimized in this project has already been presented in the previous chapter. It is the Airbus Test Case A two-dimensional airfoil in take-off configuration. The scope of this study is to find the optimal configurations of the multi-element airfoil; therefore the shape of the elements has not been modified, the aim of the parameterization is to modify only the positions of slat and flap relative to the main element.

In order to define the deployment of slat and flap two different coordinate systems are usually employed. The first one adopts the gap-overlap definition in order to identify the position of the element. Three variables are needed to localize the element, they are: gap, overlap and deflection angle. They are shown in Figure 2.4. This first system of reference is related with the flow physics, but it makes more complex the management of the parameterization inside the optimization process.

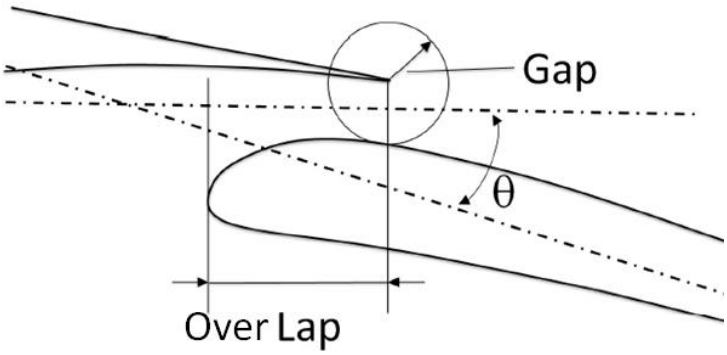

Figure 2.4: Gap-OverLap definition. Source [14], p. 157

The second method uses the Cartesian coordinate system; an example of it can be seen in Figure 2.5.

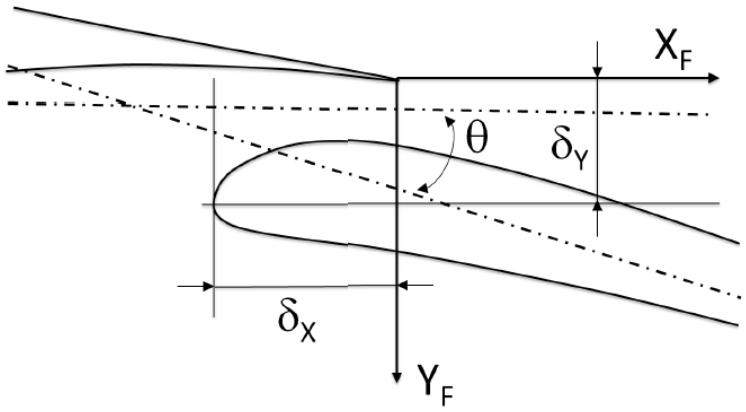

Figure 2.5: Cartesian coordinates definition. Sources[14], p. 157

This second representation has been chosen for the study since it is simpler to implement in the optimization. Also with this reference three parameters are necessary in order to define the position of each element. These parameters are:

- δx: The space along the horizontal direction between the LE of an element and the TE of the previous part.
- δy: The space along the vertical direction between the LE of an element and the TE of the previous part.
- θ: The angle that is formed by the intersection of the consecutive elements' chords.

In the practical parameterization the variables used are relative to the datum configuration. Δx and Δy are the element's translations, respectively along the horizontal and vertical directions, relative to its datum position as shown in Equations (2-6) and (2-7); Θ is defined similarly as deflection relative to the datum configuration (Equation (2-8)).

$$\Delta x = \delta x - \delta x_0 \quad \textbf{(2-6)}$$

$$\Delta y = \delta y - \delta y_0 \quad \textbf{(2-7)}$$

$$\Theta = \theta - \theta_0 \quad \textbf{(2-8)}$$

In these equations the subscript “0” indicates the datum configuration. Moreover, the coordinates are non dimensional because they are normalized dividing by the chord.

Since the position of an element is defined by three parameters and the elements that are moved during the optimization process are two, there are six design variables that compose the design vector. The first three variables define the slat position and the second three the flap one.

Two different optimization studies have been developed in this project: the first one executes three CFD evaluations for each configuration, whereas the second one runs nine CFD simulations for each design vector. Thus the second optimization process is slower compared to the first one. Further details about the two optimizations studies are presented in Chapter 5 and Chapter 6. In Table 2.1 the range of variability and the initial step for the design variables in the two cases are presented. For the second study, it has been chosen to double the initial step for the slat and the flap deflections.

	Slat			Flap		
	Δx	Δy	Θ	Δx	Δy	Θ
Initial step	0.02	0.01	1°	0.02	0.01	1°
Min range	-0.06	-0.05	-10°	-0.17	-0.06	-10°
Max Range	0.14	0.09	10°	0.09	0.00	10°

Table 2.1: Design Variables range of variation.

In order to define the values of the design vector it has been taken as reference [2], where the range of the design variables was studied in order to reproduce in a simplified way the constraints due to the deployment mechanism.

However, these choices for the design variables may not be the best one for a short optimization process. In fact, the initial step is small and it cannot guarantee a wide exploration of the design space. Nevertheless since the exploration is around the datum configuration and this configuration should be close to the optimum ones, this initial choice can be considered acceptable.

2.3.1 Parameterization Script

In order to perform the parameterization it has been chosen to follow the same strategy presented in [8]. A C++ code has been implemented in order to execute the deployment of the airfoil.

This script reads the datum geometry from a *.txt* file that contains the coordinates of the points that compose the airfoil and receives as input also the design vector, with the six variables that define the position of slat and flap. The output of this code is another *.txt* file that contains the coordinates of the points of the airfoil in the new configuration.

In order to perform the deployment the code follows some recurrent steps:

- First of all, the TE of the slat and the LE of the flap are found.
- Second, the rotation is executed using the angles specified in the design vector. Slat is rotated around its trailing edge and flap around its leading edge.
- Third, the two elements are translated in the x and y directions employing the remaining parameters of the design vector.

This simple procedure allows the execution of the deployment. In Figure 2.6 an example of deployment is shown in comparison with the datum configuration. The datum geometry is represented in white, whereas the new configuration is the one in black.

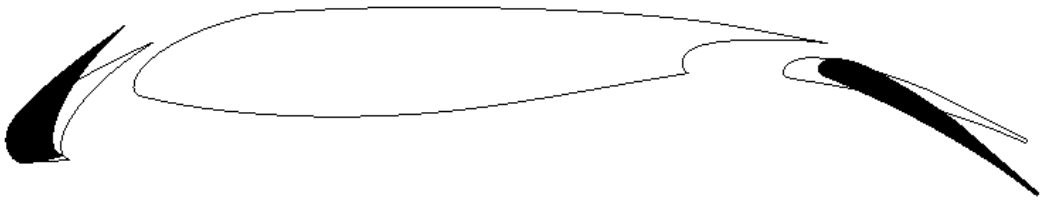

Figure 2.6: Airfoil deployment

This parameterization can lead to the creation of unfeasible configurations, where there are intersections between the different elements of the airfoil. An example can be seen in Figure 2.7.

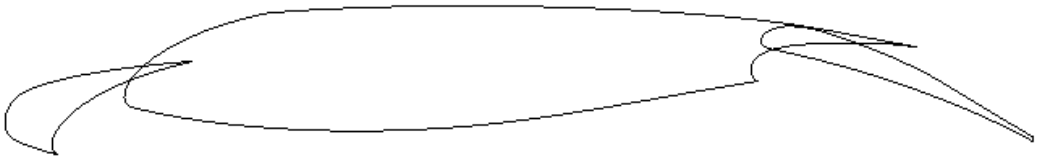

Figure 2.7: Unfeasible geometry

This event must be avoided for two main reasons. First of all, in the real life there cannot be any intersection between different elements. Second, the creation of an unfeasible geometry leads to errors in the mesh generation and can create problems in the optimization process.

For this reason, a code in C++ has been developed in order to detect unfeasible geometries and discard them. This script is quite simple; however it is efficient and works properly. This code works as presented below:

- It divides the main element of the airfoil in two curves, an upper one and a lower one.
- It checks if there are points of the slat or of the flap that are inside the area defined within the two curves.
- If there are no points of the slat and of the flap inside the main element the geometry is considered valid and the optimization process can

continue; otherwise the code returns a flag that marks the configuration as “non valid”, the geometry is then discarded and the optimization can proceed analyzing a new design vector.

The check on the geometry realized with this script cannot detect if the different elements are too close each other, causing complexities in the generation of the mesh. For this reason another check is realized in the optimization process. The program employed for the mesh generation, ANSYS ICEMCFD, in the case of lack of enough space between the elements is not able to generate the mesh. Therefore, once the grid generation process terminates a check is made in order to verify if the mesh file (*.msh*) is created. If the file *.msh* is not found the geometry is marked as unfeasible and discarded, a new design vector is then analyzed. Further information about these checks and the way they work will be presented in Chapter 6.

Once the parameterization is completed and the geometry is checked and marked as feasible the mesh generation can be executed.

3 MESH GENERATION

The following step, after the parameterization of the geometry, is the generation of the mesh around the airfoil. This is necessary in order to perform the CFD evaluations; in fact, the domain must be divided into small cells in order to find the solutions of the RANS equations. RANS equations are solved substituting the partial differential equations defined in a continuous domain in with algebraic equations defined in a discrete one. The grid composed by the total number of the cells is called mesh, and the process that leads to the creation of it is called mesh generation.

3.1 Introduction

The mesh generation is a critical phase within the development of a CFD evaluation. It is necessary in order to perform the flow study and it has a crucial influence on the quality of the solutions. In order to solve the RANS equations, in fact, an enough fine mesh is needed, since all the details of the geometry must be described with a minimum level of accuracy; otherwise, if the mesh is too coarse, the results obtained can be without physical meaning or, in the worst case, the solutions can diverge. Moreover, the mesh should not be too fine; in fact, the larger the number of nodes is the longer time is needed to execute the CFD evaluation. In an optimization process, in particular in a robust optimization study, many CFD simulations are performed; therefore a short time for the resolution of the RANS equations is one of the main goals.

Another reason why the creation of the grid has significant importance in the computational fluid dynamic study is the time needed to create it. It is still not possible to create the mesh automatically, and the procedure is time consuming and requires some experience and skills. During the years many improvements have been made on the software employed for the mesh generation; nevertheless the process cannot be realized completely in automatic and user-free. Since every single case is unique, the judgment and the skills of the designer are necessary in order to create it with technical knowledge and a little bit of art [15].

Commercial software include different methods for the mesh generation. Before describing the procedure developed in this project, different grid creation techniques are presented in the following sections.

3.2 Mesh Generation Approaches

The definition of mesh can be found in [15]: it is a structure composed by a distribution of points spread on the fluid domain with the aim of allowing the resolution of a set of partial differential equations. The grids can be divided in two main categories: Structured and Unstructured. The difference between these two groups consists in the diverse way in which the point of the mesh are created and saved, and the different shape that the elements assume. Finally there is a third category, which is a mixture of the first two and means to gather the advantages of both of them.

In this section a short presentation of the different mesh categories is presented describing the advantages and drawbacks associated with them.

3.2.1 Structured Meshes

A structured grid is obtained intersecting curvilinear coordinates surfaces, in this way the domain can be divided in regular cells that are quadrilateral in a two-dimensional case or hexahedral in a three-dimensional one. Structured meshes were born with the orthogonal Cartesian coordinates, however if the shape, which has to be analyzed, has any type of symmetry the related coordinates are orthogonal as well; some example of possible symmetries are elliptical, cylindrical or spherical symmetries. This kind of mesh can be created following two different strategies: the first is an algebraic method, whereas the second consists in finding the points as solution of a PDE.

The algebraic case requires the employment of interpolations methods that allows the creation of the grid starting from the boundary. Depending on the interpolation strategy selected, different variants of this method can be developed; however the most diffused one is the TransFinite Interpolation (TFI), which is the fastest technique for the generation of structured meshes, but on

the other hand it can have some problems related with smoothness and overlapping.

The PDE method leads to the creation of grids that are not affected by the problems that concern the algebraic ones, like low smoothness and overlapping. Therefore, PDE approach is often employed in order to smooth the algebraic meshes. A structured grid is shown in Figure 3.1.

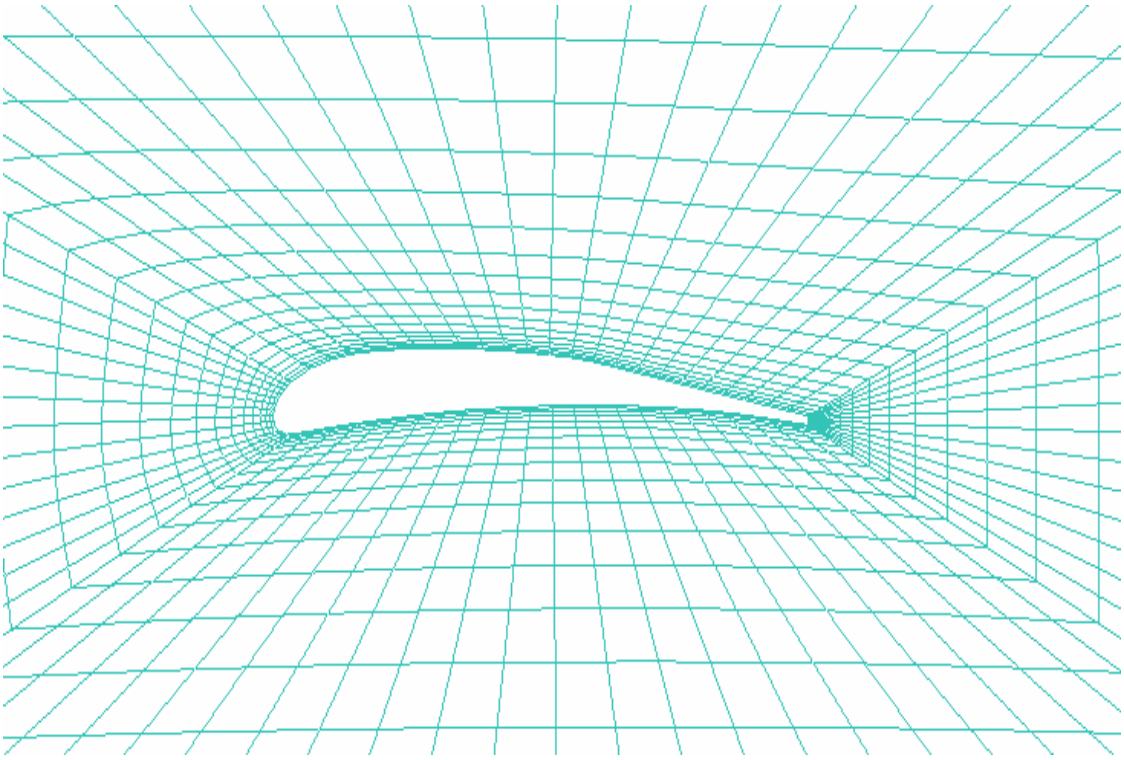

Figure 3.1: Example of structured mesh around an airfoil

The advantages associated with the use of a structured mesh are the following:

- Accurate results are obtained since the cells are usually aligned with the flow.
- Simple and easy access to the data is guaranteed: the grid can be described, in fact, using just two or three indices. The points of the mesh are stored in arrays and, without using any more information, the connectivity between the adjacent nodes can be calculated. Therefore, the memory required to store a structured grid is less than the one necessary for an unstructured one.

- The memory needed by the flow solver during the flow evaluation is less than the one employed by an unstructured mesh of the same size.

On the other hand structured meshes have some disadvantages:

- The creation of a structured mesh is more time consuming and necessitates some experience from the user to be built properly, especially if the geometry is complex.
- If the geometry is complex, containing sharp edges or mathematical discontinuities, a structured grid may not be appropriate; in fact, in order to adapt to the geometry the mesh can reach too high skewness and aspect ratio that can lead to imprecise or unphysical solutions.
- When the mesh is refined, the refinement takes place in the whole domain, leading to the increase in density of the grid also in the zones where a finer mesh is not required.

More information about characteristics, advantages and disadvantages of this particular type of mesh can be found in [15].

3.2.2 Unstructured Meshes

Unstructured grids are usually composed by triangular or quadrilateral cells, in 2D cases, tetrahedral or hexahedral cells, in 3D problems. This kind of grid is generated without using any structure to define the position of the points that compose it. Nevertheless, the procedure, which leads to the mesh generation, follows a precise scheme, even if it is not unique.

In order to create an unstructured mesh, some algorithm, based on geometrical notions, are employed. The Delaunay triangulation is one of the most popular methods used in the unstructured grid generation procedure and it is employed by most of the commercial software; this technique follows the concept stated by Dirichlet in 1850: any type of domain can be divided into elements of convex polygonal shape. The original approach employs triangular (tetrahedral in 3D) elements, but some modifications on the generating algorithms can be done in order to obtain quadrilateral cells (hexahedral in 3D). An example of unstructured mesh with triangular elements is shown in Figure 3.2.

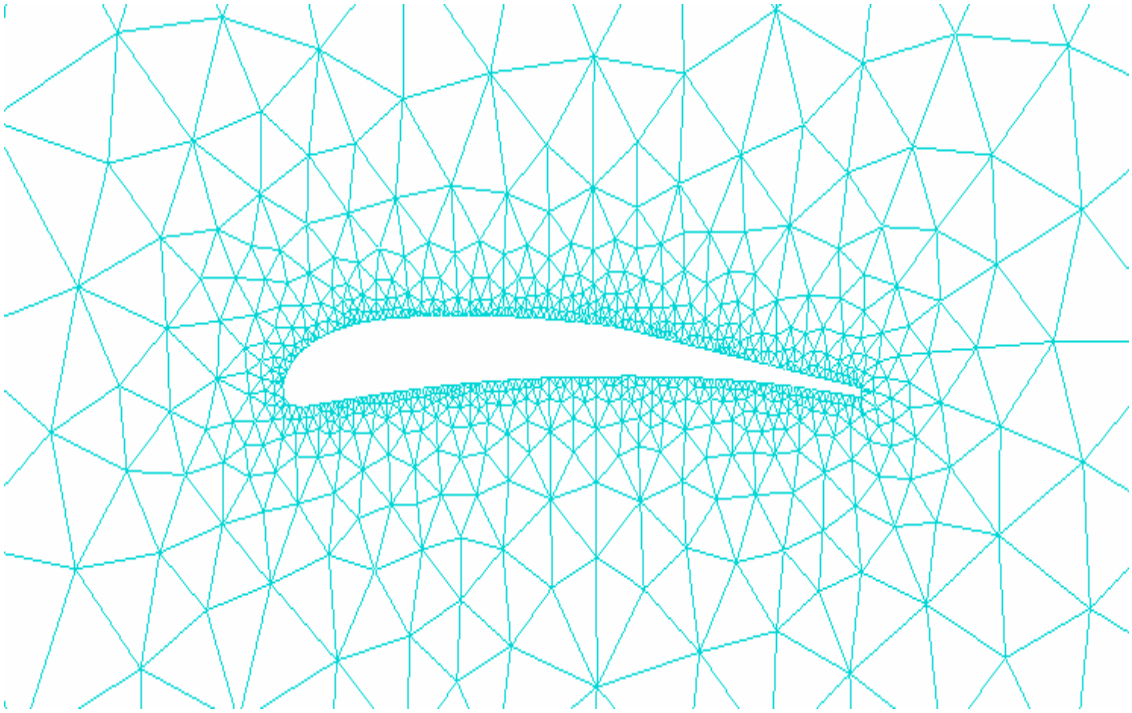

Figure 3.2: Unstructured mesh around an airfoil

This type of mesh leads to some important advantages, compared to a structured one. These advantages are listed below:

- Easy generation, it can be done nearly automatic. It requires less time and less experience from the user.
- High flexibility. The mesh can adapt to complex geometries, with sharp edges or mathematical discontinuity, without decreasing the quality to the solutions.
- The refinement can be realized automatically.

On the other hand this type of mesh has some drawbacks in comparison with structured grids:

- The flow is seldom aligned with the grid (especially if the triangular elements are adopted), leading to a low quality of the solution. In order to improve it a second order scheme for the resolution or an increase in the number of nodes must be adopted.
- For the same mesh size, more memory is needed to save an unstructured mesh compared to a structured one. In fact, more

information about the connectivity between the adjacent nodes must be stored since these data cannot be calculated as it can be done with a structured grid.

- There is a higher dependency of the quality of the solution on the aspect ratio of the elements. Low tolerance on the high aspect ratio of the tetrahedral cells near the boundaries leads to low accuracy it these parts of the domain.

More information about this kind of mesh and its characteristics can be found in [15].

3.2.3 Hybrid Meshes

This third type of mesh is created combining the two kinds of grids described previously. Hybrid meshes are usually developed in this way: a structured grid is generated in the regions close to the wall in order to reproduce accurately the boundary layer around the airfoil, while in the rest of the domain an unstructured mesh is created. The aim of this third category of grids is to combine together the advantages offered by structured and unstructured meshes and reduce their drawbacks as well.

This technique is suitable for the high-lift devices case, since a good representation of the boundary layer is needed in order to evaluate with enough precision the forces acting on the airfoil; and the geometry is quite complex making the generation of a totally structured grid difficult.

An example of hybrid mesh is presented in Figure 3.3 and a detail of it in which is possible to see the structured and the unstructured regions is shown in Figure 3.4.

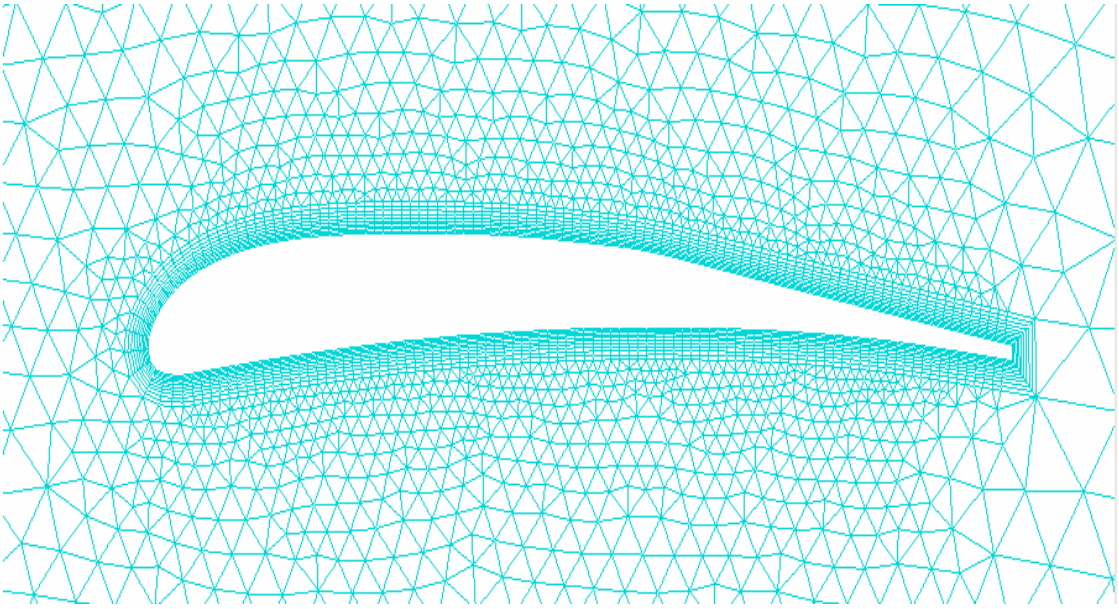

Figure 3.3: Hybrid mesh around an airfoil

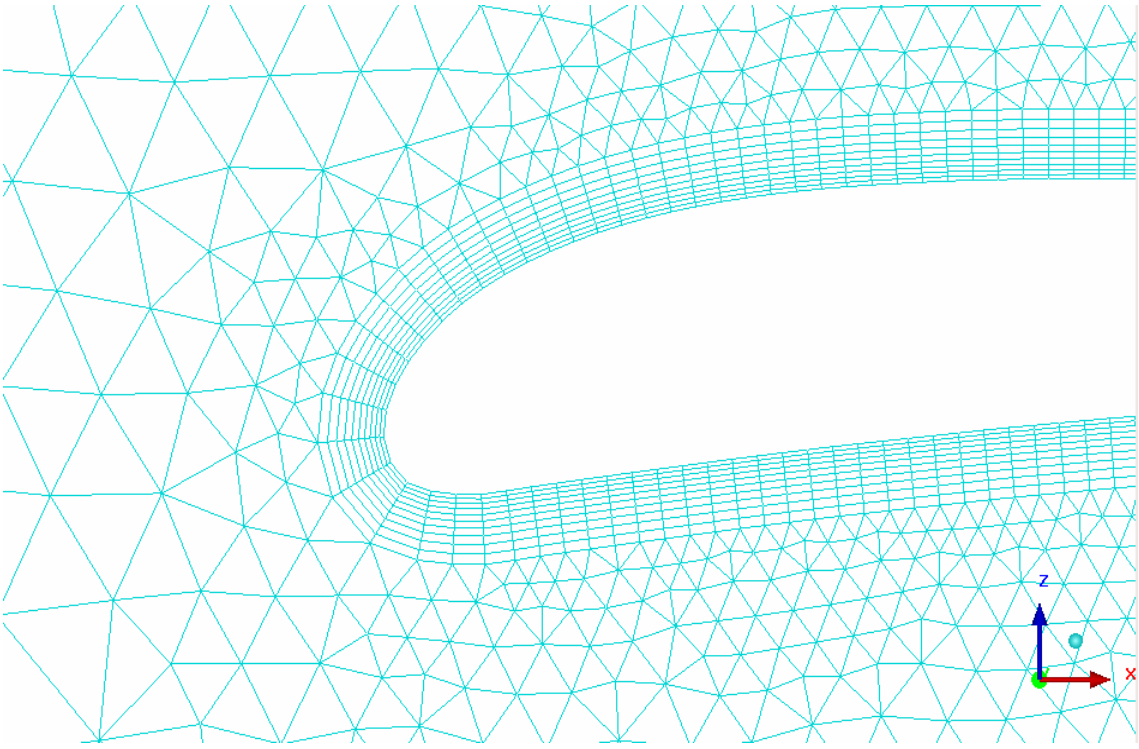

Figure 3.4: Detail of a hybrid mesh around an airfoil

3.3 Grid Generation Approach and Automation

During the last decades computational fluid dynamics grew in importance and has been employed by a continuously increasing number of users. For this reason many commercial software for mesh generation have been developed; this kind of software offers the users a powerful tool which allows the creation of the grids presented in the previous paragraphs thanks to the implementation of these methodologies in their scripts. Therefore, the creation of the grid using home-made codes is no more necessary.

In order to generate the mesh used for the optimization process commercial software has been employed, this software is ANSYS ICEM CFD 14.0. It provides a graphical interface that allows the creation of the grid in an interactive environment.

This tool is very powerful and versatile; it offers the possibility of creating structured, unstructured and hybrid meshes; all the information necessary to build them can be found in the manual [16].

3.3.1 Mesh Approach

In order to choose the best type of mesh to execute the CFD evaluation around the Test Case A, some issues have been taken into account:

- First, the geometry is quite complex and it is not easy to build an accurate structured mesh. It is likely that high values of aspect ratio and skewness or even overlapping occur; this can leads to imprecise or unphysical solutions or, in the worst case, divergence.
- Second, during the optimization cycle the elements that compose the airfoil move; therefore, the mesh has to be rebuilt in order to adapt itself to the change of the geometry. Thus, if a, accurate structured mesh were built for the datum configuration, it could not maintain its quality once the geometry moves, leading to the problems presented in the previous point.
- Third, a completely unstructured mesh has a low resolution in the boundary layer region. Therefore it can have some problems in the

evaluation of the flow in the near wall area, leading to errors in the prediction of the lift and drag coefficients.

Keeping in mind these considerations and following the example found in [2], it has been chosen to build a hybrid mesh around the airfoil. In fact, this kind of grid can adapt easily to the geometry and to the movement of the elements. Moreover, the structured near wall region allows a good representation of the boundary layer and as a consequence a good evaluation of the force coefficients acting on the airfoil.

3.3.2 Mesh Generation

In order to build the mesh it is necessary to follow a process composed by different steps, remembering that the mesh generation must be automatic in order to be performed in the optimization process.

First of all, the geometry must be imported from the *.txt* file generated by the parameterization tool. This file contains the points that define the elements of the airfoil. Once the set of point is imported in ICEM CFD the curves that represent the airfoil are created. Afterwards, the fluid domain is generated as shown in Figure 3.5.

