UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

Máster Universitario en Ingeniería Aeronáutica

Master Thesis

2D profile multi-fidelity optimization and algorithm comparison

Author: **Javier Berrueco Fernández**
Supervisors: **Jorge Izquierdo Yerón and Giovanni Bailardi**

Madrid, September 2023

Título: 2D profile multi-fidelity optimization and algorithm comparison

Autor: Javier Berrueco Fernández

Tutor: Jorge Izquierdo Yerón and Giovanni Bailardi

Titulación: Máster Universitario en Ingeniería Aeronáutica

Curso: 2022-2023

Abstract

This master's thesis focuses on optimizing foil shapes, which play a crucial role in determining their aerodynamic or hydrodynamic performance. The objective is to evaluate different techniques and methods in terms of efficiency and accuracy, to this end, a computer program has been developed. Key topics include geometry parameterization methods, Computational Fluid Dynamics (CFD) programs like XFoil and Star-CCM+, and optimization and sampling strategies using Dakota. The main contributions of this work are to provide new insights and methods for future research in aerodynamic or hydrodynamic design. Foil optimization is widely studied in various industries, including wind energy, naval, and aerospace. The optimization objective is to optimize a boat keel profile, aiming to reduce drag and achieve a certain lift coefficient.

The research found that the PARSEC optimum obtained through the SOGA optimization method performs best for the specified study case. The optimum foils exhibit similar characteristics, with thin leading and trailing edges and a maximum thickness between 0.3 and 0.5 of the chord length. The PARSEC parameterization method offers advantages over NACA formulations, providing greater control over the geometry which turns out in better profiles from the hydrodynamic performance point of view.

The overall performance of the optimization process demonstrates the efficiency of using a low-fidelity flow solver (XFoil) for optimization and verifying the results with a high-fidelity CFD solver (Star-CCM+). This methodology is a powerful approach for optimizing the aerodynamic or hydrodynamic performance of designs. Future research could explore the use of AI models to improve the accuracy of low-fidelity solvers and train models using the optimization results obtained in this study.

Keywords: aerodynamics, optimization, foil shape parameterization, computational fluid dynamics, CFD, optimization algorithms, low-fidelity solver, XFoil, high-fidelity solver, Star-CCM+, evolutionary strategies, genetic algorithm, Newton method, method of feasible directions.

Resumen

Esta tesis de máster se centra en la optimización de perfiles en 2D, lo que es crucial a la hora de determinar su rendimiento aerodinámico o hidrodinámico. El objetivo es optimizar los perfiles utilizando varios algoritmos y evaluando diferentes técnicas y métodos en términos de eficiencia y precisión, para lo cual se ha desarrollado una aplicación específica. Los temas clave tratados en este trabajo incluyen los métodos de parametrización de la geometría, programas de dinámica de fluidos computacional (CFD) como XFoil y Star-CCM+, y estrategias de optimización y muestreo utilizados mediante Dakota. Las principales aportaciones de este trabajo son proporcionar nuevas perspectivas y métodos para futuras investigaciones en diseño aerodinámico o hidrodinámico. La optimización de perfiles se estudia ampliamente en diversas industrias, como la energía eólica, la naval y la aeroespacial. Para estudiar los resultado se ha centrado el proyecto en la optimización de una quilla de velero, con el objetivo de reducir su resistencia manteniendo un determinado coeficiente de fuerza lateral.

El estudio descubrió que el óptimo utilizando la parametrización de la geometría PARSEC obtenido mediante el método de optimización SOGA es el que mejor se comporta para el caso de estudio especificado. Los perfiles óptimos presentan características similares, con bordes de ataque y salida delgados y un espesor máximo entre 0,3 y 0,5 de la longitud de cuerda. El método de parametrización PARSEC ofrece ventajas sobre las formulaciones NACA, ya que proporciona un mayor control sobre la geometría, lo que resulta en mejores perfiles desde el punto de vista hidrodinámico en este caso.

El rendimiento global del proceso de optimización demuestra la eficacia de utilizar un solver de baja fidelidad (XFoil) para la optimización y verificar los resultados con un solver CFD de alta fidelidad (Star-CCM+). Esta metodología es un potente enfoque para optimizar el rendimiento aerodinámico o hidrodinámico de los diseños. Futuras investigaciones podrían explorar el uso de modelos de IA para mejorar la precisión de los programas de baja fidelidad y entrenar modelos utilizando los resultados de optimización obtenidos en este estudio.

Keywords: aerodinámica, optimización, parametrización de perfiles, dinámica de fluidos computacional, CFD, DFC, algoritmos de optimización, XFoil, Star-CCM+, estrategias evolutivas, algoritmo genético, método de Newton, método de direcciones factibles.

Contents

List of Figures 5

List of Tables 7

List of Equations 8

Nomenclature 9

1 Introduction 11
1.1 Motivation . . . 11
1.2 Optimization objective . . . 12

2 Geometry parameterization 14
2.1 NACA formulation . . . 14
2.2 PARSEC formulation . . . 19
2.3 Geometry parameterization set-up . . . 21

3 Flow solvers 25
3.1 XFoil . . . 25
3.2 CFD and Star-CCM+ . . . 27
3.3 Solver set-up . . . 28

4 Optimization Methods 34
4.1 Sampling methods . . . 34
4.2 Optimization formulation . . . 35
4.3 Gradient methods . . . 36
4.4 Genetic or Evolutionary Algorithms . . . 37
4.5 Optimization set-up . . . 39

5 Initial design of experiments 40
5.1 Initial sampling results . . . 40
5.2 Gradient methods initial results . . . 41
5.3 GA and EA initial results . . . 43

6 Results 46
6.1 Gradient methods optimization . . . 46
6.2 Evolutionary algorithms . . . 48
6.3 Comparison with Star-CCM+ results . . . 52

7 Conclusions 58
7.1 Future works . . . 60

8 References 61

9 Annex 64
9.1 Scatter matrices and parallel coordinates graphics . . . 65
9.2 Typical Star-CCM+ simulation results . . . 72

List of Figures

1 Keel operation [22]. . . . 13
2 PARSEC parameters definition [30]. . . . 20
3 PARSEC foil with 50 points on each surface and linear spacing. . . . 22
4 PARSEC foil with 100 points on each surface and cosine spacing. . . . 22
5 PARSEC impossible geometry I. . . . 23
6 PARSEC impossible geometry II. . . . 23
7 PARSEC undesired geometry. . . . 24
8 XFoil terminal. . . . 25
9 C_p distribution of a foil produced with the panelization as described in its coordinates file. . . . 29
10 Mesh I, zoomed out. . . . 30
11 Mesh II, zooming in. . . . 30
12 Mesh III: near profile. . . . 31
13 Mesh IV: detail of LE. . . . 31
14 Mesh V: detail of TE. . . . 32
15 Pressure distribution comparison: simulation on Star-CCM+ with $k-\epsilon$ turbulence vs. experimental data from Gregory at AoA 0. . . . 33
16 Pressure distribution comparison: simulation on Star-CCM+ with $k-\epsilon$ turbulence vs. experimental data from Gregory at AoA 10. . . . 33
17 Example multidimensional parameter study [27]. . . . 34
18 Gradient methods behaviour at the presence of local minimums [33]. . . . 36
19 Description of an EA [28]. . . . 38
20 Steps and operators of a GA [9]. . . . 38
21 Initial geometry for the gradient-based optimizations with PARSEC. . . . 41
22 PARSEC geometry that almost complies with the lift constraint and has minimum drag. . . . 44
23 C_L vs. evaluation, color by C_D, size by AoA of the CONMIN MFD optimization with PARSEC. . . . 46
24 Optimum geometry of the CONMIN MFD optimization with PARSEC. . . . 47
25 C_L vs. evaluation, color by C_D, size by AoA of the OPT++ Newton optimization with PARSEC. . . . 47
26 Optimum geometry of the OPT++ Newton optimization with PARSEC. . . . 48
27 C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with NACA 4-m. . . . 49
28 Optimum of NACA 4-m optimization. . . . 49
29 C_L vs. evaluation, colour by C_D, size by AoA of the COLINY EA optimization with NACA 6. . . . 50
30 C_L vs. NACA 6 series, colour by C_D, size by AoA of the optimization. . . . 50
31 Optimum of NACA 6 optimization. . . . 50
32 C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with PARSEC. . . . 51
33 Optimum of PARSEC optimization. . . . 51
34 Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in PARSEC geometries. . . . 53

35 C_L vs. C_D of XFoil results and their respective Star-CCM+ results of PARSEC geometries. . . . 54
36 Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in NACA 4m geometries. . . . 54
37 C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 4m geometries. . . . 55
38 Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in NACA 6 geometries. . . . 55
39 C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 6 geometries. . . . 56
40 New PARSEC optimum taking into account Star-CCM+ results. . . . 57
41 New NACA 6 optimum taking into account Star-CCM+ results. . . . 57
42 C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with NACA 4-m. . . . 59
43 C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 4m geometries. . . . 59
44 Final PARSEC optimum taking into account the multi-fidelity results. . . . 60
45 Scatter matrix of an LHS using PARSEC. . . . 65
46 Parallel coordinates of an LHS using PARSEC. . . . 66
47 Scatter matrix of PARSEC eight parameters plus AoA. . . . 67
48 Parallel coordinates of PARSEC eight parameters plus AoA. . . . 68
49 Scatter matrix of NACA 4-m optimization. . . . 69
50 Scatter matrix of NACA 6 optimization. . . . 70
51 Scatter matrix of PARSEC optimization using SOGA. . . . 71
52 Residuals of a typical Star-CCM+ simulation of this work. . . . 72
53 Drag coefficient of a Star-CCM+ simulation of this work. . . . 72
54 Lift coefficient of a Star-CCM+ simulation of this work. . . . 73
55 Y+ of a Star-CCM+ simulation of this work. . . . 73
56 Pressure coefficient distribution comparison between XFoil and Star-CCM+ of a Star-CCM+ simulation of this work. . . . 73
57 Pressure scalar scene of a Star-CCM+ simulation of this work. . . . 74
58 Velocity scalar scene of a Star-CCM+ simulation of this work. . . . 74

List of Tables

1 PARSEC parameters definition [10]. 21
2 PARSEC parameters limits in the samplings and optimizations. 24
3 Comparison with simulation results using different turbulence models and experimental data from Ladson. 32
4 Comparison between all the optimums from the different optimizations. 52
5 Comparison between differences between high-fidelity results and low-fidelity for each geometry parameterization method. 56
6 Comparison between all the optimums from the different optimizations, Star-CCM+ results. 56
7 Optimums from the different optimizations taking into account Star-CCM+ results. 57
8 Final optimums from the different optimizations taking into account the multi-fidelity results. 60

List of Equations

1 NACA foils coordinates equations . . . 14
2 NACA 4 thickness distribution . . . 14
3 NACA 4 thickness leading edge radius . . . 15
4 NACA 4 trailing edge angle . . . 15
5 NACA 4 foils camber line equations I . . . 15
6 NACA 4 foils camber line equations II . . . 15
7 NACA 5 foils camber line equations I . . . 16
8 NACA 5 foils camber line equations II . . . 16
9 NACA 5 foils maximum camber position . . . 16
10 NACA 5 K1 . . . 16
11 NACA 5 Q . . . 16
12 NACA 4M thickness distribution I . . . 17
13 NACA 4M thickness distribution II . . . 17
14 NACA 6 camber line equation I . . . 18
15 NACA 6 camber line equation II . . . 18
16 NACA 6 camber line equation III . . . 19
17 NACA 6 g and h . . . 19
18 NACA 6 camber line slope . . . 19
19 NACA 6A camber line linear equation . . . 19
20 NACA 6A camber line linear equation slope . . . 19
21 PARSEC foils surfaces coordinates . . . 20
22 PARSEC a_n equations . . . 20
23 PARSEC C matrices . . . 20
24 PARSEC b matrices . . . 20
25 Mass conservation equation in RANS form . . . 27
26 Momentum equation in RANS form . . . 28
27 General optimization formulation . . . 35
28 Direction finding subproblem in a typical feasible direction method . . . 37
29 Classic Newton direction finding subproblem . . . 37
30 Difference in percentage between the force Star-CCM+ coefficients and the XFoil coefficients . . . 52

Nomenclature

AoA Angle of Attack

AI Artificial Intelligence

BL Boundary Layer

c Chord

CAE Computational Aided Engineering

CFD Computational Fluid Dynamics

C_{l_i} Design lift coefficient

C_D Drag coefficient

EA Evolutionary Algorithm

ES Evolutionary Strategies

FVM Finite Volume Method

GA Genetic Algorithm

HF or hf High fidelity

LHS Latin Hypercube Sampling

LE Leading Edge

C_L Lift coefficient

LP Linear Programming

LF or lf Low fidelity

M Mach number

MHz Megahertz

MFD Method of Feasible Directions

m Metres

NACA National Advisory Committee for Aeronautics

NLP Nonlinear Programming

C_p Pressure coefficient

PDAS Public Domain Aeronautical Software

Re Reynolds number

RANS Reynolds-Averaged Navier-Stokes

s Seconds

SOGA Single-objective Genetic Algorithm

TE Trailing Edge

UQ Uncertainty Quantification

1. Introduction

Foil optimization is a crucial aspect of engineering as the shape of a foil plays a vital role in determining its aerodynamic/hydrodynamic performance. The advancements in data management and computing offer the potential for significant improvements in aerodynamic/hydrodynamic efficiency optimization. This master's thesis aims to investigate the use of several methodologies for 2D shape optimization of foils.

The main objective of this thesis is to applicate different techniques to build a tool to optimize foil shapes. Different techniques and methods for foil optimization have been evaluated in terms of efficiency and accuracy. The research addresses to demonstrate the application of these optimizations to optimize the profiles for specific design constraints and goals, such as reducing drag while achieving a certain lift.

The thesis focuses on several key topics, including geometry parameterization methods, the usage of different Computational Fluid Dynamics (CFD) solvers, XFoil and Star-CCM+, and optimization and sampling strategies using Dakota. The main contributions of this thesis will be to provide new insights and methods for future research in the field of aerodynamic or hydrodynamic design.

1.1. Motivation

The purpose of completing this project is the application of several algorithms in foil optimization. For this reason, multiple research about airfoil optimization have been reviewed and it has been reflected how this work could change the current design procedures.

Foil shape optimization is a common study field [10] in different applications such as the wind energy, naval and aerospace industries. Even free tools have been developed that allow the user to perform a local foil shape optimization[1].

We can distinguish three or four sections in a typical foil optimization depending on the solver that is going to be used: shape parameterization/control; mesh creation/deformation, depending on the CFD solver; flow solution; and optimisation.

Shape parameterization in an optimization problem means how the geometry will be managed by the optimization algorithm. In a global optimization, we will need to cover a wide range of foils whereas, in a local optimization changing a little some of the parameters, a good result could be achieved. Previous research concludes that to cover a large database of foils, between 20 to 25 design variables are needed to cover the 90% and between 13 and 17 to cover 80 % [19]. The best parameterization method is efficient in terms of fewer variables changing significantly the airfoil shape [2].

When selecting the solver we must bear in mind the operating conditions we would like to simulate. Some conditions could not be addressed with potential flow solvers, like XFoil. Nevertheless, coupling a high-fidelity CFD solver will result in a more computationally expensive tool [10]. The optimization first approach with simplified flow solvers and the following validation of these low-fidelity results with a high-fidelity commercial, RANS or viscous CFD is currently being used in the industry [5]. The usage of CFD is mainly used at the final stages of the design process, once the problem and geometry are well defined.

[1] http://webfoil.engin.umich.edu/

Last but not least, the choice of an optimization method depends mainly on the knowledge of the objective function and the optimization landscape [17]. We must take into account that to solve the optimization problem, we will need to call the model several times, which can be computationally demanding.

To achieve the objective of this worl, the following milestones must be achieved:

- Geometry generation. Different types of geometry parameterization methods must be used to generate the foils.
- Flow solver. A fast solver (XFoil) must be coupled to the geometry generator to evaluate the different profiles.
- Optimization algorithm. Different methods must govern the geometry parameters and analyse the results to obtain an optimum foil under a given set of constraints.
- Verify and control results. High fidelity flow solver (Star-CCM+) is to be used to control the quality of the optimization results.

Every step of the process will be evaluated: from geometry parameterization to optimization methods performance, with a focus on identifying the most efficient and effective approach. By the end of this thesis, we expect to have developed a simple tool capable of automating the optimization process and have compared its performance to achieve the best results.

Future research that could be done is to implement other parameterization options, use other optimization algorithms or build an AI-driven flow solver trained with the results obtained in the optimization or design space studies performed in the development of this research.

1.2. Optimization objective

The objective of the final optimization will be to optimize a boat keel profile, reducing drag and aiming to obtain a lift coefficient bigger than 0.4. Optimizing a boat keel means optimizing a symmetrical foil. This helps us to achieve better optimization results faster as there are fewer design variables, avoiding the curse of dimensionality[2]. In addition, it is easier to learn how the optimization methods work. Nevertheless, in the results, it is detailed how the algorithm performed with a non-symmetrical foil.

Once the objective of the optimization has been chosen, we have defined our flow conditions. This is a turbulent and incompressible flow (Re = 1e+06).

The keel functionality is to provide the boat with enough lateral force to compensate for the torque of the wind force generated by the sails. For this reason, the angle of attack is also a design variable since having symmetrical profiles, it is needed to generate that torque. This problem is detailed in figure 1.

[2]The number of samples required to estimate any function with a given degree of accuracy increases exponentially as the number of input variables of the function increases, according to Bellman's [3] theory of the curse of dimensionality[7].

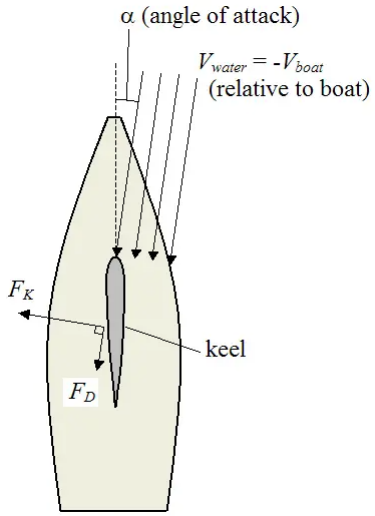

Figure 1: Keel operation [22].

Nevertheless, an initial design of experiments included non-symmetrical foil optimization.

2. Geometry parameterization

Previous research has shown the influence of geometry parameterization in the optimization process [24]. Multiple methods of foil generation have been developed up to now.

NACA formulation is probably one of the most studied parameterization methods but is not the only one. Other parameterization techniques vary from geometry representation techniques like B-Splines to foil-specific methodologies such as PARSEC. We can distinguish between constructive and deformative methods. Constructive methods build the foil surface by attending only to the geometrical features. Deformative methods obtain the foils by modifying an existing foil [19].

To develop this thesis, NACA and PARSEC formulations have been used, for this reason, they are detailed below.

2.1. NACA formulation

From 1929 to 1947, the NACA airfoils were created at the Langley Field Laboratory under the supervision of Eastman Jacobs. These foils were primarily constructed using basic geometric outlines to describe their section shape. However, the 6 and 6A series were produced through theoretical analysis and lack straightforward shape definitions The process of creating NACA foils involves the combination of a thickness envelope and a camber or mean line. This section is mainly based on the work presented by Mason [18]. The equations that detail this process are as follows:

$$\begin{aligned} x_u &= x - y_t(x)\sin\theta \\ y_u &= y_c(x) + y_t(x)\cos\theta \\ x_l &= x + y_t(x)\sin\theta \\ y_l &= y_c(x) - y_t(x)\cos\theta \\ \theta &= \tan^{-1}\left(\frac{dy_C}{dx}\right) \end{aligned} \tag{1}$$

where $y_t(x)$ is the thickness function, $y_c(x)$ is the camber function and θ is the camber line slope.

2.1.1. The NACA 4-Digit foil

The NACA 4 digit is ruled by the following structure:

NACA MPXX

where: XX is the maximum thickness in per cent of the chord, M is the maximum value of the mean line in hundredths of the chord and P is the chord-wise position of the maximum camber in the tenths of the chord

The NACA 4 thickness distribution is given by:

$$\frac{y_t}{c} = \left(\frac{t}{c}\right)\left[a_0(x/c)^{\frac{1}{2}} - a_1(x/c) - a_2(x/c)^2 + a_3(x/c)^3 - a_4(x/c)^4\right] \tag{2}$$

where:

$$a_0 = 1.4845 \quad a_2 = 1.7580 \quad a_4 = 0.5075$$
$$a_1 = 0.6300 \quad a_3 = 1.4215$$

As pointed out by Mason [18], using these equations the trailing edge is defined as having a finite thickness in it, which means the foil is not completely closed.

In the NACA 4-digit foils, the maximum thickness is always located at $x/c = 0.30$ and the leading edge radius is

$$\left(\frac{r_{LE}}{c}\right) = 1.1019\left(\frac{t}{c}\right)^2 \tag{3}$$

The angle of the trailing edge is:

$$\delta_{TE} = 2\tan^{-1}\left\{1.16925\left(\frac{t}{c}\right)\right\} \tag{4}$$

And the NACA 4-digit foil camber line is defined by:

$$\left.\begin{array}{ll} \frac{y_c}{c} & = \frac{M}{P^2}\left[2P(x/c) - (x/c)^2\right] \\ \frac{dy_c}{dx} & = \frac{2M}{P^2}(P - (x/c)) \end{array}\right\} \qquad \frac{x}{c} < P \tag{5}$$

and:

$$\left.\begin{array}{ll} \frac{y_c}{c} & = \frac{M}{(1-P)^2}\left[1 - 2P + 2P(x/c) - (x/c)^2\right] \\ \frac{dy_c}{dx} & = \frac{2M}{(1-P)^2}(P - (x/c)) \end{array}\right\} \qquad \frac{x}{c} \geq P \tag{6}$$

The foil is finally generated from the camber line slope which is found from theta in equation (1) and using equation (5) and equation (6), and the surface of the geometry (upper and lower) results from the combination of the thickness distribution equation (2) and camber line using equations of equation (1).

2.1.2. The NACA 5-Digit foil

According to Mason [18], the foil family discussed is an extension of the 4-digit series and includes additional camber lines. The source goes on to explain that the numbering system for these foils is defined as follows:

NACA LPQXX

where XX is the maximum thickness t/c in per cent of the chord, L defines the amount of camber as the design lift coefficient is 3/2 L, in tenths, P marks the location of maximum camber, $x_{maxcamber} = P/2$ given P in tenths of chord; and finally, Q characterizes the camber line being equal to 0 for a standard 5 digit foil camber and equal to 1 for a "reflexed" camber line.

The NACA 5-digit foils use the thickness distribution of the NACA 4-digit. Whereas their standard camber line is defined by:

$$\left.\begin{array}{ll} \frac{y_c}{c} & = \frac{K_1}{6}\left[(x/c)^3 - 3m(x/c)^2 + m^2(3-m)(x/c)\right] \\ \frac{dy_c}{dx} & = \frac{K_1}{6}\left[3(x/c)^2 - 6m(x/c) + m^2(3-m)\right] \end{array}\right\} \qquad 0 \leq \frac{x}{c} \leq m \tag{7}$$

and:

$$\left.\begin{array}{ll} \frac{y_c}{c} & = \frac{K_1}{6}m^3[1-(x/c)] \\ \frac{dy_c}{dx} & = -\frac{K_1}{6}m^3 \end{array}\right\} \qquad m < \frac{x}{c} \leq 1 \tag{8}$$

m delineates the maximum camber position as follows:

$$x_{maxcamber} = m\left(1 - \sqrt{\frac{m}{3}}\right) \tag{9}$$

The parameter m results from a given $x_{maxcamber}$. Additionally, the parameter K_1 above is defined to prevent a leading edge singularity for a prescribed C_{l_i} and m value.

$$K_1 = \frac{6C_{l_i}}{Q} \tag{10}$$

where

$$Q = \frac{3m - 7m^2 + 8m^3 - 4m^4}{\sqrt{m(1-m)}} - \frac{3(1-2m)}{2}[\frac{\pi}{2} - \arcsin(1-2m)] \tag{11}$$

It can be concluded that K_1 is a linear function of C_{l_i} where the values for K_1 were tabulated originally for $C_{l_i} = 0.3$, and are multiplied by $(C_{l_i}/.3)$ to obtain values for other C_{l_i} values. The values of Q and K_1 need to be determined to compute the camber line. Although the computed values of K_1 and Q might slightly differ from the tabulated values because they were calculated in the 1930s by Jacobs and Pinkerton [16], it is advisable to use the tabulated values to reproduce the official ordinates.

Finally, the geometry is designed once the camber line parameters have been chosen and using the above equations.

2.1.3. The NACA Modified 4-Digit foil

As per Mason [18], this profile is an extension of the 4-digit series, which permits a modification of the location of maximum thickness and leading edge radius. The foil's numbering system is established by:

NACA MPXX-IT

where MPXX is the NACA 4-digit designation and the IT defines the modification to the thickness distribution, being I the parameter which fixes the leading edge radius and T the chordwise position of the maximum thickness in tenths of the chord.

