UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

DEGREE IN AEROSPACE ENGINEERING

FINAL PROJECT REPORT

AERODYNAMIC OPTIMIZATION OF SUPERCRITICAL AIRFOILS USING THE ADJOINT METHOD

DAVID VERGARA BARCELÓ

YEAR 2023-2024

TITLE: AERODYNAMIC OPTIMIZATION OF SUPERCRITICAL AIRFOILS USING THE ADJOINT METHOD

AUTHOR: DAVID VERGARA BARCELÓ

SUPERVISOR: RAUL CARLOS LLAMAS SANDIN

DEGREE OR COURSE: AEROSPACE ENGINEERING

DATE: 09/06/2024

ABSTRACT

This project consists in the study and application of the continuous adjoint method for airfoil optimization under transonic flight conditions using Euler (inviscid) and RANS (viscous) approaches. SU2 is the software used as the optimizer and the solver, exploring its capabilities in the field of aerodynamic shape optimization problems. Three Study Cases are performed as practical examples of adjoint-based optimization with Hicks-Henne bump functions as the parametrization method. The first case as an introduction to the optimization process and SU2 capabilities with a simple unconstrained inviscid case of NACA 0012. The second as a constrained inviscid case for NACA 0012 to study the influence of flight speed in airfoil design with solutions for three different Mach Numbers: 0.6, 0.7 and 0.8. The third case corresponds to the ADODG case 2, showing the effective application of the continuous adjoint method in a viscous constrained optimization of RAE 2822 obtaining a reduction of 41.6% of drag coefficient and comparing results with other research studies for the same optimization problem. Therefore, the effective application of the continuous adjoint is demonstrated by producing supercritical airfoils that minimize drag in an effective computational way.

Keywords: Aerodynamics, Optimization, Adjoint method, Airfoil, CFD analysis, SU2.

RESUMEN

Este proyecto consiste en el estudio y aplicación del método adjunto continuo para la optimización de perfiles aerodinámicos en condiciones de vuelo transónico utilizando aproximaciones Euler (no viscoso) y RANS (viscoso). SU2 es el software utilizado como optimizador y solver, explorando sus capacidades en el campo de los problemas de optimización de formas aerodinámicas. Se han realizado tres casos de estudio como ejemplos prácticos de optimización mediante el uso de funciones de Hicks-Henne como método de parametrización. El primer caso sirve como introducción al proceso de optimización y a las capacidades de SU2 mediante un caso de optimización no restringido del perfil NACA 0012. El segundo como un caso no viscoso restringido para el perfil NACA 0012 para estudiar la influencia de la velocidad de vuelo en el diseño del perfil con soluciones para tres números de Mach diferentes: 0.6, 0.7 y 0.8. El tercer caso corresponde al caso 2 propuesto por el ADODG, mostrando la aplicación efectiva del método adjunto continuo en una optimización en flujo viscoso con restricciones del perfil RAE 2822 obteniendo una reducción del 43% del coeficiente de resistencia y comparando los resultados con otros estudios de investigación para el mismo problema. Por tanto, se demuestra la aplicación efectiva del método de adjuntos continuo obteniendo perfiles aerodinámicos supercríticos que minimizan la resistencia aerodinámica de una forma computacional efectiva.

Palabras clave: Aerodinámica, Optimización, Método de adjuntos, Perfil Alar, Análisis CFD, SU2.

ACKNOWLEDGEMENTS

I would like to start by thanking my tutor Raúl for his help and for encouraging me to perform the final project about this topic, which he mentioned in one lecture and kept in my mind since that moment.

Also, I'm thankful for the teachers that I've had during my university studies and to all my degree colleagues, specially to Ander, Carlos, Diego, Patricia and Javi for their help and good memories that I will always remember.

I would like to thank my girlfriend for supporting me every second, for believing in me and for being one of the fundamental pillars in my life.

Finally, I have to thank my family and friends for its unconditional support in the good and the bad moments which have made possible the completion of this project. Specially, I would like to thank my father and my mother for their company in large walks that have cleared my mind and have led to new ideas which have been key.

AGRADECIMIENTOS

Me gustaría empezar agradeciendo a mi tutor Raúl por su ayuda y por animarme a realizar el proyecto final sobre este tema, que mencionó en una vez clase y se quedó grabado en mi mente desde ese momento.

También agradecer a los profesores que he tenido durante mis estudios universitarios y a todos mis compañeros de carrera, en especial a Ander, Carlos, Diego, Patricia y Javi por su ayuda y buenos momentos que siempre recordaré.

Me gustaría dar las gracias a mi novia por apoyarme en todo momento, por creer en mí y por ser uno de los pilares fundamentales en mi vida.

Por último, tengo que agradecer a mi familia y mis amigos su apoyo incondicional en los buenos y en los malos momentos que han hecho posible la realización de este proyecto. En especial, quiero agradecer a mi padre y a mi madre por su compañía en grandes caminatas que me han despejado la mente y me han llevado a nuevas ideas que han sido claves.

Contents

ABSTRACT v

RESUMEN vi

Chapter 1. INTRODUCTION 1

1.1 Motivation 1

1.2 Objectives 1

Chapter 2. BACKGROUND AND THEORY 2

2.1 Environmental Motivation 2

2.2 Breguet equation in Aircraft Design 4

2.3 Wing sections 5

2.4 Aerodynamic Forces 7

2.4.1 Lift 8

2.4.2 Drag 9

2.5 Speed of sound and Mach Number effects 10

2.5.1 Shock waves 11

2.5.2 Critical Mach Number 12

2.5.3 Drag-divergence 13

2.6 Supercritical Airfoils 13

2.7 Design Optimization 15

2.7.1 General Design Process 15

2.7.2 Optimization process 16

2.8 Aerodynamic Optimization 18

2.8.1 Geometry 19

2.8.2 Mesh Generation 19

2.8.3 CFD simulation....20
2.8.4 Sensitivity Analysis....22
2.8.5 Gradient-Based Optimization....23
2.8.6 Methods for Computing Partial Derivatives....24
2.8.7 Methods to Compute Total Derivatives....25
2.8.8 Geometry and Mesh deformation....29
2.8.9 Convergence....31
2.9 SU2....32
2.9.1 Software Architecture....33
2.9.2 Solver Governing Equations....35
2.10 ParaView....36
Chapter 3. STUDY CASES....37
3.1 Study Case 1: Unconstrained Drag minimization of NACA 0012 in transonic inviscid flow. 37
3.1.1 Problem Statement....37
3.1.2 Freestream Conditions....38
3.1.3 Solver....38
3.1.4 Geometry and Mesh definition....38
3.1.5 Airfoil Parametrization....39
3.1.6 Key configuration settings....40
3.1.7 Running the program....41
3.1.8 Numerical Results....42
3.1.9 Post processing results....44
3.2 Study Case 2: Constrained Drag minimization of NACA 0012 in transonic inviscid flow for different Mach Numbers....48
3.2.1 Problem Statement....49
3.2.2 Freestream Conditions....49

