

POLITECNICO DI MILANO



School of Industrial and Information Engineering  
Department of Aerospace Science and Technology

Master of Science in Aeronautical Engineering

## A fully automated optimisation procedure for multi-airfoil configurations based on CFD

*Supervisor:*

Prof. Maurizio Boffadossi

*Co-supervisor:*

Ing. Francesco Fabio Semeraro

*Author:*

Stefano Bortolotti 928238

Academic Year 2020/2021



# Abstract

Designing and optimising multi-airfoil configurations usually require expensive tests in the wind tunnel and CPU-costly CFD simulations. This work presents an accurate multi-airfoil optimisation procedure based on CFD with variable-fidelity algorithms to reduce the cost related to a pure CFD approach. Several past works available in the literature highlighted the feasibility of the procedure. However, no tool available on the market has neither implemented nor tested the optimisation of multi-airfoil wing configurations. In the few cases available, tools have been limited to individual profile optimization or shape optimisation, usually involving high computational costs.

The methodology proposed in this work consists of three blocks. The first one is about geometry and computational grid. Geometries can be generated either by using default airfoil parametrisations (i.e. NACA, IGP, etc.) or by providing a list of points through external files. Grid generation is the user's most time-consuming activity in CFD requiring experience, especially when complex geometries like multi-element airfoils are concerned. Many parameters such as wall refinement, wake refinement, farfield size, boundary layer and gap description require several tunings and are usually selected with a trial-and-error procedure. Consequently, the proposed automatic tool has been optimised on several configurations, generating efficient grids with limited user inputs and guaranteeing speed and accuracy on generated grids.

The second step is CFD simulation, whose setup has been developed on various multi-element configurations by making a thorough comparison between simulation results and experimental tests. Again, the user can customize the main features of typical CFD simulations: numerical schemes, CFL, turbulence model etc.

The last and main step is the optimisation procedure. The user can select different methods, including Particle Swarm Optimisation (PSO) and Steepest Ascent (or Descent) Optimisation. Several design variables are supported for configuration optimisation, such as relative positioning and angle of attack of each element. Furthermore, the tool allows a shape optimisation of IGP multi-element airfoils. The optimisation procedure supports multi-processing for distributed computing to take advantage of HPC resources. To carry out the optimisation without an excessive cost or to augmenting procedure robustness, preliminary evaluations exploit less expensive methods like Euler CFD or the Hess-Smith panel method. As for the latter, whenever the objective function is maximum lift the user can also consider the Valarezo-Chin criterion to evaluate a first optimal solution. Preliminary studies are intended to reduce the computational cost favouring convergence by reducing bounds and selecting a proper starting point. Another consequence of preliminary studies regards benefits about the robustness of optimisation procedure since non-interesting or spurious candidates will be automatically excluded from the design space.

The entire procedure is built in Python3 and connects open-source applications: SU2 for CFD simulations, Gmsh for grid generation, ParaView for post-processing, while

Matlab is called only for the Hess-Smith panel method preliminary study. The proposed procedure opens to several developments to guarantee higher accuracy and computational speed, but also to further improve the transfer of results to aeronautical and automotive applications: extensions to 3D wings with more complex features, new optimisation algorithms or further investigating the ones already applied are few simple examples of future potentialities.

The proposed optimisation results demonstrate the feasibility and potentiality of a fully automated CFD procedure: comparing with experimental data, the algorithms found optimal fitness and slot configuration with low errors.

# Sommario

La progettazione e l'ottimizzazione di configurazioni multi-profilo solitamente richiedono test costosi nella galleria del vento e simulazioni CFD costose per la CPU.

Il lavoro riportato in questa tesi propone un'accurata procedura di ottimizzazione per profili multipli basata su CFD con algoritmi a fedeltà variabile, volta alla riduzione dei costi relativi a un approccio CFD puro.

In letteratura diversi lavori precedenti hanno evidenziato la fattibilità della procedura. Tuttavia, nessun software disponibile sul mercato ha né implementato, testato o organizzato un codice finalizzato all'ottimizzazione di profili multipli. Nei pochi casi disponibili, gli strumenti sono stati limitati all'ottimizzazione del profilo individuale o all'ottimizzazione della forma, solitamente comportando elevati costi di calcolo.

La metodologia proposta in questo lavoro consiste in tre blocchi. Il primo riguarda la definizione della geometria e generazione della griglia di calcolo CFD. Le geometrie possono essere definite sia utilizzando parametrizzazioni predefinite di profili alari (es. NACA serie 4, serie 5, IGP, ecc.) sia fornendo un elenco di punti tramite file esterni. Quest'ultima opzione garantisce una maggiore libertà all'utente, non dovendosi limitare alle parametrizzazioni proposte di default.

Dal punto di vista pratico, la generazione della griglia è nota per essere l'attività richiedente più tempo in CFD all'utente, poiché esige ampia esperienza e disponibilità di tempo.

Tali esigenze diventano ancora più evidenti quando geometrie complesse come i profili alari multipli sono l'obiettivo dell'analisi CFD. Molti parametri come la rifinitura della griglia a parete, della scia, la distanza del farfield di griglia, lo strato limite e la descrizione del gap richiedono diverse regolazioni, richiedendo solitamente una procedura "trial and error" per la loro definizione. Aggiungendo la necessità di un'ottimizzazione della configurazione o forma del profilo, la necessità di rendere l'intera procedura automatizzata diviene obbligatoria. Lo strumento automatico proposto in questo lavoro è stato ottimizzato su più configurazioni, generando griglie efficienti con input limitati e garantendo velocità e precisione sulle griglie generate. All'utente è lasciata la libertà di regolare vari parametri di griglia in base alle necessità.

Il secondo passo è la simulazione CFD, il cui setup è stato sviluppato su varie configurazioni a profili multipli effettuando un confronto approfondito tra i risultati della simulazione e i test sperimentali presenti in letteratura. Anche in questo caso, l'utente ha la possibilità di personalizzare le caratteristiche principali delle tipiche simulazioni CFD: schema numerico, CFL, modello di turbolenza ecc.

L'ultimo e principale passaggio è la procedura di ottimizzazione, la quale richiama ed include quanto già citato concernente alla generazione delle griglie e alle simulazioni CFD. Sommando tutti questi passaggi nell'intera procedura, la necessità di automatizzazione diviene obbligatoria. Anche in questo caso, l'utente può selezionare diversi

metodi, tra cui il Particle Swarm Optimisation (PSO) e lo Steepest Ascent (o Descent). Anche in questo caso, l'utente ha la possibilità di gestire direttamente i parametri e caratteristiche del metodo di ottimizzazione scelto. Sono supportate diverse variabili di progettazione per l'ottimizzazione della configurazione, come il posizionamento relativo e l'angolo di attacco di ciascun elemento. Inoltre, qualora la forma scelta utilizzasse una parametrizzazione IGP, il codice proposto consente l'ottimizzazione di forma.

La procedura di ottimizzazione supporta simulazioni multi-processing, permettendo di sfruttare le risorse HPC e ridurre drasticamente i tempi di calcolo.

Per eseguire l'ottimizzazione senza costi eccessivi o per aumentare la robustezza della procedura, le valutazioni preliminari sfruttano metodi meno costosi come Eulero CFD o il metodo a pannelli Hess-Smith. In aggiunta, qualora il problema di ottimizzazione scelto dall'utente sia la ricerca di massima portanza o deportanza come funzione obiettivo, l'utente può considerare in alternativa il criterio di Valarezo-Chin con il metodo a pannelli Hess-Smith per lo studio preliminare.

Sfruttando queste procedure preliminari meno costose, l'utente identificare delle design variables come punto iniziale così come una riduzione dello spazio delle variabili, concentrando l'ottimizzazione nella regione più promettente e, consequentemente, favorendo la convergenza all'ottimo globale.

Un'altra conseguenza degli studi preliminari riguarda i benefici sulla robustezza della procedura di ottimizzazione, poiché i candidati non interessanti o spuri verranno automaticamente esclusi dallo spazio delle variabili.

L'intera procedura è costruita in Python3 e si appoggia a software open-source: SU2 per le simulazioni CFD, Gmsh per la generazione della griglia, ParaView per il post-processing, mentre Matlab è richiamato solo per eseguire lo studio preliminare del metodo a pannelli Hess-Smith.

Ulteriori sviluppi potrebbero fornire l'accesso a differenti software di generazione della griglia (SnappyHexMesh, Salome, GridPro ecc.) e a diversi software di simulazione CFD (come OpenFOAM e Fluent), garantendo all'utente una maggiore libertà di scelta. La procedura proposta apre infine a diversi sviluppi per garantire maggiore precisione e velocità di calcolo, ma anche per migliorare ulteriormente il trasferimento dei risultati alle applicazioni aeronautiche e automobilistiche: estensioni ad ali 3D con caratteristiche più complesse, nuovi algoritmi di ottimizzazione o approfondimenti su quelli già applicati sono alcuni semplici esempi di potenzialità e sviluppi futuri.

I risultati delle ottimizzazioni proposte dimostrano la fattibilità e potenzialità di una procedura CFD completamente automatizzata: comparando con risultati sperimentali presenti in letteratura, gli algoritmi trovano fitness e configurazioni per lo slot ottimali con errori minimi.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Multi-element airfoils . . . . .	1
1.2	Optimisation of multi-element airfoils . . . . .	6
1.2.1	Experimental procedures . . . . .	7
1.2.2	Computational procedures . . . . .	8
1.3	Novelty of the thesis . . . . .	11
1.4	Structure of the thesis . . . . .	13
<b>2</b>	<b>Computational model and theory</b>	<b>15</b>
2.1	Airfoil parameterisation . . . . .	15
2.1.1	NACA Airfoils . . . . .	15
2.1.2	IGP parameterisation . . . . .	19
2.1.3	Benzing Airfoils . . . . .	21
2.2	Navier-Stokes Equations . . . . .	23
2.3	Computational Fluid Dynamics . . . . .	24
2.3.1	Navier Stokes & Euler equations in SU2 . . . . .	24
2.3.2	Turbulence Model: Spalart–Allmaras (SA) . . . . .	25
2.3.3	Bas-Cakmakcioglu Transition Model (BC) . . . . .	26
2.3.4	Numerical Schemes . . . . .	27
<b>3</b>	<b>Mesh generation and CFD simulation procedure</b>	<b>29</b>
3.1	Grid module Codes' features . . . . .	31
3.1.1	Geometry . . . . .	31
3.1.2	Grid . . . . .	32
3.1.3	Interface with Gmsh . . . . .	38
3.1.4	Element Size correction . . . . .	40
3.2	NACA 0012 FT - Baseline grid CFD results . . . . .	41
3.3	Sensitivity Analysis - NACA 0012 FT . . . . .	49
3.3.1	Farfield Size . . . . .	49
3.3.2	Global grid refinement . . . . .	50
3.3.3	Wake: refinement size and length . . . . .	52
3.3.4	Ellipse Size and grid's external semi-circle . . . . .	54
3.3.5	Wall refinement . . . . .	59
3.3.6	Structured region analysis . . . . .	60
3.4	NACA 0012 FT - Final grid CFD results . . . . .	62
3.5	Other CFD results . . . . .	66
3.5.1	NACA 0012 with BC transitional boundary-layer model . . . . .	66
3.5.2	NACA 23012 . . . . .	69

3.5.3	NLR 7301 . . . . .	70
<b>4</b>	<b>Optimisation Module</b>	<b>75</b>
4.1	Optimisation methods: introduction . . . . .	75
4.2	Optimisation Algorithms . . . . .	78
4.2.1	Steepest Ascent (or descent) method . . . . .	78
4.2.2	Particle Swarm Optimisation (PSO) . . . . .	80
4.3	Preliminary Optimisation studies . . . . .	82
4.3.1	Experimental Designs for Fitting Response Surfaces . . . . .	82
4.3.2	Hess-Smith Panel Method . . . . .	84
4.4	Tool application . . . . .	86
4.5	Results of Optimisation Module . . . . .	89
4.5.1	Slot Optimisation of GA(W)-1 . . . . .	89
4.5.2	Comparison between PSO and Steepest Ascent results . . . . .	107
4.6	Shape Optimisation of IGP multi-element airfoil . . . . .	109
<b>5</b>	<b>Conclusions</b>	<b>117</b>
<b>A</b>	<b>Appendices</b>	<b>121</b>
<b>A</b>	<b>Tool Design</b>	<b>123</b>
A.1	GA(W)-1 slot optimisation . . . . .	125
A.1.1	Points coordinates from external files . . . . .	125
A.1.2	Example of "main_config_opt" used for slot optimisation . . . . .	127
<b>B</b>	<b>Discussion about <math>y^+</math> and wall spacing</b>	<b>129</b>

# Chapter 1

## Introduction

### 1.1 Multi-element airfoils

This research aims to extensively simulate a wing optimisation procedure based on CFD analysis for multiple airfoils.

From a general point of view, multi-element airfoils work to improve lift performances of the wing or body under analysis, trying to overcome the aerodynamic limits of a single-element configuration, without affecting dimensions or other performances as much as possible. Historically the aeronautical sector has lead all the main studies, researchers and discoveries about airfoil aerodynamics.

The analysis began from single element wings' configurations for aircrafts, trying to improve lift performances in particular and demanding flight conditions, like take-off and landing. To reach maximum lift for a single element configuration, the research led to an optimal pressure distribution over the airfoil surface respecting geometrical limits imposed by normative, regulations or respective structures.

With an appropriate shape design of the airfoil, performances about maximum lift can be drastically augmented. In particular, these improvements could be reached by properly designing the mean camber line of the wing or by curving the rear portion of the section [1] [2].

Anyway, single-element wings do not provide the possibility to change configuration and setup when the vehicle is operative. This limitation becomes relevant for racing cars, for which there is a wide variability of circuit and flow conditions.

In these cases, the research of a proper and high-performing setup strictly depends on the local conditions; as a consequence, the availability of wide setup options becomes even crucial in some sectors of applications.

Moreover, many physical phenomena act to limit the continuous increase of maximum lift, like boundary-layer separation, the onset of supersonic flow etc. [3]. Boundary-layer separation usually is the main critical factor for limiting maximum lift; when the flow is two-dimensional, the most affecting factors can be traced back to the quality and the adversity of pressure gradient; another cause of separation could be boundary-layer's kinetic-energy defect at the start of the adverse pressure gradient.

The general intention is to delay the separation as much as possible: as a consequence, the ideal condition is having separation loci coincident with the airfoil's trailing edge. As reported by Houghton et al. [3], this location depends on two main factors:

- boundary-layer thickness at the beginning of adverse pressure gradient;

- Reynolds number per unit length written in the form of  $U_m/\nu$ , where  $U_m$  is the corresponding local velocity at the edge of boundary layer; from this condition, the general rule says that having a higher cited ratio the boundary layer will be able to sustain higher values of pressure coefficients.

From the point of view of the energy defect, similar reasoning leads to state that the boundary-layer can develop farther in adverse pressure gradient before separating if it is thinner at the point of maximum depression. All these notes suggest the importance of predicting the separation locus of boundary-layers and its relation with the point of maximum depression.

Various separation locus' predicting methods have been developed throughout the XX century, as deeply reported by Cebeci et al. [4].

Although predictions and investigations about boundary-layer behaviour and separation have continuously improved the knowledge about how to define the shape of a single-element airfoil to augment the maximum lift achievable, other ways have been considered all along the last century: among these solutions, multi-element configurations may occupy the most important role thanks to their high benefits on the performances, in particular exploiting high-lift and delaying separation.

The origins of multi-element configurations date back to the early XX century with the works of H. Page (Britain) and K. Lachmann (Germany) [3]. Like many other scientific deductions, the basic idea was suggested by observing the surrounding nature: in this case, multi-elements' configurations were inspired by birds' wings which usually present feathers attached to the thumb bone.

About aircrafts, since last century until today, the usage of multi-element configurations had the main objective to improve the performance by augmenting the wing lift while maintaining sufficient aircraft speeds, especially in particular flight conditions like take-off and landing [1].

Another important sector where knowledge development about multi-element configurations has been important and thriving is the automotive sector, especially referred to racing or high-performance cars.

Ing. E. Benzing [2] summarized the historical development of multi-element in the racing cars' sector. Interest in these configurations continuously increased with the improvement of theoretical aspects, positive practical and experimental results in the following years. Nowadays, applications of multi-element airfoils are wide and cover various sectors. Anyway, the physical principles are the same regardless of the application sector.

The Complexity of the flow that impinges a multi-element airfoil increases drastically with respect to single-element airfoils due to the interference and dependence of respective boundary-layers and wake of the elements (see an example in Figure 1.1) During the last century various types and configurations have been provided, but a general and common definition can be found since the aerodynamic behaviour of these various types can be summarized under the same principles: a multi-element airfoil is an airfoil composed of two or more close and coupled elements, with the main aim of increasing lift coefficient of the entire wing.

Various explanations suggest why multi-element configurations have large aerodynamic benefits on maximum lift coefficient. Prandtl provided one of the most conventional explanations about multi-element configurations: the benefits come from the slot which "blows" into the upper surface's boundary-layer providing momentum to its particles. As a consequence, particles recover velocity, previously decreased by viscous effects.

As again reported and explained by E.L. Houghton et al. [3] and widely experimentally verified (for example, by Nakayama et al. [5]), this conventional explanation to the phenomena is limited and mistaken since:

- upper and lower wing surfaces have the same stagnation pressure, then the slot cannot act like a nozzle and cannot provide higher energy to upper surface.
- the slot does not increase the flow velocity with respect to upper surface. On the contrary, the slot causes a reduction of the flow velocity over the main airfoil's upper surface [5].

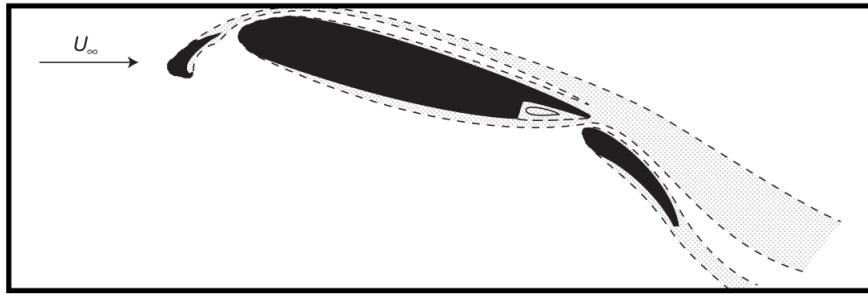


Figure 1.1: Typical boundary-layer behaviour for a multi-element configuration [3], reconstructed from experimental results of Nakayama et. al. [5].

Looking at the historical development of multi-element configurations, various options and setup have been developed trying to enhance and augment the lift performance of a specific wing.

Consequently, wing shapes' complexity increased, causing also an increment on mechanical systems' complexity to control the multi-element configurations, and costs as a consequence. Figures 1.2 and 1.3 reports various examples of upstream and downstream elements types (flap).

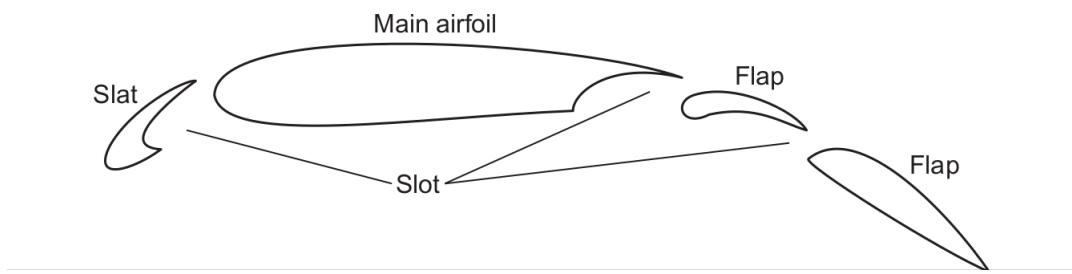


Figure 1.2: Typical multi-element airfoil scheme [3].

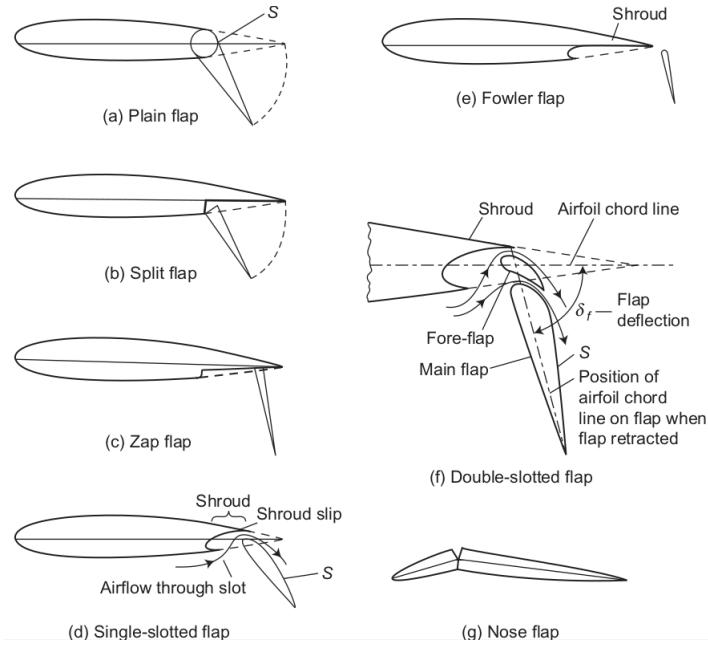


Figure 1.3: Flap Types [3].

Although these wide availability of options, two main elements' types ensemble the majority of possibilities not only by a geometrical but also aerodynamic point of view. The so-called *slats* are the upstream elements with respect to the main wing's airfoil, while the so-called *flaps* are downstream elements.

As deeply described and analysed by Smith [6] (1974), effects of slats and flaps can be directly explained by considering vortices in their respective places (Figure 1.4, 1.5).

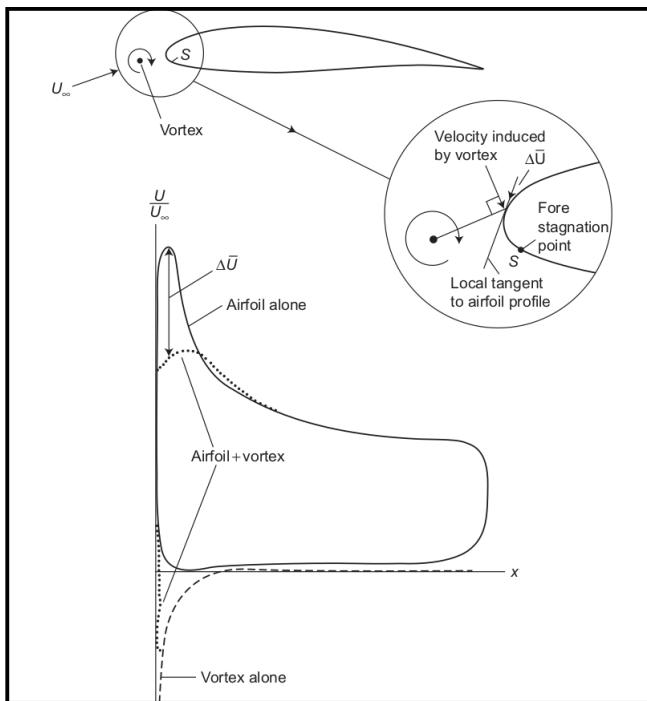


Figure 1.4: Slat effect [3].

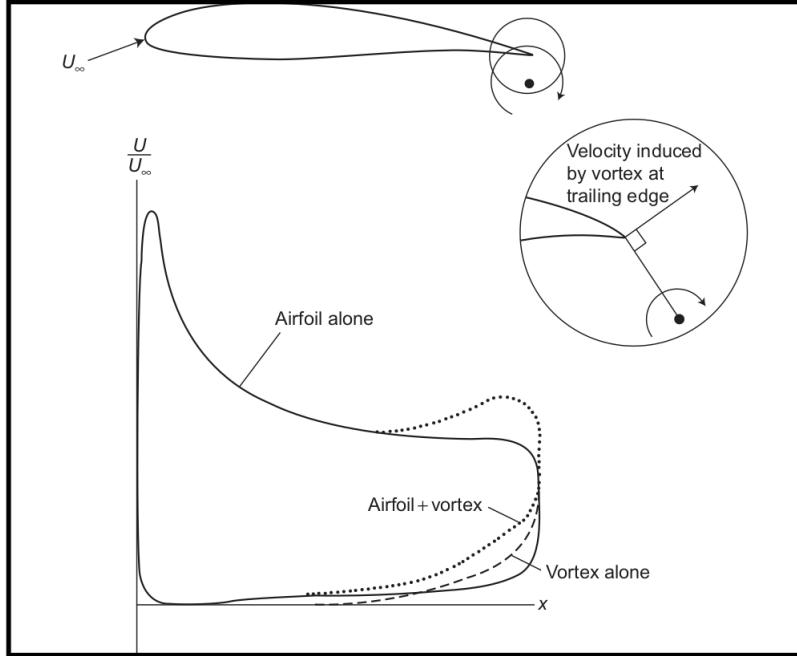


Figure 1.5: Flap effect [3].

The slat's presence causes a reduction of the adverse pressure gradient strength by reducing the boundary-layer velocity at the edge on the upper surface. The opposite happens on the lower surface of an aeronautical multi-element airfoil with a slat, where the velocity increases with respect to an airfoil without a slat system.

Since the effect on the lower surface is generally predominant, the lift coefficient increases with respect to a configuration without the presence of a slat, along with an increase of stall angle of attack.

Similarly, the flap presence leads to an increase of velocity on both surfaces, which becomes beneficial especially on the trailing edge of the main (or previous) airfoil, guaranteeing again a reduction in severity of the adverse pressure gradient. Also in this case the increase of lift can be observed: in particular, the pressure around the trailing edge of the main element is elevated as compared to a single-element configuration. The effect is due to the respective interference between two adjacent elements: in particular, the upstream element's trailing edge is located in a region of increased velocity (inclined to the mean line at the rear of the upstream element itself), related and caused by the presence of the downstream one.

The reasons can be deduced by recalling the Kutta condition, which must be verified also in the presence of the flap: this element in fact changes streamlines on upstream airfoil's trailing edge. As a consequence, viscous effects guarantee higher circulation satisfying Kutta condition, increasing the lift [3] [6] [7]. This effect is called *circulation effect*, as a consequence.

A possible disadvantage of flap usage in multi-element configurations is the slight reduction of stall angle of attack. Anyway, the presence of both slat and flap in the configuration guarantees an increase of stall angle of attack along with the maximum lift. Another benefit related to multi-element configurations regards the interaction between the element's wake and downward element boundary-layer. Considering aeronautical multi-airfoils as a reference, with proper configuration designs the slat's wake and main airfoil's boundary-layer do not interact. As a consequence, the slat's wake will interact

and decelerate downstream in the region of adverse pressure gradient above the main airfoil. The so-called *recovery* (or total velocity deceleration) of a multi-element airfoil takes place in stages along boundary-layer edge [3], delaying again flow separation as a consequence. Analogously it happens between the main element and downstream flaps. Similarly, the downstream elements appear to be "protected" from stall thanks to the presence of the previous elements [8]: upstream elements cause a downwash on the flaps, especially at relevant angles of attack close to stall. This effect is intuitively more relevant when the configuration presents upstream elements with high relative chord values.

## 1.2 Optimisation of multi-element airfoils

The following summary about optimisation of multi-element airfoils will present the problem from a generic point of view (focusing on design variables, objective function and constraints), and subsequently describing recent relevant works.

Considering an aircraft, the design project fitness could be one among several (minimum costs or maximum profits, minimum total weight etc.); anyway, as already introduced in the previous section, the objective function for a purely aerodynamic problem is increasing maximum reachable lift of a specified wing, especially when dealing with multi-element configurations. Referring to the aeronautical sector, this optimisation becomes crucial for take-off and landing aircraft's manoeuvres.

Along with maximum lift, another important objective can be maximizing efficiency; for an aircraft, maximum efficiency is sought during climbing manoeuvre [9].

Except for particular cases, the great majority of sectors where multi-element airfoils are adopted will require maximizing lift or efficiency. In some cases (like wings for racing cars) downforce or minimum negative efficiency become the target [2], but the principles do not change.

Defined the optimisation objective function, the design variables have to be imposed. Usually, they are some between the following:

- shape of each element;
- relative positioning ((x,y,z) for tridimensional problems) between each couple of elements;
- relative angles of attack;

Besides shape and relative angle of attack which characterize also single-element configurations' optimisations, optimisation of multi-element usually focus on the *slot* between each element. The slot between each couple of elements is usually defined by the *gap* and *overhang* (or *overlap*), visually reported by Figure 1.6 from Lei et al. [10].

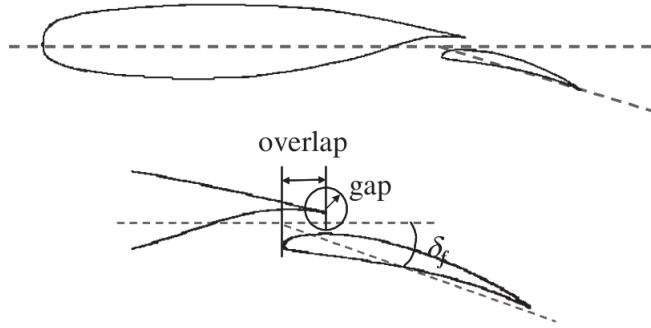


Figure 1.6: Definition of *gap* and *overlap* for multi-element airfoils configurations [10].

Along with gap and overlap representation, the figure reports also the definition of ( $\delta_f$ ), which is the relative angle of attack between an element and the upward one. The same reference was considered in several other works [7] [9] [11] [12] [13].

About the constraints applicable to optimisations' problems about multi-element airfoils, they mostly concern the obvious geometrical limitations. Anyway, the domain can be restricted by experience and excluding various configurations which will not provide an optimal solution. For example, whenever the flap is too far from its upward element, downwash induced by the latter does not protect the flap from stall at high relative AoA and does not guarantee the benefits of multi-element configurations in terms of maximum lift.

Other constraints can be imposed on secondary aerodynamic coefficients, either involved or not in the objective function definition. For example, whether maximum lift coefficient is the target of the optimisation, constraints can be imposed to limit drag and moment coefficients.

### 1.2.1 Experimental procedures

After purely theoretical and analytical analyses, the first practical optimisation procedures on multi-element configurations were historically performed in the wind tunnels. Various attempts and algorithms were applied for the optimisation of multi-element configurations in the past. Since CFD procedures were not available yet in the XX century, the typical analysis was firstly executed using wind tunnel tests.

When the experimental optimisation regards aerodynamics, the cost of each simulation becomes more relevant by both time requested and economical point of view due to the typical features of wind tunnel's experiments. Consequently, experimental optimisations regarding aerodynamics cannot rely on a high total amount of experiments in the wind tunnels.

Historically, the first methods to determine optimal gap and overlap of a multi-element configuration were based on trial and error procedure ones with a matrix of test cases generated by simple computational results [12], or from existing databases. These methods were essentially limited, not accurate or expensive.

To overcome these limitations, researchers adopted the already existing mathematical optimisation algorithms to perform the slot optimisation of multi-element airfoils: each candidate considered during the wind tunnel optimisation procedure involves a fitness evaluation through experiment. Among the several available aerodynamic optimisations'

algorithms, the ones based on a relevant amount of candidates (like Genetic Algorithm and Particle Swarm Optimisation) have to be excluded due to the high costs of wind tunnel tests.

To meet experimental requirements, low costs and high accuracies, "Design of Experiments" theories were developed and consequently applied also to experimental wind tunnel's optimisations.

Various methodologies and theoretical aspects were summarised by D.C. Montgomery [14].

To avoid high wind tunnel costs, D. Landman [7] proposed various gradient methods for multi-element configuration optimisation:

- *steepest ascent (or descent) method;*
- *fixed-size sequential simplex optimization;*
- *variable-size sequential simplex optimization.*

These methods are simple and were introduced by Landman to demonstrate the feasibility of using optimisation algorithms for multi-airfoils; they can be extended to more complex or hybrid methods.

### 1.2.2 Computational procedures

Attempts to bring optimization of multiple profiles using computational procedures have been performed on a few occasions from the last years of XX century.

One of the first important studies were performed by NASA and reported by S.M. Klausmeyer and J.C. Lin [12] (1997); the results were also summarised by D. Landman [7]. In this work, various attempts were performed by 13 researches trying to optimise a 30P-30N multi-airfoil configuration through CFD using RANS or coupled methods. The main results were gathered by Rogers, who participated in the program [7]:

1. when the user is comparing CFD results with experimental data, it is better to include wind tunnel walls into the computational model;
2. three-element geometries can be handled by most grid approaches;
3. resolution of the grid is important;
4. grid's solution adaptation to shear layers and wakes is needed;
5. transition models are still not reliable;
6. 2D modelling becomes useful for trends' identification only.

Another important result is the difficulty to predict drag polars due to the high complexity of the configurations under analysis.

A conclusion of the cited report was that wind tunnel's experiments were still necessary, since CFD procedures were still not that precise and powerful to guarantee accurate results. Anyway, CFD methods applied to multi-element airfoils presented also useful and positive results.

Firstly, they can perform and analyse a preliminary study about the gap and overhang dependence. Secondly, the dependence on freestream conditions (especially Reynolds number) can be observed, and consequently act as a guideline for experimental tests

reducing also wind tunnel occupancy time during optimisation [12].

More recent studies have been conducted using CFD to optimise the shape and configuration of multi-element airfoils.

Kim et al. (2004) [11] proposed the viscous continuous adjoint method to execute design optimisations, firstly as configuration optimisation on a 30P30N multi-element airfoil, secondly as shape optimisation on a single-element RAE2822 (both at high Reynolds numbers). Using FLO103-MB (multi-block RANS flow solver) with SA turbulence model, the procedure highlighted how CFD procedures along with adjoint method can be applied for optimisation of airfoils. In particular, the single-element configuration showed significant improvements, while the multi-element design case improvement was smaller since the baseline was already highly optimised.

A similar shape optimisation on single element configurations and multi-objective problems was performed by Arias-Montaño et al. (2012) [15] using MOEA (an open-source evolutionary computation library for Java that specializes in multi-objective optimization), but here using and developing multi-surrogate approach; results show convergence speed up by reducing objective function evaluation with respect to the same not using surrogates).

Another example of CFD application and optimisation procedures on multi-element configurations is Trapani et al. [16], then developed also for flexible wings [17] [18]. The obtained results highlighted again the potentiality of CFD solvers along with optimisations' algorithms to seek high-lift optimum of a given multi-element configuration.

In 2017 the DeSiReH (Design, Simulation and Flight Reynolds Number Testing for Advanced High-Lift Solution) project, funded by European Commission under the 7th Framework Program, gathered different flow models dimensionality, meshing techniques, geometry parametrization optimisation strategies and lastly comparing trends, advantages and disadvantages of the approaches used, trying to select efficient numerical design strategies for high-lift devices. The research is summarised by Iannelli et al. [19]. This collaboration provided various software, optimisation techniques and possibilities of analysis, thanks also to in-house software. This research demonstrated that nowadays it is possible to create the setup to face also complex 3D optimization frameworks in an industrial context, then secondly performing robust multipoint shape and configurations' optimisations for multi-element devices using CFD.

The complexity of three-dimensional problems was faced by adopting a particular correlation between 2D results and 3D wing performances, proposed and demonstrated by Wild et al. [20] years before (2009). The results did not present relevant fitness improvements for the selected configuration (which was a 3D full span slat and flap DLR-F11 (KH3Y) wing-body HL configuration). Anyway, the reasons for low efficacies can be pointed to baseline industrial configurations which were already well-designed, but also due to the flatness of objective function considered.

The support of the industrial sector also delineated that is preferable having tools which guarantee robust techniques for high-lift devices, instead of extremely precise ones. Moreover, CFD optimisation procedures can become very useful when companies are dealing with new geometries or configurations or preliminary studies about, reducing the amount of wind tunnel usage and also overall project lead-time.

In recent years, the industrial market is continuously tending to more efficient and less expensive procedures. Computational simulations are capable to satisfy these requirements.

A particular but significant example of this trend derives from the racing automotive

sector of Formula 1: starting from the present year (2021), the participating teams are bounded by the so-called "budget cap" to reduce expenses. As a consequence, various observers predict the teams will probably move on less expensive procedures.

Pat Symonds, an English engineer and technical director who worked in prestigious teams in the past (Toleman, Benetton, Renault, Marussia, Williams) and now working for FIA (Federation Internationale de l'Automobile), commented on these new rules: *"The wind tunnel has been an essential tool for the development of Formula 1 for over 50 years. [...] But it is since the 1980s that its use has become crucial, and since then the sophistication has grown enormously. [...] It might come as a surprise, therefore, to know that eight of the ten teams currently entered in the championship were in favour of the future goal of eliminating the use of tunnels within 10 years."*

Symonds points to the costs of wind tunnels, updated models and new rules of so-called "budget cap" as the main reasons for this possible historical change in automotive aerodynamics.

As he continues: *"Building a model from scratch costs almost six hundred thousand euros, such is the level of the instrumentation; [...] The tunnels have also grown in size and use of energy, and this obviously has not been achieved economically. The electricity bill for a tunnel can exceed one million euros per year"*. Symonds ended by indicating the CFD as the tool which will mostly replace wind tunnels' experiments.

The topic is still controversial and under discussion, since not all participants and technicians around the Formula 1 circus are not looking at these changes favourably; however, interest and attention to this topic are continuously growing.

Although the cited reference is referred to a particular sector, it is significant how a highly competitive and forefront sector like Formula 1 is pointing towards CFD simulations as a preferred design tool, reducing commitments of experimental wind tunnel tests.

### 1.3 Novelty of the thesis

The purpose of the present thesis is to create an open-source tool for multi-airfoil configurations' optimisation based on CFD. The importance and advantages of computational procedures were already presented in the previous section. In the following, the main potential advantages are summarised:

- decreasing of wind tunnel tests by adopting an automatic grid generation, CFD and optimisation saving a huge amount of user's time;
- like many other aerodynamic problems or sectors, CFD can avoid the usage of wind tunnel and experimental tests when the project, developments or studies are at the preliminary stages; as a consequence, costs can be reduced along with the time requested.
- CFD simulations would be able to use the same algorithms already adopted in experimental wind tunnel optimisations, but can also be extendable to more demanding optimisation algorithms (like Genetic Algorithm, Particle Swarm Optimisations etc.) from which more precise results could be obtained.

The previous works were focused only on creating grid and CFD optimisation procedures for a specific multi-airfoil geometry: there is no extension and possibility for the user to study and optimise custom geometries. Moreover, most of the previous works were executed using industrial in-house software or not free scripts.

Analysing freely software available about CFD and optimisations, SU2 provides an optimisation procedure based on adjoint, but only addressed to shape optimisation. The adjoint method has been demonstrated to provide satisfactory results [11], but it could require a significant amount of iterations to reach convergence. An alternative procedure is proposed: using the same fully automatic optimisation CFD procedure adopted for slot configuration, IGP parametrisation [21] is considered thanks to its flexibility in reproducing a wide amount of airfoils' shapes.

The purpose of the tool that has been developed is to generate a proper grid for the selected geometry, initialize a CFD simulation and iteratively search for an optimum through optimisation algorithms, repeating the grid generation step and CFD calculus for every candidate. The aim is to highlight the feasibility and wider usage of fully automated CFD procedures.

A useful consequence of making available the cited scripts is also the possibility to fast and easily generate a grid for a single CFD simulation purpose (without initializing an optimisation procedure), which again could save much user time. Nowadays on the market there are numerous open-source solutions to perform CFD simulations. Anyway, they usually provide one of the following typical services and the user has to couple them to complete a CFD simulation:

- grid generation;
- numerical computation;
- post-processing.

It is then up to the user learning, managing, interacting and finally performing each step through different software. Even considering an expert user with good knowledge of the programs mentioned, some difficulties can be encountered also with simple problems

and simple geometries (like a single-element geometry impinged by a stationary, non-separated flow) along with typically high time requested by CFD simulations. Then, the amount of time required drastically increases for complex geometries especially, furthermore when optimisation procedures are taken into account.

A typical aerodynamic complex computational example is multi-element airfoil configurations, which add new variables to be managed (gap, overlap, relative angle of attack and relative positioning).

In these cases, not only the increased geometry complexity has a significant role: as already cited in Section 1.1, the flow around each element becomes more complex, as wake. From the computational point of view, these geometrical and physical complexities affect the grid, whose generation is already the most critical and time-requesting section of the CFD procedure when the user is dealing with viscous flows. Besides a natural increase in the number of cells due to multi-elements, refining must be located properly in the domain for an accurate prediction of the flow and the aerodynamic performance.

When the user is dealing with optimisation procedures, the grids' features have to be suitable for each candidate of the optimisation procedure. Scott and Haftka [22] indicated the fitness evaluation for the single candidate as the most deciding factor in optimisation problems; moreover, other specific issues related to CFD optimisations problems are all related to the precision and suitability of the grid, as summarized by Rogers [12].

The purpose of this thesis is to provide a fully automatic procedure for optimisation of multi-element airfoils, overcoming the limitations and disadvantages of manual procedures and demonstrating the feasibility of optimising with CFD. The optimisation methods adopted during the present thesis will be examples of the procedure potential, suggesting some possible future extensions.

Since the tool is based on open-source software, all the cited advantages become firstly useful for academic usage; however, extensions and knowledge improvements about this topic can become useful and interesting also for industrial applications: a robust, reliable and efficient computational solution can drastically reduce the investigations in the wind tunnel, reducing overall costs.

## 1.4 Structure of the thesis

Chapter 2 deals with **Computational model and theory**, where several theoretical aspects adopted for the present thesis' results will be briefly introduced. It includes parametrisations, numerical schemes, optimisation algorithms etc. Chapter 3 is **Grid and CFD Module**, where a deep analysis of grid generation and CFD run is presented. All the details about grid management are reported, especially referring to the available user customization. Secondly, an extended grid sensitivity analysis for a single element geometry is reported and discussed, followed by some applications to single or multi-element airfoils. The focus will be on accuracy with respect to experimental data and convergence speed, which takes an important role in an optimisation procedure. Chapter 4 regards **Optimisation Module**, where several test cases procedures are presented. Both slot and shape optimisations will be tested, focusing again on the post-processing analysis on computational performances and accuracy.

In Chapter 5 the **Conclusions** of the work are reported. Main observations and results are summarized in this chapter, underlining features and potentialities of the procedure (especially about optimisations).

Appendix A deals with **Tool Design**, where the current structure of the tool is presented.

Appendix B regards **Discussion about  $y^+$  and wall spacing**. Since grid sensitivity and literature highlight the importance of grid structured region, a brief analysis about some tested configurations (NACA 0012, NACA 23012, NLR 7301) is performed focusing on  $y^+$  CFD results. In particular, this appendix provides hints for user's input about wall spacing, by comparing inputs values with actual  $y^+$  CFD results.



## Chapter 2

# Computational model and theory

The present chapter provides a brief recap of the various computational methods, schemes and algorithms adopted throughout the work: the required theoretical background is provided as a quick reference for understanding the choices and the tools developed.

### 2.1 Airfoil parameterisation

This section is intended to describe the default parameterisations available apart from the possibility for the user to provide external files containing airfoil's points. Within the tool, the implemented parameterisations are:

- NACA airfoils, including 4-digits, 5-digits, 4-digits modified and 16-series;
- Benzing airfoils, since common choices for high performance racing cars in the past;
- IGP airfoils, which is a modern parameterisation that has in flexibilit and robustness its main advantages.

#### 2.1.1 NACA Airfoils

The following notes about NACA 4-series, NACA 5-series, NACA 4-series modified and NACA 16-series are based [1] [23] [24] [25] [26], summarized by NASA [27]. Anyway, the first report published by NACA which gathered the first information about airfoil parameterization was *NACA Report No.460* by Jacobs et al. [28] in 1933. All the following information can be directly found into each cited papers, articles and reports in a more detailed manner.

During the II World War various families of wing sections were systematically tested by several research centers, but NACA investigations were the most relevant. This parameterisation is based on a separation of airfoil's camber (or mean) line and thickness distribution effects [1]. The method for geometry generation was based on combining mean lines and thickness distributions to obtain the desired cambered wing section. Leading and trailing edges result respectively as the forward and rearward extremities of the mean line, while the chord line is a straight line which connects the two edges. The following relations define upper and lower-surface coordinates (subscript  $U$  and  $L$ ,

respectively):

$$\begin{cases} x_U = x - y_t \sin(\theta) \\ y_U = y_c + y_t \cos(\theta) \end{cases} \quad (2.1)$$

$$\begin{cases} x_L = x + y_t \sin(\theta) \\ y_L = y_c - y_t \cos(\theta) \end{cases} \quad (2.2)$$

where  $x_U$  and  $y_U$  indicates the abscissa and ordinate of a point located on upper surface, while  $x_L$  and  $y_L$  for the lower surface;  $y_t$  represents the ordinate of the symmetrical thickness distribution at chord-wise position  $x$  and  $\tan(\theta)$  is the slope of the mean line with  $(\theta)$  the respective angle.