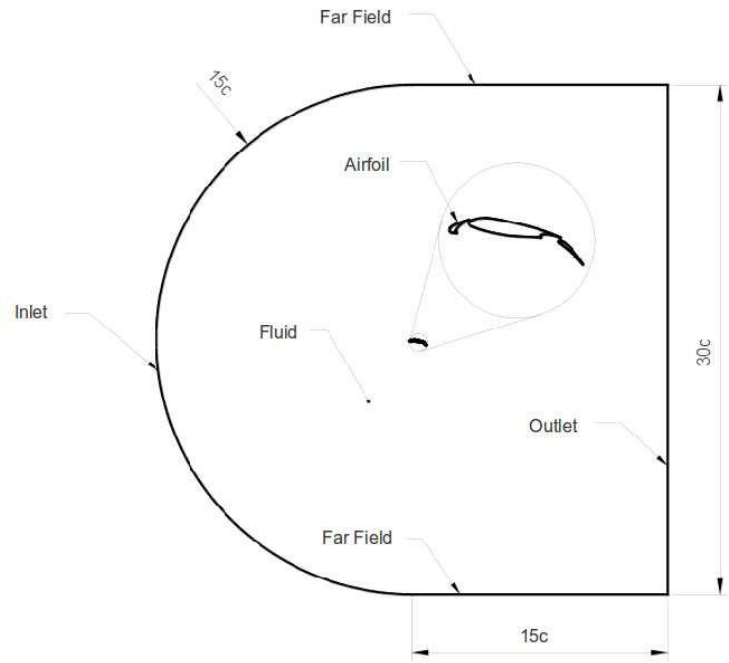

Figure 3.5: Definition of the fluid domain

Taking as example the work developed in [8], the fluid domain has been built setting the far field boundaries at a distance equal to 15 times the chord from the airfoil. This value has been chosen in order to satisfy the conditions of far field. Then, other elements (shown in Figure 3.6) have been created in the domain in order to control the mesh generation. The straight line behind the flap aims to increase the grid density in that zone in order to reproduce the wake region with enough precision; this line is set to start always at a distance equal to 0.01 times the chord downstream the flap, whatever the position of this element is. The two closed figures between the airfoil and the far field have been created in order to control the change in size of the grid, in fact a mesh coarser in the far field and finer in the near wall region, with a gradual change in size between the two zones, is pursued.

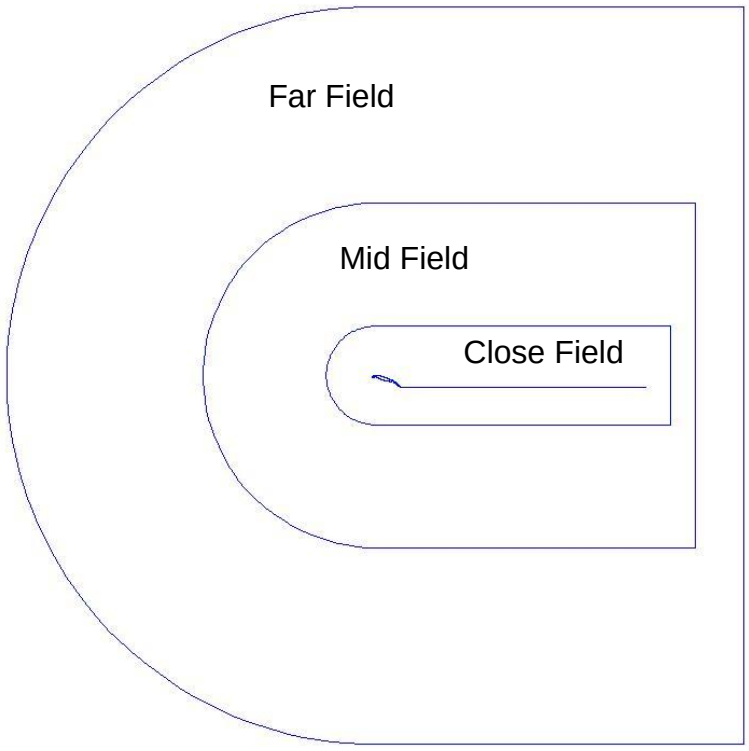

Figure 3.6: Subdivision of the fluid domain

Once the geometry was completely created the settings for the mesh have been imposed. The patch dependent method has been selected since this process builds a grid that adapts to the curves and the geometrical elements defined in the fluid domain [16]. Moreover, quad dominant unstructured mesh has been chosen, taking as reference [2]. The presence of quadrilateral cells, in fact, presents the following advantages in comparison with triangular ones:

- Decrease in the number of nodes, because of the shape of the cells; this is more effective in the far field region where the mesh can be coarser.
- A higher number of cells is aligned with the flow, thus the numerical error of the solution is reduced
- Reduction of the time needed for the calculations keeping the same mesh size of a triangular mesh.

The grid has been created defining the distances between two consecutives nodes along the curves in the fluid domain. This distance has been set for the different regions as presented below:

- 0.6 times the chord in the far-field region, inlet and outlet.
- Below 0.001 times the chord in the airfoil walls.
- Increasing with an exponential law in the wake, starting with 0.001 times the chord in proximity of the flap trailing edge, and increasing towards the far field (a detail of it can be observed in Figure 3.11).

Since the change in size is large the two control curves (shown in Figure 3.6) are needed; in fact, in the region around the airfoil the mesh must be kept fine in order to describe the complex flow around the airfoil, while in the far field a coarser mesh can be used; furthermore a gradual change in size must be realized with the aim of avoiding numerical errors during the resolution of the RANS equations.

In the end, the settings for the structured boundary layer had to be defined. Due to the turbulence model selected for CFD study (K-ω SST model, which is presented in the following chapter) a $y^+ \leq 1$ is necessary to set the first node of the grid inside the boundary layer region [17]. An iterative process, changing the

grid spacing and checking the y^+ from the CFD results, has to be done to find the appropriate spacing for the first node of the layer; in fact, the value of y^+ cannot be predicted with high precision before the execution of the CFD evaluation. To start the iterations, the grid spacing has been calculated using a tool that can be found in [18]. This tool calculates the starting value for the First Cell Height (FCH) on the basis of the values of the Reynolds number of the flow and the reference length, which is the chord for this case. A detail of the structured boundary layer can be observed in Figure 3.10.

Once all the settings were defined the mesh has been generated. The grid obtained (shown from Figure 3.7 to Figure 3.11) has 265365 cells and an FCH of the order of 10^{-6} times the chord length; this values are acceptable for this case and they are in agreement with the ones presented in [2].

The mesh has been saved in the format *.msh*, suitable to be read by ANSYS FLUENT 14.0. In contrast with what was done by the previous MSc thesis ([6] [7][8]) the mesh has not been extruded to make it 3D since ANSYS FLUENT 14.0 has been chosen as solver, and it can analyze two-dimensional flows.

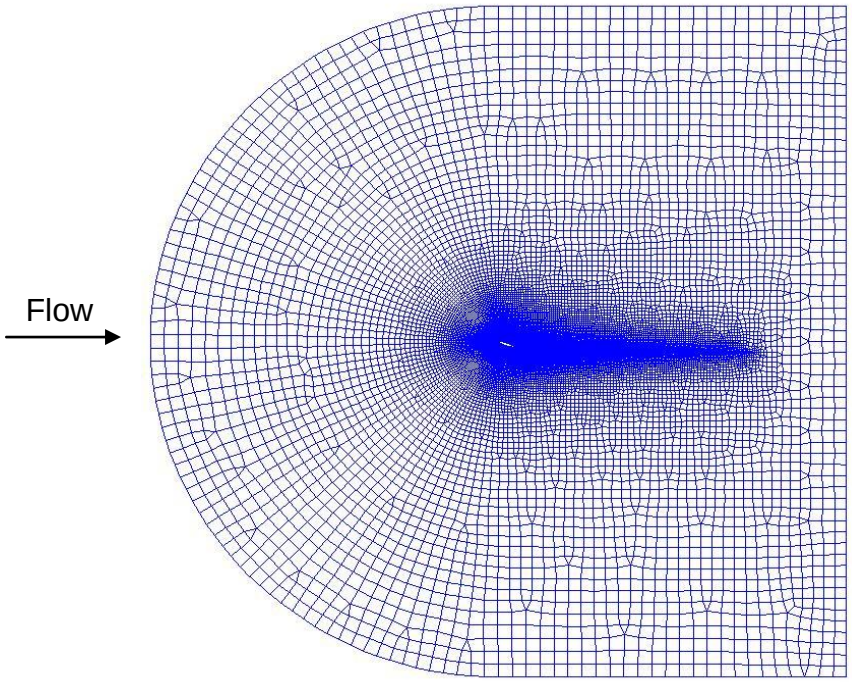

Figure 3.7: Quad dominant Hybrid mesh

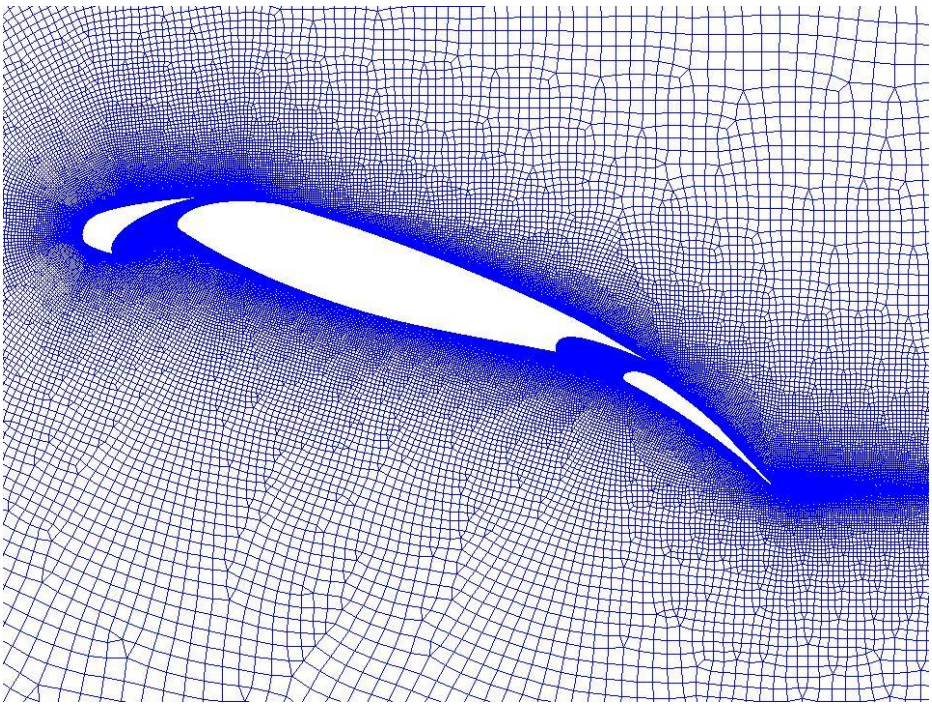

Figure 3.8: Quad dominant Hybrid mesh, detail of the airfoil

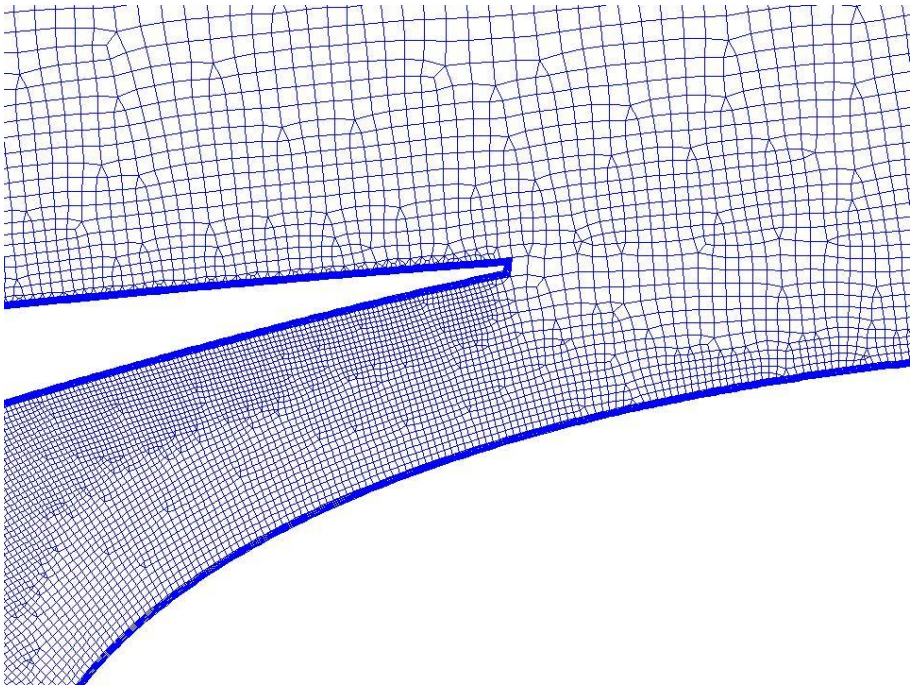

Figure 3.9: Quad dominant Hybrid mesh, detail of the near wall region

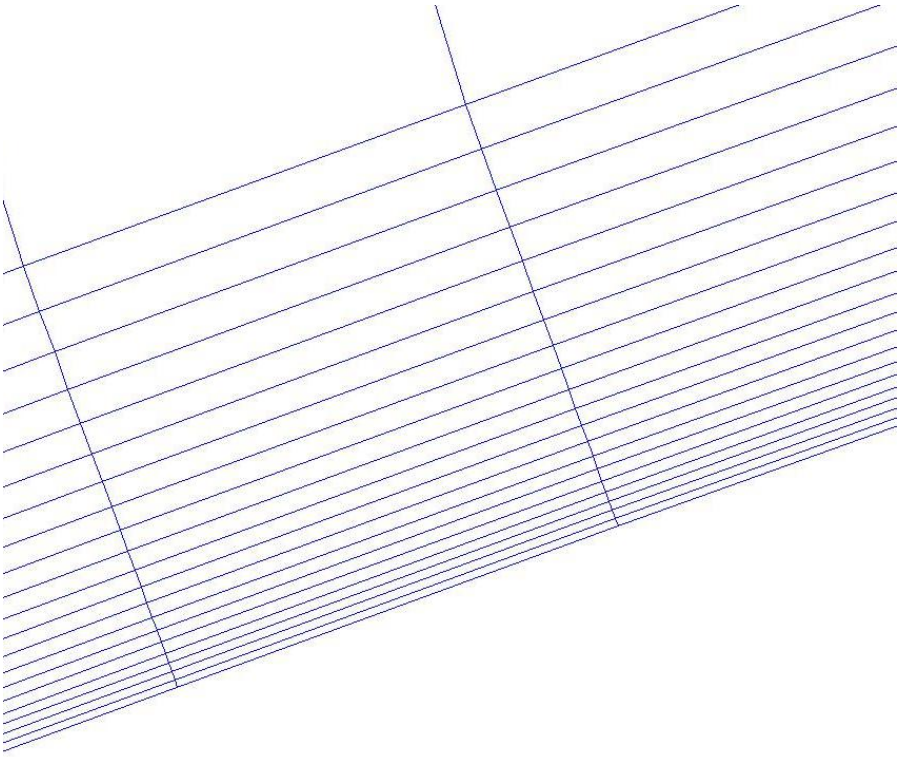

Figure 3.10: Quad dominant Hybrid mesh, detail of the structured boundary layer

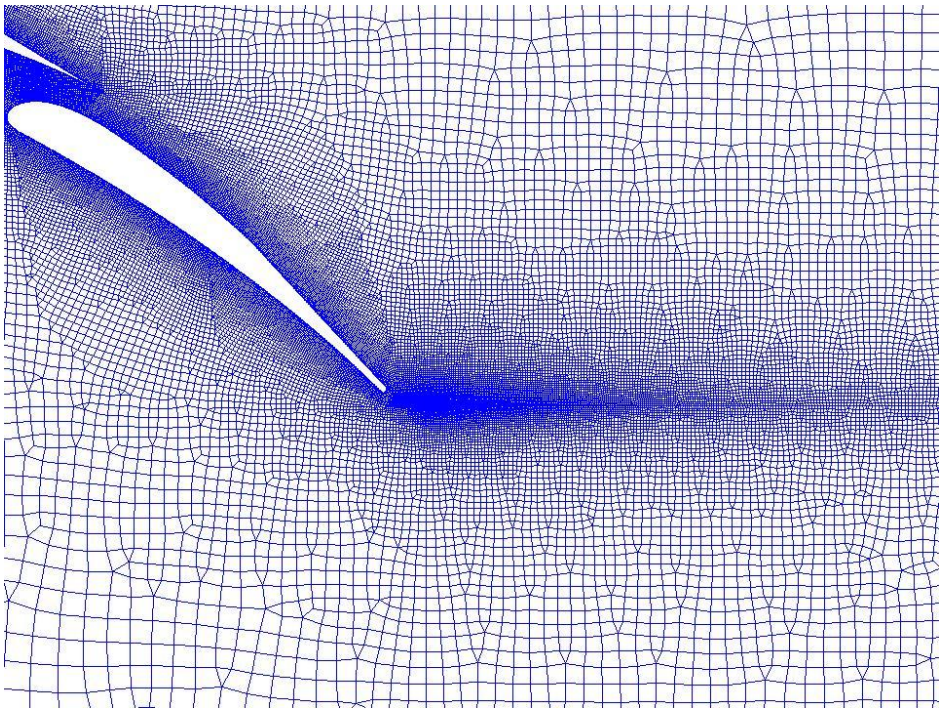

Figure 3.11: Quad dominant Hybrid mesh, detail of the wake region

3.3.3 Mesh Automation

During an optimization process the simple generation of the mesh is not sufficient; it is, in fact, necessary to automate it. ANSYS ICEM CFD offers a powerful and user friendly tool in order to realize this automation. It is possible, in fact, to record all the procedure followed in the mesh generation in a journal file (*.rpl*) just activating the recording before starting the creation of the grid. This tool can be used even though the mesh is realized using the graphical interface of the software, making the creation of the script file easy also for users with few experience. The journal file can be modified manually in order to change the settings for the grid creation and reach the quality requirements aimed. The *.rpl* script is then integrated in the optimization process and, thanks to its parametric formulation, it allows the generation of the mesh around the different geometries produced by the parameterization tool.

4 CFD ANALYSIS

4.1 Introduction

The evaluation of the flow around the airfoil employing a CFD solver is the most computational demanding and time consuming step during the optimization process. Moreover, this phase is critical for the achievement of the results that can make the optimization successful. In fact, the objective functions are calculated using the results of the CFD evaluations.

In order to reproduce the flow field, especially in a complex case like the multi-element airfoil, a fine mesh has to be built, high order schemes of resolution have to be adopted and the most appropriate turbulence model has to be chosen. Furthermore, the computational time must be taken into consideration, remembering that the higher is the accuracy of the solution the longer is the time required for the computation of the results. Therefore, a compromise between accuracy and computational time has to be reached.

The introduction of CFD evaluations in the design and development of gas turbines and aeronautical devices started in the 1960s. Nevertheless, their development was slow compared to other numerical tools that were employed in different research fields, as the Finite Element Analysis (FEA). This is due to the complexities of the flow evaluation around intricate geometries that requires high computational capabilities. With the introduction of High Performance Computers (HPC) the time required for CFD evaluations decreased, allowing the diffusion of CFD in the industrial environment in the 1990s. Thanks to the increase in the computational capability of the computers, this kind of software are nowadays involved in the design development, whereas in the past they were employed only for the validation of the design.

4.2 Commercial CFD

The increase in the employment of the CFD in the industrial environment is due to the development of the commercial software in the last decades. These programs usually employ a Graphical User Interface (GUI) that makes the interaction with the user, in order to set the flow simulation, simple and intuitive.

Furthermore, the user's manual, that is usually available with these software, allows people with little experience in the field to develop a CFD analysis.

Commercial codes are usually composed by three main modules: Pre-processor, Solver and Post-processor. They are schematically represented in Figure 1.2

.

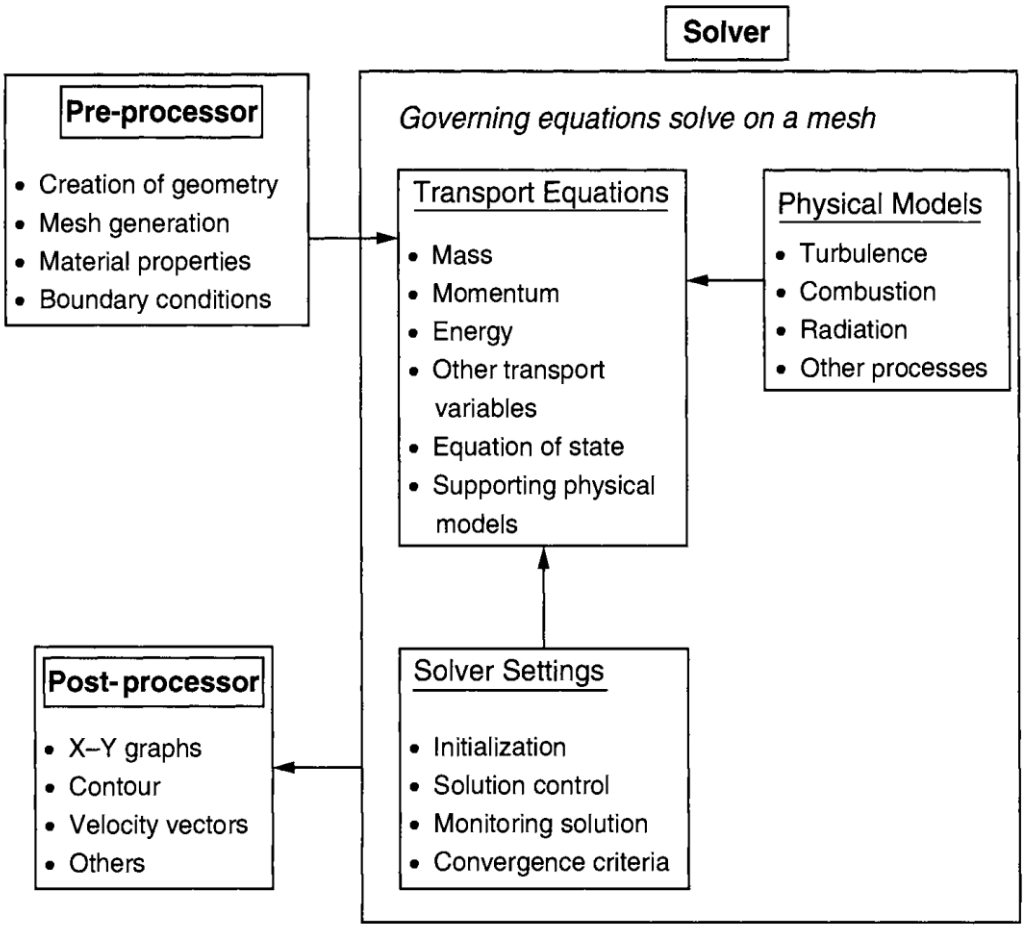

Figure 4.1: Commercial CFD modules tasks. Source [19], p. 33

The Pre-processor sets the conditions necessary in order to perform the flow analysis. First of all, the geometry and the mesh have to be defined and generated; this first step is often executed employing external software, as it is done in this case using ANSYS ICEM CFD. Second, the boundary conditions and the material properties are defined. These characteristics have to be chosen in order to represents the real problem in the most appropriate way. In

Figure 4.2 a flow chart shows the options that are usually available in a commercial CFD pre-processor for the definition of the flow characteristics.

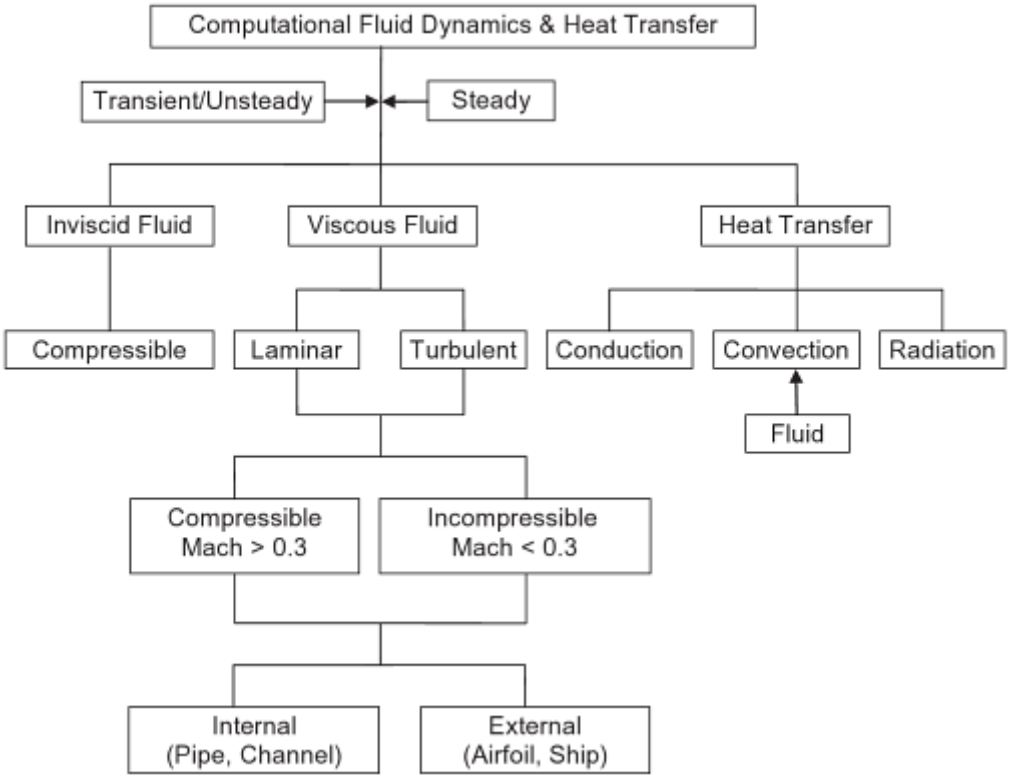

Figure 4.2: Flow chart with the options available for the flow characteristics in CFD. Source [19], p. 39

The Solver is the module of the CFD software that resolves the numerical equations that represent the flow. In this module of the software, the initialization of the solutions, their calculation and the production of the results are executed. Since the settings are defined in the pre-processor while the results are analyzed in the post-processor, the solver is automatic and the user does not interact directly with it.

The Post-processor is the last module of commercial CFD software; its task is to show the results obtained by the solver through the graphical interface, employing different devices as plots, streamlines and contours. In some cases commercial CFD scripts do not include the post-process module and external code have to be employed in order to analyze the results of the flow simulation.

The different modules that compose a commercial CFD code are linked together; they can be grouped in a unique software, as it happens for ANSYS FLUENT, or they can be split in three, as ANSYS CFX that has pre-processor (CFX5PRE), solver (CFX5SOLVE) and post-processor (CFX5POST) separated. In any case the different parts can transfer data and information from a module to the other in a quick and direct way.

The development of a CFD analysis using commercial software follows a procedure that is standardized. First of all, the geometry and the computational domain have to be defined and the mesh generated. Particular attention in this phase must be paid to set the boundaries in order not to be affected by the flow around the airfoil and its development downstream of the geometry. This is necessary for the convergence of the solutions. This first phase is usually executed in the mesh generator.

The second step consists in the importation of the mesh in the pre-processor; there the material properties, the boundary conditions and the turbulence model are defined. Moreover, some other parameters necessary for the calculations are set; these parameters are the discretization method adopted for the equations, the relaxation factors, the scaled residuals targets and other settings to control the convergence of the solutions. This phase is crucial in order to obtain accurate solutions; all the settings must be defined in order to realize the best representation of the physical problem and make the solution converge [20]. Moreover, the initialization of the solutions has to be executed; this step should be done in a clever way imposing values of the parameters that are close to the expected solution. The right choice of the values for the initialization can lead to a quick convergence of the solver; a bad choice, instead, can cause slow convergence or even divergence of the calculations.

The third step of this process consists in the solver's calculation of the solution of the equations that describe the flow; when this task is completed the last phase of the CFD study, the post-process, starts. The post-processor allows the user to analyse the solutions employing graphical devices. More over it is possible to control the validity of the assumption made at the beginning of the

process; the most recurring assumption that needs to be checked during the CFD analysis is the y^+, which is assumed during the creation of the grid.

The flow field study is then completed; the results obtained and already analyzed with the post-processor can be employed for further investigation, can be used for the development of the design process or can be read by the optimizer to carry on the optimization, as it is done in this study.

4.3 Verification and Validation

Once that a CFD evaluation has been completed it is necessary to demonstrate that what has been calculated is a representation of the real problem. It is necessary to prove the reliability and the accuracy of the results obtained. The comparison of the results with the experimental data, obtained from a study that aimed to validate the CFD model, have to be done. Nevertheless, experimental data are not always available, therefore another way to verify the validity of the solution has to be employed, and this check often consists in the comparison of the results with the ones obtained with another code [21]. However this last method is not very reliable since both the codes might contain errors or wrong models.

In order to have a unique and reliable method to check the level of accuracy of a CFD evaluation and measure its capacity to represent the real case studied, the AIAA proposed some guidelines in the AIAA G-077-1998 report [22]. In this report two important definitions have been made, they are reported below:

- *Verification*: it is the process that aims to verify that the model, developed for the flow evaluation, is the representation of the user's conceptual description of the physical problem and of its solutions.
- *Validation*: it is the process that aims to check the level of accuracy of the model for what concerns its ability to represent the real case.

From the definitions above it can be seen that two different concepts are stored in the word "model". The first meaning is the "conceptual model" which is the set of data and equations that are derived from the observation of the real phenomena. Within the CFD analysis, the "conceptual model" is composed by

all the equations (such as the mass, energy and momentum conservation, turbulence model and heat transfer equations) and boundary and initial conditions that define the flow field. The second concept is the “computerized model” that is composed by the algorithms that are implemented in the code in order to find the solution of the equations defined in the “conceptual model”.

Once that the two concepts contained in the word “model” have been explained the understanding of the definitions above becomes easier. Verification is, in fact, the procedure that checks if the “computerized model” is effectively solving the problem defined by the “conceptual model”. Whereas validation is the process that controls that the “computerized model” is simulating the real world phenomena.