3.2.3 Solver 49
3.2.4 Geometry and Mesh definition 50
3.2.5 Airfoil Parametrization 50
3.2.6 Key configuration settings 50
3.2.7 Numerical Results 51
3.2.8 Post Processing Results 57
3.2.9 Results Comparison 64
3.3 Study Case 3: Constrained Drag minimization of RAE 2822 in transonic viscous flow. 66
3.3.1 Problem Statement 66
3.3.2 Freestream Conditions 67
3.3.3 Solver 68
3.3.4 Geometry and Mesh definition 68
3.3.5 Airfoil Parametrization 69
3.3.6 Key configuration settings 69
3.3.7 Numerical Results 70
3.3.8 Comparison with other research studies 73
3.3.9 Post Processing Results 74
Chapter 4. CONCLUSIONS AND FUTURE WORK 77
4.1 Concluding remarks 77
4.2 Future development 80
APPENDICES 81
4.3 Study Case 1 Appendix 81
4.4 Study Case 3 Appendix 82
4.4.1 Numerical Results 82
4.4.2 Post Processing Results 85
BIBLIOGRAPHY 90

List of Figures

Figure 1. Forecasted evolution of air transport passenger traffic [3]. 2

Figure 2. IATA strategy towards net zero emission for 2050 [2]. 3

Figure 3. X-48B Blended Wing Body NASA aircraft concept [5]. 3

Figure 4. Boeing Truss Braced Wing aircraft concept [6]. 4

Figure 5. Wing section (Airfoil) representation [9]. 5

Figure 6. General airfoil geometries for different applications [11]. 6

Figure 7. Airfoil nomenclature [12]. 7

Figure 8. Aerodynamic forces on an airfoil [14]. 8

Figure 9. Downwash and pressure distribution scheme on an airfoil [13]. 8

Figure 10. Shock wave creation in an aircraft with increasing speed [17]. 11

Figure 11. Shockwave creation in conventional airfoils as a function of Mach number: top left corresponds to an airfoil under subsonic flow, top right corresponds to a faster speed where the upper surface produces a shock wave where Mach >1 is achieved, left bottom speed is faster with larger shockwave in the upper surface and weaker one in the bottom surface and left right is much faster speed creating stronger shock waves in upper and lower surfaces [15]. 12

Figure 12. Variation of drag coefficient with freestream Mach number increase, illustrating the critical and drag-divergence Mach numbers [10]. 13

Figure 13. Standard NACA 64-series airfoil compared to supercritical airfoil at cruise lift conditions [10]. 14

Figure 14. Conventional design phases [1]. 15

Figure 15. Comparison between conventional (top) versus design optimization process (bottom) [1]. 16

Figure 16. General Algorithm for Aerodynamic Shape Optimization (ASO). 18

Figure 17. Mesh types: structured (upper), unstructured (middle) and hybrid (lower) [20]. 20

Figure 18. Fluid Dynamics three approaches. 21

Figure 19. CFD post processing and experimental flow visualization in a wind tunnel for an aircraft wing [21]. 22

Figure 20. Gradient vector representation in a 2D case for two design variables [1]. 23

Figure 21. Example of airfoil parametrization using Hicks-Henne "Bump" functions for the RAE2822 airfoil [28]. 30

Figure 22 Folder with necessary files for Study Case 1. 37

Figure 23. Solver and mathematical problem definition in the configuration file. 38

Figure 24. NACA 0012 unstructured O-type mesh in far-field boundary. 39

Figure 25. Hicks-Henne bump locations in NACA 0012. 40

Figure 26. Numerical method definition in configuration file for study case 1. 40

Figure 27. JST definition in the configuration file for study case 1. 41

Figure 28. Adjoint-flow numerical definition in the configuration file for study case 1. 41

Figure 29. Iterations display in console. 42

Figure 30. Drag coefficient as a function of the number of function evaluations during the optimization process for Study Case 1. 43

Figure 31. Lift coefficient as a function of the number of function evaluations during the optimization process for Study Case 1. 43

Figure 32. Lift to drag ratio as a function of the number of function evaluations during the optimization process for Study Case 1. 44

Figure 33. Mach contour over NACA 0012 airfoil (baseline) for Study Case 1. 45

Figure 34. Mach contour over NACA 0012 airfoil (optimized) for Study Case 1. 45

Figure 35. Pressure coefficient comparison between baseline and modified geometry for Study Case 1. 46

Figure 36. Geometrical airfoil comparison between optimized and baseline airfoils for Study Case 1. 47

Figure 37. Adjoint-Drag density contours. 47

Figure 38. Cl driver definition setting in the configuration file. 50

Figure 39. Set of available geometric objective and constraint functions for SU2 configuration file. 51

Figure 40. Thickness constraint definition in the configuration file. 51

Figure 41. Drag coefficient convergence for Study Case 2 (Ma = 0.6). 52

Figure 42. Lift coefficient convergence for Study Case 2 (Ma = 0.6). 52

Figure 43. Thickness variation as a function of the number of function evaluations for Study Case 2 (Ma = 0.6). 53

Figure 44. Drag coefficient convergence for Study Case 2 (Ma = 0.7). 54

Figure 45. Lift coefficient variation as a function of evaluations for Study Case 2 (Ma = 0.7). .54

Figure 46. Thickness variation as a function of the number of evaluations for Case Study 2 and (Ma = 0.7).....55