Note that I equal to 6 produces the standard 4-digit foils leading edge radius and that an I=9 produces a thick leading edge and an I=1 produces a thin leading edge.

The modified thickness distribution is:

$$\frac{y_t}{c} = 5(t/c)[a_0\sqrt{(x/c} + a_1(x/c) + a_2(x/c)^2 + a_3(x/c)^3] \qquad 0 < \frac{x}{c} \leq T \tag{12}$$

and

$$\frac{y_t}{c} = 5(t/c)[0.002 + d_1(1 - \frac{x}{c}) + d_2(1 - \frac{x}{c})^2 + d_3(1 - \frac{x}{c})^3] \qquad T < \frac{x}{c} \leq 1 \tag{13}$$

According to the given information, the coefficients are obtained by solving for the d's, which are determined based on the trailing edge slope and the condition of maximum thickness at $x/c = T$. Once the coefficients are determined, the *a's* are calculated by establishing a relationship between a_0, the specified leading edge radius, the maximum thickness at $x/c = T$, and the continuity of curvature at $x/c = T$. These constants are initially determined for $t/c = 0.2$ and then scaled to other t/c values by multiplying them by $5(t/c)$. The value of d_1 controls the trailing edge slope and was originally chosen to avoid reversals of curvature. In addition to the tabulated values, Riegels has also provided an interpolation formula.

Lastly, the camber lines for this foil are the same as those for the standard 4-digit airfoils mentioned earlier. Once the camber line is established, the upper and lower ordinates are determined using standard equations.

2.1.4. The NACA 6 and 6A digit foils

The thickness distributions of the profile are numerically determined and therefore, only the mean lines have analytical definitions. Furthermore, there may be alternative sources for the tabulated coordinates, in addition to the values provided in the NACA reports.

The NACA 6-series and 6A-series foils use conformal transformations to relate the flow over a foil to that of a near-circle and a circle. NACA 6-series generators are rare but some efforts have been made to make this NACA designation reach the general public[3]. A four-step algorithm using ψ and ϵ functions is used to calculate each of the five members of the 6-series family and the three members of the 6A-series family, each with its own unique set of functions.

[3] https://www.pdas.com/naca456.html

The functions ψ and ϵ were selected to fit a certain velocity distribution around the airfoil. A scale factor is applied to the ψ and ϵ functions to produce airfoils of varying thickness to chord ratios, with the specific ratio dependent on the chosen scale factor. The resulting thickness is not known in advance [6].

Now, for a specified family and thickness, the thickness distribution may be determined without iteration. From the thickness, the scale factor is computed from the polynomial function shown above. Then, the scale factor is used to multiply the basic values of the ψ and ϵ functions for this foil family [6].

The 6-series airfoils were designed with thin foil theory to achieve a constant loading from the leading edge back to x/c = a and a linear decrease to zero at the trailing edge. The loading at the leading edge must be either zero or infinite within the context of thin foil theory analysis, and the violation of this theory is reflected by the presence of a weak singularity in the mean line at the leading edge. To address this, the slope of the mean line is held constant in front of x/c = 0.005, with the value at that point for the 6-series airfoils [1][18].

According to Abbott and Doenhoff [1], the NACA 6-series designation is usually:

NACA 6X(A),Y-LTT, a = C

where 6 is the 6-series appellative, and X is the chordwise position in tenths of the chord of the minimum pressure coefficient. (A) denotes the 6A series. Then after the comma, Y is the range of lift coefficient in tenths in which favourable pressure gradients exist on both surfaces and can be sometimes a subscript or it can be between brackets. Following the hyphen, L is the design lift coefficient in tenths and TT is the thickness in per cent of the chord. Finally, a = C, C indicates the type of mean line used, where the default assumption is that the uniform-load mean line (a = 1.0) was used when the mean line is unspecified.

To enable the foil to be constructed of almost straight line segments near the trailing edge, the 6A series airfoils utilized an empirical adjustment of the a = 0.8 camber line.

The base camber line equations are:

When $a = 1$ (uniform loading along the entire chord):

$$\frac{y}{c} = -\frac{C_{l_i}}{4\pi}\left[\left(1-\frac{x}{c}\right)\ln\left(1-\frac{x_c}{c}\right)+\frac{x}{c}\ln\left(\frac{x}{c}\right)\right] \tag{14}$$

and

$$\frac{dy}{dx} = \frac{C_{l_i}}{4\pi}\left[\ln\left(1-\frac{x}{c}\right)-\ln\left(\frac{x}{c}\right)\right] \tag{15}$$

where c_{Ii} is the "ideal" or design lift coefficient, which occurs at zero angle-of-attack.

For $a < 1$:

$$\frac{y}{c} = \frac{C_{l_i}}{2\pi(1+a)} \left\{ \begin{array}{c} \frac{1}{1-a}\left[\frac{1}{2}(a-\frac{x}{c})^2 \ln|a-\frac{x}{c}| - \frac{1}{2}(1-\frac{x}{c}^2)\ln(1-\frac{x}{c}) \right. \\ \left. +\frac{1}{4}(1+\frac{x}{c})^2 - \frac{1}{4}(a-\frac{x}{c})^2\right] \\ -\frac{x}{c}\ln(\frac{x}{c}) + g - h\frac{x}{c} \end{array} \right\} \tag{16}$$

with

$$\begin{aligned} g &= \frac{-1}{(1-a)}\left[a^2\left(\frac{1}{2}\ln a - \frac{1}{4}\right) + \frac{1}{4}\right] \\ h &= (1-a)\left[\frac{1}{2}\ln(1-a) - \frac{1}{4}\right] + g \end{aligned} \tag{17}$$

and

$$\frac{dy}{dx} = \frac{C_{l_i}}{2\pi(1+a)}\left\{\frac{1}{1-a}\left[\left(1-\frac{x}{c}\right)\ln\left(1-\frac{x}{c}\right) - \left(a-\frac{x}{c}\right)\ln\left(a-\frac{x}{c}\right)\right] - \ln\left(\frac{x}{c}\right) - 1 - h\right\} \tag{18}$$

The 6A-series means $a = 0.8$. For $0 < x/c < 0.87437$, the basic camber line is used with a modified lift coefficient $C_{l_i mod} = C_{l_i}/1.0209$. For $0.87437 < x/c < 1$ a linear equation is used:

$$\frac{y_c/c}{C_{I_i}} = 0.0302164 - 0.245209\left(\frac{x}{c} - 0.87437\right) \tag{19}$$

and

$$\frac{dy}{dx} = -0.245209 C_{l_i} \tag{20}$$

2.2. PARSEC formulation

PARSEC is a widely used and effective approach for parameterizing foil shapes. It involves utilizing eleven fundamental parameters to fully specify the geometry of the foil.

The various parameters as shown in figure 2 are leading edge radius (r_{le}), upper crest location(X_{up}, Z_{up}), lower crest location (X_{lo}, Z_{lo}), upper and lower curvature (Z_{XXup}, Z_{XXlo}), trailing edge coordinate(Z_{te}) and direction (α_{le}), trailing edge wedge angle (β_{te} and thickness ΔZ_{te}) [30].

PARSEC allows strong control over the curvature and is similar to NACA 4-digit series as it uses a polynomial of higher order (6th) [30].

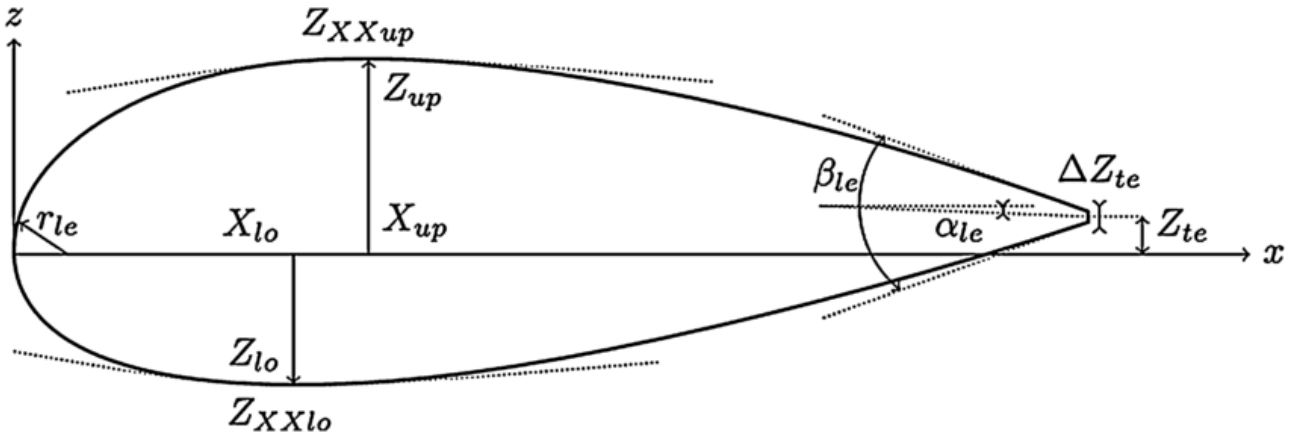

Figure 2: PARSEC parameters definition [30].

$$Z_{PARSEC} = \sum_{n=1}^{6} a_n(p) \cdot X^{\frac{n-1}{2}} \tag{21}$$

Equation (21) is used independently for the upper and lower surfaces. The coefficients a_n are determined based on the given geometric parameters as follows [10]:

$$C_{up} \times a_{up} = b_{up}, \qquad C_{lo} \times a_l = b_{lo} \tag{22}$$

where matrices C_{lo} and C_{up} are both coefficient matrices and b_{lo} and b_{up} are designated as below:

$$C_{up} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \sqrt{p_2} & p_2^{3/2} & p_2^{5/2} & p_2^{7/2} & p_2^{9/2} & p_2^{11/2} \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 \\ 0.5p_2^{-1/2} & 1.5\sqrt{p_2} & 2.5p_2^{3/2} & 3.5p_2^{5/2} & 4.5p_2^{7/2} & 5.5p_2^{9/2} \\ -0.25p_2^{-3/2} & 0.75p_2^{-1/2} & \frac{15}{4}p_2^{1/2} & \frac{35}{4}p_2^{3/2} & \frac{63}{4}p_2^{5/2} & \frac{99}{4}p_2^{7/2} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

to obtain C_{lo} we must change p_2 with p_5 in equation (23).

$$B_{\text{up}} = \begin{bmatrix} p_8 + \frac{p_9}{2} \\ p_6 \\ \tan\left(p_{10} - \frac{p_{11}}{2}\right) \\ 0 \\ p_7 \\ \sqrt{2p_1} \end{bmatrix}, \qquad B_{\text{lo}} = \begin{bmatrix} p_8 - \frac{p_9}{2} \\ p_3 \\ \tan\left(p_{10} + \frac{p_{11}}{2}\right) \\ 0 \\ p_4 \\ -\sqrt{2p_1} \end{bmatrix} \tag{24}$$

PARSEC parameter	Geometry parameter	Definition
p_1	r_{le}	leading edge radius
p_2	X_{up}	upper crest position in horizontal coordinates
p_3	Z_{up}	upper crest position in vertical coordinates
p_4	Z_{XXup}	upper crest curvature
p_5	X_{lo}	lower crest position in horizontal coordinates
p_6	Z_{lo}	lower crest position in vertical coordinates
p_7	Z_{XXlo}	lower crest curvature
p_8	Z_{te}	trailing edge offset in vertical sense
p_9	ΔZ_{te}	trailing edge thickness
p_{10}	α_{te}	trailing edge direction
p_{11}	β_{te}	trailing edge wedge angle

Table 1: PARSEC parameters definition [10].

2.3. Geometry parameterization set-up

Once some foil constraints can be fixed regarding the optimization problem presented in 1.2, from the formulations explained in the section 2 we can conclude that NACA 4 and 5 will produce the same symmetrical profiles. The ones that give more control over the thickness and foil shape are the NACA 4 modified series, NACA 6-series and PARSEC. These methodologies allow the optimizer to handle better the thickness and shape of the geometry, allowing it to cover a more interesting range of foils. Nevertheless, at the early stages of this work, cambered foils were also studied.

The design parameters of each formulation can be gathered from sections 2.1.3, 2.1.4 and table 1. In the NACA 4-digits modified, it will be mandatory to maintain the first two digits as 0s, 00XX-IT; while in the NACA 6-series, the design lift coefficient digit must be set to 0, 6X(A),0TT. The "Y" shown in the section mentioned above, is not necessary as symmetrical foils are going to be obtained, as well as the "a" parameter. Using PARSEC to obtain symmetrical foils, on one hand, we must fix that the parameters that control each surface directly (variables from 2 to 7) must be symmetrical: $p_2 = p_5$, $p_3 = -p_4$ and $p_4 = -p_7$. On the other hand, the parameters that affect the camber line must be set to 0, these are p_8 and p_{10}.

Apart from these facts, the number of points and their distribution is important to represent accurately the foil shape. This is a fact easy to deal with by plotting the foils obtained, taking note of the definition of the most curved part of the surfaces, which is mainly the leading edge.

Given the foregoing, the NACA series were used using the NACA456 Program from PDAS[4]. Using this program, we can get nearly 100 points per surface. Meanwhile, a Python program has been developed to use the PARSEC formulation. In this way, we can control

[4] https://www.pdas.com/naca456download.html

the number of points and we can choose between a linear distribution or a half-cosine spacing distribution, which increases the points at LE and TE where they are needed.

We must take into consideration this fact so later we will need a good definition of the LE to obtain valid results in the flow solver chosen.

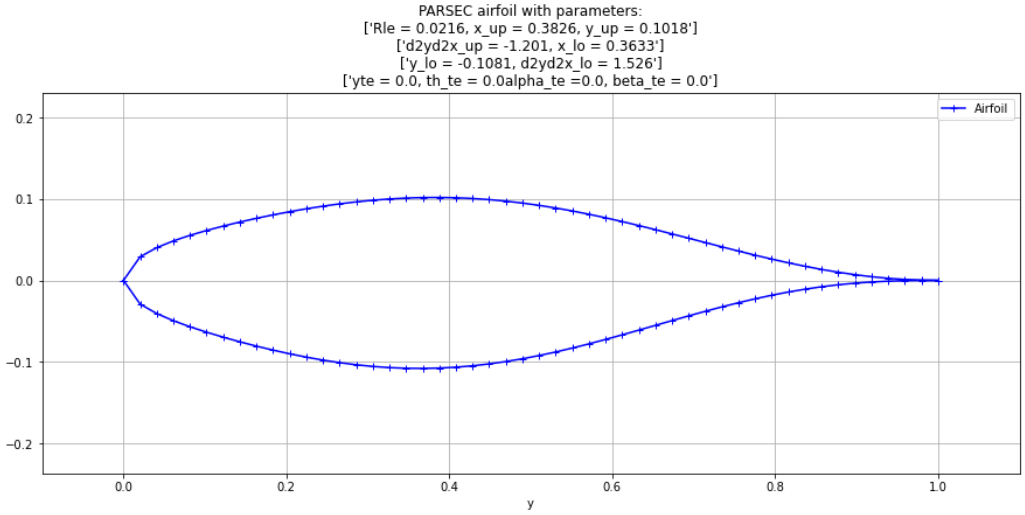

Figure 3: PARSEC foil with 50 points on each surface and linear spacing.

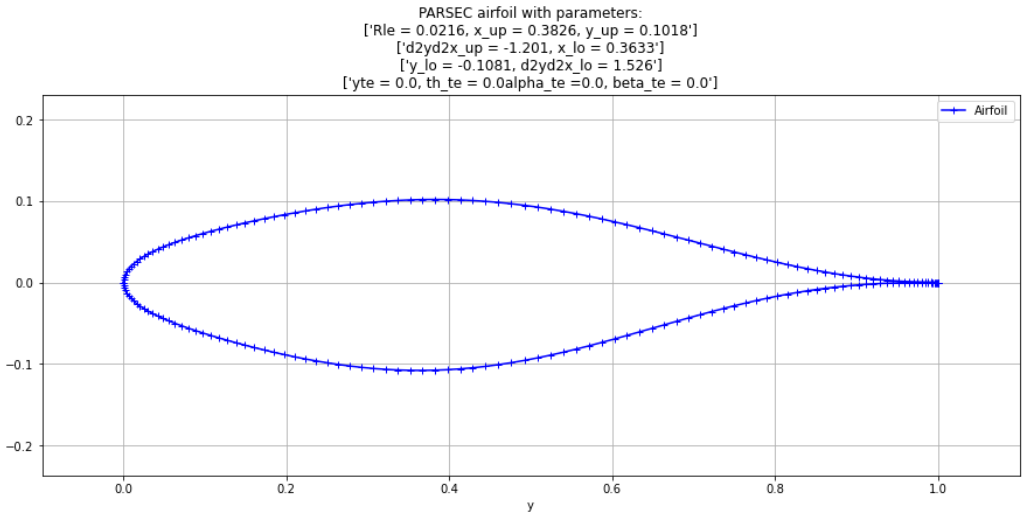

Figure 4: PARSEC foil with 100 points on each surface and cosine spacing.

Moreover, PARSEC can produce invalid geometries because PARSEC coefficients are obtained by solving the linear equations system expressed in equation 22.

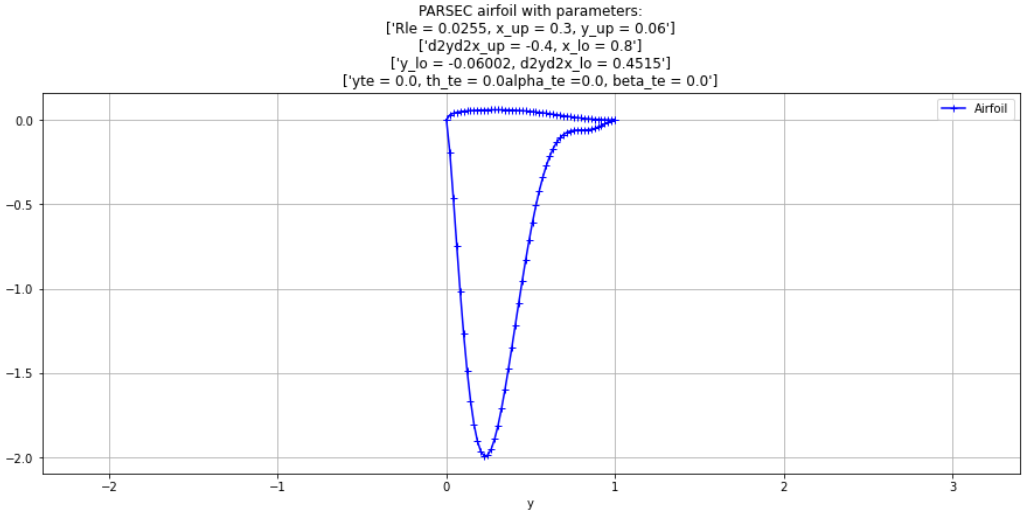

Figure 5: PARSEC impossible geometry I.

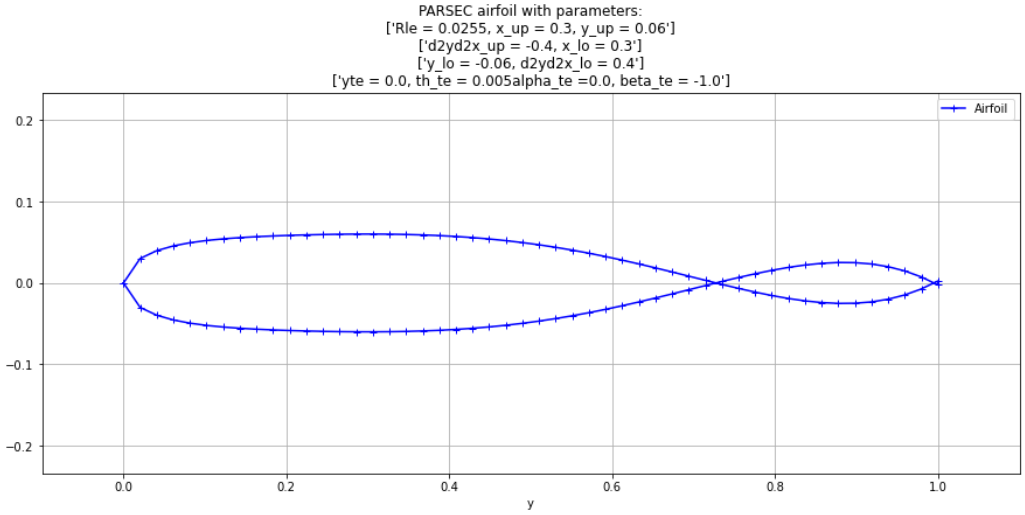

Figure 6: PARSEC impossible geometry II.

Consequentially, the combinations that result in wrong geometries must be avoided. This is achieved by sampling the design space and imposing a penalisation method on the wrong geometries, in so doing, the optimization algorithm will avoid these limits too.

Moreover, some combinations of PARSEC parameters could still produce undesired geometries. Some geometries, despite the fact that can converge in XFoil, the results it gives back can not be accurate.

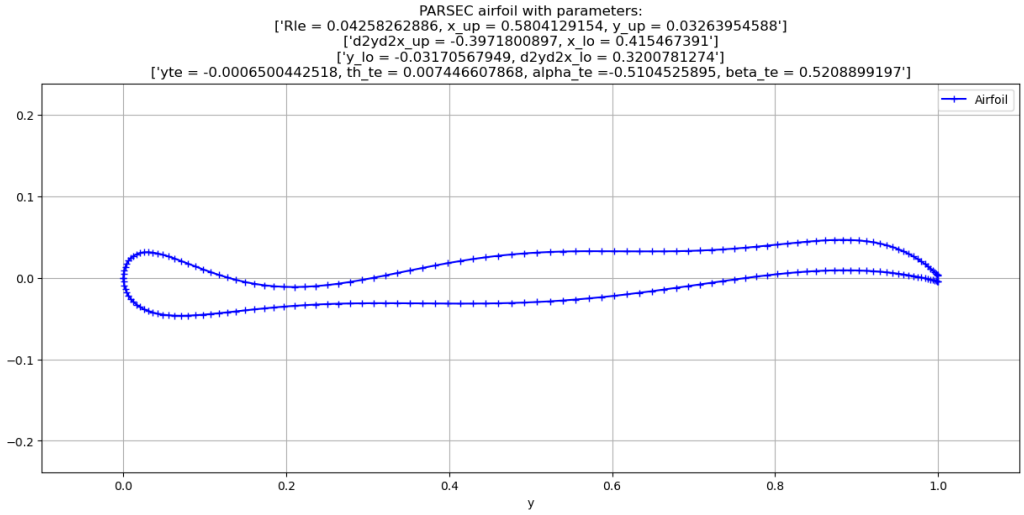

Figure 7: PARSEC undesired geometry.

To avoid this issue, each PARSEC parameter upper and lower limits must be set sensibly. To address this task, the variables were split into pairs and studied in contour plots in which a white space meant a combination that was avoided by the program. The limits in the design variables of PARSEC were fixed as stated in the following table as a consequence:

Parameter	**Lower bound**	**Upper bound**
r_{le}	0.005	0.1
X_{up}	0.1	0.7
Z_{up}	0.05	0.15
Z_{XXup}	0	-1.5
X_{lo}	0.1	0.7
Z_{lo}	-0.05	-0.15
Z_{XXlo}	0	1.5
Z_{te}	-0.1	0.1
ΔZ_{te}	0.001	0.3
α_{te} (in rad)	-1	0.5
β_{te} (in rad)	0	2

Table 2: PARSEC parameters limits in the samplings and optimizations.

3. Flow solvers

Flow solvers influence the foil optimization process significantly. Especially in the cost-efficiency relation of the whole optimization process. An accurate prediction of foil performance can significantly determine the efforts needed to improve the initial design downstream. At the same time, an initial design has to be developed efficiently: in terms of time and cost.

In this work, two of the main available options have been evaluated: an integrated boundary layer formulation and the potential flow panel method, XFoil [20]; and a commercial CFD software, Star-CCM+.

3.1. XFoil

```
 ===============================================
  XFOIL Version 6.99
  Copyright (C) 2000   Mark Drela, Harold Youngren

  This software comes with ABSOLUTELY NO WARRANTY,
    subject to the GNU General Public License.

  Caveat computor
 ===============================================

 File  xfoil.def  not found

   QUIT    Exit program

  .OPER    Direct operating point(s)
  .MDES    Complex mapping design routine
  .QDES    Surface speed design routine
  .GDES    Geometry design routine

   SAVE f  Write airfoil to labeled coordinate file
   PSAV f  Write airfoil to plain coordinate file
   ISAV f  Write airfoil to ISES coordinate file
   MSAV f  Write airfoil to MSES coordinate file
   REVE    Reverse written-airfoil node ordering

   LOAD f  Read buffer airfoil from coordinate file
   NACA i  Set NACA 4,5-digit airfoil and buffer airfoil
   INTE    Set buffer airfoil by interpolating two airfoils
   NORM    Buffer airfoil normalization toggle
   XYCM rr Change CM reference location, currently  0.25000 0.00000

   BEND    Display structural properties of current airfoil

   PCOP    Set current-airfoil panel nodes directly from buffer airfoil points
   PANE    Set current-airfoil panel nodes ( 160 ) based on curvature
  .PPAR    Show/change paneling

  .PLOP    Plotting options

   WDEF f  Write  current-settings file
   RDEF f  Reread current-settings file
   NAME s  Specify new airfoil name
   NINC    Increment name version number

   Z       Zoom    | (available in all menus)
   U       Unzoom  |

 XFOIL   c>
```

Figure 8: XFoil terminal.