4.4 CFD Software Settings

The CFD evaluation in this study has been developed employing ANSYS FLUENT 14.0. This software has been chosen since the case studied is two-dimensional, and ANSYS FLUENT can simulate either 2D or 3D flow fields. Moreover, it is a powerful software that includes advanced solver technologies and is able to calculate solutions with high accuracy. Finally, it can generate a journal file, thanks to the Text User Interface (TUI) [17], that is necessary in order to implement it in the optimization process.

The mesh generated with ANSYS ICEM CFD has been imported into FLUENT, and then it has been possible to start the pre-process. First of all the flow characteristics have to be defined; referring to Figure 4.2, the flow is external and it has been studied as steady, composed by a compressible viscous fluid in turbulent regime. The choice of employing compressible flow has been made in order to describe in an accurate way the physics of the case, since in some regions of the domain the Mach number (Ma) can reach high values; therefore if the flow were modelled as incompressible the solutions could be not precise enough [19]. The flow conditions (angle of attack, Ma, Re, etc.) are not shown due to confidentiality issues, they have been provided by Airbus UK through the GARTEUR report referring to the case A2 [4]. The air has been chosen as fluid material and modelled as ideal gas with constant specific heat, thermal

conductivity and viscosity. The k-ω Shear Stress Transportation (SST) model has been selected as turbulence model taking as reference the previous thesis projects; in particular in [6], through a comparison between the Spalart Allmaras (SA) and the k-ω SST, the better prediction of the flow of the latter one is shown.

The boundary conditions have been defined as follows:

- **Inlet, Far field and Outlet**: Pressure far field, defining Ma, flow direction ambient pressure and total temperature.
- **Airfoil**: Wall, imposing the no-slip condition.
- All the other elements that have been used to realize the mesh and are in the fluid domain have been set as interiors.

The following step has been the setting of the parameters to control the convergence of the solutions. With the aim of improving the convergence and the accuracy of the solutions, coupled velocity-pressure scheme and second order equations for pressure, density, momentum, turbulent kinetic energy, specific dissipation rate and energy have been selected. Furthermore the relaxation factors have been changed with the same aim, following the example provided by [8]; the explicit relaxation factors for momentum and pressure, and the density under-relaxation factor have been reduced from their default values to 0.5.

Monitors on the residuals, c_l, c_d and mass convergence have been set. In particular the target for the residuals has been set to 10^{-6} and the maximum number of iterations chosen for the validation case has been 1000. The results of the flow simulation for the datum configuration are analyzed with the post processor of FLUENT and they are shown in the following section. From Figure 4.3 to Figure 4.6 the residual and the convergence of c_l, c_d and mass flow rate are shown. Due to the confidentiality of the results, the c_l and the c_d values in the y axis of the plots are not displayed.

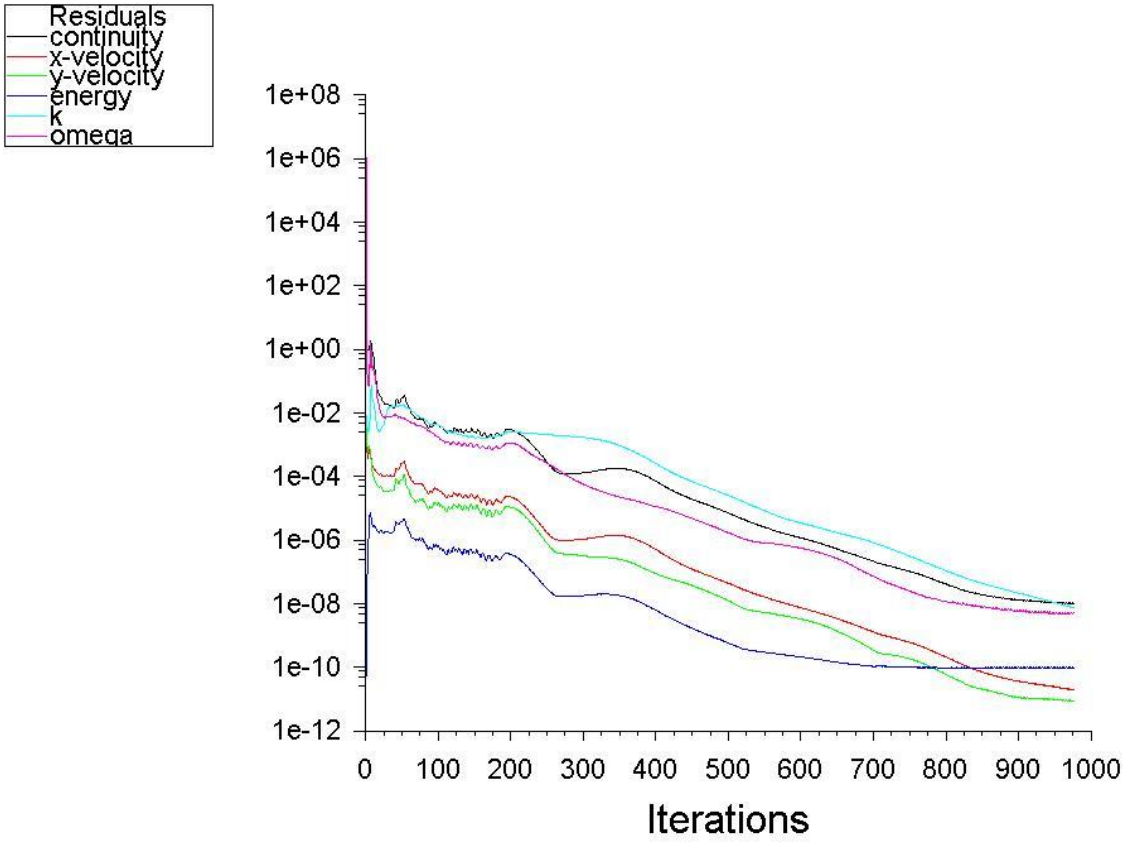

Figure 4.3: Residuals history

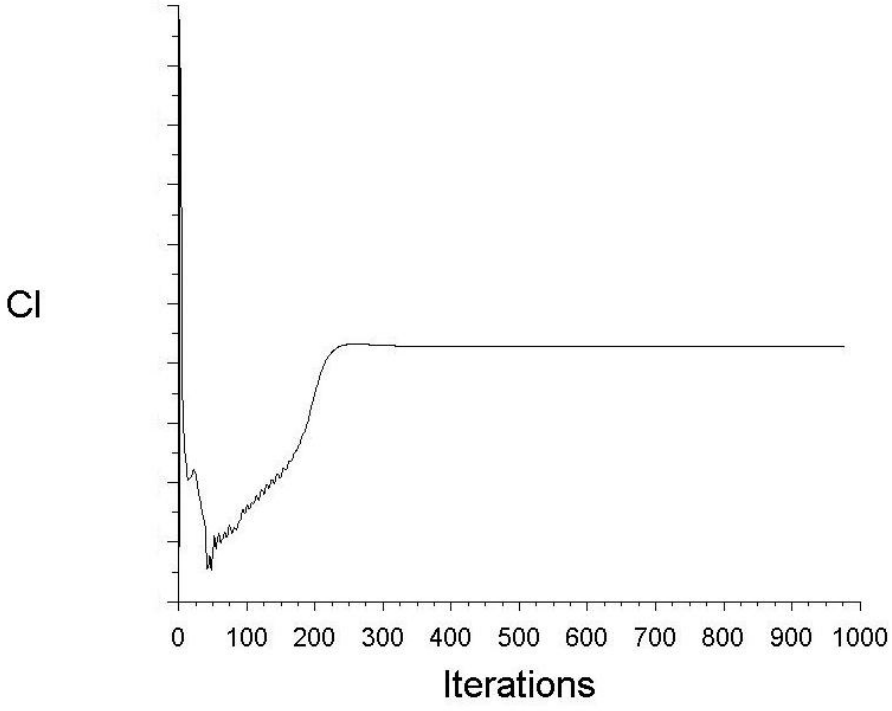

Figure 4.4: cl convergence history

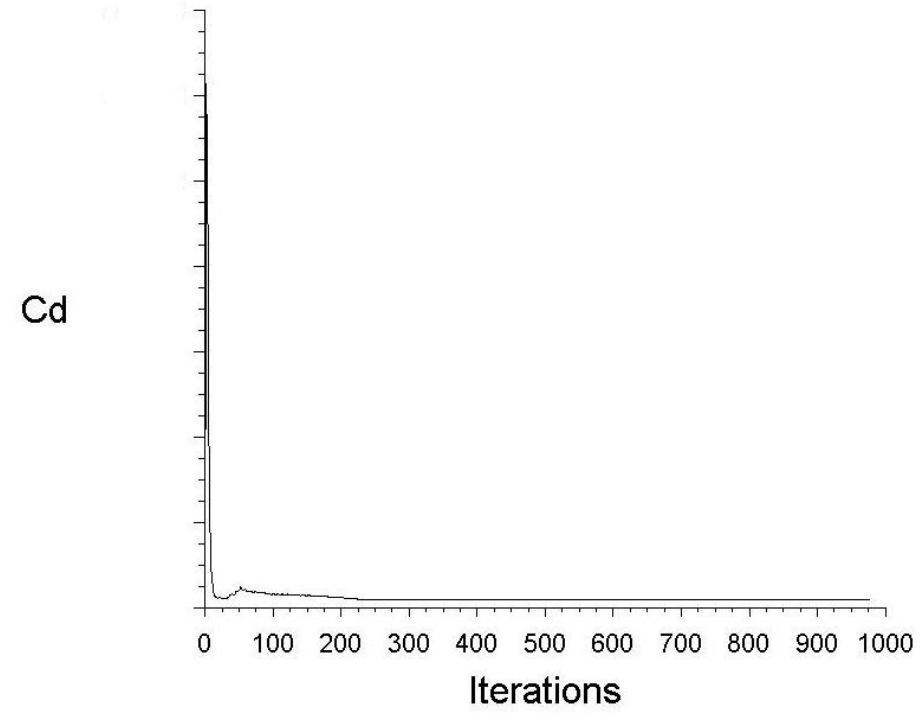

Figure 4.5: cd convergence history

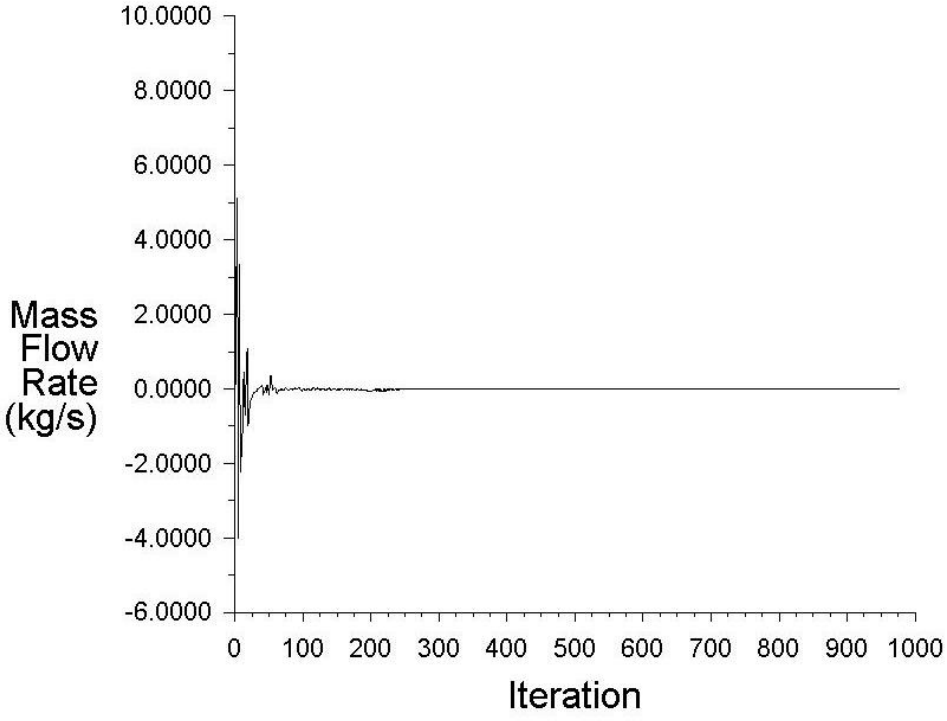

Figure 4.6: Mass flow rate convergence history

4.5 Results and Comparison with the Experimental Data

In this section the results obtained in from the CFD evaluation are presented. From Figure 4.3 it can be noticed that the residuals achieve the target value at around 500 iterations, while the convergence of c_l (Figure 4.4), c_d (Figure 4.5) and mass flow rate (Figure 4.6) are reached between 300 and 400 iterations. Therefore the simulation has been repeated stopping the calculation once the target residual is reached and the time needed to complete this evaluation has been recorded.

In Figure 4.7 it can be observed that the value of the wall y^+ is lower than 1 for all the elements that compose the airfoil. The constraints for this parameter are therefore respected and a good representation of the boundary layer, also in the zone of low Re, can be carried out.

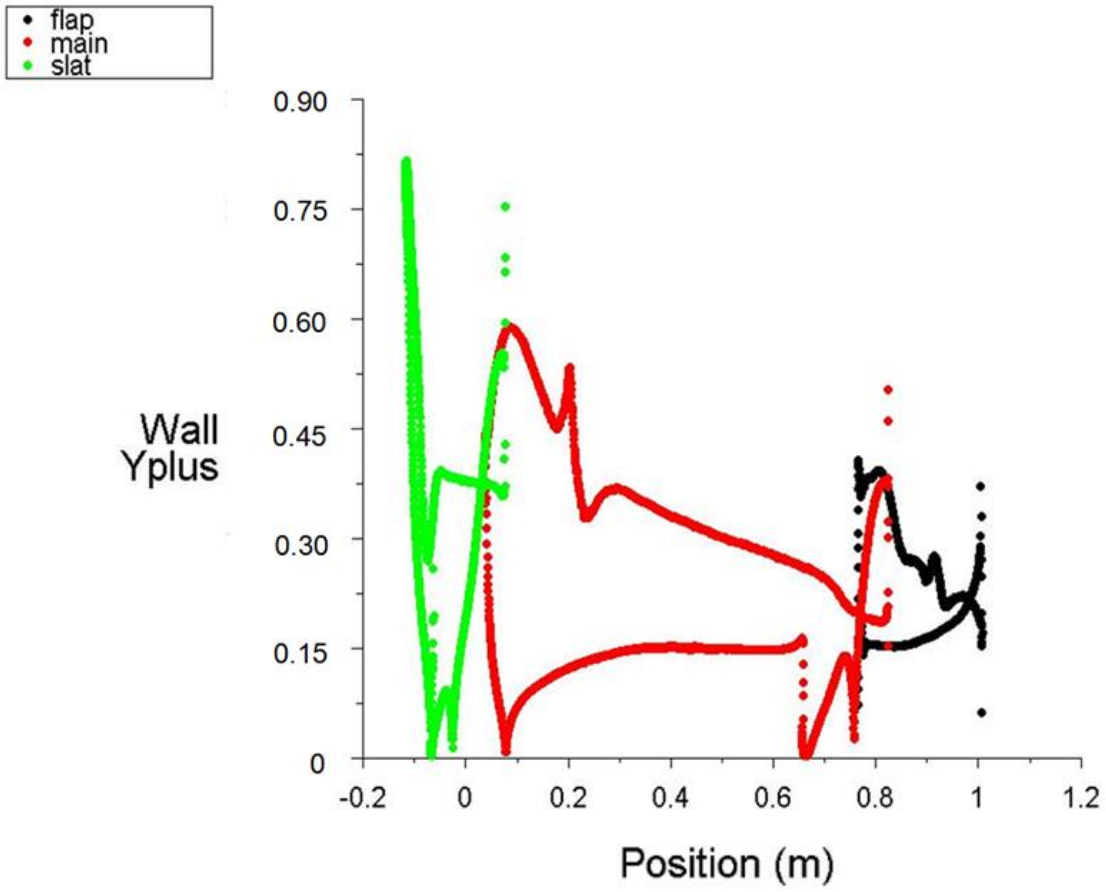

Figure 4.7: y^+ on the airfoil's walls

The outcome of the flow analysis is presented in Table 4.1. The values of c_l and c_d obtained from the CFD and the ones provided by the experimental data are not shown due to the confidentiality of the results; nevertheless the errors of the calculated values compared to the experimental data are shown.

CPU time	Iterations	c_l Error	c_d Error
16 min	500	0.2%	13.2%

Table 4.1: CFD evaluation results

It can be observed that the c_l has been predicted with a very good precision. It matches the value measured in the experiment presented in [4]. The c_d, instead, presents a much bigger error. Some attempts to reduce the error increasing the density of the mesh and changing the distribution of the nodes have been done. However, no improvements or too small advances, compared to the increase in computational time, have been obtained. This error has been compared to the one obtained in the previous theses ([6], [7] and [8]) and in [2], which were all simulating the flow around the same airfoil; it has been noticed that the error of c_d was quite high in all these studies, this is due to two main reasons: first, the experimental data provide two different values of c_d, without giving a satisfactory explanation of their meaning; second, the CFD software are not as precise in the prediction of the drag as they are in the calculation of the lift; thus an error of 10-15% can be considered acceptable for this parameter [19].

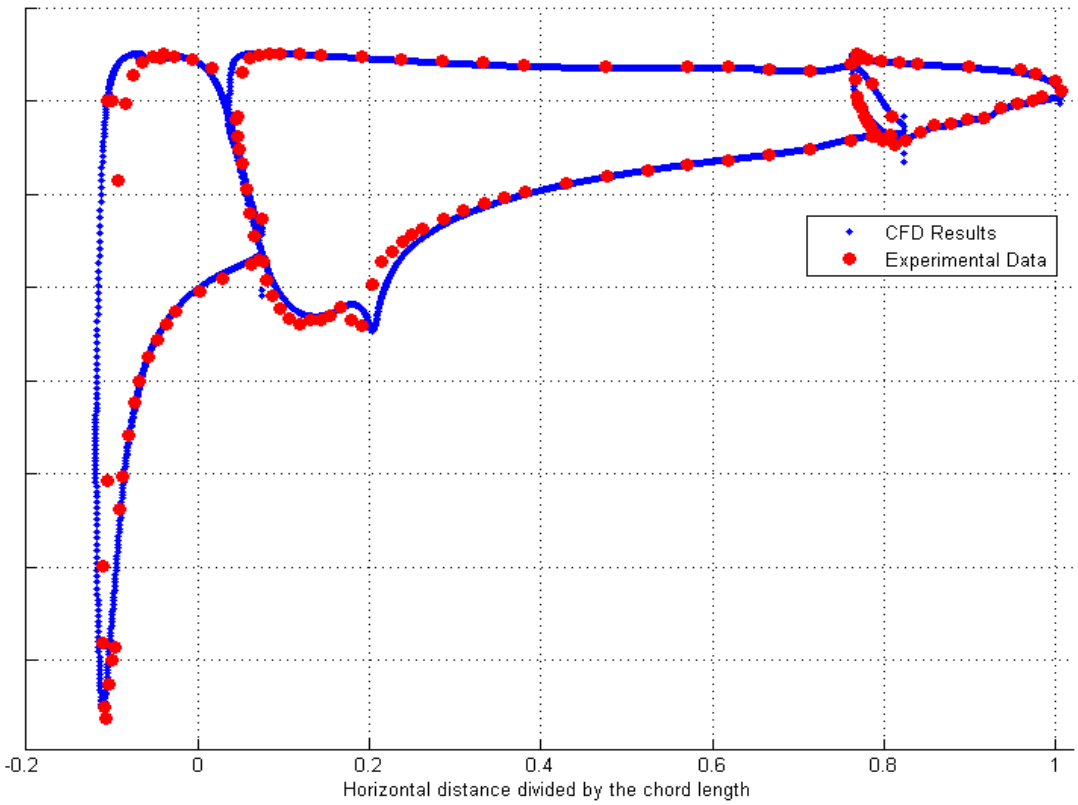

Figure 4.8: Pressure coefficients comparison

In order to proceed in the validation of the CFD analysis the comparison of the pressure coefficient c_p along the elements of the airfoil between CFD results and experimental data has been carried out. This comparison is shown in Figure 4.8; from the image it is clear that the experimental data and the results of the flow simulation have a very good matching. The values of the pressure coefficients on the y axis are again omitted due to the confidentiality nature of the data.

The streamlines, and the velocity and pressure contours around the airfoil are presented from Figure 4.9 to Figure 4.12. From these figures can be seen that the flow field is well represented by the software. Through the picture representing the streamlines (Figure 4.9), besides the stagnation points, the recirculation behind the slat and in the rear part of the main element can be noticed. In the contours images (Figure 4.10, Figure 4.11 and Figure 4.12) the stagnation point, the peak of velocity in the front part of the slat and the wake behind each element can be appreciated.

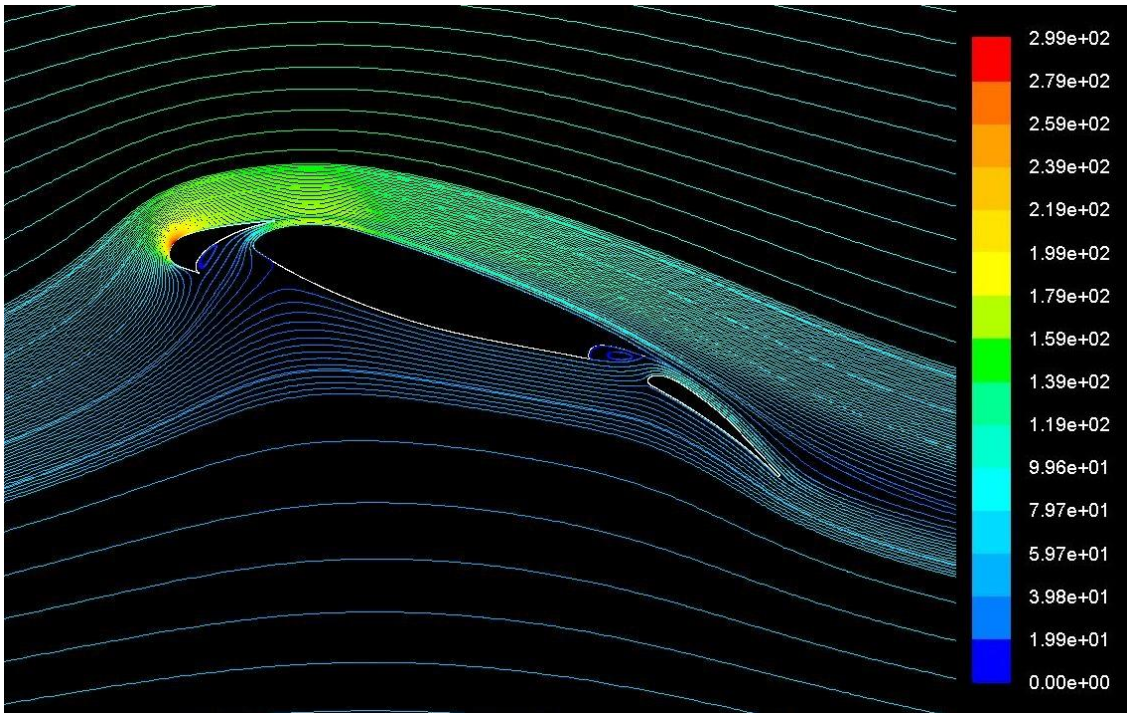

Figure 4.9: Streamlines around the datum geometry (parameter: velocity magnitude, units: m/s)

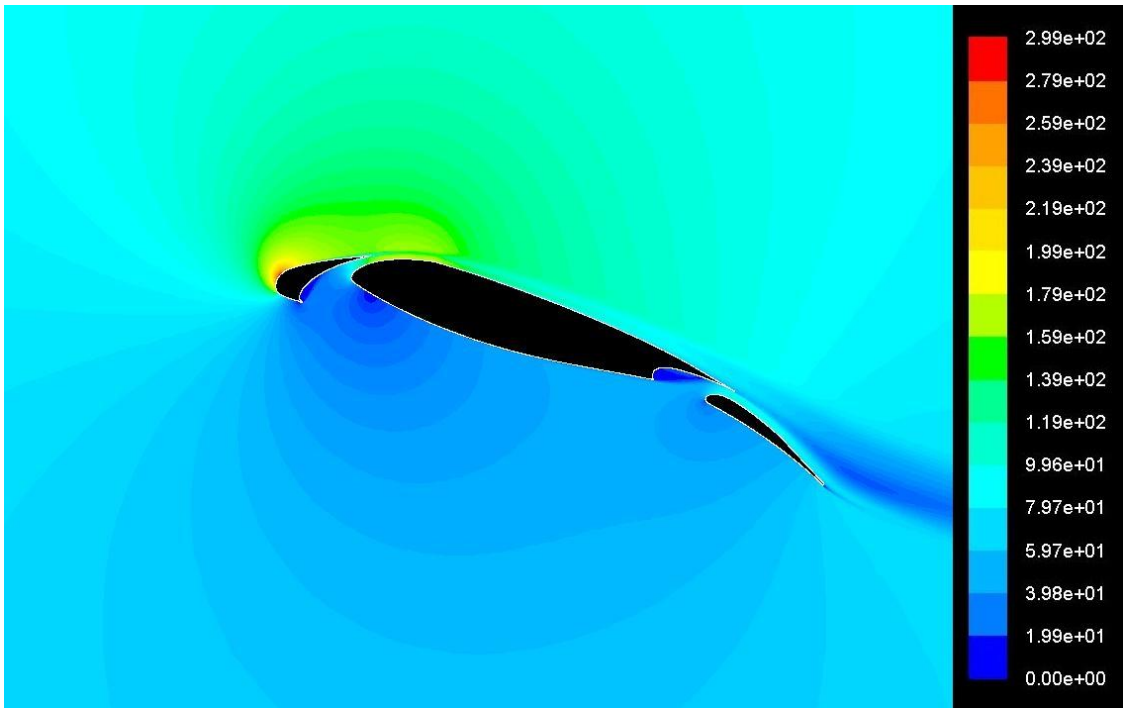

Figure 4.10: Velocity contours around the datum geometry (units: m/s)

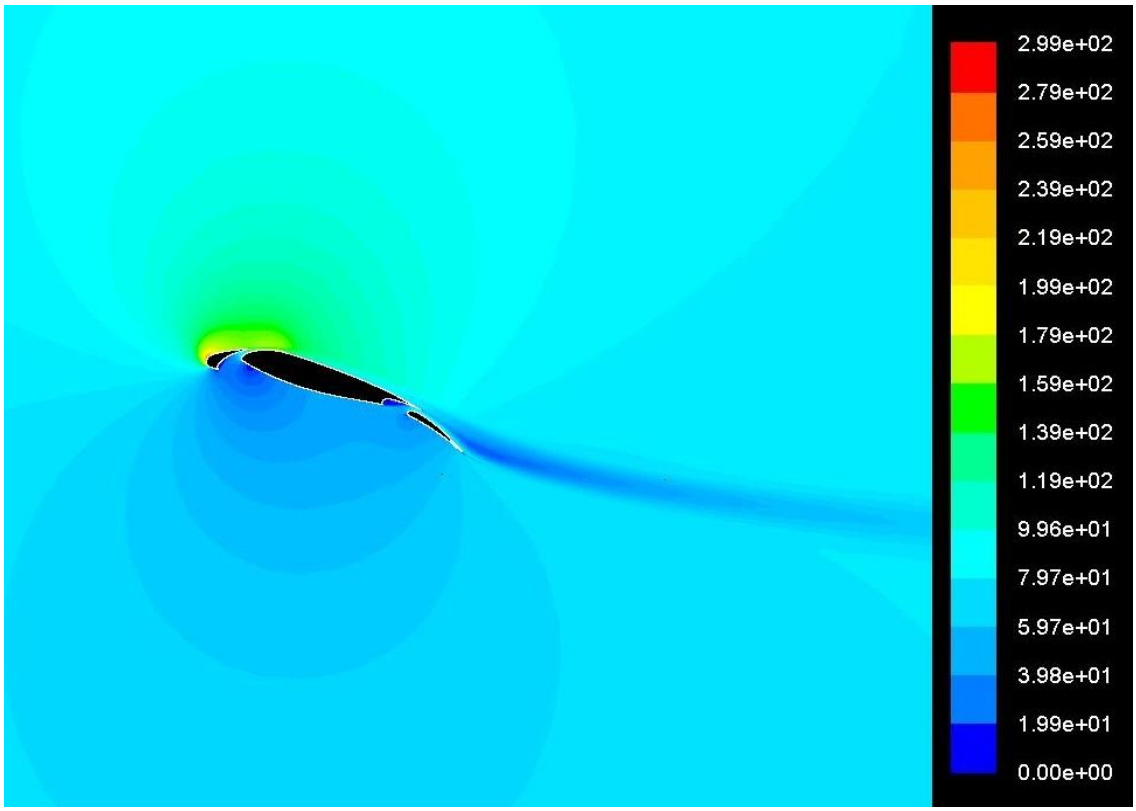

Figure 4.11Velocity contours around datum geometry, zooming out to appreciate the wake (units: m/s)

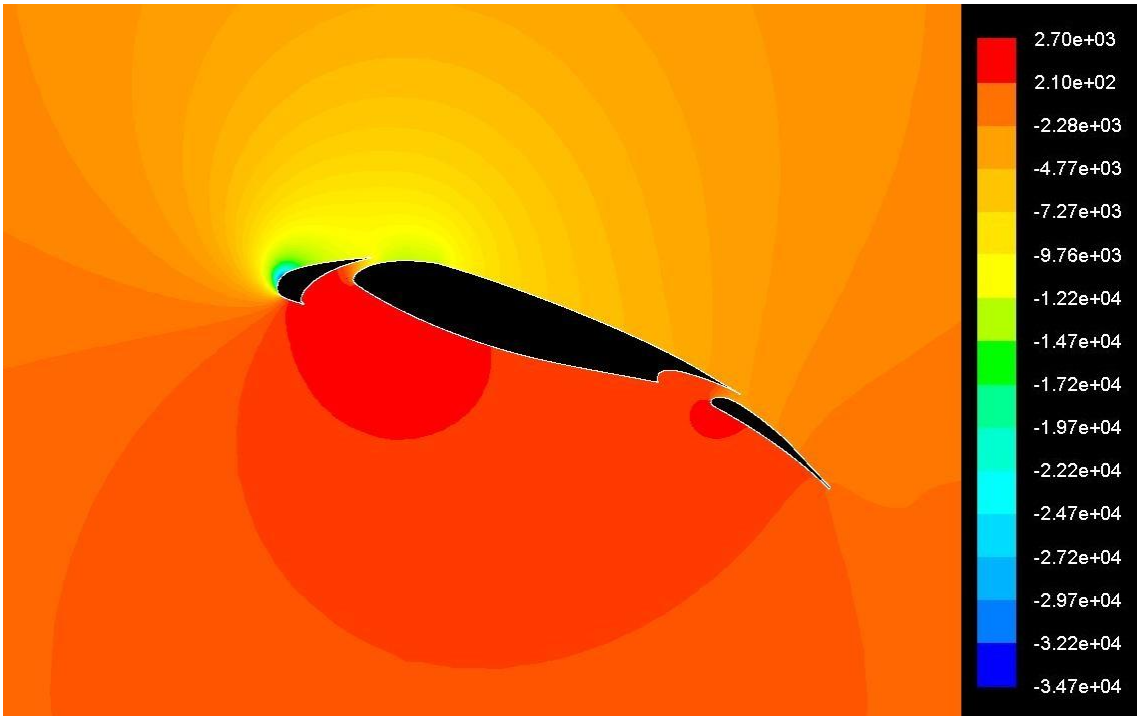

Figure 4.12: Pressure contours around the datum geometry (units: Pa)

4.6 Solver Automation

In order to perform the optimization it is necessary to automate the solver, as it has been done for the mesh generation. It is possible to generate a journal (*.jou*) that allows the simulation of the flow in batch mode. ANSYS FLUENT permits the generation of the journal file recording the procedure followed by the user to set the solver. This feature is similar to the one that ICEM CFD employs in order to create the *.rpl* file. There is only one difference: the commands given using the graphical interface cannot be recorded in the FLUENT journal file, for this reason the TUI has to be employed. The instruction for use this device can be found in [17].