Figure 47. Drag coefficient convergence for Study Case 2 (Ma = 0.8). 56

Figure 48. Lift coefficient variation as a function of number of function evaluations for Study Case 2 (Ma = 0.8). 56

Figure 49. Thickness variation as a function of the number of function evaluations for Study Case 2 (Ma = 0.8). 57

Figure 50. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.6). 58

Figure 51. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.6).....58

Figure 52. Pressure coefficient comparison between baseline and modified geometry for Study Case 2 (Ma = 0.6). 59

Figure 53. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Mach = 0.6). 59

Figure 54. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.7). 60

Figure 55. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.7).....60

Figure 56. Pressure coefficient comparison between baseline and modified geometry for Study Case 2 (Ma = 0.7). 61

Figure 57. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.7). 61

Figure 58. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.8). 62

Figure 59. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.8).....63

Figure 60. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.8). 63

Figure 61. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.8). 64

Figure 62 Solver and mathematical problem definition for Study Case 3. 68

Figure 63. Hybrid mesh used for Study Case 3. 68

Figure 64. RAE 2822 Hicks-Henne bump locations. 69

Figure 65. Constrains definition for Case Study 3. 69

Figure 66. Compressible free-stream definition for Study Case 3. 70

Figure 67. Drag coefficient convergence for Study Case 3. 71

Figure 68. Airfoil area variation as a function of the number of function evaluations for Study Case 3. 71

Figure 69. Lift coefficient variation as a function of number of function evaluations for Study Case 3. 72

Figure 70. Moment coefficient variation as a function of number of function evaluations for Study Case 3. 72

Figure 71. RAE 2822 baseline airfoil Mach contour for Study Case 3. 74

Figure 72. Optimized airfoil Mach contour for Study Case 3. 75

Figure 73. Pressure coefficient comparison between baseline and modified geometry for Study Case 3. 75

Figure 74. Geometrical comparison between baseline and optimized airfoils for Study Case 3. 76

Figure 75. Density contour for NACA 0012 (baseline airfoil) in Case study 1. 81

Figure 76. Pressure contour for NACA 0012 (baseline airfoil) in Case study 1. 81

Figure 77. Temperature contour for NACA 0012 in Case study 1. 82

Figure 78. Constrains definition for Study Case 3 Annex. 82

Figure 79. Drag coefficient convergence for Case study 3. 83

Figure 80. Airfoil area variation as a function of the number of function evaluations for Case Study 3. 83

Figure 81. Lift coefficient variation as a function of number of function evaluations for Case Study 3. 84

Figure 82. Moment coefficient variation as a function of number of function evaluations for Case Study 3. 84

Figure 83. RAE 2822 baseline airfoil Mach contour for Study Case 3 (wrong Cm definition). .85

Figure 84. Optimized airfoil Mach contour for Study Case 3 (wrong Cm definition).....86

Figure 85. Pressure coefficient comparison between baseline and modified geometry for Study Case 3 (wrong Cm definition). 87

Figure 86. Geometrical comparison between baseline and optimized airfoils for Study Case 3 (wrong Cm definition).....87

Figure 87. Comparison between two evaluations for ADODG case 2 obtaining non-unique flow solutions [47].....88

Figure 88. Analytical design pressure distributions for 14-percent-thick phase 2 and 3 supercritical airfoils (M = 0.73; Cl= 0.7; Re = 30E6 [49].....88

List of Tables

Table 1. Computational cost between Direct and Adjoint methods [1]. 28

Table 2. Airflow conditions for Study Case 1. 38

Table 3. Airflow conditions for Study Case 2. 49

Table 4. Comparison for drag coefficient optimization and number of functions evaluated for Ma = 0.6, Ma = 0.7 and Ma = 0.8 for Study Case 2.....64

Table 5. Freestream conditions for Study Case 3. 67

Table 6. ADODG benchmark Case 2 comparison with other research studies [27].....73

Table 7. Study Case 3 drag coefficient results comparison between correct Cm constrain (green) and erroneous Cm constrain (red).....85

Chapter 1. INTRODUCTION

1.1 Motivation

The concept of optimization is fundamental in any discipline in order to seek for advancement and evolution, it is even intrinsic of humans as an instinct to improve their lives and most of things that surround them [1]. In the same way, it is essential in engineering in order to build more sophisticated machines, systems, and components, with multiple objectives such as reducing weight, materials use, environmental impact, cost or also improve its performance.

In the field of aerospace engineering this practice is also followed, as it represents an industry with very high technological advancements that constantly needs to improve their products for the benefit of human lives, safety, and environmental impact reduction in a very competitive market.

Most specifically, the aerodynamics field has suffered many improvements from each generation of aircrafts to the next one, thanks to the parallel development of complex software that are able to simulate real conditions such as airflow simulations by using computational fluid dynamics (CFD) or automatic optimization tools that allow to increase the complexity of problems with the possibility to be solved autonomously, being driven by a computer algorithm.

On the other hand, as problems turn to be more complex, improvement is not only necessary in designs themselves, but in the way these designs are obtained by seeking to reduce the computational time and cost improving the optimization process itself.

1.2 Objectives

For this motivation, the contribution of this project will be focused on improving the aerodynamic performance of aircrafts by improving the wing airfoil characteristics combining high fidelity CFD simulations and automatic optimization based on the continuous adjoint method that allows to reduce computational time while allowing to push further airfoil shape design improvement and optimization.

The theoretical background and technical insights of the optimization problem will be explained and then applied to three Study Cases covering both, Euler and RANS simulations.

The key objective of the project is to explore SU2 automatic and continuous adjoint design tools to generate supercritical airfoil designs that improve transonic performance by reducing drag.

Chapter 2. BACKGROUND AND THEORY

In order to understand this project in the context of modern aviation, this chapter introduces the environmental aviation objectives and summarizes the fundamental physical, the theory behind and the software information that made this project possible.

2.1 Environmental Motivation

An important part of the future of aviation related to aircraft design will be strongly dominated by IATA initiative towards net-zero carbon emissions. At the 77th IATA Annual General Meeting in Boston, USA, on 4 October 2021, IATA member airlines committed to achieve zero carbon emissions for 2050 on their operations [2]. This commitment is fundamental in aviation to adapt to new times with environmental purpose for the benefit of humanity and a better future for the earth.