XFoil is a versatile and interactive program that is specifically designed for the analysis and design of subsonic isolated airfoils. It offers a range of menu-driven routines that perform various functions such as viscous analysis, airfoil design, and Karman-Tsien compressibility correction. One of the key features of XFoil is its ability to facilitate airfoil design and redesign through the interactive specification of a surface speed distribution. This allows users to easily modify geometric parameters such as maximum thickness, camber, leading-edge radius, trailing edge thickness, camber line, loading change specification, flap deflection, and explicit contour geometry. XFoil also offers additional functionalities such as blending airfoils, drag polar calculation, writing and reading of airfoil geometry, saving polar files, and plotting geometry, pressure distributions, and polar curves [11].

XFoil 1.0, developed by Mark Drela in 1986, aimed to combine the speed and accuracy of

high-order panel methods with the newly-introduced fully-coupled viscous/inviscid interaction method used in the ISES code by Drela and Giles.

The inviscid formulation of XFoil utilizes a linear-vorticity stream function panel method with a finite trailing edge base thickness modelled using a source panel. It also incorporates a Karman-Tsien compressibility correction, enabling accurate predictions in compressible flows up to sonic conditions. However, the accuracy of XFoil diminishes rapidly in the transonic regime, and it is unable to reliably predict shocked flows.

XFoil employs two types of inverse methods: Full-Inverse and Mixed-Inverse. Full-Inverse utilizes Lighthill's and van Ingen's complex mapping method, which calculates the complete airfoil geometry based on the entire surface speed distribution. On the other hand, Mixed-Inverse is an inviscid panel formulation in which the panel node coordinates are treated as unknowns at locations where the surface speed is prescribed. This allows for the alteration of only a portion of the airfoil at a time.

The boundary layers and wake in XFoil are described using a two-equation lagged dissipation integral boundary layer formulation, along with an envelope transition criterion that is adopted from the transonic analysis/design ISES code. The viscous solution is strongly coupled with the incompressible potential flow through the surface transpiration model. Drag is determined from the momentum thickness of the wake far downstream. The total velocity at each point on the airfoil surface and wake is obtained from the panel solution, with the Karman-Tsien correction added. The execution times of XFoil are rapid, typically taking around 1 or 2 seconds on a nowadays computer.

In cases where the lift is specified, the wake trajectory is obtained from an inviscid solution at the specified lift or angle of attack. However, XFoil does not perform a secondary correction as a new source influence matrix would need to be calculated each time the wake trajectory is changed. The impact of this approximation on overall accuracy is small and mainly noticeable near or past stall conditions. In the attached cases, the effect of an incorrect wake trajectory is usually imperceptible.

The lift and moment coefficients (C_L, C_M, and C_p) in XFoil are calculated through direct surface pressure integration. The pressure coefficient (C_p) is determined using the Karman-Tsien compressibility correction to account for compressibility effects. The drag coefficient (C_D) is obtained using the Squire-Young formula at the last point in the wake, but not at the trailing edge. XFoil also provides information on the friction and pressure drag components (C_{Df}, C_{Dp}) of the total drag coefficient, allowing for a detailed analysis of the drag characteristics of the airfoil.

Transition in an XFoil solution can be triggered by either free or forced transition. The e^n method, used in XFoil, incorporates a user-specified parameter known as Ncrit, which represents the logarithm of the amplification factor of the most-amplified frequency that triggers the transition. The appropriate value for Ncrit depends on the ambient disturbance level, and it can be adjusted by the user to achieve an accurate prediction of transition location on the airfoil surface.

Panel density requirements are crucial for accurately capturing strong separation bubbles in viscous solutions using XFoil. If a separation bubble appears to be poorly resolved, it is important to re-panel the airfoil with a higher number of points and/or concentrated around the bubble region. For moderate chord Reynolds numbers (ranging from 1 to 3 million), finer

panelling is typically necessary. Insufficient resolution of the separation bubble can result in a "ragged" or "scalloped" lift coefficient (C_L) versus drag coefficient (C_D) polar curve.

XFoil incorporates upwinding into the equations automatically only in regions of rapid change, ensuring that the overall numerical scheme is stable and as accurate as possible. Some small oscillations in the shape parameter H may occur near the stagnation point, but these are cosmetic defects that do not significantly affect the downstream development of the boundary layer [11].

3.2. CFD and Star-CCM+

The usage of simulation tools has proven essential to the creation of many commonplace technology. In actuality, modern numerical simulation tools serve as technological enablers. CFD is this kind of tool. The Navier-Stokes equation, which ingeniously represents an entire range of flow phenomena, including turbulent or laminar single-phase incompressible flows, compressible all-speed flows, and multiphase flows, is at the core of this instrument. The Finite Volume Method has emerged as a standout among the numerical techniques applied to CFD. The Finite Volume Method (FVM) is a numerical technique that transforms partial differential equations into discrete algebraic equations over finite volumes [21].

In addition, some of the effects that are challenging to identify in experimental experiments are determined and shown using CFD approaches based on RANS flow modelling. Every CFD code might be broken down into three sections, each of which corresponds to a different stage of the problem-analysis process.

1. The pre-processor builds the problem, imports a mesh, and implements the physical problem into a mathematical model, dividing the computational domain into elements. Next, the domain and analysis type (multiphase, VOF, turbulence) are defined, and boundary conditions are set. Global accuracy depends on mesh quality, with finer meshing for smooth flow changes and coarser for higher gradients.

2. The solver module is the core of a CFD code, which uses numerical solution algorithms to solve algebraic systems of equations. These programs can run even in the background and build backup and result files. Users can control the analysis using residuals or variables, allowing for quick and easy monitoring of convergence. Launching and relaunching the solver without affecting the solver allows for quick problem resolution and monitoring of divergence.

3. The postprocessor module analyzes solution results, generating solution variables for grid nodes or volumes, which are collected and elaborated for a physical representation.

It has to be taken into account that the majority of flows in both nature and technology display turbulence, a type of instability. When the flow velocity, or more accurately, the Reynolds number, hits a specific critical threshold, the regime changes. This makes it difficult to solve the equations analytically and necessitates the formulation of numerical methods to solve for certain (statistically stationary) states within the flow. All flow instability is averaged out and considered to be a component of turbulence in the Reynolds-averaged Navier-Stokes (RANS) approach to turbulence. Those are presented as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{25}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_i}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_j}(\rho\overline{u_i' u_j'}) \tag{26}$$

These Reynolds stresses $\rho\overline{u_i' u_j'}$ are to be modeled to close equation 26.

The selection of the numerical method and the solution process depends on the code's application domain and the modelling approach used. Even though there are many models offered in any commercial program, this research has only employed two-equation turbulence models: Realizable $k-\epsilon$ and $k-\omega$ SST.

On the one hand, the most popular is the $k-\epsilon$ model. Being stable and numerically robust, it provides a fair balance between accuracy and resilience. In this model ϵ is the pace at which the smallest eddies' turbulent energy dissipates due to viscosity. The model is said to be "realizable" as it fulfills specific mathematical restrictions on the Reynolds stresses that are in line with the physics of turbulent flows.

On the other hand, in the $k-\omega$ model, $\omega = \epsilon/k$ is the dissipation rate. The SST (Shear Stress Transport) model implements an enhanced approximation of the principal shear stress in adverse pressure gradients [15].

3.2.1. Star-CCM+

Simcenter Star-CCM+ Software is a Computational Aided Engineering (CAE) solution developed by Siemens Digital Industries Software for solving multidisciplinary problems in both fluid and solid continuum mechanics. It provides one of the world's most comprehensive engineering physics simulations inside a single integrated package. It offers all stages required for carrying out engineering analyses, including import and creation of geometries, mesh generation, solution of governing equations, analysis of results, automation of simulation workflows, and connection to other CAE software for co-simulation analysis [29].

Numerous researchers [13][34] have studied its performance in comparison with other commercial software with promising results.

3.3. Solver set-up

Hereunder it is described the configuration and models adopted to solve the flow in the described study case in section 1.2.

3.3.1. XFoil configuration

Setting up XFoil requires only a few steps:

- Load the coordinate file of the geometry.
- Use the `pane` option to re-pane the profile.
- Increase the limit of iterations to 200.
- Set viscosity equal to Re = 1e+06.
- Force Boundary Layer transition at x = 0.01.

- Run the AoA.

We will use the XFOIL repanelization option because it was studied that using the distribution of points how it appears in the text file resulted in a C_p distribution erroneous. In figure 9 looking at the emphasized sections, we can observe that the pressure distribution is not smooth.

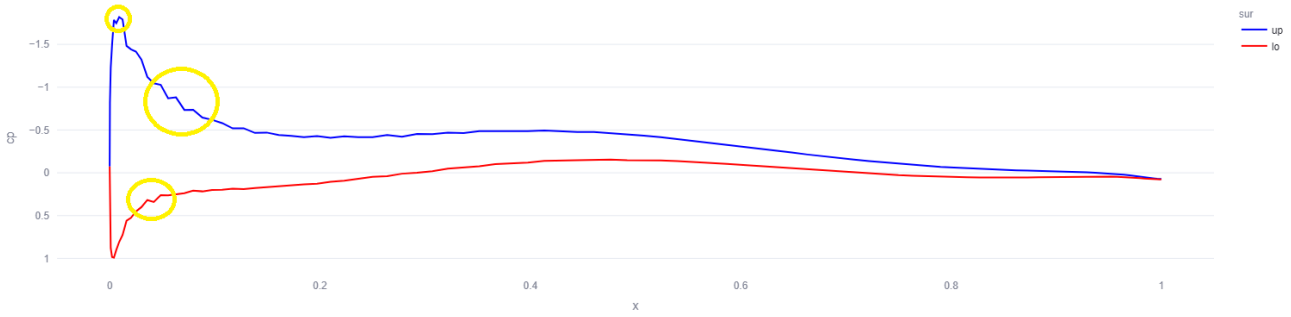

Figure 9: C_p distribution of a foil produced with the panelization as described in its coordinates file.

3.3.2. Star-CCM+: models and mesh

In Star-CCM+, the models used were the ones recommended by the user guide for incompressible external aerodynamics:

- Bi-dimensional.
- Time: steady
- Material: liquid, constant density, water ($\rho = 997.561 kg/m^3$, μ =8.8871e-4$Pa-s$).
- Flow: segregated flow.
- Viscous Regime: turbulent.
- Turbulence: Reynolds-Averaged Navier-Stokes, Realizable $k-\epsilon$.

The other main aspect to set up properly is the mesh. First of all, the domain in which the geometry is located is 150 chords long and 100 wide, being the foil at 50 chords of the inlet (velocity) and 100 from the outlet (pressure). The geometry is imported as an open spline at the TE and then closed. Polyhedral mesh with prim layer controls was used. Some insights of a typical mesh for any of the geometries simulated will look like this:

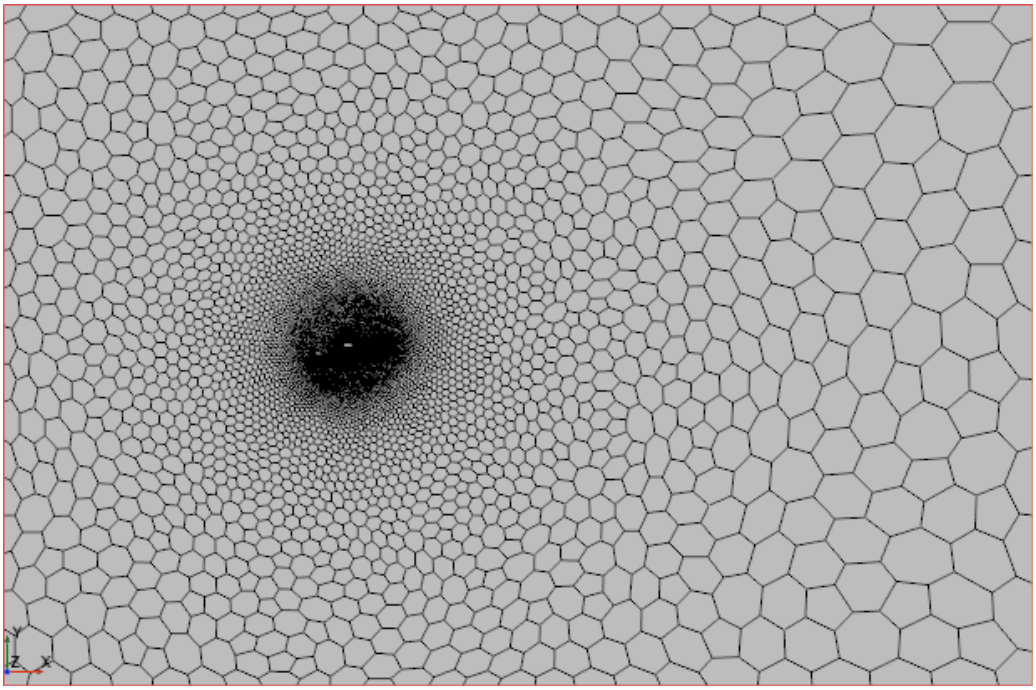

Figure 10: Mesh I, zoomed out.

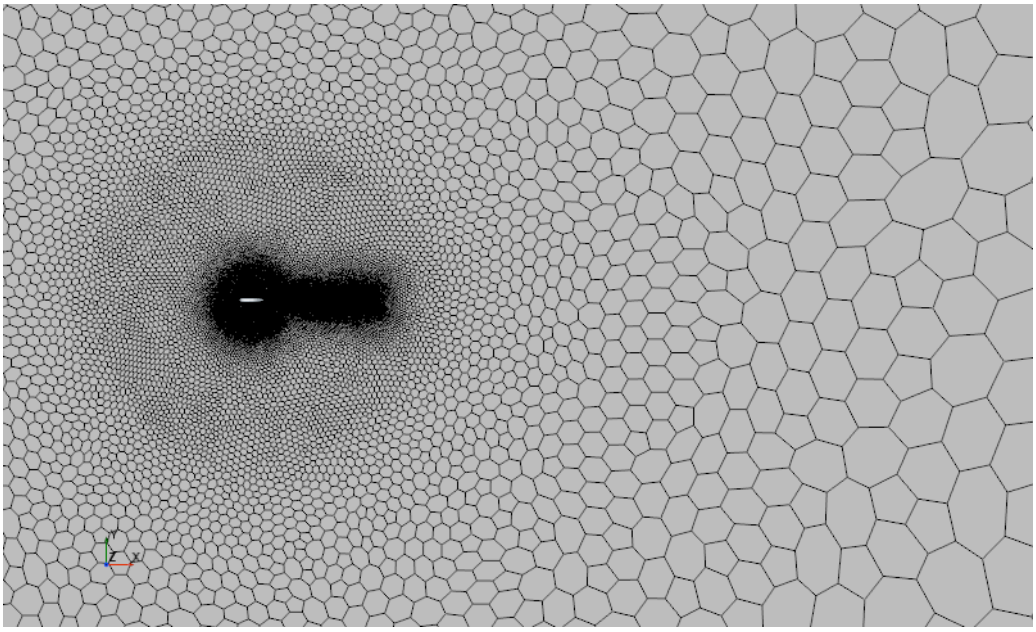

Figure 11: Mesh II, zooming in.

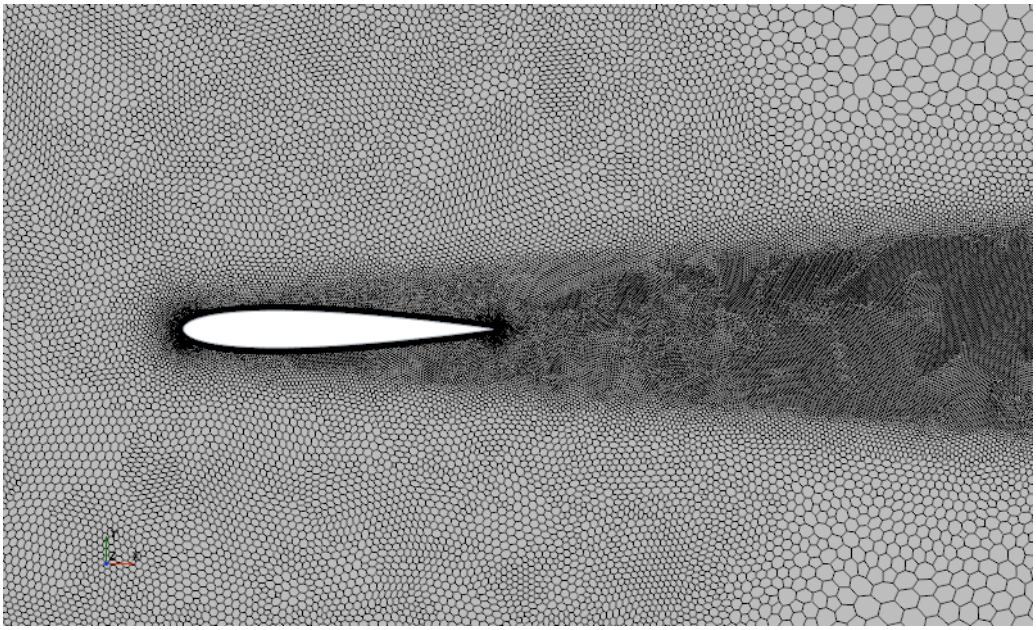

Figure 12: Mesh III: near profile.

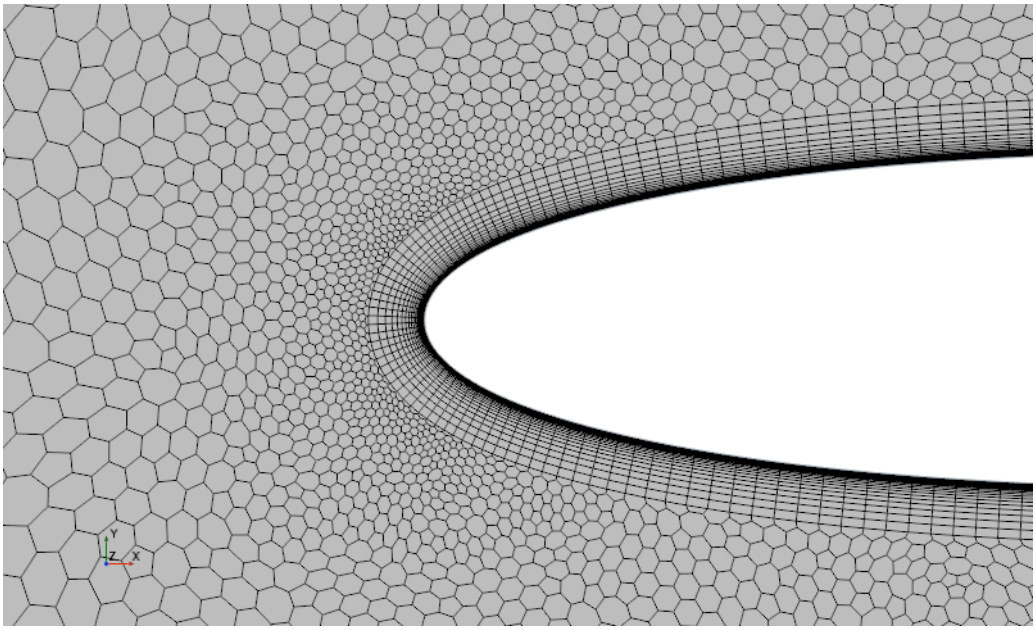

Figure 13: Mesh IV: detail of LE.

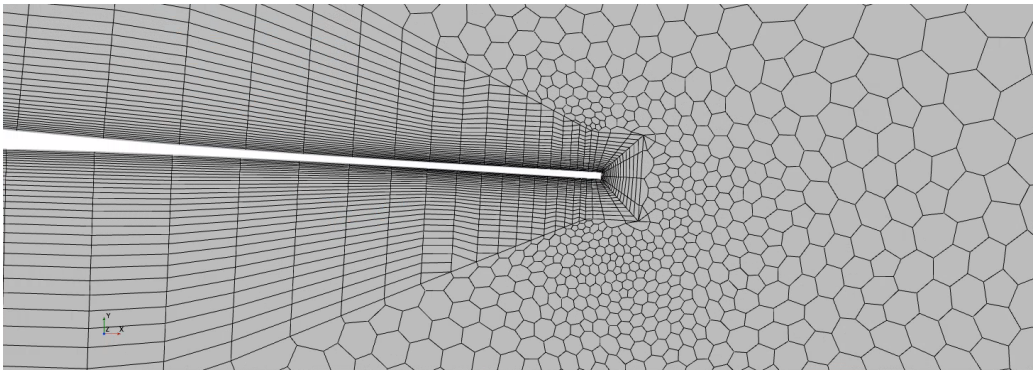

Figure 14: Mesh V: detail of TE.

3.3.3. NACA 0012 Validation

This configuration was tested following the 2D NACA 0012 Airfoil Validation Case[5]. The results were compared with the force data from Ladson, Hill, & Johnson (NASA TM 100526, 1987) and the pressure data with the data from Gregory & O'Reilly (NASA R&M 3726, Jan 1970). The results were also compared with a simulation using $k-\omega$ turbulence.

AoA	Source	C_D	C_L
0	$k-\omega$	8.19e-03	-6.67e-07
	$k-\epsilon$	8.11e-03	2.84e-05
	Experimental data	8.11e-03	-1.30e-03
10	$k-\omega$	1.35e-02	1.05e+00
	$k-\epsilon$	1.33e-02	1.08e+00
	Experimental data	1.16e-02	1.08e+00

Table 3: Comparison with simulation results using different turbulence models and experimental data from Ladson.

[5] https://turbmodels.larc.nasa.gov/naca0012_val.html

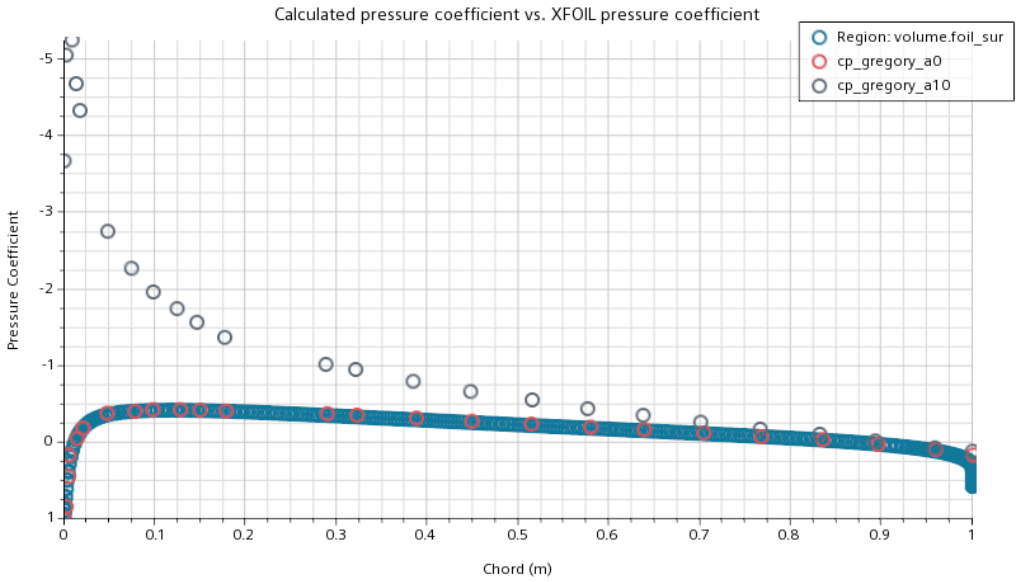

Figure 15: Pressure distribution comparison: simulation on Star-CCM+ with $k-\epsilon$ turbulence vs. experimental data from Gregory at AoA 0.

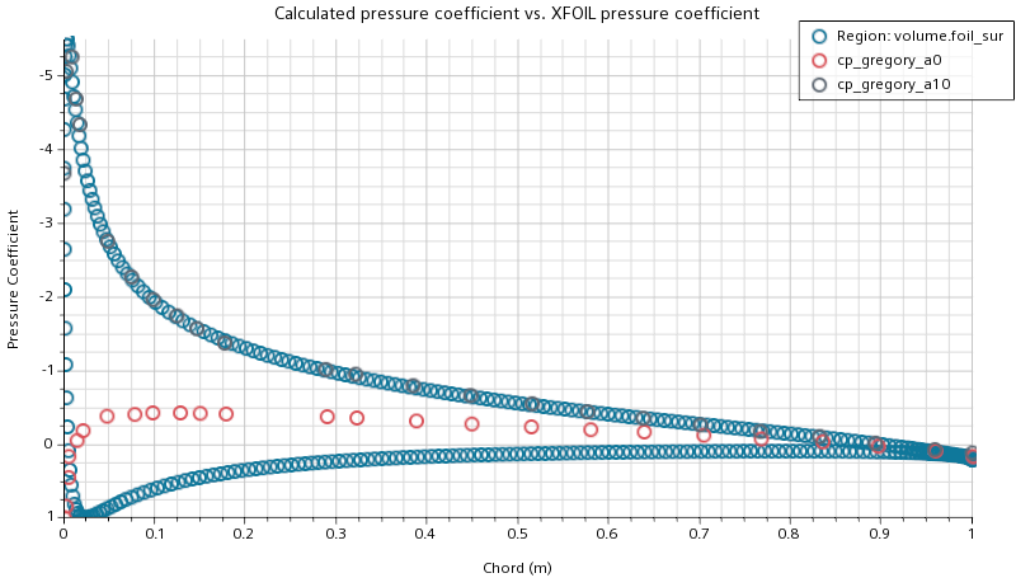

Figure 16: Pressure distribution comparison: simulation on Star-CCM+ with $k-\epsilon$ turbulence vs. experimental data from Gregory at AoA 10.