5 MULTI-OBJECTIVE OPTIMIZATION

5.1 Introduction

The numerical optimization process is usually composed by two main parts: there is an engineering code that produces data from the study of a particular shape for the design, and the optimizer that analyses the data produced and develops new configurations to be studied. In order to improve the design, many shapes have to be generated and then analysed. The parameters that can be controlled by the designer in order to modify the shape are named design variables and are included in the design vector. The optimizer, on the basis of the data obtained from the study of a particular shape, proposes new design vectors with the aim of generating improved shapes, this process continues in a loop until convergence is reached.

The idea of employing numerical optimization in order to achieve a good design for aerodynamic elements was proposed for the first time in the late 1970s [21]. At that time the tools available and the computational power of the computers were not enough developed to carry out a numerical optimization of real-life problems. Nevertheless, nowadays important improvements on the tools employed to develop this kind of studies have been made; therefore, numerical optimization can now be applied successfully to real-world cases.

Optimization can be divided in two main categories, on the basis of the number of functions to optimize [23]:

- Single-Objective optimization: it is realized when there is only one function to optimize. The result of this process is the global optimum of the problem.
- Multi-Objective optimization: it is developed when more functions, usually conflicting, have to be optimized. In this case it is not possible to find a global optimum and the solution of the problem is more challenging.

Most of the real-life problems have to be analyzed with a multi-objective method; in fact, many different aspects, which are often contradictory, affect the design of engineering products.

In order to perform an optimization process many different algorithms can be chosen; each of them has advantages and drawbacks. For this reason there is not a method that can be applied to any problem and can be considered superior compared to all the other techniques. However, for each case, the method that best adapts in the research for the optimal solutions can be found [24]. For aerodynamics optimization heuristic methods are preferred to gradient based ones. This is due to the fact that this kind of problems presents many local minima that affect gradient-based codes' research for optimal solutions. There are several methods included in the heuristic (or stochastic) category; the most employed approaches in the engineering field are Genetic Algorithms, Simulated Annealing and Tabu Search. The last one is employed in order to carry out this work.

5.2 Multi-Objective Optimization Background

In order to understand how a multi-objective optimization case is studied, it is necessary to make a step backward observing a single-objective optimization problem, which can be defined in this way:

Minimize $$f(\bar{x}), \qquad \bar{x} \in \mathbb{R}^n \tag{5-1}$$

Subject to the constraints $$\begin{cases} c_i = 0, & i = 1, 2, 3, \dots, m' \\ c_j \geq 0, & j = m' + 1, \dots, m \end{cases} \tag{5-2}$$

$f(\bar{x})$ is the objective function, $\bar{x}$ the design vector, n the number of design variables that compose $\bar{x}$ and c_i and c_j are the constraints that affect $f(\bar{x})$. This problem is solved finding the design vector that minimizes the objective function, respecting the limitations imposed as constraints.

It is common practice in the optimization environment to minimize the objective functions. For this reason, if there is a function $g(x)$ that needs to be maximized it is inserted in the optimization process changing its sign: $-g(x)$. A design vector $\bar{x}_1$ is defined "better", from an optimization point of view, than $\bar{x}_2$ if and

only if $f(\bar{x}_1) \leq f(\bar{x}_2)$. Since only one function is analyzed, there is a unique solution for a single-objective problem [23].

A multi-objective optimization problem can be defined similarly to the single-objective one, with the difference that there are more functions to minimize. Using the same notation as before the representation of the problem is:

Minimize $$\bar{F}(\bar{x}) = \{f_1(\bar{x}), \dots, f_p(\bar{x})\}, \qquad \bar{x} \in \mathbb{R}^n \tag{5-3}$$

Subject to the constraints $$\begin{cases} c_i = 0, & i = 1, 2, 3, \dots, m' \\ c_j \geq 0, & j = m' + 1, \dots, m \end{cases} \tag{5-4}$$

For this kind of problem there is not a unique optimal solution; the multi-objective optimization produce as result a group of solutions called Pareto optimal set. This concept was introduced by Ysidro Edgeworth (cited in [25], p. 2) and afterwards it was generalized by Vilfredo Pareto (cited in [25], p. 2). A solution can be defined as Pareto optimal if there are no other feasible design vectors that can decrease more, at the same time, all the objective functions. The Pareto set is composed by all these solutions, which are also called non-dominated solutions. This concept can be expressed in mathematical language. Considering two design vectors $\bar{x}_1$ and $\bar{x}_2$, it can be asserted that $\bar{x}_1$ dominates $\bar{x}_2$ if all the elements of $\bar{F}(\bar{x}_1)$ are smaller than or equal to the analogue ones of $\bar{F}(\bar{x}_2)$ and no less than one element is strictly smaller than its correspondent. On the other hand, $\bar{x}_1$ and $\bar{x}_2$ are defined Pareto-equivalent if there are some component of $\bar{F}(\bar{x}_1)$ greater and some others smaller than the analogous ones of $\bar{F}(\bar{x}_2)$. All the Pareto-equivalent solutions, which are not dominated by any other solution, compose the Pareto-optimal set or Pareto-front (it can be noticed in Figure 5.1). These solutions represent a limit: no more improvement of the objective functions can be reached without penalizing at least one of them [26].

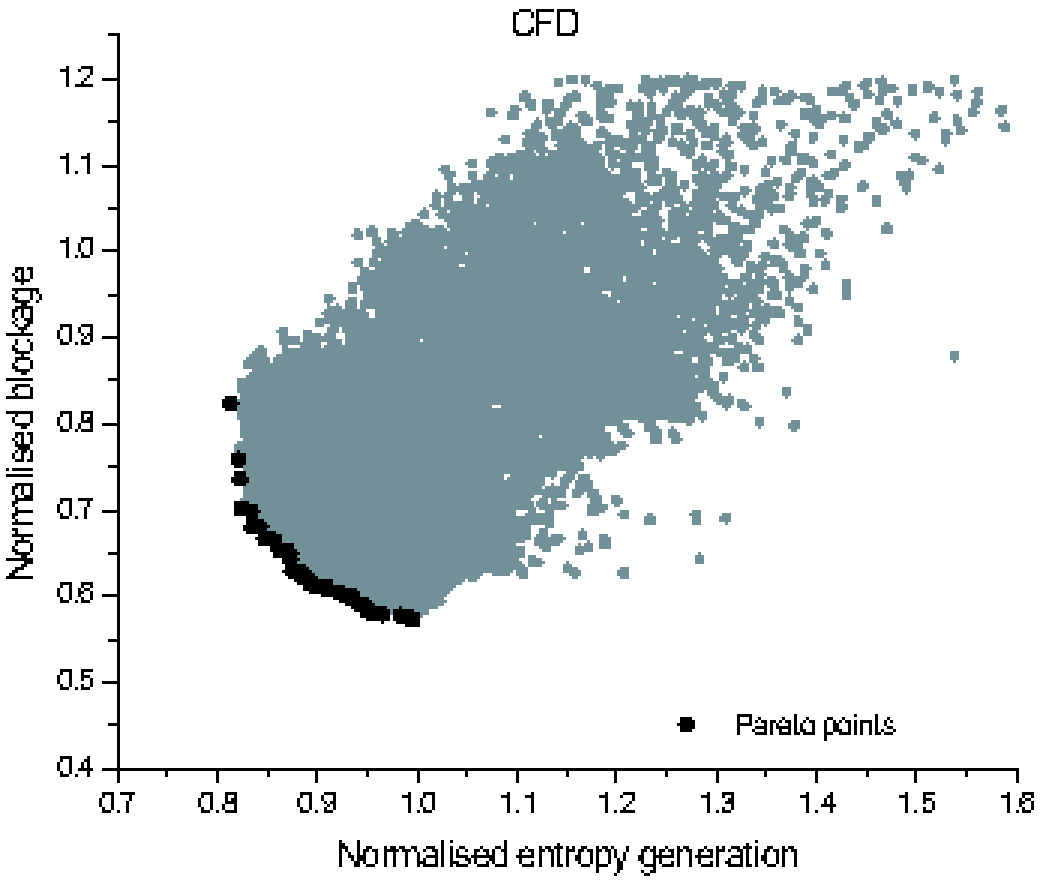

Figure 5.1: Example of Optimization research pattern and Pareto front. Source [27], p. 485.

At the end of the multi-objective study a Pareto front is obtained. All the solutions contained in it are considered equivalent, therefore there is no solution that can be considered better than the others. A Decision Maker (DM) has to be employed to select the final solution within the ones contained in the Pareto front. There are different DM that can be chosen, three main categories are available in literature and presented here:

- A Posteriori: The choice of the solution is made once the optimization process has terminated and the Pareto front has been created. There are some disadvantages related with this method: it is difficult to present the solutions when more than three objective functions are evaluated. Moreover, the DM can struggle with the choice of the solution if there is a large number of design vectors composing the Pareto front.

- A Priori: The choice of the solution is realized combining the different objective functions employing some scalar factors as weights. The multi-objective optimization is, therefore, turned into a single-objective one. This method has the disadvantage that the choice of the weights can lead to unrealistic results since it is not always known how the weights employed are a good representation of the reality.
- Interactive: This third method combines the two previous. The optimization process and the DM are alternated, this technique reduces the drawback of the former two. The creation of the complete Pareto front is not needed and the DM has a higher confidence since it can correct during the process the preferences on the objective functions.

5.3 Heuristic Methods

In this section a brief introduction to the main stochastic methods employed in the engineering optimization process are presented. These techniques are the Simulated Annealign, the Genetic Algorithms and the Multi-Objective Tabu Search (MOTS). The last one is the one employed for this work.

5.3.1 Simulated Annealing

This technique is founded on the Metropolis algorithm and exploits the analogy that can be observed between the optimization research for a minimum and the process that bring to the generation of a crystalline structure of minimum energy during the slow cooling of a metal. A more detailed presentation of this method can be found in [28].

Some advantages are obtained by the application of this method. First, it does not get stuck in the local minima because of its random research method. Second, the codification of the algorithm representing this method is quite easy, even for complex optimization problems. The disadvantage related to this technique, instead, is the need of data management in the case of the generation of a big amount of values.

5.3.2 Genetic Algorithms

The Genetic Algorithms (GA) method takes inspiration from the natural selection: different populations of design vectors are combined together in order to find the ones that are the best for the objective functions. The first attempt of application of this technique can be found in a PhD thesis dated 1967 (cited in [25], p.29) even if the problem analyzed was reduced to a single-objective optimization.

GA begins the optimization process with the creation of a first population characterized by a random selection of design vectors. Afterwards, new generations are realized combining the population of the first one with different operations, as cross-over and mutation. Cross-over consists in the creation of a new design vector combining two “parents” sets of design variables; whereas the generation by mutation takes place when some parameters of only one design vector are modified. The best individuals from every new generation are selected in order to produce new generations with the two operations described above; the process keeps on going until the convergence is achieved.

Many versions of this method have been developed during the years; at the moment, the most popular is the NGSA-II, more information about it can be found in [29].

The main advantage related with the employment of this approach is the wide exploration of the design space. Nevertheless, this technique can get stuck in non-optimal solutions since it moves away from the worse solutions rather than approaching the optimal ones. Moreover, due to its formulations no derivatives are calculated, and exaggerated variations of the design vectors can be achieved.

5.4 Multi-Objective Tabu Search

The MOTS algorithm can be considered the improvement of local search techniques, employing an adaptive memory in order to execute an intelligent exploration of the design space. In this way unnecessary movements are avoided and the optimization can reach the convergence quickly.

Two types of memory are employed by MOTS: explicit and attributive. Explicit memory is used to store the Pareto set of optimal solutions and record new regions that are interesting for the analysis but are still unexplored. Attributive memory is employed for guiding the solutions towards convergence; in fact, it keeps the information about the attributes related to solutions that change moving from one point to another [12].

The core of the local search technique adopted by MOTS consists of the Hooke and Jeeves (H&J) algorithm. This algorithm follows the same pattern at each iteration:

- It creates $2n$ new points (where n is the number of design variables), using a preselected value, called initial step (δ_i), in order to increment ($x_i + \delta_i$) or reduce ($x_i - \delta_i$) each design variable.
- For each of the $2n$ points created the values assumed by the objective functions are calculated.
- From this set of points a selection is made in order to keep the best one as starting point for the next iteration. The points that violate the constraints and the ones that are Tabu are discarded before the selection.

A classification of the points is needed in order to perform the selection presented above. There are three categories in which the points can be divided: A, B and C. If all the objective functions evaluated in the new point are decreased in comparison with the previous one, this point belongs to category A; if some objectives decrease whereas some others increase it is part of category B; if all the objective functions increment it is a point of category C.

The points of type A are non-dominated points; they are the ones that are aimed for the optimization. For this reason, if after the Hooke and Jeeves movement only one point belonging to the category A is obtained it is automatically selected as best point. Otherwise, if more points of class A are found, a random selection is executed with the aim of choosing one of them; the remaining points of type A are stored in an Intensification memory in order to be analyzed later. If no points of type A are found, the same process is applied to

the ones classified in B. If there are no point of type A and B, one of class C is selected through a random process.

Connor and Tilley (cited in [26], p.1199) developed a movement procedure that aims to improve the simple H&J move. This process is executed following the path presented below:

- The first Hooke and Jeeves movement is executed and then repeated.
- If the new solution found is non-dominated it is stored as current optimum point.
- If the point obtained it is not dominating, it is discarded and the method comes back to the previous point and executes a normal H&J movement.

MOTS method executes three main kinds of search; each of them is related to a different type of memory. The points that have been visited recently are stored in the STM (Short Term Memory); these points are called also Tabu because it is forbidden to visit them again. This is what gives the name to this optimization method.

Medium Term Memory (MTM) records the points that are in the optimal or in the near-optimal region. The algorithm executes a Search Intensification (SI) around these points, since they guarantee good results of the objective functions.

The third kind of memory is the Long Term Memory (LTM); the regions that have already been widely explored are saved in this type of memory. Search Diversification (SD) is associated with this memory. This particular search method consists in the investigation of the areas that are still unexplored.

In the end, another action can be executed by MOTS. It is named Step Size Reduction (SSR) and it is executed when no successful points are found during the last iterations. The step (δ_i) is reduced in order to investigate intensively the regions around the current Pareto front. This search is executed either in the proximity of the current optima points or in a region around a random point chosen from the MTM. Once the SSR search terminates the counters that checks the different search strategies is reset.

The different memories and search methods are summarized and presented in Figure 5.2.

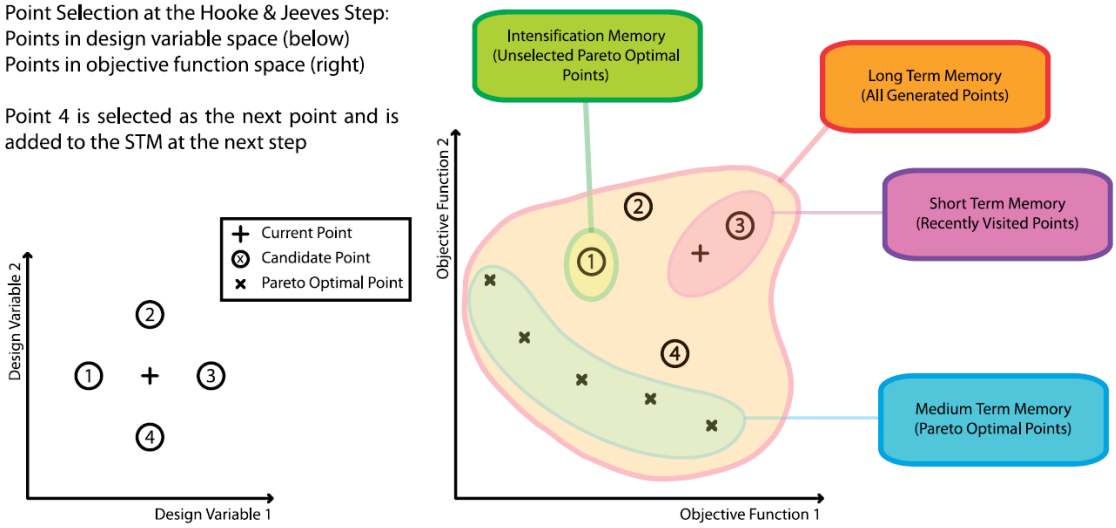

Figure 5.2: Selection of the points for the H&J move and MOTS different memories explanation. Source[26], p.1199

5.4.1 Software Description

The MOTS algorithm employed in this work is an adaptation, made by Jaeggi et al. [26], of the code created by Connor and Tilley (cited in [26], p.1195) in order to perform single objective Tabu Search. This code has been checked and improved by Kipouros through the work presented in [12].

The flow chart representing the code is presented in Figure 5.3. From this image it can be noticed that there is a local counter called i_{local} that counts the iterations and it is reset for each update of the Medium Term Memory. Each of the different research methods presented in the previous section (SI, SD and SSR) is associated with a predefined value of i_{local}; therefore, once i_{local} reaches one of these predetermined values, the related research method is executed starting from a point chosen through a random process from the related kind of memory (MTM for ID and SSR, LTM for SD).

The optimization process stops when one of the convergence criteria is reached; these criteria can be defined as maximum number of iterations or as

maximum time available for the optimization. The Pareto front is determined once the process terminates; nevertheless, it is possible to visualize the temporary version of the Pareto set and stop the optimization at any moment.

The code contains an algorithm in order to manage the constraints, which can be grouped into two main categories:

- Binary Constraint: this type of constraints divides the geometries in feasible and unfeasible. If a geometry is classified as unfeasible it is stored in the Tabu memory and it will not be visited again. Moreover, no information about the constraint violation is recorded.
- Penalty Functions: this class records information about the constraint violation; this information is used in order to guide the research towards a feasible region. Employing this class of constraints it is possible to have a more flexible optimizer and explore the entire design space [26].

In the end, another interesting characteristic of the MOTS software is its parallelization. In fact, the code is divided in one Master and some Slaves; the Master code contains the core of the optimization algorithm, it selects the design vectors that have to be analyzed and sends them to the Slaves; on the other hand, the Slaves analyze the design vectors received from the Master evaluating the objective functions. Several Slaves work in parallel allowing the study of many design vectors at the same time. This parallelization is called Functional Decomposition and it is created in order to make faster the process since the CFD simulations are the most time consuming part of the optimization; therefore the execution of many of them in parallel can speed up the whole process. The parallelization can be executed also in the CFD software, it is called Domain Decomposition; in this case the fluid domain is divided in small regions that are analysed by different processors in parallel, making the CFD evaluation faster. The Functional Decomposition of the optimizer can be coupled with the Domain Decomposition; in this way also the time required by the flow evaluation can be shortened even more. In this study both the parallelizations have been applied in order to reduce as much as possible the time needed to execute the optimization.

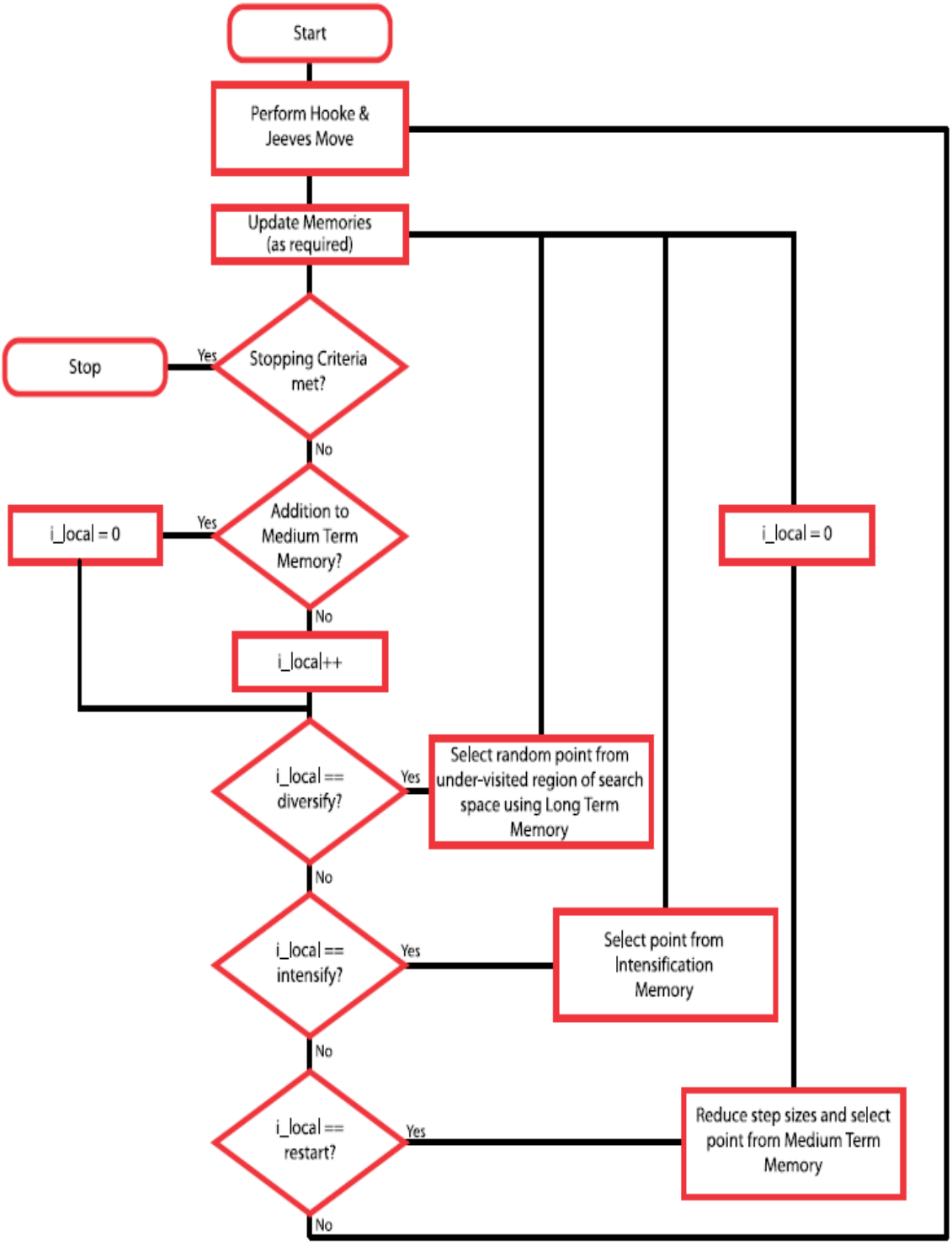

Figure 5.3: Flow chart representing the MOTS algorithm. Source [26], p. 1196

5.5 Robust Optimization

In the previous sections the concept of optimization and different methods employed to execute it have been described. The aim of the researches conducted with the tools presented before is to minimize (or maximize) the objective defined by the user, finding the absolute minimum (or maximum). In the last decade, thanks to some interesting studies, it has been observed that the optimal solutions found after an optimization process can have high sensitivity to small variations of the operating conditions. This sensitivity can lead to a large degradation of the performance, represented by the objective functions, even for small variations in the conditions. For this reason, nowadays many researches focus on finding robustly optimized designs, which are less sensitive to the input variations [30]. The robust optimum that is aimed does not have to be the global optimum; but has to keep almost constant the value of the objective function through small variations of the conditions. In Figure 5.4 a comparison between a robust design point (on the right) and a non-robust one (on the left) is shown.

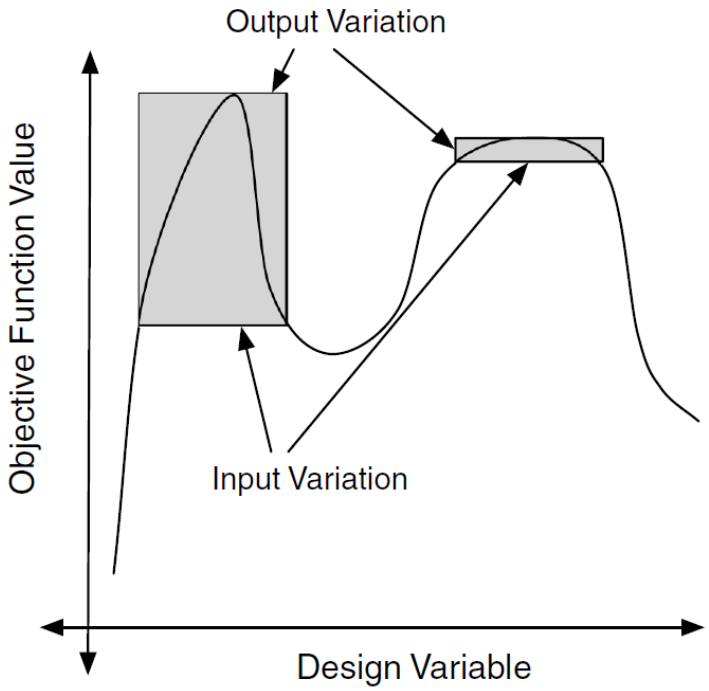

Figure 5.4: Comparison between a non-robust design point (on the left) and a robust one (on the right). Source[30], p. 639

In order to execute a robust optimization, different techniques can be employed; some methods aim to quantify the uncertainties using stochastic methods, whereas some others evaluates the performance employing the interval analysis method to reproduce the change in the operating conditions.

Many techniques belong to the first category; some examples are Monte Carlo Simulation (MCS), Polynomial Chaos (PC) and σ-distribution. However, the procedure followed is similar for each of them:

- The uncertainties are expressed as random variables in terms of mean value and variance.
- The algorithm produces some deterministic values of these variables that are given as input for the engineering simulations, which can be CFD evaluations for example
- The deterministic outputs of the simulations are again treated by the algorithm in order to give back the objective functions in terms of mean value and variance.

Some studies were developed in order to find out which method performs better the uncertainty quantification. A research executed in 2002 [31] recognized the MCS as the most accurate. Nevertheless, its computational cost is too high in order to perform complex CFD evaluations; therefore other methods that can guarantee an acceptable accuracy with lower computational cost are preferred. PC is a valid substitute of MCS, as demonstrated in [30], and, in the last years, several researches have been developed in order to perform robust optimization with this approach. More information about stochastic models applied to the uncertainties quantification can be found in literature; it is possible to find some examples of comparisons between different models ([31] and [32]) and their application for robust optimization [30].