This initiative is also derived from the aviation passenger traffic forecast Figure 1, which projects future demand for 2050 exceeding 10 billion, increasing therefore its contribution to carbon and overall emissions.

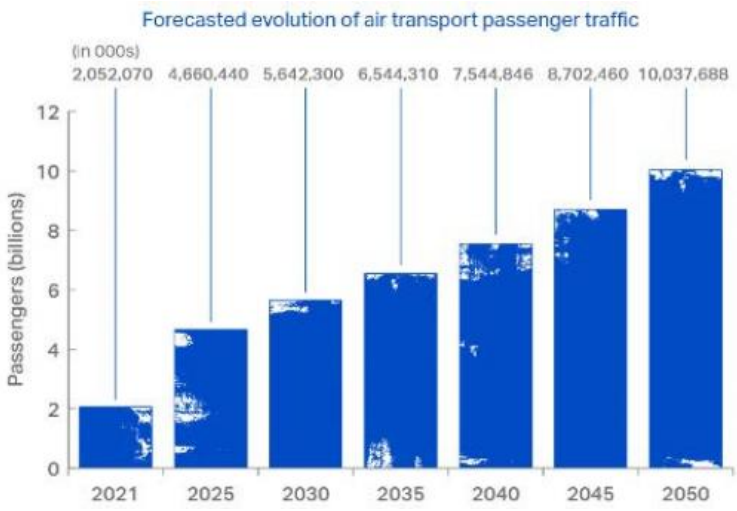

Figure 1. Forecasted evolution of air transport passenger traffic [3].

Also, a strategy has been defined to achieve this goal, combining a set of developments to be made for 2050. This strategy is based on the elimination of emissions by the development and use of SAFs (Sustainable Aviation Fuels), by the introduction of new technologies such as electric and hydrogen, new aircraft concepts derived from this new energies, infrastructure and operational improvements and offsets and carbon capture as seen in Figure 2.

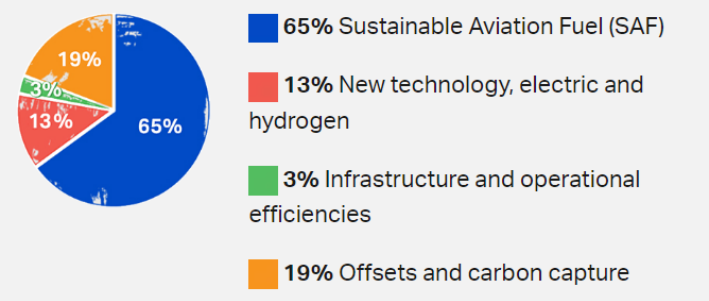

Figure 2. IATA strategy towards net zero emission for 2050 [2].

From these aspects, the matter of this project is to contribute with the development of new technology, electric and hydrogen concepts.

Historically, fuel-efficiency of aircraft has consistently been improved since the introduction of passenger jets in commercial aviation in the 1950s, reducing emissions generation after generation around 15-20% on each, resulting in an efficiency improvement around 80% from 50 years ago [4].

This improvement has come from the use of new and more efficient engines, better aerodynamics and reduced weight of materials and components (with the substitution of aluminium by composites for example).

Future technologies and aircraft concepts will play a fundamental role in emissions reduction, allowing to take advantage of new types of energy as that coming from hybrid, hydrogen, or electric engines, as well as new aircraft design concepts that improve aerodynamic efficiency such as the Blended Wing Body (Figure 3) or a less extreme design change as the Truss Braced Wing (Figure 4).

Figure 3. X-48B Blended Wing Body NASA aircraft concept [5].

Figure 4. Boeing Truss Braced Wing aircraft concept [6].

2.2 Breguet equation in Aircraft Design

The objective of this project is to contribute to short and long-term environmental goals, by improving aerodynamic design in order to either reduce fossil fuel consumption or in the case of new energy sources, reduce drag and therefore consume less energy to fly, facilitating these new designs development.

To reduce the fuel consumed in a flight, there are multiple design variables that can be modified in order to achieve such goal. The Breguet equation for jet aircraft, which is used in performance evaluation for aircraft design also shows a simplified way to define the range that an aircraft is capable to reach at specific values of speed, weight, aerodynamic characteristics, and engine efficiency [7].

$$R = \frac{V}{SFC}\frac{CL}{CD}\ln\frac{W_1}{W_2}$$

V = flight speed

SFC = Specific fuel consumption

$\frac{CL}{CD}$*= Aerodynamic efficiency*

W_1 *and* W_2 *the ratio of weight between two flight points*

This equation helps to understand that in order to achieve a larger range (with same fuel consumed) or the same range (with less fuel consumed) there would be many possible ways to do it from the point of view of aircraft design by increasing numerator terms and decreasing denominator ones. Some of these design improvements are the following:

- Flying faster (Larger V)
- Consuming les fuel for the same thrust produced (lower SFC)
- Improving aerodynamic design (reducing CD and therefore increasing $\frac{CL}{CD}$)
- Decreasing the weight of the aircraft structure and/or increasing the capacity of fuel carried (increasing $\frac{W_1}{W_2}$)

However, these design variables are not trivial to optimize and most of them either have physical or technological limitations for the moment. Some of these limitations are related to fluid properties as critical Mach number, or flight limits as the coffin corner.

2.3 Wing sections

One of the most important aircraft components are wings, which is a specialized aerodynamic surface that generates lift as an aircraft moves through the air. This component is designed to support the aircraft air by means of dynamic reaction on the air [8] while also enabling controlled flight. They normally have optimized streamline shapes with a specific geometry that enables to generate lift forces and maintain its structural strength while optimizing the drag to reduce fuel consumption.

This three-dimensional component characteristics are strongly affected by the shape of the wing section (a two-dimensional component) [8], being called “airfoil” or “aerofoil”, which can be visualized in Figure 5.