Consequently, this angle  $(\theta)$  can be found as:

$$\theta = \arctan \left( \frac{dy_c}{dx} \right) \quad (2.3)$$

Once thickness distribution and camber relations are given, upper and lower surfaces of the airfoil can be found.

### NACA 4-digits

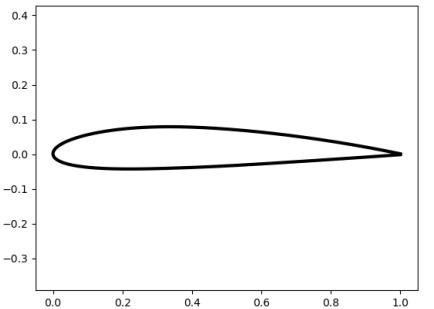
NACA 4-digits is the most known parameterisation in aerodynamics, since it easily provides various geometries by simple analytical relations. Although dated and with a limited geometrical variability (defined by only 3 parameters), NACA 4-series airfoils still remain a very simple parametrization still used for test cases thanks to the wide availability of experimental data.

The parameters which represent the main features of a NACA 4-digits airfoil are the following:

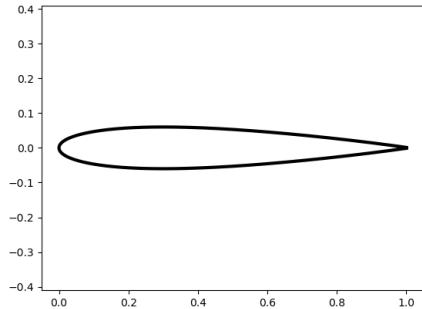
1. maximum camber (as percentage of the chord);
2. distance of the maximum camber from the airfoil's leading edge (in tenths of the chord);
3. maximum thickness of the airfoil (as percentage of the chord, defined by two digits).

for which four digits are obtained.

For example, NACA 2412 presents a maximum camber of 2% located at 40% of the chord from leading edge, with a maximum thickness of 12% of the chord. As reported by the parameters' definition, NACA 4-series guarantees the possibility to build symmetric and asymmetric airfoils: for example, NACA 2412 and NACA 0012 have the same thickness (12% of the chord), but the first uses camber along the chord (2% located at 40% of the chord), generating a cambered airfoil. Figures 2.1a and 2.1b visually report the respective shapes, with a reference chord equal to one.



(a) Visualisation of a NACA 2412.



(b) Visualisation of a NACA 0012.

### NACA 5-digits

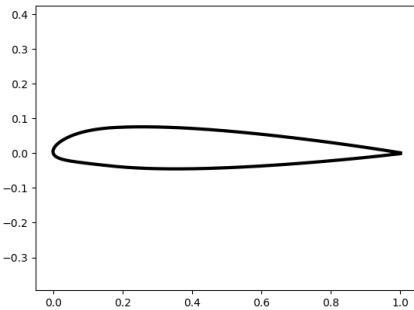
First observations about NACA 4-digit indicated that the maximum lift coefficient increases considering a shifting (either forward or aft of mid-chord position, approximately) in the position of maximum chamber. The geometry shape equations follows from these observations.

This new type of geometry definition needs 4 parameters (instead of 3 needed for NACA 4-digits):

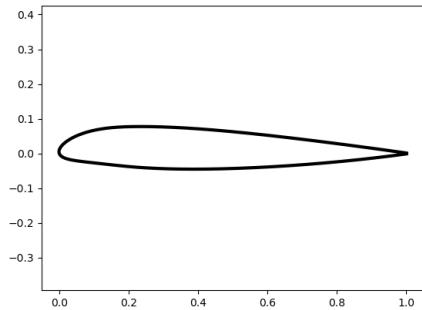
- the first, second and third parameter (first three digits) define the mean line ( $y_c$ );
- the fourth parameter (last two digits) define again the thickness (like for NACA 4-digits), such that  $t$  is equal to the value of these two digits divided by 100.

Thickness distributions ( $y_t$ ) of NACA 5-digits sections are equal to the ones presented previously for NACA 4-digits.

Changes are related to equations of mean line, which can be subdivided in two groups depending on the input values: with **non-reflexed** and with **reflexed camber line**. The difference is indicated by the third digit, which can take only value 0 (non-reflexed) or 1 (reflexed camber line). Figures 2.2a and 2.2b show NACA 23012 and NACA 23112, which visually exemplified the differences between non-reflexed and reflexed camber line's airfoil.



(a) Visualisation of NACA 23012.



(b) Visualisation of NACA 23112.

## NACA 4-digits modified

The family of 4-digits airfoils received various modifications, especially from thickness' studies. In particular, an objective was trying to change the normal position of the maximum thickness (allocated at 0.30 chord for previous NACA airfoils).

An important family of airfoils derived from these modifications were the so-called NACA 4-digits modified. These airfoils allow variations on leading edge radius and location of maximum thickness.

The numbering system is defined by:

NACA MPXX-IT

where MPXX is the standard designation used for NACA 4-digits, while the last two (IT) describe the modification to thickness distribution.

These new digits' respective definition is the following:

- $I$ : leading edge radius' designation;
- $T$ : chord-wise position of maximum thickness (in tenths of chord)

The camber line ( $y_c$ ) of these airfoil remains the same of standard NACA 4-digits.

What changes is the relation about thickness distributions ( $y_t$ ), which directly depends on the new input digits ( $I, T$ ).

For example: NACA 0012-74 means an uncambered 12% thick airfoil with a maximum thickness at  $x/c = 0.4$ ; moreover, leading edge radius is 0.0216, which is 36% larger than the standard 4-digits value. The example is reported in Figure 2.3.

Note that the value of  $I = 6$  provides leading edge radius of the standard 4-digits airfoils [26].

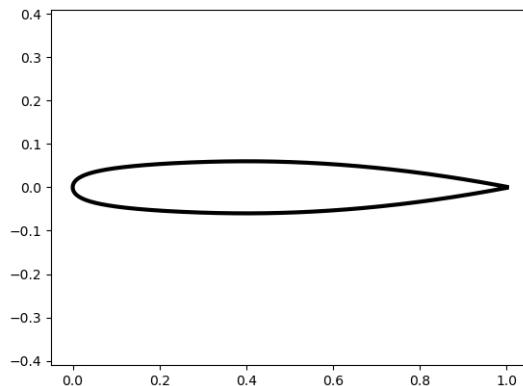


Figure 2.3: Visualisation of NACA 0012-74.

## NACA 16-series

NACA 16-series is a special case of NACA 4-digits modified. It has been mentioned and implemented in the Python code to facilitate the procedure for the user and to avoid the "translation" from one nomenclature to the other. For example, NACA 16-012 and NACA 0012-45 are the same airfoil geometry.

This series has a leading edge radius defined by the index  $I = 4$ , with maximum thickness located at  $x/c = 0.5$  (which is equal to say  $T = 5$ ) [26].

### 2.1.2 IGP parameterisation

IGP (Improved Geometrical Parameterisation) is a modern parameterisation proposed by Xiaoqiang et al. [21] in 2018. This method is a so-called *constructive method* since it does not need a basic airfoil for geometrical definition.

IGP method is conceptually based on *thin airfoil theory*, that is a simple theory related to inviscid incompressible flows which impinges airfoils characterized by small angle of attack, camber and thickness; *thin airfoil theory* was firstly devised by mathematician the Max Munk, and in a second instance by aerodynamicists Hermann Glauert et al. 1920s. The most important result of *thin airfoil theory* is the substantial separation of angle of attack, thickness and camber respective effects if the they are relatively small. In particular, typical small-thickness airfoils provides lift performances mostly due to their camber, rather than thickness [1].

IGP parameterisation is based on these notes, considering thickness and camber effects decoupled, as two independent problems. In particular, the camber line is expressed using Bézier curve while thickness is described by the polynomial basis function based on curve NACA 4-digits airfoil series.

The most important consequences of thickness and camber lines' decoupling about an ideal shape optimisation are:

- reduction of design space dimensions;
- faster optimisation procedure;
- possibility to consider thickness and camber optimisation problems as separated;
- easier shape definition, thanks to lower number of control parameter.

The latter cited advantage is due to the reduction of control geometric parameters with respect to other older parameterisation, as Table 2.1 reports. Note that the method should need (9) parameters, but considering an airfoil with finite angle at trailing edge the parameters decreases to (8).

Methods	Number of parameters
PARSEC	11
CST	11
OBF	10
IGP	8

Table 2.1: Comparison with others typical constructive method's number of parameters.

Reduction from (10) or (11) design parameters of older methods to (8) can be a deciding factor for a shape optimisation problems.

To demonstrate flexibility of IGP, the authors tried to reproduce 2199 already existing airfoils provided by UIUC Airfoil Data [29], and it was able to reconstruct 2166 airfoils (98.5% of the total library). The reconstruction precision along with lower number of design variables with respect to other constructive methods strongly makes IGP practical for preliminary optimisations procedures.

In the following, only design variables and respective ranges will be presented, since

they can be useful if the user wants to adopt this parameterisation.  
Basic geometrical parameters used by IGP method are visually reported in Figure 2.4.

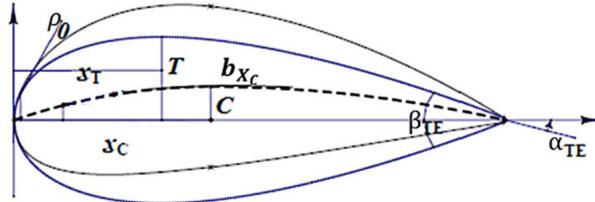


Figure 2.4: Illustration of geometric parameters [21].

To find these values, eight equations are managed and rewritten to provide a solvable linear system. Moreover, since there is an intrinsic decoupling between thickness and camber lines' problems, two groups of four equations can be considered as separated. The first four parameters regard camber line relations, while the second four define thickness distribution.

$$\begin{cases} c_1, c_2 \text{ define } x\text{-coordinate definition of camber line } (x_c). \\ c_3, c_4 \text{ define } y\text{-coordinate definition of camber line } (y_c). \end{cases} \quad (2.4)$$

$$\begin{cases} X_T \text{ is the chord location of maximum thickness.} \\ T \text{ defines the maximum thickness, such that } t(X_T) = T. \\ \bar{\rho}_0 \text{ is dimensionless leading edge radius.} \\ \bar{\beta}_{TE} \text{ is dimensionless trailing edge boat-tail angle.} \end{cases} \quad (2.5)$$

where the two dimensionless values are directly computed from the following relations:

$$\bar{\rho}_0 = \bar{\rho}_0 = \frac{\rho_0}{(\frac{T}{X_T})^2} \quad (2.6)$$

$$\bar{\beta}_{TE} = \frac{\beta_{TE}}{\arctan \frac{T}{1-X_T}} \quad (2.7)$$

The geometrical parameters' bounds were determined by various tests performed by the authors:

$$c_1 \in [0.010, 0.960] \quad (2.8)$$

$$c_2 \in [0.020, 0.970] \quad (2.9)$$

$$c_3 \in [-0.074, 0.247] \quad (2.10)$$

$$c_4 \in [-0.102, 0.206] \quad (2.11)$$

$$X_T \in [0.2002, 0.4813] \quad (2.12)$$

$$T \in [0.0246, 0.3227] \quad (2.13)$$

$$\bar{\rho}_0 \in [0.1750, 1.4944] \quad (2.14)$$

$$\bar{\beta}_{TE} \in [0.1452, 4.8724] \quad (2.15)$$

Note that the cited bounds are not so restrictive: they could be exceeded, but the probability to encounter "strange points" increases, as reported by the authors.

### 2.1.3 Benzing Airfoils

The airfoils proposed by E. Benzing [2] have been developed for racing cars. Differently from aeronautical airfoils, airfoils applied to ground vehicles have to generate downforce; moreover, the typical Reynolds numbers are quite different. Another difference with respect to aeronautical sector is the less favourable geometrical limits imposed naturally by a ground vehicle, but also by regulations. The continuous seek of higher downforce throughout the years showed that the single-element configurations were not sufficient for high performances: as knowledge about multiple configurations grew, the presence of flaps mounted increase. Without any regulation, racing cars categories tried retractable airfoils (like for aircraft) to reduce drag along the straights of the circuits. For safety reasons, FIA blocked these developments imposing fixed multi-element configurations. Secondly, another regulation was about the total number of elements allowed, to avoid extremely complex configurations. E. Benzing developed a series of airfoils with the aim of guaranteeing high efficiencies based on aerodynamic notes related to racing cars' typical conditions. As a consequence, these airfoils are not suggested for the aeronautical sector, but are specifically designed for racing cars since a very specific pressure and velocity distribution were researched in their definition.

When these airfoils were developed, the methodology adopted for curvature definition was similar to typical NACA airfoils, but with more design characteristics and parameters. Figure 2.5 reports the classification features of these type of airfoils. Note that the terms reported are not aligned with the nomenclature used in the present thesis; moreover, they are not sufficient for the determination of the geometry, but simply reports the main features of the airfoil under analysis.

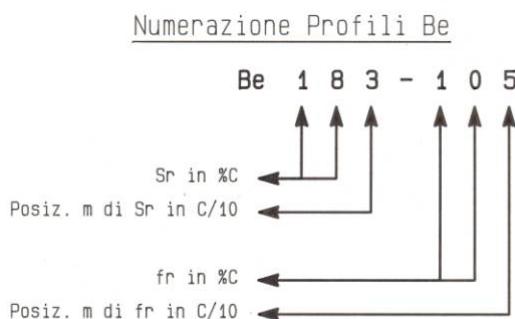


Figure 2.5: Features related to Be-series airfoils developed by E. Benzing [2].

In the following a clarification about the nomenclature used by Benzing is presented in the following list. Note that, typically, Benzing airfoils avoid pressure peak value inside airfoil denomination since mostly very high.

- **Sr:** relative maximum thickness;
- **$m_{Sr}$ :** position of relative maximum thickness along the chord;
- **fr:** relative maximum camber;
- **$m_{fr}$ :** position of relative maximum camber along the chord.

E. Benzing provided formula for (15) airfoils, reported in the following table.

<b>Be 092-046</b>	<b>Be 152-075</b>	<b>Be 123-077</b>
<b>Be 122-046</b>	<b>Be 094-056</b>	<b>Be 153-077</b>
<b>Be 152-046</b>	<b>Be 124-056</b>	<b>Be 103-115</b>
<b>Be 092-075</b>	<b>Be 154-056</b>	<b>Be 123-115</b>
<b>Be 122-075</b>	<b>Be 103-077</b>	<b>Be 153-115</b>

Table 2.2: Benzing airfoils implemented in the present thesis.

Anyway, these formula are somehow not-precise in geometry definition, especially on leading and trailing edge. Some corrections were implemented inside the Python code of the present thesis:

- the curvature on leading edge has been reconstructed by using radius and centre relative to the specific airfoil (provided in the cited reference);
- trailing edge has been cut since the upper and lower curves usually intersect each other close to trailing edge. The cut location strictly depends on how far this intersection occurs; anyway, it is never lower than 95% of the chord.

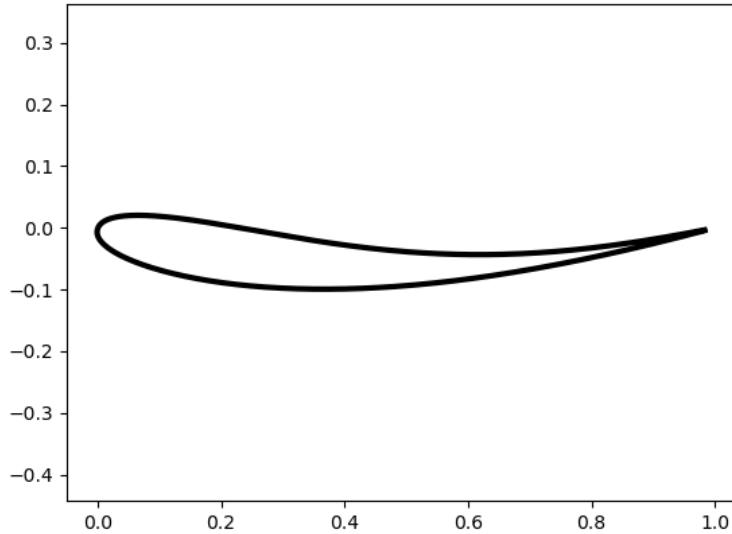


Figure 2.6: Visualisation of Be 092-075, printed using present thesis' tool.

## 2.2 Navier-Stokes Equations

The Navier-Stokes equations are the governing equations for a viscous heat conducting fluid. They are composed by the mass conservation, momentum equation and energy equation.

For a compressible fluid, the 2D Navier-Stokes equations can be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \quad (2.16)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + P) + \frac{\partial}{\partial y}(\rho uv) &= \frac{\partial}{\partial x} \left[ (2\mu + \lambda) \frac{\partial u}{\partial x} \lambda \frac{\partial v}{\partial y} \right] + \\ &+ \frac{\partial}{\partial y} \left[ \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \end{aligned} \quad (2.17)$$

$$\begin{aligned} \frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho v^2 + P) &= \frac{\partial}{\partial x} \left[ \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \\ &+ \frac{\partial}{\partial y} \left[ \lambda \frac{\partial u}{\partial x} + (2\mu + \lambda) \frac{\partial v}{\partial y} \right] \end{aligned} \quad (2.18)$$

$$\begin{aligned} \frac{\partial E^t}{\partial t} + \frac{\partial}{\partial x} [u(E^t + P)] + \frac{\partial}{\partial y} [v(E^t + P)] &= \frac{\partial}{\partial x} \left( K \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( K \frac{\partial T}{\partial y} \right) + \\ &+ \frac{\partial}{\partial x} \left[ (2\mu + \lambda) u \frac{\partial u}{\partial x} + \lambda u \frac{\partial v}{\partial y} \right] + \\ &+ \frac{\partial}{\partial x} \left[ \mu v \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \\ &+ \frac{\partial}{\partial y} \left[ \mu u \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \\ &+ \frac{\partial}{\partial y} \left[ (2\mu + \lambda) v + \frac{\partial v}{\partial y} + \lambda v \frac{\partial u}{\partial x} \right] \end{aligned} \quad (2.19)$$

where  $\underline{u} = (u, v)^T \in \mathbb{R}$  is the 2D flow velocity vector.

## 2.3 Computational Fluid Dynamics

### 2.3.1 Navier Stokes & Euler equations in SU2

SU2 adopts and solves the following compressible Navier-Stokes equations in differential form:

$$R(\mathbf{u}) = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \bar{\mathbf{f}}^c(\mathbf{u}) - \nabla \cdot \bar{\mathbf{f}}^v(\mathbf{u}, \nabla \mathbf{u}) - A = \mathbf{0} \quad (2.20)$$

where:

- $A$ : generic source term;
- $\mathbf{u}$ : vector of conservative variables, such that:  $\mathbf{u} = \{\rho, \rho\bar{v}, \rho E\}^T$ ;
- convective flux:  $\bar{\mathbf{f}}^c = \begin{Bmatrix} \rho\bar{v} \\ \rho\bar{v} \otimes \bar{v} + \bar{I}p \\ \rho E\bar{v} + p\bar{v} \end{Bmatrix}$
- viscous flux:  $\bar{\mathbf{f}}^v = \begin{Bmatrix} \cdot \\ \bar{\tau} \\ \bar{\tau} \cdot \bar{v} + \kappa \nabla T \end{Bmatrix}$

with:

- $\rho$ : fluid density;
- $\bar{v}$ : flow speed in Cartesian system of reference s.t.  $\bar{v} = \{u, v, w\}^T \in \mathbb{R}^3$ ;
- $p$ : static pressure;
- $E$ : total energy per unit mass; ,
- $\bar{\tau}$ : viscous stress tensor, which can be expressed as:  $\bar{\tau} = \mu(\nabla \bar{v} + \nabla \bar{v}^T) - \mu \frac{2}{3} \bar{I}(\nabla \cdot \bar{v})$ , with  $\mu$ : viscosity;
- $\kappa$ : thermal conductivity;
- $T$ : temperature.

Further notes about each term especially for ideal gases can be found directly in SU2 documentation [32].

When viscosity and thermal conductivity are absent can be neglected, SU2 provides the so-called compressible Euler equations.

These equations are a simplified version of compressible Navier-Stokes' ones represented by equation 2.20, described previously.

Consequently, compressible Euler equations can be expressed in differential form as:

$$R(\mathbf{u}) = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \bar{\mathbf{f}}^c(\mathbf{u}) - A = \mathbf{0} \quad (2.21)$$

### 2.3.2 Turbulence Model: Spalart–Allmaras (SA)

The Spalart-Allmaras model (SA) [33] is a one-equation eddy viscosity model, which is added to the RANS equations.

By a wider point of view, it belongs to the methods based on the concept of eddy (or turbulent) viscosity ( $\nu_t(\mathbf{x})$ ) and Boussinesq hypothesis [34] for the closure of RANS equations.

Although the definition of turbulent viscosity finds various applications and performs well on predicting several turbulent flows, it is limited and consequently inadequate in many occasions: flows with abrupt change of shear rate [35], over curved surfaces, in ducts with separations or secondary motions are only examples where eddy-viscosity models do not perform well. These cases reminds that turbulent viscosity is a non-physical term, added with the only purpose to close the RANS problem with respect to turbulent flows.

For the determination of  $\nu_t(\mathbf{x})$ , various models consider the definition of *mixing length* (proposed by Prandtl [36] for different purposes and reformulated for this kind of problem in computational fluid-dynamics). SA model does not belong to this wider group, since it aspires to directly create a partial differential equation for the computation of eddy viscosity.

In SA model  $\nu_t(\mathbf{x})$  is defined as:  $\nu_t = \tilde{\nu} f_{v1}$  and the model is given by the following equation:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[ c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2} \right] \left( \frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \left[ \frac{\partial}{\partial x_j} \left( (\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\partial \tilde{\nu}}{\partial x_i} \right] \quad (2.22)$$

with the boundary conditions:

$$\tilde{\nu}_{wall} = 0 \rightarrow \nu_{t,wall} = 0 \quad (2.23)$$

$$\tilde{\nu}_{farf} = 3 \div 5\nu_\infty \rightarrow \nu_{t,farf} = 0.210438 \div 1.294234\nu_\infty \quad (2.24)$$

The values and respective determination of constants and coefficients inside the equation 2.22 are provided by Spalart and Allmaras [33].

This turbulence model received various developments and modifications throughout the years to face particular problems or to alleviate some disadvantages.

SU2 provides the following versions (besides classical **SA** already presented):

- **SA\_NEG**: SA Negative [37].
- **SA\_E**: SA with Edwards Correction [38].
- **SA\_COMP**: SA with compressibility correction [39].
- **SA\_E\_COMP**: SA with both Edwards and compressibility corrections.

### 2.3.3 Bas-Cakmakcioglu Transition Model (BC)

For the description and evaluation of boundary-layer transition several models have been described throughout the years. At the current stage, SU2 provides the Bas-Cakmakcioglu Transition Model (BC) [40] [41] which can be applied along with Spalart-Allmaras turbulence model described in previous section. The complete transition turbulence model is called SA-BC [42].

BC model is a correlation-based algebraic transition model which relies on local flow information from the computational point of view, this model uses the already existing terms provided by SA turbulence model instead of deriving an intermittency equation. In particular, the proposed intermittency function  $\gamma_{BC}$  multiplies the existing production term of SA model, which is damped with respect to onset requirements.

The BC model depends on local flow information from which empirical correlations are derived for transition detection. In particular, the main onset correlation is the following:

$$Re_{\theta_c} = 803.73(Tu_{\infty} + 0.6067)^{-1.027} \quad (2.25)$$

where  $Re_{\theta_c}$  is the experimental transition onset critical momentum thickness Reynolds number, while  $Tu_{\infty}$  is the freestream turbulence intensity, defined as:

$$Tu_{\infty} = \frac{u'_{\infty}}{U_{\infty}} \quad (2.26)$$

where  $u'_{\infty}$  is the root-mean-square of freestream turbulent velocity fluctuations, while  $U_{\infty}$  is the mean freestream velocity. For the purposes of the present thesis (aeronautical or automotive multi-element airfoils), freestream turbulence intensity is usually much lower than 1%. As also BC authors report, Drela  $e^N$  type model [43] is curve fitted (Figure 2.7): consequently, results are aligned with already existing and accurate models.

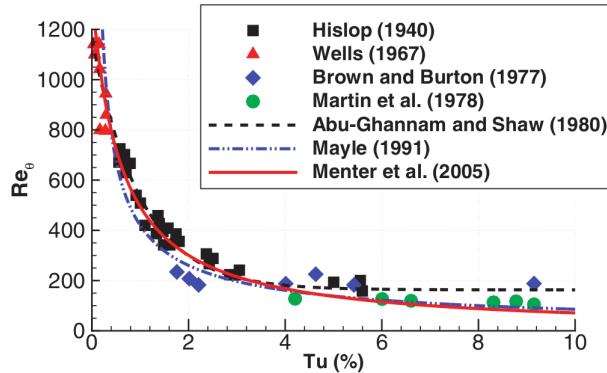


Figure 2.7: Comparison of transition onset correlations with experimental database.

Less equations and less constants' calibrations provide a much easier model with respect to already existing ones, like Menter et al.  $\gamma - Re_{\theta}$  model [44] or Kubacki et al. [45]. For example, comparisons with  $\gamma - Re_{\theta}$  model about a 3D problem see a reduction from (9) to (6) equations to be solved, guaranteeing important resources' saving. About constants' calibration, it reduces to (2).

Limitations of BC Transition Model have been found when the adverse pressure gradients become significantly strong; anyway, in this approach it provides accurate results in majority of cases.

### 2.3.4 Numerical Schemes

Inside the present work, the adopted numerical schemes are:

- Second order Roe scheme with MUSCL approach (*Monotonic Upstream-centred Scheme for Conservation Laws*);
- Jameson-Schmidt-Turkel scheme (JST).

Both the methods are 2nd order accurate and provide various advantages with respect to 1st order schemes.

**MUSCL scheme** was indeed developed to face various issues and limitations caused by simpler 1st order schemes, like Godunov's scheme or first methods based on Godunov. Proposed by Bram van Leer [46] [47] in 1979, the MUSCL scheme (*Monotonic Upstream-centered Scheme for Conservation Laws*), is a method that can provide highly accurate numerical solutions for a given system even in case of large gradients, shocks and discontinuities, where 1st order schemes fail causing smearing introducing so-called *numerical viscosity*.

The main idea of MUSCL scheme is to calculate fluxes at the cell edges by reconstructing left and right states for each cell, obtained from cell-averaged states of previous time-step. As most of the available 2nd order schemes, MUSCL can present oscillations close to large gradients, like discontinuities.

On the other hand, **JST** (Jameson-Schmidt-Turkel scheme) [48] is another 2nd order scheme, proposed in 1981 to reduce these oscillations close to large gradients.

This advantage can be achieved by managing (2) terms (named *artificial dissipation coefficients*) inside the scheme. The simpler version for one-dimensional Euler equation is reported as clarification:

$$\frac{dU_j}{dt} + \frac{F_{j+1} - F_{j-1}}{2h} = \frac{d_{j+\frac{1}{2}}^{(2)} - d_{j-\frac{1}{2}}^{(2)}}{h} + \frac{d_{j+\frac{1}{2}}^{(4)} - d_{j-\frac{1}{2}}^{(4)}}{h} \quad (2.27)$$

In particular, the second-order artificial dissipation term is activated only close to shocks, thanks to the design of the flux, while the fourth-order dissipation term only acts in smooth regions to suppress high-frequency oscillations. Since they add dissipation, second and fourth order terms lead to loss of energy/amplitude.

When using SU2 solver, these two parameters can be managed inside the configuration file at the following item:

$$\text{JST\_SENSOR\_COEFF} = (0.5, 0.02)$$

These notes about behaviour close to discontinuities are less important in the present thesis, since the intent is not to investigate airfoils impinged by transonic flows.



## Chapter 3

# Mesh generation and CFD simulation procedure

In this chapter the analysis focus on how the tool builds a CFD grid and prepares a CFD simulation, starting from user input data. If requested, the CFD simulation can be run right after the grid generation. The procedure of mesh generation and CFD simulation is summarised by a block diagram in Figure 3.1.

Firstly, the tool generates grids utilizing Gmsh [49] using its syntax. The generation procedure consists of writing a text file with the suffix *.geo*, required by Gmsh. Grid generation is completed, and Gmsh is automatically opened with the following command:

```
$ gmsh filename.geo -rand 1.e-14
```

Secondly, the tool reads the user CFD input about the configuration file, and it subsequently runs SU2 RANS simulation by executing the following command:

```
$ SU2_CFD ConfigurationFile.cfg
```

The description is executed using block diagrams, for which a legend is provided in the following. Firstly, the blocks' types:

- yellow rectangle gathers the inputs provided by the user about geometry and grid;
- grey rectangles are proceedings code blocks, with various roles defined by what written inside;
- green blocks refer to writing single grid file and executing CFD simulation related;
- cyan or blue blocks refer to optimisation methods, steps or results;
- red blocks report input errors due to inconsistencies, causing the tool's running stops.

Secondly, rectangles defined by dashed lines report which Python script executes which action, step or operation:

- red dashed lines `--` refer to Python scripts modifiable by the user. Here the user has to provide geometry, grid and others custom options.
- blue dashed lines `- - -` refer to proceeding Python scripts, where the user is not called to modify codes' lines.

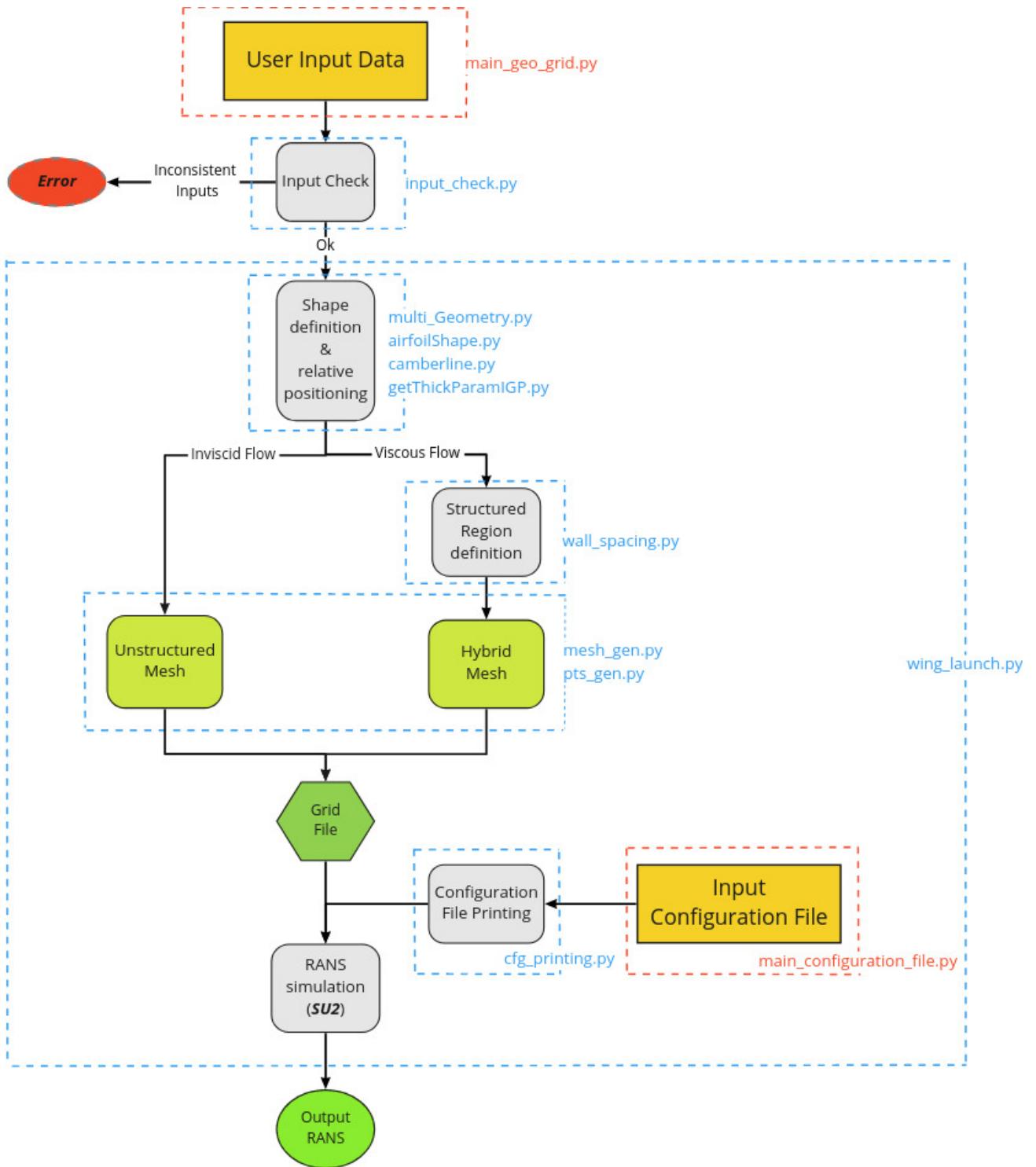


Figure 3.1: Grid generation and CFD simulation module.

### 3.1 Grid module Codes' features

The user can manage several features of the geometries and the grid. These inputs have to be consistent between them or providing reasonable values.

The Python file `input_check.py` executes several checks about input inconsistencies.

#### 3.1.1 Geometry

About the geometry, the user can provide:

- default parameterisation airfoils. Available default options have already been described in Section 2.1;
- external input files: in these cases, the user must provide text files named `MY_FILE_i`, where  $i$  indicates the airfoil number positioning and it has to be consistent with the other lists provided in input (AoA, relative positioning etc.) at index  $i$ .

Inside the various options, the user has to write the homonym string '`MY_FILE_i`'. An example is reported inside Appendix A, where the input file adopted for slot optimisation of GA(W)-1 airfoil is presented along with used points coordinates.

The code firstly considers airfoils with unit chord (with  $x \in [0, 1]$ ) and with points' coordinate provided from TE to LE passing through the lower surface, then proceedings on the upper; chords' scaling and rotations are executed only secondly.

The user has to respect these input rules, otherwise errors may occur. Codes provided some basic check about, trying to fix whether geometry definition is wrong.

Secondly, the user can customize the trailing edge of each element by choosing between a finite angle TE, or truncated TE defined by a line which links the upper and lower surface. The unique exception is about trailing edges of IGP airfoils, which is mandatory defined by finite angles [21].

#### Geometry scaling and rotation

Depending on the values of respective chords and absolute angles of attack provided in input by the user, the Python code (in particular, script `multiGeometry.py`) performs scaling and relative rotation of the elements.

The first one is executed by firstly determining a preliminary airfoil's element shape of unit chord from the selected parameterisation or an external file; secondly, scaling is performed by multiplying x and y-coordinates by relative chord in input.

Rotation is executed by considering a proper rotation matrix, which requires a rotation angle and a reference point.

The rotation angle is defined by the relative angle of attack ( $\Delta\alpha_i$ ), evaluated as follows:

$$\Delta\alpha_i = \alpha_i - \alpha_0 \quad (3.1)$$

where  $\alpha_i$  is the absolute angle of attack of the  $i$ -th element, while  $\alpha_0$  is the absolute angle of attack of the main element with unit chord, respectively.

The centre of rotation is defined as located on the trailing edge of the  $i$ -th profile.

In linear algebra, a rotation matrix is a transformation matrix used to perform a rotation

in Euclidean space, defined as follows:

$$R = \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

where  $(\Delta\alpha_i)$  is the relative angle of attack of the  $i$ -th airfoil. The last row and column are irrelevant since a 2D problem is taken into account, with z-coordinate unchanged. The rotation is given by the following matrix multiplication:

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) \\ \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} \quad (3.3)$$

where  $\underline{x} = \begin{Bmatrix} x \\ y \end{Bmatrix}$  and  $\underline{x}' = \begin{Bmatrix} x' \\ y' \end{Bmatrix}$  are the original and rotated coordinates' vectors, respectively.

### 3.1.2 Grid

About the grid, the user can provide several custom options. From a general point of view, the user can select the following type of flow (through a homonym option *flow*):

- EULER option for inviscid RANS simulations;
- VISCOUS.

Two different functions inside Python script **mesh\_gen.py** are present to generate a grid that is suitable for the selected flow.

As a consequence, two grid options are:

- unstructured grid for inviscid flows (option EULER);
- hybrid grid for viscous flows (option VISCOUS);

The descriptions provided in the following are focused on the hybrid grid since related to the viscous flow which occupies the main interests and details.

Completely unstructured grids for inviscid flows are however based on these concepts and, consequently, very similar. In particular, inviscid RANS simulations do not need structured regions close to wall surfaces, like no particular need of refining in the wake's region.

In line with the customization of Gmsh, the tool allows for using different algorithms for unstructured regions. In the present thesis, the chosen unstructured region algorithm is Frontal-Delaunay (provided as default by Gmsh); different algorithms are addressed for future studies. See Reference [50] for Frontal-Delaunay theory details, while [49] for Gmsh implementations and options.

As an example of a hybrid grid for viscous flows involving multi-element configurations, C-11 airfoil's grid for wind turbines [51] is presented.

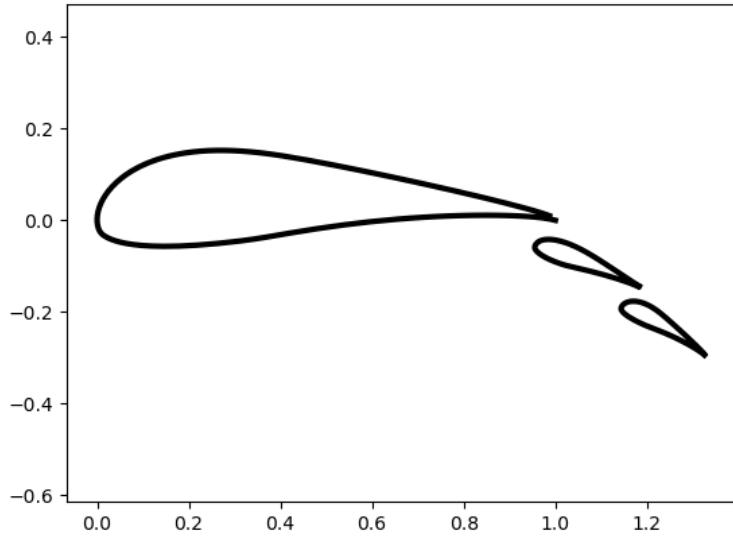


Figure 3.2: C-11 airfoil sketch.

Figure 3.3 shows a grid example about C-11 configuration (complete and with two levels of zooms). All the available implemented grid features are observable, and will be deeply described in the following.

The user can manage directly refining by customizing the parameters listed below:

- *nodes* defines the general refinement on the grid; increasing this value the smallest element size decreases, guaranteeing a finer grid on the entire domain.
- *ellipse\_refining* defines the nodes' refinement on the ellipse's grid splines.
- *wall\_refining* performs a proportional refinement on airfoils' wall surfaces.
- *wake\_length* and *wake\_progr*, which defines the refining lengths and progression of elements' size for the wake region.

These input parameters directly affect refining, but several other ones can be customized by the user to manage structured region, ellipse and external grid semi-circle, described in the following paragraphs.

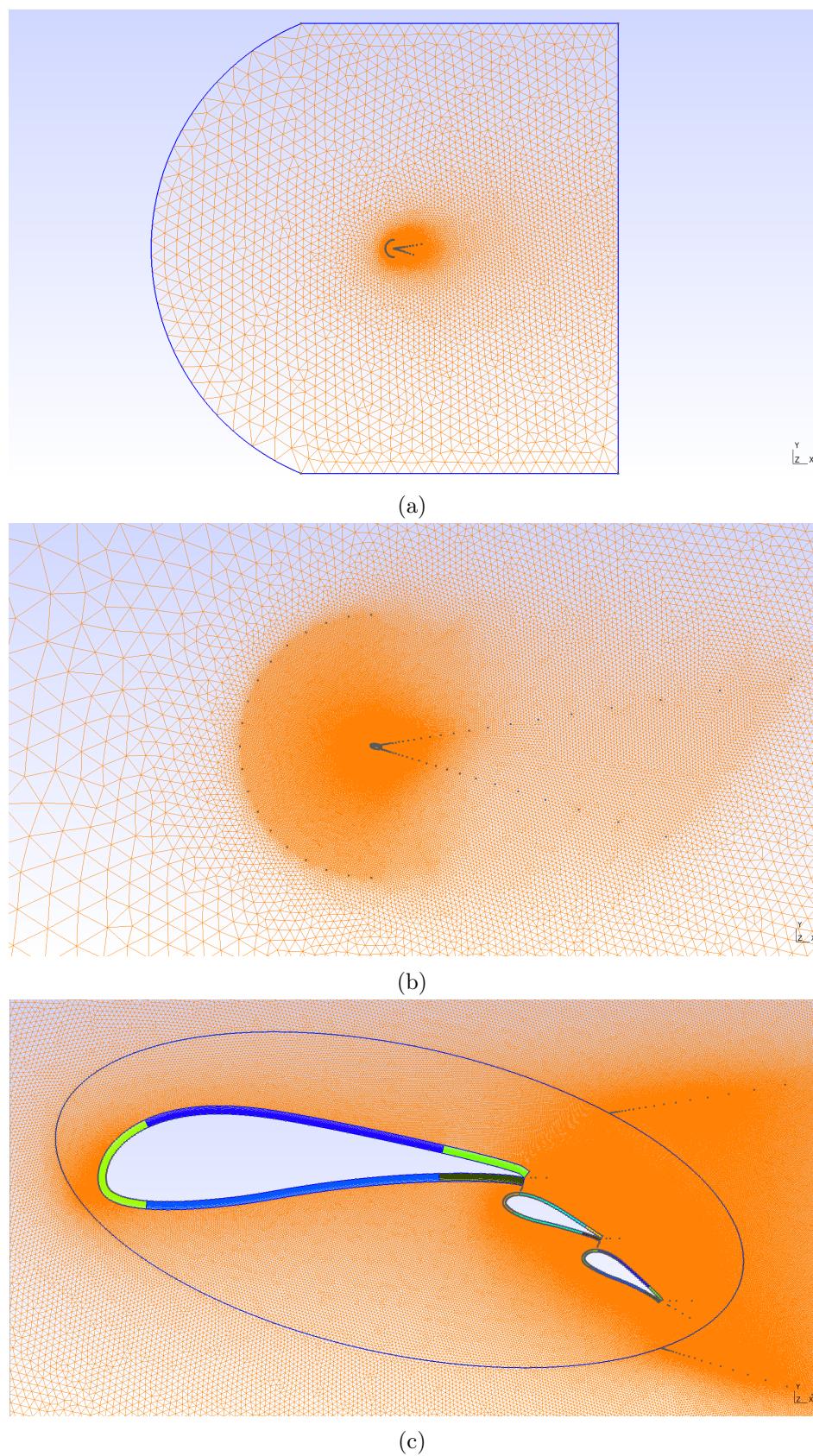


Figure 3.3: C-11 airfoil grid (complete and zoomed) on Gmsh.

## Structured region

A viscous RANS simulation requires a proper structured region close to wall surfaces. This feature is needed to correctly detect boundary layer's viscous scales close to the airfoil surface [52].

From a geometrical point of view the structured region is created by considering the following scheme:

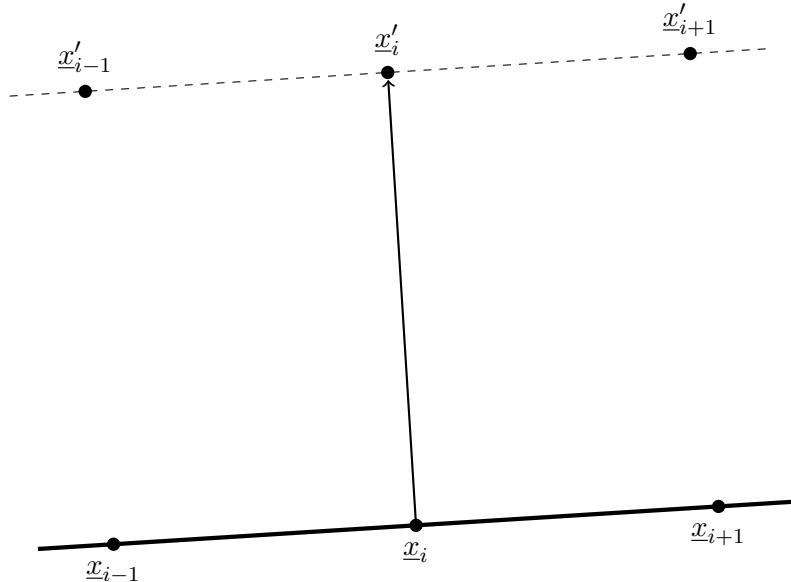


Figure 3.4: Schematic visualization of the adopted method for structured region generation.