A different method to perform robust optimization is the Multi-Point approach. The theory behind it is quite simple and no stochastic methods are applied: during the optimization for each analysed design, several different evaluations are executed applying discrete variations on the operating conditions, instead of performing only one analysis at the nominal state (as it is usually done for non

robust optimization). Once that the design has been analysed, the results from the evaluations are combined together in the objective functions in two different ways: in one objective function the results are summed together in order to find the optimum designs, in the other function the differences between the performance of the design at the nominal operating conditions and at the modified ones are done with the aim of finding the robust designs. Some examples of the applications of this method are represented by [2] and [33]. In [2] multi-point optimization has been carried out on a three-element airfoil; the angle of attack has been considered uncertain and for each design the CFD analysis has been executed for three different angles of attack: α (the nominal one), $\alpha - 1°$ and $\alpha + 1°$. Two objective functions have been created: the first one considers the sum of the c_l obtained from the three CFD evaluations, whereas the second one contains the difference of the c_l obtained with the nominal angle of attack and the ones obtained with the modified ones. The first objective function looks for the design that allows the maximum c_l while the second one searches the designs whose performance are less sensitive to the variations of the angle of attack.

In this work Multi-point optimization has been executed in order to find a robust design. The uncertainties considered are the angle of attack and the deflection of the flap. In order to perform this optimization the work presented in [2] has been taken as example.

5.6 Objective Functions

The aim of this work is to find the configurations of the multi-element airfoil that reach high value of the c_l and that have a low sensitivity on the variations of the operating conditions. As mentioned before the operating conditions that are considered variable are the angle of attack and the deflection of the flap. Using multi-point optimization it has been decided to discretize the uncertainties in this way:

- In order to reproduce the uncertainty of the angle of attack α, three angles of attack have been considered: $\alpha - 1°$, α, and $\alpha + 1°$

- In order to represent the variation of the deflection of the flap Θ three deflections have been considered: $\Theta - 1°$, Θ and $\Theta + 1°$.

Considering all the combinations between these two discretizations nine cases are obtained. This means that for each configuration of the multi-element airfoil nine CFD evaluations should be executed.

It has been decided to carry out two different optimization studies. The first one executes just three CFD evaluations for each configuration; whereas the second one performs all the nine CFD simulations, considering all the possible combinations of angle of attack and deflection of the flap, for each design vector. This choice has been made in order to compare the two approaches and see if the approach with three CFD evaluations besides the gain in time can guarantee reliable results that are comparable to the ones obtained with nine evaluations.

In order to perform the study with three CFD evaluations, a brief analysis of the performance of the airfoil has been executed, leading to the decision of simulating the flow for the nominal operating conditions and for the conditions that are supposed to cause the largest variations of the performance. They are resumed in Table 5.1.

CFD evaluation 1	CFD evaluation 2	CFD evaluation 3
$\boldsymbol{\alpha - 1°}$ & $\boldsymbol{\Theta + 1°}$	α & Θ	$\alpha + 1°$ & $\Theta - 1°$

Table 5.1: Operating condition for the optimization with three CFD evaluations for each configuration

The objective functions employed for the first optimization process, the one with three CFD evaluations for each configuration, are presented below:

$$\mathrm{obj}_1 = -\frac{c_{l_{\alpha-1°,\Theta+1°}} + c_{l_{\alpha,\Theta}} + c_{l_{\alpha+1°,\Theta-1°}}}{\left(c_{l_{\alpha-1°,\Theta+1°}} + c_{l_{\alpha,\Theta}} + c_{l_{\alpha+1°,\Theta-1°}}\right)_0} + P \qquad \textbf{(5-5)}$$

$$\mathrm{obj}_2 = \frac{\left|c_{l_{\alpha-1°,\Theta+1°}} - c_{l_{\alpha,\Theta}}\right| + \left|c_{l_{\alpha+1°,\Theta-1°}} - c_{l_{\alpha,\Theta}}\right|}{\left(\left|c_{l_{\alpha-1°,\Theta+1°}} - c_{l_{\alpha,\Theta}}\right| + \left|c_{l_{\alpha+1°,\Theta-1°}} - c_{l_{\alpha,\Theta}}\right|\right)_0} + P \quad \textbf{(5-6)}$$

$$\mathrm{P} = max\left[0, \frac{1}{2}\left(\frac{c_{d_{\alpha-1°,\Theta+1°}} + c_{d_{\alpha,\Theta}} + c_{d_{\alpha+1°,\Theta-1°}}}{\left(c_{d_{\alpha-1°,\Theta+1°}} + c_{d_{\alpha,\Theta}} + c_{d_{\alpha+1°,\Theta-1°}}\right)_0} - 1\right)\right] \quad \textbf{(5-7)}$$

In the optimization study developed executing nine CFD evaluations for each configuration of the airfoil, instead, the following objective functions have been used. Since these equations are long they are cut in the representation below; the entire formulas are shown in Appendix A.

$$\mathrm{obj}_1 = -\frac{c_{l_{\alpha-1°,\Theta+1°}} + c_{l_{\alpha-1°,\Theta}} + c_{l_{\alpha-1°,\Theta-1°}} + \cdots + c_{l_{\alpha+1°,\Theta-1°}}}{\left(c_{l_{\alpha-1°,\Theta+1°}} + c_{l_{\alpha-1°,\Theta}} + c_{l_{\alpha-1°,\Theta-1°}} + \cdots + c_{l_{\alpha+1°,\Theta-1°}}\right)_0} + P \quad \textbf{(5-8)}$$

$$\mathrm{obj}_2 = \frac{\left|c_{l_{\alpha-1°,\Theta+1°}} - c_{l_{\alpha,\Theta}}\right| + \left|c_{l_{\alpha-1°,\Theta}} - c_{l_{\alpha,\Theta}}\right| + \cdots + \left|c_{l_{\alpha+1°,\Theta-1°}} - c_{l_{\alpha,\Theta}}\right|}{\left(\left|c_{l_{\alpha-1°,\Theta+1°}} - c_{l_{\alpha,\Theta}}\right| + \left|c_{l_{\alpha-1°,\Theta}} - c_{l_{\alpha,\Theta}}\right| + .. + \left|c_{l_{\alpha+1°,\Theta-1°}} - c_{l_{\alpha,\Theta}}\right|\right)_0} + P \quad \textbf{(5-9)}$$

$$\mathrm{P} = max\left[0, \frac{1}{2}\left(\frac{c_{d_{\alpha-1°,\Theta+1°}} + c_{d_{\alpha-1°,\Theta}} + \cdots + c_{d_{\alpha+1°,\Theta-1°}}}{\left(c_{d_{\alpha-1°,\Theta+1°}} + c_{d_{\alpha-1°,\Theta}} + \cdots + c_{d_{\alpha+1°,\Theta-1°}}\right)_0} - 1\right)\right] \quad \textbf{(5-10)}$$

For both the cases the first objective function sums the c_l obtained at different operating conditions (three values in the first case, nine in the second) and divides this sum by the one calculated for the datum configuration. This function looks for the design vectors that allows the increase in the performance of the airfoil, in particular it search the designs that increment the c_l of the Test Case A. Since the aim of the research is to maximize the c_l, but the optimizer minimizes the objective functions the negative sign is employed in the "*Objective Function 1*". The second objective function instead is the one that looks for robust designs; it sums the absolute values of the differences in c_l between the nominal and the varied operating conditions, and divides this sum

by the same operation executed on the datum configuration. The aim of this function is to find the designs that reduce these differences and guarantee performance less sensitive to the variations on the operating conditions. A penalty function P has been employed, for both the objective functions, in order to take into account also the drag in the optimization; this function penalizes the configurations that are affected by an increase in the c_d compared to the datum configuration.

The two objective functions and the penalty function (P) have a similar formulation for both the optimization studies performed. The only difference consists in the number of elements that compose them.

6 OPTIMIZATION PROCESS IMPLEMENTATION

This Chapter aims to describe how the whole optimization process has been assembled. In the previous sections all the tools and the steps of the study have been presented; in this part the links between the different parts and the technical issues employed to make all the process automatic are shown.

6.1 Optimization Cycle

The algorithm employed for the optimization cycle has been presented in the previous chapter. The whole process is implemented in C++ language. In Figure 6.1 a flow chart shows how it works.

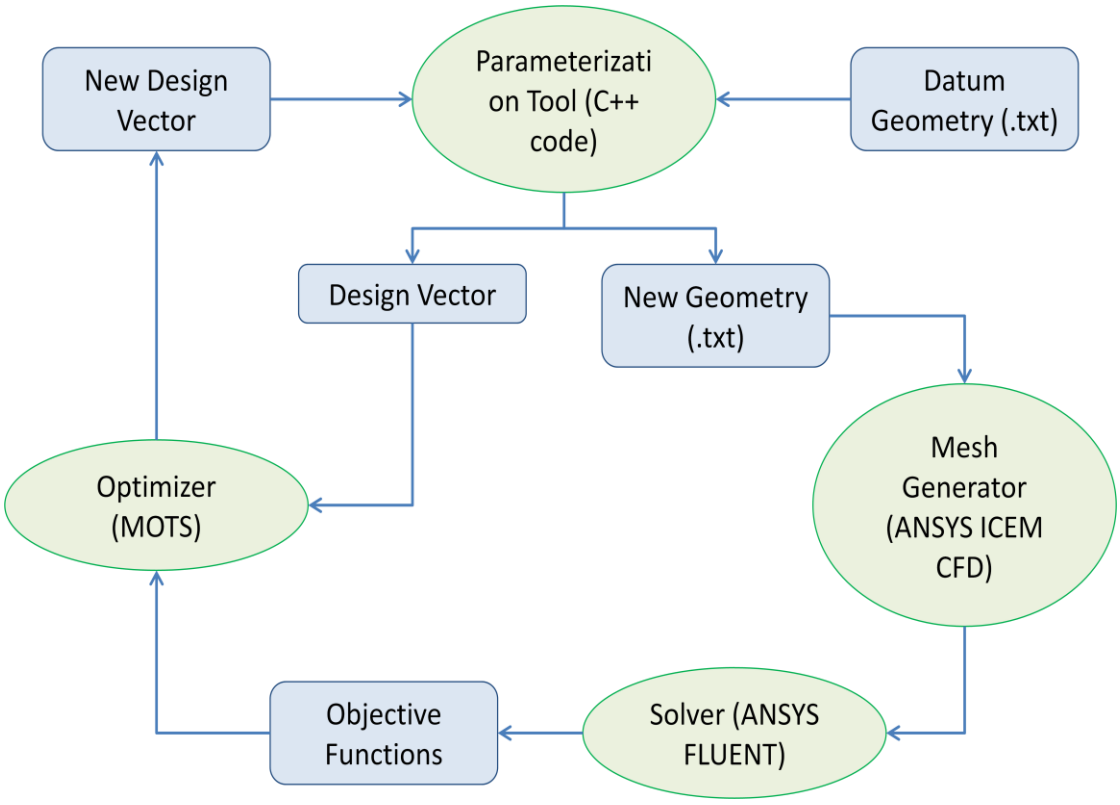

Figure 6.1: Schematic representation of the optimization process

The optimization process can be summarized with a brief description of the flow chart: giving as input to the parameterization tool the datum geometry and the new design vector, the new geometry is created; the mesh is built around it and the CFD evaluation is performed; the results obtained from the CFD solver are combined together in the objective functions that are read, together with the

design vector, by the optimizer. The optimizer, with the information obtained from the inputs, generates a new design vector to start the cycle again. This loop is repeated until convergence is reached.

It is necessary to describe the cycle step by step in order to have a better understanding of it. The process starts with the parameterization tool reading the datum geometry and the design vector files and creating the new configuration for the airfoil, as explained in Chapter 2. This tool gives back two outputs; the first one is a text file (*.txt*) the set of points defining the new geometry, whereas the second one is the design vector that has been given as input before. The new geometry file is then checked using the tool developed in C++ and described in Chapter 2. If the new geometry presents intersections between the different elements, the algorithm that performs the control gives back a flag with the value 1; this value of the flag makes the code call the function InvalidGeometryException. This function records the design vector in the Tabu memory, preventing the solver to visit it again; moreover a message is printed in a text file in order to inform the user that an overlapping has been detected. When InvalidGeometryException is called the current design vector is discarded, the analysis stopped and a new design vector is selected to continue the optimization process. If the geometry is valid the flag assumes the value 0 and the process can go on.

Once the new geometry has been created and checked, the ANSYS ICEM CFD journal file (*.rpl,*) used for the mesh generation, is generated. This file is created for every design vector employing a function called *script_writer*, this code substitute the datum geometry with the new geometry in the journal file. Therefore, the mesh around the new geometry can be built following the same procedure described in Chapter 3. Using the command *system()*, ANSYS ICEM CFD is then run in batch mode (without using the GUI). Once the mesh is created, it is saved in a *.msh* file. If the geometry presents some defects that are not detected in the previous check or the elements are too close each other some errors can occur during the mesh generation. In this case the grid is not built and the file *.msh* is not created. For this reason a second control is

performed at this point. The code checks if the *.msh* file has been generated, if it finds it the process can continue; if the mesh is not found the InvalidGeometryException function is called stopping the analysis and passing to the evaluation of the following design vector. In this case a text file is produced informing the user that the mesh generation failed.

After the mesh generation, the CFD evaluation using ANSYS FLUENT is realized. The ANSYS FLUENT journal (*.jou*) has to be created employing the *script_writer* function in order to give as input the proper angle of attack and the proper mesh file, built around the configuration analysed, to the solver. ANSYS FLUENT is then executed in batch mode employing again the command *system()*. Once the solver completed the analysis two checks are made. First the results file of the CFD simulation is searched in order to find out if the solver has been successful; then if the code found the file, the convergence of the solutions is checked calculating the variance of the c_l in the last 100 iterations, and controlling that this value is lower than 10^{-5}. If both the checks are positive the process can continue; otherwise, if one of them gives a negative response, the InvalidGeometryException function is called and a text file, which informs the user that a failure occurs in the CFD evaluation, is created.

The results obtained from the solver (c_l and c_d) are combined in the objective functions (shown in Chapter 5). The design vector, together with the associated values of the objective functions is then sent to the Master code that records it and produces a new design vector. The new design vector created by the Master undergoes to the same process, and this procedure is repeated until the stopping criterion is reached.

The description of the process has been presented. As mentioned before in order to perform the multi-point optimization several CFD simulations have to be performed for a single design vector. Therefore in the following paragraphs the method employed in order to perform the multiple CFD evaluations in the two different processes developed is presented.

6.1.1 First Study: Three CFD Evaluations for Each Design Vector

In order to perform the study with three CFD evaluations it is necessary to build the mesh three times, since the geometry changes due to the modifications in the deflection of the flap. For each mesh generated one simulation is run. The combinations of flap deflections and angles of attack analyzed are presented in Table 5.1.

The Slave code includes a loop that repeats parameterization, mesh generation and CFD evaluations three times, reproducing the three conditions presented in Table 5.1. Once the loop terminates the three values obtained for the c_l and the three obtained for the c_d are combined in the objective functions ((5-5), (5-6) and (5-7)) and the optimization proceeds as presented before.

6.1.2 Second Study: Nine CFD Evaluations for Each Design Vector

The study developed performing nine CFD evaluations, considering all the combinations of angles of attack and deflections of the flap, is a little bit more complex than the previous one. In this case three grids are generated and for each grid three CFD simulations are run.

In the code two loops are employed in order to reproduce this feature. The first loop repeats three times parameterization and mesh generation, replicating the three flap deflections. The second loop is inside the first one and repeats three times the CFD evaluation changing the angle of attack at each iteration. In this way for each deflection of the flap the three CFD evaluations with the three angles of attack are realized. Once both the loops concluded their job, nine values of c_l and nine of c_d are obtained. These parameters are employed in order to evaluate the objective functions (5-8), (5-9) and (5-10). Once the objective functions are evaluated the optimization can continued as presented before.

6.2 Parallelization

The code used in order to develop the optimization process employs different parallelization techniques in order to reduce the time necessary to carry out the work. In Chapter 5.2 the parallel mode and the parallelization methods have been presented; in this paragraph the description of how the process has been parallelized is shown.

Functional parallelization for the optimizer and domain parallelization for the CFD solver are adopted. Due to the crowded cluster and the time available, the two studies developed present a little difference in the parallelization. The first study uses three nodes (therefore Slaves); whereas it has been possible to employ just two nodes, which means two Slaves, for the second study. For what concerns the parallelization of the solver, instead, the same strategy has been applied for both the cases: four processors in parallel are employed to solve the fluid domain.

The Functional parallelization can create some problems due to the fact that the different Slaves are carrying out diverse evaluations saving them in the same folder. For this reason two Slaves might write in the same file leading to errors in the optimization process. In order to overcome this problem to each Slave is associated to a number, called PID number. Each file saved by a Slave contains the PID number in its name; in this way each Slave works with its own files without enter in conflict with the other Slaves.

Thanks to the parallelisation a reduction of the time necessary to execute the optimization has been possible. Nevertheless a robust optimization process is slow and requires long time to run.

7 RESULTS

In this Chapter the results obtained with the two optimization studies performed are presented. The Pareto front and the more interesting configurations are showed. Moreover, the failures of the process have been recorded and classified on the basis of the reason that leads to the error. Finally, a comparison between the results obtained in the two different cases is realized.

7.1 First Study: Three CFD Evaluation for Each Design Vector

The first optimization process has been run for a longer time, and it is faster than the second one, since it uses three Slaves (see Chapter 6) and it executes less CFD evaluations. The optimization process realized 76 iterations obtaining 409 designs. Figure 7.1 shows a resume of successful designs and failures, classifying the latter.

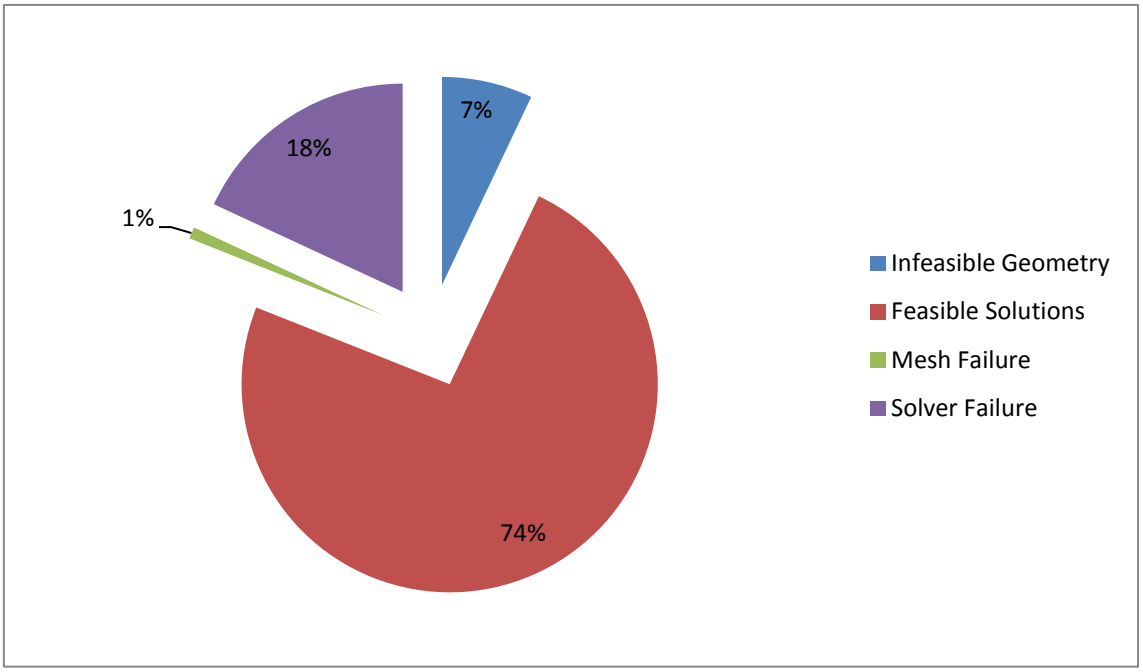

Figure 7.1: Resume of the outcomes of the optimization process for the first study

It can be noticed from the chart that most of the design vectors evaluated have lead to valid solutions; the 7% of the configurations created have been affected by overlapping and just for 1% of the design vectors the geometry has been considered feasible, but the creation of the mesh around the airfoil was not

possible. Finally, the 18% of the configurations has caused a failure of the solver. The high percentage of feasible solutions (74%) proves the reliability of the process; nevertheless, it can be improved in order to reach a higher number of valid solutions, focusing on the reduction of the solver failures.

The Pareto front obtained from this study, together with the other solutions composing the search pattern, is presented in Figure 7.2.

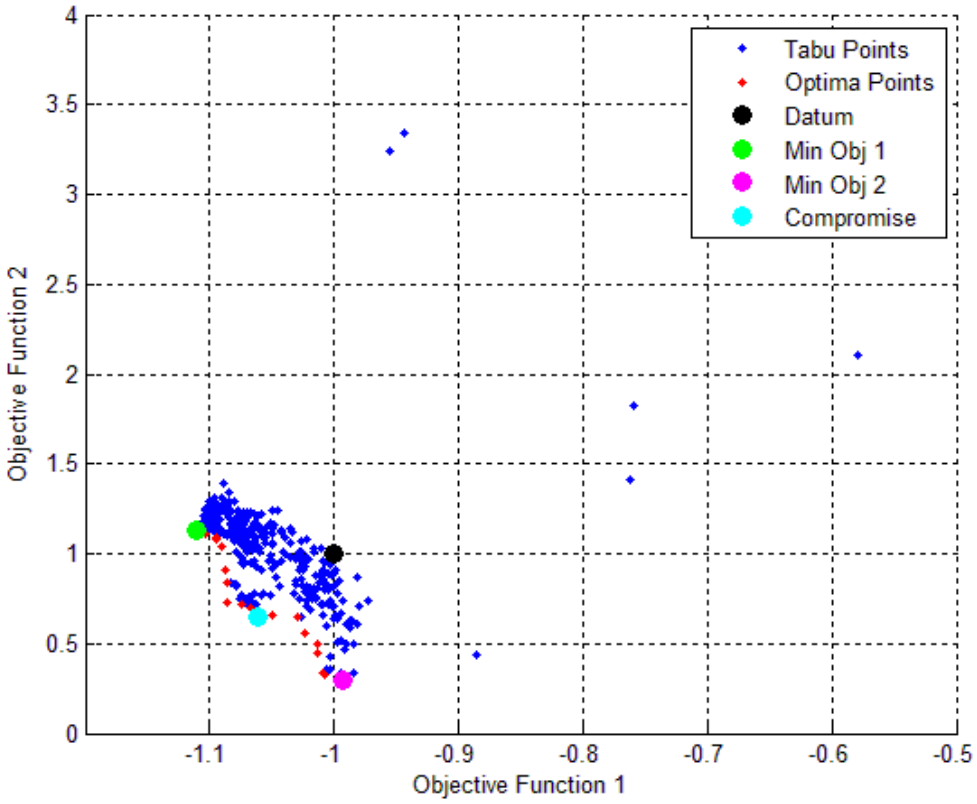

Figure 7.2: Pareto front and search pattern in the first study

In the chart above the "*Objective Function 1*" is Equation (5-5), this function aims to find the configurations that allow the maximum value for the c_l. "*Objective Function 2*", instead, is presented in Equation (5-6) and its purpose is to find the robust configurations that are less sensitive to the variations of the operating conditions.

In Figure 7.2 it possible to observe that most of the design vectors analyzed present a reduction of the first objective function, in comparison with the datum

configuration. On the other hand, there is a more equal distribution of the points that improve and that do not improve the "*Objective Function 2*". Furthermore, it is interesting to observe that the Pareto optimal set is not very uniform, it presents some scatter and some discontinuities are visible in the central region. This is due to the complexities related with the objective functions definition in a multi-point study and to the reduced exploration of the design space. Similar problems have been encountered also in the multi-point optimization process carried out in [2]. However, most of the design vectors evaluated are in a region close to the optimal one confirming the good convergence of the MOTS; there are some exceptions, some points positioned far from the Pareto front, due to the diversification process (described in Chapter 5).

From the observation of the Pareto optimal set, three configurations have been selected in order to be analyzed in a more detailed way; they represent the minimum for the "*Objective Function 1*", the minimum for the "*Objective Function 2*" and a compromise design.

The configuration of the airfoil that minimizes the first objective function, together with the datum geometry, is shown in Figure 7.3.

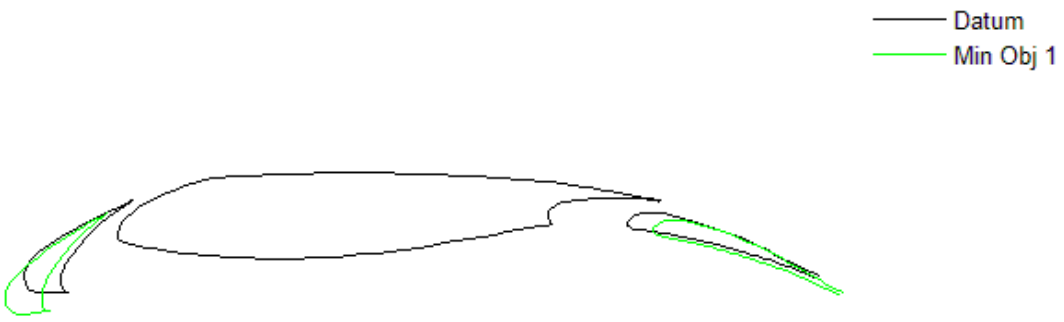

Figure 7.3: Datum geometry and configuration that minimizes the Objective Function 1 in the first study

This configuration allows an increase of the 11% for the lift performance; on the other hand it increases the second objective function of the 13%. Moreover, this design vector causes an increase of the drag coefficient compared to the datum geometry that leads to a value of the 1.3% for the penalty function (5-7). This

improvement in the lift capacity of the airfoil is reached thanks to the increase in the flow deflection angle and the increase in the gaps between the different elements that allows the boundary layers of the different part to remain separate each other.

In Figure 7.4 the configuration that minimizes the “Objective Function 2” is presented.

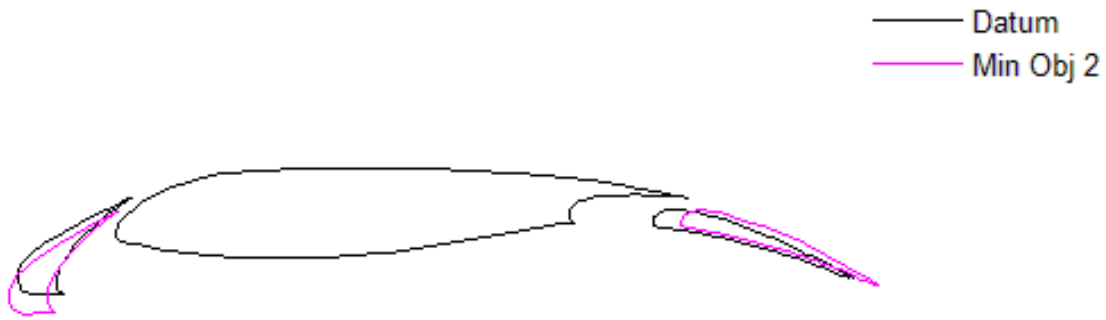

Figure 7.4: Datum geometry and configuration that minimizes the Objective Function 2 in the first study

These positions of flap and slat lead to a small reduction of the first objective (+1% in Objective Function 1); whereas the second objective function decreases by 71%. This design vector is affected by a small increment of the c_d that is quantified by a value of 3.2% of the penalty function. This configuration of the airfoil guarantees a good stability of the performance, in terms of lift, and low degradation due to the variations on the angle of attack and deflection of the flap.

The compromise configuration is shown in Figure 7.5. It is selected from the central region of the Pareto front with the aim of improving both the objective functions in comparison with the datum geometry.

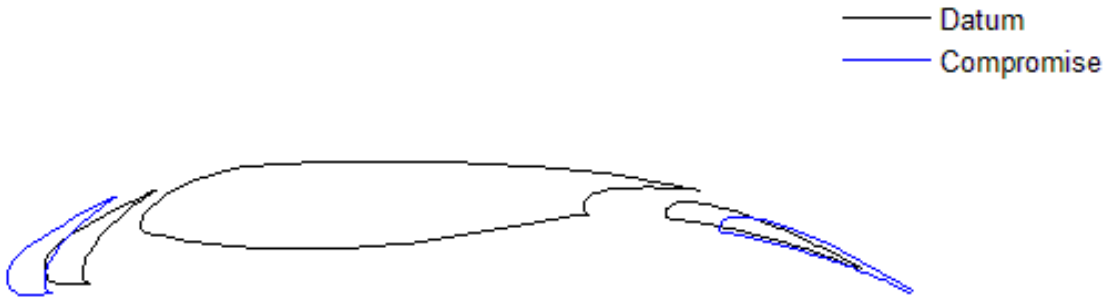

Figure 7.5: Datum geometry and compromise configuration in the first study

This configuration is characterized by a noticeable increase in the gaps between the elements. The first objective function is improved by 6% the second one by 34%. There is also a small increase in the drag coefficient that is represented with the 1.2% value of the penalty function. This geometry leads to improvement in both the objective functions with a low increment of the drag.

In Figure 7.6, all the optimal design vectors selected from the Pareto front are shown together. It can be observed that the configurations that minimize the first and the second objective functions have small differences whereas the compromise one presents larger gaps between the elements.