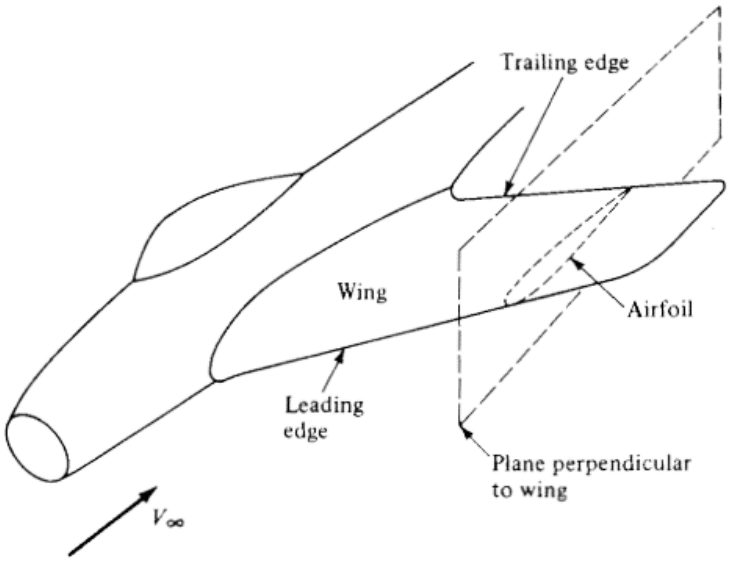

Figure 5. Wing section (Airfoil) representation [9].

Airfoils are some of the most important and basic parameters in an aircraft, as they define the shape of the main wings and the vertical and horizontal stabilizers, having very important effects on the cruise speed, take-off and landing distances, stall speed, handling qualities or the overall aerodynamic efficiency during flight phases [7].

Since the first development of airfoils in 1884 by Horatio F. Phillips (who carried first wind-tunnel experiments on airfoil) [10], this component has been widely studied as some of the fundamentals in aerodynamic design for aircrafts. Due to its relevance, airfoils are cautiously studied resulting in multiple geometries that vary depending on its application as shown in Figure *6*.

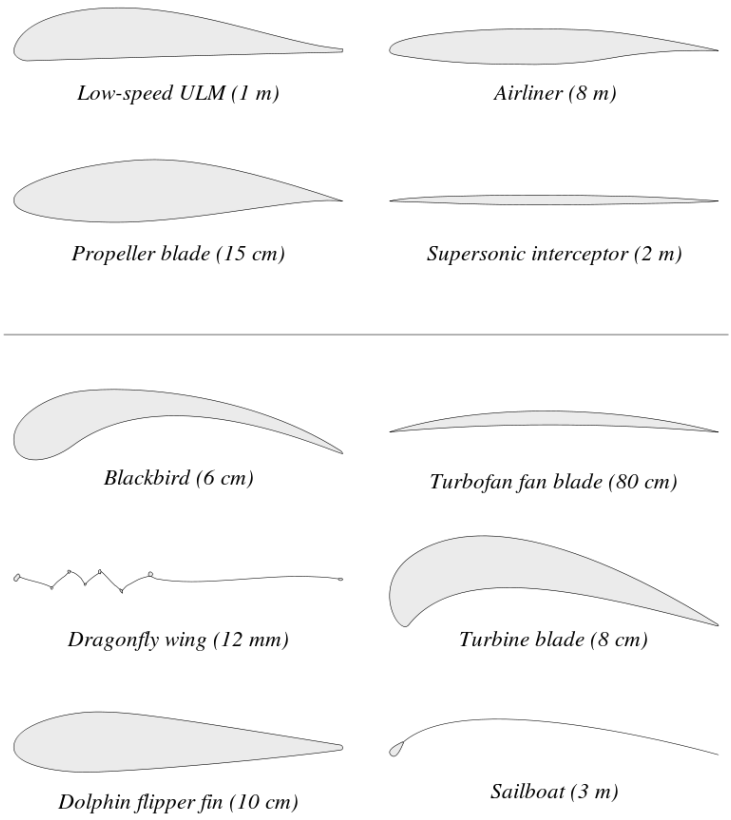

Figure 6. General airfoil geometries for different applications [11].

Clear changes can be seen from the different geometries, that are a result of performance and structural requirements such as:

- Size of the wing
- Stall characteristics
- Flight speed

- Weight to lift
- Maneuverability

Even though these geometries are very different from each other, some important parameters that define them will be introduced for better understanding of future nomenclature.

Figure 7 represents a general airfoil which is submerged in a free stream of air, where the following nomenclature can be defined:

- **Angle of attack (α)**: it is the angle between the chord line and the relative wind vector.
- **Leading edge**: is the point at the front where the airfoil has minimum radius.
- **Trailing edge**: is the point at the rear where there is minimum radius.
- **Chord line**: is the straight line that joins the leading and trailing edge.
- **Camber line**: is the result of joining midway points between upper and lower surfaces.

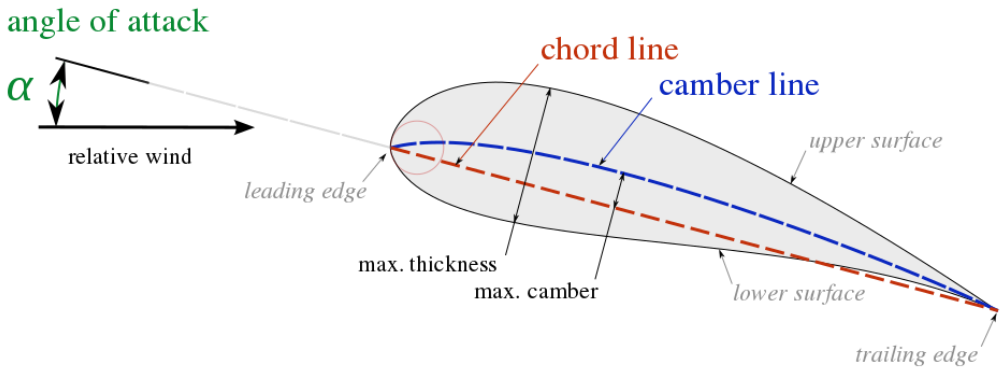

Figure 7. Airfoil nomenclature [12].

2.4 Aerodynamic Forces

When fluids flow through solid bodies, they produce pressure forces that act normal to the surface and also shear forces being those parallel to the body [13]. The composition of these two components produces a resultant force, that is by convention decomposed in two main aerodynamic forces: lift and drag.