Regarding these results, it was concluded that the mesh was adequately defined and, regarding the table 3, the most correct turbulence model was the $k-\epsilon$ because using it, the results approached better the experimental data than using the $k-\omega$ model, although this model has shown to be better for foils applications [15].

4. Optimization Methods

In this project, two optimization methods have been used: gradient methods and genetic or evolutionary algorithms. They have been implemented using Dakota[6].

The Dakota project provides state-of-the-art research and robust software for optimization and UQ (Uncertainty Quantification). Its advanced parametric analyses enable design exploration, model calibration, risk analysis, and quantification of margins and uncertainty. The toolkit provides a flexible interface between simulation codes and its iterative systems analysis methods, such as optimization, uncertainty quantification, parameter estimation, and sensitivity/variance analysis. These capabilities can be used on their own or as components within advanced strategies [25].

It is also needed to mention that previous to the optimization process, when the response function behaviour is unknown, sampling methods are recommended to set up properly the optimization process. In this project, parametric studies and Latin Hypercube Sampling have been used and compared to study the design space.

4.1. Sampling methods

As it has been said, to study the design space parametric studies or other sampling methods such as LHS can be used.

Parametric study

There are several parametric study methods available. In parametric studies, one sort of sensitivity analysis is performed by calculating response data sets at various places in the parameter space to examine the impact of parametric modifications on simulation models. In Dakota there are available: *vector*, *list*, *list*, and *multidimensional* methods that perform parameter studies in n-dimensional parameter space. *vector* uses a line between two points, *list* uses a list of points, *list* evaluates nearby points along the coordinate axes, and *multidimensional* forms a regular lattice or hypergrid. For example, a *multidimensional* parametric study will result in $(p+1)^n$ function evaluations being p the number of partitions and n the number of dimensions assuming all dimensions will be divided in the same number of partitions [27].

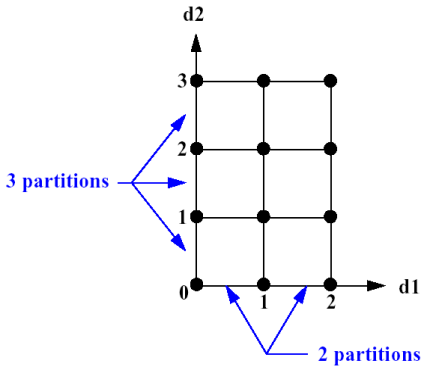

Figure 17: Example multidimensional parameter study [27].

[6] https://dakota.sandia.gov/

Latin Hypercube Sampling

The LHS technique of stratified random sampling was initially created for effective uncertainty analysis. LHS divides the parameter space into bins with equal probabilities in order to get a more uniform distribution of sample points in the parameter space than would be feasible with pure random sampling [32].

4.2. Optimization formulation

A general optimization problem can be defined as follows:

$$\begin{gathered} \text{minimize:} \quad f(\mathbf{x}) \\ \mathbf{x} \in \Re^n \\ \text{subject to:} \quad \mathbf{g}_L \leq \mathbf{g}(\mathbf{x}) \leq \mathbf{g}_U \\ \mathbf{h}(\mathbf{x}) = \mathbf{h}_t \\ \mathbf{a}_L \leq \mathbf{A}_i \mathbf{x} \leq \mathbf{a}_U \\ \mathbf{A}_e \mathbf{x} = \mathbf{a}_t \\ \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \end{gathered} \tag{27}$$

Vectors and matrices are marked in bold, being $\mathbf{x}$ an n-dimensional vector of design variables or design parameters. $\mathbf{x}_L$ and $\mathbf{x}_U$ are the lower and upper bounds of the design parameters.

There is no need to mention that as stated in equation 27, the goal is to minimize $f(\mathbf{x})$ while satisfying the constraints. These constraints are defined with $\mathbf{g}$ and $\mathbf{h}$ with upper $\mathbf{g}_U$, lower $\mathbf{g}_L$ and target values $\mathbf{h}_t$ correspondingly. We can distinguish here from non-linear inequality constraints, $\mathbf{g}$; and non-linear equality constraints, $\mathbf{h}$. These limitations build linear systems, $\mathbf{A}_i$ and $\mathbf{A}_e$ with their limits and target values respectively.

The optimization issue can be solved in a variety of ways, all of which iterate on $\mathbf{x}$. The response values, $f(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, and $\mathbf{h}(\mathbf{x})$, are determined, frequently by conducting a simulation, and then some technique is performed to produce a new $\mathbf{x}$ that will either lower the objective function, reduce the degree of infeasibility, or both. Three criteria will be used to distinguish these strategies in the discussion that follows to ease a broad presentation of them: type of optimization issue, search objective, and search strategy.

The types of constraints and the linearity or non-linearity of the goal and constraint functions can be used to identify the type of optimization issue. A hierarchy of complexity, ranging from straightforward limit constraints to fully nonlinear constraints, is used to categorize constraints. Nonlinear programming (NLP) issues are frequent in engineering applications, whereas linear programming (LP) issues are frequent in scheduling, logistics, and resource allocation applications.

We must distinguish between global and local optimization. Global optimization aims to find the design point that gives the lowest feasible objective function value over the entire parameter space, while local optimization aims to find the lowest design point relative to a nearby region. Both are computationally expensive, so users must choose an optimization algorithm with an appropriate search scope that best fits their problem goals and budget.

The search method is the approach taken in the optimization algorithm to locate a new design point that has a lower objective function or is more feasible than the current design point. It can be classified as either gradient-based or non-gradient-based. In a gradient-based algorithm, gradients of the response functions are computed to find the direction of improvement. However, this approach can be computationally expensive, inaccurate, or even nonexistent, so non-gradient-based search methods may be useful. Examples of non-gradient-based optimization include pattern search methods and genetic algorithms [26].

4.3. Gradient methods

For effective navigation to a local minimum close to the starting point, gradient-based optimizers are best suited. Of all local optimization techniques, they are the most effective and have the highest convergence rates. They may, however, be less reliable if the issue displays non-smooth, discontinuous, or multi-modal behaviour [26].

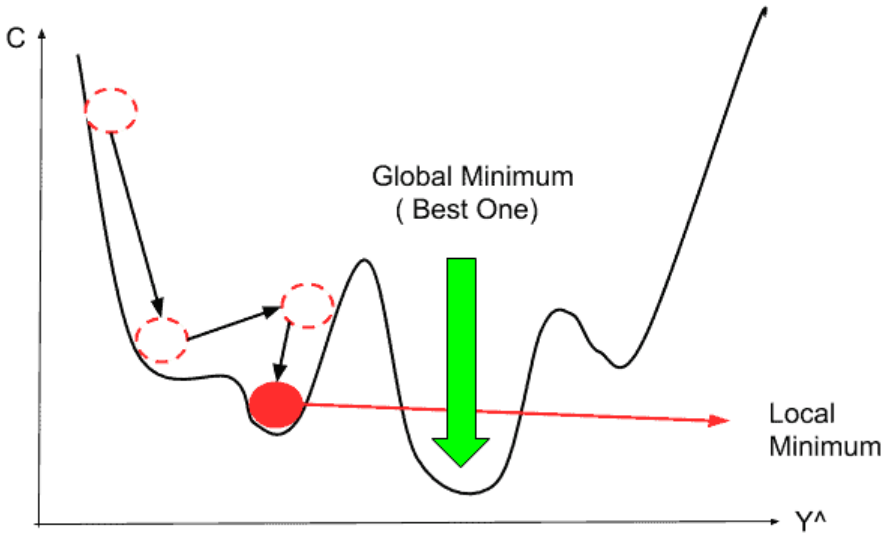

Figure 18: Gradient methods behaviour at the presence of local minimums [33].

In derivative methods, scales play an important factor. A bad scaling can prevent reaching a solution. Scales can define weights within the design parameters: the optimization algorithm sees some variables as more relevant than others because as they are badly scaled, these relevant variables impact heavily in the optimization algorithm [14]. For these reasons, the algorithm must use parameters relative to its corresponding design variable magnitude order, otherwise, if absolute values are going to be used, variables must be correctly scaled.

For gradient-based optimizers, gradient accuracy is crucial since erroneous derivatives frequently result in search failures or early method termination. Hessians and analytical gradients are desirable yet frequently unavailable. A complete Newton approach will reach quadratic convergence rates close to the solution if an application code can give analytic gradient and Hessian information. Super-linear convergence rates can be determined if just gradient information is given and the Hessian information is roughly inferred from a collection of gradient data. However, the optimization technique will often employ a finite difference approach to determine gradient values for engineering applications. To allow for local precision and

convergence, the finite difference step size should be chosen as tiny as feasible; nevertheless, it should not be chosen so small that the steps are "in the noise". This calls for evaluating the response functions' local smoothness using, for instance, a parameter research approach. In general, central differences will result in more trustworthy gradients than forward differences, but at about twice the cost [26].

In this research, two gradient-based methods have been studied: a method of feasible direction and a Newton method. Gradient methods rely on the same idea which is the use of the gradient projection of the objective function to look for a descent direction on it. The typical gradient method has the following steps [8]:

1. Select an initial point. This is a crucial step because selecting a point close to the optimum can prevent the algorithm to find a descent direction whereas choosing a bad point can make the optimization take longer.
2. Find a descent direction d_i. From a set of points obtained iterating, the algorithm search for a descent direction. If no direction is found, the method stops.
3. Determine a step length such that $f(x_i + \delta d_i) \leq f(x_i)$.
4. Update the point $x_{i+1} = x_i + \delta_i d_i$ and go to step 2 until a termination criterion is reached.

The main differences between descent methods are how each algorithm computes the descent direction. The direction-finding problem in a method of feasible direction can be defined as follows [8]:

$$\min\{\max\left\{\nabla f(x_i)^T d; f(x_i) + \nabla f(x_i)^T d\right\}\} \tag{28}$$

While for a Newton method [12]:

$$\min\left\{\max\left\{\nabla f(x_i)^T d + \frac{1}{2} d^T \nabla^2 f(x_i) d\right\}\right\} \tag{29}$$

Newton's methods and MFD can be applied to non-linear optimization problems while through all optimization iterations, both techniques remain feasible. Nevertheless, in Newton's methods, finding a solution to a linear system of equations that was created by setting the derivative of a second-order Taylor series expansion to zero is the main goal of the subproblems connected to these techniques [26].

4.4. Genetic or Evolutionary Algorithms

The foundation of evolutionary algorithms (EA) and genetic algorithms (GA) is Darwin's hypothesis of the survival of the fittest. They begin with a population of design points in the parameter space that is randomly chosen, where the values of the design parameters combine to create a "genetic string" that is used to identify each design point. The algorithm then proceeds through a series of generations, only allowing the best design points to live and reproduce because they are deemed to be the most "fit" ones. It mimics evolution by using mathematical representations of phenomena including mutation, breeding, and natural selection. A design

point that minimizes the optimization problem's objective function is ultimately found by the optimizer [26]. The GA and the EA differ slightly in the way they deal with design parameters and algorithm parameters: in classic GA there is no distinction between types of algorithm parameters. In actuality, all of the factors are "outside"-set, or exogenous in terms of Evolution Strategies (ES) [4][23].

The fittest individuals are evaluated taking into account the value of objective function whilst satisfying the constraints. If a constraint is not respected the algorithm can penalise the fitness of the individual.

As described in the Dakota reference manual, the steps of an evolutionary algorithm are to randomly select an initial population, select parents based on relative fitness, apply crossover and mutation to generate new individuals, perform function evaluations on the new individuals, perform replacement to determine the new population, and continue until convergence criteria are satisfied or iteration limits are exceeded.

Meanwhile, the steps of the genetic algorithm are the followings: initializing a population, evaluating it and looping until converging: performing crossover, mutation, assessing fitness, replacing members, and testing for convergence.

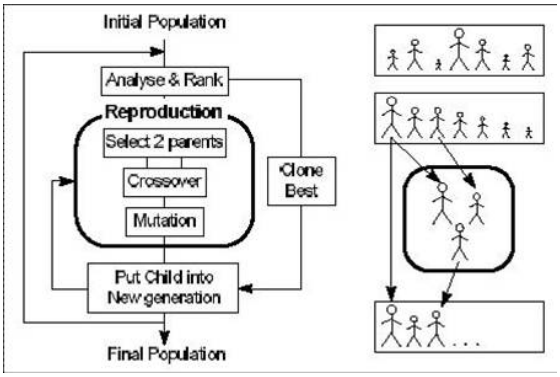

Figure 19: Description of an EA [28].

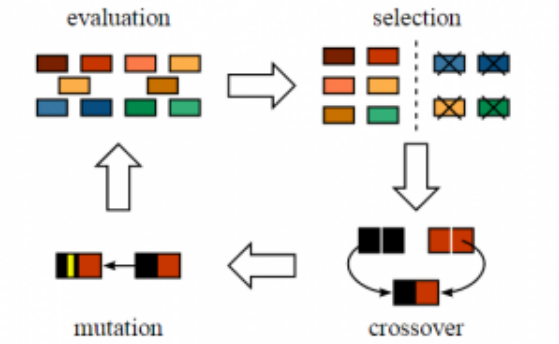

Figure 20: Steps and operators of a GA [9].

4.5. Optimization set-up

To implement the gradient methods, it has been used `conmin_mfd` and `opt_newton`. These methods have been implemented only with PARSEC as they are not able to manage discrete variables which NACA uses (see optimization usage guidelines in the Dakota documentation[7]). Scaling was also tried and numerical gradients and Hessians were computed for some cases.

`soga` and `coliny_ea` were chosen to execute the genetic and evolutionary algorithm. It has to be highlighted the wide range of possible configurations these methods have. In this work, it has been studied the influence of some of their settings in the results.

Many of the possible configurations in Dakota for each method were studied as part of the initial design of experiments and will be analysed later on.

[7] https://dakota.sandia.gov/sites/default/files/docs/6.17.0-release/user-html/usingdakota/studytypes/optimization.html

5. Initial design of experiments

Before facing the objective of the optimization declared in section 1.2, several approaches were taken. These experiments focused on understanding the geometry parameterization and the Dakota configuration for each method. At this stage, the flow conditions were M = 0.3 and Re = 1e+06.

5.1. Initial sampling results

First of all, the design space of the PARSEC parameterization method was studied.

PARSEC limits

To search for the PARSEC variable's feasibility limits, the variables were split into pairs. Divided into pairs a parametric study and LHS sampling were performed for each pair, fixing the rest of the variables. To visualize the response of some functions to these pairs of variables, contour plots were chosen. They are useful to visualize the value of the objective function depending on the pair of variables chosen. Moreover, comparing the two studies (parametric and LHS) with these variables pairs allow us to rate which performs better in our model. The functions evaluated were C_d, C_l and C_m. These studies were performed for the full set of PARSEC parameters. As a result, the PARSEC variables limits were defined as stated in 2.

Improving data visualization

Nevertheless, analysing the variables in pairs is not time-efficient. With better data visualization options, we can get valuable results. With these methods, we can see how the combination of multiple variables influences the response functions. In addition, the sampling methods can be compared better. On the one hand, we have seen in figure 17 the structured distribution of the parametric study. On the other hand, in figure 45 we have the randomized LHS for the full PARSEC parameters. For these visualization options, it was used `streamlit`[8] and `plotly`[9].

In both figures 45 and 46, each point (or line) represents a different combination of geometry and AoA. Its colour is in this case its C_d value. In figure 45, the last three rows (and columns) are response functions: C_d, C_l and max_{th}, whereas in the figure 46 they are represented in the right three columns. The maximum thickness is a response function as at first the thickness of the foil using PARSEC is unknown, but it could be approximated by the difference between the parameters Z_{up} and Z_{lo}. Note that the Z parameters are called y for the rest of this work. In the left columns, we find the number of evaluations out of 600 samples, which is the ID of each foil and AoA. Both data views are interactive. Zooming in any region of figure 45 or selecting ranges in the figure 46 bars it can be seen how different combinations of variables influence the response functions.

The sampling provided above is a 600-sample LHS, which took 20 minutes approximately. When the geometries are irregular, sometimes XFoil can stay on error open, even after having reached a solution. For this reason, an XFoil time limit of 9 seconds was imposed. A full PARSEC parametric study has not been performed because it will require at least 4096 evaluations, as described in the paragraph 4.1. The LHS method intends to describe a whole design space

[8] https://streamlit.io/
[9] https://plotly.com/python/

with a user-defined number of samples. For these reasons, LHS is preferred for this work as the number of variables starts to be significant.

5.2. Gradient methods initial results

In this project, two gradient-based methods have been used, CONMIN MFD and OPT++ Newton Method.

5.2.1. CONMIN

Firstly, it was tried to achieve an optimum foil from an approximation of a NACA 0012 built with PARSEC [31], aiming for a minimum drag only using an alternative CONMIN method `conmin_frcg`. Then `conmin_mfd` allows to set up non-linear constraints, which consists in looking for a lift coefficient of at least 0.4 and a minimum thickness of 10% of the chord. In both cases, the algorithm could not find a feasible direction.

Continuing with `conmin_mfd`, the first approach to solve the issue was changing the initial point. NACA 0012 is not a thick profile nor has a huge base drag, in addition, the initial point influences heavily gradient optimization. A thicker profile, which can be found in figure 21, was tested then but the result would be the same. Despite the fact Dakota output was showing that the finite difference gradient was being calculated with relative steps for each parameter, the scaling option was toggled. It raised the same issue: the algorithm still could not find a feasible direction. In light of the unpromising results, it was tried to increase the step size for the finite difference gradient, which seemed to be the origin of the issue as the gradient was being measured between adjacent points. This option produced the same problem.

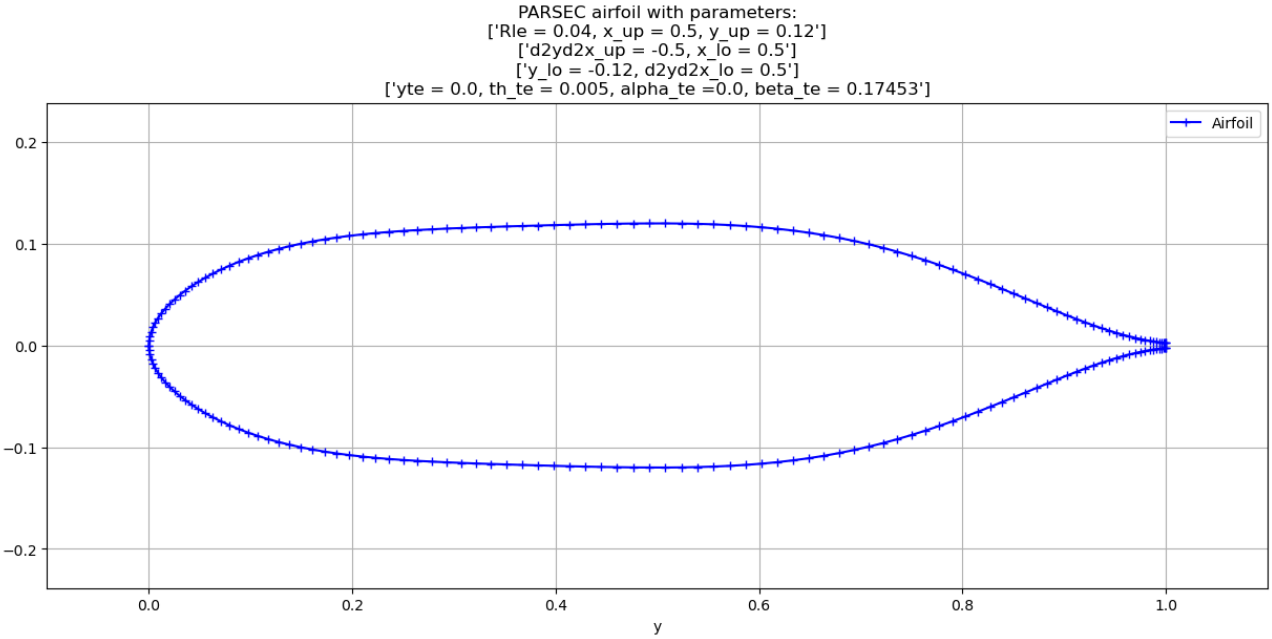

Figure 21: Initial geometry for the gradient-based optimizations with PARSEC.

To avoid the curse of dimensionality, the design variables were reduced from 12 to 9, controlling some aspects of the profile but being able to produce cambered foils. The omitted parameters were: y_{te}, th_{te} and $beta_{te}$. Moreover, the geometry generator programme was revised

to introduce stronger constraints inside it. This was implemented for the purpose of penalising heavier thick foils. As a consequence, the profiles are to be between 30 and 10 per cent of the chord. These intents had no improvement. The algorithm evaluated the objective function and constraints 31 times and calculated gradients 6 times. It wrote down 31 points for which it performed 67 function evaluations taking 262.78s. The termination criterion for the algorithm was again that it was not able to produce a feasible design. The reported results were geometries with exactly the same parameters.

Finally, the optimization proposed for this work was approached, searching for symmetric profiles with fixed thicknesses at the TE. This approach was successful and the results are presented below.

5.2.2. OPT++ Newton Method

At first, the implementation of the Newton method using the OPT++ package in Dakota was not successful neither. It was approached after reducing the number of variables and it was also tried with multiple step sizes. The algorithm was able to write down 36 points but it needed 1656 function evaluations which took 1973.14s. Moreover analysing the results, the last profiles did not satisfy the constraints. Nevertheless, it was able to report different profiles even though once again these geometries were not optimum at all.

After setting the study case of this project, the Newton method returned already a good result. The optimizations using the Newton method were at first executed by `optpp_fd_newton` in which the calculation of the Hessian is performed by the algorithm. For this reason, Dakota could not use the advantage of the asynchronous evaluation concurrency. This optimization performed 4462 function evaluations (2590 new, 1872 duplicate) and took 5571.94s

The final optimization using the Newton method was performed using `optpp_newton` to exploit the evaluation concurrency function of Dakota. Its results will be presented later in this work.

5.2.3. Gradient-based method set-up in the optimization

Some conclusions were deducted from the results of both methods:

- The design space is not smooth enough to allow the gradient to work correctly without reducing the number of variables.
- Scaling the variables does not improve the results because the algorithm manages already relative values for each variable range.
- Setting the thickness of the TE as a variable in the symmetric case holds the algorithm to reach a feasible direction complying with the constraints.
- The optimization result is heavily dependent on the starting point.
- The step size for gradients was released and managed by Dakota and the programme.

The NACA formulation was not used in the gradient-based optimization because the variables were defined as integers or strings in the case of the NACA 6-series which are not supported by the chosen algorithms.

5.3. GA and EA initial results

As has already been mentioned, two ES have been used: SOGA and COLINY EA.

5.3.1. SOGA

After the intent of implementing gradient-based optimizers with PARSEC, a SOGA was executed. At first, it was tried with the complete set of PARSEC parameters. Proceeding as in the case of the earliest descent methods implementation, minimum drag was only pursued. This method was able to iterate in a wider range of profiles than a gradient method and was not sensitive to the initial point. It produced a total of 752 combinations of geometries and angles of attack and its execution time was 1123.1s. Although we might think it performed well, Dakota's election as the best result does not look like a typical foil profile, it can be seen in figure 7. This geometry at an angle of attack of 0.72 is supposed to have a drag coefficient of 0.003 according to the XFoil solution.

This "foil" was extremely thin and irregular. This geometry could not be managed well in XFoil despite it returning a solution. To improve the results, the same steps that were followed in the gradient-based optimizations were implemented: the thickness of the foil was bounded between 30 and 10 per cent of the chord and the number of variables was reduced to 9 and finally to 7. Even though the first results using `soga` were not perfect, they were useful to observe how the algorithm works. With the data visualization improvements, it could be seen that the algorithm reached convergence in the response function C_D.

To avoid the curse of dimensionality once again, the number of variables was set to 9. This way, cambered foils were being looked for but with some design restrictions, these parameters were fixed to a sensible value. Moreover, the C_L non-linear constraint of more than 0.4 was imposed. Dakota produced a total of 450 evaluations in 671.42s.

In figure 47 it can be observed how:

- C_D reached convergence again.
- C_L in the last generations/evaluations tend to be near the given constraint.
- max_{th} in the last generations foils tend to be lower than the initial values.