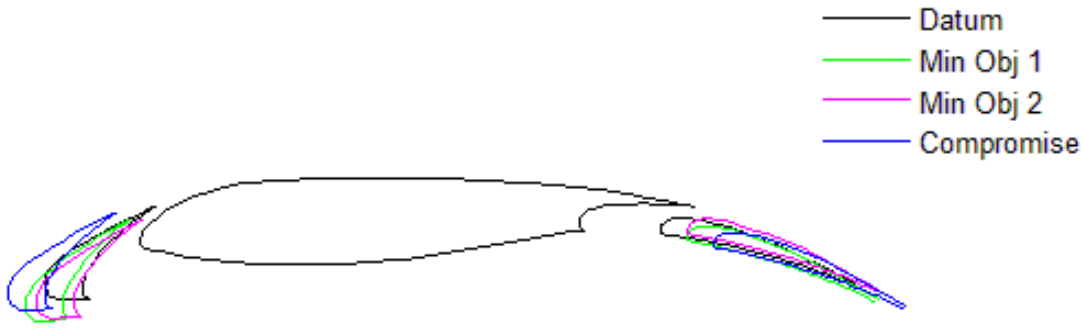

Figure 7.6: Comparison of the datum and all the optimal configurations selected in the first study

The results obtained from these three configurations are summarized in Table 7.1.

	Min Obj 1	Min Obj 2	Compromise
Δ_{Obj1}	-11.0%	+1.1%	-6.2%
Δ_{Obj2}	+13.4%	-70.9%	-34.4%
P	1.3%	3.2%	1.2%

Table 7.1: Resume of the variations of the objective functions and penalty function for the three configurations analysed in the first study

In Figure 7.7 a chart resume the value of the c_l for the three optimal geometries and the datum one at the different operating conditions considered. For each configuration three lift coefficients are presented; Condition 1 represent the CFD evaluation performed with angle of attack equal to $\alpha - 1°$ and deflection of the flap equal to $\Theta + 1°$, Nominal condition set these parameters respectively equal to α and Θ, and in Condition 2 they are equal to $\alpha + 1°$ and $\Theta - 1°$. Due to the confidentiality of the results the values of the lift coefficients are not shown in this chart and in the following ones.

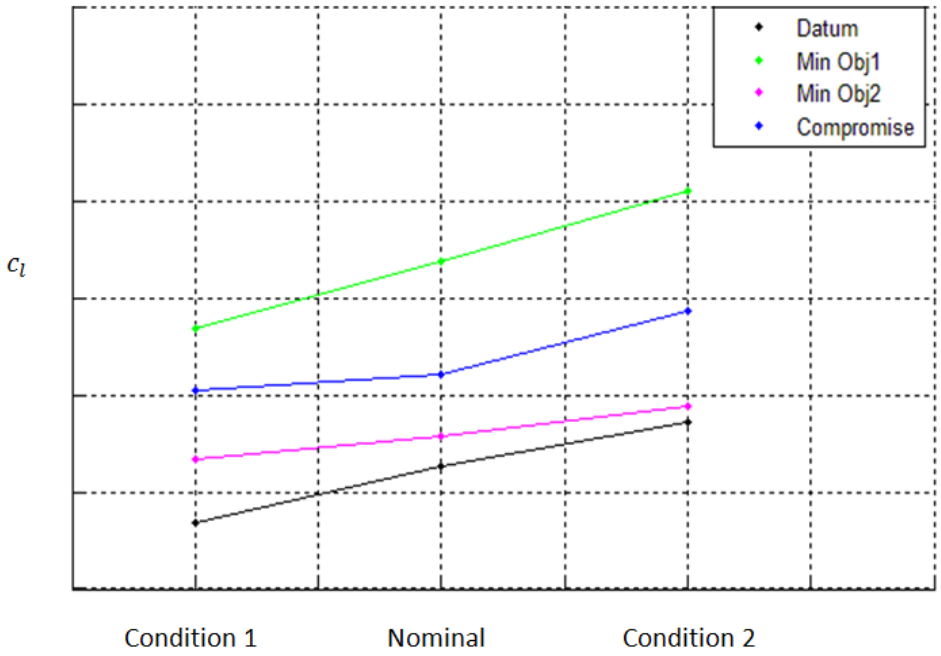

Figure 7.7: Comparison of the lift coefficients between the different configurations for the first study

It can be observed that all the all the optimal configurations improve the c_l compared to the datum geometry for each operating condition. Nevertheless, the design vector that minimizes the "*Objective Function 2*" has a higher value of the first objective in comparison with the datum; this is due to the increase in drag that is taken into account using the penalty function. Observing the graph it is possible to notice the sensitivity of the lift coefficient to the different operating conditions; the geometry that minimizes the second objective is almost insensitive to the variation of the operating conditions, whereas the other configurations are more affected by these variations.

The configurations selected in the Pareto front have been analysed in the post process executing the CFD evaluation with ANSYS FLUENT. In Figure 7.8 the comparison of the pressure coefficient between datum and optimal designs is made.

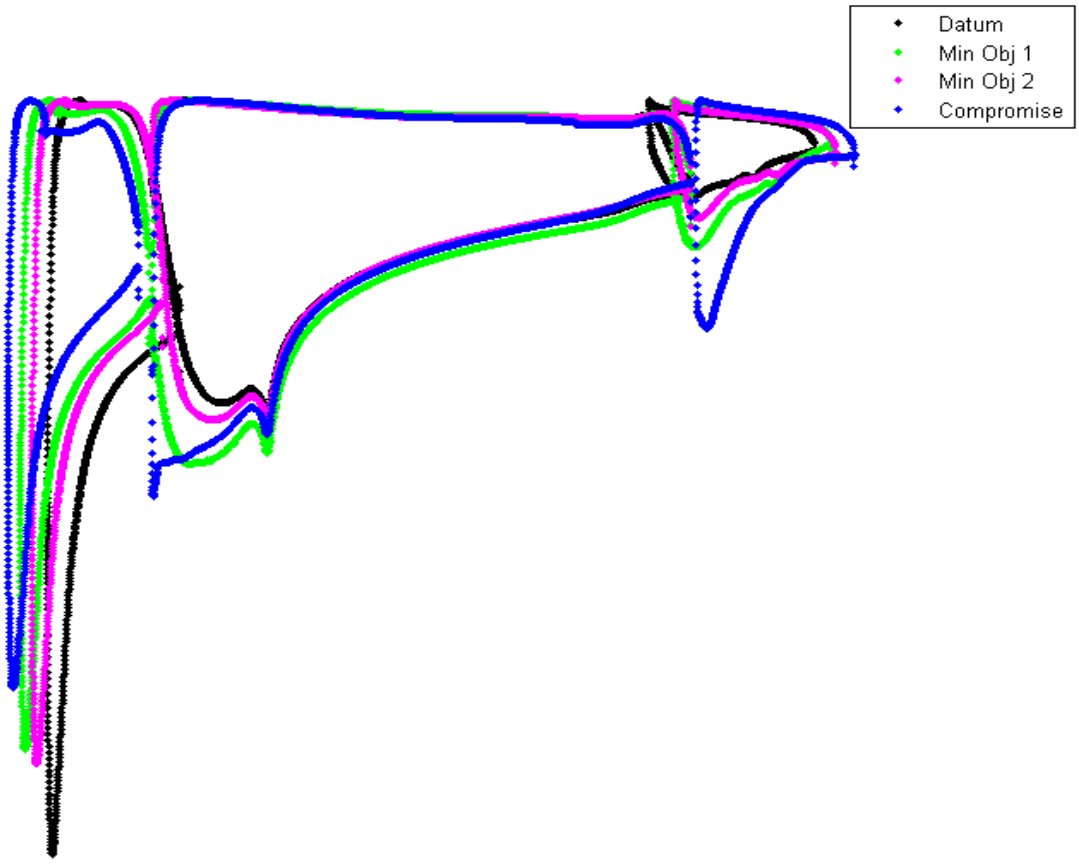

Figure 7.8: Comparison of the cp distribution for different configurations analysed in the first study

It can be observed that all the geometries chosen from the Pareto front reduce the peak of suction of the slat and redistribute it in through the three elements that compose the airfoil. This effect is greater in the compromise configuration, but it is noticeable also in the other two.

The contours of static pressure and velocity for the three optimal configurations studied are shown from Figure 7.9 to Figure 7.14. In these images it is possible to observe the new distribution of pressure, as presented for the pressure coefficient. The velocity contours change in a similar way reducing the fluid speed in the slat leading edge and increasing it in the main and in the flap leading edges. These new distribution of pressure and velocity are more evident for the compromise design.

In the end, the streamlines for the three optimal configurations are presented in Figure 7.15, Figure 7.16 and Figure 7.17.

Figure 7.9: Static pressure contours for the configuration that minimizes the Objective Function 1 in the first study (units Pa)

Figure 7.10: Static pressure contours for the configuration that minimizes the Objective Function 2 in the first study (units Pa)

Figure 7.11: Static pressure contours for the compromise configuration in the first study (units Pa)

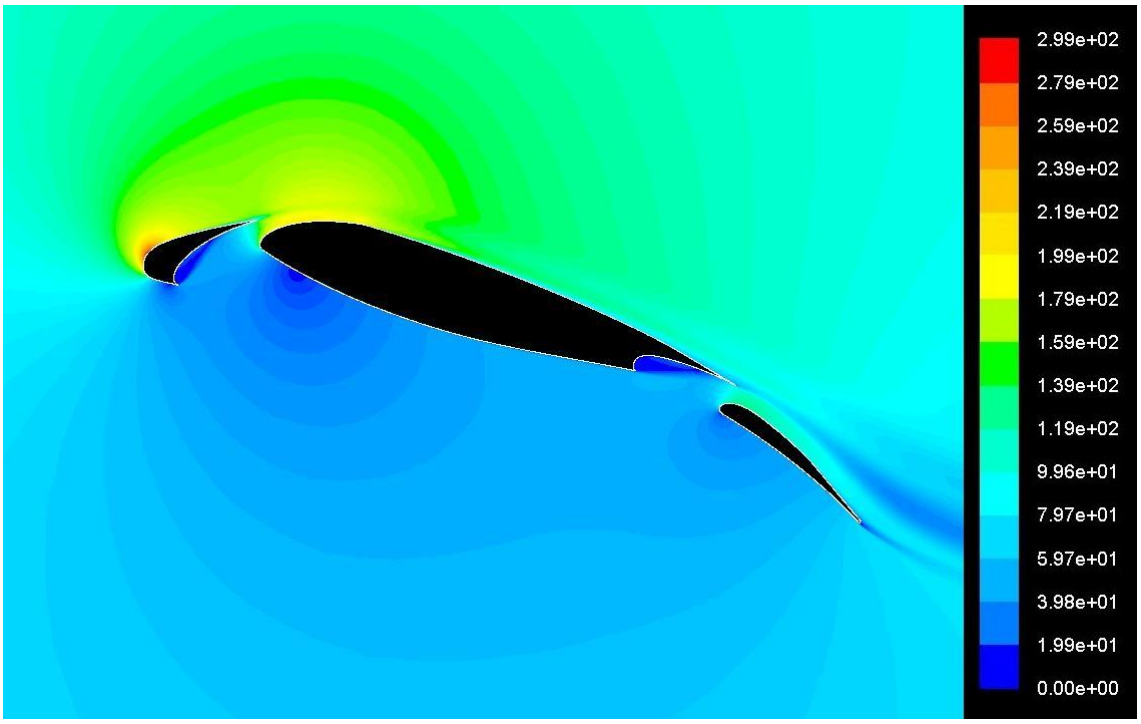

Figure 7.12: Velocity contours for the configuration that minimizes the Objective Function 1 in the first study (units m/s)

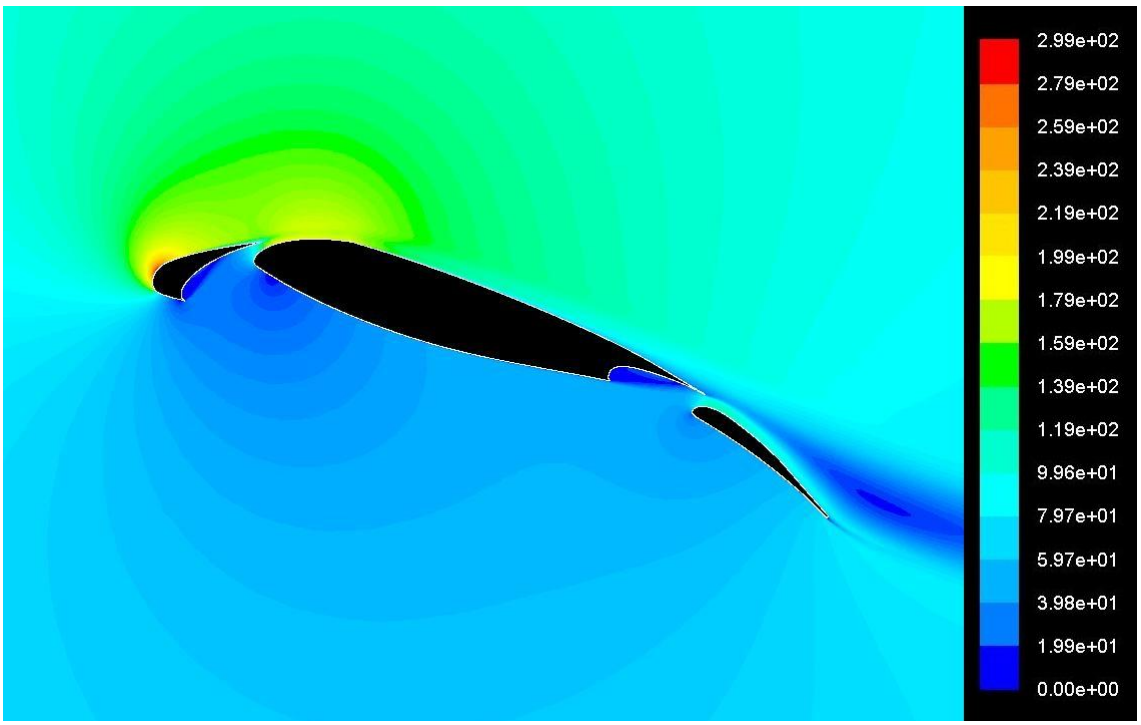

Figure 7.13: Velocity contours for the configuration that minimizes the Objective Function 2 in the first study (units m/s)

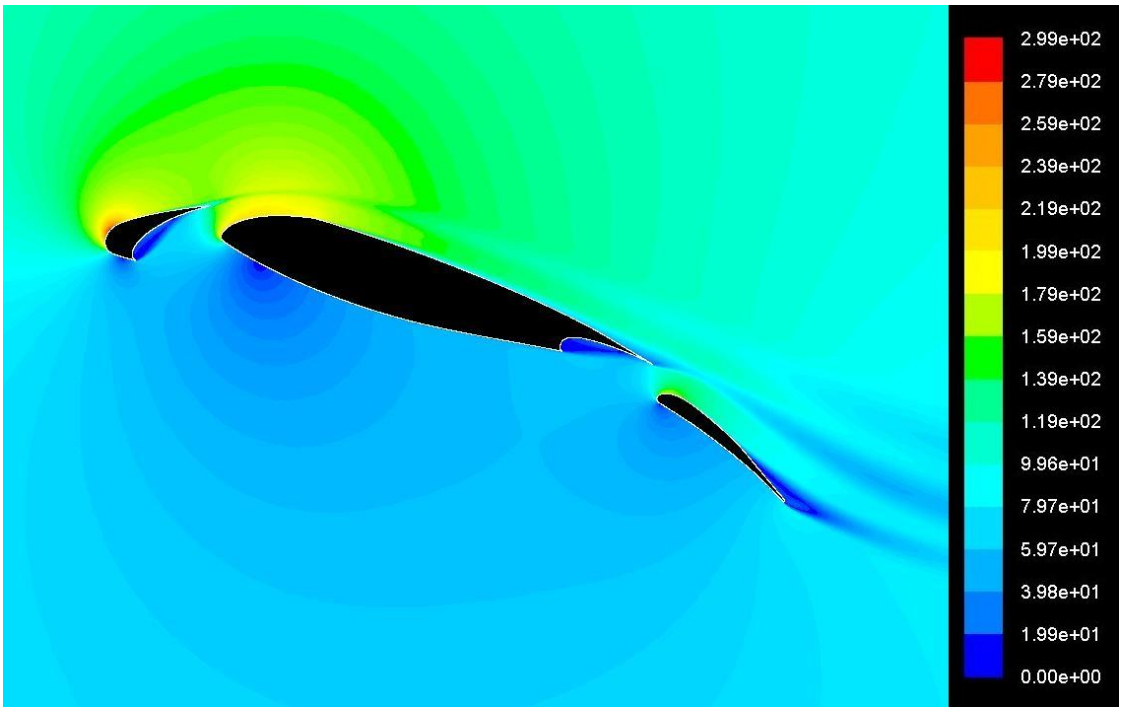

Figure 7.14: Velocity contours for the compromise configuration in the first study (units m/s)

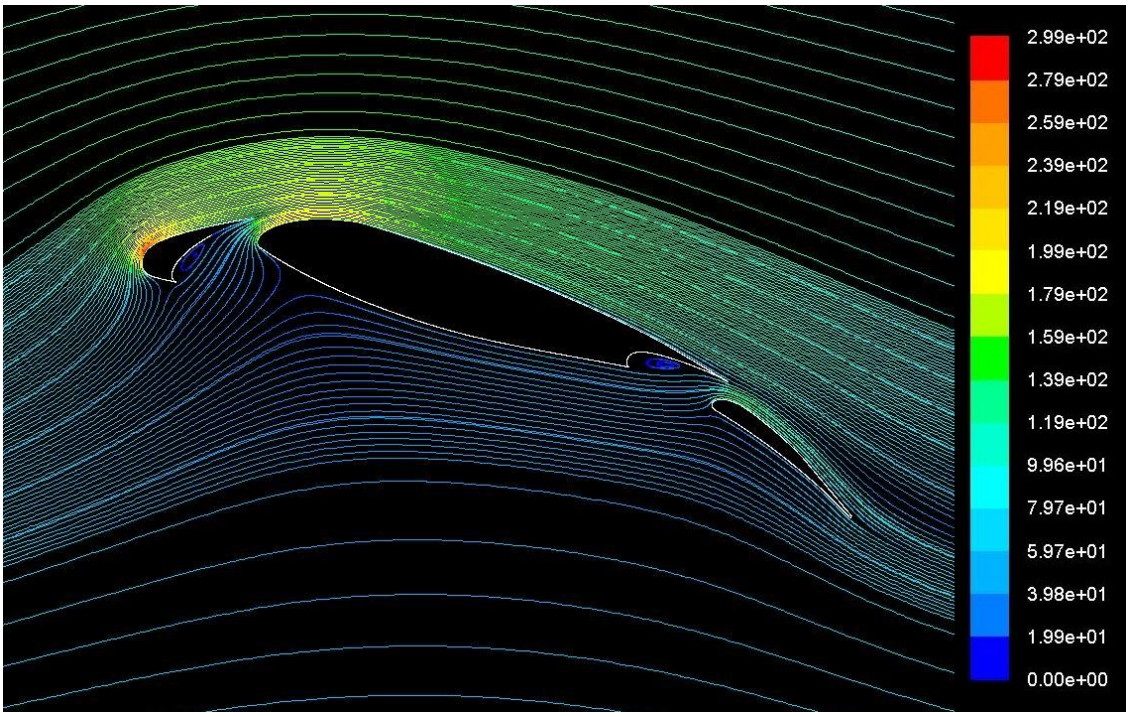

Figure 7.15: Streamlines for the configuration that minimizes the Objective Function 1 in the first study (parameter: velocity magnitude, units: m/s)

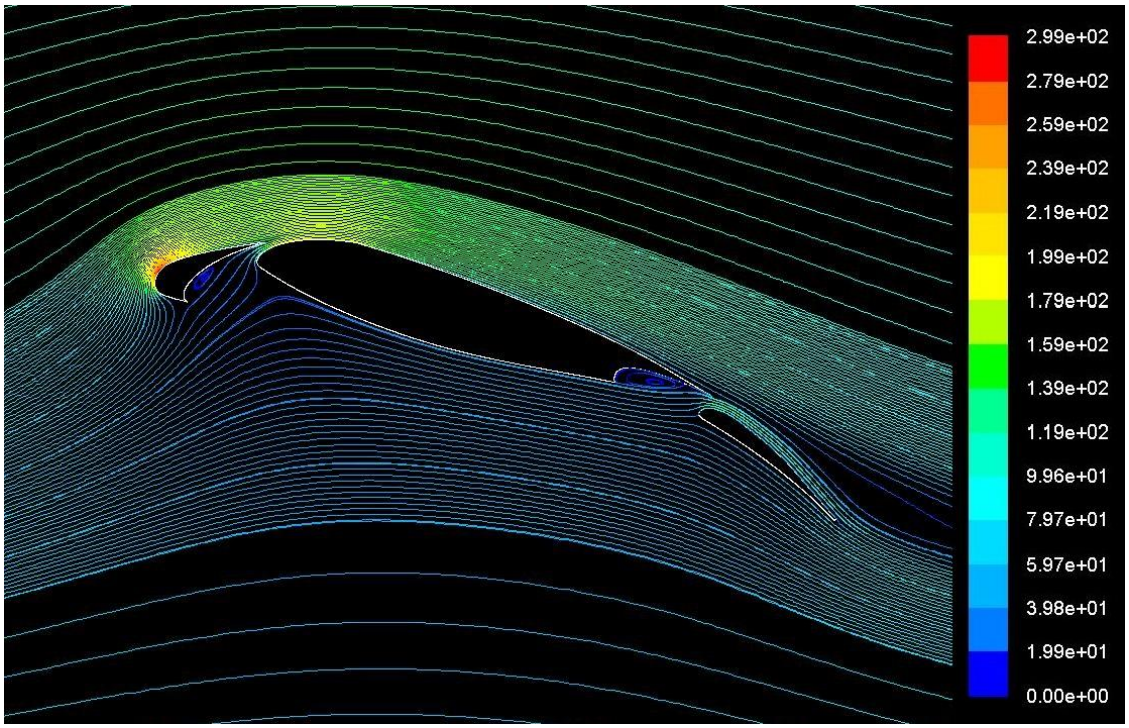

Figure 7.16: Streamlines for the configuration that minimizes the Objective Function 2 in the first study (parameter: velocity magnitude, units: m/s)

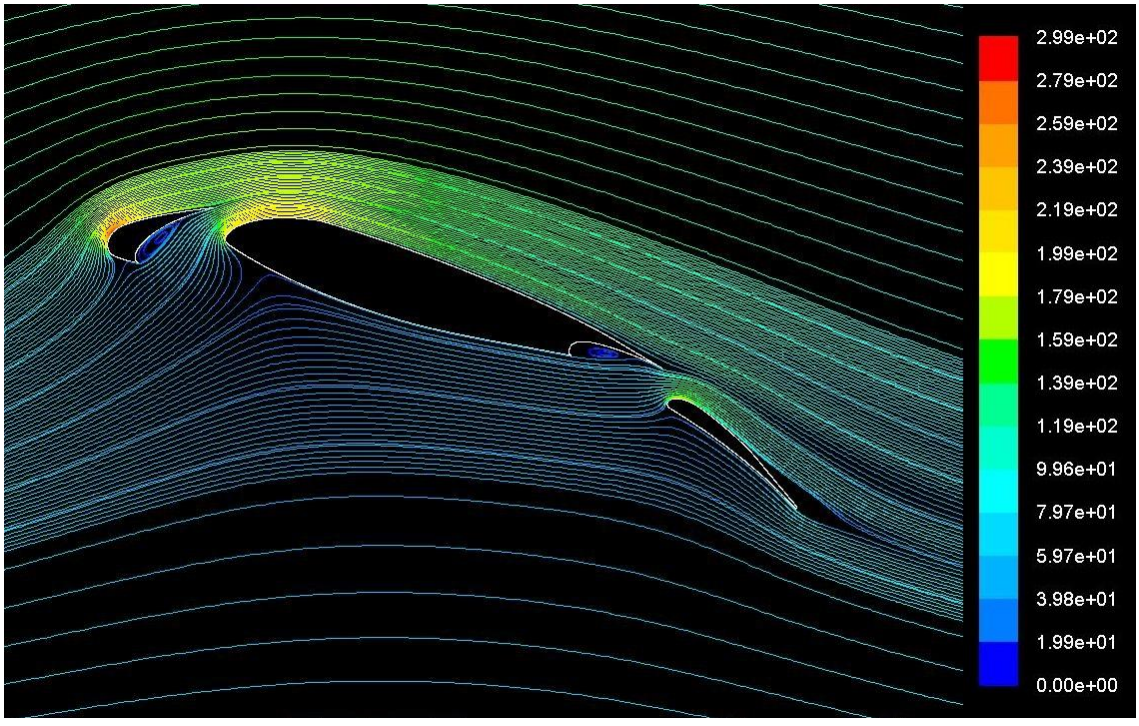

Figure 7.17: Streamlines contours for the compromise configuration in the first study (parameter: velocity magnitude, units: m/s)

7.2 Second Study: Nine CFD Evaluation for Each Design Vector

The second optimization study has been slower due to the large number of CFD evaluations required and the fact that it has been possible to use only two Slaves. Furthermore, short time has been available to execute this analysis. For all these reasons, only 17 iterations of the optimization process have been completed, obtaining 135 valid designs. In Figure 7.18 a chart represents the valid solutions and the failures classified on the basis of the type. It can be noticed also this second optimization process has a good reliability; in fact the 82% of the design vectors analysed lead to feasible solutions. Few failures are caused by unfeasible geometries (2%) and errors occurring during the mesh generation (1%). Most of the failures are related with the solver (15%). The percentage of total failures is lower than the one obtained in the first study; but it must be taken into account that the exploration of the design space is not very wide in this case, therefore most of the design vector analysed are close to the datum configuration.

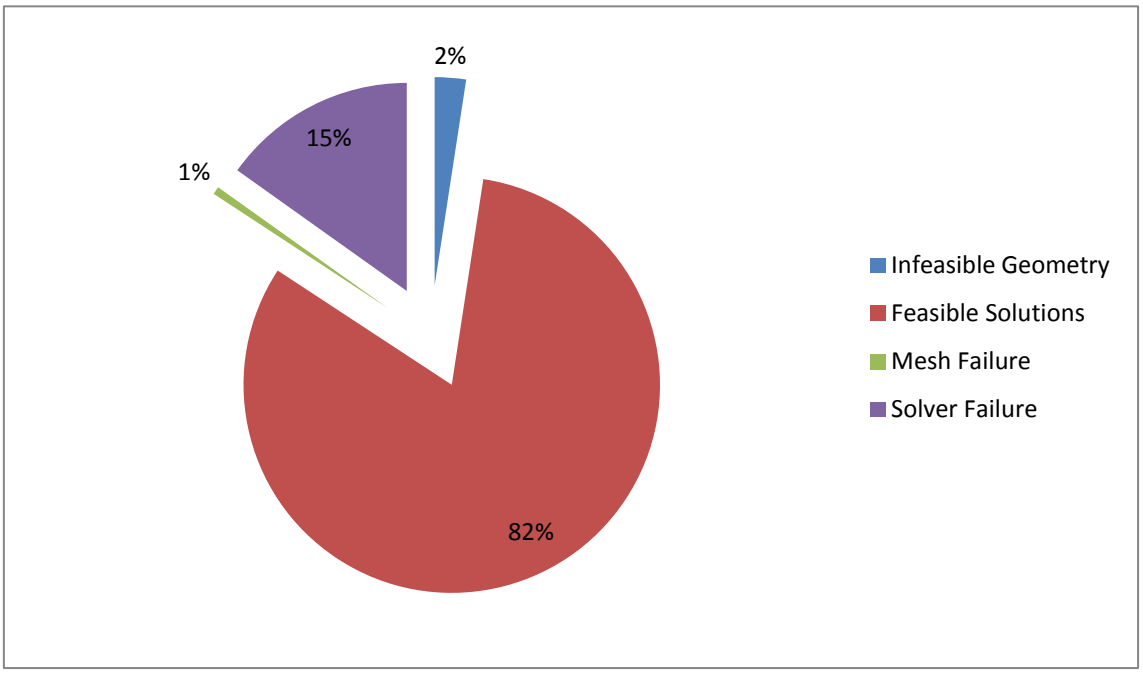

Figure 7.18: Resume of the optimization outcomes for the second study

The search pattern and the Pareto Front are shown in Figure 7.19.