These two components are perpendicular to each other and are established as the decomposition of the resultant shear and pressure forces perpendicular to the flow direction (lift force) and the parallel components of pressure and shear (drag force) as seen in Figure 8.

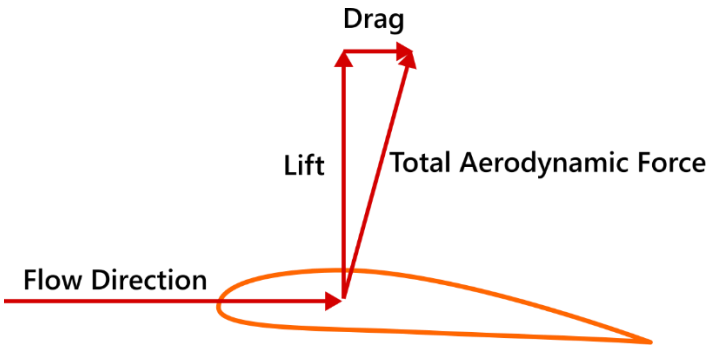

Figure 8. Aerodynamic forces on an airfoil [14].

2.4.1 Lift

Lift is the force that makes possible flying, by opposing to the weight of the object or the aircraft in this case. A full explanation of the lift phenomena is much more complex than it looks, but it can be attributed as the result of the cause and effect of the airflow that is deflected downwards by the airfoil and the pressure difference that is generated along the geometry as seen in Figure 9.

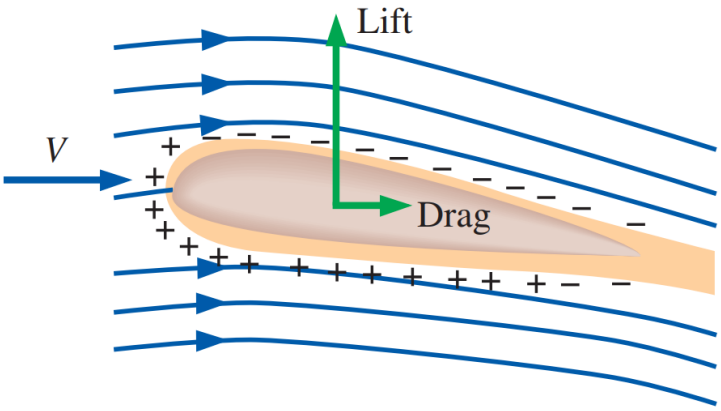

Figure 9. Downwash and pressure distribution scheme on an airfoil [13].

The lift force, as seen in the following equation, depends on the upstream velocity V^2, the density (where $\frac{1}{2} \cdot \rho \cdot V^2$ represents the dynamic pressure of the fluid), the size (represented by the area A, note that this area is referred to a wing by multiplying span times chord so for an airfoil only chord "c" is considered) and a coefficient called "Lift coefficient" [13].

$$L = \frac{1}{2} \cdot \rho \cdot V^2 \cdot A \cdot C_L$$

As the order of magnitude for this force varies depending on multiple factors, it is more convenient to work with dimensionless numbers that allow to compare between different

bodies and be able to attribute the lifting characteristics of the specific body without being related to its dimensions or the flow conditions.

Therefore, the lift coefficient is used, as it will represent how efficient is a body in generating lift under specific conditions (or in a more intuitive way, how good is a geometry in deflecting air downwards). Higher camber and larger downwash angles are correlated with higher CL numbers. The lift coefficient for an airfoil (notice low case letter for lift coefficient is:

$$C_l = \frac{L}{\frac{1}{2} \cdot \rho \cdot V^2 \cdot c}$$

2.4.2 Drag

This Project will focus its main effort on improving the aerodynamic efficiency by reducing drag, so it is important to define what drag is and what types exist.

Drag of an aircraft can be defined as: “the component of force acting in the opposite direction to the line of flight, or in the same direction as the motion of the undisturbed stream” [15], and more generally as “the force corresponding to the rate of decrease in momentum in the direction of the undisturbed external flow around the body” [15]. However, these definitions are very general while drag is a complex force that is a contribution of multiple fluid effects, so it can be decomposed as following.

There are multiple sources of drag, but the most common classification is in two types of drag: parasitic and induced drag.

Parasitic drag refers to the resistance that is independent of the lift generated and is a consequence of the friction of air particles with the aircraft surface (skin friction drag) or the entropy production due to the effect of the boundary layer and wake (profile drag) [16] or associated with the formation of shock waves in high-speed flight (wave drag) [15].

While induced drag (also called drag due to lift) refers to the energy lost in the process of producing lift, by the generation of downwash and the wing tip vortices that produce energy losses from the aircraft to the atmosphere. However, this drag is only considered for three dimensional studies as in an infinite wing there would be no wing tip vortices and therefore no losses due to this effect.

As a matter of this Project and as a simplification of the problem to solve, parasitic drag will be the case of study, most specifically wave drag.

As in the lift force, the drag equation follows the same expression, but this time referring to the drag coefficient [13].

$$D = \frac{1}{2} \cdot \rho \cdot V^2 \cdot A \cdot C_D$$

The drag coefficient for an airfoil (notice "d" low case letter) is:

$$C_d = \frac{D}{\frac{1}{2} \cdot \rho \cdot V^2 \cdot c}$$

Drag coefficient (it will result as a sum of coefficients accounting for the different types of drag) is in the same way a non-dimensional way to define how a body is slowed down by a fluid or how much resistance acts on the body. Higher drag coefficients are correlated with large resistance as in bluff or blunt bodies (as a parachute), while lower values with streamlined geometries (as airfoils) and therefore low resistance.

2.5 Speed of sound and Mach Number effects

An important parameter in the study of high-speed flight is the speed of sound, which is the speed at which an infinitesimally pressure wave travels through a medium, and beyond this speed, fluid particles cannot transmit the information to adjacent particles [13]. This speed can be obtained by the following expression:

$$c = \sqrt{k\,R\,T}$$

Where k is the specific heat ratio of the fluid, R is the gas constat (a fixed value for a specified ideal gas) and T the temperature of the fluid (so it can be deduced that this speed will change as altitude changes due to the variations of temperature).