Gmsh provides some commands to generate a structured region close to wall surfaces automatically, but after some previous tests it has been chosen to discard these options since they affected the robustness of grid generation by causing some errors.

As a consequence, the tool adopts the procedure presented in Figure 3.4 to generate the structured regions' points (and consequently respective surfaces and perimeter splines). For each  $i$ -th point on the wall surface ( $x_i$ ), it exists a respective one on the external spline of a structured region; the determination is executed by evaluating a line and its relative slope determined by previous and after wall surfaces' points ( $x_{i-1}$ ,  $x_{i+1}$ ). This procedure is sufficiently robust since it is simply based on geometrical considerations. Some issues can be encountered if the wall surfaces' points are too coarse: the consequences could be a structured region defined by few points on its external perimeter and, consequently, relative splines could be bad-defined.

Anyway, the consequences on structured regions due to poor wall surfaces' points definition came later, since the primary bad effect would be directly on wall surfaces: in these cases, the airfoils would be poorly defined and splines would not be as regular as desired.

Default geometries do not present these possible issues since the user can directly define the geometrical total amount of points for the elements by giving an integer value to the item  $n$  in input. Issues may be faced when the user provides external files in inputs with a low total amount of points for geometries' definitions.

Lastly, the code needs to understand where these points have to be placed depending on the upper or lower surfaces of the airfoil. Considering again the sketch example

of Figure 3.4, the displayed positioning of structured region points ( $x'_{i-1}$ ,  $x'_i$ ,  $x'_{i+1}$ ) is correct only if these points are related to the upper surface, otherwise they must be placed below the wall surface line (determined by points  $\underline{x}_{i-1}$ ,  $\underline{x}_i$ ,  $\underline{x}_{i+1}$ ). The tool code simply solves this topic by firstly calculating the structured regions' points before airfoils positioning, scaling and rotation based on their chords and angles of attack; secondly, wall surface points are ordered starting from the trailing edge through the lower surface, then proceeding to the leading edge and the upper surface lastly.

The next step regards how grids' details inside structured regions are computed, along with a proper definition of the thicknesses of the structured regions.

In particular, a proper CFD grid has to achieve the following condition:

$$y^+ < 1 \quad (3.4)$$

along airfoils' surfaces to properly detect turbulent viscous flows, where  $y^+$  is the dimensionless wall distance.

Its definition derives from the length and velocity scales formed with viscosity ( $\nu$ ) and wall stress ( $\tau_w$ ):

$$\text{Friction velocity: } u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (3.5)$$

$$\text{Viscous length: } \delta_v = \frac{\nu}{u_\tau} \quad (3.6)$$

$$\text{Dimensionless wall distance: } y^+ = \frac{yu_\tau}{\nu} \quad (3.7)$$

In order to avoid manual definition of *wall spacing* ( $\delta_s$ , normal thickness of grid's closest element to wall surface) by the user, a reference was taken into account to facilitate the generation of the structured region. In particular, its thickness is based on Schlichting experimental formula for turbulent boundary layers along a flat plate channel [53]:

$$\delta(x) \approx 0.37 \frac{x}{Re_x^{1/5}} \quad (3.8)$$

The thickness of the BL is dependent on the position along the airfoil chord ( $x$ ) and Reynolds number per unit chord ( $Re_x$ ). Inside the code, the dependence on x-coordinate was omitted, imposing  $x = c$  (respectively for each airfoil).

Considering the following relations:

$$\tau_w = 0.5 C_f \rho U^2 \quad (3.9)$$

$$C_f = 0.0576 Re_x^{-1/5} \quad (3.10)$$

wall spacing is eventually obtained:

$$\delta_s = \frac{y^+ \nu}{\rho u_\tau} \quad (3.11)$$

The user has to provide flow conditions values (Reynolds number, dynamic viscosity and temperature) along with a theoretical  $y^+$  value for each airfoil.

The structured region thickness can be scaled with another item (*thick*) inside the Python input file.

Using Schlichting formula is not precise since referred to flat plates and based on simple experimental observations. For example, the empirical relation about friction coefficient ( $C_f$ ) is approximately valid for a range of Reynolds per unit chord  $Re_x \in [5 \cdot 10^5, 10^7]$  [53].

Anyway, these relations provide an easy reference to generate the structured region with a dependence on Reynolds number. Further developments could base structured regions on more complex and specific relations.

### Ellipse and External semi-circle

As depicted in Figure 3.3, a dummy ellipse and semicircle have been imposed inside the unstructured region.

These objects are in fact only fictitious and finalized to provide a refinement in the unstructured regions close to airfoils, avoiding a sudden "jump" from coarse freestream elements to finer ones on airfoils' surfaces.

In particular, the ellipse has been constructed following the main geometry relations [54] [55]:

$$\text{Ellipse equation: } 1 = \left( \frac{x - h}{a} \right)^2 + \left( \frac{y - k}{b} \right)^2 \quad (3.12)$$

$$\text{Area: } A = \pi ab \quad (3.13)$$

$$\text{Eccentricity: } e = \sqrt{1 - \left( \frac{b}{a} \right)^2} \quad (3.14)$$

The tilt angle ( $\tau$ ) of the ellipse has the following effect on (x,y) ellipse points' coordinates:

$$\text{Ellipse with Tilt angle: } \begin{Bmatrix} x(t) \\ y(t) \end{Bmatrix} = \begin{Bmatrix} h \\ k \end{Bmatrix} + \begin{Bmatrix} \cos(\tau) & -\sin(\tau) \\ \sin(\tau) & \cos(\tau) \end{Bmatrix} \begin{Bmatrix} a \cos(t) \\ b \sin(t) \end{Bmatrix} \quad (3.15)$$

such that  $0 \leq t \leq 2\pi$ .

In the cited relations,  $(h, k)$  are the abscissa and ordinate coordinates of the ellipse's centre, while  $a, b$  are the length of major and minor semi-axes, respectively. If  $a > b$ , the ellipse is stretched further in the horizontal direction, otherwise vertically.

The ellipse centre has (x,y) coordinates based on airfoil coordinates weighted by respective relative chords. Tilt angle is aligned with the line which links the more left airfoil leading edge to airfoil's trailing edge on the right.

Since the ellipse has to enclose the airfoils' geometries (and the structured regions in viscous flows problems), the relation becomes:

$$1 > \left( \frac{x - h}{a} \right)^2 + \left( \frac{y - k}{b} \right)^2 \quad (3.16)$$

To ensure a margin between inner points and the ellipse perimeter, the user can assign a value lower than 1 at the left end of the inequality.

From the grid point of view, nodes on ellipse do not belong to a unique spline on the ellipse's perimeter, but to an ensemble; this subdivision has been imposed for refining where necessary. In particular, the ellipse ensures a refinement close to more left airfoil's leading edge and all elements' trailing edges, depending on the respective angles of attack.

If needed, the user can proportionally increase the ellipse refinement anyway through item *ellipse\_refining*.

On the other hand, a semi-circle is simply obtained by centring it using airfoils' coordinates. Its dimensions are obtained scaling with respect to ellipse and wake sizes. Again, the user can manage refinement related to the semi-circle's grid points.

### Wake refining

Airfoils require refining downward, where the wake has a fundamental role due to the wake's effects on aerodynamic performances, especially pressure drag [3].

To entirely detect the wake, the grid considers the absolute angle of attack and the relative ones of each element.

As a consequence, the wake detection is performed by two straight lines downward the airfoil configuration, which respectively should contain the overall wake region, guaranteeing a refinement.

These lines are composed of 50 grid points, positioned in the domain using a geometric progression.

Elements' grid size of the first grid wake line starts from a smaller value (defined in input by the user with input item *nodes*) and then increase with a factor of 1.20 from the 20th point.

The second line follows the same idea, with a elements' grid size of 1.21 to possibly avoid overlapping of the two lines' points.

Like grid ellipse and semi-circle, the user can customize refinement and wake length.

### 3.1.3 Interface with Gmsh

Gmsh requires a numerical ID assignment for each point, line/spline and surface inside grid generation file. As a consequence, the tool's Python3 scripts manage to avoid repetitions which may end up in errors when Gmsh tries to generate the proposed grid. Repetitions are only allowed if the entity of the element is different: for example, Gmsh accepts simultaneous existence of Point ID with 10, Line with ID 10 and Surface with ID 10.

Since the code should work automatically and must be adaptable to every input configuration, the robustness of the grid generation method is fundamental.

Table 3.1 summarises this topic: the more left column reports the name of each grid or geometry feature, while the cells report the range assigned for respective grid points, lines and surfaces.

Geometry or Grid feature	Points ID	Lines ID	Surfaces ID
<b>Farfield</b>	1, 2, 3, 4	1, 2, 3, 4	1000
<b>Airfoil (1)</b>	100, ..., $N_1 - 1$	100, ..., 104, (105)	/
<b>Struct. Region (1)</b>	10000, ..., 10000 + $N_1 - 1$	1000, ..., 10011, (206, 207, 208)	10, ..., 14, (15)
<b>Refining wake/slot points (1)</b>	10, ..., 19	/	/
<b>Airfoil (2)</b>	$N_1, \dots, N_2 - 1$	200, ..., 204, (205)	/
<b>Struct. Region (2)</b>	10000 + $N_1, \dots,$ 10000 + $N_1 + N_2 - 1$	2000, ..., 2011, (206, 207, 208)	20, ..., 24, (25)
<b>Refining wake/slot points (2)</b>	20, ..., 29	/	/
...	...	...	...
<b>Ellipse</b>	100000, ..., 100359	5, ..., $N_E$	2000
<b>Wake</b>	100360, ..., 100459	/	/
<b>Ext. semicircle</b>	100460, ..., 100478	/	/

Table 3.1: Grid management performed by the tool for Gmsh - Points ID.

where:

- Number inside round brackets ( $i$ ) indicates the element number inside chosen airfoil configuration;
- $N_i$ : number of geometrical points which define  $i$ -th airfoil.
- $N_E$ : number of required splines for ellipse definition. It is strictly dependent on the total amount of elements considered for the airfoil. In the example proposed at the beginning of present chapter (airfoil C-11, Figure 3.3 ), three elements involves ellipses ID from 5 to 8.
- oblique bar (/) indicates no needs of ID assignment for the related feature.
- Round brackets are reserved for IDs used for truncated trailing edges cases only.

Looking at the table data it is possible notice that repetitions are avoided by the presented proper organisation; issues may only appear whether user inserts huge amount of airfoils in input or each element is defined by an unnecessary high number of points for its geometrical definition. Anyway, if future needs require higher robustness by this point of view, modifications to the tool are quite simple (for example: increasing the ranges by increasing the ID bounds).

Each line is defined by extreme points if straight. Each surface is defined by an ensemble of closed loop lines. As a consequence, some of these surfaces took the useful lines to complete the closed loop from "other" entities: for example, each structured region needs the external splines but also the airfoil wall surface's ones.

### 3.1.4 Element Size correction

To verify the robustness of grid generation procedure, several tests were performed. Firstly, the command which opens and generates grid files using Gmsh is the following:

```
gmsh -2 filename.geo
```

which is also the one suggested on Gmsh website.

First tests executed to verify grid sensitivity showed an error inside Gmsh software about unstructured region. An error example is reported in Figure 3.5a.

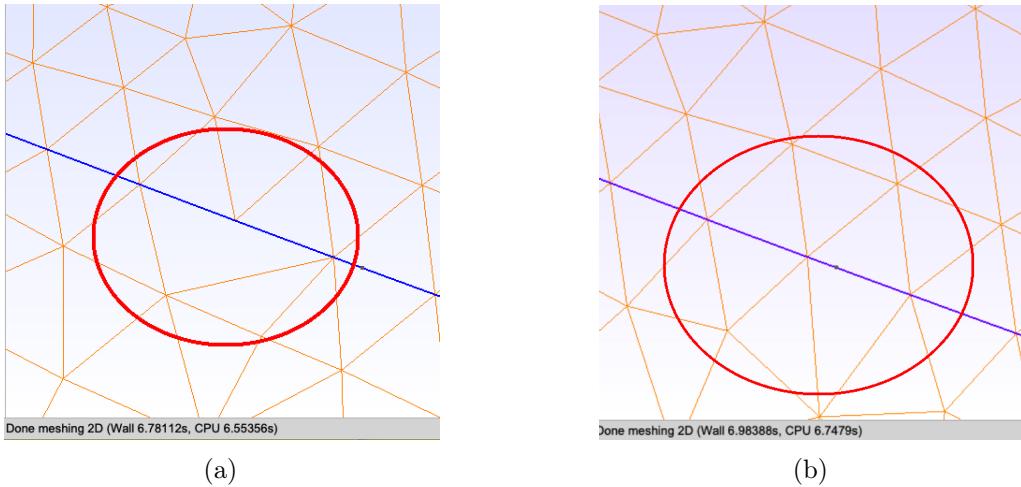


Figure 3.5: Example of error from typical Gmsh usage inside unstructured region (a) and relative correction (b).

This error occurred in various occasions, especially when high refining was requested; the domain location of this wrong unstructured element was always along interfaces regions (external ellipse or structured-unstructured interface); moreover, this error usually presented in multiple locations.

The effects of this "hole" inside the grid caused troubles when RANS simulations run. In particular, this domain element presented unphysical results, usually velocity peaks, resulting in difficulties to converge. The effect about hard-convergence became more evident if these holes were located close to the airfoil and near the leading edge.

This error was solved using the command:

```
$ gmsh filename.geo -rand 1.e-14
```

already presented at the beginning of present Chapter.

Visually, the correction was reached as Figure 3.5b exemplifies.

Eventually this command is the one implemented inside the code since it appears sufficient to avoid the cited error, at least for the present thesis' domain of multi-element airfoils.

## 3.2 NACA 0012 FT - Baseline grid CFD results

This section deals with grid sensitivity, one of the main CFD results' influencing factors. Due to the wide availability of default geometries along with the possibility to provide external data files, grid sensitivity studies are performed on a simple single-element geometry (NACA 0012). The CFD results are compared with experimental ones performed by Charles L. Ladson [56]; these experimental data are used also by NASA in Turbulence Modeling Resources [57] for comparison with CFD simulations of a NACA 0012. As reported in the references, Ladson's experimental data are performed with both free and artificial fixed transition (*tripped*). In the latter case, various grid roughness were applied at about 5% of the chord:

- 60-W grit;
- 80 grit;
- 120 grit;
- 180 grit;

In this chapter about grid sensitivity analysis the comparison has been performed using CFD simulations by SU2 with fully turbulent boundary-layer (FT) for a better comparison with experimental data. The chapter is organised as follows:

1. CFD results about a baseline grid will be presented; this grid is created using tool's parameters without any previous analysis. CFD results will be commented making comparisons with experimental data.
2. grid modifications will be applied, trying to delineate grid sensitivity with respect to these feature changes. The intent of these modifications is to identify the parameters which guarantee convergence of numerical simulations to physical results. Secondly, tool's users can look at these analysis exporting some informations about the possible effects of parameters' changes on aerodynamic coefficients.
3. after some modifications, a new polar about NACA 0012 is computed and presented; comparisons with experimental data are performed again.

The following flow and numerical conditions are used for flow numerical simulations about NACA 0012 with fully turbulent boundary-layer:

- Mach number: 0.15;
- freestream temperature: 288.15 K;
- Reynolds number:  $6 \cdot 10^6$ ;
- AoA: from 0.0 to 16.0 degrees;
- Turbulence Model: SA;
- Convective numerical method: JST;
- CFL: 1500.
- Time discretization: Euler Implicit;

- Numerical method for spatial gradients: Weighted Least Squares.

Flow conditions have been defined according with Ladson experimental data [56]. A series of viscous CFD simulations have been executed using the tool and hybrid grids. Since no previous analysis was performed, the first parameters setup was blind and simply based on experience and literatures CFD grids' examples. The main grid features analysed are reported in the following list, along with the respective starting values for the baseline setup.

- *farfield\_size* defines the farfield distance from the airfoil as chords times.  
Baseline value: 100;
- parameter *nodes*, which defines the general refinement of the grid.  
Baseline value: 1.
- parameters *progr* and *thick* define structured region thickness and elements' progression in normal direction with respect to wall surface of the airfoil.  
Baseline values, respectively: 1.1 and 1.
- parameter *ellipse\_dimension* = 0.85 for baseline, such that the enclosure ellipse formula presented in Section 3.1.2 becomes:  $\left(\frac{x-h}{a}\right)^2 + \left(\frac{y-k}{b}\right)^2 < 0.85$ . It corresponds to an area equals about to ( $0.17 \text{ m}^2$  for an airfoil with 1 m chord) for baseline.
- wake length of 50 and wake elements' size progression of 1.20 for baseline. The respective parameters for these purposes are: *wake\_length*, *wake\_progr*.

Some of these parameters have been normalized to simplify the analysis. For example, *thick* = 1 related to structured region thickness corresponds to the boundary-layer thickness value proposed by Schlichting formula (already explained in Section 3.1.2) which depends on flow conditions in input.

On the other hand, wake length and elements' size are based on the ellipse area and the smallest element size of the entire grid, as again explained in Section 3.1.2.

Table 3.2 and plots report the main aerodynamic and computational results performed about NACA 0012 with fully turbulent boundary-layer.

AoA [deg]	Lift Coefficient ( $C_L$ ) [-]	Drag Coefficient ( $C_D$ ) [-]	Grid N° of elements	Time Required [min]
0	-0.000021	0.008521	304080	33
2	0.226879	0.008664	299338	31
4	0.452083	0.009110	357348	38
6	0.673736	0.009909	372944	42
8	0.889369	0.011135	383898	66
10	1.094604	0.012924	388696	45
12	1.285253	0.015563	382730	84
14	1.452752	0.019606	421808	116
16	1.576785	0.026758	435756	196

Table 3.2: CFD simulation of NACA 0012 - Baseline grid.

It is important to report the convergence criterion applied for each simulation:

- convergence criterion: Cauchy, both on lift and drag.  
Epsilon to control the Cauchy series:  $10^{-4}$ .  
Number of elements to apply the criterion: 100.
- convergence criterion: density residual.  
Minimum value of the residual ( $\log_{10}$  of the residual): -8.5.

The criterion started to evaluate convergence from iteration 500 to keep out all initial RANS iterations from convergence criterion evaluations.

Since the purpose of the code is providing a tool for optimisation studies, computational time takes a relevant role.

The last columns give an overview of the general behaviour of the simulations performed: the computational time required increases when higher angles of attack are taken into account. Flow complexity along with getting closer to stall condition cause difficulties about convergence.

Anyway, a more complete analysis can be performed looking at errors. Along with the convergence criterion, two types of errors were considered in post-processing:

1. iteration from which the relative error w.r.t. the previous iteration stays under 0.1%.

$$\begin{aligned} \left| 100 \frac{C_{L_{i-1}} - C_{L_i}}{C_{L_{i-1}}} \right| &< 0.1 \\ \left| 100 \frac{C_{D_{i-1}} - C_{D_i}}{C_{D_{i-1}}} \right| &< 0.1. \end{aligned}$$

2. iteration from which the relative error w.r.t. (iteration - 100) stays under 0.1%.

$$\begin{aligned} \left| 100 \frac{C_{L_{i-100}} - C_{L_i}}{C_{L_{i-100}}} \right| &< 0.1 \\ \left| 100 \frac{C_{D_{i-100}} - C_{D_i}}{C_{D_{i-100}}} \right| &< 0.1. \end{aligned}$$

Plots in Figures 3.6, 3.7 and 3.8 propose the history of the aerodynamic coefficient with a coloured curve (red or blue) with reference values on the left y-axis at different angles of attack.

The respective Cauchy residuals are proposed with a black line with reference values on the right y-axis.

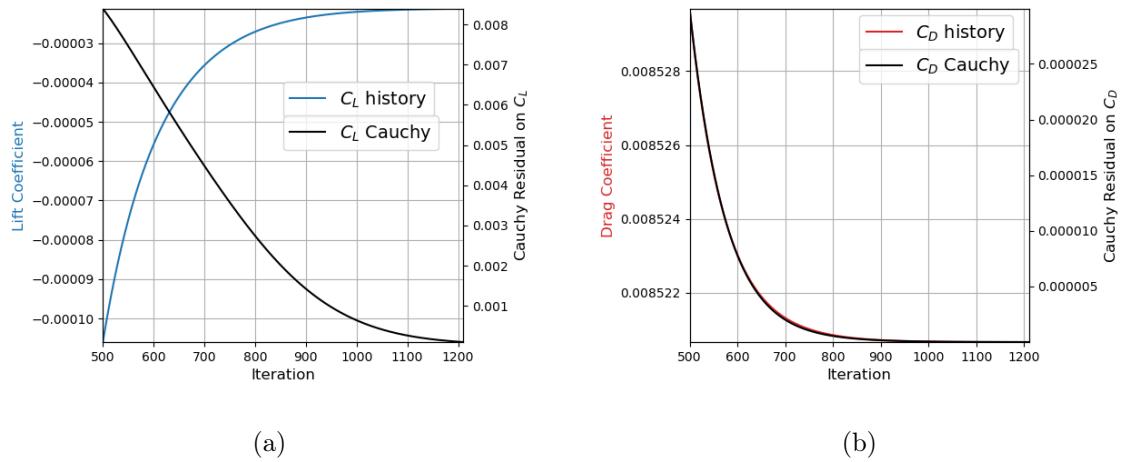


Figure 3.6: RANS History for NACA 0012 - Baseline, 0 deg.

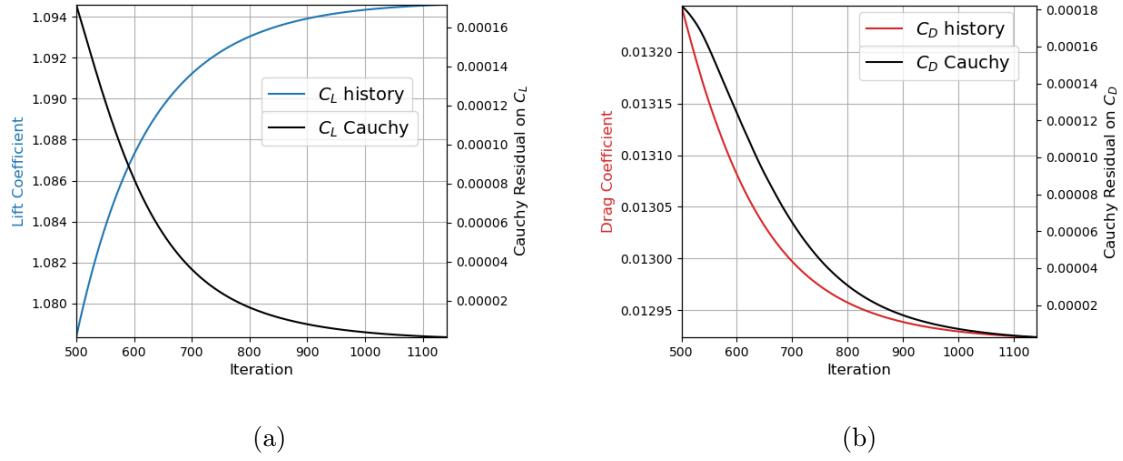


Figure 3.7: RANS History for NACA 0012 - Baseline, 10 deg.

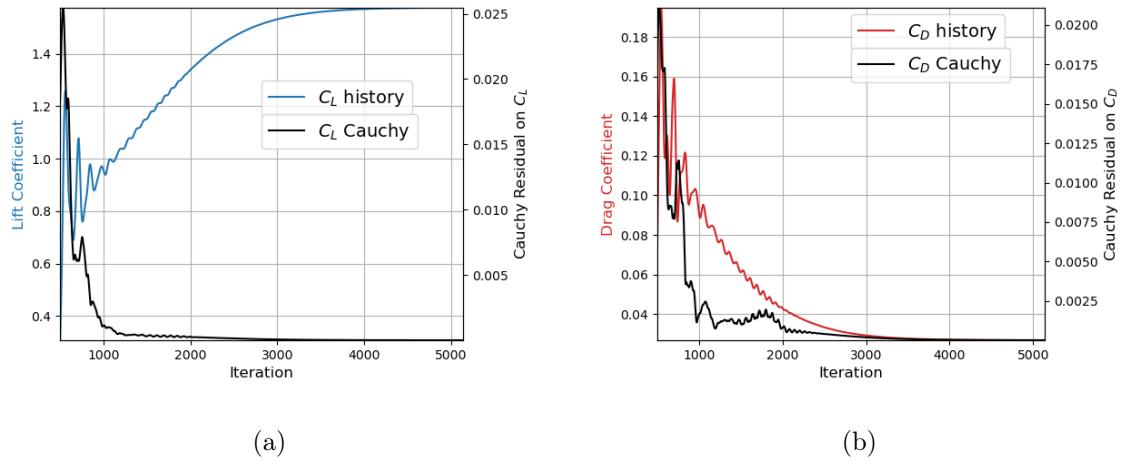


Figure 3.8: RANS History for NACA 0012 - Baseline, 16 deg.

By analysing history plots and table results proposed for the second criterion, it is possible to assert that convergence criterion can be relaxed, saving an important amount of time and without losing relevant amounts of information about aerodynamic coefficients' outputs.

Adopting the two relative errors criterion proposed in Tables 3.3 and 3.4, it is possible to assert criterion number 2 (based on a slot of 100 iterations for evaluation) is more reliable since iteration from which criterion holds approximately corresponds to where  $C_L$  and  $C_D$  respectively not change significantly.

<b>AoA [deg]</b>	<b>Iterations Required [-]</b>	Iteration from which error criterion (1) holds - $C_L$	Iteration from which error criterion (2) holds - $C_L$
<b>0</b>	<b>1210</b>	894	1210
<b>10</b>	<b>1142</b>	< 500	870
<b>16</b>	<b>5141</b>	1061	3948

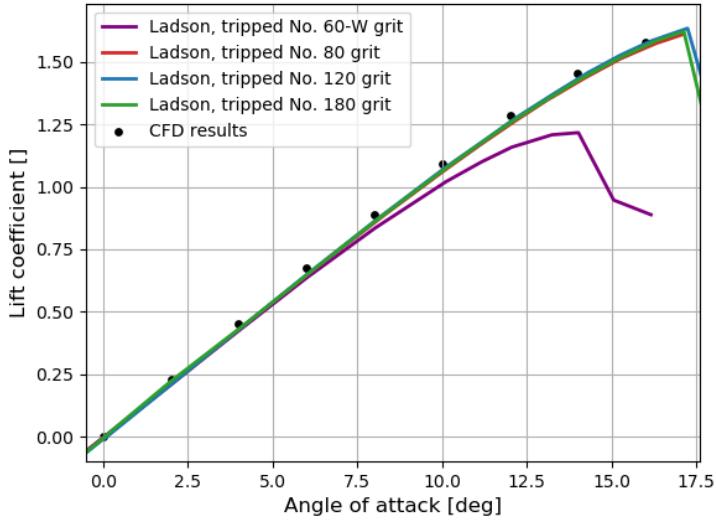
Table 3.3: Errors about lift coefficient of NACA 0012 at 0, 10, 16 deg - Baseline grid.

<b>AoA [deg]</b>	<b>Iterations Required [-]</b>	Iteration from which error criterion (1) holds - $C_D$	Iteration from which error criterion (2) holds - $C_D$
<b>0</b>	<b>1210</b>	< 500	582
<b>10</b>	<b>1142</b>	< 500	952
<b>16</b>	<b>5141</b>	2126	4473

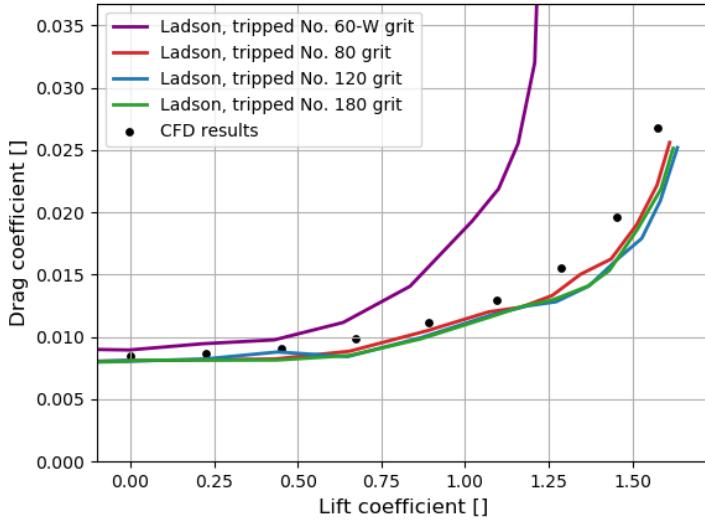
Table 3.4: Errors about drag coefficient of NACA 0012 at 0, 10, 16 deg- Baseline grid.

Figures 3.9a - 3.9b report the plots of lift coefficient with respect to the AoA and drag coefficient with respect to lift coefficient, respectively.

Results have been compared with various experimental data proposed by Ladson with tripped boundary-layer [56]; note that the experimental results were proposed with data clouds: interpolations were executed for better visualization of the trends, and for a subsequent error evaluation.



(a)



(b)

Figure 3.9: Comparison of Ladson experimental data with CFD results of a NACA 0012 with fully turbulent boundary-layer - Baseline grid.

The most important notes about these first CFD results are listed below:

- CFD plots generally have correct trends from a theoretical point of view: linear for what concerns  $C_L(AoA)$ , quadratic for  $C_D(AoA)$ ;
- comparison with experimental data depends on how the artificial boundary-layer transition has been imposed experimentally.

As for the latter, plots in Figures 3.9a - 3.9b show how 60-W grit causes huge effects on aerodynamic experimental performances on NACA 0012, such that 60-W grit's plots are quite different from the other experimental data. Consequently, it does not make

sense comparing the results with tripped 60-W grit proposed by Ladson.

Similar reasoning can be done comparing 80, 120 and 180 grits' experimental results: firstly, the  $C_L(AoA)$  plots are very similar, while  $C_D(C_L)$  plots show a slight difference between each other at higher angles of attack.

In particular, NACA 0012 with 180 grit at 5% of the chord has lower drag coefficient values with respect to 80 and 120 grit; analogously for 120 grit with respect to 80 grit. As a consequence, the comparison between CFD results and experimental data with 80 grit is the most reasonable benchmark.

Before proposing a direct comparison, an observation must be stated: CFD and experimental results are physically slightly different since CFD has been performed using fully turbulent boundary-layer equations, while Ladson experimental data with an artificial transition fixed at 5% of 80 grit.

Considering also the possibility of a low but still existing experimental error in Ladson results, a CFD difference with experimental data is natural. Consequently, the following CFD results' comparison with 80 grit is the most reasonable, but not completely correct; Figures 3.10 and 3.11 report comparison between CFD results and 80 grit experimental data.

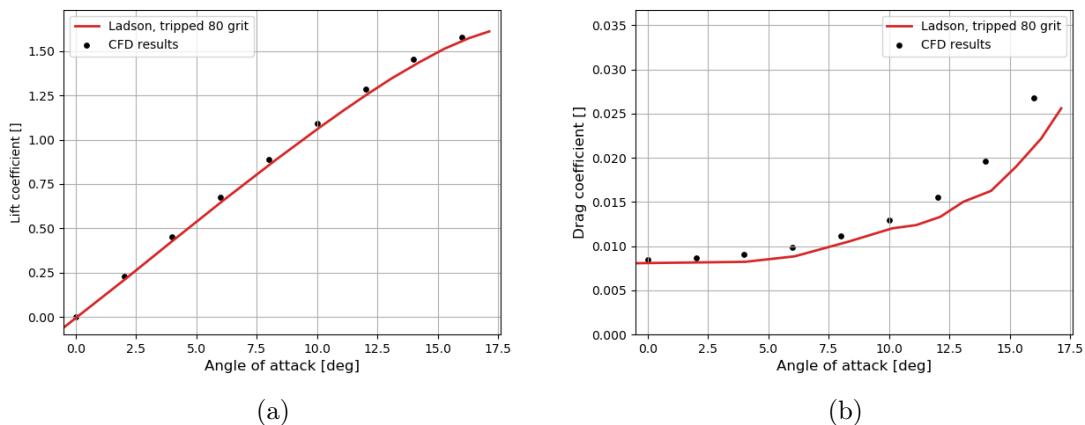


Figure 3.10: Comparison between CFD results and experimental data (80 grit) of a NACA 0012 with fully turbulent boundary-layer - Baseline grid.

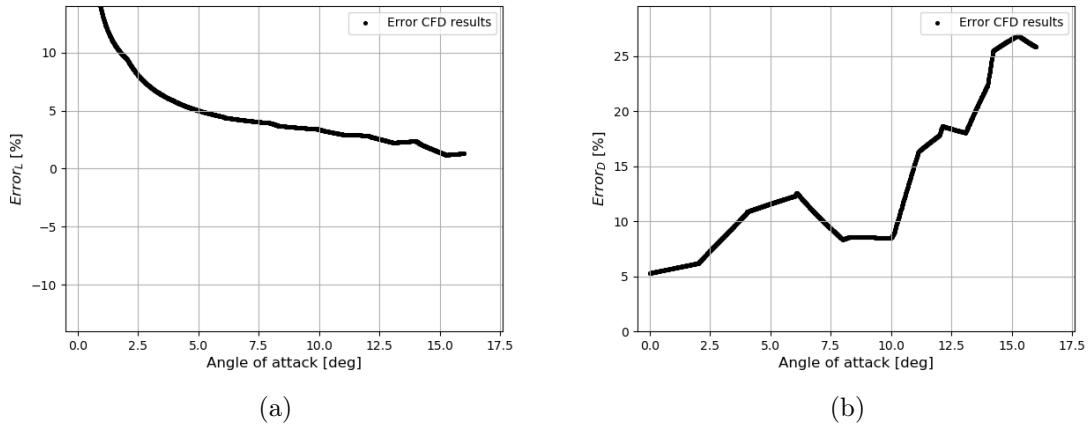


Figure 3.11: Error of CFD results w.r.t. Ladson experimental data (80 grit) about NACA 0012 with fully turbulent boundary-layer - Baseline grid.

Firstly, looking at  $C_L(AoA)$  plots the error is defined by a decreasing curve with higher errors at low angles of attack, although the effective  $C_L(AoA)$  plot does not show relevant differences between experimental data and CFD results, especially at low angles of attack.

Since at low angles of attack the lift coefficient is theoretically zero NACA 0012 is a symmetrical airfoil, slight differences between experimental interpolated results and CFD ones causes high relative (%) errors.

For example, at 0 degrees:

- CFD results:  $C_L = -0.000021$ ;
- Ladson experimental data tripped 80 grit, interpolated:  $C_L = -0.003$ .

Although the difference is almost negligible, the computed error becomes relevant. As a consequence, this comparison about lift makes sense only for higher angles of attack. If the lift coefficient values are rounded to the fourth digit, (for example), the error tends to zero also at low AoA.

Here, a general overestimation of CFD results about lift coefficient is noted, but the errors stay under 10% with a continuous percentage decreasing when higher angles of attack are taken into account.

About drag coefficient results, the most important observation to make is about the increasing difference at higher angles of attack.

Considering for example an angle of attack of 12 deg, baseline provided a lift coefficient of 1.285253 and drag coefficient of 0.015563, with a relative error w.r.t. angle of attack of approximately 17.5%.

As already cited, 80 grit is the most reasonable comparison with CFD fully turbulent boundary-layer results, but an amount of uncertainty remains since the behaviour of drag coefficient directly depends on grit considered [56] for artificially boundary-layer tripping.

### 3.3 Sensitivity Analysis - NACA 0012 FT

In the following, some attempts of grid performance improvements are presented, trying to reduce the overall error, especially focusing on drag coefficient since it generated the most relevant differences with respect to experimental Ladson data 80 grit. The focus of the following grid changes is finalized to present sensitivity with respect to some parameter variations, evaluating when further refining or parameter changes do not provide CFD output variations.

The angle of attack will be always 12 deg with the same flow conditions on NACA 0012. The focus will be on the relative error of  $C_D$ , which has a starting value of approximately 17.5%.

A serial method was applied for the present sensitivity analysis: the grid parameter which guarantees performance improvements is selected and used for the following studies. As a consequence, it is important to underline that results can change if a different order of parameter variation is considered; anyway, these possible results' changes are not expected to be relevant.

#### 3.3.1 Farfield Size

Firstly, the farfield size was varied to analyse how it influences the aerodynamic coefficients. Literature suggests an error on drag coefficient if the farfield is too small in size. NASA Turbulence Modeling Resources about NACA 0012 [57] suggests a farfield size of 500 chords to avoid adverse impact on aerodynamic performances.

As a consequence, from the choice adopted of 100 chords in baseline grid for NACA 0012 it is expected a variation if the farfield size is augmented, possibly reducing computational error.

Farfield Size per chords	Lift Coefficient ( $C_L$ ) [-]	Drag Coefficient ( $C_D$ ) [-]
50	1.283297	0.015880
100	1.285253	0.015563
250	1.285463	0.015367
500	1.286332	0.015306
750	1.286040	0.015286

Table 3.5: CFD aerodynamic performances of NACA 0012 - Farfield size variations.

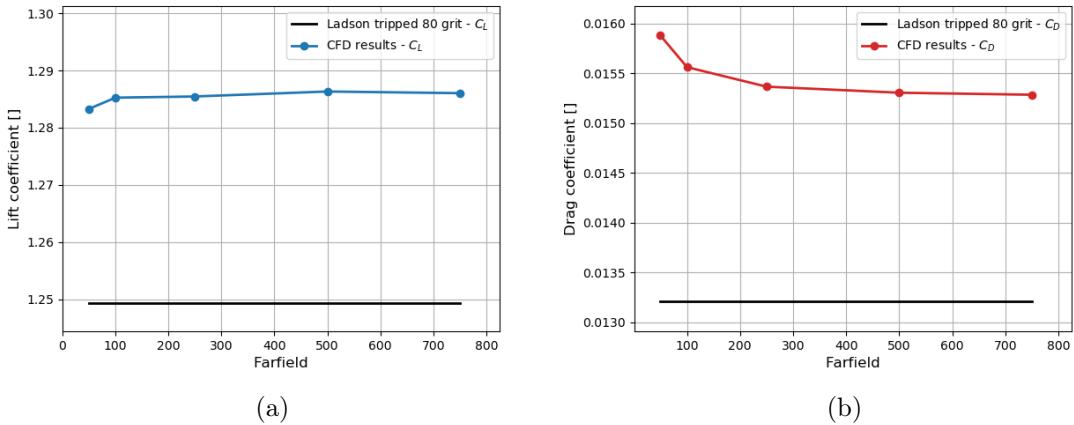


Figure 3.12: Comparison between CFD results and experimental data for a NACA 0012 at 12 deg - Farfield size variations.

About the lift coefficient (Figure 3.12a), the plot shows relevant changes only when the farfield size is reduced to 50 chords. The lift prediction seems not to be significantly affected by the grid when farfield size is above 100 chords, where also the error with respect to Ladson data stabilizes at around 2.8%-2.9%.

Looking at the drag coefficient plot (Figure 3.12b), changes become relevant at bigger farfield sizes with respect to lift coefficient: from approximately 250 farfield size, drag coefficient does not significantly change; at lower values, an effect of farfield size on drag coefficient output can be observed. The error follows the same trend, going from approximately 20% at 50 farfield size to an almost constant value of 16% when the farfield size overcomes 250.

Increasing the bounds of the farfield does not present relevant changes in the total amount of grid elements, remaining at approximately 380000 - 385000 cells. The reason for this is because the farfield presents a coarser grid with respect to near airfoils' grid region. As a consequence, the computational time does not vary significantly.

For the next analyses about grid sensitivities, the farfield size is increased to 500. This value is in agreement with the similar NACA 0012 CFD test case presented by NASA [57].

### 3.3.2 Global grid refinement

Through a variation of the parameter *nodes*, a general refinement on the entire grid is performed. This parameter simply scales the number of nodes on each relevant line or spline of the grid (farfield, airfoils' surfaces etc.) and reduces the smallest grid element's dimension.

Table 3.6 and Figures 3.13a and 3.13b report the main CFD results obtained.

Parameter <i>nodes</i>	Grid Elements	Lift Coefficient ( $C_L$ ) [-]	Drag Coefficient ( $C_D$ ) [-]	Time requested [min]
<b>0.70</b>	215276	1.288443	0.015438	26
<b>0.85</b>	312079	1.287094	0.015326	72
<b>1.00</b>	380100	1.286332	0.015306	75
<b>1.15</b>	532615	1.285043	0.015278	90
<b>1.30</b>	600022	1.284649	0.015270	102

Table 3.6: CFD aerodynamic performances of NACA 0012 - Global refining.

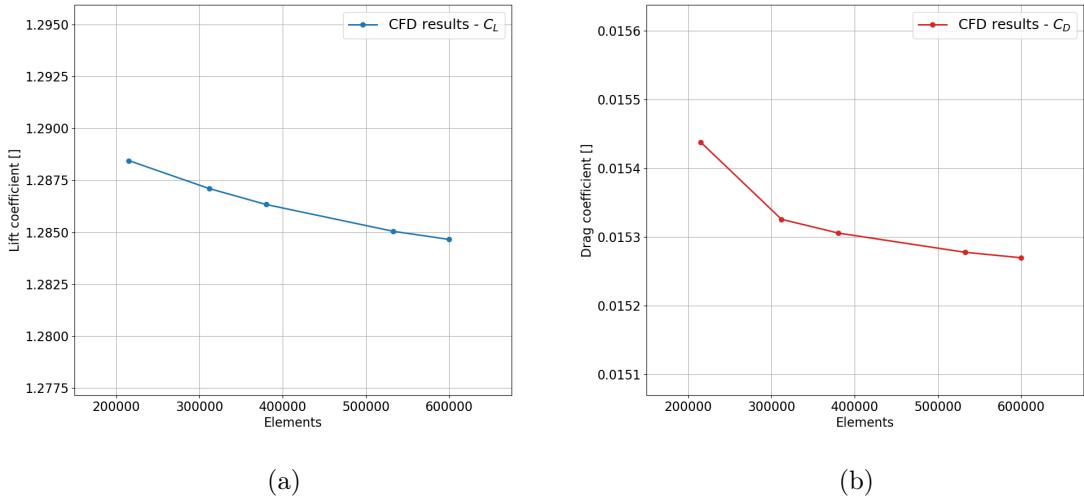


Figure 3.13: CFD results of NACA 0012, 12 deg - Global refining.

Results show that the drag coefficient does not change when the refinement increases, while the lift coefficient sees a slight decrease. Anyway, these changes are very low and almost negligible, especially when considering the computational time required for these fined grids. The differences with Ladson experimental when large refining is imposed is limited under approximately 0.2% for lift coefficient and 1% for drag coefficients when the grid refining generates grids with a total amount of element greater than 300000 elements.

Table 3.6 last column reports the computational time required, considering the same convergence criterion of the previous section. The trend is not respected by the second row (parameter  $nodes = 0.85$ ), where the requested computational time to satisfy the convergence criterion is higher than expected. This unexpected result is due to RANS stability needs about this simulation: in particular, using a CFL 1500 (like the previous simulations for baseline grid and farfield changes) makes the RANS simulation diverging; as a consequence, the CFL has been reduced to 500. The simulation reaches convergence criterion, but CFL reduction usually means lower speed. This effect will appear present again in the following grid sensitivity analyses.

Since general grid refinement seems not to affect aerodynamic coefficients significantly, the parameter  $nodes$  is then maintained with the value 1.00 (like baseline was) to avoid high computational times.

### 3.3.3 Wake: refinement size and length

The next sensitivity analysis is focused on grid wake refinement. As already presented in Section 3.1.2, the wake refinement is defined by two lines of grid points. These lines have the following main features:

- line slope defined by the airfoils' elements angles of attack, such that the entire wake is enclosed and detected;
- grid points are 50 and spatially proceed with a geometrical series (of coefficient 1.20 - 1.21) from point number 20 to guarantee an immediate refinement closer to the airfoil;
- grid points define refining according again to a geometrical series. The elements' size progression parameter is customizable by the user through the item *wake\_progr*;
- the maximum distance reached by these points can be customized by the user through the item *wake\_length*. The value is proportional to the major axis of the ellipse.