Using the parallel coordinates view, selecting the lowest drag configurations and the highest lift coefficients, we can see which combinations have led the algorithm to these results and when were they evaluated:

Some patterns can be observed:

- r_{le} value is the highest or the lowest in its range.
- y_{up} coordinate around 0.1.
- d^2y/dx^2_{up} around -0.5.
- y_{lo} near the maximum in its range.
- d^2y/dx^2_{lo} near 1.
- $alpha_{te}$ between 0.2 and 0.4 or -0.2 and -0.4.
- AoA higher than 1, most of them at 2 degrees.

- max_{th} of the best geometries is lower than 20% of the chord.

  Nevertheless, the geometries still not be clean, they were irregular or thick.

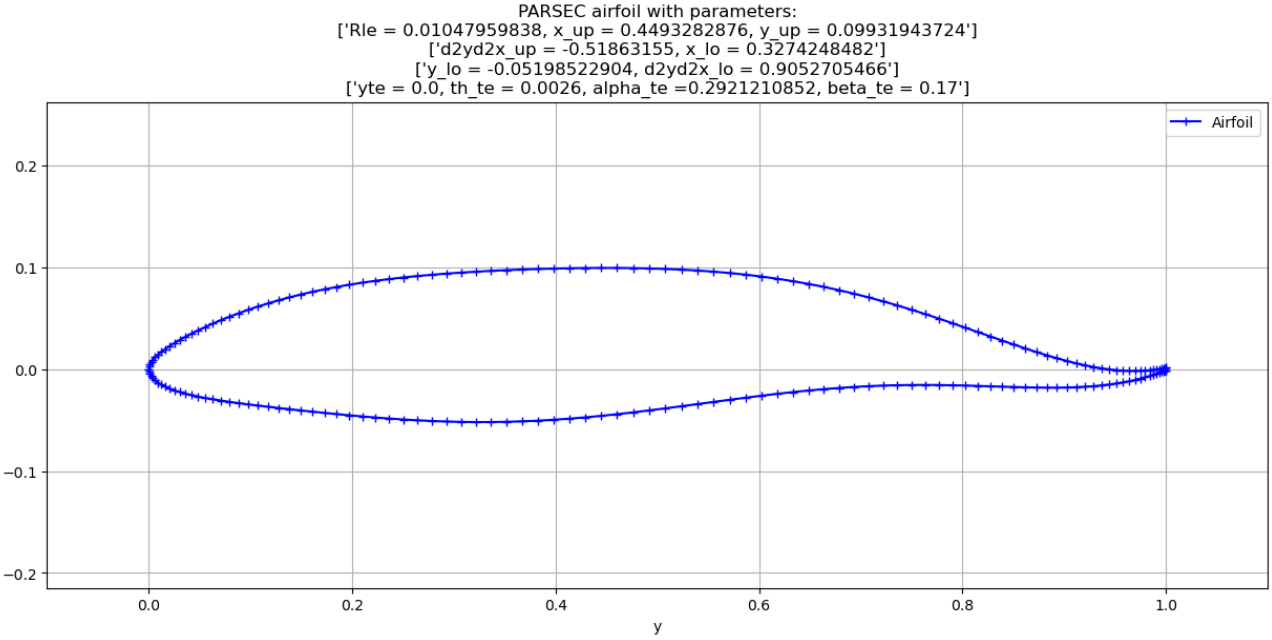

Figure 22: PARSEC geometry that almost complies with the lift constraint and has minimum drag.

At that point, the optimization was producing better results, thus the algorithm's parameters were changed to approach the configuration which would provide more results satisfying the lift constraint. The parameters that were changed to modify the behaviour of the algorithm were:

- `fitness_type` determines how individuals are compared.
- `num_generations` affects the termination criteria of the algorithm, delaying its stopping as it is increased.
- `percent_change` affects the termination criteria of the algorithm, delaying its stopping as it is decreased.
- `constrain_penalty` in the merit function of each profile, not satisfying the constraint is multiplied by the input value.
- `convergence_type` defines how the fitness of the population can be computed, an average of the generation or the best-fitted of the individuals.
- `population_size` defines the number of each population within the merit function is evaluated.
- `replacement_type` instruct how the algorithm replaces the individuals between each generation.
- `crossover` controls how the crossover between individuals occurs and its probability.
- `mutation` equal to `crossover`, controls the mutation events.

- `convergence_tolerance` set the tolerance that indicates convergence.

It was noted that increasing the population size, the function evaluations increased but the results improved: more foils met the minimum lift while keeping the drag low. The convergence tolerance and `num_generations` have a direct relationship as well with the results and the number of function evaluations. The smaller the tolerance, the longer the optimization but the better the results, The number of generations set the minimum number of consecutive generations that have to change less than the percent change to stop. Crossover and mutation could help the algorithm to reach a wider range of profiles faster.

As is this algorithm able to manage non-continuous variables, the angle of attack was simplified to a predefined set of real numbers: 0, 0.1, 0.2 and so on. With these changes, we expect to avoid the algorithm populating several profiles with approximately the same AoA.

5.3.2. COLINY EA

Using `coliny_ea` is similar to using `soga` because it has almost the same settings options. Although, it must be highlighted that using `coliny_ea` the algorithm did not reach any stopping criteria apart from the maximum function evaluations. This fact indicates that `coliny` "learns" slower than `soga`. The main differences between both evolutionary strategies can be found in some of their default settings and the fact that COLINY EA performs crossover and mutation only on a selection of the best individuals of each population.

6. Results

In this chapter, the results of this project will be presented. The optimizations will try to solve the problem stated at 1.2. The best results are presented down below.

All the optimizations have been evaluated with XFoil. Typically, a foil evaluation and communication results to Dakota takes less than 2 seconds using 4 cores out of 6 on an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2601 Mhz, 6 cores, 12 threads.

6.1. Gradient methods optimization

Hereafter, the optimizations with PARSEC geometries and gradient methods are presented. These optimizations were local having as the initial point the geometry already mentioned in figure 21.

6.1.1. CONMIN Method of feasible directions

The local optimization using the method of feasible direction that performed better for our objective, had the thickness of the TE fixed. It ran 76 evaluations in 352.33s. The tolerance to convergence and to the lift constraint was 1e-03. Numerical forward difference gradients were computed by Dakota.

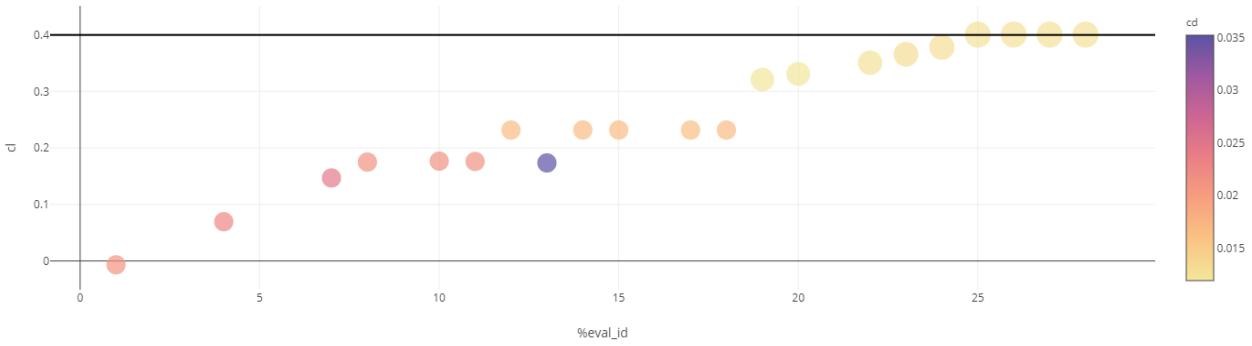

Figure 23: C_L vs. evaluation, color by C_D, size by AoA of the CONMIN MFD optimization with PARSEC.

The optimization tends to have the AoA high keeping the geometry relatively thin, reaching a local optimum.

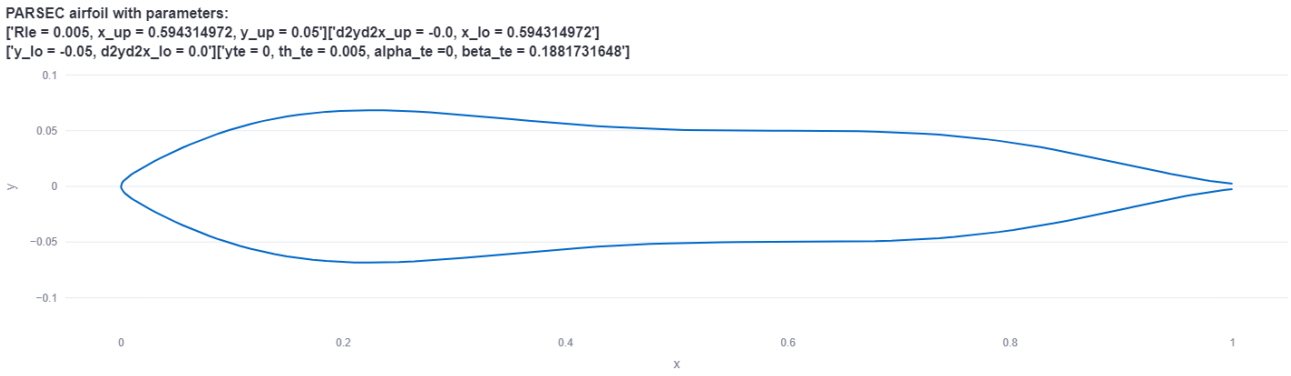

Figure 24: Optimum geometry of the CONMIN MFD optimization with PARSEC.

This profile at 3.6368 degrees of AoA, has a drag coefficient of 0.0126 and a lift coefficient of 0.4006.

6.1.2. OPT++ Newton method

The optimization using the Newton method was executed by `optpp_newton` and Hessians were calculated by Dakota. This way the optimization performed 714 new function evaluations in 1623.02s.

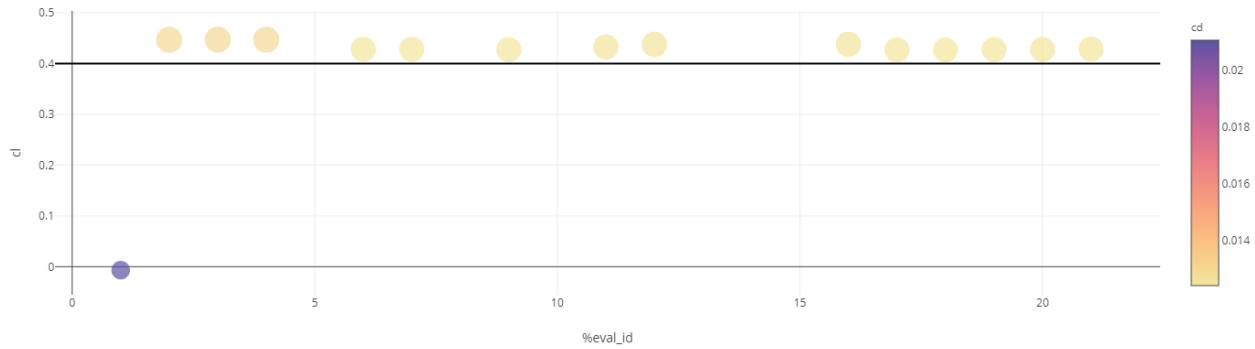

Figure 25: C_L vs. evaluation, color by C_D, size by AoA of the OPT++ Newton optimization with PARSEC.

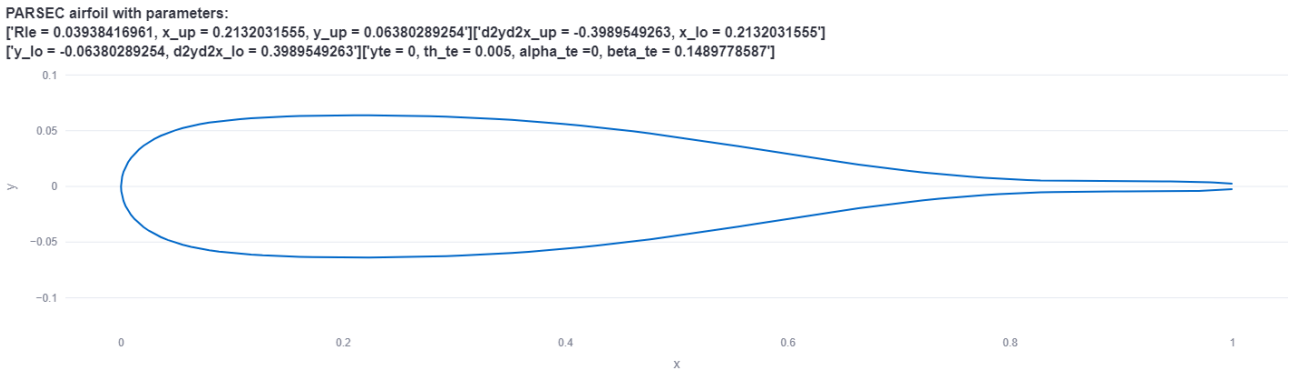

Figure 26: Optimum geometry of the OPT++ Newton optimization with PARSEC.

This profile at 3.6111 degrees of AoA, has a drag coefficient of 0.01243 and a lift coefficient of 0.4283.

6.2. Evolutionary algorithms

NACA 4-modified and NACA 6 was used in the optimizations with the EAs algorithms, in addition to PARSEC parameterization. These optimizations report more results comparing the gradient-based optimizations. For this reason, selecting the optimum is not that simple. The optimum geometries and AoA combination below, are selected manually and subjectively after analysing the data.

6.2.1. NACA 4-m

The optimization with the NACA 4-m formulation was performed using `soga`. It took 1799.26s and evaluate 896 cases. With the amount of data reported, it is interesting to analyse the results using the data visualization options mentioned at 5.1.

Some patterns can be seen within the foils at the Pareto front:

- The thickness is the lowest possible: 10% or 11%.
- The LE index is between 2 and 4 being the most common within the best foils 3.
- The most repeated position of the maximum thickness is 4.
- The TE is thin, under the 1% of the chord.
- The AoA is between 3 and 4 degrees.

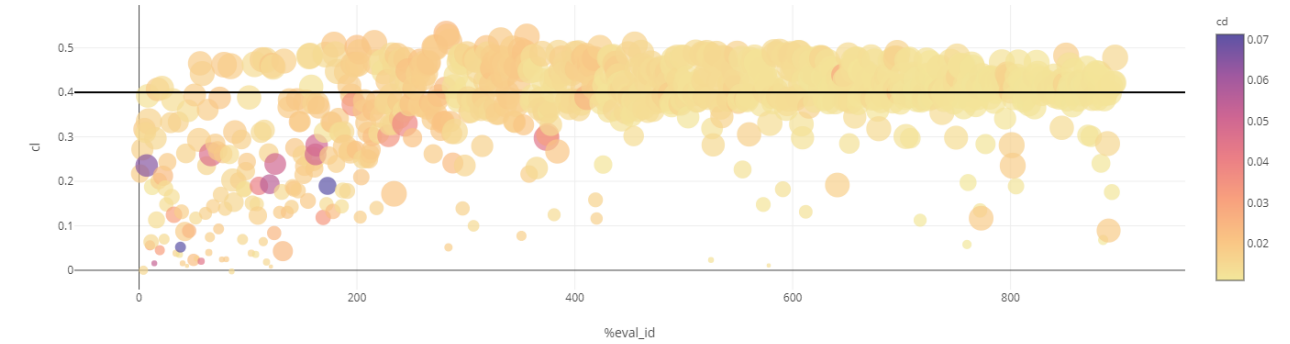

Figure 27: C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with NACA 4-m.

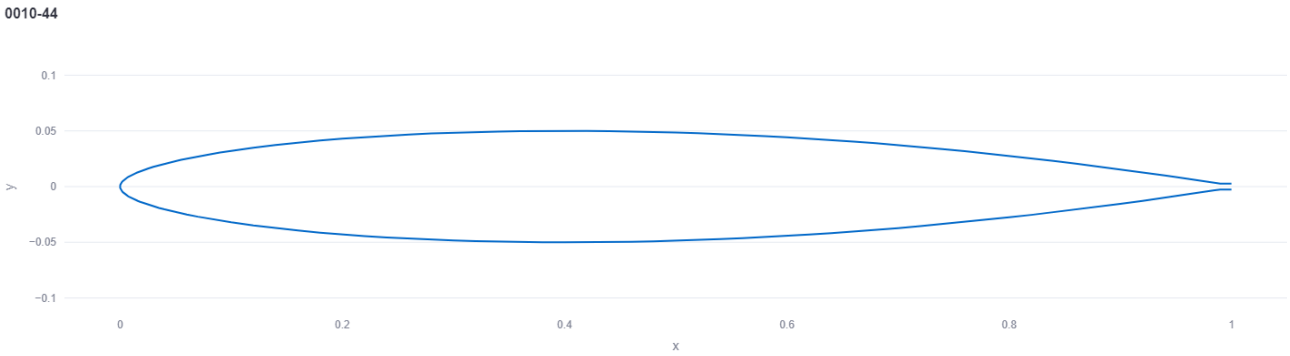

Figure 28: Optimum of NACA 4-m optimization.

This geometry 0010-44 at 3.7 degrees of AoA was the evaluation number 855 and it has a drag coefficient of 0.01142 and a lift coefficient of 0.4159.

6.2.2. NACA 6

The best optimization using the NACA 6-digits formulation was performed by the `coliny_ea` `algorithm`. This algorithm evaluated the functions 1504 times and it took 3020.19s.

It can be observed how the algorithm could not identify some visible relationships between some variables and the functions. It did not find a relationship between the thickness of the TE and the C_D along the evaluations nor the relationship with the 6 or 6A series, although this is not clear even after analysing the results. Even though, it can be highlighted that:

- The thickness is the lowest possible: 10% or 11%.
- The 6 series is dispersed between the possible values, as it can be seen in figure 30.
- The TE is thin, under the 1% of the chord.
- The AoA is between 3 and 4 degrees.

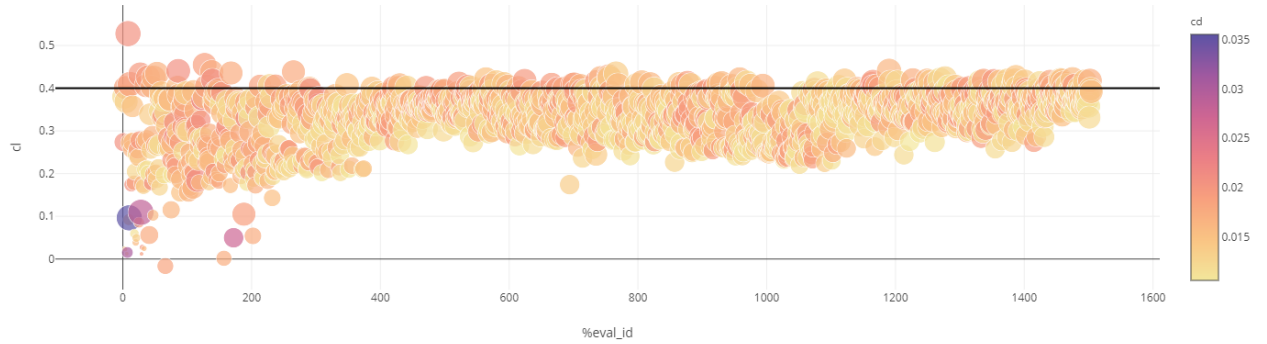

Figure 29: C_L vs. evaluation, colour by C_D, size by AoA of the COLINY EA optimization with NACA 6.

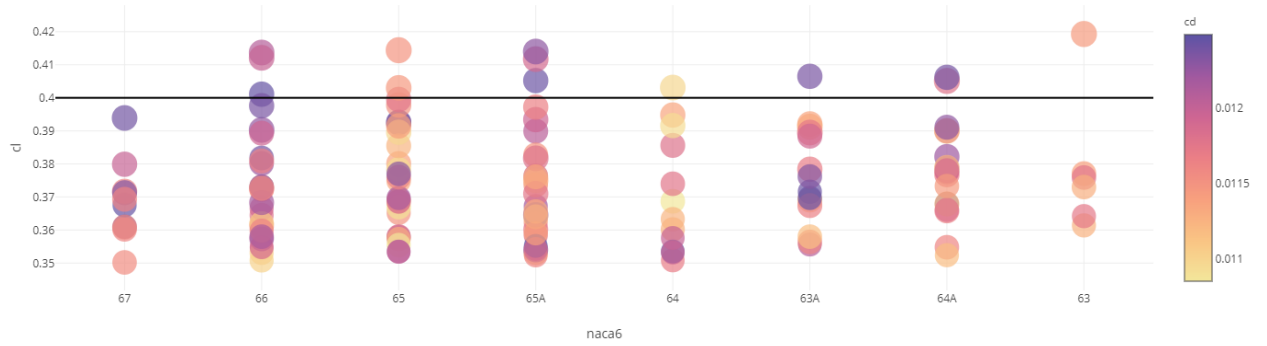

Figure 30: C_L vs. NACA 6 series, colour by C_D, size by AoA of the optimization.

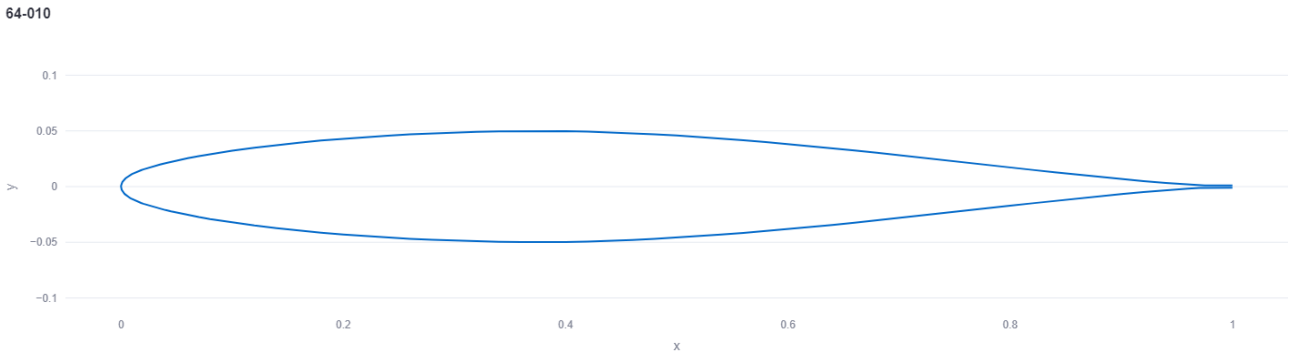

Figure 31: Optimum of NACA 6 optimization.

This geometry at AoA 3.5 degrees was the evaluation number 1283 and it has a drag coefficient of 0.01097 and a lift coefficient of 0.4031.

6.2.3. PARSEC

The optimization with PARSEC was executed by soga algorithm. It evaluated the function 1707 times and took 3225.06s.

Some relationships between variables and the objective function and constraint can be observed in figure 51.

- Low r_{LE} are related with low C_D.
- Thin geometries has a low C_D, y_{up} is low as a result.
- The position of the control point of the upper and lower surface has a parabolic relation with the C_D with the minimum at the middle point of the chord
- The TE must be thin for a low drag coefficient.
- The AoA has to be between 3 and 4 degrees to comply with the lift constraint.

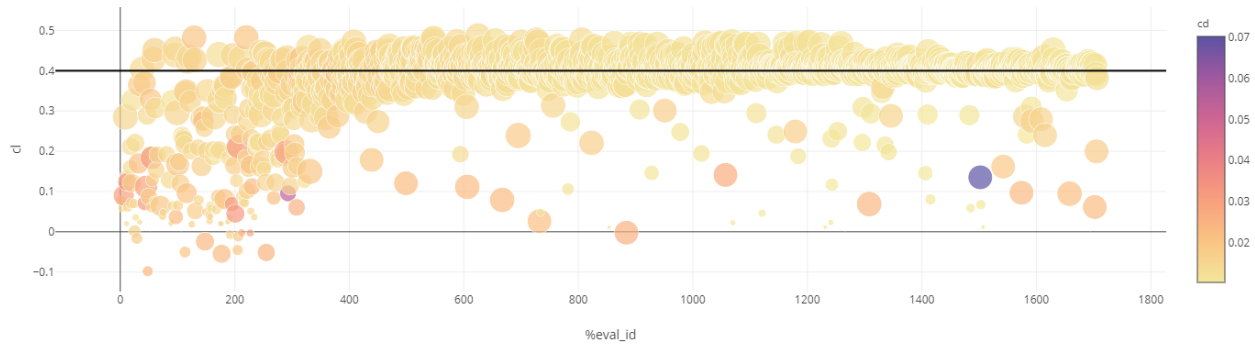

Figure 32: C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with PARSEC.