In relation to speed of sound, another important parameter is the Mach number, which is the ratio of velocity of the fluid (or a moving object in a fluid) to that of the speed of sound.

$$Ma = \frac{V}{c}$$

The main flow regimes that are considered in terms of Mach number as stated in [13] are:

Ma < 1: subsonic

Ma ≅ 1: transonic

Ma = 1: sonic

Ma > 1: supersonic

The Mach number is an important parameter to consider in order to design an aircraft due to the effect of shock waves.

2.5.1 Shock waves

As an object moves through a fluid in a subsonic speed, particles that surround it are being influenced through a mechanism of pressure waves (travelling at the speed of sound), so forward particles have information about the object which is behind them and are able to adapt to this presence by moving away from it. However, as the object accelerates to supersonic speeds, these waves cannot be propagated ahead of the object fast enough (see Figure *10*), so they form a sharp discontinuity known as shock wave that produces drastic changes in the flow conditions.

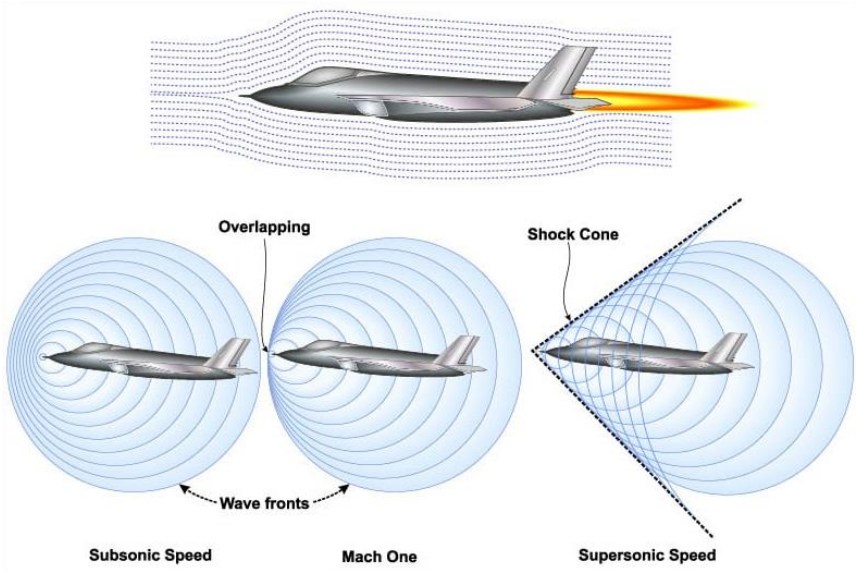

Figure 10. Shock wave creation in an aircraft with increasing speed [17].

Air molecules in front accumulate and are rapidly compressed forming a high-pressure region and also producing a significant increase in temperature and density of the molecules, while after the shock wave, the Mach number significantly decreases (depending in the type of shock wave) and flow properties change discontinuously [10]. Also, there are changes in the flow order, changing from a smoother flow into exhibiting turbulence and separation or other kinds of complex flow phenomena.

2.5.2 Critical Mach Number

Once, Mach number and drag have been defined, it is important to understand its interconnection in the context of commercial aviation and how this effect should be approached by the automated optimizer.

When an aircraft is flying at high speeds (close to Mach 1), as it is the case of modern commercial aviation, as air travels along the aircraft wings surface it is accelerated due to the effect of the curvature as seen from Figure *11*. This means that the local velocity of the airflow at these points will be higher than the freestream speed that the aircraft is actually flying at. As the aircraft accelerates, the speed will reach an important condition as the aircraft will fly at a certain speed below Mach 1, while air on the wing will be actually at Mach = 1 or higher. This speed is called critical speed, and its associated Mach number is the one at which sonic flow is first obtained on the surface [15]. This speed and associated Mach number is called Mach critic (Mcr) and its one of the physical limits that a commercial aircraft has.

As airflow velocity increases beyond Mc, a region where the flow is supersonic will develop from the point where the flow is firstly sonic, ending in a shock wave that is normal to the surface. As speed keeps increasing, this shock wave becomes stronger and moves rearward, reaching a value where also the lower surface passes the sonic regime, and a weak shock wave also appears [15].

As Mach number increases beyond this maximum speed, shockwave intensity increases being developed in both, upper and lower surfaces of the airfoil

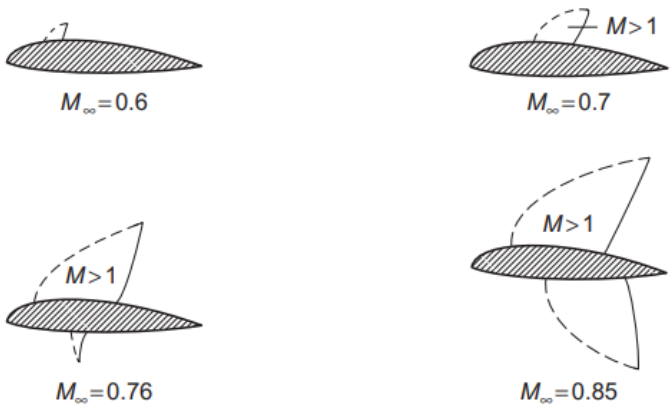

Figure 11. Shockwave creation in conventional airfoils as a function of Mach number: top left corresponds to an airfoil under subsonic flow, top right corresponds to a faster speed where the upper surface produces a shock wave where Mach >1 is achieved, left bottom speed is faster with larger shockwave in the upper surface and weaker one in the bottom surface and left right is much faster speed creating stronger shock waves in upper and lower surfaces [15].

2.5.3 Drag-divergence

In order to understand what wave drag is and why shockwaves are very harmful to the aerodynamic efficiency of commercial aircraft, it is fundamental to understand the drag-divergence Mach number.

As seen from Figure *12*, at subsonic speeds the drag coefficient can be approximated constant up to the point where Mach critic is achieved. After increasing Mach number (beyond this point), a region of supersonic flow will appear on the airfoil as shown in Figure *11*. Then, there is a point where drag coefficient suddenly increases, which can be defined as the drag-divergence Mach number [10]. This high drag increase is associated with a large supersonic region over the airfoil that ends in a shockwave, resulting in flow separation and therefore turbulence downstream. Increasing Mach number further than this point can significantly increase the drag coefficient.