Previous analyses were performed with a wake length of *wake\_length* = 50 and *wake\_progr* = 1.20.

Wake refinement is a key point since it directly affects aerodynamic coefficients, especially pressure drag. Then, its gradient detection becomes relevant, but exceeding in the refining could be also inefficient from a computational time point of view. As a consequence, ideal tuning of these parameters has been performed on the NACA 0012, trying to delineate both aerodynamic coefficients changes and the total number of elements inside the grid (and, consequently, computational costs).

Like for farfield size's sensitivity analysis, the aerodynamic coefficients reported in Tables 3.8 - 3.7 show slight variations in the results. Figures 3.14 and 3.15 propose coefficients' errors with respect to Ladson data.

Length \ Size progr.	1.15	1.20	1.30	1.40
Length				
30	1.286195	1.285852	1.286122	1.286618
50	1.285729	1.286332	1.286330	1.286399
70	1.286522	1.286134	1.286235	1.286599

Table 3.7: CFD aerodynamic performances of NACA 0012 - Lift coefficient results w.r.t. grids' wake variations.

Length \ Size progr.	1.15	1.20	1.30	1.40
Length				
30	0.015299	0.015317	0.015351	0.015391
50	0.015300	0.015306	0.015315	0.015338
70	0.015300	0.015306	0.015307	0.015315

Table 3.8: CFD aerodynamic performances of NACA 0012 - Lift coefficient results w.r.t. grids' wake variations.

Length	Size progr.	1.15	1.20	1.30	1.40
		30	50	70	
30		448334	286712	197374	173516
50		632030	380100	256142	216144
70		911980	488130	324630	266556

Table 3.9: CFD aerodynamic performances of NACA 0012 - Number of elements for each grid with wake's features variations.

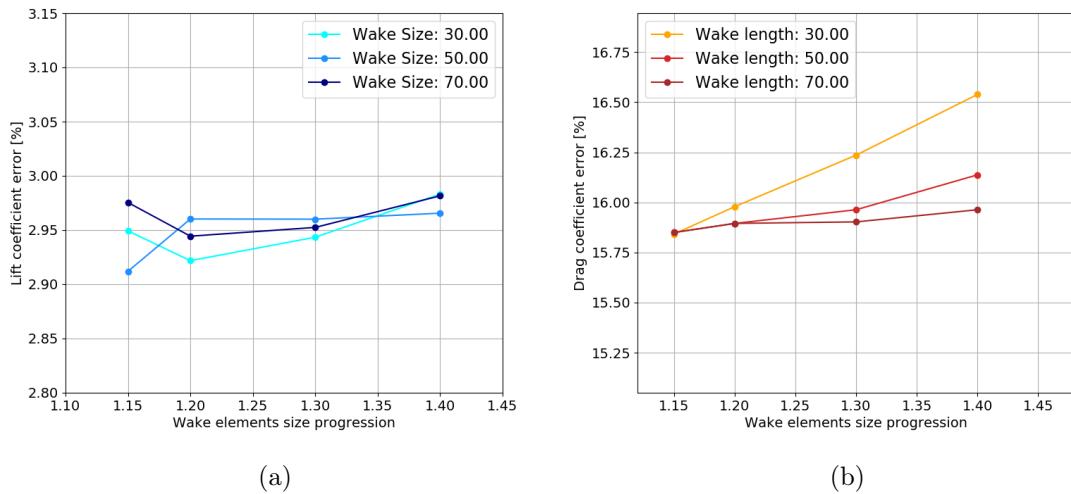


Figure 3.14: Error of CFD results with respect to Ladson experimental data (80 grit), 12 deg - Grid Wake analysis.

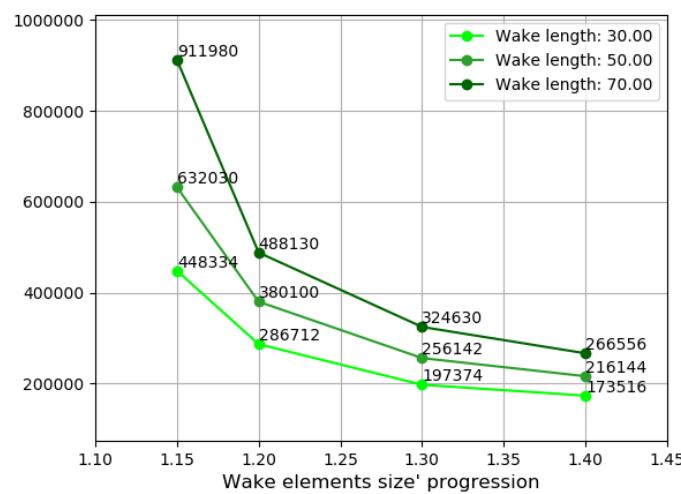


Figure 3.15: Number of elements for each CFD simulation performed about grid wake analysis.

Input parameter *wake\_length* defines how far wake grid refinement gets. Consequently, higher values of this input parameter guarantees much more refined grid.

Secondly, by reducing wake elements size progression through parameter *wake\_progr*, the grid results more refined: for example, 1.15 means each grid point belonging to the wake will have a grid size dimension 15% higher than the previous one located closer to the airfoils. Consequently, at a particular distance downward to the airfoil configuration, between (for example)  $wake\_progr = 1.15$  and  $wake\_progr = 1.40$ , the first will generate a much more refined grid, as Figure 3.15 shows plotting the total amount of elements of each test case.

Some observations can be made by analysing the plots:

- the lift coefficient seems not to be significantly affected by a wake length increase or refining by wake elements' size progression.
- the drag coefficient is affected by both parameters, but slightly. The most important variations are observable wake length and elements' size progression are excessively decreased (separately or contemporaneously) to values 30 and 1.40 respectively since the grid becomes too coarse.

As a consequence, to meet both requirements of computational costs and accuracy in aerodynamic coefficients, an intermediate choice between wake length and elements' size progression can be done.

From the baseline parameters of  $wake\_length = 50$  and  $wake\_progr = 1.20$ , the first will be maintained while the second is increased to a value of 1.30, reducing the number of total elements (from 380000 to 256000 grid's elements, approximately) and consequently computational time, without affecting the aerodynamic coefficients results.

### 3.3.4 Ellipse Size and grid's external semi-circle

Without proper grid management upwards of the airfoil's configuration, the results risk providing a relevant error. The tool applies a two-level structure to manage this upward refinement: a grid ellipse close to wall surfaces and an external semi-circular ensemble of grid points, far from the airfoils.

Without the presence of a refinement close to the airfoil, the grid immediately appears too coarse. The presence of a grid ellipse close to the wall surfaces guarantees also a refinement where the flow thermodynamic gradients become crucial, like more left foil leading edge region.

Anyway, the simple presence of a grid ellipse results insufficient using the unstructured mesh algorithm of Frontal-Delaunay: although its positioning, first tests found a large error without further upward refinement.

Recall that item *ellipse\_dimension* regards the left-hand side value related to the inequality 3.16.

The tests performed without external refinement are reported in Table 3.10 for various grid ellipse dimensions.

Ellipse equation parameter	Corresponding Ellipse Area	Lift Coefficient ( $C_L$ ) [-]	Drag Coefficient ( $C_D$ ) [-]	Number of elements
<b>0.35</b>	0.482	1.286726	0.015371	412664
<b>0.55</b>	0.305	1.282093	0.015451	284238
<b>0.85</b>	0.193	1.287409	0.015698	200356

Table 3.10: Effect of grid ellipse with various sizes when external semi-circle points are not implemented - CFD results.

Table 3.10 shows how lower refinement around the airfoil configuration causes an increase in the drag coefficient. On the other hand, with a large dimension of the grid ellipse (first row in the table), a proper refinement has been reached, since the aerodynamic coefficients are similar to the ones obtained with the presence of an external semi-circle refinement. These results demonstrate that refining around and upward airfoil configuration is necessary to reduce the errors caused by too coarse grids.

To guarantee wider customization to the user, two refinement levels have been proposed. Tables 3.11 - 3.12 - 3.13 report the aerodynamic coefficients of various tests executed on a single grid ellipse configuration ( $ellipse\_dimension = 0.85$ , which corresponds to the ellipse with lower area 0.193) with several combinations of external grid semi-circle. The following list reports the meaning of the two variables related to these analyses along with the assigned baseline values:

- geometrical dimension:  $semicircle\_dimension = 13$ , which is proportional to ellipse and/or wake length.
- elements size:  $semicircle\_elem\_factor = 250$ . This term multiplies the smallest grid element size to obtain the external grid semi-circle points' size.

To simplify the discussion, the dimensions of the ellipse will be no longer presented using their respective input parameter  $ellipse\_dimension$ , but directly by their respective area.

Moreover, like it was for previous sensitivity analysis since the coefficients do not change significantly, the plots are respectively proposed in terms of aerodynamic coefficients compared to Ladson experimental data.

Dimension \ Element size	50	75	150	250	300
4	299252	265206	231118	212380	205878
8	-	359784	270626	238102	229112
<b>13</b>	-	392518	303870	256142	247482
18	-	466766	330006	282492	267870
23	-	514082	372288	295722	278876

Table 3.11: CFD aerodynamic performances of NACA 0012 - Number of elements for each grid with external grid's semi-circle variations.

Dimension \ Element size	50	75	150	250	300
4	0.015311	0.015313	0.015339	0.015397	0.015459
8	-	0.015297	0.015319	0.015331	0.015347
13	-	0.015294	0.015308	0.015315	0.015324
18	-	0.015294	0.015308	0.015305	0.015314
23	-	0.015298	0.015299	0.015305	0.015319

Table 3.12: CFD aerodynamic performances of NACA 0012 - Drag coefficient for each grid with external grid's semi-circle variations.

Dimension \ Element size	50	75	150	250	300
4	1.286556	1.285801	1.286512	1.286555	1.284534
8	-	1.286140	1.286138	1.286164	1.285811
13	-	1.286132	1.285870	1.286330	1.285806
18	-	1.286177	1.286025	1.286109	1.286080
23	-	1.286166	1.286035	1.286153	1.285913

Table 3.13: CFD aerodynamic performances of NACA 0012 - Lift coefficient for each grid with external grid's semi-circle variations.

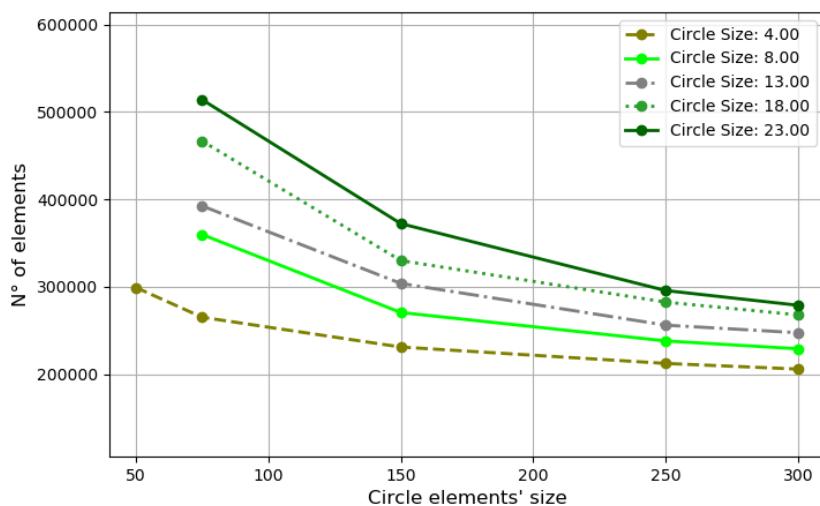


Figure 3.16: Amount of elements of each test case for external semi-circle grid sensitive analysis.

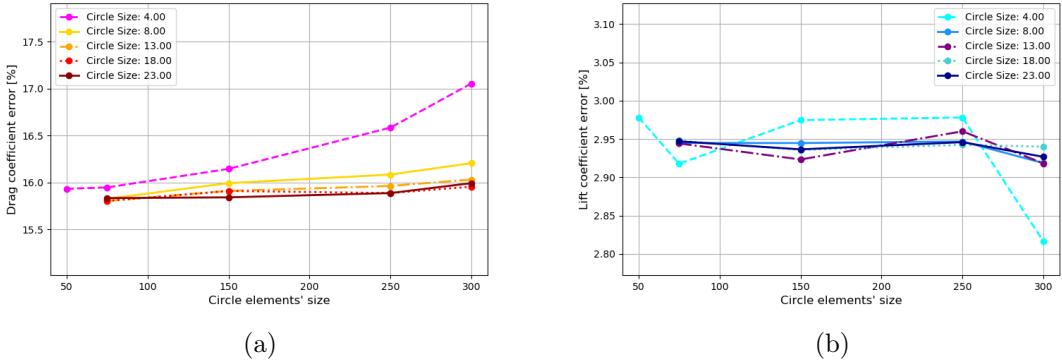


Figure 3.17: Errors of CFD computed results about lift and drag coefficients considering a single ellipse dimension 0.85 and various external grid semi-circle features.

Tables 3.11, 3.12, 3.13 and Figures 3.16, 3.17 regarding the comparison between one single ellipse and several external grid semi-circles propose some interesting results. The majority of combinations about semi-circle dimension and elements' size do not involve relevant variations about aerodynamic coefficients.

Figures 3.17a and 3.17b show that the only semi-circle with relevant changes about aerodynamic coefficients is the one with a lower size (4, magenta and cyan dashed lines). About the lift coefficient, this setup is the only one which steps away from an almost constant error of approximately 2.95% value found for the other setups.

About the drag coefficient, the setup with 4 grid semi-circle's dimension sees a general increase of error; moreover, the differences increase when the coarser external semi-circle of dimension 4 is considered.

As a consequence, the main deduction from these plots and results is the following: it makes sense to avoid external semi-circle with lower geometrical dimensions and with high elements' sizes, probably because both involve a grid that is too coarse upward of the airfoils. The opposite condition (higher semi-circle geometrical size) guarantees sufficiently fine grids with results almost unvaried and independent from ellipse size. On the other hand, a finer grid (lower semi-circle elements' sizes) seems to reduce the differences between outputs of various semi-circle geometrical dimensions; anyway, as Figure 3.16 shows, this condition involves a higher total amount of elements inside the grid and consequently higher computational costs.

Other results are proposed for different ellipses sizes: since the combinations are several and not easily explainable through tables, trends will be displayed in Figures 3.18 and 3.19.

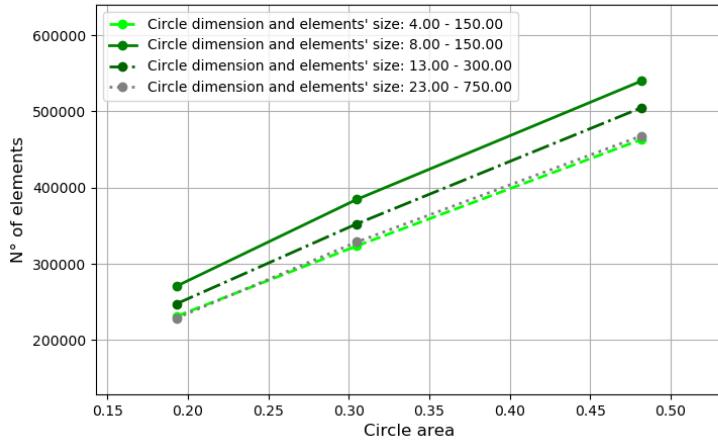


Figure 3.18: Number of elements for each CFD grid simulation considering various grid's ellipse and external semi-circle points.

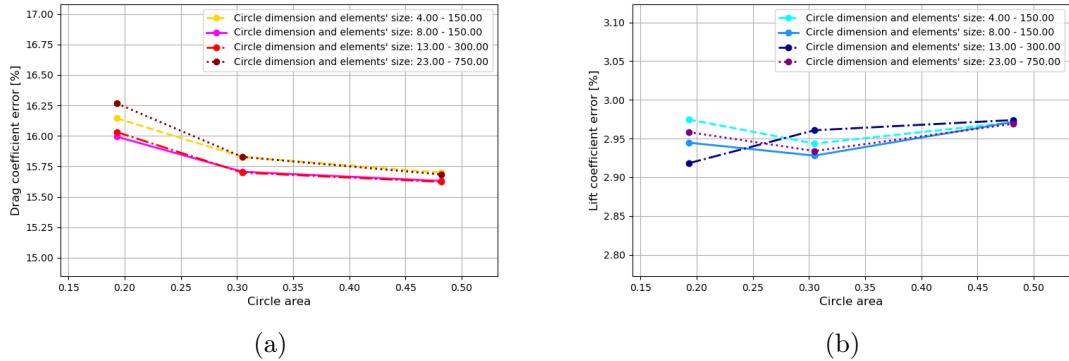


Figure 3.19: CFD Errors about lift and drag coefficients considering various grid's ellipse and external semi-circle points.

Between Figures 3.18 and 3.19 where several combinations of upward refinement through external grid ellipse and external semi-circle, a unique relevant conclusion can be exported: increasing the area of the ellipse, the drag seems to converge to the same results (Figure 3.19a), reducing the error. Anyway, a higher ellipse area involves a higher total amount of elements as for computational cost, this variation (although visible) remains inconvenient.

From the cited results and observations, from a starting configuration of:

- $\text{ellipse\_dimension} = 0.85$ ;
- $\text{semicircle\_dimension} = 13$ ;
- $\text{semicircle\_elem\_factor} = 250$ ;

the only variation which makes sense is increasing the geometrical size of the grid's semi-circle, while others changes are not so reliable or may cause unnecessarily fine grids.

Consequently, the next analysis will use  $\text{semicircle\_dimension} = 23$ , which means considering a general finer grid with respect to the previous section (going from 256000 to 296000 grid's elements, approximately).

### 3.3.5 Wall refinement

A wall refinement has been performed firstly. This section is governed by the item  $\text{wall\_refining}$ , fixed at unit value during the previous sensitivity analysis and in baseline grid.

This parameter affects proportionally the total amount of nodes on each airfoil's element surface.

Wall refinement item	Total amount of grid's elements	Wall nodes*	Lift Coefficient ( $C_L$ )	Drag Coefficient ( $C_D$ )
1.00	295722	596	1.286153	0.015305
1.25	314722	749	1.285111	0.015285
1.50	333960	899	1.284162	0.015283
1.75	352878	1047	1.284053	0.015285

Table 3.14: CFD aerodynamic performances of a NACA 0012 - wall refinement variations.

Note \*: the nodes placed on cut TE line are fixed to 50 and excluded from Table 3.14.

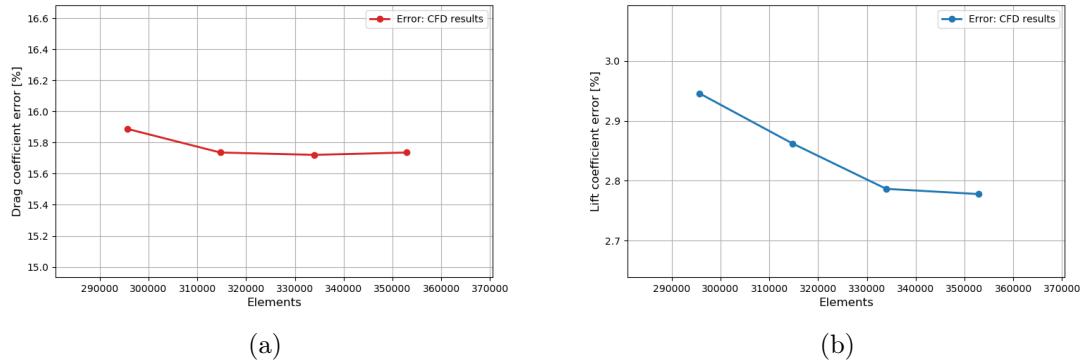


Figure 3.20: Errors of CFD computed results about lift and drag coefficients considering increased wall refinement.

The results do not show relevant changes: the values seem to stabilize when  $\text{wall\_refining}$  parameter is fixed on 1.25 - 1.50, from which drag and lift coefficient do not show further changes higher than 0.1% approximately. Furthermore, with these input values the total amount of wall surface nodes corresponds to a similar test case executed on NACA 0012 on SU2 website [32].

For the next and final sensitivity analysis the wall surface nodes has been increased imposing  $\text{wall\_refining} = 1.25$ , which means passing from a total amount of wall surface nodes of 596 to 749 for the NACA 0012 test case.

### 3.3.6 Structured region analysis

The most interesting results about sensitivity analysis of the proposed hybrid grid are the ones related to the structured region variations.

Progression \ Thickness	0.5	1.0	1.5
1.05	1.292113	1.289809	1.297345
1.10	1.280544	1.285111	1.293521
1.15	1.272630	1.278291	1.287829

Table 3.15: CFD aerodynamic performances of NACA 0012 - Lift coefficient. structured region's changes.

Progression \ Thickness	0.5	1.0	1.5
1.05	0.014972	0.015134	0.015067
1.10	0.015258	0.015285	0.015257
1.15	0.015597	0.015603	0.015550

Table 3.16: CFD aerodynamic performances of NACA 0012 - Drag coefficient, structured region's changes.

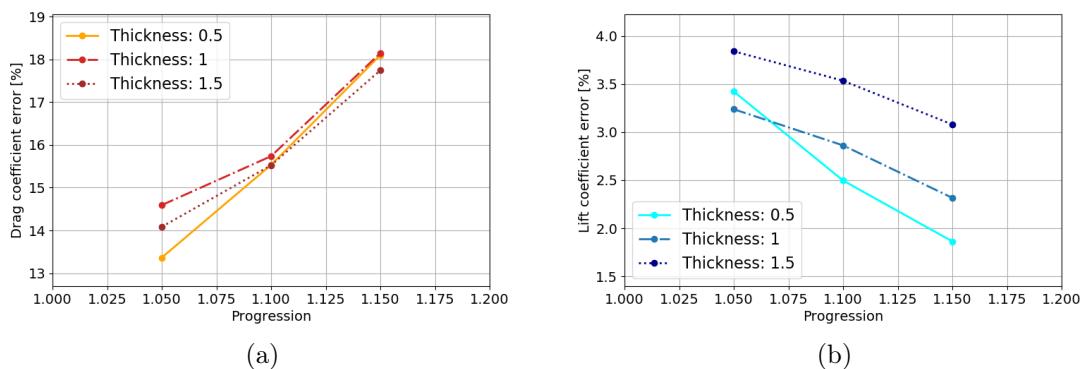


Figure 3.21: Errors of CFD computed results about lift and drag coefficients considering several options about structured region.

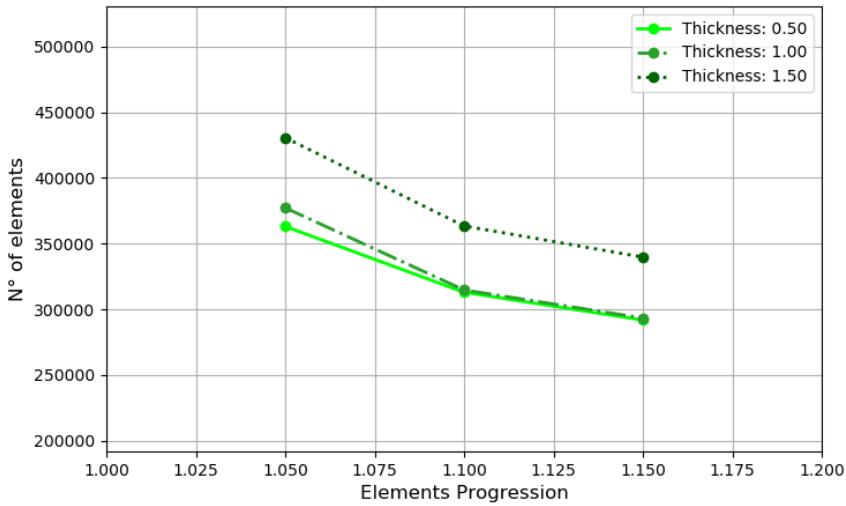


Figure 3.22: Number of elements for each CFD grid simulation considering various structured region's changes.

As expected, decreasing the progression of normal structured element size causes a general increase in the total amount of elements inside the grid. On the other hand, structured region thickness causes an increase of total grid elements' amount for high thickness cases only.

By looking at the lift coefficient, the grid behaves with a general decrease when higher progressions are taken into account; the drag coefficient increases for higher structured regions' progressions.

About thickness, results show a decrease in the error when a lower structured region's thickness is considered.

It is important to remark that these results have been obtained using hybrid grid: the interface between structured and unstructured region becomes crucial, since there the elements' sizes have to be comparable.

The results report some hints about this topic: as shown by Table 3.17, lower thickness and higher progressions provides faster results, with the latter being more relevant. Anyway, the fastest possibility (progression 1.15 and thickness 0.5) presents the highest value of the lift coefficient and, consequently, of the error. For the cases of multi-element airfoils where optimisations are usually performed about maximum lift coefficient, speed is as desired as precision; as a consequence, this fastest structured region's setup could be not appropriate for the analysis, while other setups with a proper balance between speed and precision can be considered.

As last modification for the test case under analysis about NACA 0012, the thickness has been reduced to 0.5 along with a progression of structured elements reduced to 1.05. As a consequence, the total amount of elements become 363000 approximately. These choices were selected to analyse how much computational cost and aerodynamic coefficients could vary after these changes maintaining approximately the same total grid elements amount of baseline test case.

Thickness Progression	0.5	1.0	1.5
1.05	1622	1938	1897
1.10	1373	1482	1445
1.15	1174	1283	1335

Table 3.17: CFD aerodynamic performances of NACA 0012 - Iterations required.

### 3.4 NACA 0012 FT - Final grid CFD results

With the previous considerations and notes about grid sensitivity, a new polar analysis has been performed. The results are reported below. It is possible to note a slight error decrease with respect to Ladson experimental data 80 grit. The comparison is valid within the possible uncertainties already presented at the beginning of the present chapter.

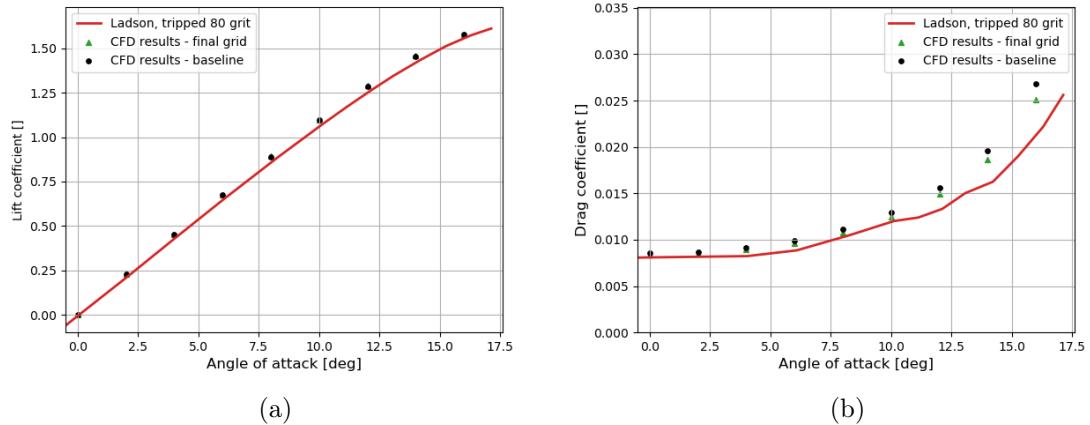


Figure 3.23: Comparison between CFD results (both baseline and final grid) and experimental data of a NACA 0012 with fully turbulent boundary-layer.

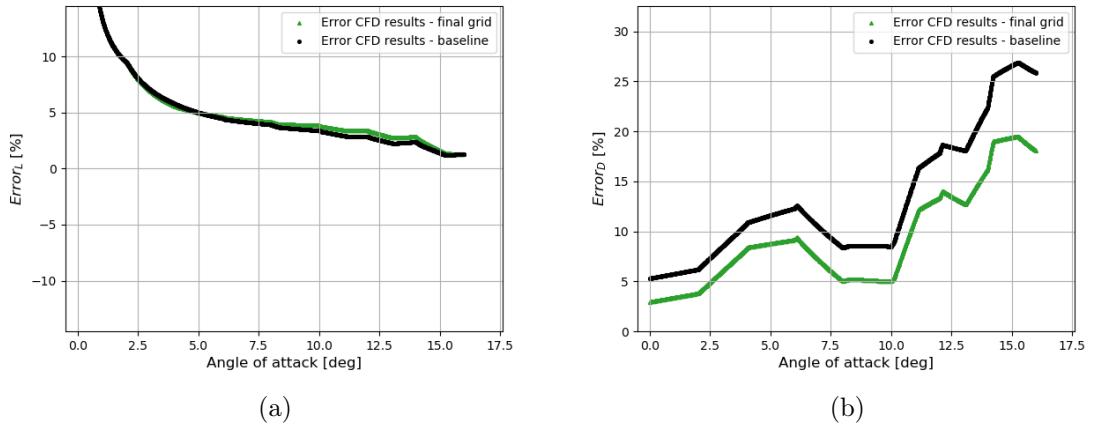


Figure 3.24: Error of CFD results (both baseline and final grid) w.r.t. experimental data (Ladson 80 grit) of a NACA 0012 with fully turbulent boundary-layer.

The differences are appreciable when the errors are compared: a significant reduction in the drag error has been achieved (from approximately 2.5 % at low AoA to 8 % at high AoA), in particular at lower angles of attack. Although the highest drag errors' reduction can be noted at high AoA, the difference with respect to Ladson experimental data 80 grit is still significant. However, it is important to remind that the direct comparison with Ladson experimental data with tripped boundary layer (80 grit) has a margin of uncertainty, as presented at the beginning of the Section (Figures 3.9b). For these reasons, next Section 3.5 will firstly perform other CFD results with different geometries comparing with experimental data.

Resuming the analysis, lift coefficient remained almost unvaried. Regarding the computational cost, at an angle of attack of 12 degrees the total amount of elements was reduced from approximately 383000 to 363000 after the grid sensitivity analysis changes. Since these changes regarded all the angles of attack, each one received a reduction of total amount of elements within the respective grid.

Looking at Tables 3.18 - 3.19, the two convergence criterion defined at the beginning of the present chapter are satisfied at an iteration much lower than exit convergence one, which means less thresholds could be used for these simulations without causing relevant additional error, like it was for baseline results. On the other hand, the computational time required for each simulation using this final grid has increased, along with the respective iteration for the satisfaction of two error criterion. Within Tables 3.18 and 3.19 the baseline history performances are presented inside round brackets: although still acceptable, it is important underlying that the test case was deal with a quite simple geometry as the NACA 0012 is.

The only CFD simulation which took less time was at AoA 12 deg, the one used for the ssensitivity analysis. As a consequence, the entire study presented should be extended to several angles of attack for a proper evaluation of computational times and respective RANS speed.

Low speed to convergence could become unacceptable when more complex geometries like multi-element airfoils are taken into account, since they naturally involve higher number of grid elements and a more complex flow.

Firstly, although pointing to convergence, the final grid's simulations show higher oscillations during RANS history, as Figures 3.25 - 3.26 report (example reported at

$\text{AoA} = 8$  deg about drag coefficient). The magnitude of these oscillations are not so relevant, since limited inside a range of 0.005% approximately; moreover, the result tends to convergence like baseline does.

Tables 3.18 and 3.19 report that final grid tends to convergence slower than baseline grid. The reason could be related to the more relevant oscillations in RANS histories, affecting satisfaction speed of the adopted convergence criterion (which were density residual and Cauchy series on lift and drag coefficients).

AoA [deg]	Computational Time	Iterations Required [-]	Iteration from which error criterion (1) holds - $C_D$	Iteration from which error criterion (2) holds - $C_D$
0	50 (33)	1618 (1210)	1176 (< 500)	1618 (582)
10	46 (45)	1389 (1142)	< 500 (< 500)	806 (952)
16	258 (196)	7524 (5141)	1264 (2126)	5655 (4473)

Table 3.18: CFD errors about drag coefficient of NACA 0012 - Comparison between Final and baseline grid.

AoA [deg]	Computational Time	Iterations Required [-]	Iteration from which error criterion (1) holds - $C_D$	Iteration from which error criterion (2) holds - $C_D$
0	50 (33)	1618 (1210)	< 500 (< 500)	843 (582)
10	46 (45)	1389 (1142)	< 500 (< 500)	1050 (952)
16	258 (196)	7524 (5141)	1383 (2126)	6465 (4473)

Table 3.19: CFD errors about drag coefficient of NACA 0012 - Comparison between Final and Baseline grid.

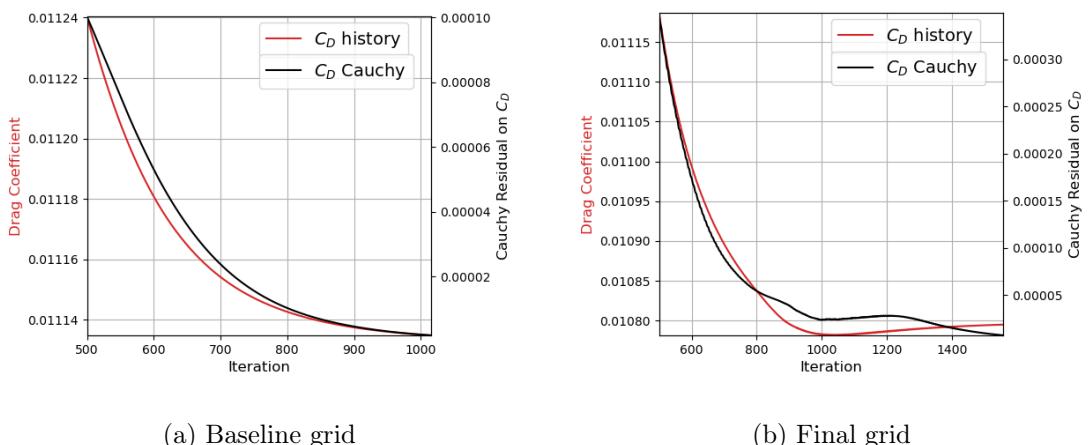


Figure 3.25: Comparison between final and baseline grid's RANS History performances about NACA 0012 - 8 deg.

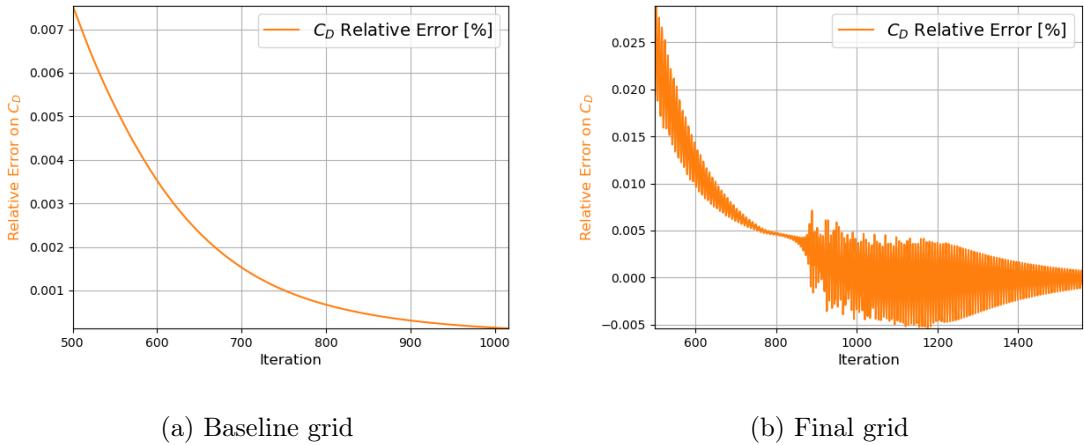


Figure 3.26: Comparison between final and baseline grid's RANS History performances about NACA 0012 - 8 deg.

Summarising, the presented results report the following possibilities:

- a reduction of aerodynamic coefficients' errors can be achieved by properly setting tool's refining parameters;
- although the final setup guarantees a general reduction on the total amount of elements along with errors decrease, various parameters could affect computational speed. Consequently, the total amount of elements is only one of the features to take into account when treating computational speeds' performances. About this topic, structured region features (thickness and progression) appear as relevant aspects.

The user can customize various parameters, leading the simulations to lower errors or increasing the computational speeds. The sensitivity analysis proposed in previous paragraphs become useful for the tool usage; further analyses about could provide wider results and observations about computational speed, accuracies, may extending to wider ranges, combinations, geometries and flow conditions. Since the structure of grid generation code is not intended to be drastically changed in future tool's upgrades, the grid sensitivity analysis performed in this chapter will remain valid.

### 3.5 Other CFD results

The grid structure obtained from the previously-presented sensitivity analysis is used to simulate the NACA 0012 and other airfoils. In this Section, the results of these additional simulations are presented and discussed.

These results are intended to extend the grid validation to different geometries and flow conditions.

Each simulation reported has a  $y^+ < 1$ : further details about comparison between input  $y_{\text{plus}}$  and actual  $y^+$  CFD result can be found in Appendix B.

#### 3.5.1 NACA 0012 with BC transitional boundary-layer model

The simple geometry of NACA 0012 is considered again, but this time imposing a transitional-boundary layer model.

The tool's parameters are the same of the final grid, except for structured region where it has been considered a thickness of 0.5 and elements' normal progression of 1.10.

As a consequence, aerodynamic coefficients can be compared with proper experimental results which involve free-transition boundary-layer, like ones proposed again by Ladson [56] or by Abbott and Von Doenhoff [1]. Although older, the results proposed by Abbott and Von Doenhoff are more reliable since they are based on known free-stream turbulence intensity at Langley Wind Tunnel (fundamental data for free transitional experimental tests). Oppositely, Ladson results have a margin of uncertainty since the experimental tests were performed after Langley Wind Tunnel suffered damage, with a consequent limited but not irrelevant increasing in free-stream turbulence intensity; comparing the two reference experimental results, Ladson reports higher drag coefficients, probably symptom of different free-stream turbulence intensity.

Experimental tests of Abbott and Von Doenhoff were performed with a free-stream turbulence intensity ( $Tu_\infty$ ) of 0.15% at Reynolds number of  $6 \cdot 10^6$ . Since reference considers an untripped boundary-layer, the transitional model SA-BC proposed by Bas et al. [40] [41] along with Spalart-Allmaras turbulence model was adopted. As already presented in Section 2.3.3, the model is almost entirely based on empirical correlation with free-stream turbulence intensity, which consequently takes a fundamental role.

Since the available results of Abbott are limited and do not provide drag coefficients for high angles of attack close to stall conditions for the cited case, the results of Ladson [56] are anyway provided in Figures 3.27 and 3.28 for comparison.

The range previously adopted on NACA 0012 fully turbulent boundary-layer tests of  $\text{AoA} = [0, 16]$  is extended to 17 and 18 degrees to delineate aerodynamic coefficients on stall and post-stall conditions.

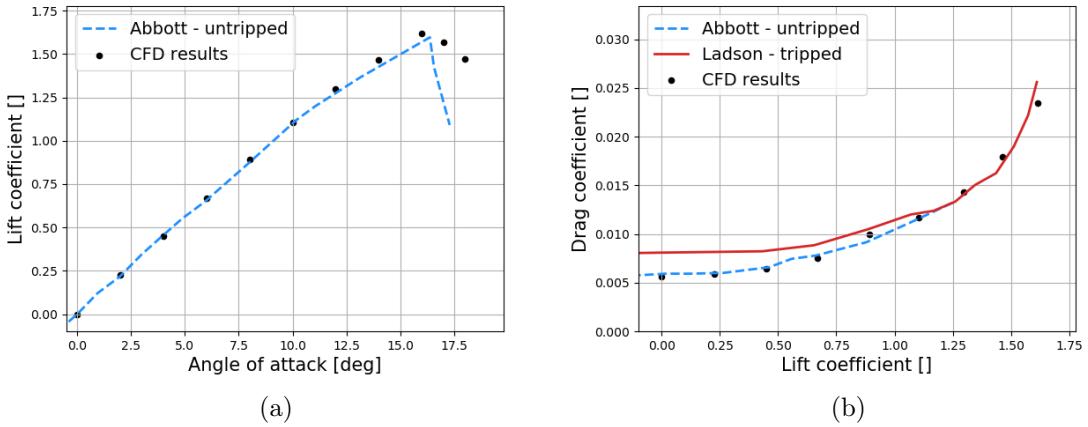


Figure 3.27: Comparison between CFD results and Abbott (in blue) and Ladson (in red) experimental data about NACA 0012 with free transitional boundary-layer.

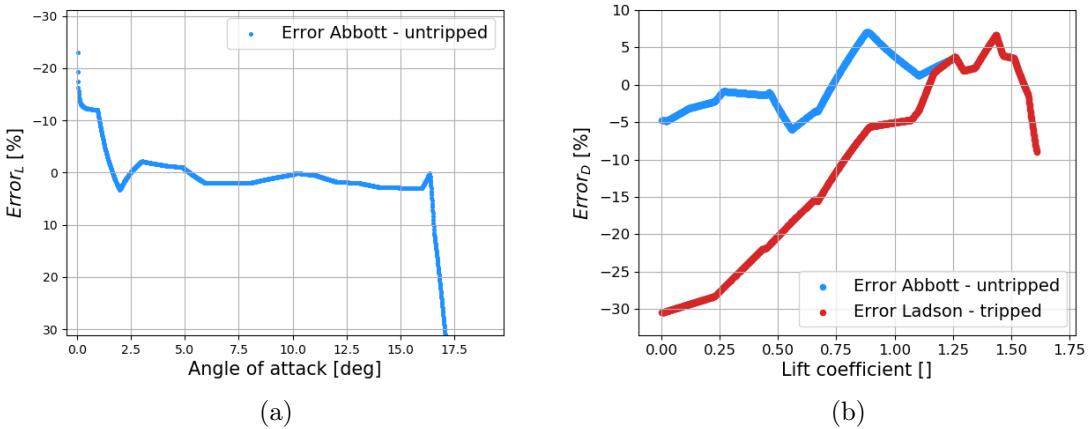


Figure 3.28: Error between CFD results and Abbott (in blue) and Ladson (in red) experimental data of a NACA 0012 with free transitional boundary-layer.

As it was for NACA 0012 with fully turbulent boundary-layer (Sections 3.2 - 3.4), lift error evaluation loses importance at low angles of attack since a slight difference with respect to theoretical zero lift causes huge errors. Again, from approximately  $AoA = 2$  degrees it is possible to note a very low error for the entire abscissa range up to stall condition. There, CFD results are satisfying since they are able to detect maximum lift. After stall conditions with higher AoA, the error drastically increases since the lift coefficient found by the CFD solver is much higher with respect to Abbott experimental data.

About drag, CFD results provide coefficients with very low error (maximum absolute value of 7.77%) with respect to Abbott experimental data. The comparison with Ladson has not real validity since those experimental results were performed at higher freestream turbulence intensity, as already cited; however, close to stall conditions the experimental results performed by Abbott and Ladson both seem comparable.

## Observations about SA-BC within SU2

During the various preliminary tests performed using BC transition model, some inconsistencies arised looking at CFD results.

Application of SA-BC inside SU2 is based on Cakmakcioglu et al. [40] [41] [58] and on NASA summary [42].

As already presented in Section 2.3.3, BC transition model is based on correlations with flow features, simplifying and reducing computational costs. This model requires freestream turbulence intensity ( $Tu_\infty$ ) value; for SU2 it is provided by input variable `FREESTREAM_TURBULENCEINTENSITY`.

Figure 2.7 reports the correlation between freestream turbulence intensity and experimental transition onset critical momentum thickness Reynolds number.

The inconsistency of SU2 version 7 is about the respective references of freestream turbulence intensity used inside BC model and for freestream turbulent kinetic energy. Considering the example adopted for the validation test about NACA 0012 with free transition boundary-layer of present Section, the input value was:

`FREESTREAM_TURBULENCEINTENSITY = 0.15`

Going inside SU2 code, inconsistencies emerge since:

- the BC model reads the percentage value in a range of [0, 100]; then, the input 0.15 corresponds to 0.15%, consistent with the BC papers [40] [41] [58].
- the freestream turbulent kinetic energy reads the percentage value in a range of [0, 1]; consequently, in this case 0.15 corresponds to 15%.

The latter reference causes an output extremely high for freestream turbulent kinetic energy with respect to the order expected: in the cited test case about NACA 0012, freestream turbulent kinetic energy per unit mass output was  $87.9372 \text{ m}^2/\text{s}^2$ , while it is expected a value of approximately  $0.00879372 \text{ m}^2/\text{s}^2$ .

Anyway, since Spalart-Allmaras turbulence model does not directly use freestream turbulence intensity and turbulent kinetic energy, this inconsistent and wrong value does not affect the CFD results. On the other hand BC model is based on freestream turbulence intensity and it uses the cited reference range of [0, 100]; then, input should be consisted with respect to BC reference.