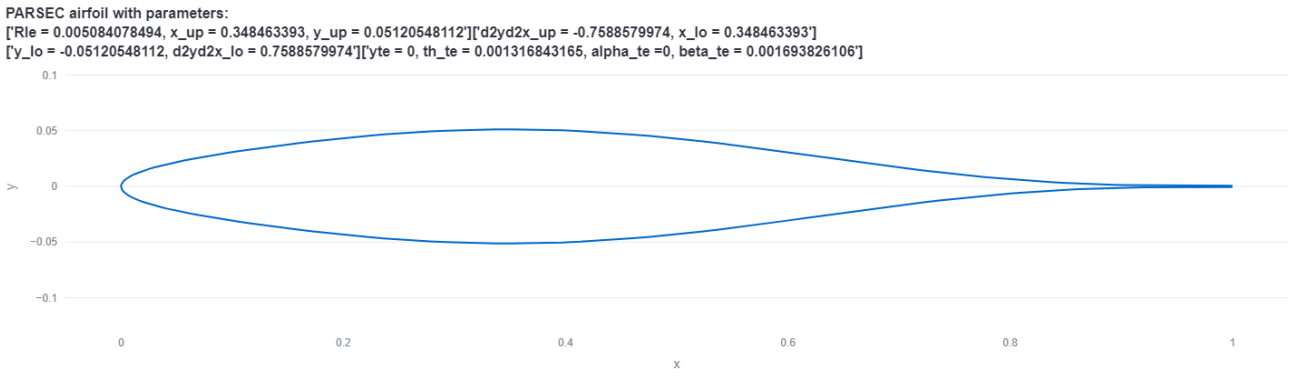

Figure 33: Optimum of PARSEC optimization.

This geometry at AoA 3.5 degrees was the evaluation number 1686 and it has a drag coefficient of 0.01078 and a lift coefficient of 0.4036.

6.2.4. Comparison of optimums from the different methods of optimization and geometry parameterization

In table 4 the best optimization results for each methodology have been summarised:

Geometry parameterization	Optimization method	C_D	C_L	Evaluations	CPU time (s)
PARSEC	**Gradients**				
	conmin_mfd	0,01257	0.4006	76	352.33
	optpp_newton	0.01243	0.4283	714	1623.02
	EA				
	soga	0.01078	0.4036	1707	3225.06
NACA 4m	soga	0.01142	0.4159	896	1799.26
NACA 6	coliny_ea	0.01097	0.4031	1504	3020.19

Table 4: Comparison between all the optimums from the different optimizations.

6.3. Comparison with Star-CCM+ results

In order to estimate the validity or accuracy of the results, from the ES optimizations for each parametrization method, some of the best combinations have been run in Star-CCM+ using the set-up described in section 3.3.2. The process from preparing the folder for each simulation to running and tabulating the results takes between 100 and 200 seconds. The solver elapsed time per iteration is less than 0.1s and the total solver elapsed time is usually less than 1 minute depending on the number of iterations. The rest of the time is wasted in the server opening and closing processes in Star-CCM+. The simulation is run in parallel using 15 cores on an AMD EPYC 7V12 64-Core Processor, 2445 Mhz, 16 Core(s) and 32 Logical Processor(s), so the solver CPU times are the values of the elapsed time multiplied by the number of processors used in the simulation. Taking into account that the number of geometries that are going to be simulated is approximately a hundred, the whole process usually takes between 12000 and 16000 seconds (3-4 hours).

Because of the nature of the optimization results, the best geometries plus the AoA tend to be the same or very close. For this reason, is difficult to stand out correlations between the variables and the results from Star-CCM+ because the whole design is not being simulated.

The difference between both solvers is expressed in % following:

$$\Delta C_X = \left[\frac{C_{X_{hf}} - C_{X_{lf}}}{\frac{C_{X_{hf}} + C_{X_{lf}}}{2}} \right] \cdot 100 \tag{30}$$

where C_X can be C_L or C_D and the sub indexes hf and lf means high fidelity (Star-CCM+) and low fidelity (XFoil) respectively.

The simulation stops when asymptotic convergence at the force coefficients is reached and all the residuals are less than 1e-03 or the number of iterations is equal to 1500. The simulations have usually a y+ lower than 0.1 on most of the surface. The pressure distribution of XFoil has been compared for random individuals with their pressure distribution of Star showing an almost perfect match. These comparisons and results can be found in the section 9.2.

6.3.1. PARSEC

With the best optimization using PARSEC (which was the one executed by the `soga` algorithm), 126 combinations were evaluated in Star-CCM+. The differences between the XFoil drag coefficient and lift coefficient between the Star-CCM+ ones were evaluated:

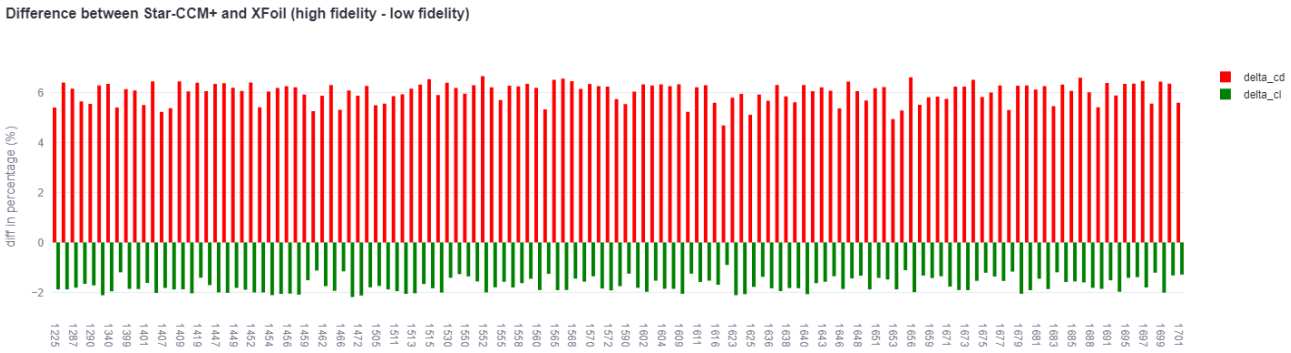

Figure 34: Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in PARSEC geometries.

The difference between the high-fidelity results and low-fidelity results is computed as indicated in equation 30. It can be observed that XFoil calculated a drag coefficient lower than the one calculated by Star whereas to the lift coefficient, the opposite happened. On the one hand, the mean of the deviation for the drag coefficient was 6.0073% and the standard deviation was 0.4033%. On the other hand, the mean of the lift coefficient was -1.7028% and the standard deviation 0.2882%.

These differences are evident if the low-fidelity results are plotted with their respective high-fidelity ones:

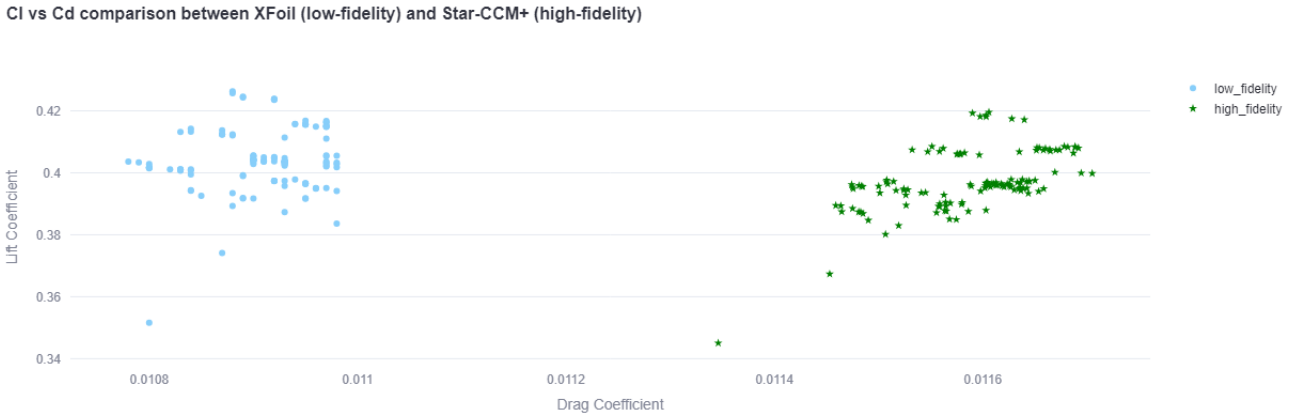

Figure 35: C_L vs. C_D of XFoil results and their respective Star-CCM+ results of PARSEC geometries.

6.3.2. NACA 4m

83 combinations of NACA 4m geometries were evaluated in Star-CCM+.

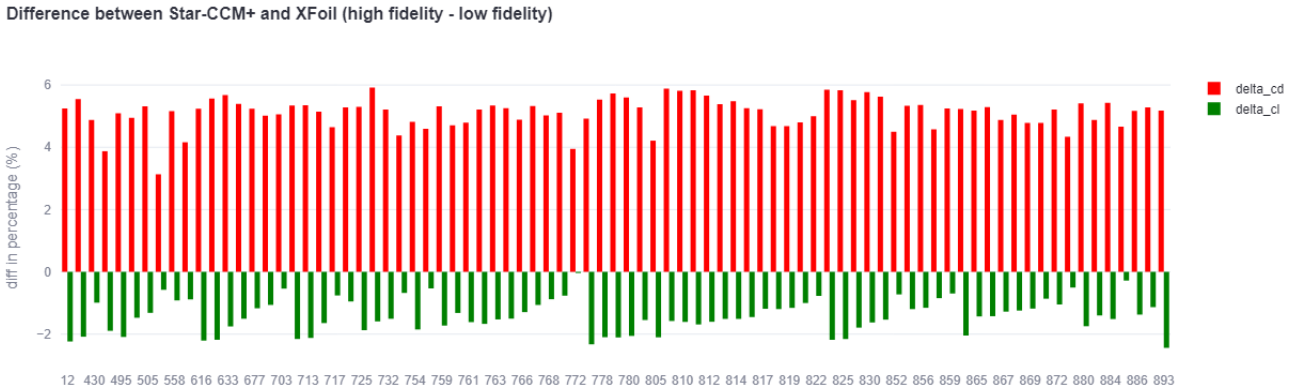

Figure 36: Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in NACA 4m geometries.

It can be observed that XFoil calculated a drag coefficient lower than the one calculated by Star whereas to the lift coefficient, the opposite happened again. On the one hand, the mean of the deviation for the drag coefficient was 5.1179% and the standard deviation was 0.4886%. On the other hand, the mean of the lift coefficient was -1.4131% and the standard deviation 0.5265%.

These differences are evident if the low-fidelity results are plotted with their respective high-fidelity ones:

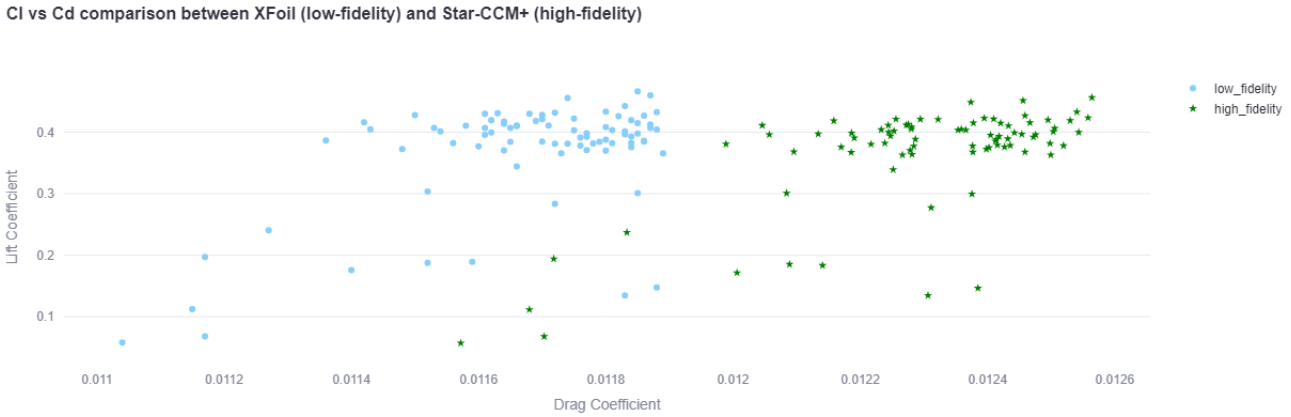

Figure 37: C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 4m geometries.

6.3.3. NACA 6

107 combinations of NACA 6 geometries were evaluated in Star-CCM+.

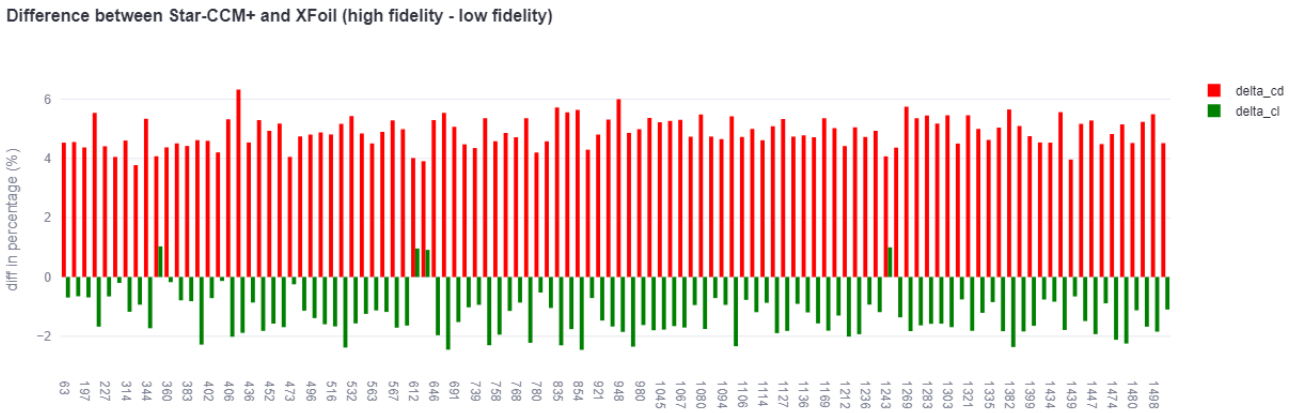

Figure 38: Difference between the Star-CCM+ coefficients and the XFoil coefficients in percentage in NACA 6 geometries.

It can be observed that XFoil calculated a drag coefficient lower than the one calculated by Star whereas the lift coefficient delta is negative in most of the cases. On the one hand, the mean of the deviation for the drag coefficient was 4.9011% and the standard deviation was 0.4903%. On the other hand, the mean of the lift coefficient was -1.3315% and the standard deviation 0.7173%.

These differences are evident if the low-fidelity results are plotted with their respective high-fidelity ones:

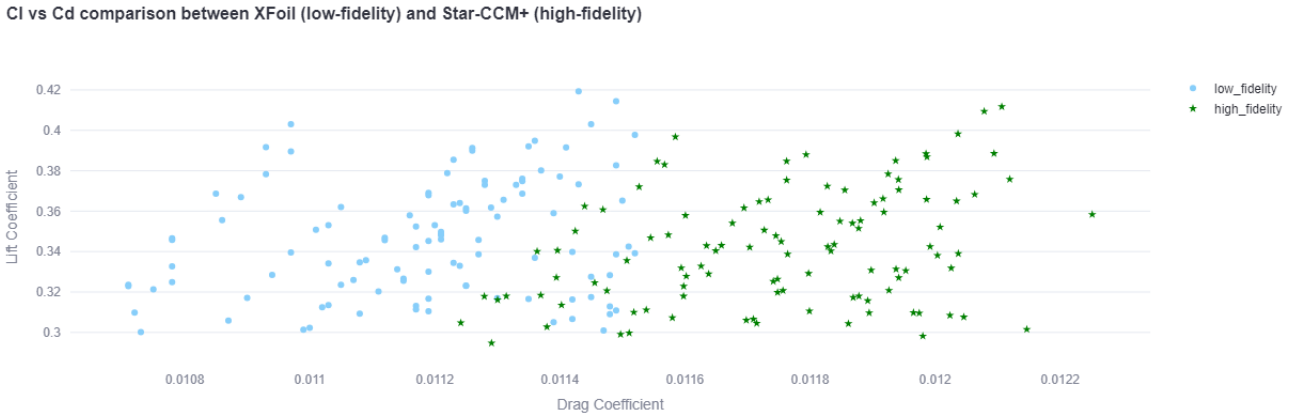

Figure 39: C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 6 geometries.

6.3.4. Summary of comparison between Star-CCM+ and XFoil

The statistics for each method are presented below.

Geometry parameterization	$\mathbf{\Delta C_D}$ (%)	$\mathbf{\Delta C_L}$ (%)
PARSEC	$\mu = 6.0073$ $\sigma = 0.4033$	$\mu = -1.7028$ $\sigma = 0.2882$
NACA 4m	$\mu = 5.1179$ $\sigma = 0.4886$	$\mu = -1.4131$ $\sigma = 0.5265$
NACA 6	$\mu = 4.9011$ $\sigma = 0.4903$	$\mu = -1.3315$ $\sigma = 0.7173$

Table 5: Comparison between differences between high-fidelity results and low-fidelity for each geometry parameterization method.

In addition, the previous optimums can be re-evaluated taking into account the Star-CCM+ results.

Geometry parameterization	**Optimization method**	$\mathbf{C_D}$	$\mathbf{C_L}$
PARSEC	`soga`	0.01151	0.3972
NACA 4m	`soga`	0.01205	0.4109
NACA 6	`coliny_ea`	0.01158	0.3968

Table 6: Comparison between all the optimums from the different optimizations, Star-CCM+ results.

Regarding the table 6 and the figures 35, 37 and 39, new optimums for the NACA 6 optimization and PARSEC optimization have emerged.

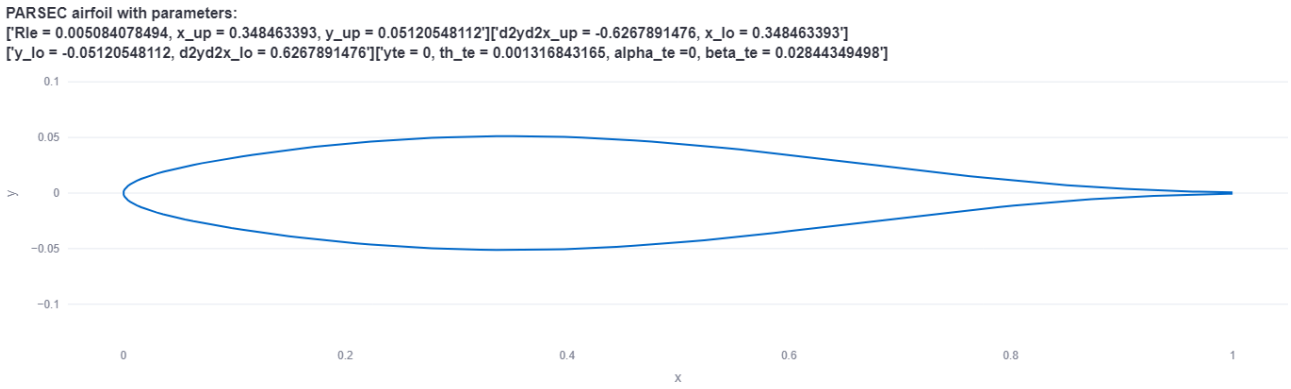

Figure 40: New PARSEC optimum taking into account Star-CCM+ results.

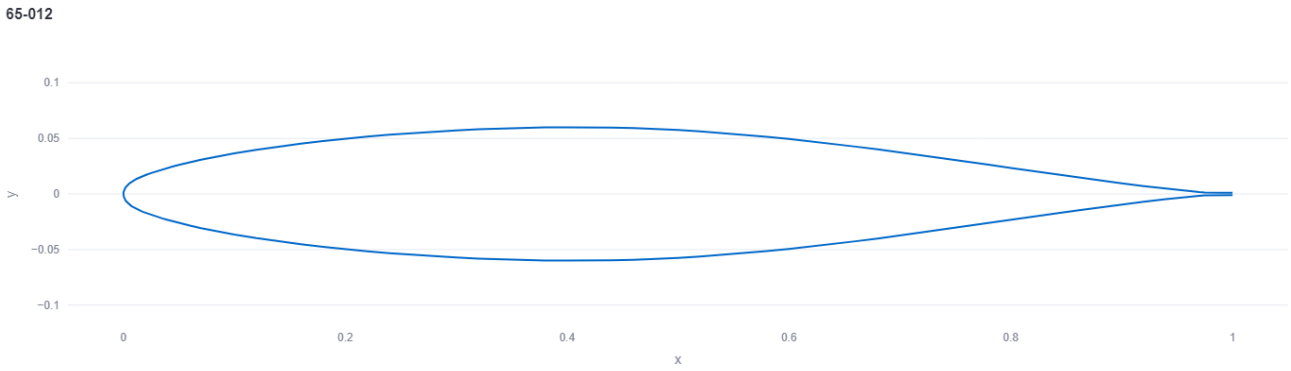

Figure 41: New NACA 6 optimum taking into account Star-CCM+ results.

Geometry parameterization	**Optimization method**	$\mathbf{C_D}$	$\mathbf{C_L}$
PARSEC	`soga`	0.01153	0.4074
NACA 4m	`soga`	0.01205	0.4109
NACA 6	`coliny_ea`	0.01208	0.4094

Table 7: Optimums from the different optimizations taking into account Star-CCM+ results.

7. Conclusions

In this project, a review of profile geometry parameterization methods, optimization algorithms and flow solvers has been performed. This review aimed to test different techniques to optimize foil shapes under different conditions and constraints. For this reason, an optimization tool was developed to meet this mission. The result of this project is a comparison in efficiency and accuracy between the different options of the three topics, parameterization methods, optimization algorithms and flow solvers, and the optimization tool which can be adapted to future research.

In the geometry parameterization, PARSEC, eleven parameters formulation of profile parameterization, and NACA were used in the tool and the results of using these in the optimization process were compared. At the same time, different optimization algorithms were tested. Finally, a percentage of the results of the best optimization was simulated in a commercial CFD solver, which was Star-CCM+. e

Many conclusions can be drawn from this research, depending on which part is focused on. The main focus is on which process (geometry parameterization plus optimization method) performed better.

Table 7 shows the performance of the optimum foils of the algorithm and parameterization method presented, calculated with a high fidelity CFD program. Regarding this comparison, it can be concluded that the best foil for the study case described at the top of this work in chapter 1.2, seems to be the PARSEC optimum of the SOGA optimization. Even before comparing the XFoil results with the Star-CCM+ results, it seemed to be the best. After running Star-CCM+, a new optimum needs to be selected as some of the previous ones were not meeting the requirements. Subsequently, comparing the newest optimums relying on the Star-CCM+ results, the PARSEC still to be the best.

However, it can be observed how the 3 profiles look alike: thin LE, thin TE and the maximum thickness between 0.3 and 0.5 of the chord.

Regarding the geometry parameterization and taking into account the best results, not only seems PARSEC to be the best for the objective of this work, it has visible advantages to the NACA formulations. It provides more control over the geometry which allows the user to set more constraints, like shape constraints because of structural reasons, and still could modify the rest of the surface of the geometry. Future research could compare these results with an optimization using other parameterization methods for example B-Splines.

Respecting the optimization algorithms, for the study case, the global methods have dominated. On the one hand, the gradient-based method had nothing to do in light of the unawareness of the objective function and constraints in the design space, selecting an initial point attaches the results to a specific region of the design space. They were included in this work in order to learn and measure their operation. As expected, the Newton method is much more expansive than the MFD and the result was only a 1% better than the MFD optimum. Both optimums were different in contrast to both ES optimizations performed in which the optimums were similar.

On the other hand, the chosen ES and the selected configuration make both algorithms operate analogously. Besides the fact that `coliny_ea` did not reach the convergence criteria in this work, they provide almost the same results but `soga` is preferred in 2 out of 3 cases.

An example of optimization data can be visualized in figure 42 below and more detailed in figures 49, 50, 51. Future research on this topic could focus on the optimization algorithms configuration in order to look for a better approach.

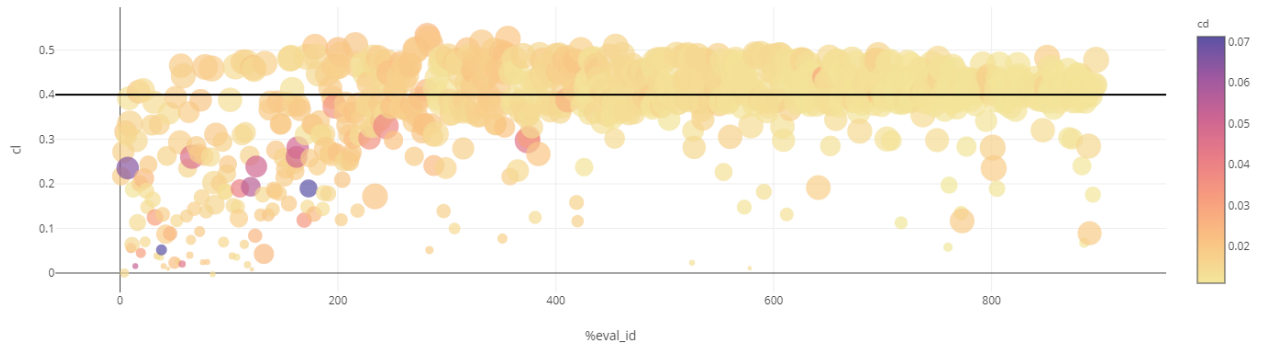

Figure 42: C_L vs. evaluation, colour by C_D, size by AoA of the SOGA optimization with NACA 4-m.