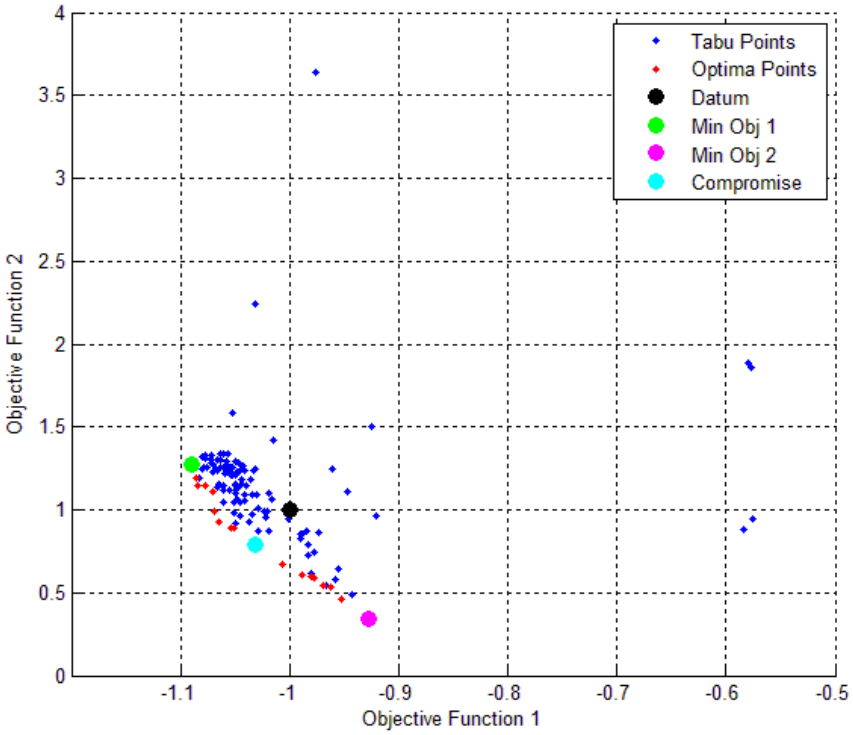

Figure 7.19: Search pattern and Pareto front for the second study

The objective functions evaluated are the ones for the second studies, presented in Chapter 5; the first and the second objectives are represented, respectively, by Equations (5-8) and (5-9) and their meaning have been already explained. The research is focused in the neighbourhood of the Pareto optimal set, underlining the convergence of the method. As it has been noticed for the previous study, the Pareto front presents some scatter and some discontinuities, especially in the central region, this is due to the already mentioned complexities related with multi-point optimization that are increased performing nine CFD simulations for each design vector. It can be observed that all the optimal solutions improve both the objectives compared to the datum geometry.

From the Pareto front the design vectors that minimize "*Objective Function 1*" and "*Objective Function 2*" and the compromise design are selected and analysed.

In Figure 7.20 the design vector that minimizes the first objective function is presented. This configuration increases the lift performance by 9%; on the other hand it is related with an increment in the second objective function by 28%. Furthermore, a negligible increase in the drag is obtained, represented by a value of 0.3% of the penalty function (5-10). Similarly to the previous study, the configuration that allows the best increase in the lift performance present an increment in the gaps between the elements that compose the airfoil.

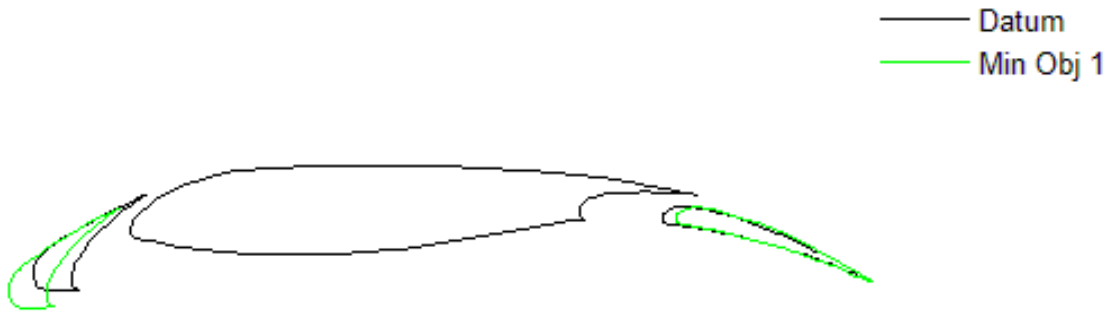

Figure 7.20: Datum geometry and configuration that minimizes the Objective Function 1 in the second study

The configuration of the airfoil corresponding to the minimum for the "*Objective Function 2*" is shown in Figure 7.21.

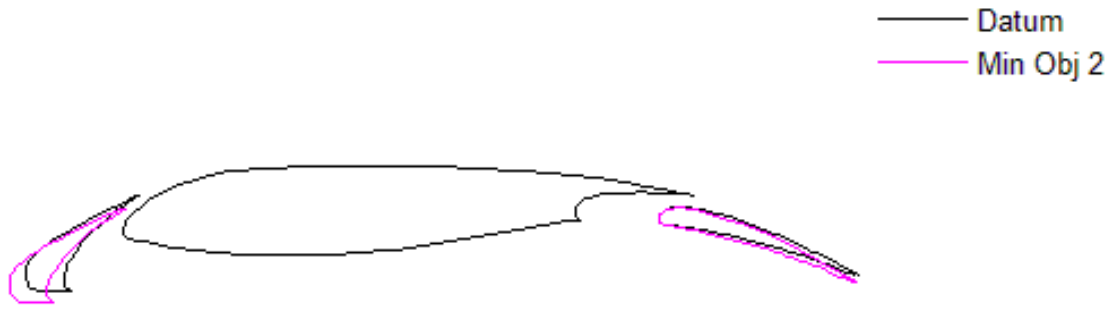

Figure 7.21: Datum geometry and configuration that minimizes the Objective Function 2 in the second study

This geometry decreases the second objective function of the 66% making the performance of the airfoil affected by varying operating conditions almost stable. "*Objective Function 1*" and drag, instead, are increased; the first objective rises by 7% while the penalty function assumes the value of 4.9%.

Finally, Figure 7.22 represents the configuration of compromise. This design vector allows the improvement of both the objective functions: the first objective is reduced by 3%, whereas the second one is decreased by 21%. On the other hand, there is an increase in drag represented by a value of the penalty function of 6.4%.

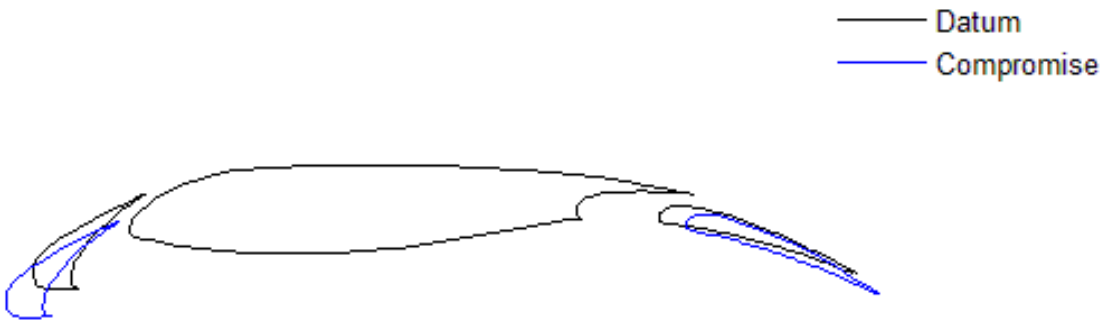

Figure 7.22: Compromise configuration in the second study

In Figure 7.23 all the optimal configurations selected are presented together with the datum geometry. It can be observed that the optimal configurations have small differences; this is due to the reduced exploration of the design space that has been realized.

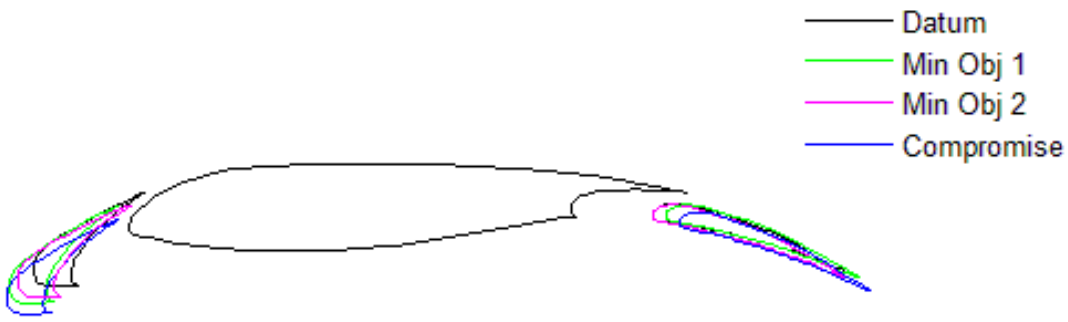

Figure 7.23: Comparison of the datum and all the optimal configurations selected in the second study

The results obtained with these optimal designs are summarised in Table 7.2. Moreover, Figure 7.24 presents a graphical representation of the lift coefficients, at different operating conditions, for the different configurations analysed.

	Min Obj 1	Min Obj 2	Compromise
Δ_{Obj1}	-9.0%	+7.3%	-3.1%
Δ_{Obj2}	+27.7%	-66.0%	-21.5%
P	0.3%	4.9%	6.4%

Table 7.2: Resume of the variations of the objective functions and penalty function for the three configurations analysed in the second study

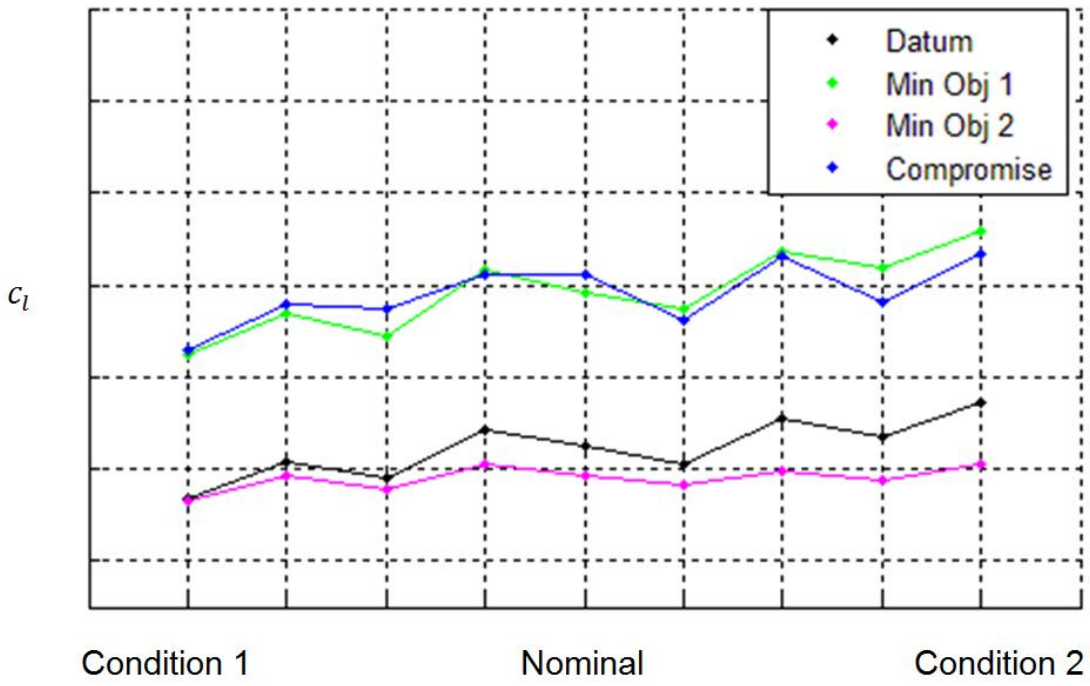

Figure 7.24: Comparison of the lift coefficients between the different configurations in the second study

The chart presented in Figure 7.24 shows the value of the lift coefficient on the y-axis and the different operating conditions on the x-axis, using diverse colours in order to distinguish the configurations. The operating conditions in this study are all the nine combinations of angles of attack and deflections of the flap; in the x-axis “Condition 1”, “Nominal” and “Condition 2” that are shown have been described in the previous study case; the remaining points represents all the other operating conditions evaluated. It is interesting to observe that, for each configuration, the lift coefficients are included in the interval between the c_l at

Condition 1 and the one at Condition 2. Analysing the graph it can be stated that: the configuration that minimizes the first objective function can improve the lift coefficient considerably, even if it is affected by an high sensitivity on the operating conditions; the design vector that minimizes the "*Objective function 2*" presents a very good stability and a slight reduction of the c_l in comparison with the datum configuration; in the end, the compromise design leads to an interesting increase in lift coefficient, at the nominal condition it is higher than the configuration corresponding to the minimum for the first objective, and a quite low sensitivity to the varying condition, nevertheless it is penalised by the increase in drag coefficient.

The post process using ANSYS FLUENT has been carried out for the optimal configurations obtained from this study. A comparison between the different pressure coefficients is shown in Figure 7.25.

Figure 7.25: Comparison of the cp distribution for different configurations analysed in the second study

The values of the pressure coefficient are not shown since these data are confidential. From the graph it is possible to observe that all the optimal configurations reduce the peak in suction in the slat; this effect is not as

remarkable as it is in the first study case. In the compromise configuration and in the one that minimizes "*Objective function 1*" the c_p is redistributed on the two remaining elements; whereas, the design that leads to the lowest value of the second objective presents a reduction in suction for all the elements of the airfoil.

These redistributions of pressure coefficients are visible in the pressure and velocity contours; in the configuration of compromise and in the one that minimizes the first objective function it is possible to notice the decrease in pressure and the corresponding increase in velocity at the leading edge of main element and flap.

The contours of static pressure and velocity magnitude, and the streamlines for the optimal configuration are presented from Figure 7.26 to Figure 7.34.

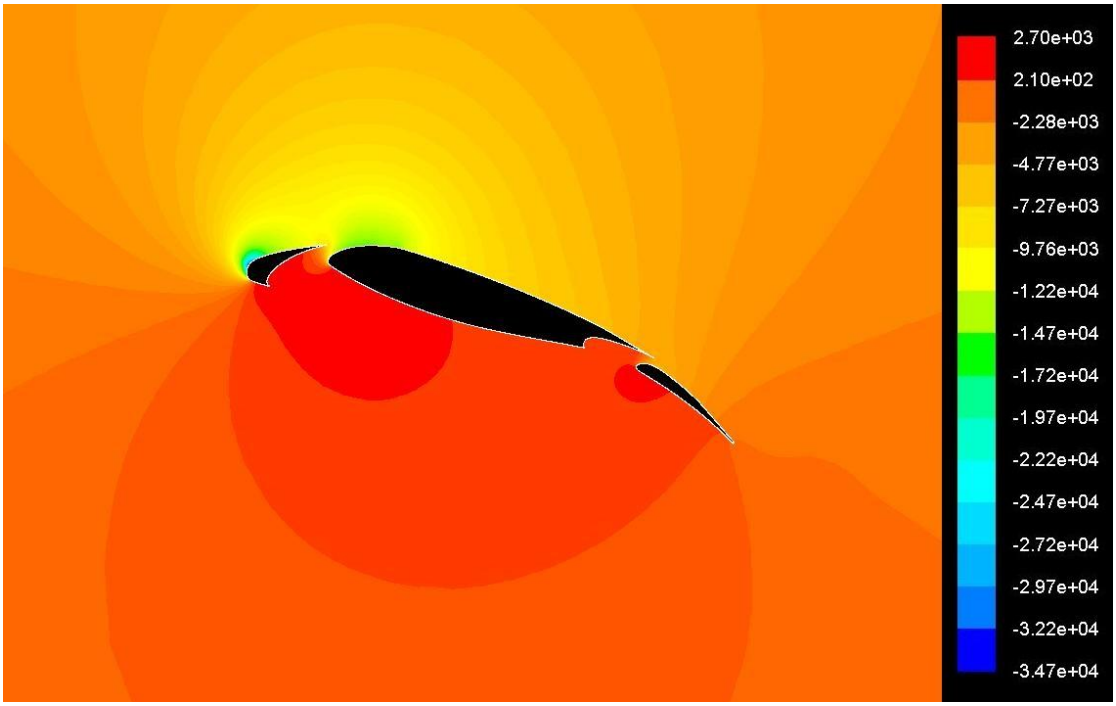

Figure 7.26: Static pressure contours for the configuration that minimize Objective Function 1 in the second study (units: Pa)

Figure 7.27: Static pressure contours for the configuration that minimizes Objective Function 2 in the second study (units: Pa)

Figure 7.28: Static pressure contours for the compromise configuration in the second study (units: Pa)

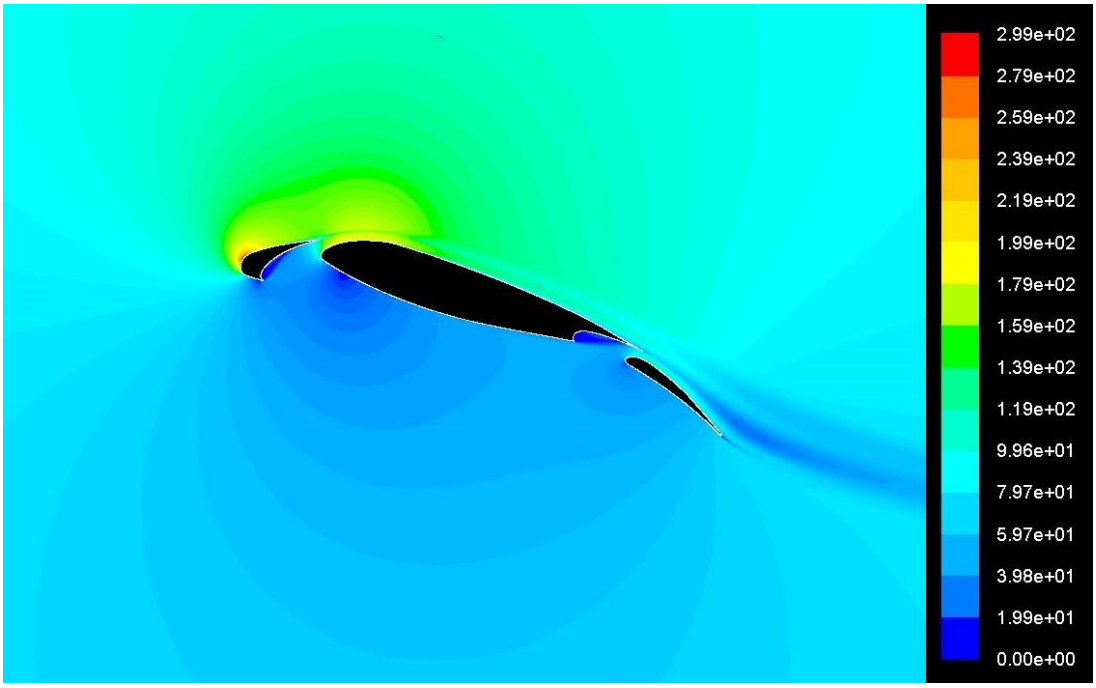

Figure 7.29: Velocity magnitude contours for the configuration that minimizes Objective Function 1 in the second study (units: m/s)

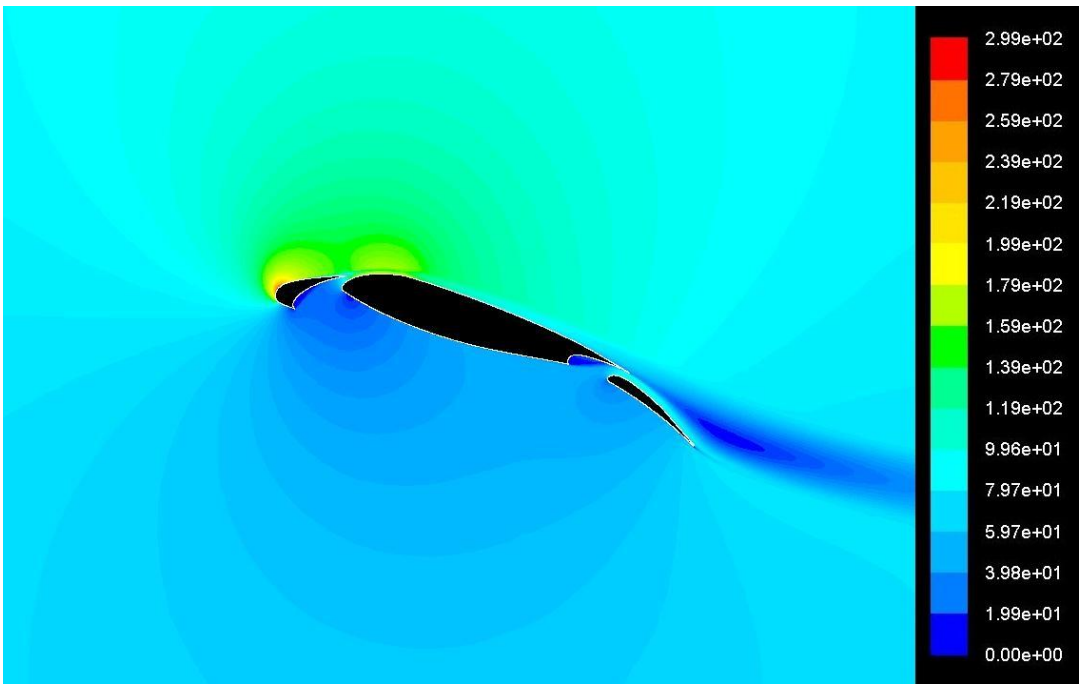

Figure 7.30: Velocity magnitude contours for the configuration that minimizes Objective Function 2 in the second study (units: m/s)

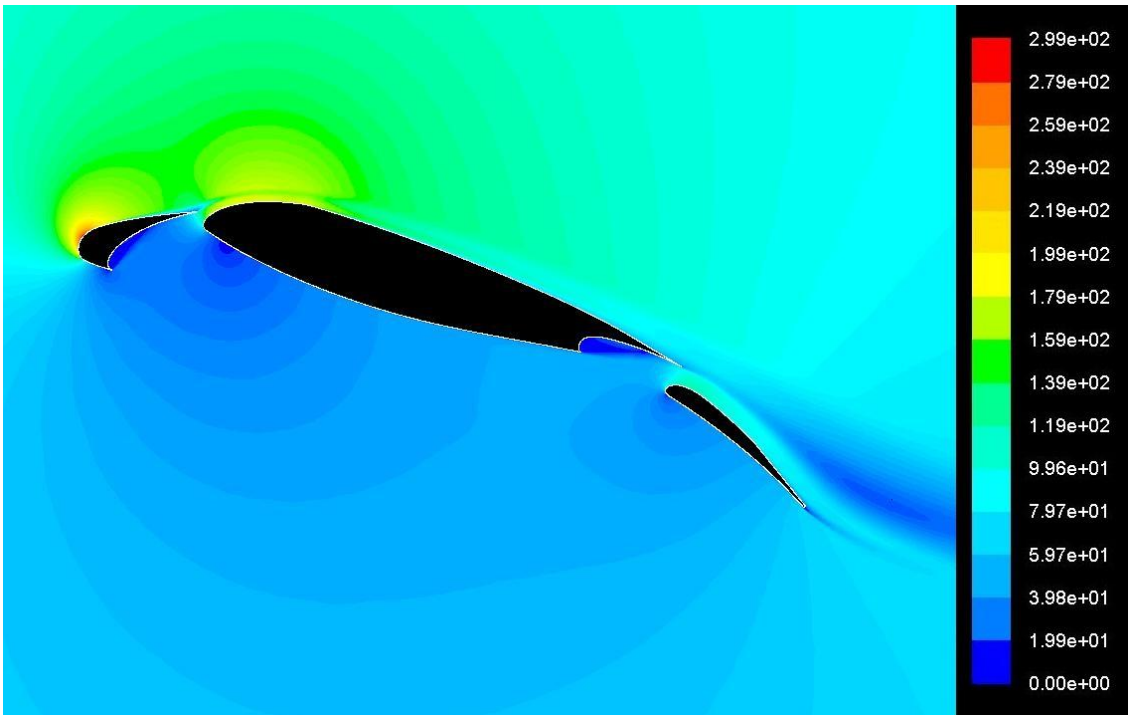

Figure 7.31: Velocity magnitude contours for the compromise configuration in the second study (units: m/s)

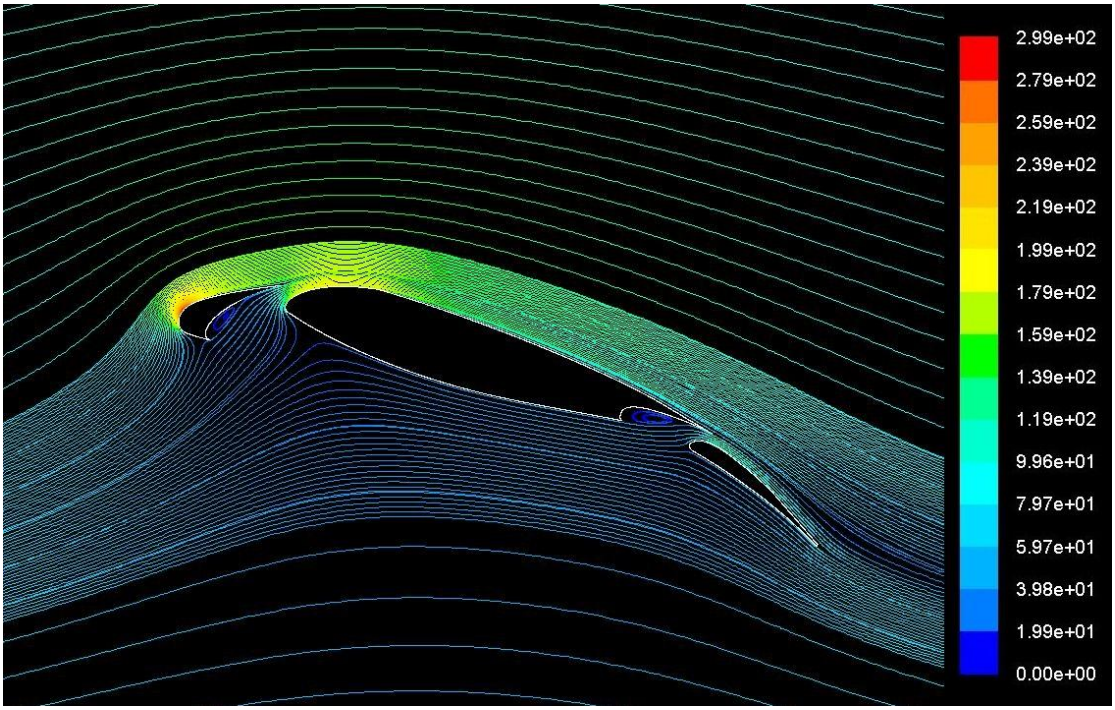

Figure 7.32: Streamlines for the configuration that minimizes Objective Function 1 in the second study (parameter: velocity magnitude, units: m/s)

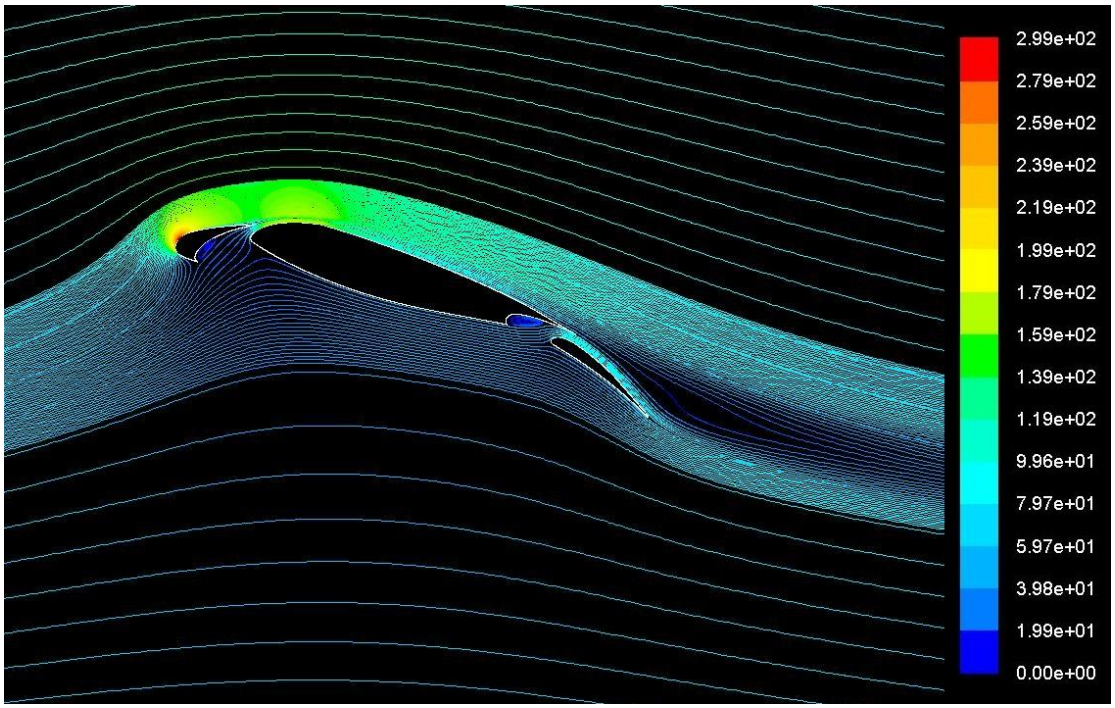

Figure 7.33: Streamlines for the configuration that minimizes Objective Function 2 in the second study (parameter: velocity magnitude, units: m/s)

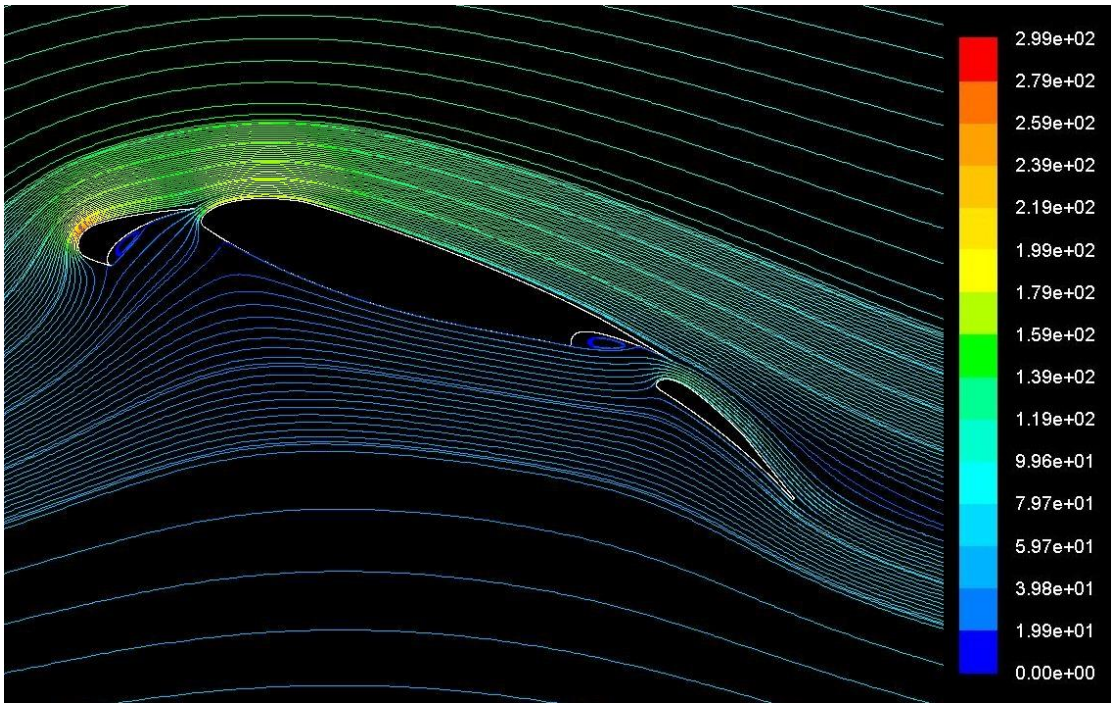

Figure 7.34: Streamlines magnitude contours for the compromise configuration in the second study (parameter: velocity magnitude, units: m/s)

7.3 Comparison of the Different Studies

In this section the two optimization studies developed are compared. In order to carry out a proper comparison it would be necessary that the two studies execute a comparable number of iterations and, as a consequence, have a similar exploration of the design space. This has not been possible since first study reached a higher number of iterations than the second one. Nevertheless, the results obtained are put side by side in order to see if the two studies can be interchangeable.