A reporting has been sent to SU2 developers, who assured that next versions of the software will fix the cited inconsistency since freestream turbulence intensity is used by SST and other transition models under development/implementation.

### 3.5.2 NACA 23012

Another geometry is proposed for validation: NACA 23012 with transitional boundary-layer. The transition model for boundary-layer is again BC. Since this airfoil is asymmetric, both positive and negative angles of attack need to be analysed. The results are compared again with experimental data by Abbott and Von Doenhoff [1]. The flow conditions are the same used for NACA 0012 (Reynolds number:  $6 \cdot 10^6$ , Mach: 0.15, free-stream temperature: 288.15 K, free-stream turbulence intensity: 0.15%).

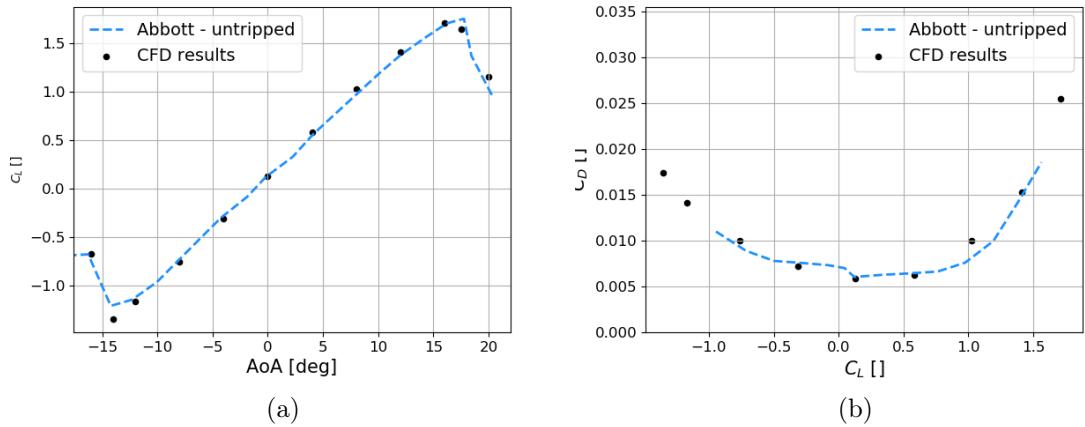


Figure 3.29: Comparison between CFD results and Abbott experimental data about NACA 23012 with free transitional boundary-layer.

CFD simulations provided acceptable and very precise aerodynamic coefficients for NACA 23012, except for lift coefficient approximately equal to 1.0 ( $\text{AoA} \simeq 8\text{deg}$ ) where CFD drag coefficient seems quite overestimated.

### 3.5.3 NLR 7301

Up to here the analysis were proposed about single element geometries, providing satisfying results.

This section presents a comparison between CFD results obtained with the present thesis' tool and experimental results for the NLR 7301. The comparison treats pressure coefficient distribution on the two elements of the configuration. Experimental data were recovered from Van der Berg and Gooden report [45] since not available in digital version on web.

Figures 3.30 and 3.31 present NLR 7301 geometry and grid.

Flow conditions and configuration details are reported in the following list:

- Mach number: 0.185;
- Reynolds number:  $2.51 \cdot 10^6$ ;
- Absolute angle of attack w.r.t. main element: + 6 deg;
- Absolute angle of attack w.r.t. flap: + 20 deg.
- Since the experimental data of Reference [45] has been obtained with untripped boundary-layer, CFD simulations have adopted the SA-BC turbulence-transition model.

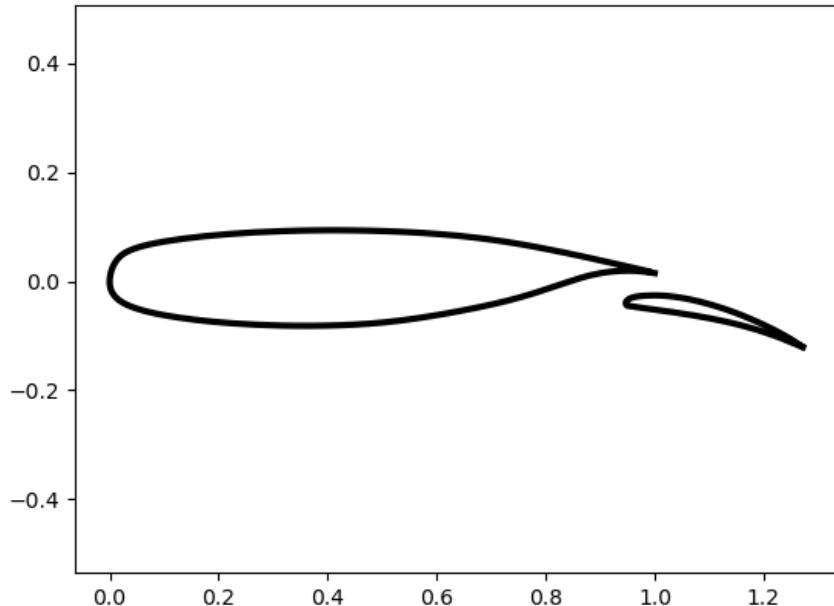


Figure 3.30: Sketch of two-element airfoil NLR 7301 geometry.

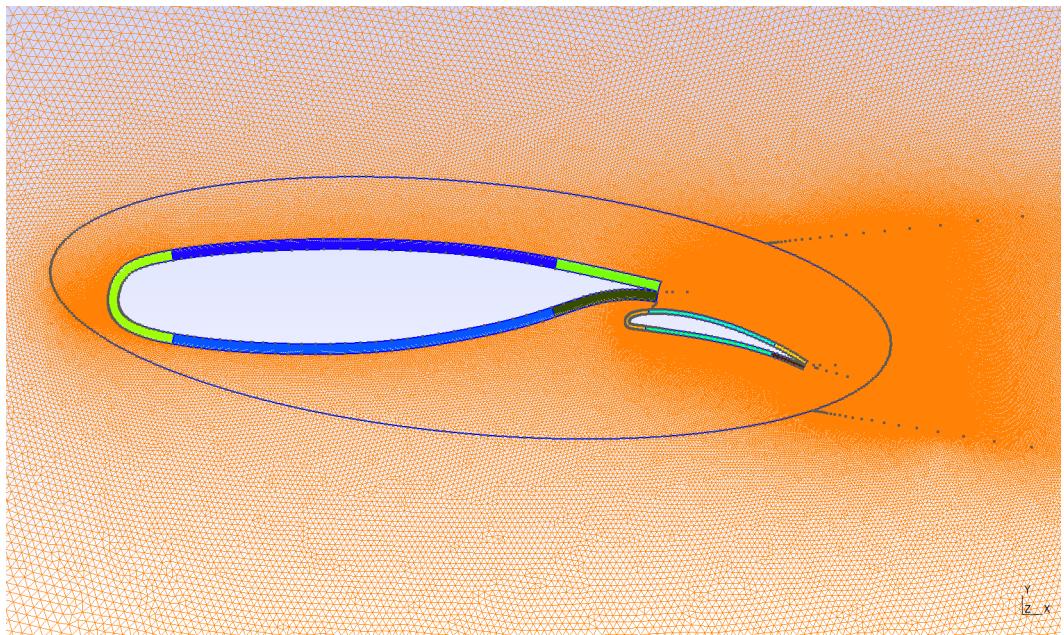


Figure 3.31: Zoom on NLR 7301 grid close to wall surface.

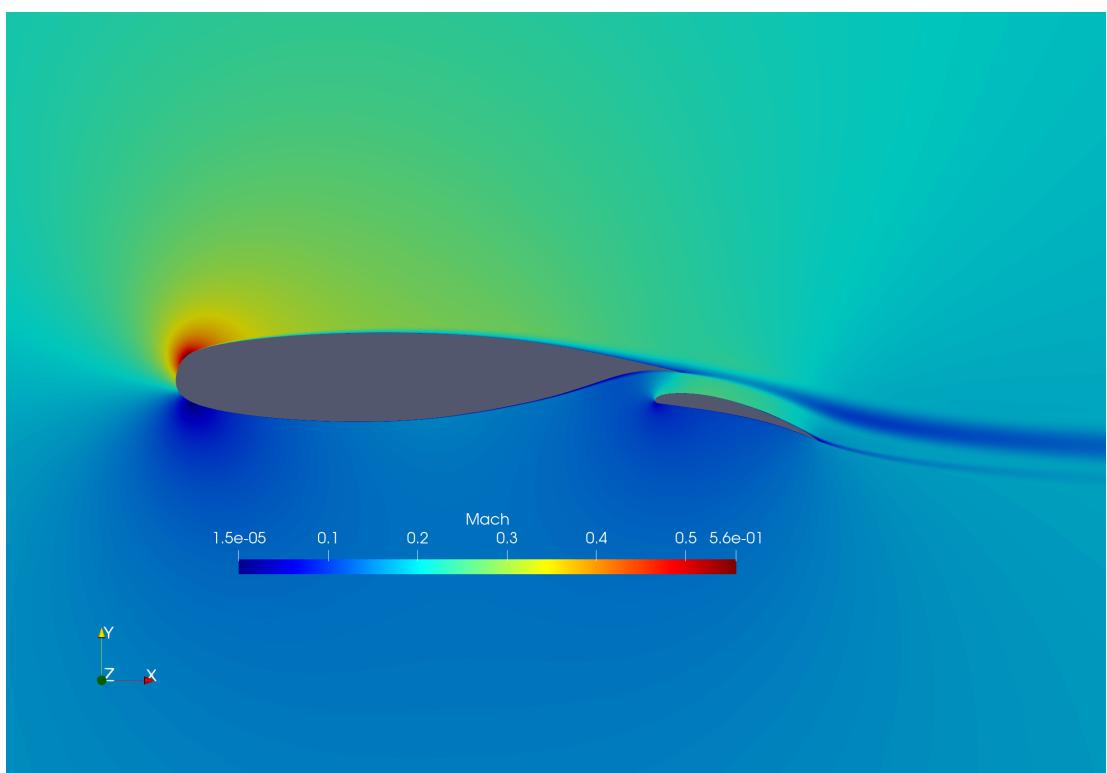
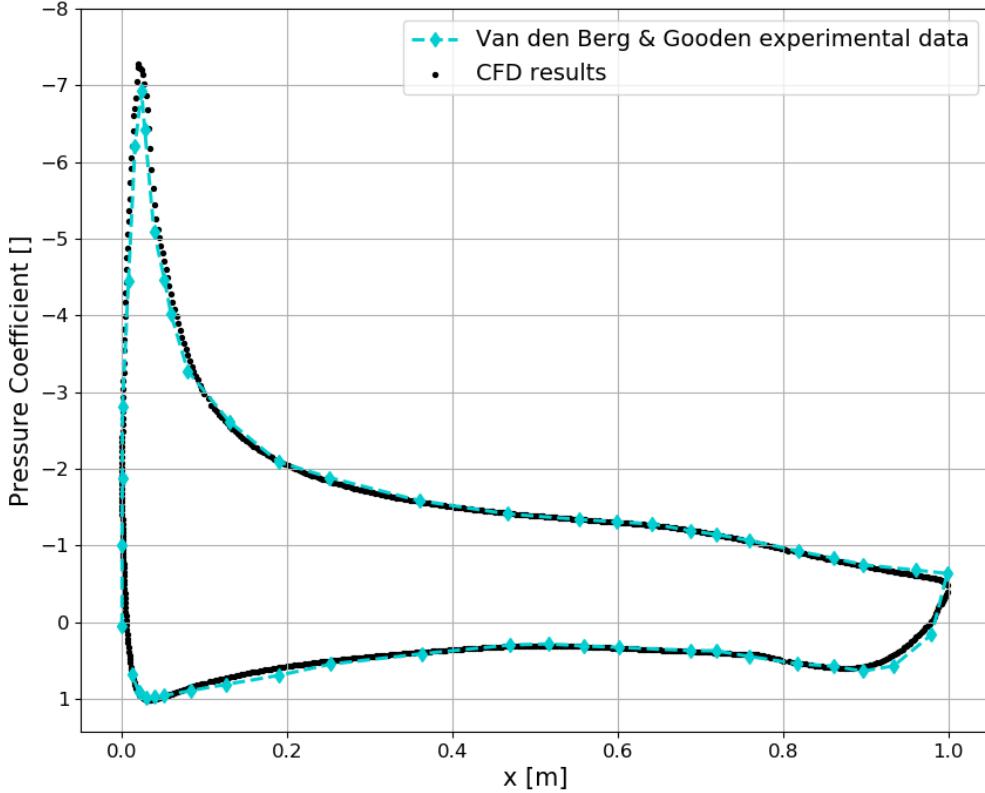
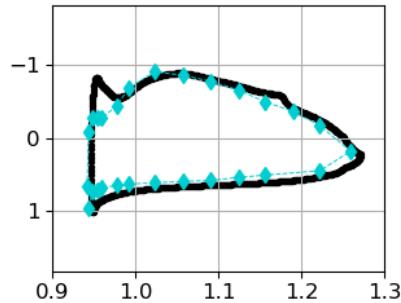


Figure 3.32: Mach visualization of NLR 7301 at absolute AoA 6 deg.



(a) Main element



(b) Flap

Figure 3.33: Comparison between CFD results and experimental data [45] about pressure coefficient distribution.

The NLR 7301 pressure coefficient comparison's plots between CFD and experimental data [45] display a good matching, especially on the main element of the airfoil. Some mismatching about pressure coefficient present on the flap; in particular:

- the pressure coefficient on the flap LE (upper surface) presents a depression peak not reported in experimental data.
- the upper surface at  $x = 1.175$ , where smoothness of  $C_P$  distribution is affected by a small depression peak.

About the last observation made, experimental data report the transition point on flap at approximately (1.16) x-coordinate [45]. In literature the relationship between pressure coefficient jump and transition point [59] [60] is widely described, both experimentally and using computational means. About the main element, the transition point has been also detected, but can be observed visually by looking at x-component of friction coefficient (Figure 3.34) zoomed on interesting coordinates. In this case the experimental boundary-layer transition position is inside the range [63%, 71%] of the chord [45], while by CFD means it has been detected at approximately 78% since here a significant gradient is revealed [32].

To conclude, although an amount of uncertainty derives from recovering experimental data and geometry positioning without direct availability of coordinates and values, CFD results performed by the procedure on NLR 7301 about pressure coefficient can be considered overall acceptable.

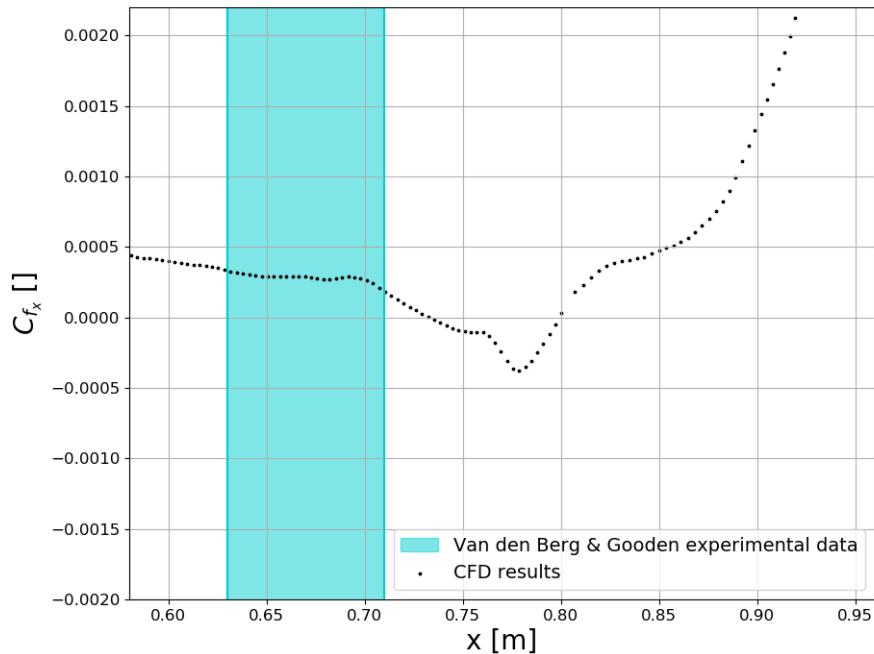


Figure 3.34: Computed CFD x-component of friction coefficient on main element lower surface of NLR 7301, compared with transition experimental location [45] (light-blue coloured).



# Chapter 4

## Optimisation Module

### 4.1 Optimisation methods: introduction

Optimisation of multi-airfoils systems becomes crucial since performance directly depends on the configuration and shape chosen.

From a general point of view, an optimisation problem consists of a collective process to find one or various sets of conditions to achieve the best (or optimal) result for a given situation [61].

The present section will briefly describe optimisation problems from a general point of view, and then focusing on aerodynamic problems.

The nomenclature related to a generic optimisation problem is the following:

- *objective function* (or *cost function* or *fitness* or *response*), that is the aim of the analysis to be minimized or maximized;
- *design variables* are the optimisation problem unknowns; they could be either discrete or continuous, depending on the problem;
- *constraints* consist of equalities and/or inequalities about the design variables;
- *noise* is the uncertainty associated with the response;
- *design space* (or *domain*) is the n-dimensional space defined by a range of continuous design variables [62];
- *design point* is a point located in design space and defined by a unique set of design variables [22]

The first classification about optimisation procedures regards the objective function, which can be defined by a single equation or multiple ones.

From a purely mathematical point of view, many procedures are available for optimisation problems to seek and find the global optimum.

The optimisation procedure can be classified in *single-objective* and *multi-objective*, depending on the presence of single or multiple objective functions.

Many other classifications exist, depending on many different characteristics. In the following some classification examples are reported:

- heuristic or mathematical-based;
- deterministic or stochastic;

- evolutionary, direct or surrogate research methods;
- gradient-free or gradient-based;
- with the feature to be local optimum research methods or global optimum ones;
- hybrid methods.

As easily guessed, optimisation problems' subject is wide and strictly dependent on the domain for which the optimum is searched. A first general presentation of the basic concepts and interests about optimisation algorithms procedures will be described below, while the details about optimisation procedures focused on multi-element configurations will be presented in the following section.

D. Landman [7] presents and analyses various optimisation methods and procedures from a general and mathematical point of view, but then applied to multi-element configurations and aerodynamic optimisation problems. As a consequence, it becomes a nice guide about what can be extrapolated from the large mathematical subject of optimisation methods and applied to multi-airfoil configuration optimisations.

In this dissertation, the process is sought based on experimental results, since historically the first way to find optimal solutions for multi-element configurations (but also for many other different sectors) was by using practical experiments, while simulations like CFD became relevant only recently. As a consequence, these optimisations are called *experimental optimisations* and can be executed using *on-line* or *off-line* approaches. Literature offers various details about experimental optimisation with the respective advantages and disadvantages of each method or approach from a purely mathematical and theoretical point of view, but also applied to practical experimental problems. D.C. Montgomery [14] offers a wide view of experimental optimisation procedures from a theoretical a mathematical point of view. Scott and Haftka [22] indicate the cost of each experiment as the strongest deciding factor about the method's selection, along with derivatives' approximations and noise. Several works respectively provide extensive analysis about experimental optimisations' details both applied to different sectors and multi-element experimental optimisations [63] [64] [65] [66].

When each experiment about optimisation is expensive, the user has to adopt methods that possibly converge in relatively few iterations. Another possibility is modifying existing optimisation methods trying to reduce costs. Some examples of cost-reducing methods are the following:

- *Evolutionary Operation*;
- *Taguchi methods*;
- *D-Optimal criterion*, otherwise called parametric studies or interactive methods [7] [22];
- *Box-Hunter methods*.

Most of the cases studied with the cited optimisation methods and procedures have an important limitation: higher complexity of the domain could reduce the possibilities to reach a so-called *global optimum*.

Depending on the starting point location in the domain, various methods risk failing and converging directly in a *local optimum*. In fact, no optimisation algorithm can distinguish the local optimum from the global one, unless the objective function under

analysis is convex [7]. Almost every practical case in scientific optimisation problems does not provide a convex objective function, which is also mostly unknown.

For this reason, over the years particular models have been developed to overcome this limitation. Among them, the most known techniques are:

- Genetic Algorithm optimisation (GA) is a meta-heuristic optimisation algorithm based on Darwin's theory of "fittest candidate survival" belonged to a population [62]. In particular, the population evolve by mostly inheriting (so-called *crossover* step) design variables or *genes* of the optimal configuration found at each iteration. These genes are variated by a random amount, then the process is repeated up to the convergence criterion. Many variations are available in the literature for various method's features: selection of each step optimum found, whether or not saving previous iterations' optimum (like creating an "elite" of best optimum individuals), how crossover should be executed etc.
- Particle Swarm Optimisation (PSO) is a meta-heuristic optimization algorithm, originally proposed by Kennedy, Eberhart and Shi [67] (1995) [68] (1998). It is based again on the evolution of a population in the design space. In this case, individuals (or *particles*) evolves directly translating in the design space. Velocity and direction of translation are dictated by the optimal design position found from the procedure beginning. Further details will be presented in Section 4.2.

A large population along with an amount of random variation of individual's design variables drastically reduces the probability to converge to a local optimum [69]. Moreover, the methods are widely known and are easy to program nowadays.

Anyway, the most important typical disadvantage of optimisation algorithms based on evolutionary populations is the high number of required objective function evaluations [7].

Consequently, simpler and *inexpensive* methods could become useful to face experimental optimisations' problems. Their potential lies precisely in the low cost required for each experiment, guaranteeing a practical method for experimental optimisations.

Among them, the so-called *gradient methods* are the ensemble that contains the most known and practical ones. All gradient methods are based on the research of specific design variables which minimize (or maximize) objective function's gradient, which provides a practical way to determine the response's rate of change [70]. They become practical if the number of design variables and the level of noise are not too high.

Various versions of gradient-based methods were developed since the second half of the last century. Since this ensemble of methods is particularly wide, it is suggested to refer again to Montgomery's work [14] for further details.

## 4.2 Optimisation Algorithms

In this section, the theory about optimisation algorithms implemented inside the present thesis' tool is briefly presented.

### 4.2.1 Steepest Ascent (or descent) method

Also known as the *saddle-point method*, the Steepest Ascent (or Descent) method is a development of Laplace's method applied to the asymptotic estimate of integrals of analytic functions. Today the method is classified as belonged to the category of gradient methods [61] [70] [71] since gradient evaluation at a given point is the main matter of the method.

Starting from a given point (design variable set,  $\underline{x}_0$ ), the Steepest Ascent method depends on the evaluation of the response gradient which determines the moving direction based on successive gradient magnitudes and scaling factors (obtained by experience) [7]. The displacement in design space proceeds until optimum response does not significantly vary, meaning an optimum has been reached; alternatively, the procedure stops when a maximum number of iterations has been reached.

As for other gradient methods, the main relation is the following:

$$\underline{x}_{i+1} = \underline{x}_i + \alpha \underline{s} \quad (4.1)$$

where  $(\underline{x}_{i+1}, \underline{x}_i)$  are the *centroids* of iterations  $i+1, i$  respectively. The method depends on the values of  $\alpha$  (step length) and  $\underline{s}$  (unit vector in the direction of steepest) so that the objective function  $F = F(\underline{x})$  is minimized or optimised. As a consequence, the unit vector is found to be:

$$\underline{s} = -\frac{\nabla F}{|\nabla F|} \quad (4.2)$$

since the gradient of  $F$  (vector  $\nabla F$ ) in the direction of the greatest rate of change of  $F$ . On the other hand,  $\alpha$  can be directly determined by  $F$  between the following iterations [62].

Like other versions or methods which belong to the gradient-method category, the main advantages (with respect to other methods' typologies) are:

- straightforward mathematical concepts;
- convergence in fewer iterations;
- lower computational costs;
- high probability to find local optima.

On the other hand, the major disadvantage can be extremely relevant: gradient methods find difficulties in detecting global optima. Looking at the Steepest Ascent, the starting point and step size can cause issues from this point of view. Their tuning is important to avoid stuck into the local optimum.

Another aspect that influences the reaching of global optima, that is the definition of the objective function. If its trend in design space is highly irregular presenting various local optimums, probabilities to stuck the method in local optimum increase.

Looking at the aerodynamic optimisation problem about multi-element airfoils configurations, the objective function is often unknown [7] and the presence of various local

optimums is probable. An example is about the configuration GA(W)-1, whose performances and response surfaces about maximum lift have been analysed experimentally by Wentz and Seetharam [9]; GA(W)-1 will be the subject of configuration optimisation in the following Sections.

D. Landman [7] deeply analysed the experimental optimisation problem about multi-element airfoils by using the Steepest Ascent, referring to previous studies of several authors [70] [72] [73] [74]. A bi-dimensional visualization of the method behaviour is shown in Figure 4.1. Moreover, Landman compared the Steepest Ascent with few other versions or methods: Steepest Ascent seems to perform slightly better, but each method guarantees sufficient results highlighting again the potentiality of optimisation algorithms for multi-element airfoils' configurations (in this case, experimentally).

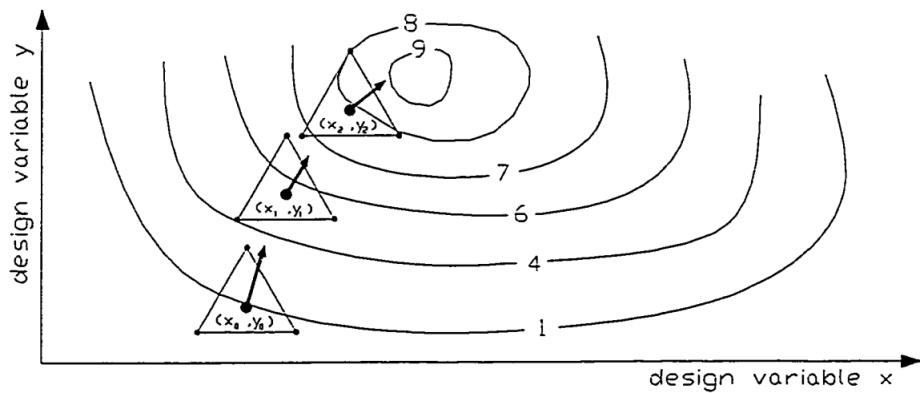


Figure 4.1: Example of bi-dimensional optimisation problem using Steepest Ascent [7].

Since analytically unknown in aerodynamic problems, the objective function has been proposed with a linear model:

$$C_L = a + bx + cy \quad (4.3)$$

for an hypothetical problem in two variables ( $x, y$ ), like the typical problem of slot optimisation for multi-element airfoils. Since the three coefficients ( $a, b, c$ ) are unknown, the methods need three objective functions evaluations (lift coefficients) to complete the linear relation. These evaluations are executed on three candidates obtained from centroid perturbation; for the example of a bi-dimensional problem, they could be the vertexes of an equilateral triangle with centroid equals to last optimum found.

At each step, the new centroid coordinates are determined as follows:

$$x_{i+1} = x_i + S_i \frac{b_i}{\sqrt{b_i^2 + c_i^2}} \quad (4.4)$$

$$y_{i+1} = x_i + S_i \frac{c_i}{\sqrt{b_i^2 + c_i^2}} \quad (4.5)$$

while the scaling parameter  $S_i$  is:

$$S_i = \left[ \frac{\sqrt{b_i^2 + c_i^2}}{\sqrt{b_{i-1}^2 + c_{i-1}^2}} \right]^M \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (4.6)$$

The user has to provide in input a starting point (set of design variables  $\underline{x}_0$ ), starting step ( $S_0$ ) and element size ( $h$ ), which strictly depend on the problem under analysis;

Landman indicated a 0.5 as a proper value for the exponent  $M$ .

Based on the works of Jacoby et al. [72] and Semones et al. [73], an important concept was underlined by Landman about optimal scaling factor, that is better choosing a smaller than optimal step size (and hence scaling factor) since optimal does not perform significantly better concerning convergence speed.

The Steepest Ascent algorithm version proposed by Landman has been adopted in the present thesis. An extension has been executed on the custom design variables for the user inside the tool: in particular, the user can consider respective angles of attack of each element as design variables, besides (x,y) slot's coordinates. Since the number of design variables possibly increase from (2) to a higher amount, the generation of a hyper-pyramid in design space has been implemented whenever the user considers a high amount of design variables.

With respect to Landman's algorithm, the other main differences implemented inside the tool concerns mainly the *noise* term, which has been neglected and not considered inside the optimisation procedure. However, note that the experimental results performed by Landman did not observe particular influences on fitness about this term.

The last difference with respect to Landman's algorithm regards tolerance and stall management, trying to avoid or escape from possible local minima after centroid stalls at a position in design space. In particular, whenever the hyper-pyramid candidates do not provide better results in terms of output fitness, the centroid does not move inside the design space for the subsequent iteration. Then, the algorithm will try an increasing reference size of hyper-pyramid (doubling, in the case of the present thesis) and subsequently a continuous halving. Each time centroid does not move since hyper-pyramid candidates do not provide better fitnesses, a stall iteration will be counted. The code will exit when an upper stall iteration bound has been reached.

#### 4.2.2 Particle Swarm Optimisation (PSO)

Particle Swarm Optimization (PSO) is a meta-heuristic optimization algorithm, originally proposed by Kennedy and Eberhart [67] (1995) along with Shi [68] (1998). Extensive surveys about the applications were made by Poli [75] [76].

Being a meta-heuristic algorithm, PSO involves no guarantee to find the global optimal solution. Anyway, it usually increases the probabilities to find global optimum with respect to other methods (like gradient-based methods) thanks to the high number of candidates and objective function evaluations along with partial random characteristics in their design variables' definition.

In fact, a PSO considers a population (*swarm*,  $S$ ) of candidate solutions (*particles*,  $i$  to denote the singular one) which respectively and iteratively moves into a search-space looking for minimization of a specific cost function ( $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ).

Each particle is described by its position ( $\underline{x}_i \in \mathbb{R}^n$ ) and velocity ( $\underline{v}_i \in \mathbb{R}^n$ ) in the search-space.

At each iteration, each particle moves in the search-space using a defined vector (with its direction and magnitude). The displacement of each particle is respectively influenced by:

- velocity vector;
- its best-known position w.r.t. fitness value;
- shared best fitness value found by the entire population.

Note that the relationship between new iterations, better positions in search-space and previous iterations' solutions is balanced and regulated by some parameters, defined by the user and strictly dependent on the problem under analysis. For example, considering a simple version of PSO [77], the velocity and position of a particle can be updated as follows:

$$\underline{v}_{i+1} = \omega \underline{v}_i + \phi_p r_p (\underline{p}_i - \underline{x}_i) + \phi_g r_g (\underline{g} - \underline{x}_i) \quad (4.7)$$

considering  $\underline{p}_i$  as the best-known position of the  $i$ -th particle, while  $\underline{g}$  is the best-known position of the entire swarm. A correct definition of these parameters (in this example,  $\omega$ ,  $\phi_p$ ,  $\phi_g$ ) along with random terms ( $r_p$  and  $r_g$ ), the algorithm can drastically reduce the possibility for the particle to stuck into local optimum. However, also these parameters usually have an optimal definition which strictly depends on the number of design variables and problem domain.

This thesis will adopt the simple version of Particle Swarm Optimisation proposed by M. Clerc [77], but it is not intended to analyse the correct or optimal selection of cited parameters; as a consequence, these parameters will not be further investigated. For the same reasons, there will be no analysis of proper swarm size or tolerance definition. For the optimisation presented in the following sections with PSO, the parameters were fixed to:

- $\omega = 0.5$ ;
- $\phi_p = 0.5$ ;
- $\phi_g = 0.5$ .

Various and deep studies available in the literature have been performed in past and recent years, related also to different sectors of applications (examples: references [78] [79] [80]).

Lastly, about convergence criterion, typical options are the following [81]:

- reaching limit number of iterations;
- step size of the optimum under a defined tolerance;
- objective function variation under a defined tolerance.

The latter two convergence criterion can be extended to a higher amount of iterations to be satisfied (so-called *max stall iterations*).

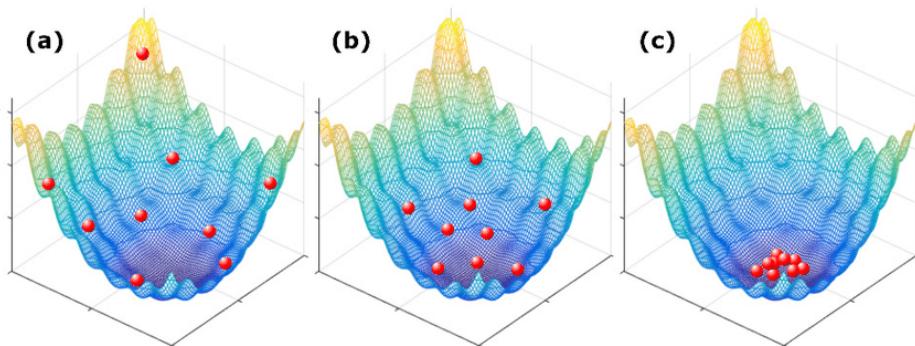


Figure 4.2: PSO example: starting population points to the global minimum (a-b-c).

## 4.3 Preliminary Optimisation studies

In the present thesis, three methods have been implemented to perform preliminary studies about design space under analysis about multi-airfoil configurations.

These preliminary studies try to define proper bounds and starting point based on less expensive and low-fidelity methods, like Euler CFD and Hess-Smith panel method. Consequently, spurious and uninteresting design variables would be excluded, improving convergence speed of the subsequent optimisation procedure.

### 4.3.1 Experimental Designs for Fitting Response Surfaces

At the current stage of the tool, candidates of the preliminary population are selected using *Latin-Hypercube Sampling* and *Box-Behnken design*, which belongs to "Experimental Designs for Fitting Response Surfaces".

These methods already found wide applications in experimental optimisation procedures. These designs have the main advantage to guarantee a proper analysis of response surfaces, specifically when they are unknown or performing a high number of tests (experimental or computational) can be too expensive. Details about this topic can be found in D.C. Montgomery work [14].

When selecting one of these designs, some features are desired:

- providing sufficient and reasonable distribution of data points;
- guaranteeing investigation of model adequacy;
- providing an internal estimation of error;
- ensuring simplicity of model parameters' calculation;
- limiting number of runs;
- providing reasonable robustness of the method;
- guaranteeing precision about coefficients' estimation;

and many others.

These features can be sometimes conflicting with each other: for this reason, the designs built throughout the years have tried to enhance the advantages. Further details about performances can be found inside works of Khuri and Cornell [82] (1996) or Myers et al. [83] (2016).

#### Latin-Hypercube Sampling (LHS)

One of the first space-filling designs proposed was the so-called *Latin-Hypercube sampling* by McKay et al. [14] [84]. Further deepening has been executed by Iman et al. [85] [86] (1981).

A Latin-Hypercube is a statistical method used for the generation of a sample of specific parameters from a multi-dimensional distribution. It consists of an  $n \times k$  matrix, given by  $n$  runs for  $k$  factors. The columns of the matrix respectively consists of a random permutation of the levels from 1 to  $n$ .

In two dimensions, a Latin-Hypercube becomes a square for which there is only one sample in each row and each column. The global definition of a Latin-Hypercube is the

generalisation of this concept to an arbitrary number of dimensions.

The advantage of this sampling scheme is that more samples are not required for more dimensions. Another advantage is that random samples can be taken one at a time, remembering which samples were taken so far.

In Python, the Latin-Hypercube can be accessed by importing **pyDOE** [87]. For its creation, it can be used the following syntax:

```
lhs( $n$ , [samples, criterion, iterations])
```

where:

- $n$ : an integer value that defines the number of factors;
- *samples*: an integer which designates the number of sample points for each factor;
- *criterion*: a string which defines how to sample the points. There are numerous options accepted for this string:
  - "center" or "c": points' centre within the intervals' sampling;
  - "maximin" or "m": maximize the minimum distance between points, but place the point in a randomized location within its interval;
  - "centermaximin" or "cm": like "maximin", but centred within the relative intervals;
  - "correlation" or "corr": minimize the maximum correlation coefficient.

### Box-Behnken design

The *Box-Behnken design* was proposed in 1960. For fitting response surfaces, this approach considers some three-levels designs combining  $2^k$  factorials with incomplete block designs, usually guaranteeing efficiency in terms of the number of required runs. Anyway, having three-levels means Box-Behnken design can be applied to problems with no less than three design variables [14].

Figure 4.3 shows a visualization of a Box-Behnken design for three factors.

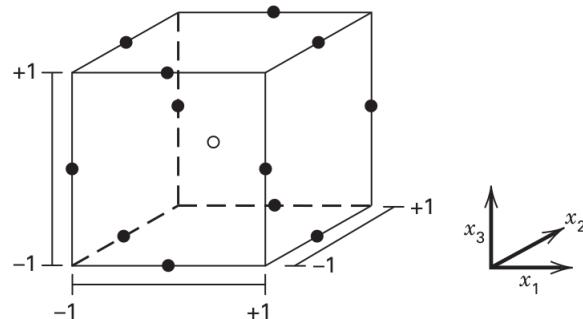


Figure 4.3: Visualization of a Box-Behnken design (for three factors) [14].

As the figure reports, this design places the factors on a sphere of radius  $\sqrt{2}$ , resulting equally spaced (usually coded as  $-1, 0, 1$ ) and having poor design space corners' coverage; anyway, it is still considered one of the most reliable approaches between response

surface methodologies.

With the following syntax it is possible to generate a Box-Behnken design in Python [87]:

`bbdesign(n, centre)`

where:

- *n*: integer value which defines the number of factors; in this case, it must be major or equal to (3);
- *centre*: number of points to include.

#### 4.3.2 Hess-Smith Panel Method

Panel methods adopt the solution of Laplace's equation placing sources, vortices and sinks on airfoil's body, using also the exact flow tangency boundary conditions without the approximations used in thin airfoil theory.

These methods consider the following assumptions about flow properties:

- inviscid;
- irrotational;
- steady;
- incompressible (not mandatory).

A continuous distribution of sources/sinks and vortices lead to integral equations which cannot be treated analytically; discretization is therefore necessary. For this reason, the airfoil's surface is considered as a series of segments (or *panels*).

One of the simplest versions of panel methods is the one proposed by Hess and Smith (1966) [88].

HS method considers sources and vortices on geometry surface, such that:

$$\phi = \phi_\infty + \phi_S + \phi_V \quad (4.8)$$

where  $\phi$  is the total potential function, while the other three terms are the potential of freestream, sources and vortexes distributions, respectively.

In particular:

$$\phi_S = \int \frac{q(s)}{2\pi} \ln(r) ds \quad (4.9)$$

$$\phi_V = \int \frac{\gamma(s)}{2\pi} \theta ds \quad (4.10)$$

where  $q(s)$ ,  $\gamma(s)$  are the respective strengths along  $s$  (arc-length coordinate). For the Hess-Smith panel method, vortex strength  $\gamma(s)$  is constant, while source strength  $q(s)$  varies on the airfoil.

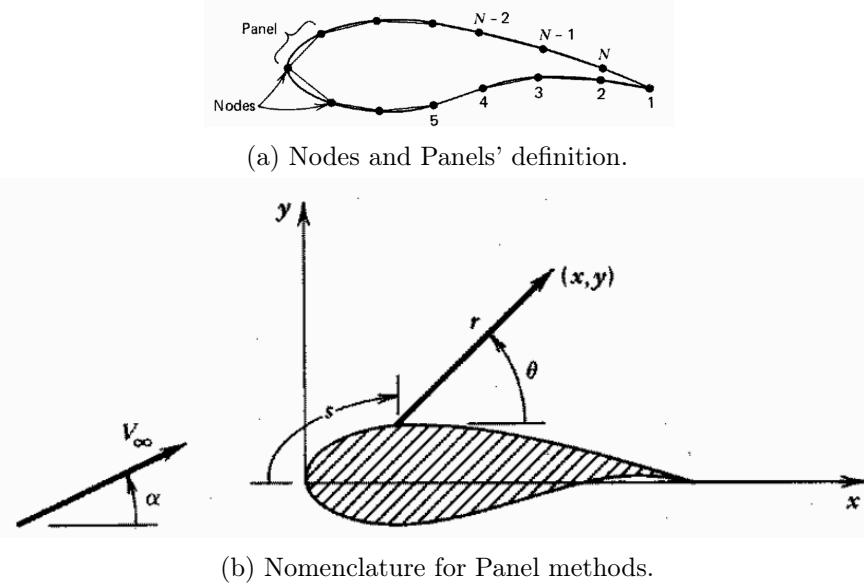


Figure 4.4: Schematic details about Panel methods [88].

The discretised total potential function can be derived from 4.8:

$$\phi = U_\infty(x \cos\alpha + y \sin\alpha) + \sum_{j=1}^N \int_{panel_j} \left[ \frac{q(s)}{2\pi} \ln(r) - \frac{\gamma}{2\pi} \theta \right] \quad (4.11)$$

where  $N$  is the total amount of panels.

Thanks to the assumptions on sources and vortexes respective strengths, the total amount of variables is  $N + 1$ . To accomplish problem resolution,  $N$  equations are obtained from flow tangency boundary conditions for each panel (in the case of Hess-Smith, evaluated at the midpoint), plus one equation imposing Kutta condition on TE. The resolution will then require the inversion of an  $(N + 1) \times (N + 1)$  matrix.

Although simpler with respect to CFD, panel methods are still widely used for preliminary analysis since the required computational time of each run is very low.

For this reason, the present thesis tool allows preliminary studies using the Hess-Smith Panel method. If requested, Python code opens Matlab and execute the panel method's calculus for each candidate (generated using Latin Hypercube Design or Box-Behnken, described previously).

J.L. Hess [89] [90] Junaidin et al. [91] provide further information about panel methods.

### Valarezo-Chin criterion

Inside a preliminary study using the Hess-Smith panel method in Matlab, whenever the objective function is the maximum lift, the user can adopt the Valarezo-Chin criterion [8] as preliminary research.

Analysing experimental data at maximum lift, Valarezo and Chin empirically noted that:

$$|C_{P_{peak}} - C_{P_{TE}}| \simeq 13 \div 14 \quad (4.12)$$

where  $C_{P_{peak}}$  is the maximum pressure coefficient, while  $C_{P_{TE}}$  pressure coefficient at trailing edge. The relation is valid for each element of the configuration.

A dependence from Mach and Reynolds numbers was observed by the authors: in particular, the limit values of (13 ÷ 14) decrease when high flow velocities or lower Reynolds numbers are taken into account.

Since the observations made by Valarezo and Chin were referred to experimental data, if the criterion is applied in Hess-Smith panel methods the viscous terms could cause differences with respect to actual values. In particular, the maximum pressure peaks detected by a panel method could be much higher than the limit values proposed by Valarezo and Chin of (13 ÷ 14).

For this reason, an upper bound has been consequently implemented inside the HS preliminary study with Valarezo-Chin criterion, excluding any candidate which presents a pressure coefficient difference (at each configuration's element) higher than 13.5. All the candidates which overcome this value at any configuration's element will be automatically excluded.

## 4.4 Tool application

"Optimisation Module" refers to the configuration and IGP shape optimisations.

The lower and upper bounds of each design variable must be provided by the user, respecting the natural geometrical features of the configuration. For example, in configuration optimisations the slots' bounds must not allow intersecting of elements' geometries or structured regions; another example is about IGP parameters in shape optimisations, which have to respect bounds provided by the IGP method's authors [21]. The block diagram in Figure 4.5 summarises the procedure of multi-element optimisations applied by the present thesis' tool. Note that any time the green block "Grid & CFD Module" appears, the procedure presented in Chapter 3 is executed for grid generation and CFD simulations.

The optimisation methods are Steepest Ascent (or Descent) and Particle Swarm Optimisation, which features can be customized by the user in input (total amount and which design variables, max iterations' number, tolerances etc.).

The respective algorithms and theory principles are the ones already described previously in Section 4.2; the user has to provide the following inputs about the respective optimisation algorithms:

- Steepest Ascent (or Descent):
  - Starting step size ( $S0$ ), which is the dimension of centroid design variables' variation at the first iteration;
  - Maximum step size ( $SMAX$ ), which is the maximum dimension of centroid design variables' variation at following iterations;
  - Reference size ( $REF\_SIZE$ ), which defines candidates' design variables based on centroid ones;
  - $min\_iter$ : minimum iterations for convergence;
  - $max\_iter$ : maximum iterations for convergence;
  - $max\_stall\_iterations$ : maximum stall iterations, for which best fitness is allowed to not vary, then the procedure ends;
  - $tolerance$ , under which stall iterations counting holds;

- Particle Swarm Optimisation:
  - *swarmsize*, which indicates the total amount of candidates (or population size) evaluated at each iteration;
  - *max\_iter*: maximum iterations for convergence;
  - *min\_step*: minimum step size allowable to optimum solution to proceed PSO, otherwise convergence applied;
  - *min\_func*: minimum fitness variation allowable for the optimisation to go on, otherwise stop for convergence.