Measuring the whole process performance, and optimization in a low-fidelity or fast-flow solver like XFoil and then verifying the best results with a RANS-based CFD, can save up to the 98% of time that an optimization with the RANS-based CFD could take depending on how many results are running in the RANS CFD. 1500 evaluations optimization longs $\sim$ 3000 s with XFoil, a simulation in Star-CCM+ is $\sim$ 120 s. In this study case, the lack of accuracy of the results (concerning the high-fidelity results) has been only a 6% maximum. This difference, even though it is not negligible, is enough to trust the optimization process. In addition, in figure 56, we can see that the pressure distributions of both solvers are overlapping which supports the first optimization results. Hereafter a typical difference between the low-fidelity and high-fidelity results is presented.

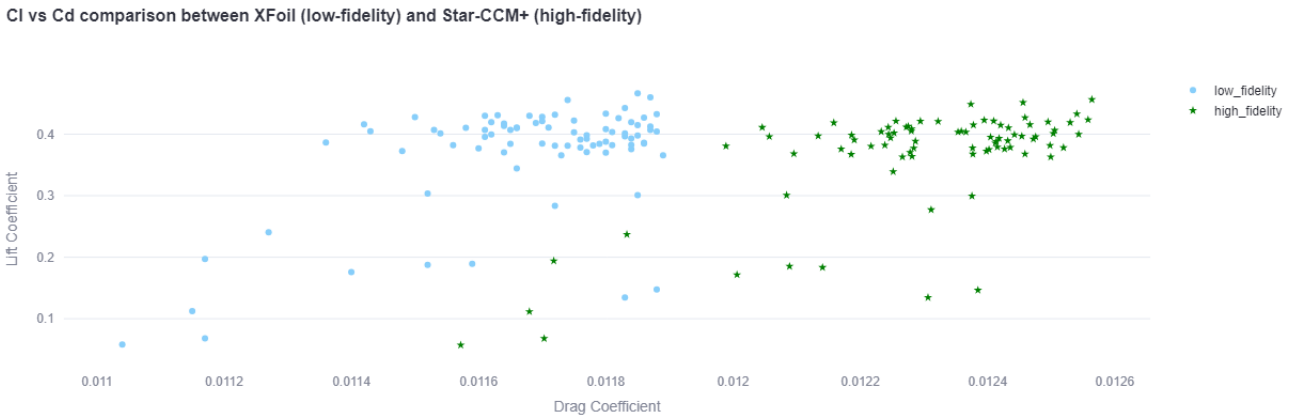

Figure 43: C_L vs. C_D of XFoil results and their respective Star-CCM+ results of NACA 4m geometries.

The evaluation of the optimum with the high-fidelity CFD, force to re-evaluate if the

previous optimum foil still to be the optimum. After the re-evaluation, the final optimums are presented below:

Geometry parameterization	**Optimization method**	$\mathbf{C_D}$	$\mathbf{C_L}$
PARSEC	`soga`	0.01153	0.4074
NACA 4m	`soga`	0.01205	0.4109
NACA 6	`coliny_ea`	0.01208	0.4094

Table 8: Final optimums from the different optimizations taking into account the multi-fidelity results.

Between this individuals, as it has already been mentioned, PARSEC and `soga` result was the best:

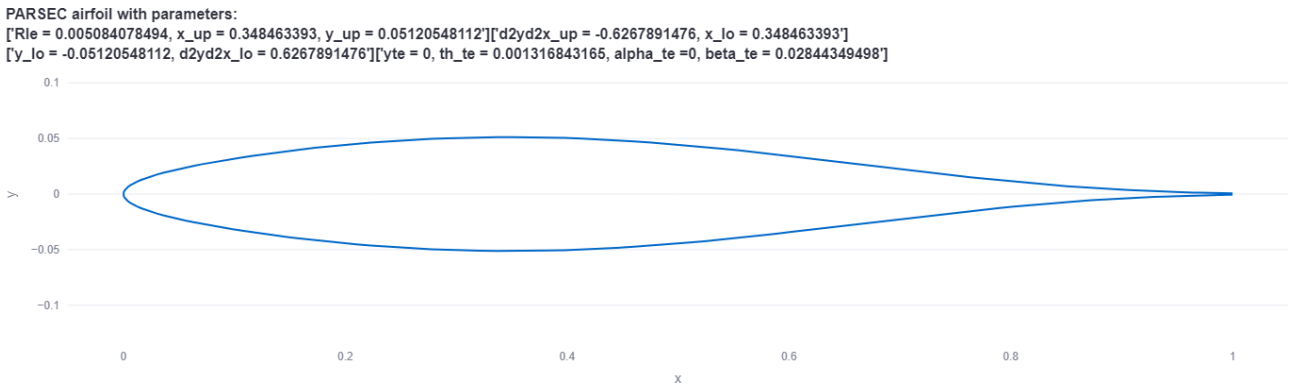

Figure 44: Final PARSEC optimum taking into account the multi-fidelity results.

In conclusion, the process implemented in this optimization is a powerful approach to optimising the aerodynamic or hydrodynamic performance of a design when the design space it is not intensively studied. As it was mentioned in the introduction of this work, it is a practice that is becoming more common within the industry. The full potential of this multi-fidelity and global optimization using Evolutionary Algorithms, can be delivered in research and development tasks.

7.1. Future works

Nevertheless, to improve this methodology others could emerge, for example using AI models. AI or Machine Learing (ML) models can overcome the accuracy of panel methods or velocity prediction programs with a minimum cost compared with the RANS. Along with the method developed in this project, huge databases come up as a result. This databases of the firstly unknown design spaces, can be combined to train these models.

In fact, at Bailardi Engineering S.L., the developed tool in this project is being used by interns in future research to train multi-fidelity ML models to improve optimization.

8. References

[1] I. H. Abbott and A. E. Doenhoff. *Theory of Wing Sections*. 1959. URL https://www.semanticscholar.org/paper/Theory-of-Wing-Sections-Abbott-Doenhoff/12181b8a5e838fd0a461b3d26cd711265ae3d9d5.

[2] M. T. Akram and M.-H. Kim. Aerodynamic Shape Optimization of NREL S809 Airfoil for Wind Turbine Blades Using Reynolds-Averaged Navier Stokes Model—Part II. *Applied Sciences*, 11(5):2211, Jan. 2021. ISSN 2076-3417. doi: 10.3390/app11052211. URL https://www.mdpi.com/2076-3417/11/5/2211. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[3] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, Nov. 1959. ISSN 0096-199X. doi: 10.1109/TAC.1959.1104847. URL http://ieeexplore.ieee.org/document/1104847/.

[4] H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[5] Cape Horn Engineering. FastRig Digital Twin. Technical report, Dec. 2022. URL https://media.licdn.com/dms/document/media/D561FAQHDWPOBP_rt2w/feedshare-document-pdf-analyzed/0/1684393179712?e=1685577600&v=beta&t=ySlqyHU9iym_letmHN2S_Q-oYwAxlKoBXXrebpDlLSU.

[6] R. Carmichael. Algorithm for calculating coordinates of cambered NACA airfoils at specified chord locations. In *1st AIAA, Aircraft, Technology Integration, and Operations Forum*, Los Angeles,CA,U.S.A., Oct. 2001. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2001-5235. URL https://arc.aiaa.org/doi/10.2514/6.2001-5235.

[7] L. Chen. Curse of Dimensionality. In L. LIU and M. T. ÖZSU, editors, *Encyclopedia of Database Systems*, pages 545–546. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_133. URL https://doi.org/10.1007/978-0-387-39940-9_133.

[8] X. Chen and M. M. Kostreva. Methods of Feasible Directions: A Review. In P. M. Pardalos, D. Hearn, X. Yang, A. I. Mees, M. Fisher, and L. Jennings, editors, *Progress in Optimization*, volume 39, pages 205–219. Springer US, Boston, MA, 2000. ISBN 978-1-4613-7986-7 978-1-4613-0301-5. doi: 10.1007/978-1-4613-0301-5_14. URL https://link.springer.com/10.1007/978-1-4613-0301-5_14. Series Title: Applied Optimization.

[9] J. Cheng. Evolutionary Optimization: A Review and Implementation of Several Algorithms, Oct. 2018. URL https://www.strong.io/blog/evolutionary-optimization.

[10] P. Della Vecchia, E. Daniele, and E. DAmato. An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm. *Aerospace Science and Technology*, 32(1):103–110, Jan. 2014. ISSN 1270-9638. doi: 10.1016/j.ast.2013.11.006. URL https://www.sciencedirect.com/science/article/pii/S1270963813002046.

[11] M. Drela and H. Youngren. XFOIL 6.9 User Primer, Nov. 2001. URL https://web.mit.edu/drela/Public/web/xfoil/xfoil_doc.txt.

[12] J. Fliege, L. M. G. Drummond, and B. F. Svaiter. Newton's Method for Multiobjective Optimization. *SIAM Journal on Optimization*, 20(2):602–626, Jan. 2009. ISSN 1052-

6234, 1095-7189. doi: 10.1137/08071692X. URL http://epubs.siam.org/doi/10.1137/08071692X.

[13] M. M. Grigoriev, C. V. Swiatek, and J. A. Hitt. Benchmarking CD-Adapco's Star-CCM+ in a Production Design Environment. In *Volume 7: Turbomachinery, Parts A, B, and C*, pages 1019–1025, Glasgow, UK, Oct. 2010. ASMEDC. ISBN 978-0-7918-4402-1. doi: 10.1115/GT2010-23627. URL https://asmedigitalcollection.asme.org/GT/proceedings/GT2010/44021/1019/346925.

[14] I. Gómez, J. M. Perales, J. M. Vega, and Velázquez. Métodos de optimización de tipo gradiente. In *Técnicas de Optimización y Control para el Diseño de Aerorreactores*. 2020.

[15] J. Izquierdo Yerón. *Numerical and experimental study of appendages in racing yachts and their influence on the free surface*. PhD Thesis, Universidad Politécnica de Madrid, 2015. URL http://oa.upm.es/39819/.

[16] E. N. Jacobs and R. M. Pinkerton. Tests in the variable-density wind tunnel of related airfoils having the maximum camber unusually far forward. *Report National Advisory Committee for Aeronautics*, (537):521–529, Jan. 1936. URL https://ntrs.nasa.gov/citations/19930091610.

[17] R. Marler and J. Arora. Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, Apr. 2004. doi: 10.1007/s00158-003-0368-6.

[18] W. Mason. Appendix A: Geometry for Aerodynamicists, Oct. 1997. URL https://archive.aoe.vt.edu/mason/Mason_f/CAtxtAppA.pdf.

[19] D. A. Masters, N. J. Taylor, T. C. S. Rendall, C. B. Allen, and D. J. Poole. Geometric Comparison of Aerofoil Shape Parameterization Methods. *AIAA Journal*, 55(5):1575–1589, May 2017. ISSN 0001-1452. doi: 10.2514/1.J054943. URL https://arc.aiaa.org/doi/10.2514/1.J054943. Publisher: American Institute of Aeronautics and Astronautics.

[20] J. Morgado, R. Vizinho, M. A. R. Silvestre, and J. C. Páscoa. XFOIL vs CFD performance predictions for high lift low Reynolds number airfoils. *Aerospace Science and Technology*, 52:207–214, May 2016. ISSN 1270-9638. doi: 10.1016/j.ast.2016.02.031. URL https://www.sciencedirect.com/science/article/pii/S1270963816300839.

[21] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, volume 113 of *Fluid Mechanics and Its Applications*. Springer International Publishing, Cham, 2016. ISBN 978-3-319-16873-9 978-3-319-16874-6. doi: 10.1007/978-3-319-16874-6. URL https://link.springer.com/10.1007/978-3-319-16874-6.

[22] F. Normani. Physics Of Sailing. URL https://www.real-world-physics-problems.com/physics-of-sailing.html.

[23] pkoperek. Answer to "What are the differences between genetic algorithms and evolution strategies?", Jan. 2019. URL https://stackoverflow.com/a/54107130.

[24] J. A. Samareh. Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization. *AIAA Journal*, May 2012. doi: 10.2514/2.1391. URL https://arc.aiaa.org/doi/10.2514/2.1391.

[25] Sandia National Laboratories. About Dakota, . URL https://dakota.sandia.gov/content/about.

[26] Sandia National Laboratories. Optimization, . URL https://dakota.sandia.gov/sites/default/files/docs/6.17.0-release/user-html/usingdakota/studytypes/optimization.html.

[27] Sandia National Laboratories. Parameter Studies, . URL https://dakota.sandia.gov/sites/default/files/docs/6.17.0-release/user-html/usingdakota/studytypes/parameterstudies.html.

[28] Sandia National Laboratories. 6.16.0 Reference Manual | Dakota, 2022. URL https://dakota.sandia.gov/content/6160-reference-manual.

[29] Siemens Digital Industries Software. Product Overview. In *Simcenter STAR-CCM+ User Guide, version 2021.1*. Siemens, 2022.

[30] H. Sobieczky. Parametric Airfoils and Wings. In K. Fujii and G. S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies: Inverse Design and Optimization*, Notes on Numerical Fluid Mechanics (NNFM), pages 71–87. Vieweg+Teubner Verlag, Wiesbaden, 1999. ISBN 978-3-322-89952-1. doi: 10.1007/978-3-322-89952-1_4. URL https://doi.org/10.1007/978-3-322-89952-1_4.

[31] Suprayitno, J. C. Yu, Aminnudin, and R. Wulandari. Airfoil aerodynamics optimization under uncertain operating conditions. *Journal of Physics: Conference Series*, 1446(1): 012014, Jan. 2020. ISSN 1742-6588, 1742-6596. doi: 10.1088/1742-6596/1446/1/012014. URL https://iopscience.iop.org/article/10.1088/1742-6596/1446/1/012014.

[32] L. Swiler, R. Slepoy, and A. Giunta. Evaluation of Sampling Methods in Constructing Response Surface Approximations. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 14th AIAA/ASME/AHS Adaptive Structures Conference 7th*, Newport, Rhode Island, May 2006. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-040-6. doi: 10.2514/6.2006-1827. URL http://arc.aiaa.org/doi/abs/10.2514/6.2006-1827.

[33] A. Zafar. What is Stochastic Gradient Descent- A Super Easy Complete Guide!, Apr. 2020. URL https://www.mltut.com/stochastic-gradient-descent-a-super-easy-complete-guide/.

[34] Y. Zou, X. Zhao, and Q. Chen. Comparison of STAR-CCM+ and ANSYS Fluent for simulating indoor airflows. *Building Simulation*, 11(1):165–174, Feb. 2018. ISSN 1996-3599, 1996-8744. doi: 10.1007/s12273-017-0378-8. URL http://link.springer.com/10.1007/s12273-017-0378-8.

9. Annex

9.1. Scatter matrices and parallel coordinates graphics

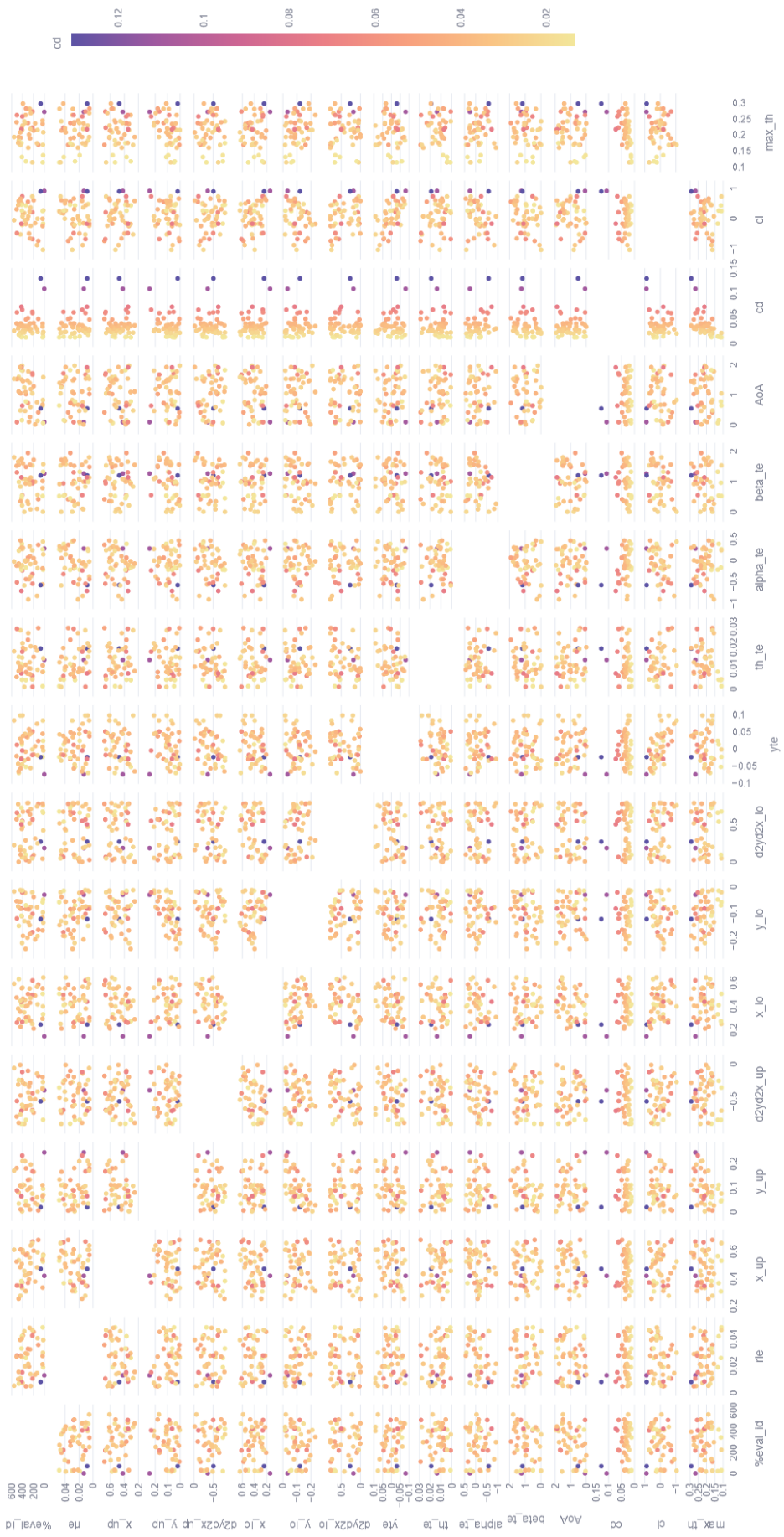

Figure 45: Scatter matrix of an LHS using PARSEC.

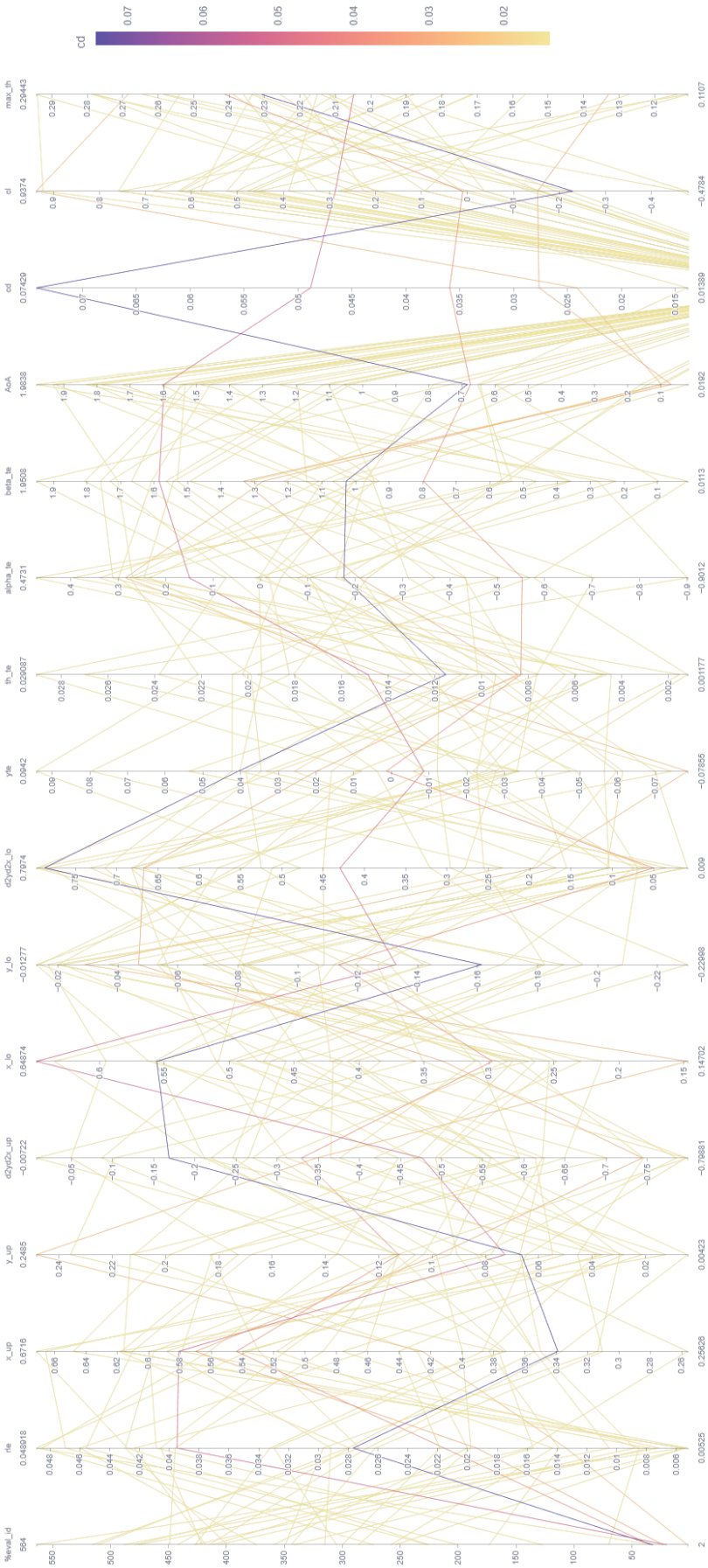

Figure 46: Parallel coordinates of an LHS using PARSEC.

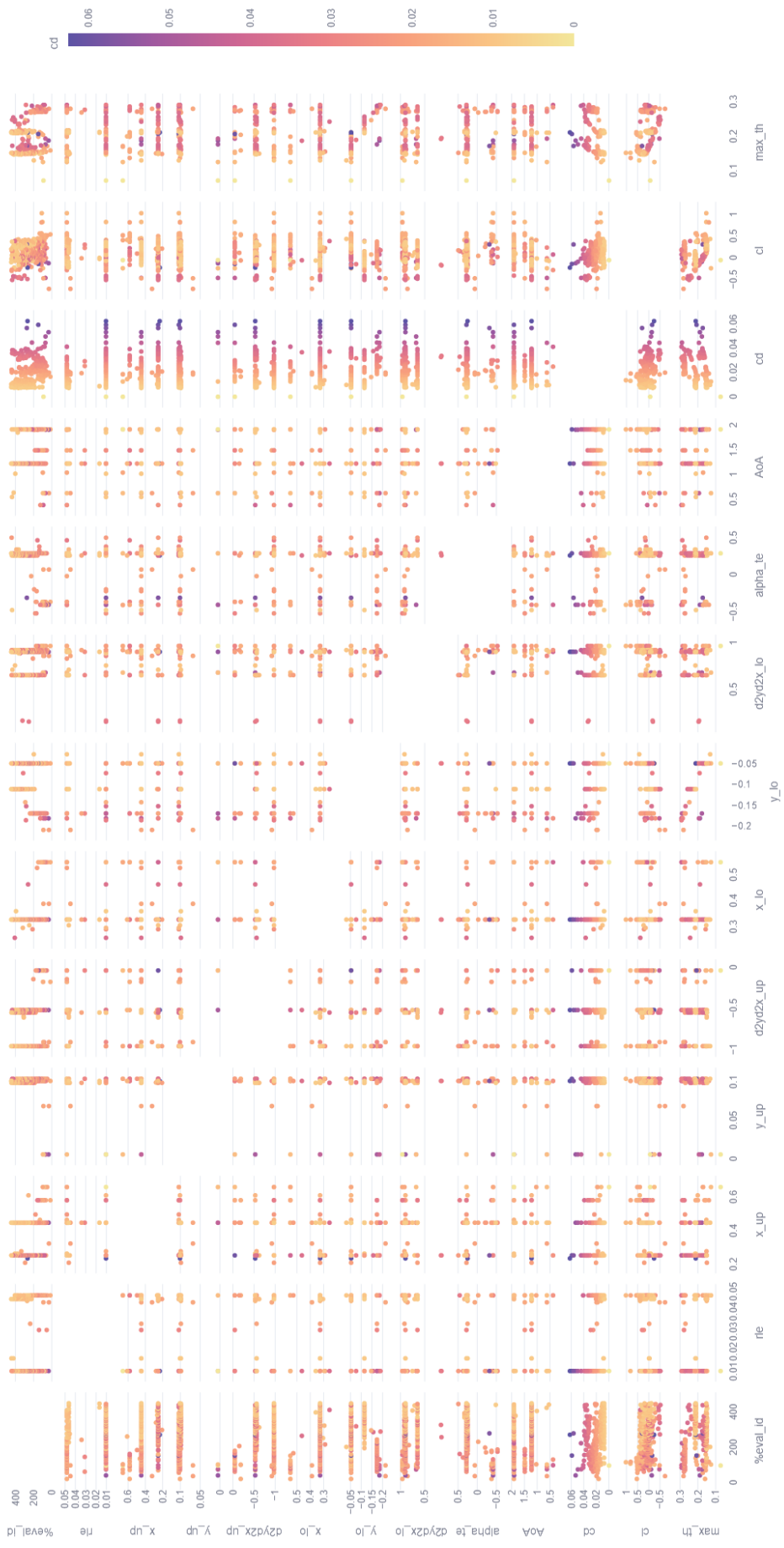

Figure 47: Scatter matrix of PARSEC eight parameters plus AoA.