It has been chosen to compare only the two solutions corresponding to the extremes of the Pareto front, since the two compromise configuration chosen in the two studies are not comparable.

The designs that minimize the "*Objective function 1*" in the two studies are presented in Figure 7.35.

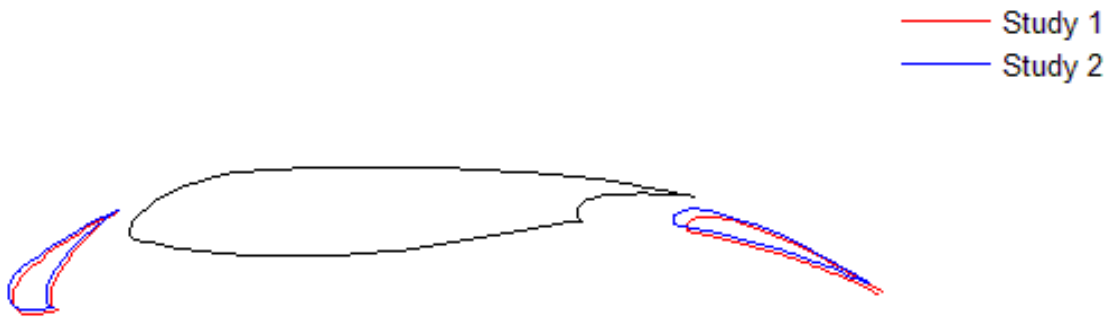

Figure 7.35: Comparison of the configurations that minimize Objective Function 1 in the two studies

It can be noticed that the two configurations are similar but not exactly the same. However, it is interesting to observe that the gap between slat and main element and the deflection angle of the flap are the same for both the configurations. In Figure 7.36, the c_l at different states for the two geometries taken into account are compared. The operating conditions displayed on the x axis have already been described previously. It can be noticed that the lift coefficients for the configuration obtained from the second study are lower than

the ones of the other one; nevertheless, they present a similar variation of performance due to the change in operating conditions.

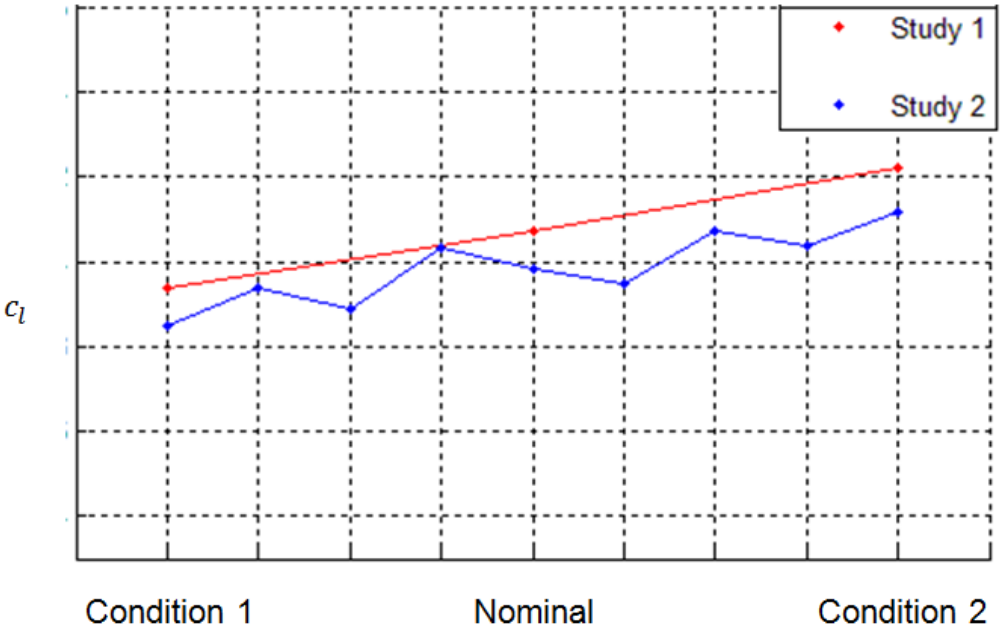

Figure 7.36: Comparison of the lift coefficients of the configurations for minimum objective function 1 in the two studies

The two configurations that minimize the second objective function are compared in Figure 7.37. Also in this case it is possible to see that the geometries are similar, in particular they have the same gap between slat and main element and the same deflection angle for the flap. Nevertheless, they are not exactly the same.

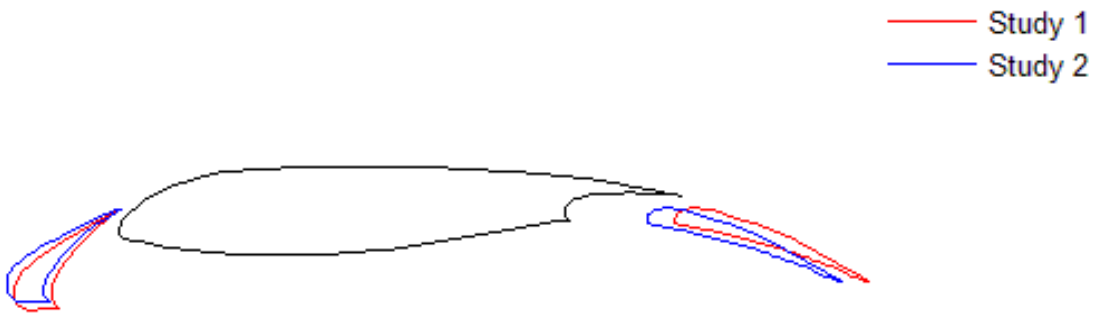

Figure 7.37: Comparison of the configurations that minimize Objective Function 2 in the two studies

In Figure 7.38 the comparison of the lift coefficients for the configurations presented in Figure 7.37 is shown. Again there is a slight difference in the corresponding c_l, whereas its variation with the change in operating condition is comparable in the two cases.

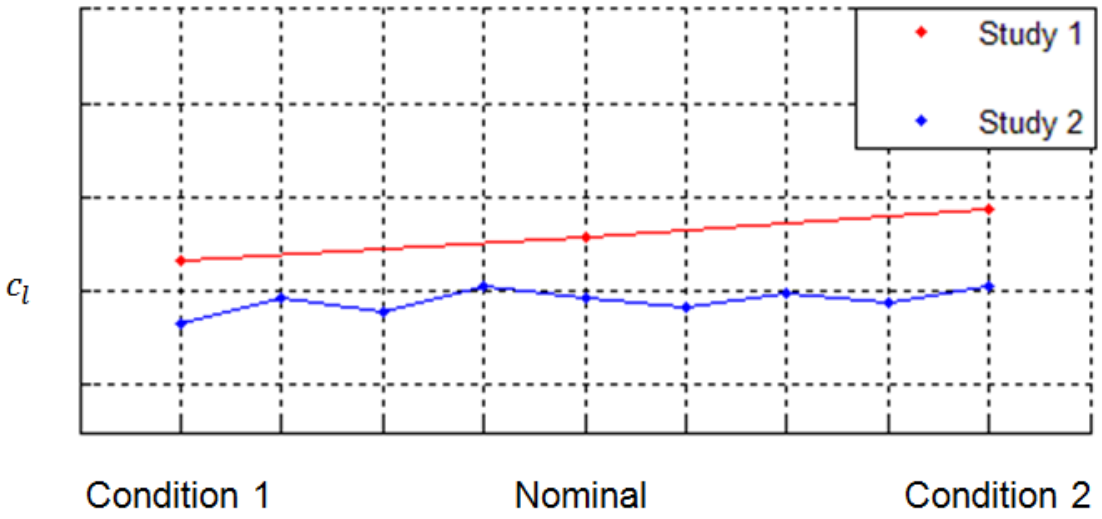

Figure 7.38: Comparison of the lift coefficients of the configurations for minimum objective function 2 in the two studies

In Figure 7.39 and Figure 7.40 the representations with parallel coordinates of the points composing the Pareto fronts of the two studies are shown; the eight vertical axes represent the six design variables plus the two objective functions; each optimal configuration is represented by a black line that connect the six design variables that defines the geometry and the values of the two objective functions associated with that design. Due to the small number of optimal solutions and to the different exploration of the design space performed by the two studies, it is difficult to notice similarities between the figures. Nevertheless, it can be observed that for both the studies the optimal design vectors are characterized by a good variety of slat positions whereas fewer changes in the flap locations can be noticed.

Due to the small number of iterations executed, the exploration of the design space in the second study is not as wide as the one carried out in the first one. This is the main reason why the results of the two processes are not perfectly

matching. In order to carry out a more precise comparison it is necessary to execute the same number of iterations in both the studies, in this work the time to do it was not available therefore it has been possible to realize just this preliminary comparison. However, this preliminary comparison hints good agreement between the two approaches.

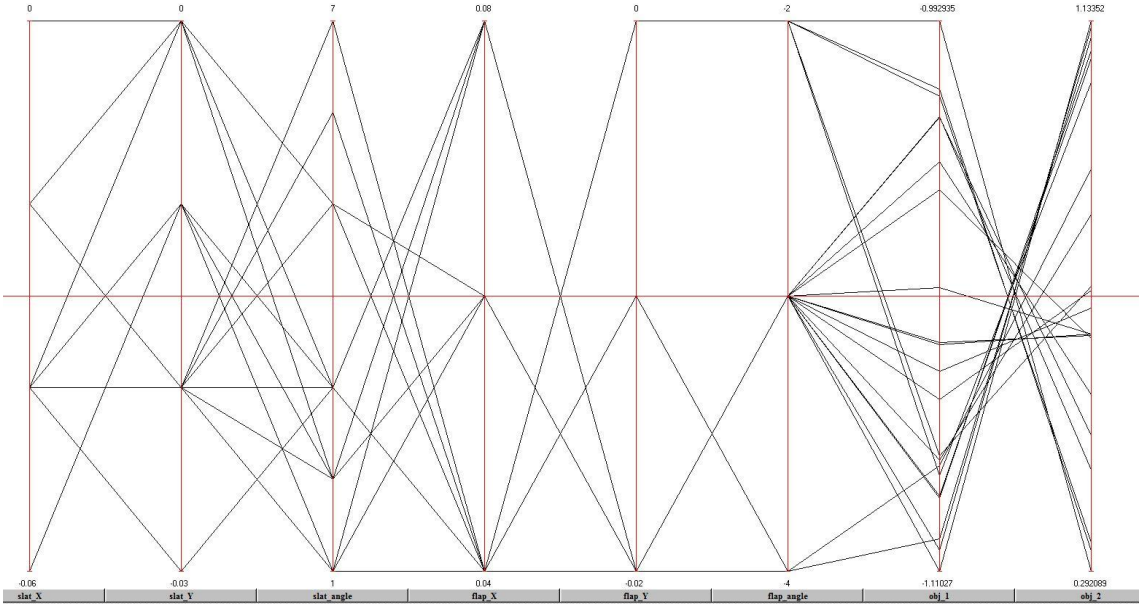

Figure 7.39: Representation of the Pareto front obtained in the first study with parallel coordinates

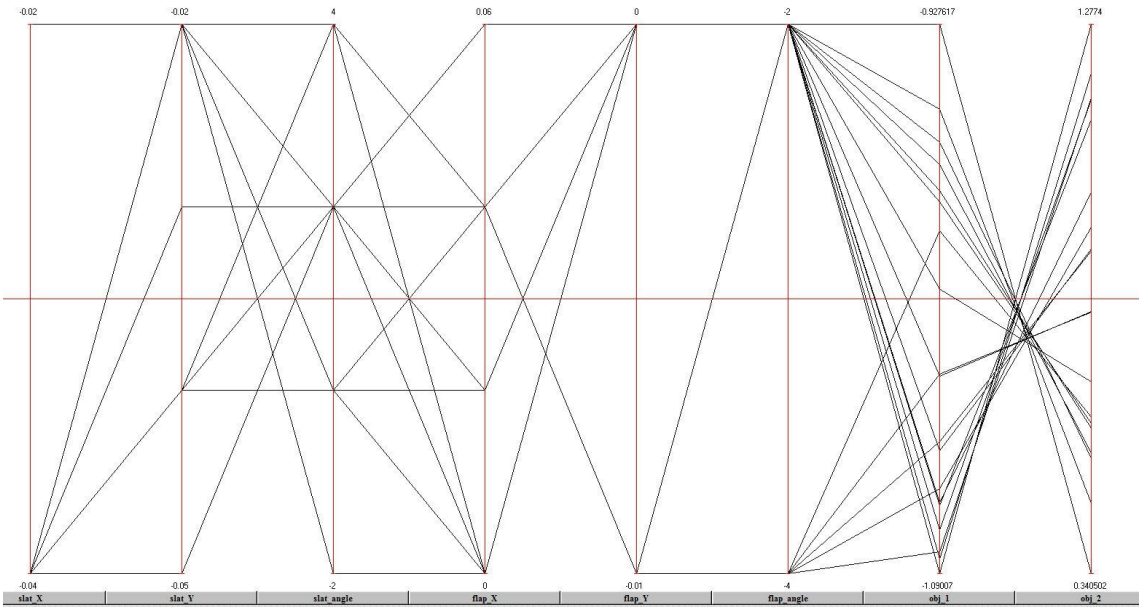

Figure 7.40: Representation of the Pareto front obtained in the second study with parallel coordinates

8 CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The studies developed in this work mean to carry out a robust optimization for an aerodynamic application, employing the multi-point method. The single-point optimization used in the previous MSc thesis projects has been taken as basis and it has been modified in order to develop the multi-point one. The optimization process includes different tools that are necessary in order to perform it. These tools are the parameterization script, the mesh generator, the CFD solver and the optimizer.

The parameterization code has been developed taking as example the previous work presented in [8]. It consists in a C++ code that allows the user to modify the relative positions of the elements that compose the airfoil, using a small number of design variables to identify the locations of the high lift devices. Another C++ code is then employed in order to check if the geometry created is feasible or if it presents overlapping.

The mesh is generated using the commercial software ANSYS ICEM CFD 14.0. Using the journal tool it is possible to make the mesh generation automatic and implement it in the optimization process.

The solver employed in order to find the solutions of the RANS equations is ANSYS FLUENT 14.0. It has been selected due to its versatility and possibility of analysing two-dimensional flow fields. Also this software allows the user to create a journal file that makes the CFD simulation a completely automatic process that can be implemented in the optimization loop. This step of the optimization is the most time consuming part of the process, especially in a robustness study, where multiple CFD evaluations are performed for each design vector.

The MOTS optimizer executes the research for the optimal configurations; the algorithm employed in this study is an adaptation of the one created by Jaeggi D. M. and modified by T. Kipouros. This C++ code has been adjusted in order to implement the tools described above and perform a multi-point optimization.

This optimizer adopts the Master and Slave configuration: the Slaves evaluate the objective functions, whereas the Master manages the entire process.

Two different studies have been performed in order to find the configurations of the Test Case; multi-element airfoil that optimizes the lift coefficient and minimizes the sensitivity to the variations in operating condition, which are limited to uncertainties on angle of attack and deflection of the flap in this work. The first approach analyses the flow for only three combinations of these operating conditions: the nominal one and two extremes which are supposed to be the ones that affect more the performance of the airfoil. The second study instead analyses all the combinations of angle of attack and deflection of the flap for each design vector evaluated. Both the studies have been run for a relatively short time; therefore a wide exploration of the design space has not been possible. Nevertheless, they have produced results that create a preliminary Pareto front composed by designs that improve the performance of the airfoil in comparison with the datum geometry. These new configurations lead to an increase up to the 11% on the c_l and a reduction up to the 70% on the sensitivity to the operating conditions. Moreover, both the studies have proved the reliability and the fast convergence of the MOTS.

The two studies have been compared, in order to realize a proper comparison it would have been necessary to run them for the same number of iterations but it has not been possible. However, some conclusion can be extrapolated from the preliminary comparison. Some analogies between the configurations that minimize the same objective functions in the two studies have been found. Furthermore, the second study shows that range of variation of the c_l for the same configuration, affected by all the combinations of uncertainties, is included in the interval, whose extremities are the lift coefficients corresponding to the extreme operating states selected in the first approach. This means that the first study evaluates the “worst case scenario” variations of operating conditions.

Finally, it can be conclude that: the first study is faster and can explore more the design space. The second one is more accurate and gives a better description of the variations of performance through all the variations of operating

conditions. Nevertheless it is too time-consuming in order to develop an exhaustive optimization process. For this reason the first approach can be employed in order to develop a preliminary research, whereas the second one is more suitable for accurate studies in a reduced design space.

8.2 Recommendations for Future Works

The outcomes of this work demonstrate how powerful the robust, and in particular multi-point, optimization is. Some improvements can be done in order to perform a complete optimization that explores the whole design space. Moreover this tool can be employed in order to develop new studies. Some suggestions can be made for the future works that can be developed:

- The C++ code that checks the geometry can be improved detecting, and marking as unfeasible, the valid geometries that can lead to failures in the mesh generation. In this way some time can be saved and the mesh failure can be avoided.
- The mesh generated around the airfoil can be improved either to have a better representation of the wakes and of the boundary layer or to have a faster convergence of the CFD evaluations.
- New design variables can be considered. The optimization process can evaluate also the change in shape of the elements instead of, or together with, the modification of the position of the elements. Nevertheless, it should be taken into account that the increase in the number of the design variables leads to a wider design space that requires more time to be explored.
- The objective functions can be analysed and a different definition of them might be done in order to improve the method.
- More uncertainties can be taken into account. At first all the three variables that identify the position of the flap (x, y and Θ) can be considered as uncertainty. Afterwards also the slat position can be set as variable condition. Moreover, uncertainties on the flow conditions can be added. Nevertheless, it is necessary to remember that the more uncertainties are considered the larger will be the number of

combinations that can be created with them; therefore, it would be advisable to make a preliminary study in order to analyse only the more interesting combinations.

- The multi-point optimization can be employed in order to execute robust multidisciplinary optimizations. It can be interesting to apply it to the previous works that evaluated the noise prediction or the structural stress on the wing.

REFERENCES

[1] M. Murayama and K. Yammamoto, “Validation of Flows on High-Lift Configurations by Structured- and Unstructured- Mesh Method,” in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2005.

[2] G. Trapani, T. Kipouros and A. M. Savill, “The Design of Multi-Element Airfoils Through Multi-Objective Optimization Techniques,” *CMES: Computer Modeling in Engineering & Sciences,* vol. 88, no. 2, pp. 107-140, 2012.

[3] A. M. O. Smith, “High-lift aerodynamics,” *Journal of Aircraft,* vol. 2, p. 501–530, 1975.

[4] K. de Cock, I. Lindblad and e. al., “2D Maximum Lift Prediction for the 59 Percent Span Wing Section of the A310 Aircraft,” GARTEUR, 1 June 1999.

[5] I. A. A. Lindblad and d. C. K. M. J., “CFD Prediction of Maximum Lift of a 2D High-lift Configuration.,” *AIAA,* pp. 99-3180, 28 June - 1 July 1999.

[6] G. Trapani, Multi-Objective Optimization of 2D high-Lift Airfoil Configurations using Tabu Search (MSc Thesis), Cranfield University, 2009.

[7] D. Capotorto, Multidisciplinary Design Optimization for Aircraft Configuration (MSc Thesis), Cranfield University, 2010.

[8] J. A. Exposito Carrillo, Aeroacustic Design Optimization for Take-off (MSc Thesis), Cranfield University, 2011.

[9] J. Samareh, “Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization,” *AIAA Journal,* vol. 39, no. 5, pp. 877-889, 2001.

[10] A. J. Keane and P. B. Nair, Computational approaches for aerospace

designs: the pursuit of excellence, Chichester: John Wiley & Sons Ltd, 2005.

[11] G. Farin, J. Hoschek and M.-S. Kim, Handbook of computer aided geometric design, Amsterdam: Elsevier Science B.V., 2002.

[12] T. Kipouros, Multiobjective Aerodynamic Design Optimization, PhD thesis, Cambridge University, 2006.

[13] J. Samareh, “A novel shape parameterization approach,” *NASA 1999-209116,* 1999.

[14] J. Wild, “Multi-objectiveconstrained optimisation in aerodynamic design of high-lift systems,” *International Journal of Computational Fluid Dynamics,* vol. 22, no. 3, pp. 153-168, 2008.

[15] J. F. Thompson, B. Soni and N. Weatherill, Handbook of Grid Generation, New York: CRC Press, 1998.

[16] ANSYS Inc., “ANSYS ICEM CFD 14.0 User Manual,” SAS IP Inc., 2011.

[17] ANSYS Inc., “ANSYS FLUENT 14.0 User's Guide,” SAS IP Inc, 2011.

[18] B. Jones, “Viscous Grid Spacing Calculator,” Nasa, 3 July 1997. [Online]. Available: http://geolab.larc.nasa.gov/APPS/YPlus/. [Accessed 5 June 2013].

[19] J. Tu, G. H. Yeoh and C. Liu, Computational Fluid Dynamics: A Practical Approach, 2nd ed., Oxford: Elsevier Inc, 2008.

[20] H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics, Second ed., Harlow: Pearson Education Limited, 2007.

[21] T. Kipouros, D. Jaeggi, W. Dawes, G. Parks, P. Clarkson and A. M. Savill, “Biobjective Design Optimisation for Axial Compressors Using Tabu Search,” *AIAA Journal,* vol. 46, no. 3, pp. 701-711, 2008.

[22] American Institute of Aeronautics and Astronautics, AIAA guide for the verification and validation of computational fluid dynamics simulations, Reston: AIAA, 1998.

[23] K. Deb, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons Ltd, 2001.

[24] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems for Optimization,” *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* vol. 1, no. 1, pp. 67-82, April 1997.

[25] C. A. Coello, “Evolutionary Multi-Objective Optimization: A Historical View of the Field,” *IEEE Computational Intelligence Magazine,* vol. 1, no. 1, pp. 28-36, 2006.

[26] D. M. Jaeggi, G. T. Parks, T. Kipouros and P. J. Clarkson, “The Development of a Multi-Objective Tabu Search Algorithm for Continuous Optimisation Problems,” *European Journal of Operational Research (EJOR) ,* vol. 185, no. 3, pp. 1192-1212, 2008.

[27] T. Kipouros, “Stochastic Optimisation in Computational Engineering Design,” *Advances in Intelligent and Soft Computing,* vol. 175, pp. 475-490, 2012.

[28] X. S. de Souza, J. Suykens, J. Vandewalle and D. Bolle, “Coupled simulated annealing,” *IEEE Trans Syst Man Cybern B Cybern,* vol. 40, no. 2, pp. 320-335, 2010.

[29] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on,* vol. 6, no. 22, pp. 182 - 197, 2002.

[30] M. Dodson and G. T. Parks, “Robust Aerodynamic Design Optimization,” *JOURNAL OF AIRCRAFT,* vol. 46, no. 2, pp. 635-646, 2009.

[31] R. W. Walters and L. Huyse, “Uncertainty Analysys for Fluid Mechanics with Applications,” NASA, Hampton, Virginia, 2002.

[32] D. Xiu and G. E. Karniadakis, “Modelling uncertainty in flow simulation via generalized polinomial chaos,” *Journal of Computational Physics,* vol. 187, pp. 137-167, 2003.

[33] D. N. Srinath, S. Mittal and V. Manek, “Multi-point shape optimization of airfoils at low Reynolds numbers,” *Computer Modeling in Engineering and Sciences (CMES),* vol. 51, no. 2, pp. 169-190, 2009.

APPENDICES

Appendix A Definition of the Objective Functions for the Second Study

Objective Function 1

$$\text{obj}_1 = -\frac{c_{l_{\alpha-1^\circ,\Theta+1^\circ}} + c_{l_{\alpha-1^\circ,\Theta}} + c_{l_{\alpha-1^\circ,\Theta-1^\circ}} + c_{l_{\alpha,\Theta+1^\circ}} + c_{l_{\alpha,\Theta}} + c_{l_{\alpha,\Theta-1^\circ}} + c_{l_{\alpha+1^\circ,\Theta+1^\circ}} + c_{l_{\alpha+1^\circ,\Theta}} + c_{l_{\alpha+1^\circ,\Theta-1^\circ}}}{\left(c_{l_{\alpha-1^\circ,\Theta+1^\circ}} + c_{l_{\alpha-1^\circ,\Theta}} + c_{l_{\alpha-1^\circ,\Theta-1^\circ}} + c_{l_{\alpha,\Theta+1^\circ}} + c_{l_{\alpha,\Theta}} + c_{l_{\alpha,\Theta-1^\circ}} + c_{l_{\alpha+1^\circ,\Theta+1^\circ}} + c_{l_{\alpha+1^\circ,\Theta}} + c_{l_{\alpha+1^\circ,\Theta-1^\circ}}\right)_0} + P$$

Objective Function 2

$$\text{obj}_2 = \frac{\left|c_{l_{\alpha-1^\circ,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{\alpha-1^\circ,\Theta}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{\alpha-1^\circ,\Theta-1}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a,\Theta-1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta-1^\circ}} - c_{l_{a,\Theta}}\right|}{\left(\left|c_{l_{\alpha-1^\circ,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{\alpha-1^\circ,\Theta}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{\alpha-1^\circ,\Theta-1}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a,\Theta-1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta+1^\circ}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta}} - c_{l_{a,\Theta}}\right| + \left|c_{l_{a+1^\circ,\Theta-1^\circ}} - c_{l_{a,\Theta}}\right|\right)_0} + P$$

Penalty Function

$$\text{P} = max\left[0, \frac{1}{2}\left(\frac{c_{d_{\alpha-1^\circ,\Theta+1^\circ}} + c_{d_{\alpha-1^\circ,\Theta}} + c_{d_{\alpha-1^\circ,\Theta-1^\circ}} + c_{d_{\alpha,\Theta+1^\circ}} + c_{d_{\alpha,\Theta}} + c_{d_{\alpha,\Theta-1^\circ}} + c_{d_{\alpha+1^\circ,\Theta+1^\circ}} + c_{d_{\alpha+1^\circ,\Theta}} + c_{d_{\alpha+1^\circ,\Theta-1^\circ}}}{\left(c_{d_{\alpha-1^\circ,\Theta+1^\circ}} + c_{d_{\alpha-1^\circ,\Theta}} + c_{d_{\alpha-1^\circ,\Theta-1^\circ}} + c_{d_{\alpha,\Theta+1^\circ}} + c_{d_{\alpha,\Theta}} + c_{d_{\alpha,\Theta-1^\circ}} + c_{d_{\alpha+1^\circ,\Theta+1^\circ}} + c_{d_{\alpha+1^\circ,\Theta}} + c_{d_{\alpha+1^\circ,\Theta-1^\circ}}\right)_0} - 1\right)\right]$$