Because proposed optimisation methods could require a significant computational time due to large bounds or inappropriate starting point, the user has also the possibility to apply a preliminary study of the optimisation procedure, customizing the methodology between a panel method (Hess-Smith method) or CFD analysis (either inviscid or viscous) on an ensemble of candidates.

Moreover, the user can define the method for design variables' generation for each candidate, their features and the size of the ensemble too. At the current stage, the tool provides Box-Behnken and Latin-Hypercube designs for the design variables' assignment to each preliminary candidate, described in the following Section 4.3.

Consequently, the role of an efficient preliminary study is to favour and speed up convergence to the optimal solution, may also adopting low-costs computational simulations (like panel method or Euler CFD).

An investigation about their efficacy is reported in Section 4.5.1.

Note that the Hess-Smith panel method is performed using Matlab, while a preliminary study with inviscid or viscous CFD is performed using the usual procedure (Gmsh along with SU2).

In the latter case, note that two scripts related to configuration files for SU2 can be used: one is related to preliminary studies, one for the actual optimisation calculus.

As a consequence, the user can consider "lighter" requirements to perform preliminary studies and saving computational time, like less restrictive RANS convergence criterion.

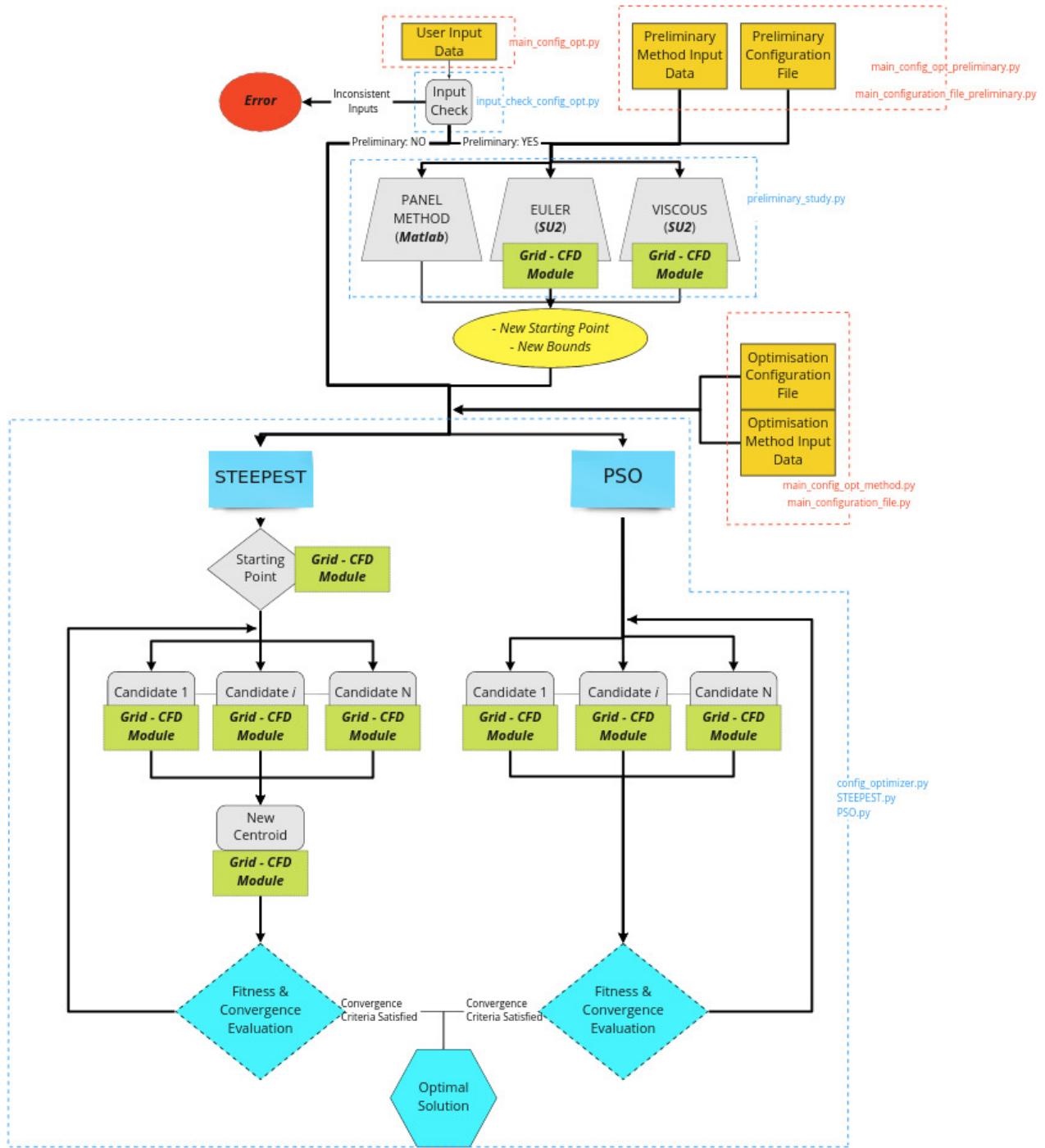


Figure 4.5: Optimisation module.

## 4.5 Results of Optimisation Module

Several and different results of the optimisation procedures are presented in this section.

Results of configuration optimisation of the GA(W)-1 airfoil are reported. Several test cases were performed applying various combinations of methods, preliminary studies and optimisation features for comparisons. The attention will be focused on performances (both accuracy and convergence speed), underlining also possible developments and improving through preliminary studies using Euler CFD or Hess-Smith panel method. An important premise has to be cited: since the geometry is web-provided, there is uncertainty about comparison with respect to experimental data [9].

Beyond geometrical definition, the uncertainties regard whether flap chord is 29% or 30% chord: although similar, the two results proposed experimentally [9] provides different aerodynamic performances.

In the present work, the 29% chord flap was chosen as a reference for comparison.

Relative elements' chords and their positioning are fundamental whenever a slot optimisation and related comparison with experimental data are taken into account, as the following optimisation results will show.

The shape optimisation of a double-element IGP airfoil intended for automotive applications will be presented to highlight the potentialities of a fully automated optimisation procedure for multi-airfoil configurations based on CFD also in different domain cases.

### 4.5.1 Slot Optimisation of GA(W)-1

This paragraph is intended to present results about slot optimisation for a GA(W)-1 configuration for lift maximisation, adopting methods of Steepest Ascent and Particle Swarm Optimisation, already discussed in previous Sections.

The design variables under analysis are ( $x, y$ ) coordinates of flap leading edge distance from the main element trailing edge; these two design variables entirely describe the slot between the main element and flap. The objective function is the maximum lift. From general literature but also specifically about GA(W)-1 [9] experimental results, the optimal solution is located close to the main element trailing edge, but with a certain margin: exceeding in slot reduction, the lift coefficient typically decreases. The statement is valid essentially for any relative AoA of the flap.

At the current status of the tool, bounds can be imposed only using upper and lower limits, defined by practical observations on generated grids.

Bounds must avoid overlapping of the two elements' geometries and their respective structured regions. Further developments can investigate ways to admits closer analysis between the elements and geometrical relations to avoid overlapping without simply relying on upper/lower bounds values.

Bounds and starting points will be presented each time a new test is proposed since they are a critical input for optimisation. Anyway, each Test proposed considers a good starting point and bounds, focusing the optimisation attention on the interesting region. Figures 4.6 - 4.7 report an example of optimisation starting configuration geometry and grid of GA(W)-1 adopted in Test A (presented later). Location of ( $x, y$ ) axes slot reference is also reported.

As already introduced, several tests have been performed; for some, preliminary studies have been executed using the Hess-Smith panel method or Euler simulations, trying to

delineate whether they become helpful to increase procedure speed or not. Tables 4.1 - 4.2 - 4.3 report fixed parameters or conditions for all tests regarding GA(W)-1 slot optimisation; subsequently, details and peculiarities of each test will be presented, followed by the discussion and analysis of results.

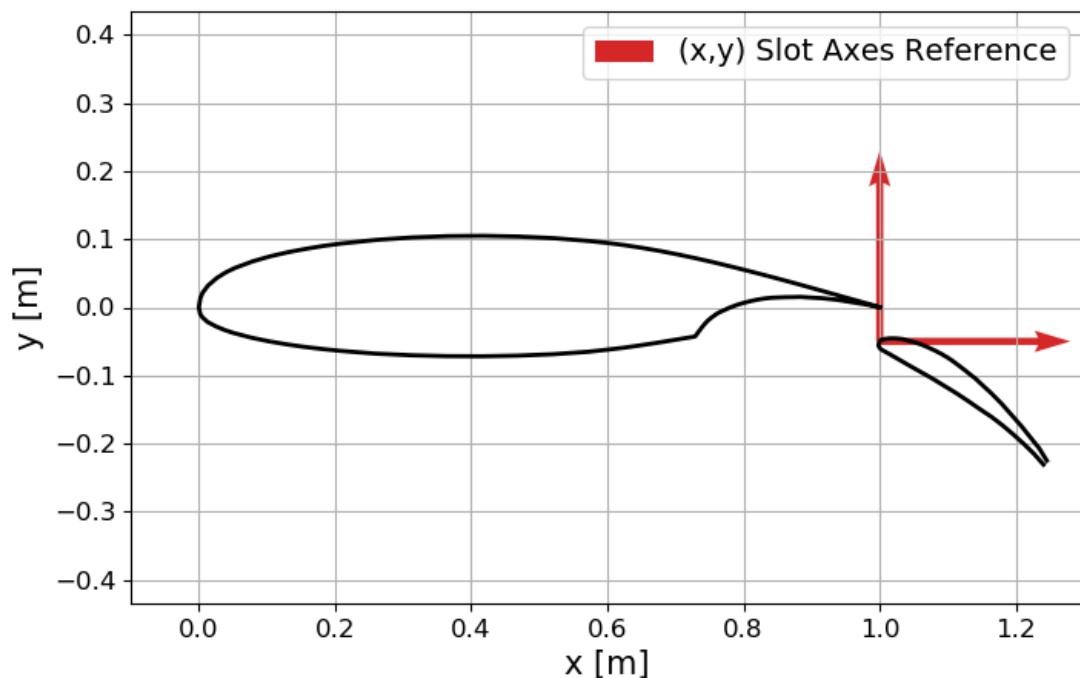


Figure 4.6: GA(W)-1 geometry at an example slot starting point  $[x, y] = [0.000, -0.050]$ .

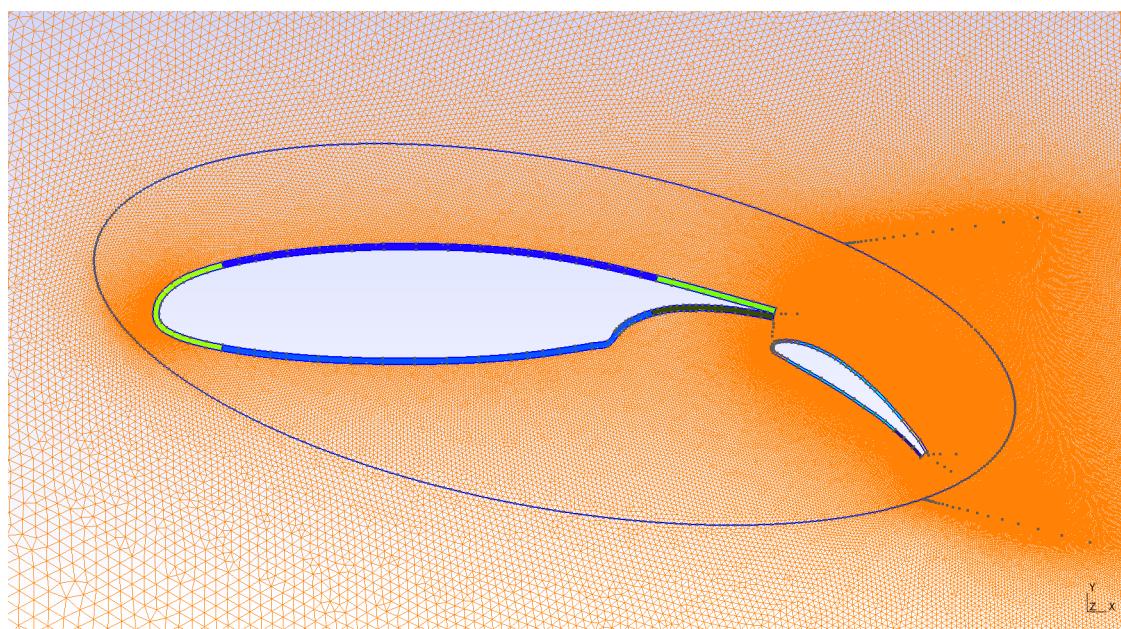


Figure 4.7: GA(W)-1 grid at an example slot starting point  $[x, y] = [0.000, -0.050]$ .

Flow condition	Value
Mach number [-]	0.21
Reynolds number [-]	$2.2 \cdot 10^6$
Temperature [K]	288.15
Dynamic Viscosity [kg/m · s]	$1.853 \cdot 10^{-5}$
Boundary-Layer	Fully Turbulent

Table 4.1: Flow and geometry conditions used for GA(W)-1 CFD optimisation, based on Reference [45].

CFD Configuration Feature	Value/Method
CFL	500
Adaptive CFL	NO
Numerical Scheme	Roe + MUSCL
Method for spatial gradients	Weighted Least Squares
Time discretization	Euler Implicit
Convective method	Scalar Upwind

Table 4.2: Computational configuration features used for GA(W)-1 CFD optimisation.

Grid Parameter	Value
<i>nodes</i>	0.95
<i>ellipse_dimension</i>	0.85
<i>ellipse_refining</i>	0.9
<i>y_plus</i>	[0.08, 0.12]
<i>thick</i>	[0.5, 0.5]
<i>progr</i>	[1.15, 1.15]
<i>wall_refining</i>	[1.2, 1.2]
<i>wake_length</i>	50
<i>wake_progr</i>	1.3
<i>semicircle_dimension</i>	18
<i>semicircle_elem_factor</i>	250

Table 4.3: Grid features used for GA(W)-1 CFD optimisation.

The primary analysis will focus on whether CFD optimisation is feasible or not, its advantages and disadvantages, the issues of each methodology and first possible corrections' to reach satisfying results. Furthermore, results will be compared with experimental data as far as possible based on Wentz and Seetharam experimental report [9] about GA(W)-1. In particular, the main core of the tests will consider an absolute AoA for the two elements of [7.7, 37.7] deg, which is an already extreme condition for GA(W)-1 analysing experimental behaviour reported in the cited Reference:

- flap relative AoA has been firstly fixed at 30 deg since slightly higher values (35-40 deg) do not provide further lift performances improvements.

- absolute angle of attack was fixed at 7.7 deg since stall conditions are approximately located to 10 deg, with slight differences depending on flap relative AoA).

Reference [9] does not provide directly the position of optimal slots when GA(W)-1 configuration has flap at relative AoA of 30 deg, but only maximum lift reachable: consequently, Tests from A to P will be compared with experimental data about lift coefficient only.

Test V will increase relative flap AoA at 35 deg with absolute values of [3.3, 38.3] deg to make a further comparison with Wentz and Seetharam experimental data also about optimal slot.

The following Table summarises the main optimisation features of tests performed about GA(W)-1.

Test ID	AoA	Starting Point	Preliminary study	Opt. Method
<b>A</b>	[7.7, 37.7]	[0.0000, -0.0500]	/	Steepest
<b>B</b>	[7.7, 37.7]	[0.0100, -0.0300]	/	Steepest
<b>C</b>	[7.7, 37.7]	[-0.0400, -0.0400]	/	Steepest
<b>D</b>	[7.7, 37.7]	[-0.0000, -0.0500]	HS	Steepest
<b>E</b>	[7.7, 37.7]	[-0.0000, -0.0250]	HS	Steepest
<b>F</b>	[7.7, 37.7]	[-0.0000, -0.0500]	Euler	Steepest
<b>M</b>	[7.7, 37.7]	[0.0000, -0.0500]	/	PSO
<b>N</b>	[7.7, 37.7]	[0.0000, -0.0500]	HS	PSO
<b>O</b>	[7.7, 37.7]	[0.0000, -0.0500]	Euler	PSO
<b>P</b>	[7.7, 37.7]	[0.0000, -0.0250]	/	PSO
<b>U</b>	[3.3, 38.3]	[0.0000, -0.0300]	Euler	PSO
<b>V</b>	[3.3, 38.3]	[0.0000, -0.0300]	HS	Steepest

Table 4.4: Performed GA(W)-1 optimisation tests summary.

### Tests A, B, C: Steepest Ascent w/o Preliminary Study

The adopted features about Steepest Ascent optimisation algorithm used for Tests A, B and C are reported in Table 4.5.

Steepest Parameter	Value
Starting step size ( $S_0$ )	0.0075
Max. step size ( $S_{MAX}$ )	0.0050
Reference size ( $REF\_SIZE$ )	0.0075
Min. iterations ( $min\_iter$ )	5
Max. iterations ( $max\_iter$ )	20
Max. stall iterations	5
Tolerance	0.03
Bounds	x-coord.
	y-coord.

Table 4.5: Steepest method features used for GA(W)-1 slot CFD optimisation.

Tests A, B, C do not involve a preliminary study; consequently, design space bounds and starting point are the ones provided in input, and they will not change during the optimisation. Method's parameters (like reference or step size) were chosen based on meaningful dimensions: for Tests A, B and C, these values will remain fixed, while the only difference between the three cited tests is about respective starting point.

As already cited in Section 4.2 about Steepest Ascent features, gradient methods risk to fail and stuck into local optimum since based on gradient evaluation: proper starting point and method's parameters can drastically improve convergence performances.

To verify and investigate the sensitivity, further tests with different methods' features, new starting points and various geometries or flow conditions can be performed.

Figure 4.8 reports the respective computational history of Tests A, B, C inside design space, while Figure 4.10 plots the respective objective function (maximum lift) histories. Figure 4.9 reports the respective location of each candidate considered by Test C.

The starting configuration is printed in grey; note: whenever an iteration does not involve a displacement of centroid and centroids are overlapped as a consequence, only later iteration centroid marker is displayed.

Table 4.6 reports starting and final design variables and fitnesses of each Test.

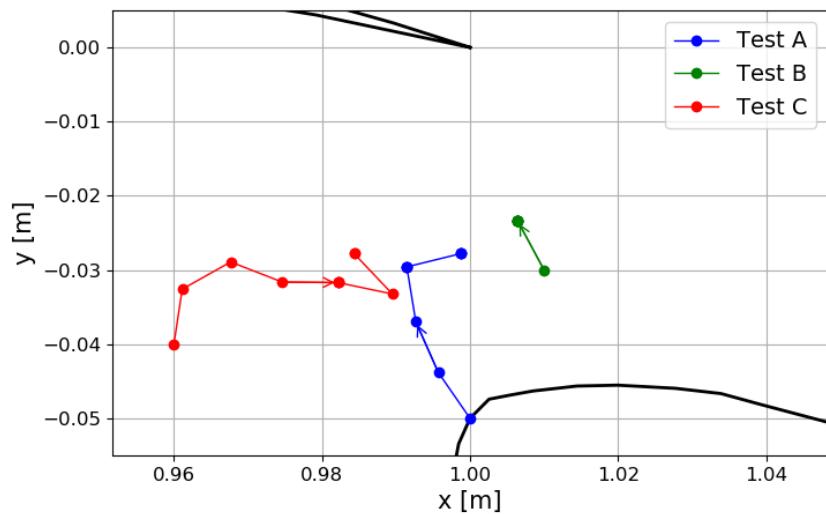


Figure 4.8: Respective histories of optimisations Tests A, B, C inside design space.

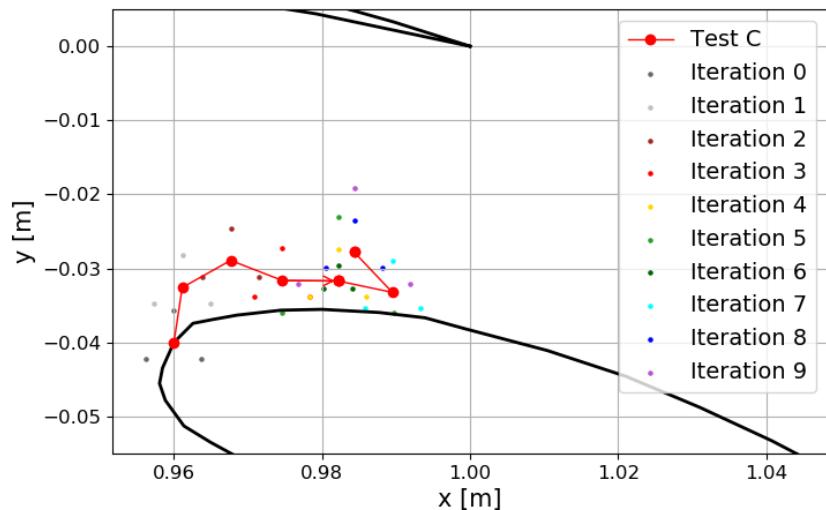


Figure 4.9: Candidates considered by Test C slot optimisation.

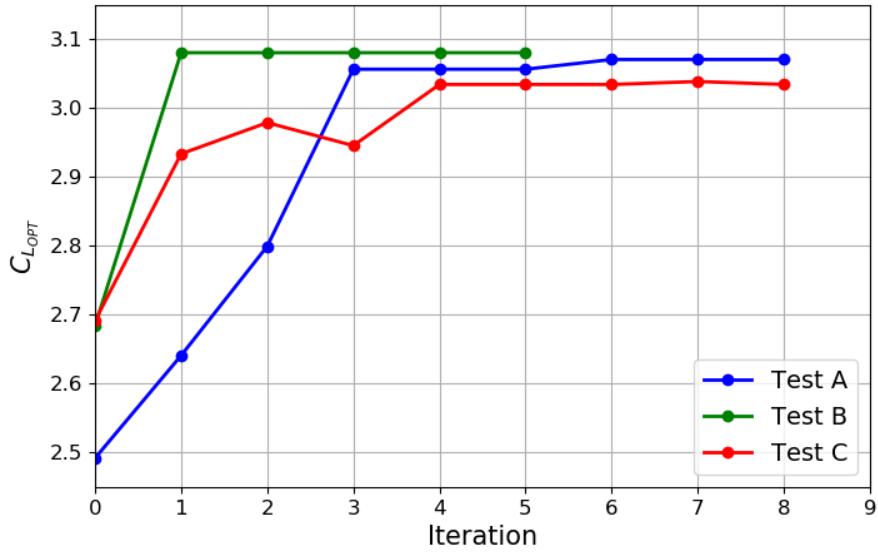


Figure 4.10: Respective histories of optimisations Tests A, B, C with respect to fitness (maximum lift).

	Test A	Test B	Test C
<b>Starting Point</b>	[0.0000, -0.0500]	[0.0100, -0.0300]	[-0.0400, -0.0400]
<b>Starting Fitness</b>	2.4920	2.6841	2.6914
<b>Optimal Slot</b>	[0.0013, -0.0278]	[0.0064, -0.0234]	[-0.0157, -0.0278]
<b>Optimal Fitness</b>	3.0710	3.0809	3.0348
<b>% Error w.r.t. [9]</b>	-4.33	-4.02	-5.46

Table 4.6: Starting and Optimal points in design space and fitnesses for Tests A, B, C, respectively.

The percentage error is calculated with respect to Wentz and Seetharam experimental result  $C_L = 3.21$  maximum lift, evaluated with the same configuration.

The optimal solutions converge to the same region, in line with the one found experimentally by Wentz and Seetharam for similar flap AoA (with a gap of approximately 2%  $\div$  3% and overlap close to 0%). Furthermore, the maximum lift coefficient found is at approximately 4%  $\div$  5% error w.r.t. experimental data.

Consequently, the CFD optimisation procedure applied provide good results with low errors, similarly to the previous test cases executed on NACA 0012, NACA 23012 and NLR 7301.

The slight differences are however explainable recalling again that geometry was web-provided and not so particularly refined in its shape; moreover, the parameters for the optimisations are the default ones, while the grid parameters were based on sensitivity analysis proposed in Section 3.3.

Another interesting result is about convergence speed of Steepest Ascent in the three different Tests: looking at Figure 4.10, the number of max stall iterations imposed 5 can be considered excessive.

For example, after the fifth iteration (number 4 in the plot) Test C does not present significant fitness improvements, remaining in the same position for the next iterations. For the other two tests, the optimisation could have been stopped at even lower iteration for the same reasons.

Consequently, if the optimisation stops at the 5th iteration and if the user can perform parallel computing at each iteration (in these cases with two design variables, using 4 cores: 3 for the hyper-pyramid, 1 for the centroid), the duration of the entire optimisation would have been approximately 8-9 hours, which is a reasonable computational requested time without requesting HPC.

Another important aspect about convergence speed regards the proper definition of starting point, which could become extremely crucial: comparing Test A and C, the total amount of required iterations before objective function stalling goes from 2 to 5. Consequently, a definition of starting point based on low-expensive preliminary studies could save a relevant amount of time, favouring convergence of optimisation algorithms. For this reason, the following paragraph presents further tests which firstly considered a Hess-Smith panel method or Euler solutions to define a proper starting point and, possibly, further reducing the bounds to focus the domain on the main region of interest.

### Tests D, E, F: Steepest Ascent with Preliminary Study

Tests A, B, C were exemplifications of starting point importance in Steepest Ascent optimisation with respect to total required time and for convergence.

In particular, Tests A and C has taken more iterations to reach convergence due to a not optimal starting point, while Test B immediately converged to an optimal slot region. Moreover, a proper starting point can also avoid stuck into the local optimum as can may happen for typical gradient-based methods.

Taking as reference Test A starting point ([0.0000, -0.0500]) and bounds ([-0.1000, 0.1000], [-0.1000, -0.0190]), further optimisations were performed considering preliminary studies. Firstly, new Tests D, E, F consider a population of candidates inside the design space obtained with Latin Hypercube design (LHS), which considers:

- starting point as centre;
- standard deviation w.r.t. centre of [0.030, 0.025];

for (x,y) coordinates of each candidate's slot, for all Tests D, E, F. Then the procedure calculates respective fitness of each candidate based on less expensive simulations with respect to viscous ones: Hess-Smith panel method or Euler CFD.

As already presented at the beginning of the present chapter, the user has also the possibility to perform these preliminary studies using directly viscous CFD, guaranteeing higher precision but at the cost of low computational speed. For this reason, only low-fidelity methodologies (like Euler CFD and Hess-Smith panel method) are investigated in the following test cases, trying to meet sufficient accuracy and speed.

Note that whenever the starting point is already a good choice but a preliminary study is imposed, the low-fidelity methods adopted (inviscid CFD or panel method) may change the starting point worsening the actual fitness. For this reason, performing a preliminary study is suggested for those cases the user does not have practical knowledge of or experience of a possible optimal position.

Once the preliminary study ends, the algorithm will select the candidate with the best fitness (optimal slot) as the new starting point; furthermore, the bounds will be reduced

to the candidates' design variables values with at least a particular fitness (named as "Bounds margin").

The following Table reports the features of Tests D, E, F respective preliminary study, while Steepest Ascent optimisation performed considers the same parameters of Test A.

	<b>Test D</b>	<b>Test E</b>	<b>Test F</b>
<b>Method</b>	HS	HS	Euler
<b>Samples</b>	100	100	16
<i>criterion</i>	<i>Valarezo-Chin</i>	<i>Max. Lift</i>	<i>Max. Lift</i>
<b>Bounds margin</b>	15	4.5	3.7

Table 4.7: Preliminary Studies features for Tests D, E, F.

Row "*criterion*" defines the fitness evaluation criterion adopted for preliminary study: Tests E, F evaluate directly the maximum lift, while Test D adopts Valarezo-Chin criterion already presented in Section 4.3.

Reduction of the bounds is based on cited criterion and "Bounds margin" value: all the candidates with fitnesses higher than this value will be included inside the new bounds. Recall that whenever Valarezo-Chin criterion is adopted (as done for Test D), this limit value will be based on pressure coefficient difference using the following relation:

$$\sum_{i=1}^2 (|C_{P_{peak}} - C_{P_{TE}}|)_i \geq 15 \quad (4.13)$$

where index  $i$  indicates the  $i$ -th element's pressure difference.

As already presented in Section 4.3, for elements close to stall (and consequently to maximum lift) Valarezo-Chin criterion defines a maximum pressure coefficient difference of  $13 \div 14$ : whenever panel method is considered, all the candidates with higher values has to be ignored, since in viscous problems these cases probably involve less performances about maximum lift due to viscous effects (like separation). Then, the code automatically ignores these candidates, analysing and consequently selecting all the ones with  $(|C_{P_{peak}} - C_{P_{TE}}|)_i \leq 13.5$  for each  $i$ -th element, as the Valarezo-Chin criterion dictates.

On the other hand, Tests E, F consider higher "Bounds margin" values for maximum lift: either Euler CFD or Hess-Smith panel method are expected to provide higher values of lift coefficients with respect to viscous problem, as the following results show. Test E differs also about starting point and bounds: in particular, upper bound of  $y$ -coordinate is imposed as less restrictive to guarantee closer flap positions to main element. This modification was defined to demonstrate that smaller slots do not involve higher lift performances, as experimental data show.

To guarantee smaller slots' evaluations, grid structured region was reduced to  $[0.2, 0.4]$  for parameter *thick*.

Results of Tests D, E, F are reported in Table 4.8, including respective preliminary studies' ones.

	<b>Test D</b>	<b>Test E</b>	<b>Test F</b>
<b>Start Point</b>	[0.0000, -0.0500]	[0.0000, -0.0250]	[0.0000, -0.0500]
<i>Start Bounds</i> ( $x,y$ )	[-0.1000, 0.1000] [-0.1000, -0.0190]	[-0.0500, 0.0500] [-0.0500, -0.0105]	[-0.1000, 0.1000] [-0.1000, -0.0190]
Prel. Opt. Fitness	22.2194 (***)	4.7270 (**)	3.8663 (*)
Prel. Opt. Point	[-0.0433, -0.0422]	[0.0060, -0.0190]	[-0.0064, -0.0280]
<i>New Bounds</i> ( $x,y$ )	[-0.0451, 0.0619], [-0.0870, -0.0190]	[-0.0148, 0.0574] [-0.0442, -0.0190]	[-0.0500, 0.0370] [-0.0833, -0.0190]
<b>Iterations</b>	9	9	6
<b>Optimal Slot</b>	[-0.0082, -0.0319]	[0.0119, -0.0177]	[-0.0064, -0.0314]
<b>Optimal Fitness</b>	3.0614	3.0682	3.0570
<b>% Error w.r.t.</b> [9]	-4.63	-4.42	-4.77

Table 4.8: Preliminary Studies results for Tests D, E, F.

(\*): preliminary study result computed with Euler CFD.

(\*\*): preliminary study result computed with Hess-Smith panel method; objective function is maximum lift.

(\*\*\*): preliminary study result computed with Hess-Smith panel method; objective function is maximum pressure coefficient according to Valarezo-Chin criterion.

For the optimisation procedure both Tests E and F have considered a reduced max stall iterations to (3), based on previously results; the minimum amount of iterations before exiting was maintained anyway at (5), guaranteeing an inspection of design space before exiting. Furthermore, Test E considered different parameters for Steepest Ascent, trying to test accuracy:

- $S0 = 0.005$ ;
- $SMAX = 0.0015$ ;
- $REF\_SIZE = 0.004$ .

which means closer candidates one each other in design space and lower step size at each iteration.

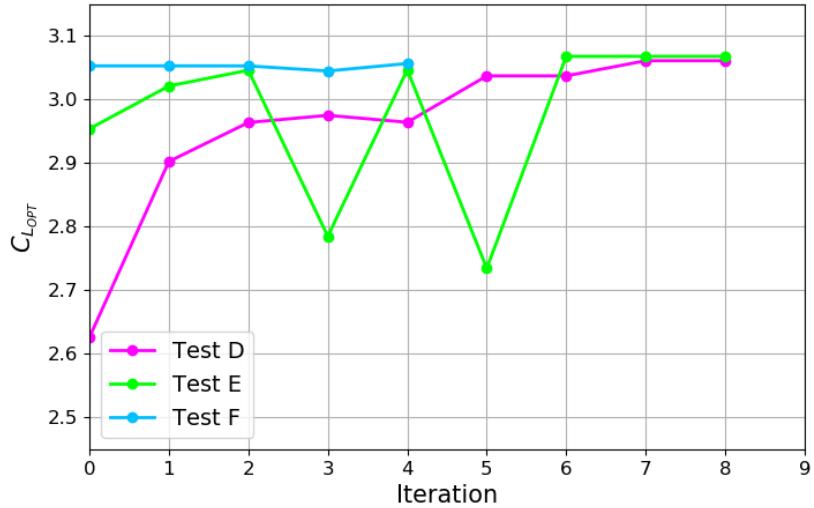


Figure 4.11: Tests E, F history of optimisation inside design space.

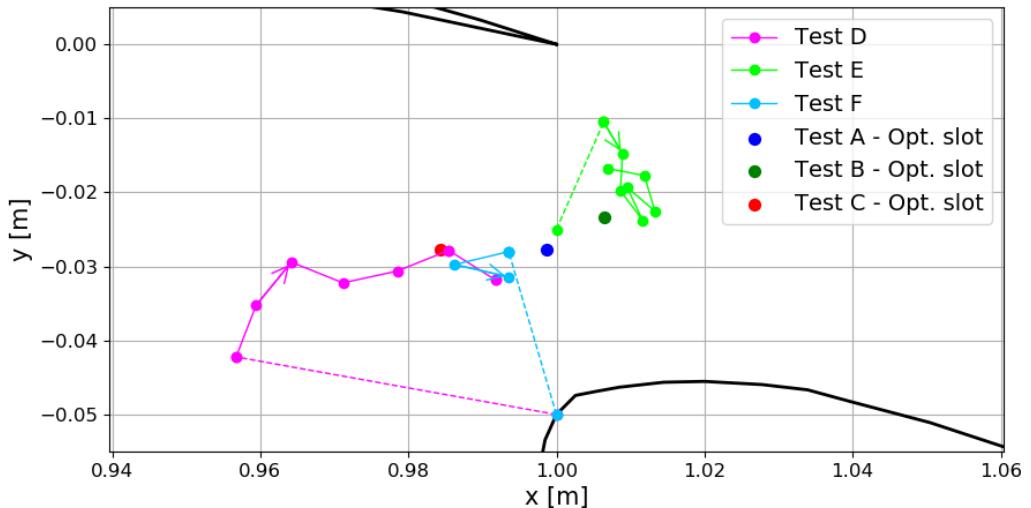


Figure 4.12: Tests E, F history of optimisations with respect to fitness (maximum lift).

The reported results present overall successful optimisation for all the three Tests D, E, F involving preliminary low-fidelities studies.

The first observation is about the Hess-Smith panel method preliminary study output, adopted for Test E: although the starting point was centred in an already optimal region, the panel method is not that precise due to its hypothesis and slightly moves the starting point closer to the main element.

However, all Tests D, E, F generate good starting points for the following optimisations, guaranteeing fast convergence performances: Hess-Smith panel method and Euler CFD can be stated as efficient for determining a proper starting point. After the preliminary study, the optimal solution moves from the main element trailing edge, converging to the same area of the other tests, as expected.

Apparently, only Test E has found issues along the way to reach convergence, requiring a higher total amount of iterations: in particular, the optimal solution appears oscillates shown in Figure 4.12. This observation underlines how tight and small is the optimal region with at least (3.0) as lift.

The design space trend is in line with what measured experimentally by Wentz and Seetharam [9] about GA(W)-1 airfoil slot: around the optimal region with approximately  $C_L$  equals  $3.0 \div 3.1$ , fitness drastically decreases with a large gradient, especially for negative overlaps. Although the step size was reduced, the variations in optimal fitness are relevant. The second observation is about the Steepest Ascent method: since based on the gradient of close candidates, the centroid moves based on their fitnesses values, but the risks to make an excessively big step is possible.

However, the algorithm immediately responds going back to the optimal region in the design space at the following iterations and eventually stabilizing there.

Test F involves a preliminary study using Euler CFD on a population of 16 candidates: the new starting point is already in the region of interest, guaranteeing Steepest Ascent iterations to drastically reduce in number. Again, the convergence criterion required at least a minimum of five iterations before exiting, but the optimal solution found remains in the same location inside the design space and the fitness has high values at the first iteration already.

Consequently, it appears that an Euler CFD preliminary study can be more effective with respect to the Hess-Smith panel method; on the other hand, it requires more computational time to perform RANS inviscid CFD of each preliminary candidate.

To conclude the comparisons of Tests with the Steepest Ascent method, results show a procedure that tends to an optimal region in few iterations, guaranteeing low computational times.

Steepest Ascent has the advantage of requiring only (4) CFD simulations at each iteration, guaranteeing applicability also without HPC. Regarding accuracy, the method still requires proper tuning to maximize its efficiency and reducing the risks of oscillations or sticking into local minima.

The results are however promising and guarantee low fitness' errors with respect to experimental data, as Tests A, B, C, D E, F show.

### Tests M, N, O, P: Particle Swarm Optimisation

Tests M, N, O and P were performed using Particle Swarm Optimisation (PSO). Theory about the method was described inside Section 4.2. The Table 4.9 reports PSO parameters adopted.

PSO Parameter	Value
<i>swarmsize</i>	32
<i>max_iter</i>	25
<i>min_step</i>	$5 \cdot 10^{-5}$
<i>min_func</i>	$10^{-3}$

Table 4.9: PSO features used for GA(W)-1 slot CFD optimisation.

Test M was the first one performed and it did not involve a preliminary study. Consequently, design variables' chosen bounds were the widest already considered for Tests A, B, C, which were [-0.100, 0.100], [-0.100, -0.190] respectively for (x,y) slot coordinates.

The starting point was the main source of issues for Steepest Ascent optimisation method from a computational point of view instead, PSO may face troubles whenever large bounds are considered. In particular, Test M was aborted at iteration (8) since the optimum was unreliable: lift coefficient of 7.2102 at position [0.0221, -0.0779].

Due to unreliability of Test M results, it is not worth reporting the complete optimisation procedure. As already presented in Section 1.1, the proximity between two elements guarantees reduction of effective flap AoA thanks to downwash of upward element, retarding stall condition [3] [8]. Consequently, whenever the flap is too far from main element, no benefits are expected by multi-element configuration and flap will stall at high AoA as a typical single element configuration. To face these conditions, bounds become crucial.

Consequently, CFD simulations tend to oscillate: if not excluded, these wrong results could interfere with the optimisation analysis, providing spurious data. To avoid these issues, a criterion has been imposed to RANS history: whenever the last N iterations overcome a limit value of standard deviation (implying RANS history large oscillations), the candidate will be automatically excluded from optimisation by imposing null lift, for example.

For the following tests like N and O, the adopted standard deviation lower limit considered has been of (0.025) calculated among the last (200) RANS history iterations. In the case of PSO, the probability of encountering these particular affecting cases increases since several candidates are considered inside design space. Consequently, the definition of proper bounds for the chosen design space becomes more crucial: preliminary studies using panel methods or Euler simulations could extremely favour the optimisation procedure.

Based on these notes, Tests N, O consider preliminary studies to reduce the bounds and a proper starting point, which features are reported in the Table 4.10.

	Test N	Test O
<b>Method</b>	HS	Euler
<b>Samples</b>	100	32
<i>criterion</i>	<i>Max. Lift</i>	<i>Max. Lift</i>
<b>Bounds margin</b>	4.5	3.7

Table 4.10: Preliminary Studies features for Tests N, O.

Test P is an additional test case performed without preliminary studies and directly performing PSO with reduced bounds and different starting point. Its feature are reported in the following list.

- starting point: [0, -0.025];
- bounds for (x,y), respectively: [-0.050, 0.050], [-0.050, -0.0105];
- structured grid region thickness' (parameter *thick*): [0.2, 0.4].

The reason why Test P considered different grid, design variables' bounds and starting point was related to observation made on Tests N and O optimal slot found (reported in Table 4.11): in these test cases, the optimal solution was found closer to upper y-coordinate bound. To verify that no optimal solution can be found closer to main element trailing edge, Test P was performed considering a reduced structured region to favour evaluation of candidates who involve particularly small slots' configurations. Moreover, Test P already excludes configurations involving flap far from main element considering restricted starting bounds.

Figures 4.14 - 4.15 report history in design space and optimal fitness found at each iteration, while Figure 4.13 reports Test P swarms' particle positions and optimal slots for some iterations to exemplify convergence to optimum slot found.

	Test N	Test O	Test P
<b>Start Point</b>	[0.0000, -0.0500]	[0.0000, -0.0500]	[0.0000, -0.0250]
<i>Start Bounds</i> (x,y)	[-0.1000, 0.1000] [-0.1000, -0.0190]	[-0.1000, 0.1000] [-0.1000, -0.0190]	[-0.0400, 0.0400] [-0.0500, -0.0110]
Prel. Opt. Fitness	4.7364 (**)	3.8927 (*)	/
Prel. Opt. Point	[0.0087, -0.0190]	[-0.0021, -0.0190]	/
<i>New Bounds</i> (x,y)	[-0.0155, 0.0571] [-0.0445, -0.0190]	[-0.0494, 0.0584] [-0.0885, -0.0190]	/
<b>Iterations</b>	10	10	8
<b>Optimal Fitness</b>	3.1045	3.1031	3.0742
<b>Optimal Slot</b>	[0.0094, -0.0204]	[0.0099, -0.0203]	[0.0094, -0.0220]
<b>% Error w.r.t. [9]</b>	-3.29	-3.33	-4.23

Table 4.11: PSO results for Tests N, O, P.

(\*): preliminary study result computed with Euler CFD.

(\*\*): preliminary study result computed with Hess-Smith panel method; objective function is maximum lift.

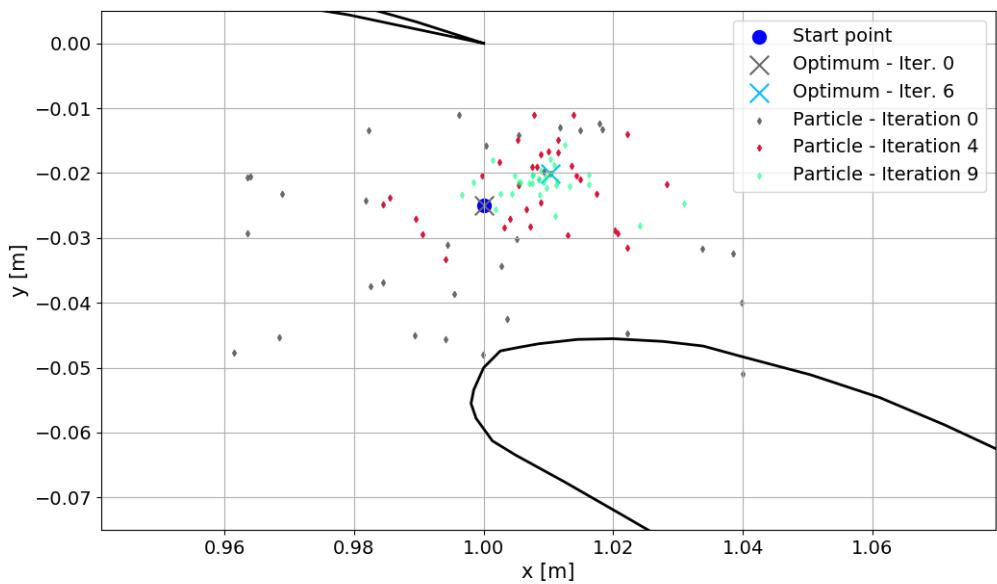


Figure 4.13: Test P PSO history in design space, with particle positions examples.