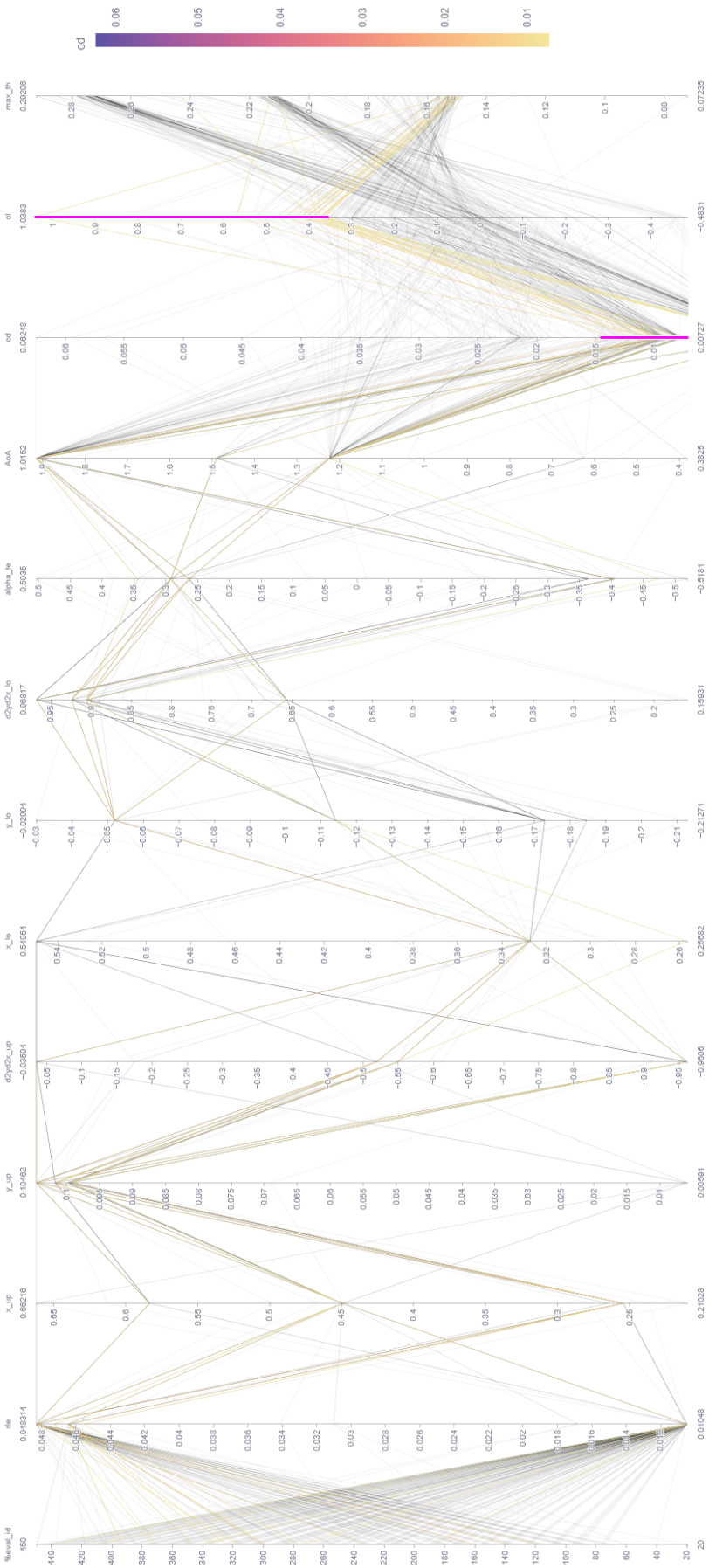

Figure 48: Parallel coordinates of PARSEC eight parameters plus AoA.

Figure 49: Scatter matrix of NACA 4-m optimization.

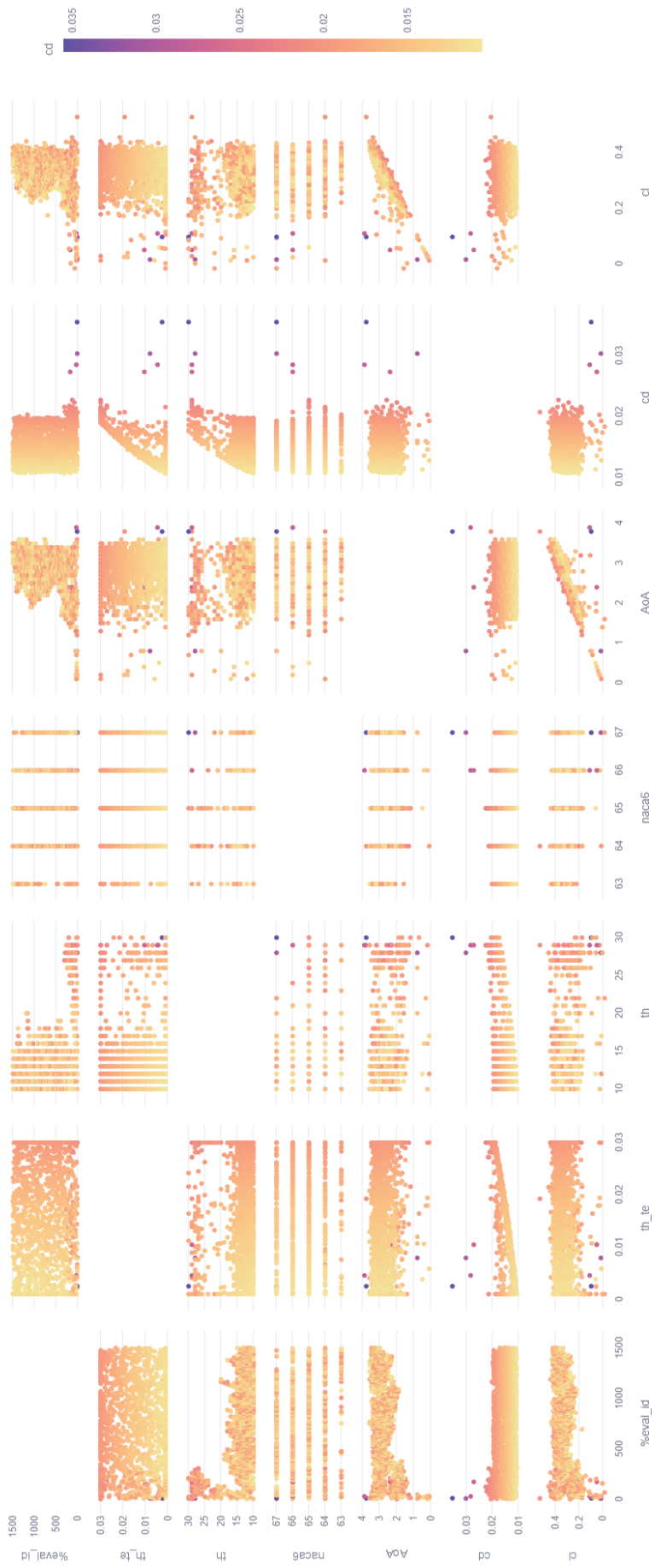

Figure 50: Scatter matrix of NACA 6 optimization.

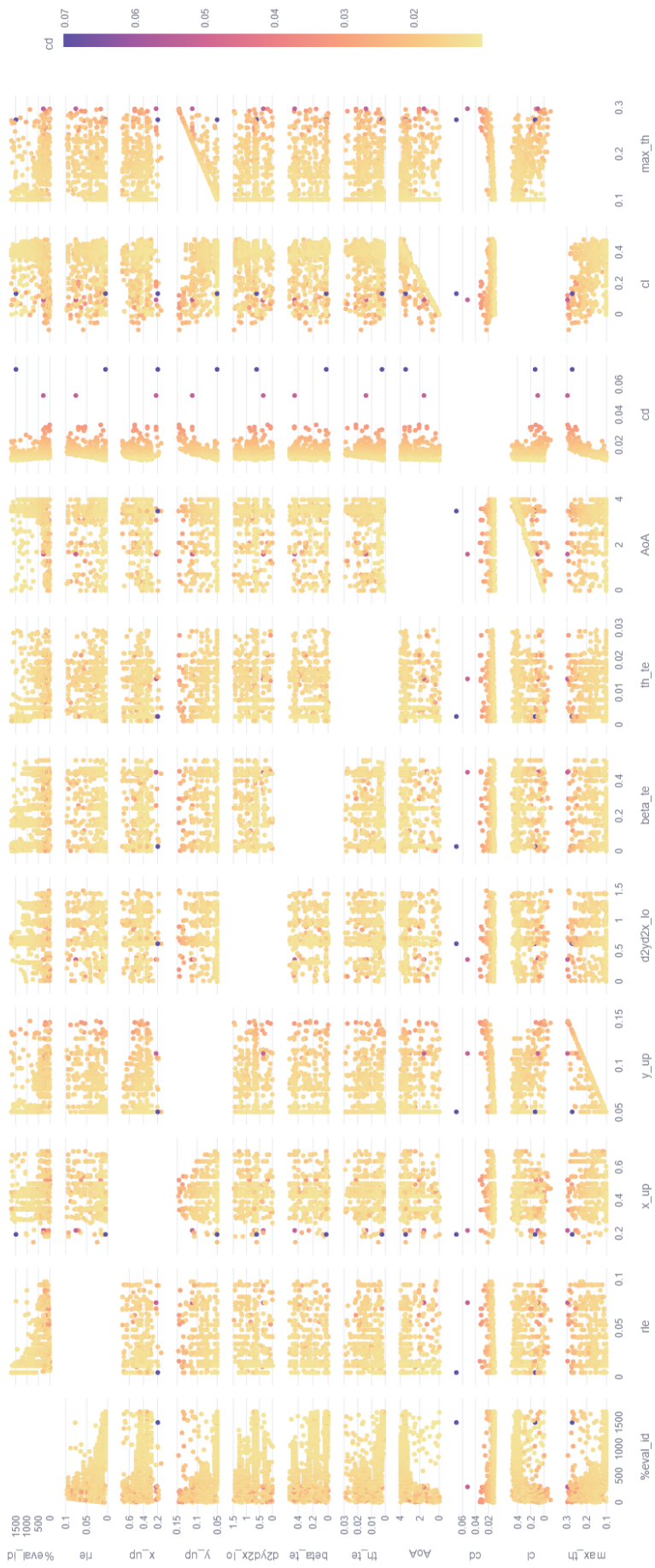

Figure 51: Scatter matrix of PARSEC optimization using SOGA.

9.2. Typical Star-CCM+ simulation results

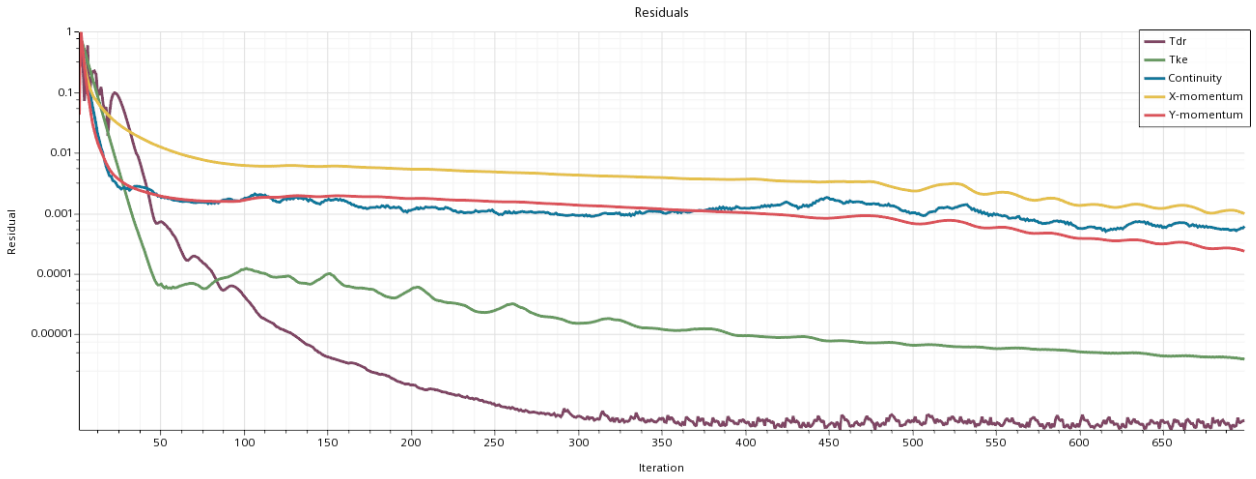

Figure 52: Residuals of a typical Star-CCM+ simulation of this work.

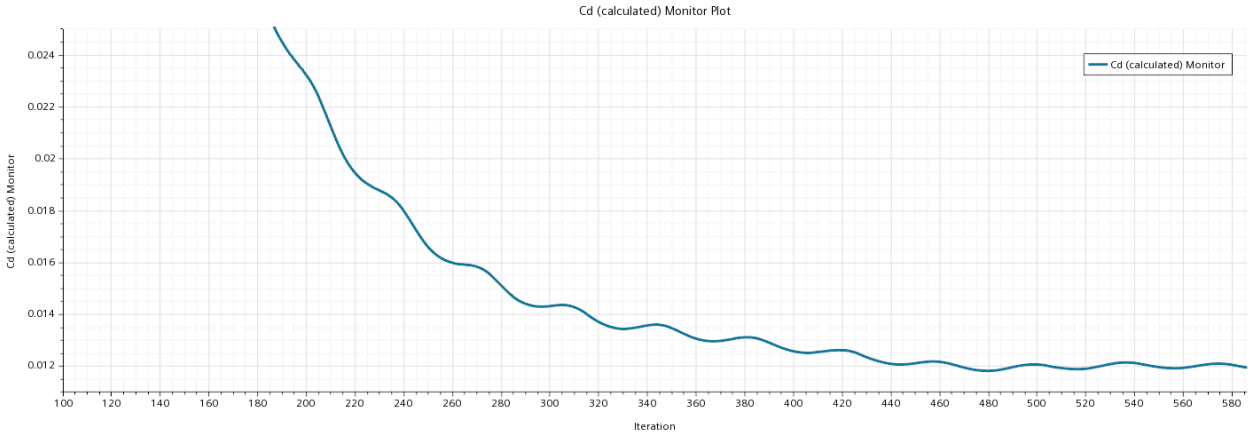

Figure 53: Drag coefficient of a Star-CCM+ simulation of this work.

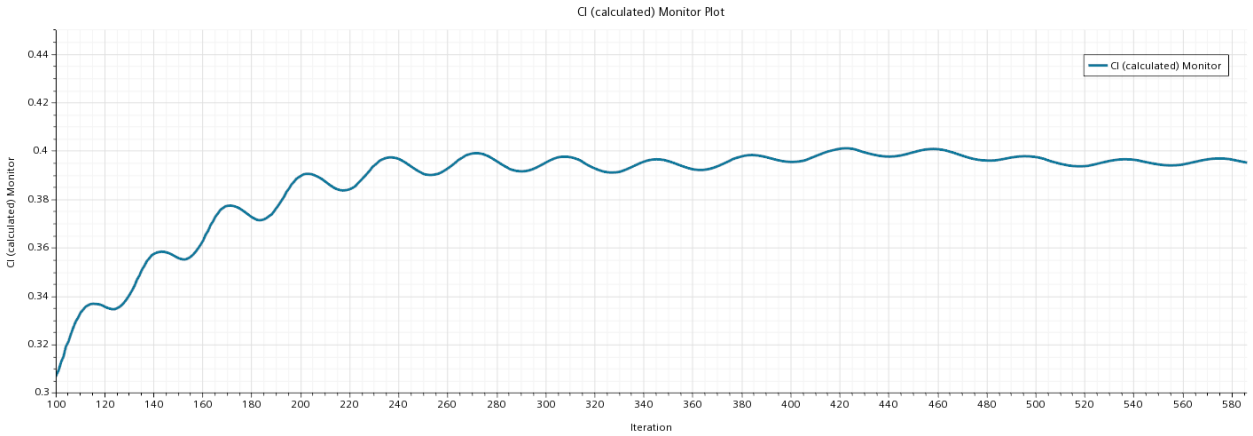

Figure 54: Lift coefficient of a Star-CCM+ simulation of this work.

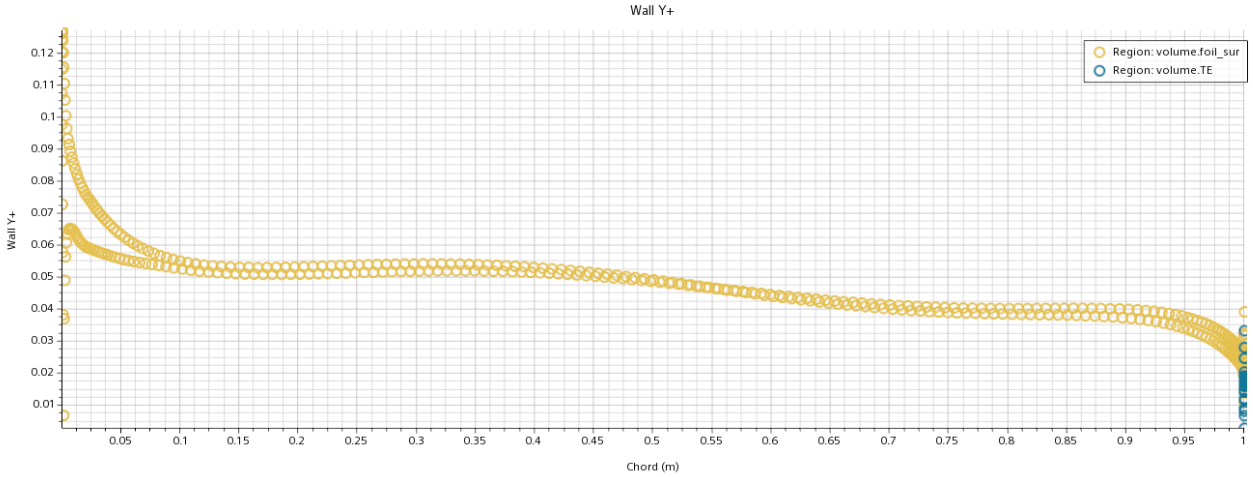

Figure 55: Y+ of a Star-CCM+ simulation of this work.

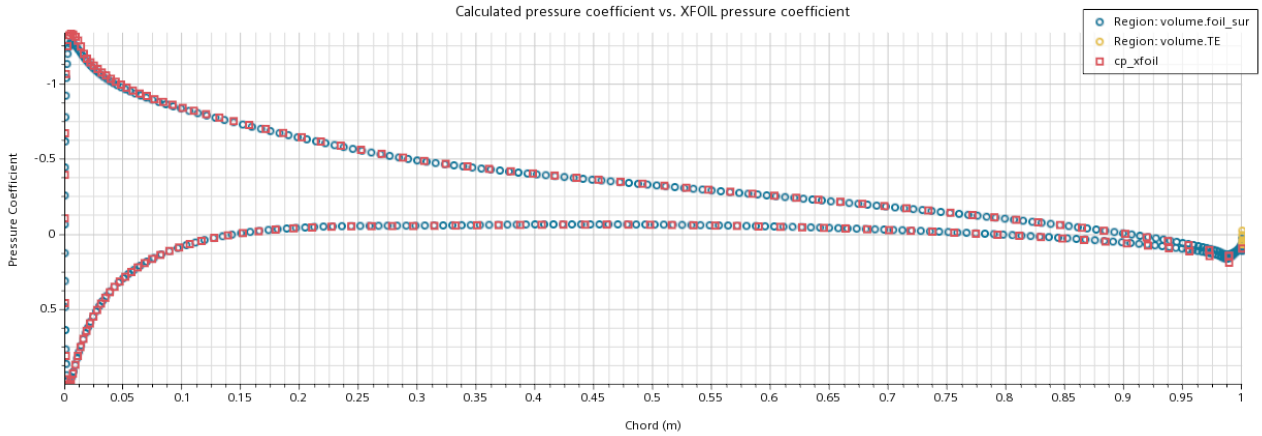

Figure 56: Pressure coefficient distribution comparison between XFoil and Star-CCM+ of a Star-CCM+ simulation of this work.

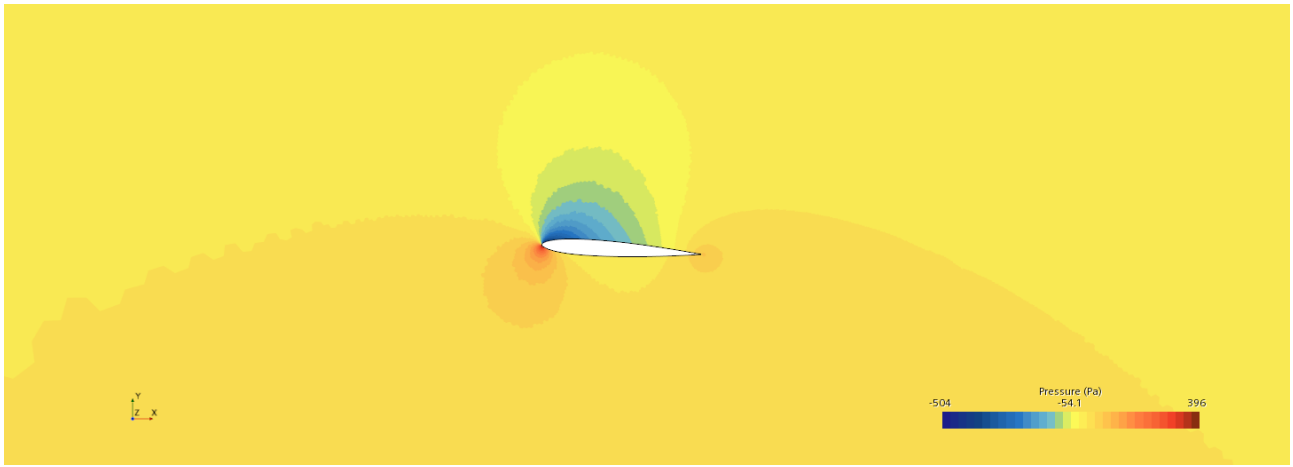

Figure 57: Pressure scalar scene of a Star-CCM+ simulation of this work.

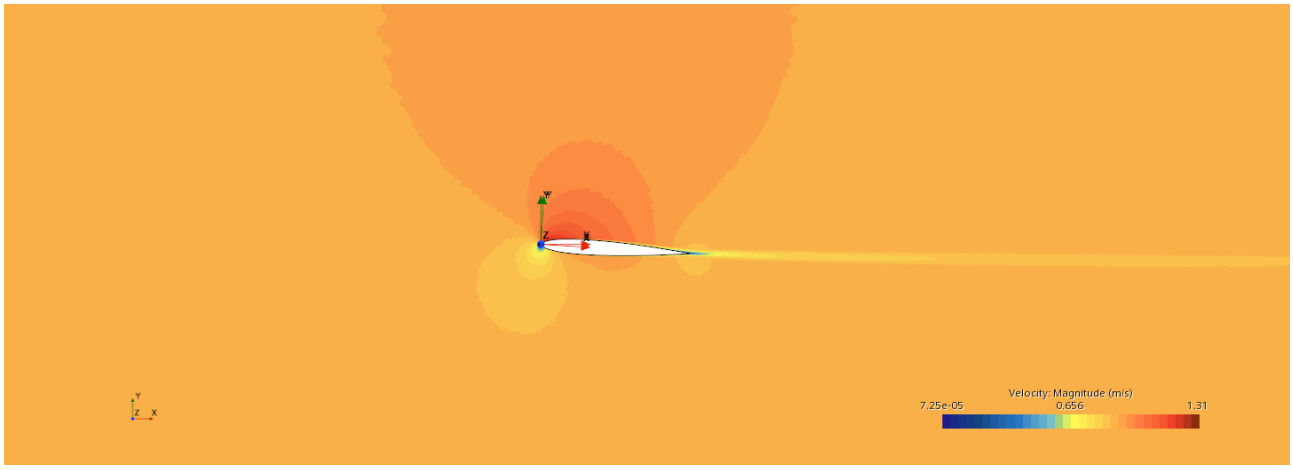

Figure 58: Velocity scalar scene of a Star-CCM+ simulation of this work.