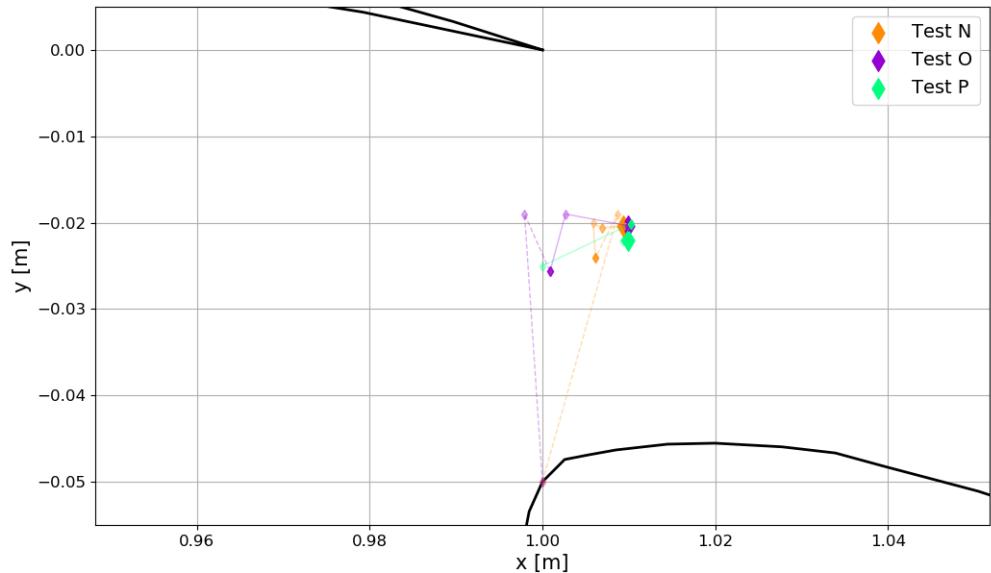


Figure 4.14: Tests N, O, P respective histories in design space.

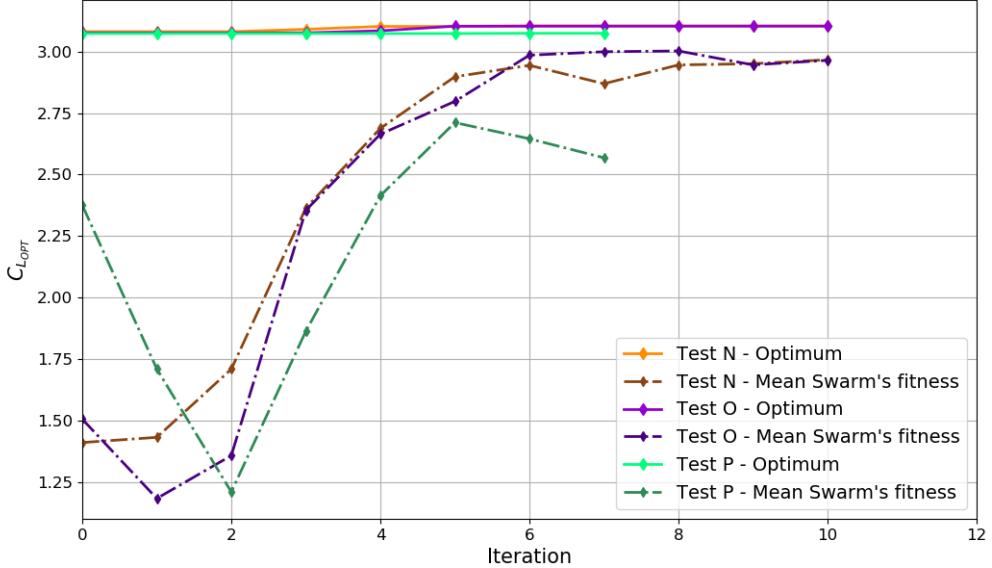


Figure 4.15: Test N, O, P history optimisation w.r.t. optimal and swarm's mean fitnesses found at each iteration.

As it was for previous Tests with Steepest Ascent, preliminary studies try to reduce design variables bounds' sizes and to find a new starting point, based on Hess-Smith panel method or Euler CFD. Once preliminary study ends, PSO considers a swarm of particles inside the new bounds, with also the new starting point. Subsequently, as already explained in Section 4.2, PSO moves particles in the direction of found optimal positions (plus an additional randomizing factor to avoid stuck into local optima). About Tests N, O a significant reduction of the bounds thanks to preliminary studies based on panel method or Euler CFD is observable, making these first procedures particularly efficacies.

The optimal slot was found in the same region of Tests N, O, confirming optimal solution. Test P involved different grid settings to allow higher y-coordinate bounds with respect to the other two tests, as already cited: results show anyway an approximately equal optimal slot among all three tests.

### Tests U, V: Slot optimisation with 35 deg relative flap AoA

Tests U, V consider a higher value for flap's relative AoA of 35 deg, while absolute value is fixed at 3.3 deg. This new configuration was to verify optimal slot position since not available for 30 deg in Wentz and Seetharam experimental data for comparison. At 35 deg relative flap AoA, the flow condition is particularly difficult since at higher relative flap AoA, the experimental data show a decrease in performance, suggesting a stall condition.

For this reason, CFD could be particularly difficult and may require a higher iterations amount to reach convergence close to stall condition. Consequently, the minimum amount of RANS iterations to be performed was increased from 1500 of previous tests to 2000.

As already explained, a proper preliminary study can avoid issues with RANS history due to oscillations, and can favour convergence to optimum. Test U will consider PSO with Euler CFD preliminary study, while Test V will consider Steepest Ascent with Hess-Smith panel method preliminary study, with maximum lift as objective function. The respective minimum lift values for the candidates to be maintained inside the subsequent bounds of optimisation procedure are (3.4) and (4.0).

Except from AoA and RANS minimum iterations, the selected parameters for preliminary study and optimisation procedure were the same of previous Tests.

	<b>Test U</b>	<b>Test V</b>
<b>Start Point</b>	[0.00, -0.03]	[0.00, -0.03]
<i>Start Bounds</i> ( $x,y$ )	$[-0.0500, 0.0500]$ $[-0.0500, -0.0105]$	$[-0.0500, 0.0500]$ $[-0.0500, -0.0105]$
Prel. Opt. Fitness	3.7857 (*)	4.4541 (**)
Prel. Opt. Point	$[-0.0039, -0.0105]$	$[0.0118, -0.0105]$
<i>New Bounds</i> ( $x,y$ )	$[-0.0500, 0.0474]$ $[-0.0500, -0.0105]$	$[-0.0206, 0.0500]$ $[-0.0500, -0.0105]$
Iterations	14	8
Optimal Fitness	3.0053	2.9958
Optimal Slot	$[-0.0073, -0.0181]$	$[-0.0054, -0.0171]$

Table 4.12: Tests U, V optimisation results.

(\*): preliminary study result computed with Euler CFD.

(\*\*): preliminary study result computed with Hess-Smith panel method; objective function is maximum lift.

	<b>Test U</b>	<b>% Error/Difference</b>	<b>Wentz &amp; Seetharam</b>
$C_{L_{MAX}}$ Optimal Slot	3.0053 $[-0.0073, -0.0181]$	-6.67 % $[1.23 \%, 1.70 \%)$	3.22 $[-0.0196, -0.0351]$

Table 4.13: Test U comparison with experimental data [9].

	Test V	Error/Difference	Wentz & Seetharam
$C_{L_{MAX}}$ Optimal Slot	2.9958 [-0.0054, -0.0171]	-6.96 % [1.42 %, 1.80 %]	3.22 [-0.0196, -0.0351]

Table 4.14: Test V comparison with experimental data [9].

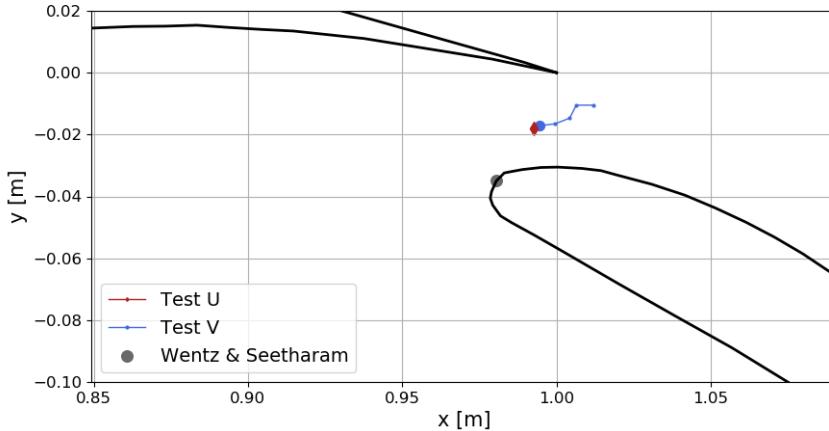


Figure 4.16: Optimal slot locations found for 35 deg relative AoA flap condition of GA(W)-1.

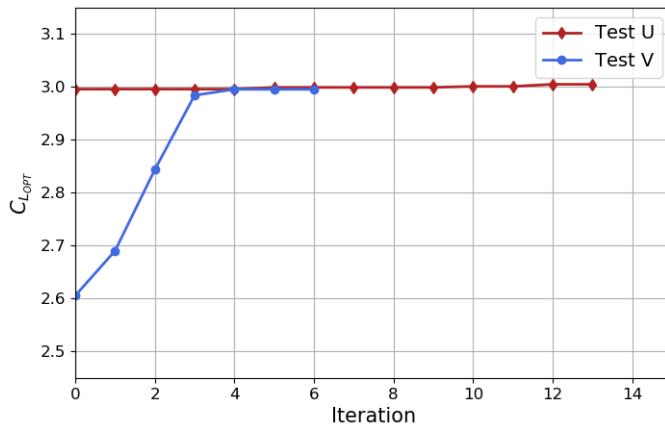


Figure 4.17: Optimal fitness histories performed for 35 deg relative AoA flap condition of GA(W)-1.

Figure 4.16 shows the optimal slot position: about Test U performed with PSO, optimal slot found at each iteration sees negligible changes both on (x,y) location and fitness.

The two methods essentially found the same optimum slot, while experimental results by Wentz and Seetharam showed a larger slot as optimum. However, maximum lift is again under a 7% error, while slot coordinates under 2% chords.

Recalling again geometries were web-provided, results are acceptable.

#### 4.5.2 Comparison between PSO and Steepest Ascent results

The several optimisations performed for the GA(W)-1 slot allow a more specific comparison between Steepest Ascent and PSO performances, referring especially to GA(W)-1 with AoA = [7.7, 37.7] results. Figure 4.18 reports the optimal fitness found using the two optimisation algorithms.

Note that it essentially reports the entire domain of analysis below the main airfoil TE; furthermore, recall Tests E and P were performed with lower structured region thickness: consequently, slight differences in optimal slot and lift can be noticed also due to different grid features (as sensitivity analysis executed reported in Section 3.3).

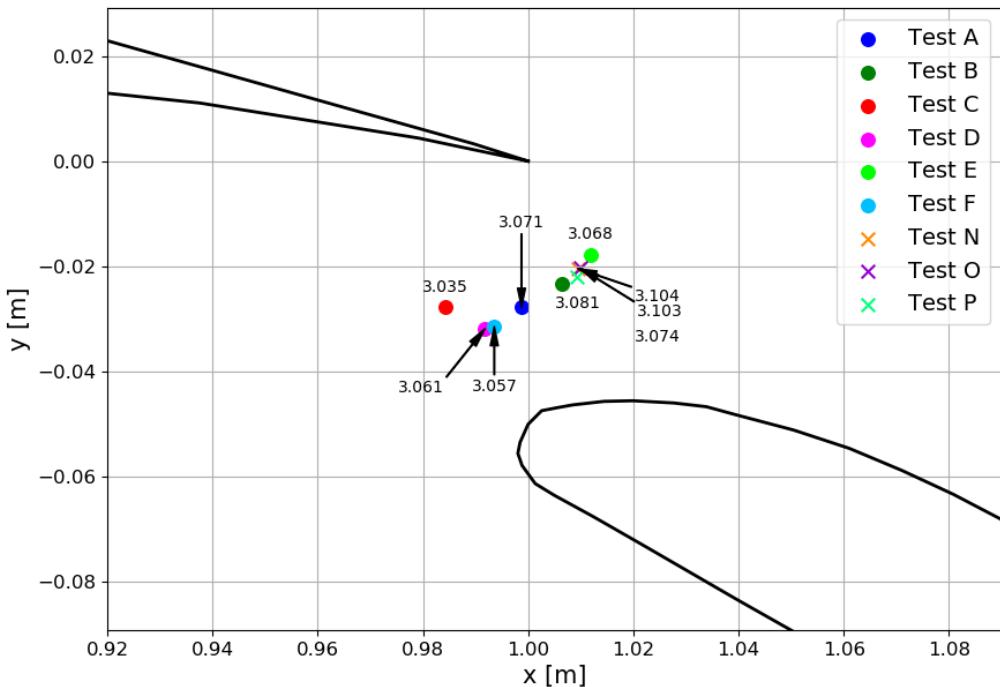


Figure 4.18: Optimal slot location and fitness found by respective PSO and Steepest Ascent tests.

Results show a region where the maximum lift coefficient increases to approximately 3.10 corresponding to a gap of -2% and an overlap of +1% of the main element chord. About the maximum lift coefficient, these results satisfy a low error requirement with respect to experimental data, staying under an absolute value of 5.5%.

Comparing the two methods, Particle Swarm Optimisation Tests N, O provide more precise results with respect to Steepest Ascent tests, which could stuck into local optimum typical of gradient-based methods (examples: Tests C, E, F). Consequently, PSO is more robust since it favours detection of global optima, while Steepest Ascent appears more sensible to parameter settings like reference size of the optimisation candidates selection's triangle (or generally hyper-pyramid for N design variables).

The starting point is quite important for Steepest Ascent, as Tests A, B, C showed. Another observation can be stated about the orientation and dimension of the triangle

(or hyper-pyramid, for N-dimensional problem): Test E showed that proper management of the triangle could avoid oscillations in the design space, favouring convergence. Furthermore, the history and result of Test E confirmed what already found experimentally for GA(W)-1 configuration [9]: around the "peak" of maximum and optimal fitness, fitness' isolines are very close one each other, proposing a large gradient close to the optimal region which brings lift coefficient from a value of  $2.7 \div 2.8$  to optimal values of  $3.0 \div 3.1$ .

Future developments of Steepest Ascent optimisation methodology could favour convergence; for example, a possibility may be setting the orientation of the hyper-pyramid as random or gradient-based.

Another possibility is determining proper starting design variables and bounds through the Hess-Smith panel method or Euler CFD as proposed for preliminary studies. Results are usually positive (like Tests F, N, O) favouring convergence, with Euler CFD generally being more precise.

To conclude the summary about preliminary studies, these low-fidelity methodologies help finding a proper starting point and reducing the bounds to the interested region without using expensive computational resources.

However, the aim of the present analysis was not defining an optimal setup for both optimisation algorithms, but demonstrating the feasibility of reaching optimal results by using a fully automated CFD optimisation procedure, an aim that was widely satisfied with proper measures on bounds and starting point.

## 4.6 Shape Optimisation of IGP multi-element airfoil

To highlight the potentiality of a fully automated CFD optimisation procedure about multi-element airfoils, a different problem was set evaluating performances about shape optimisation of a double-element airfoil generated using IGP parametrisation. The geometry and flow conditions were selected about typical racing cars' multi-element airfoils inspired by Benzing airfoils [2].

The chosen starting geometry has the following IGP parameters:

- main element: [0.35, 0.6, -0.1, -0.1, 0.3, 0.15, 0.3, 2];
- flap element: [0.3, 0.6, -0.1, -0.1, 0.3, 0.16, 0.3, 1.5];

Starting geometry and related grid are reported Figures 4.19 - 4.20.

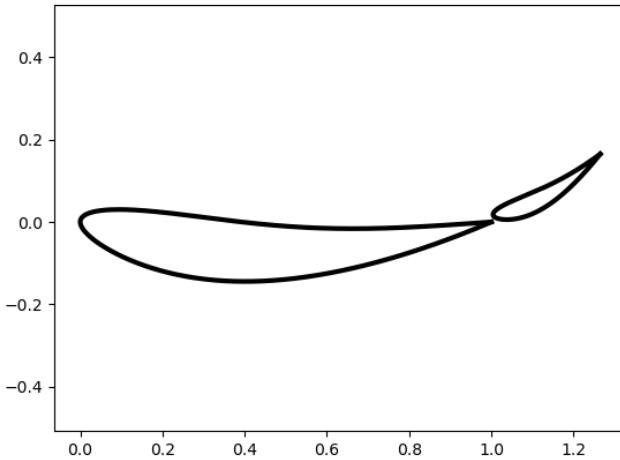


Figure 4.19: IGP shape optimisation: starting airfoil geometry.

The absolute angles of attack are, respectively: [-10, -40] deg. Flap chord is 30% of main element one.

Note third and fourth parameters of both elements have been imposed outside IGP authors' proposed bounds [21] to create a proper configuration for racing cars applications.

To avoid issues due to variations inside their respective domains, the third and fourth parameters of both elements were fixed during optimisation, limiting the total amount of design variables to (12). Due to the high total amount of design variables and due to previous observations about accuracy, Particle Swarm was chosen for the optimisation. Table 4.20 reports starting bounds of each design variable.

The main features about flow conditions, grid details and optimisation parameters are reported in Tables 4.15 - 4.16 - 4.17.

Flow condition	Value
Mach number [-]	0.122
Reynolds number [-]	$10^6$
Temperature [K]	293.15
Dynamic Viscosity [ $\text{kg}/\text{m} \cdot \text{s}$ ]	$1.853 \cdot 10^{-5}$
Slot coordinates (x,y)	[0.005, 0.015]
Boundary-Layer	Fully Turbulent

Table 4.15: Flow and geometry conditions used for IGP shape optimisation.

CFD Configuration Feature	Value/Method
CFL	500
Adaptive CFL	NO
Numerical Scheme	Roe + MUSCL
Method for spatial gradients	Weighted Least Squares
Time discretization	Euler Implicit
Convective method	Scalar Upwind

Table 4.16: Computational configuration features used for IGP shape optimisation.

Grid Parameter	Value
<i>nodes</i>	0.95
<i>ellipse_dimension</i>	0.85
<i>ellipse_refining</i>	0.9
<i>y_plus</i>	[0.04, 0.04]
<i>thick</i>	[0.5, 0.5]
<i>progr</i>	[1.15, 1.15]
<i>wall_refining</i>	[1.2, 1.2]
<i>wake_length</i>	50
<i>wake_progr</i>	1.3
<i>semicircle_dimension</i>	18
<i>semicircle_elem_factor</i>	250

Table 4.17: Grid features used for IGP shape optimisation.

About the optimisation procedure, the Particle Swarm was adopted plus a preliminary study using the Hess-Smith panel method with minimum lift as objective function. The related details are reported in Tables 4.18 and 4.19.

Method	HS
Samples	300
criterion	<i>Min. Lift</i>
Bounds margin	-4.75

Table 4.18: Preliminary Studies features for IGP airfoil shape optimisation.

PSO Parameter	Value
<i>swarmsize</i>	40
<i>max_iter</i>	15
<i>min_step</i>	$5 \cdot 10^{-4}$
<i>min_func</i>	$5 \cdot 10^{-3}$

Table 4.19: PSO features used for IGP airfoil shape optimisation.

Preliminary study's candidates were selected using Latin Hypercube design with starting design variables as centre and respective standard deviation of:

$$\text{Element (1)} : [0.075, 0.08, 0.08, 0.08, 0.085, 0.85]$$

$$\text{Element (2)} : [0.075, 0.08, 0.08, 0.075, 0.085, 0.8]$$

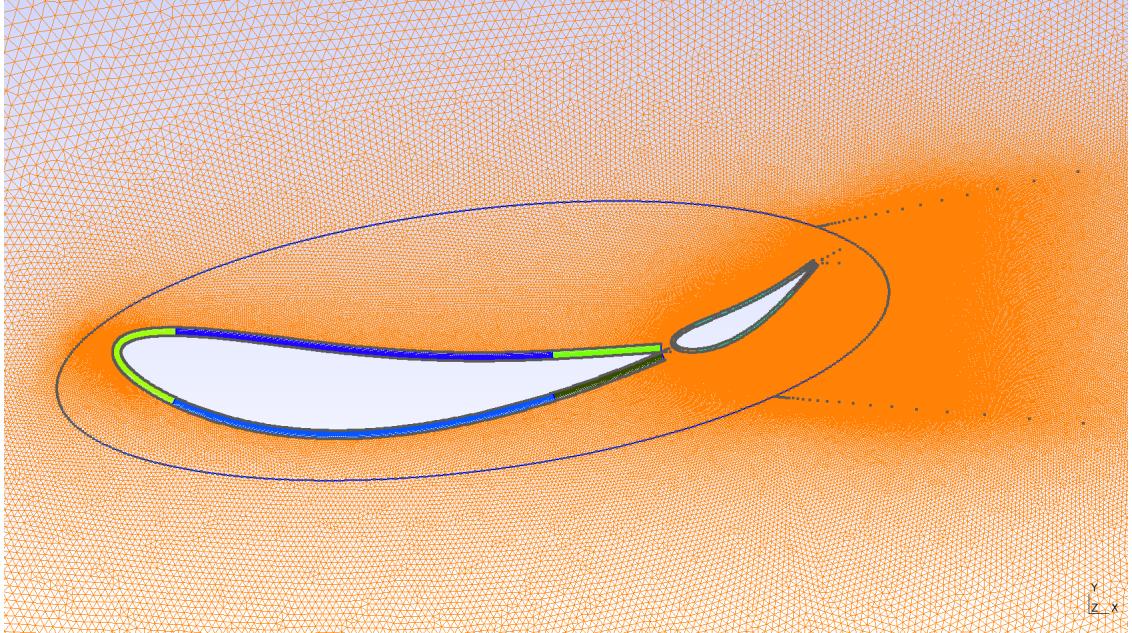


Figure 4.20: IGP shape optimisation: starting airfoil grid.

Table 4.20 displays the variation of bounds thanks to the preliminary study using the Hess-Smith panel method and minimum lift as the objective function. Comparisons make evidence of not so relevant variations in design variables' bounds, except for some parameters allowing exclusions of extreme (and not interesting) configurations.

Table 4.21 reports the main results and performances obtained thanks to the entire optimisation procedure. To favour readability, IGP parameters have been rounded to the fourth digit.

Results are overall positive: the final shape of double-element IGP airfoil sees a higher lift coefficient. Consequently, the procedure appears successful also for a very different problem like shape optimisation. An observation must be executed about the preliminary study adopted for this test case: although the panel method allowed a slight reduction in bounds (which may help optimisation convergence), the starting point through this preliminary study is controversial since an additional CFD simulation performed among this configuration provided less downforce (-1.9834). In the case of PSO,

this note is about the optimal preliminary solution is not quite relevant, as final results show; however, it underlines what already observed in GA(W)-1 slot configuration: preliminary studies are useful whenever the user does not have motivating reasons to select a starting point for design variables or bounds; on the other hand, CFD Euler and Hess-Smith panel method respectively guarantee starting point and bounds based on low-fidelity conditions.

	Design variable - IGP ID	Start Bounds	Bounds after Preliminary
Main element	1	[0.2, 0.4]	[0.2, 0.4]
	2	[0.4, 0.9]	[0.4, 0.7769]
	5	[0.21, 0.4]	[0.21, 0.4]
	6	[0.0700, 0.25]	[0.0700, 0.25]
	7	[0.290, 0.9]	[0.290, 0.5572]
	8	[0.5, 3]	[0.5, 3]
Flap	1	[0.2, 0.4]	[0.2, 0.4]
	2	[0.4, 0.9]	[0.4, 0.8173]
	5	[0.21, 0.4]	[0.21, 0.4]
	6	[0.0800, 0.25]	[0.0800, 0.25]
	7	[0.290, 0.9]	[0.290, 0.5574]
	8	[0.5, 3]	[0.5, 3]

Table 4.20: Bounds considered for shape optimisation of IGP racing car double-element airfoil, (before and after preliminary study with HS panel method).

IGP double-element airfoil - shape optimisation	
<b>Start Geometry</b>	[0.35, 0.6, -0.1, -0.1, 0.3, 0.15, 0.3, 2] [0.3, 0.6, -0.1, -0.1, 0.3, 0.16, 0.3, 1.5]
<b>Start <math>C_L</math></b>	-2.8265
<b>Geometry after Prel.</b>	[0.3645, 0.6263, -0.1, -0.1, 0.2889, 0.2392, 0.3967, 1.8037] [0.2279, 0.7631, -0.1, -0.1, 0.4000, 0.2061, 0.3644, 2.2311]
<b><math>C_L</math> after Prel.</b>	-4.9435 (**)
<b>Optimal Geometry</b>	[0.2413, 0.5259, -0.1, -0.1, 0.2139, 0.1287, 0.4450, 1.1047] [0.4000, 0.7948, -0.1, -0.1, 0.2817, 0.0819, 0.3340, 1.7963]
<b>Optimal <math>C_L</math></b>	-3.0398

Table 4.21: Shape optimisation of IGP airfoil results and performances.

(\*\*): preliminary study result computed with Hess-Smith panel method; objective function is maximum lift.

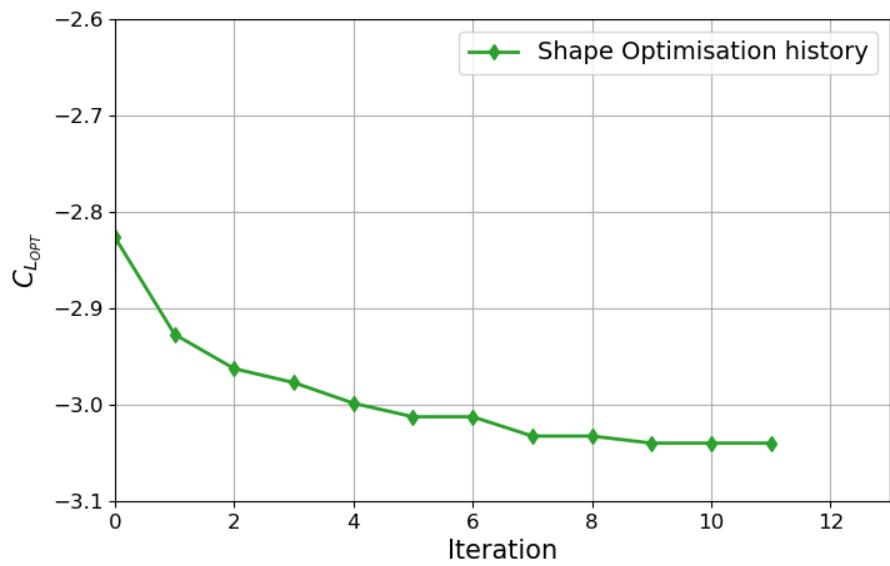


Figure 4.21: Fitness' history for IGP shape optimisation.

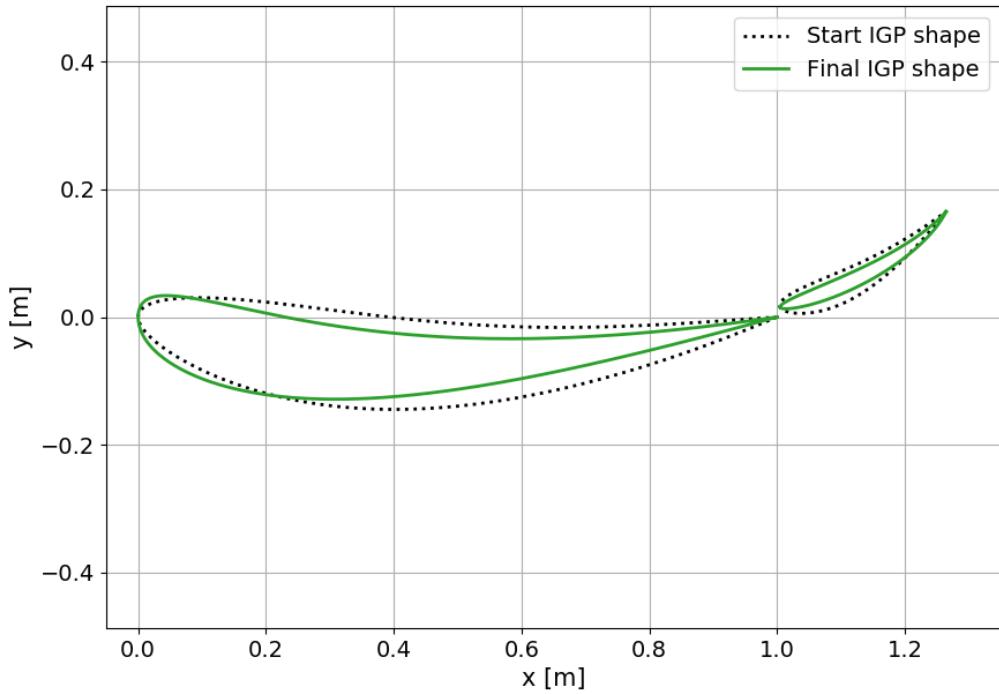
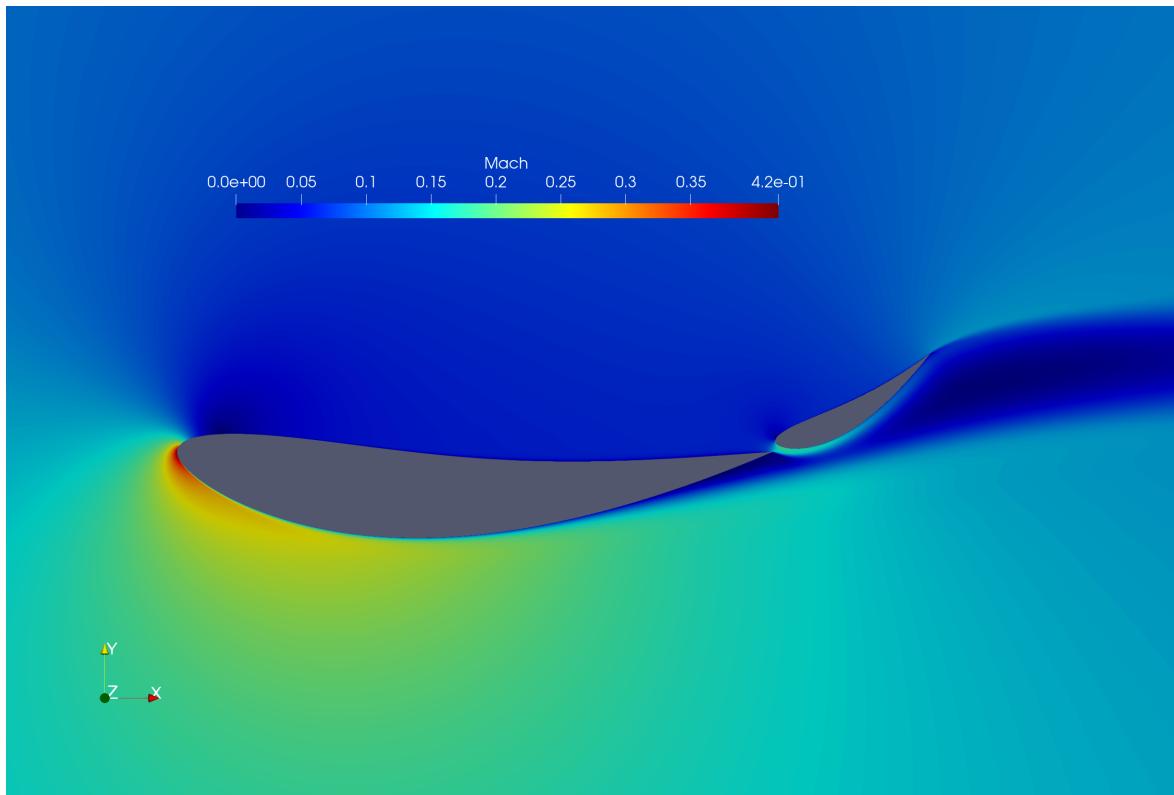
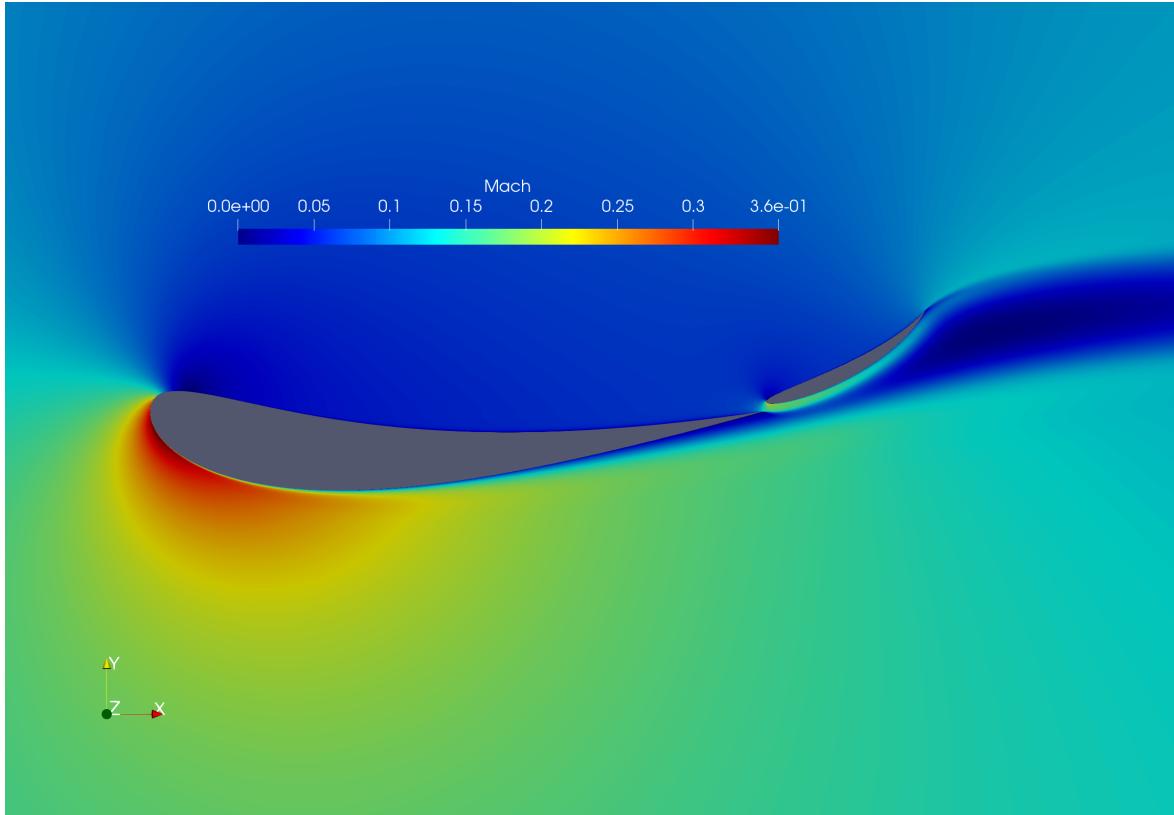


Figure 4.22: Comparison between starting and optimised respective IGP shapes.



(a)



(b)

Figure 4.23: Comparison between starting (a) and optimal IGP shape optimisation (b) Mach visualizations.

Results report a significant improvement in downforce, specifically 7.55% in 12 iterations as Figure 4.21 shows. Consequently, although the starting geometry was a good starting design already, PSO provided a more accurate shape with improved downforce performances.

Comparing the starting and final geometries (Figures 4.22 and 4.23), the difference between them is evident, especially on thickness reduction. Moreover, the flap has the first parameter equal to the lower bound setted, suggesting lower limits could have provided even better fitnesses. Again, the importance of bounds is underlined, like it was for slot optimisation of GA(W)-1 airfoil.

Comparing with similar literature airfoils, the final shape is comparable again with Benzing airfoils, thought and developed for racing cars applications.

To conclude, the procedure responds well also to very different problems like shape optimisation with a higher amount of design variables.



# Chapter 5

## Conclusions

A complete automated methodology for multi-element configurations' optimisations using CFD has been presented to demonstrate feasibility and potentialities of the procedure.

Along with this general aim, a tool was written using completely free-based software (Python3, Gmsh, SU2, Paraview), while Matlab was adopted only for Hess-Smith panel method preliminary studies. The user can customize several aspects: from geometry to grid features, optimisation parameters and objectives, preliminary studies and CFD RANS details.

About the geometries, the user can import an external file with the geometry points of each element or using a default parametrisation between IGP, NACA 4-digits, 5- digits, 4-digits modified, 16 series or some Benzing airfoils.

About the grid, the procedure uses a hybrid grid to guarantee adaptability to various customized problems: the proposed code generates a grid that can adapt to flow conditions and configurations. Refinement is guaranteed in various ways and locations, again customizable by the user: close to airfoils' surfaces through an external grid ellipse, inside slots, along wake and close to the wall surface in the normal direction. About the latter, a structured region based on Schlichting empirical formulas is generated: the user can manage it with parameters to eventually guarantee  $y^+ < 1$ .

To validate and analyse the proposed code to generate grids, a deep detailed grid sensitivity was performed in Section 3.3. Here, various tests were executed trying to delineate how the grid behaves and responds to different input parameters' settings.

Deeply described in Chapter 3, grid and single CFD simulations were validated executing various tests on NACA 0012, NACA 23012 and NLR 7301 geometries, comparing with experimental data available in literature.

The proposed results have been satisfying overall.

Once the proposed code for grid generation and CFD running was widely verified and validated, it was adopted inside optimisation procedures. In particular, the "Optimisation module" implemented in the tool allows users to customize several parameters and details. If the input airfoil is generated using IGP parametrisation, the user can also perform a shape optimisation.

The methods currently available are Particle Swarm Optimisation (PSO) and Steepest Ascent (or Descent). If needed, the user can add a preliminary study to individuate a proper starting point or bounds for the optimisation, based on either Hess-Smith panel method or Euler CFD simulations about an ensemble of candidates generated using Latin Hypercube or Box-Behnken designs. In the case of maximum lift objective func-

tion and panel method, user can base also the preliminary study on the Valarezo-Chin criterion, if desired.

To validate the procedure, several tests were performed combining all the available optimisation features, firstly focusing on a GA(W)-1 multi-element configuration.

Among all test cases, the importance of starting point location and bounds' definition was underlined. Firstly, when adopting a gradient-based method like Steepest Ascent, a proper starting point can significantly improve the convergence speed of the entire optimisation procedure. Bounds reduction to a region of interests help again from the point of view of convergence speed, focusing the analysis on the interesting region; furthermore, large permissive bounds could involve computational issues: whenever the flap is too far from the main element, from a physical point of view the beneficial effects of multi-airfoil configuration drop, causing stall on the flap element due to its high absolute angle of attack. From a computational point of view, RANS usually struggle to reach convergence due to the unsteady nature of stall conditions. For these reasons, adequate bounds definition becomes mandatory to avoid convergence issues of a CFD simulation. This topic becomes much more relevant when an optimisation algorithm based on a large population of candidates (like Particle Swarm Optimisation or Genetic Algorithms) is considered since the random features of these methods usually lead to considering design variables close to design space's bounds.

Preliminary studies were adopted to define a first optimal solution along with new bounds. The first optimum is not precise due to the low-fidelity nature of preliminary methods adopted. About GA(W)-1 slot optimisation and IGP shape optimisation, performed test cases showed the Hess-Smith panel method is less precise with respect to Euler CFD, as expected. Panel method becomes anyway useful as a preliminary study since each candidate require few seconds of evaluation (depending on the total amount of nodes considered for each element).

However, test cases showed preliminary studies implemented appear to be successful overall as low-fidelity first methodologies, indicating in most cases a proper starting point, new bounds and improving robustness and speed of the entire optimisation procedure. Euler CFD appears more precise with respect to the Hess-Smith panel method. In the latter case, also the Valarezo-Chin criterion provided satisfying results whenever called for maximum lift optimisations.

Although preliminary studies may be not so precise (since using low-fidelity methodologies), they can reduce the design space to an interesting region and providing a proper starting point for the following high-fidelity viscous CFD optimisation which eventually achieves a more accurate optimum, confirming the trends of experimental data. The proposed preliminary methodologies are suggested whenever the user has no idea or experience to properly select starting variables and bounds.

Comparing the two optimisation methods, Particle Swarm appeared to be more accurate and better performing with respect to Steepest Ascent: thanks to the large availability of candidates to evaluate optimal fitness inside design space, PSO has higher probabilities to detect the global optimum. However, Steepest Ascent outputs involved a smaller amount of CFD simulations, while the optimal fitness found was anyway satisfying in several cases. Consequently, a proper setup of the Steepest Ascent method along with some developments can guarantee good performances to the methodology, maintaining its typical lower requirement of computational resources.

Shape optimisation of a double-element IGP airfoil was performed to demonstrate feasibility also in a very different problem: along with the changed typology of design

variable, the chosen airfoil recalls a racing car multi-element airfoil, with lower Reynolds number and flow velocities. Although starting from an already well-defined configuration, the proposed procedure provided an optimal shape with sensibly higher downforce, demonstrating again the feasibility and good performances of the procedure.

Although without an optimal tuning of PSO and Steepest Ascent respective parameters, results are overall positive, suggesting strong potentialities at the current stage already, but especially for further developments.

## **Further Studies and Developments**

Based on the obtained results and feasibility of the procedure, several developments can be considered in future studies, referring especially to the tool built in the present thesis.

An interesting development of the tool would be adding support for different grid generators and CFD solvers.

For the grid, the possibility of generating a structured grid could be considered, but this involves several complexities from the programming point of view especially when multi-airfoil configurations have to be meshed.

Also, the user cannot study excessively small gaps due to the constraint of non-overlapping structured regions. This is something that could be addressed by locally shrinking the size of the structured region or by decreasing its thickness globally.

Looking at the optimisation procedure, the implemented methods can be deeper investigated to evaluate optimal settings specifically for multi-element configurations' optimisations. Moreover, possible extensions can regard implementing other new algorithms like Genetic Algorithm, Simplex method, etc.

Another important topic is about guaranteeing a full optimisation, performing configuration and shape optimisation at the same time: at the current stage, the two optimisations have been implemented separately inside respective folders. The upgrade can guarantee the possibility to consider all design variables about configuration and shape, extending to the maximum amount of design variables and opening the analysis to complete multi-element optimisations.

This upgrade is ready, but it will be tested in the next future since it requires a high computational power: a contemporaneous optimisation of shape and configuration of a double-element IGP configuration involves 21 design variables and, consequently, several configurations to be tested.

Lastly, results reported the importance of delineating proper starting points and bounds for both convergence speed and procedure robustness; the preliminary studies implemented was successful by this point of view, but further studies with different multi-element airfoils or different configurations should be performed to further investigate potentialities, accuracies and advantages/disadvantages.



# Appendices



## Appendix A

# Tool Design

This chapter briefly describes the features and characteristics of the tool by a conceptual point of view.

The present thesis' Python code is able to perform the following CFD features:

1. grid generation, without initializing a CFD simulation;
2. a single complete CFD simulation;
3. configuration optimisation of a multi-element airfoil;
4. shape optimisation of a multi-element IGP airfoil;
5. post-processing for a single CFD simulation.

All these features can be executed separately. To facilitate the management of external input data (may web-provided), Folder (0) has been added to perform geometry scaling and rotation separately, providing a new data file in output.

Configuration and shape optimisation respectively gather all the other modules including grid generation, CFD simulation and specific post-processing. The possibility to generate a single grid or perform a single CFD simulation was included in Folders (1) and (2) to allow for simple or preliminary analysis without initialising the optimisation procedure. On the other side, Folders (3) and (4) respectively give the possibility to perform a configuration optimisation (slot and AoA of a multi-element airfoil) and a IGP shape optimisation.

In each Folder, the user can impose his own inputs by customizing all the Python files with **main\_** as starting filename.

About post-processing, note SU2 already provides various informations in output. Further useful post-processing details can be exported using Folder (5), where the user can easily find out various computational details (polars, convergence, pressure distribution etc.) typically interesting in CFD simulations about airfoils.

In particular, Folder (5) aims at guaranteeing an easy post-processing analysis for CFD simulations providing various Python scripts:

- convergence studies' scripts, which analyse the history of the RANS;
- polar post-processing, useful when user has performed various CFD simulations on the same airfoil configuration at several angles of attack. These scripts require manual writing of aerodynamic coefficients obtained;

- wall surface analysis' scripts, from which plots about thermodynamic quantities on airfoils' surfaces are provided.

Convergence study requires the file *history.csv*, while wall surface analysis requires *surface\_flow.vtu*. These files are provided in output after CFD simulation by SU2 [32] [92].

The results about polars and wall surface quantities can be compared with experimental results, which should be provided using proper text files. Some of these experimental results are organised inside the directory, with the purpose of creating a database of experimental data in the future.

Consequently, Folder (5) allows for post-processing related to single CFD simulations: Folder (3) and (4) directly provide specific post-processing data files about respective optimisations.

Based on the tool organisation, Chapter 3 will essentially describe how Folders (1) and (2) provide grid and initialize CFD simulation along with CFD single simulation results about NACA 0012, NAC 23012, NLR 7301.

Chapter 4 describes how Folders (3) and (4) respectively perform configuration and shape optimisation, ending with validation results again. The test cases considered refer to a GA(W)-1 double element configuration where optimal slot is searched, and shape optimisation of a double-element configuration for racing cars applications.

In all the cases, the proposed results were obtained using the grids generated by the tool: along with possible computational errors or experimental ones which occur in comparisons, the grids have not been "optimised" to obtain the best possible results obtainable using the tool. Lastly, geometries obtained from external files (NLR 7301, GA(W)-1) are web-provided and not fined fitted about geometries' points; then, consequences can affect results' accuracy.

## A.1 GA(W)-1 slot optimisation

### A.1.1 Points coordinates from external files

#### Main Element

1.0000029054 0.0000000070	0.0776694644 -0.0446398017
0.9792142118 0.0043781809	0.0516974194 -0.0378132782
0.9584148857 0.0077149656	0.0387193711 -0.0336189745
0.9376155597 0.0110517503	0.0257530185 -0.0282791426
0.9147235423 0.0134725463	0.0128015511 -0.0214813657
0.9043121837 0.0139954104	0.0050610793 -0.0144243262
0.8939018883 0.0146224134	0.0019842414 -0.0097061820
0.8834926562 0.0153535554	0.0000000000 0.0000000000
0.8730727917 0.0150433081	0.0022213429 0.0135167989
0.8626550537 0.0149413387	0.0054247058 0.0211911870
0.8522341259 0.0145269525	0.0133459278 0.0318377653
0.8418131982 0.0141125663	0.0264823981 0.0431601618
0.8313858911 0.0130733465	0.0395864398 0.0513063211
0.8209522045 0.0114092931	0.0526723001 0.0576717046
0.8105132018 0.0092245451	0.0788136123 0.0674240984
0.8000688829 0.0065191025	0.1049322775 0.0749583331
0.7896203110 0.0033971041	0.1310351005 0.0809408977
0.7791621701 -0.0006621446	0.1571272911 0.0858820731
0.7686965864 -0.0054503657	0.1832131024 0.0901984148
0.7634643263 -0.0077924069	0.2092904077 0.0936816452
0.7582214337 -0.0111758373	0.2614290699 0.0990860220
0.7519242212 -0.0157983040	0.3135485938 0.1026158981
0.7477207633 -0.0194006431	0.3656521692 0.1045836903
0.7435098628 -0.0237319547	0.4177399023 0.1049998125
0.7414012228 -0.0262100273	0.4698130692 0.1039892313
0.7396082409 -0.0283788723	0.5218712444 0.1015102913
0.7287313080 -0.0428488245	0.5739123015 0.0973547145
0.7026454968 -0.0471651663	0.5999256001 0.0945687814
0.6765596856 -0.0514815081	0.6259330508 0.0912100842
0.6504770640 -0.0554854331	0.6519332716 0.0871432423
0.6243965690 -0.0592810802	0.6779290268 0.0826390169
0.5983235166 -0.0623477549	0.7039188278 0.0775516135
0.5722525907 -0.0652061517	0.7299044823 0.0720580683
0.5201320035 -0.0688401667	0.7558858837 0.0661479674
0.4680273649 -0.0709120978	0.7818633513 0.0598525524
0.4159344219 -0.0718385007	0.8078379480 0.0532759624
0.3638531744 -0.0716193754	0.8338109500 0.0465431639
0.3117825594 -0.0703588609	0.8597819318 0.0396125015
0.2597236399 -0.0679528181	0.8857509997 0.0324943890
0.2076817322 -0.0638805525	0.9117184728 0.0252200682
0.1816671577 -0.0612195861	0.9376856269 0.0179145056
0.1556568362 -0.0581420640	0.9636525684 0.0105881153
0.1296528941 -0.0544397083	0.9896199352 0.0033033805
0.1036563947 -0.0500083802	

## Flap element

0.9956221136	-0.1211985345	0.0502289652	0.0462011593
0.9293987185	-0.0990009488	0.0733344213	0.0579079660
0.8622757623	-0.0819050537	0.0923190938	0.0661331320
0.7946730328	-0.0675300590	0.1107040602	0.0709571709
0.7267104803	-0.0551957411	0.1475339610	0.0809453587
0.6581482180	-0.0462625469	0.1839440636	0.0885527596
0.5888063311	-0.0417508165	0.2198144282	0.0930991502
0.5196443557	-0.0362187528	0.2554449097	0.0962850873
0.4502424970	-0.0320471358	0.2907755328	0.0977704609
0.3804208437	-0.0302563091	0.3258063045	0.0975552743
0.3105991869	-0.0284654790	0.3604772462	0.0952994074
0.2402977638	-0.0293955526	0.3947883649	0.0910028670
0.1700563123	-0.0299855128	0.4290994801	0.0867063266
0.0999348006	-0.0298952497	0.4632306873	0.0813894494
0.0648440607	-0.0300201731	0.4974818309	0.0767527991
0.0366755114	-0.0306642926	0.5311332682	0.0687150182
0.0227111835	-0.0303061272	0.5648446737	0.0610173507
0.0097063883	-0.0245061613	0.5986160544	0.0536598001
0.0036237872	-0.0192253881	0.6721814382	0.0333238052
0.0013021397	-0.0125036510	0.7384048333	0.0111262230
0.0000000000	0.0000000000	0.8046282250	-0.0110713627
0.0031193242	0.0117240298	0.8702519069	-0.0366700754
0.0101606292	0.0178468894	0.9359355640	-0.0619286747
0.0184021859	0.0248101709	1.0000000000	-0.0963703117
0.0345254764	0.0366960602		

### A.1.2 Example of "main\_config\_opt" used for slot optimisation

```

1      # Define chosen parametrization of each airfoil.
2      typ = [ "MY_FILE_1" , "MY_FILE_2" ]
3
4      # Specify geometric parameters.
5      params = [ [ "MY_FILE_1" ] , [ "MY_FILE_2" ] ]
6
7      # Specify Trailing Edge type
8      TE = [ 'N' , 'Y' ]
9
10     n = 500
11
12     # Relative respective chord (%).
13     crel = [1 , 0.3]
14
15     # Free-stream features
16     Mach = 0.21
17     Re = 2.2e6
18     mu = 1.853e-05
19     temp = 288.15
20
21     # ===== OPTIMISATION ALGORITHM
22     # Optimisation Objective function
23     objective_function = [ 'MAX' , 'LIFT' ]
24
25     # Optimization Method
26     optim_method = 'STEEPEST'
27
28     # Design Variables and bounds
29     design_variables_alpha = [ 'N' , 'N' ]
30     design_variables_dist = [ [ 'Y' , 'Y' ] ]
31
32     # Preliminary study
33     preliminary = 'YES'
34     preliminary_method = [ 'HS' , 'LHS' ]
35
36     # Optimisation design space bounds.
37     start_alpha = [ 7.7 , 37.7 ]
38
39     # Relative slot coordinates
40     start_dist = [ [ 0 , 0 ] , [ 0.0 , -0.05 ] ]
41
42     # ===== GRID DATA
43     flow = 'VISCOUS'
44     farfield_size = 300
45     nodes = 0.95
46     ellipse_dimension = 0.85

```

```

47 ellipse_refining = 0.9
48
49 # Structured region
50 y_plus = [0.08, 0.12]
51 thick = [0.5, 0.5]
52 progr = [1.15, 1.15]
53 wall_refining = [1.2, 1.2]
54
55 # ===== ADVANCED PARAMETERS
56 # Oscillating RANS history exclusion criterion .
57 limit_exclusions = [ 'YES' , 0]
58 iter_range = 200
59 dev_std = 0.025
60
61 # Reference airfoil
62 ref_airfoil = "DEFAULT"
63
64 # Wake length and elements progression inside .
65 wake_length = 50
66 wake_progr = 1.3
67
68 # Meshing algorithm (an integer from 1 to 9 or 'DEFAULT').
69 Mesh_Algo = 6
70
71 external_pts = 'YES'
72 semicircle_dimension = 18
73 semicircle_elem_factor = 250

```

## Appendix B

# Discussion about $y^+$ and wall spacing

The crucial role of the structured region was presented in Chapter 3: input parameters must guarantee  $y^+ < 1$  along wall surfaces to ensure a proper detection of viscous near wall phenomena.

Among the structured region parameters already presented in Section 3.3.6, input item  $y\_plus$  takes the most important role since it directly affects wall spacing dimension. Several aspects could interfere on actual CFD computed  $y^+$  value on wall surfaces, like airfoil curvature, local velocity or Reynolds number etc.

For these reasons it is suggested a conservative tool input about  $y\_plus$ , imposing a value much lower than 1 and guaranteeing small wall spacings.

However, if the user imposes too small wall spacing values, the amount of total elements could increase unnecessarily. As a consequence, further studies can be executed about this topic, trying to delineate dependencies, particular cases etc.

In the following, comparisons between input and actual computed dimensionless wall distance are performed looking at already presented geometries (NACA 0012, NACA 23012, NLR7301) considering also various angles of attack where possible.

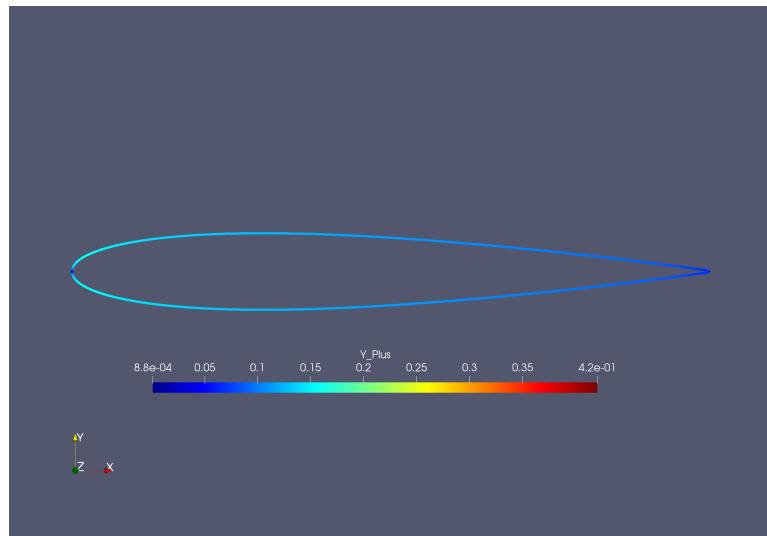
About NACA 0012, results are reported using BC the transition model and fully-turbulent boundary-layer; for the latter, results are referred to final grid (Section 3.4) CFD data.

In the following figures, the coloured band about  $y^+$  range has been maintained the same for NACA 0012 and NACA 23012, while band's bounds have been changed for NLR 7301 since this case presents a lower order of  $y^+$ .

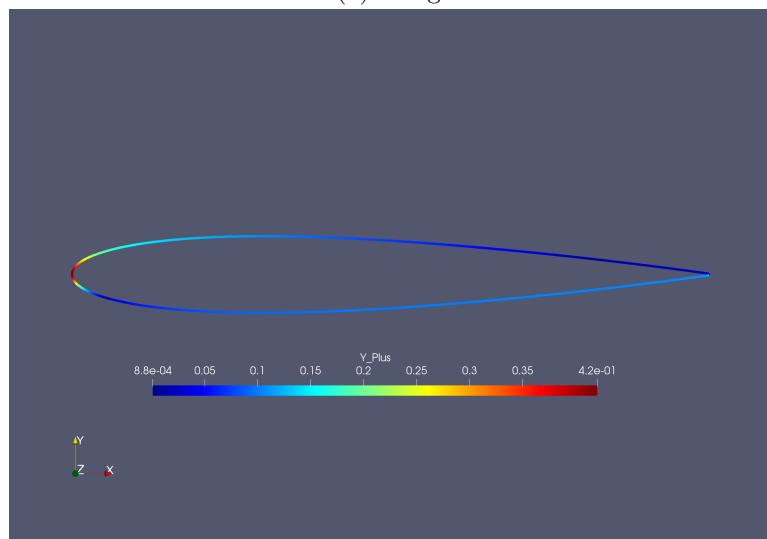
Moreover, the two values inside round brackets regard respectively main and flap element of NLR 7301 data.

Figures B.1 - B.2 - B.3 - B.4 report each case visualization of computed  $y^+$  distribution on airfoil surface.

Table B.1 summarises the respective input parameters  $y\_plus$  and computed CFD  $y^+$ .

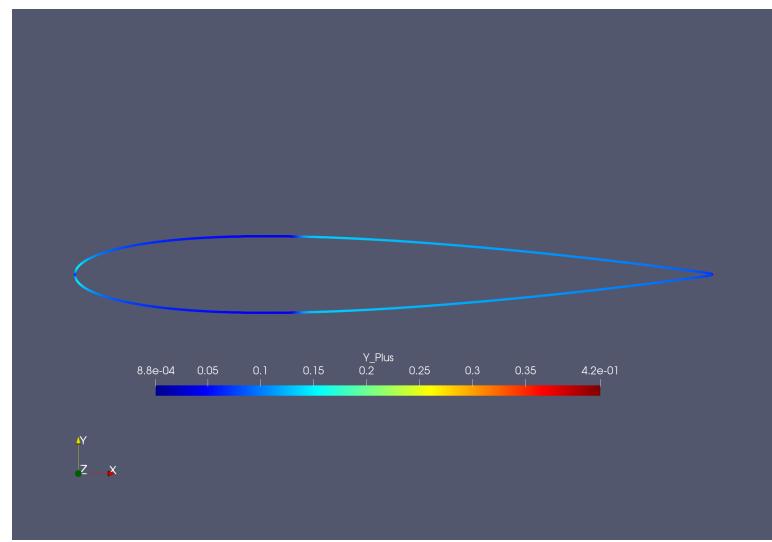


(a) 0 deg

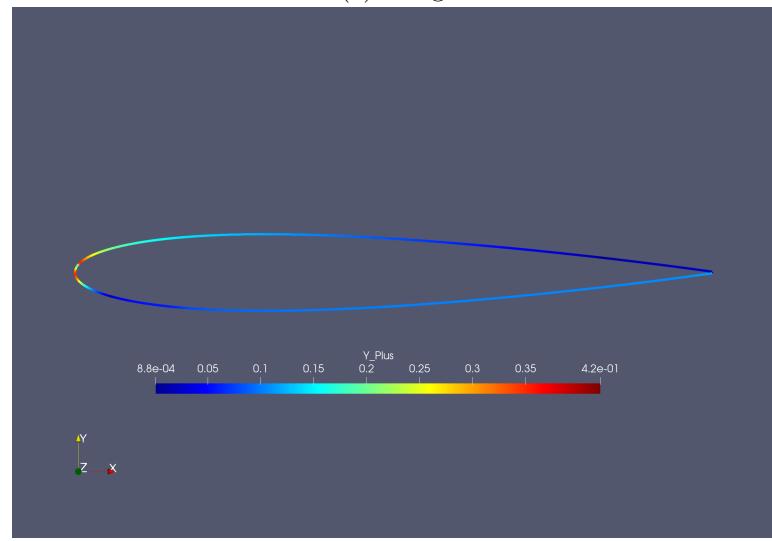


(b) 16 deg

Figure B.1: Surface distributions of  $y^+$  NACA 0012 with fully turbulent boundary-layer.

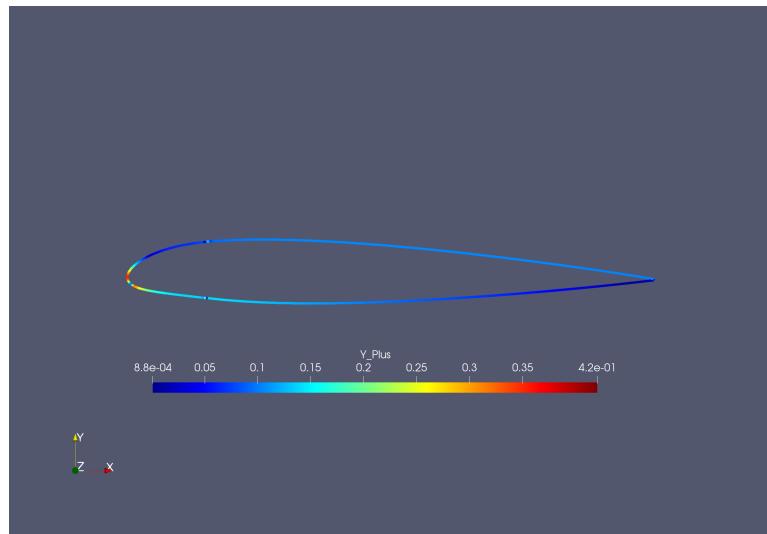


(a) 0 deg

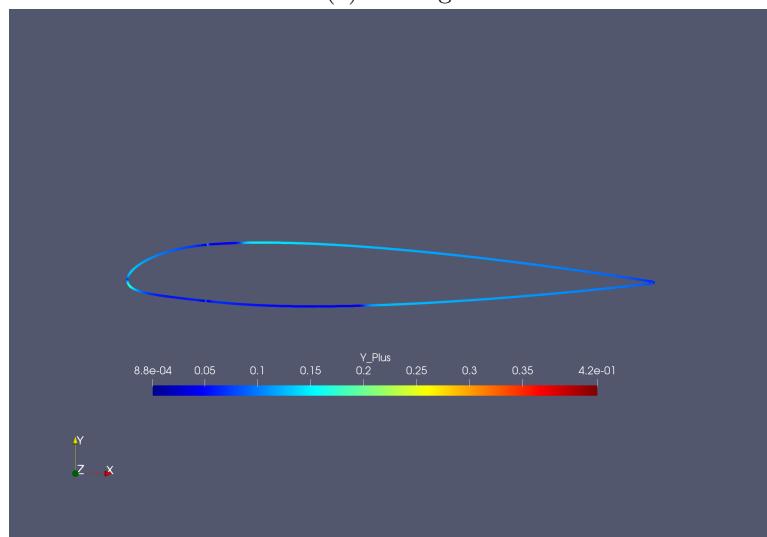


(b) 16 deg

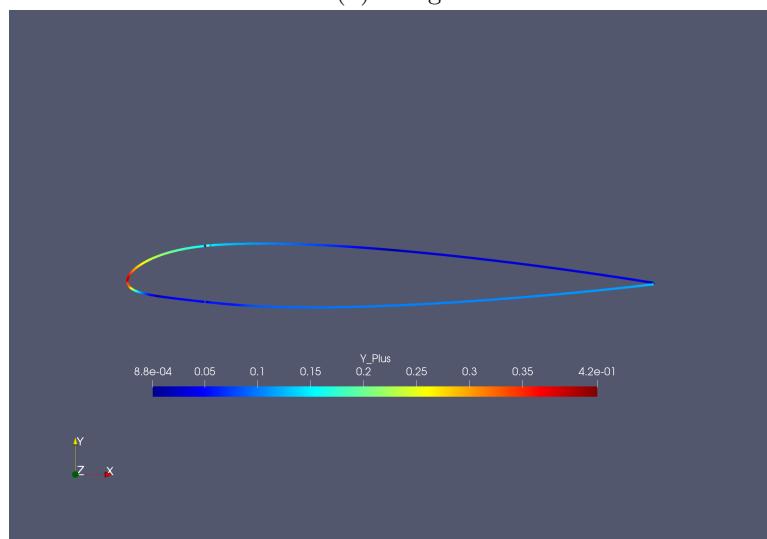
Figure B.2: Surface distributions of  $y^+$  NACA 0012 with BC.



(a) -14 deg



(b) 0 deg



(c) 17.5 deg

Figure B.3: Surface distributions of  $y^+$  NACA 23012.

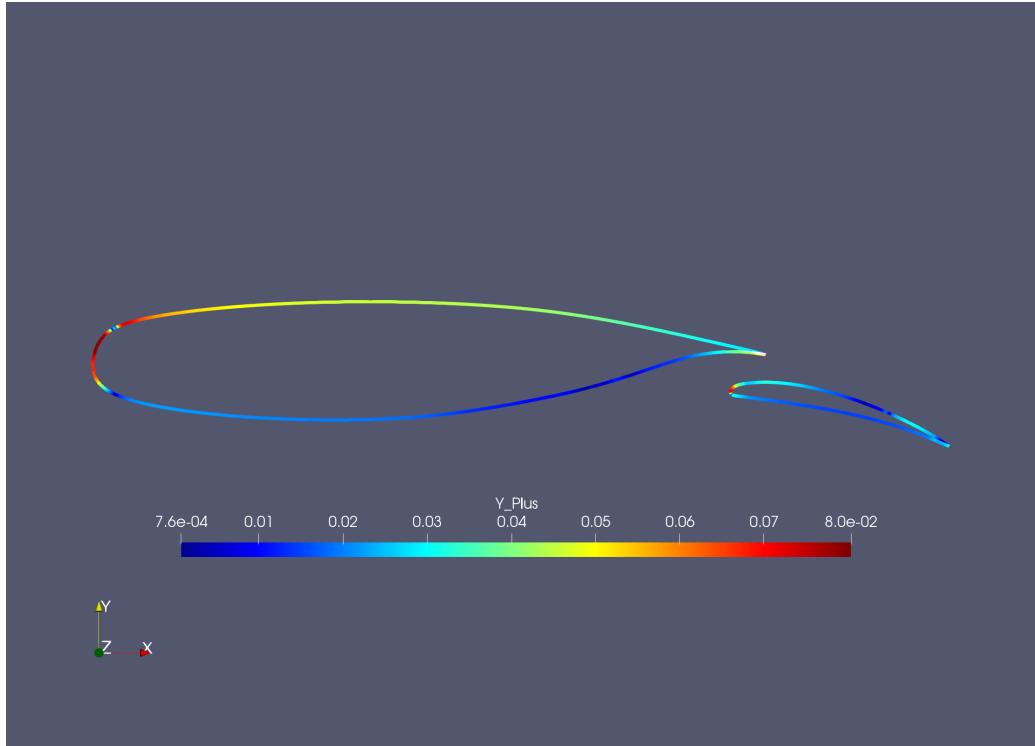


Figure B.4: Surface distribution of  $y^+$  NLR 7301.

Configuration	AoA [deg]	Input parameter $y_{plus}$	Computed CFD $y^+$
<b>NACA 0012 - FT</b>	<b>0</b>	0.20	0.1568
	<b>10</b>	0.20	0.3340
	<b>16</b>	0.20	0.4205
<b>NACA 0012 - BC</b>	<b>0</b>	0.20	0.1449
	<b>10</b>	0.20	0.2835
	<b>16</b>	0.20	0.3519
<b>NACA 23012</b>	<b>-14</b>	0.20	0.3536
	<b>0</b>	0.20	0.1656
	<b>17.5</b>	0.20	0.3991
<b>NLR 7301</b>	<b>(6, 20)</b>	(0.02, 0.02)	(0.0803, 0.0753)

Table B.1: Summary of  $y^+$  computed CFD results using tool inputs about NACA 0012 (both FT and using BC), NACA 23012 and NLR 7301.

The main observations are listed below:

- comparing FT and BC on NACA 0012 with same flow conditions, results show that FT has higher maximum  $y^+$  value.
- using the same reference for coloured band (NACA 0012 and NACA 23012), highest values of  $y^+$  are detected when AOA is close to stall conditions.
- related to previous point, input parameter  $y_{plus}$  seems a good prediction for

maximum actual  $y^+$  when small AoA are considered. On the other hand, underestimations occur close to stall, where actual  $y^+$  approximately doubles the input value.

- highest values of  $y^+$  are all located at LE, also for the cases at low AoA. Similarly,  $y^+$  values appear to increase in magnitude along accelerating flow surfaces.
- NLR 7301 has a very low peak of  $y^+$ , approximately equals to 0.08. Since the margin from unit value is significant, this CFD simulation could have been performed with less refining in structured region.
- lowest values of  $y^+$  are located where flows decelerates at on stagnation point.
- comparing NACA 0012 FT and BC, results are similar. However, when BC model is adopted the region of maximum values on LE is subdivided, presenting a lower value of  $y^+$  in the middle. Also the other geometries' results (NACA 23012 and NLR 7301, which were computed using BC) mention this trend.

About the latter point, a further observation can be made: considering NLR 7301, the x-position of these peaks approximately corresponds to the laminar bubble location detected by experimental data.

For example, on main element theoretical transition position is at  $x = 0.0252$  [45]. On the other hand, CFD results show the following values:

- $y^+$  value which is approximately 0.07-0.08 for  $x < 0.02$  and for  $x > 0.04$ , approximately.
- inside range (approximately)  $[0.02, 0.04]$ ,  $y^+$  drops. The minimum is at  $y^+ = 0.00349$  at  $x = 0.0289$ .

Based on these notes, further tool developments could consider methods for a correct detection of bubble locations or, by another point of view, improving grid structured region management and possibly increase computational speed.

# List of Symbols

$\alpha$	Angle of Attack
$\alpha_i$	Absolute angle of attack of $i$ -th element
$\alpha_0$	Main element angle of attack
$\alpha_{TE}$	Angle between camber line and chord line on trailing edge
$\beta_{TE}$	Trailing edge boat-tail angle
$\gamma$	Ratio of specific heat of a gas at a constant pressure to heat
$\gamma_{BC}$	Intermittency function in BC model
$\gamma - Re_\theta$	Menter et al. transition prediction model
$\theta$	Slope angle of an airfoil the mean line
$\lambda$	Volume viscosity coefficient
$\mu_\infty$	Freestream dynamic viscosity
$\nu_\infty$	Freestream cinematic viscosity
$\rho_\infty$	radius of leading edge
$\phi$	Total potential function
$\phi_\infty$	Potential function of freestream
$\phi_g$	PSO coefficient for velocity evaluation w.r.t. known optima position
$\phi_p$	PSO coefficient for velocity evaluation w.r.t. particle best known position
$\phi_s$	Potential function of sources distribution
$\phi_V$	Potential function of vortexes distribution
$\omega$	PSO coefficient for velocity evaluation w.r.t. previous iteration
$b_{x_c}$	Curvature of camber line on maximum camber location
$C$	Maximum camber
$C_D$	Drag coefficient
$C_{D_{max}}$	Maximum drag coefficient
$C_{D_{min}}$	Minimum drag coefficient
$C_F$	Friction coefficient
$C_L$	Lift coefficient
$C_{L_{max}}$	Maximum lift coefficient
$C_{L_{min}}$	Minimum lift coefficient

$c_i$	Chord of the $i$ -th element
$E$	Aerodynamic Efficiency ( $C_L/C_D$ )
$E^t$	Total energy of the flow
$e^N$	Transition prediction method, developed at TU Delft
$\underline{g}$	Best known position of the entire swarm in PSO
$K$	Thermal conductivity coefficient
$Ma$	Mach Number
$P$	Static pressure
$P_\infty$	Freestream static pressure
$\underline{p}_i$	Best known position of $i$ -th particle in PSO
$R_*$	Specific gas constant
$r_p$	Random term about particle's best known position for velocity evaluation
$r_g$	Random term about best swarm known position for velocity evaluation
$Re$	Reynolds Number
$Re_{\theta_c}$	Experimental transition onset critical momentum thickness Reynolds number
$Re_x$	Reynolds Number per unit chord
$s$	Wall spacing
$t$	Maximum thickness as chord's fraction of an airfoil
$T$	Maximum thickness
$T_\infty$	Mean freestream temperature
$Tu_\infty$	Freestream turbulence intensity
$U_\infty$	Freestream velocity
$u'_\infty$	Root-mean-square of freestream turbulent velocity fluctuations
$x$	X-coordinate in Cartesian coordinate system
$X_C$	Chord-wise location of maximum camber
$x_L$	Abscissa of a point located on lower surface of an airfoil
$X_T$	Chord-wise location of maximum thickness
$x_U$	Abscissa of a point located on upper surface of an airfoil
$y$	Y-coordinate in Cartesian coordinate system
$y^+$	Dimensionless wall distance
$y_c$	Y-coordinate of a specific airfoil's mean line
$y_L$	Ordinate of a point located on lower surface of an airfoil
$y_t$	Y-coordinate of a specific airfoil's thickness distributions
$y_U$	Ordinate of a point located on upper surface of an airfoil

# List of Abbreviations

<b>AoA</b>	Angle of Attack
<b>BB</b>	Box-Behnken design
<b>BC</b>	Bas-Cakmakcioglu transition model
<b>CFD</b>	Computational Fluid Dynamics
<b>DoE</b>	Design of Experiment
<b>FT</b>	Fully Turbulent boundary-layer
<b>GA</b>	Genetic Algorithm optimisation
<b>HL</b>	High-Lift
<b>HPC</b>	High Performance Computer
<b>IGP</b>	Improved Geometrical Parameterisation
<b>JST</b>	Jameson-Schmidt-Turkel scheme
<b>LE</b>	Leading Edge
<b>LHS</b>	Latin Hypercube Sampling
<b>MUSCL</b>	Monotonic Upstream-centred Scheme for Conservation Laws
<b>NACA</b>	National Advisory Committee for Aeronautics
<b>NASA</b>	National Aeronautics and Space Administration
<b>PSO</b>	Particle Swarm Optimisation
<b>RANS</b>	Reynolds-Averaged Navier-Stokes
<b>SA</b>	Spalart–Allmaras turbulence model
<b>SA-BC</b>	Spalart-Allmaras and Bas-Cakmakcioglu turbulence-transition model
<b>TE</b>	Trailing Edge
<b>TU</b>	Turbulence Intensity
<b>TVD</b>	Total Variation Diminishing method
<b>SU2</b>	Stanford University Unstructured



# References

- [1] Ira H. Abbott and Albert E. Von Doenhoff. *Theory of Wing Sections*. Dover Publications Inc., 1958.
- [2] E. Benzing. *Ali-Wings: Their Design and Application to Racing Cars*. Giorgio Nada Editore s.r.l., 2012.
- [3] E.L. Houghton, P.W. Carpenter, Steven H. Collicott, and Daniel T. Valentine. *Aerodynamics for Engineering Students*. Elsevier, 2013.
- [4] T. Cebeci, J. Mosinskis, and M.O. Smith. Calculation of separation points in incompressible turbulent flows. *J. Aircraft*, 9(9):618–624, 1972.
- [5] A. Nakayama, H.P. Kreplin, and H.L. Morgan. Experimental investigation of flowfield about a multi-element airfoil. *AIAA Journal*, 28(1):14–21, 1990.
- [6] A.M.O. Smith. High-lift aerodynamics. *J. Aircraft*, 12(6):501–530, June 1975.
- [7] D. Landman. *Experimental Geometry Optimization Techniques for Multi-Element Airfoils*. Old Dominion University, 1998.
- [8] W.O. Valarezo and V.D. Chi. Method for the prediction of wing maximum lift. *Journal of Aircraft*, 31(1):103–109, Jan.-Feb. 1994.
- [9] W.H. Wentz Jr. and H.C. Seetharam. Development of a fowler flap system for a high performance general aviation airfoil. Technical report, National Aeronautics and Space Administration, Washington D.C., December 1974.
- [10] Z. Lei, M. Murayama, K. Takenaka, and K. Yamamoto. Cfd validation for high-lift devices: Two-element airfoil. *Trans. Japan Soc. Aero. Space Sci.*, 49(163):31–39, 2006.
- [11] S. Kim, J.J. Alonso, and A. Jameson. Multi-element high-lift configuration design optimization using viscous continuous adjoint method. *Journal of Aircraft*, 41(5):1082–1097, Sept.-Oct. 2004.
- [12] S.M. Klausmeyer and J.C. Lin. Comparative results from a cfd challange over a 2d three-element high-high lift airfoil. *NASA Technical Memorandum 112858, Langley Research Center*, May 1997.
- [13] J. Park, N. Van Nguyen, M. Kim, S. Tyan, and J.W. Lee. Flap design optimization for very light aircraft in compliance with airworthiness certification. In *Fluid Dynamics and Co-located Conferences - 21st AIAA Computational Fluid Dynamics Conference*, July 2013.

- [14] D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons Inc., 2017.
- [15] A. Arias-Montano, C.A. Coello, and E. Mezura-Montes. Multi-objective airfoil shape optimization using a multiple-surrogate approach. *2012 IEEE Congress on Evolutionary Computation*, 2012.
- [16] G. Trapani, T. Kipouros, and M. Savill. Multi-objective optimization of 2d high-lift airfoil configurations using tabu search. In *AIAA Student Conference*, 2010.
- [17] G. Trapani, T. Kipouros, M. Savill, C. Agostinelli, and A. Rampurawala. Aerodynamic optimization of a flexible high-lift wing. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, 2013.
- [18] G. Trapani, T. Kipouros, M. Savill, C. Agostinelli, and A. Rampurawala. A rapid approach to the aerodynamic design of a flexible high-lift wing. In *AIAA Aviation*, 2014.
- [19] P. Iannelli, F. Moens, M. Minervino, R. Ponza, and E. Benini. Comparison of optimization strategies for high-lift design. *Journal of Aircraft*, 54(2):642–680, 2017.
- [20] J. Wild, J. Brezillon, O. Amoignon, J. Quest, F. Moens, and D. Quagliarella. Advanced high-lift design by numerical methods and wind tunnel verification within the european project eurolift ii. In *25th AIAA Applied Aerodynamics Conference*, June 2007.
- [21] Lu Xiaoqiang, Huang Jun, Song Lei, and Li Jing. An improved geometric parameter airfoil parameterization method. *Aerospace Science and Technology* 78, page 241–247, 2018.
- [22] E.P. Scott and R.T. Haftka. Optimization and experiments. *AIAA 95-0121*, 1995.
- [23] J.D. Jr. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill Education, 2011.
- [24] P. Marzocca. The naca airfoil series. Technical report, Clarkson University, 2009.
- [25] E.N. Jacobs and R.M. Pinkerton. Report no. 537, tests in the variable-density wind tunnel of related airfoils having the maximum camber unusually far forward. Technical report, NACA Technical Reports, 1935.
- [26] R.M. Cummings, W.H. Mason, and D.R. Morton, S.A. McDaniel. *Applied Computational Aerodynamics: a Modern Engineering Approach*. Cambridge: Cambridge University Press, 2015.
- [27] Naca airfoil. [https://nasa.fandom.com/wiki/NACA\\_airfoil](https://nasa.fandom.com/wiki/NACA_airfoil). Accessed: 2021-05-28.
- [28] E.N. Jacobs, K.E. Ward, and R.M. Pinkerton. Naca report no. 460, the characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Technical report, NACA Technical Reports, 1933.
- [29] University of Illinois Urbana-Champaign Department of Aerospace Engineering. Uiuc applied aerodynamics group. [https://m-selig.ae.illinois.edu/ads/coord\\_database.html](https://m-selig.ae.illinois.edu/ads/coord_database.html). Accessed: 2021-05-29.

- [30] J.R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [31] O. Rübenkönig, Z. Liu, and J. Linemann. The finite volume method (fvm) - an introduction. [https://web.archive.org/web/20091002233707/http://www.imtek.uni-freiburg.de/simulation/mathematica/IMSweb/imsTOC/Lectures%20and%20Tips/Simulation%20I/FVM\\_introDocu.html](https://web.archive.org/web/20091002233707/http://www.imtek.uni-freiburg.de/simulation/mathematica/IMSweb/imsTOC/Lectures%20and%20Tips/Simulation%20I/FVM_introDocu.html). 2009-10-02.
- [32] Thomas D. Economou. Compressible flow. <https://su2code.github.io/tutorials/>. Accessed: 2021-05-18.
- [33] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale (originally presented at AIAA 30th Aerospace Sciences Meeting and Exhibit - 1992)*, 1:5–21, 1994.
- [34] J. Boussinesq. *Theorie Analytique de la Chaleur mise en harmonie avec la Thermodynamique et avec la Theorie Mecanique de la Lumiere*. Cours de physique mathematique - Tome II - Refroidissement et echauffement par Rayonnement conductibilite des tiges, lames et masses cristallines courants de convection theorie Mecanique de la Lumiere, 1903.
- [35] M.S. Uberoi. Effect of grid geometry on turbulence decay. *The Physics of Fluids*, 10(6), 1967.
- [36] L. Prandtl. *Bericht über Untersuchungen zur ausgebildeten Turbulenz*. Zs. Angew. Math. Mech. 2, 1925.
- [37] S.R. Allmaras, F.T. Johnson, and P.R. Spalart. Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In *ICCFD7-1902, 7th International Conference on Computational Fluid Dynamics*, 2012.
- [38] J.R. Edwards and S. Chandra. Comparison of eddy viscosity-transport turbulence models for three-dimensional, shock-separated flowfields. *AIAA Journal*, 34(4):756–763, 1996.
- [39] P.R. Spalart. Trends in turbulence treatments. *AIAA 2000-2306*, June 2000.
- [40] U. Kaynak, O. Bas, and S.C. Cakmakcioglu. A correlation-based algebraic transition model. *Journal of Mechanical Engineering Science*, 232(21):3915–3929, 2018.
- [41] U. Kaynak, O. Bas, S.C. Cakmakcioglu, and I.H. Tuncer. *Transition Modeling for Low to High Speed Boundary Layer Flows with CFD Applications*. IntechOpen, 2019.
- [42] The spalart-allmaras 1-equation bc transitional model. [https://turbmodels.larc.nasa.gov/sa-bc\\_1eqn.html](https://turbmodels.larc.nasa.gov/sa-bc_1eqn.html). Accessed: 2021-06-13.
- [43] M. Drela. Mises implementation of modified abu-ghannam and shaw transition criteria. *MISES Code Documentation*, 1995.
- [44] F.R. Menter, R.B. Langtry, S.R. Likki, Y.B. Suzen, P.G. Huang, and S. Völker. A correlation-based transitionmodel using local variables — part i: Model formulation. *Journal of Turbomachinery*, 126:413–422, July 2006.

- [45] B. Van der Berg and J.H.M. Gooden. Low-speed surface pressure and boundary layer measurement data for the nlr 7301 airfoil section with trailing edge flap. *AGARD Advisory Report*, 303(2):A9.1–A9.12, August 1994.
- [46] J. Park, N. Van Nguyen, M. Kim, S. Tyan, and J.W. Lee. The muscle code for compressible flow: Philosophy and results. In *2nd International Conference on Computational Methods in Nonlinear Mechanics*, Jan-1979.
- [47] B. Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, July 1979.
- [48] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the euler equations by finite volume methods using runge-kutta time stepping schemes. *AIAA Journal, American Institute of Aeronautics and Astronautics*, 1981.
- [49] C. Geuzaine and J.F. Remacle. A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <https://gmsh.info/doc/texinfo/gmsh.html>. Accessed: 2021-05-18.
- [50] S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm. *J. Comput. Phys.*, 106:25–138, May 1993.
- [51] S. Narsipur, W.P. Brent, and M.S. Selig. Cfd analysis of multielement airfoils for wind turbines. In *AIAA 2012-2781*, 2012.
- [52] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [53] H. Schlichting. *Boundary-Layer Theory*. McGraw Hill, New York, U.S.A., 7th edition, 1979.
- [54] C.F. Van Loan. Using the ellipse to fit and enclose data points, a first look at scientific computing and numerical optimization. Technical report, Department of Computer Science, Cornell University.
- [55] J. Abramson. *College Algebra*. LibreTexts, OpenStax, 2015.
- [56] C.L. Ladson. Effects of independent variation of mach and reynolds numbers on the low-speed aerodynamic characteristics of the naca 0012 airfoil section. Technical report, NASA Technical Memorandum 4074, 1988.
- [57] NASA Turbulence Modeling Resources. 2dn00: 2d naca 0012 airfoil validation case. [https://turbmodels.larc.nasa.gov/naca0012\\_val.html](https://turbmodels.larc.nasa.gov/naca0012_val.html). Accessed: 2021-06-06.
- [58] S. Kubacki and E. Dick. A revised one-equation transitional model for external aerodynamics - part i: Theory, validation and base cases. *AIAA 2020-2714, Applied CFD Modeling and Numerical Corrections with Experimental Data II*, 2020.
- [59] V. Popov, M. Labib, and R.M. Botez. Method for the transition point detection method from the pressure coefficients distribution for controller design. *Canadian Aeronautics and Space Institute Annual General Meeting Aircraft Design and Development Symposium*, 2007.

- [60] M. Islam. Modeling of boundary layer transition. Technical report, Ahsanullah University of Science and Tech., May 2015.
- [61] S.G.G. Beveridge and R.S. Schechter. *Optimization: Theory and Practice*. McGraw-Hill Book Company, 1970.
- [62] R.T. Haftka and Z. Gürdal. *Elements of Structural Optimization*. Kluwer Academic Publishers, 3rd ed. edition, 1992.
- [63] D. Landman and C.P. Britcher. Experimental optimization methods for multi-element airfoils. *American Institute of Aeronautics and Astronautics*, October 1996.
- [64] J.C. Otto, D. Landman, and A.T. Patera. A surrogate approach to the experimental optimization of multi-element airfoils. *AIAA 96-4138-CP*, 1996.
- [65] J.C. Otto, M. Paraschivoiu, S. Yesilyurt, and A.T. Patera. Bayesian-validated computer-simulation surrogates for optimization and design: Error estimates and applications. *Mathematics and Computers in Simulation*, 44:347–367, 1997.
- [66] S. Yesilyurt and A.T. Patera. Surrogates for numerical simulations; optimization of eddy-promoter heat exchangers. *Computer Methods in Applied Mechanics and Engineering*, 121:231–257, 1995.
- [67] J. Kennedy and R. Eberhart. Particle swarm optimisation. *Proceedings of ICNN'95 - International Conference on Neural Networks*, pages 1942–1948, 1995.
- [68] J. Kennedy, R. Eberhart, and Y. Shi. A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, 1998.
- [69] J. Krottmaier. *Optimizing Engineering Designs*. McGraw-Hill, March 1993.
- [70] R.L. Fox. *Optimizing Methods for Engineering Design*. Addison Wesley, 1971.
- [71] S.S. Petrova and A.D. Solov'ev. The origin of the method of steepest descent. *Historia Mathematica, ARTICLE NO. HM962146*, 24:361–375, 1997.
- [72] S.L.S. Jacoby, J.S. Kowalik, and J.T. Pizzo. *Iterative Methods for Nonlinear Optimization Problems*. Prentice-Hall, 1972.
- [73] G.B. Semones and H.C. Lim. Experimental multivariate adaptive optimization of the steady-state cellular productivity of a continuous baker's yeast culture. *Biotechnology and Bioengineering*, 33:16–25, 1989.
- [74] A. Shenk. *Calculus and Analytic Geometry*. Goodyear Publishing Company, 1979.
- [75] R. Poli. An analysis of publications on particle swarm optimisation applications. Technical report, Technical Report CSM-469, 2007.
- [76] R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, pages 1–10, 2008.
- [77] M. Clerc. Standard particle swarm optimisation. Technical report, HAL Open Access Archive, hal-00764996, 2012.

- [78] M. Taherkhani and R. Safabakhsh. A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, pages 281–295, 2015.
- [79] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi. Fuzzy self-tuning pso: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*, 39:70–85, 2018.
- [80] M. Meissner, M. Schmuker, and G. Schneider. Optimized particle swarm optimization (opso) and its application to artificial neural network training. *BMC Bioinformatics*, 7:70–85, 2006.
- [81] D. Bratton and J. J. Kennedy. Defining a standard for particle swarm optimization. *IEEE Swarm Intelligence Symposium*, doi: 10.1109/SIS.2007.368035, pages 120–127, 2007.
- [82] A.I. Khuri and J.A. Cornell. *Response Surfaces: Designs and Analyses*. CRC Press, Taylor and Francis Group, 1996.
- [83] R.H. Myers, D.C. Montgomery, and C.M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 2016.
- [84] M.D. McKay, W. Conover, and R. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(1):239–245, 1979.
- [85] R.L. Iman, J.M. Davenport, and D.K. Zeigler. *Latin hypercube sampling (program user's guide). [LHC, in FORTRAN]*. Technical Information Center Oak Ridge Tennessee, 1980.
- [86] R.L. Iman, J.C. Helton, and J.E. Campbell. An approach to sensitivity analysis of computer models: Part ii - ranking of input variables, response surface validation, distribution effect and technique synopsis. *The Journal of Quality Technology*, 13(4):232–240, 1981.
- [87] pydoe: Design of experiments for python. <https://pythonhosted.org/pyDOE/index.html>. Accessed: 2021-05-30.
- [88] J.L. Hess and A.M.O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8:1–138, 1967.
- [89] J.L. Hess. Aerodynamic characteristics of airfoil using panel method. *Cruse T.A. (eds) Advanced Boundary Element Methods. International Union of Theoretical and Applied Mechanics*, pages 165–177, 1988.
- [90] J.L. Hess. Panel methods in computational fluid dynamics. *Annual Rev. Fluid Mech.*, 22:255–274, 1990.
- [91] B. Junaidin, P.T. Penerbangan, and S.T.T. Adisutjipto. Aerodynamic characteristics of airfoil using panel method. *Jurnal Angkasa*, 2016.
- [92] T.D. Economon, F. Palacios, S.R. Copeland, T.W. Lukaczyk, and J.J. Alonso. Su2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2015.