An interesting note is that drag coefficient does not increase up to infinity as it could be thought, but wind tunnel testing demonstrated in the late 1940s that the Cd peak around Mach 1 actually decreased after the freestream entered in supersonic regime [10]. However, the aircraft has to be extremely powerful to overcome this large drag increase to fly further from this point.

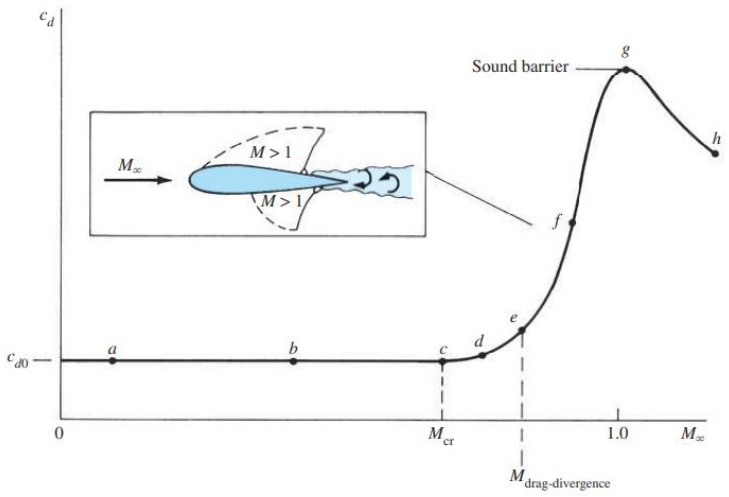

Figure 12. Variation of drag coefficient with freestream Mach number increase, illustrating the critical and drag-divergence Mach numbers [10].

2.6 Supercritical Airfoils

As a conclusion from the Breguet equation from section 2.2 and from the critical Mach number in 2.6, it can be stated that in order to increase the range and reduce emissions, a fast subsonic airfoil that is capable to reduce the critical Mach number and therefore to also reduce the drag-

divergence Mach number so it is possible to fly faster but with the drag benefits of doing it more slowly from the wing perspective.

A solution to this problem can be to use very thin airfoils that produce a small acceleration of the flow on the upper surface, however the use of these types of airfoils has some other important implications that are not considered such as the structural strength, the stall characteristics or even the capacity to storage fuel (which is mainly done in the wings). So, the important study must be dedicated to delay sudden increase in drag while maintaining a certain airfoil thickness.

Therefore, a solution to this problem can be attributed to supercritical airfoils developed by Richard Whitcomb in 1965 at the NASA Langley Research Center, which philosophy approach was towards increasing the separation between Mcr and drag-divergence Mach critic [10].

These kinds of airfoils are characterized by the intended purpose of reducing local peaks of low pressure and therefore high velocity (see Figure *13* for a comparison of the pressure distribution for a supercritical airfoil and a conventional airfoil), which reduce the intensity and extend backwards the shock wave formation (reducing flow separation) by using relatively flat upper surfaces that rise the local Mach number to delay also drag-divergence, while using a larger downwards kick angle near to the trailing edge to maintain lift generation necessary to enable normal flight. So, this airfoil allows aircraft to fly faster for the same drag produced (or reducing the drag for the same speed), significantly reducing the amount of energy lost during flight.

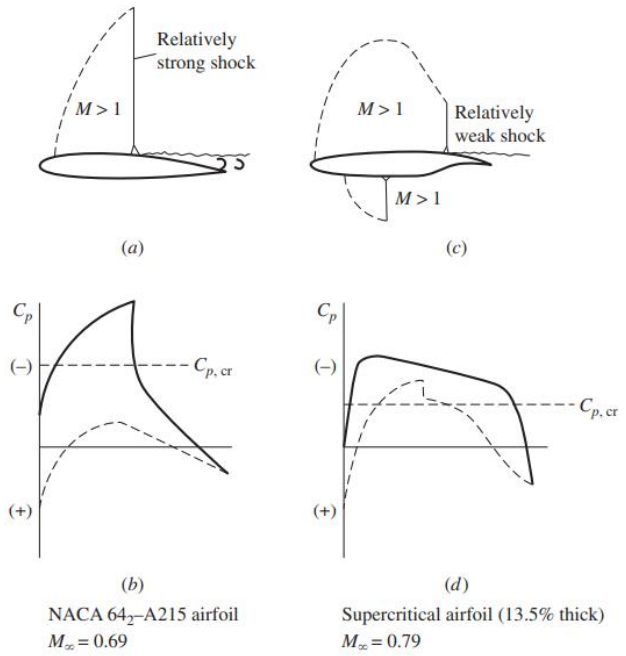

Figure 13. Standard NACA 64-series airfoil compared to supercritical airfoil at cruise lift conditions [10].

2.7 Design Optimization

2.7.1 General Design Process

As previously stated, automatic optimization is fundamental in modern design problems, as the complexity of geometries and problems to approach can't be only solved as a pure human exercise of trial and error.

Engineering design is always based on iterative processes that are followed by engineers in order to produce a component with a specific intended use. The main phases of a conventional design problem can be seen in Figure 14.

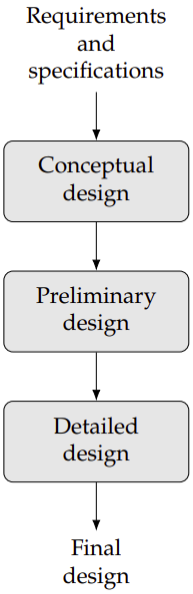

Figure 14. Conventional design phases [1].

First, a market research and analysis of similar designs is performed in order to product a final component that is able to be competitive with the current technology and that is able to satisfy customer requirements. This stage will result in a set of requirements of specification that are defined in accordance with these previous characteristics mentioned and will result in the goals that the final design will have to satisfy at the end of the process to be valid.

Once these goals have been set is the moment to enter in the conceptual phase, where multiple concepts or ideas are considered, and creative approaches are investigated to consider all the possibilities that could result in the optimal design. This phase is normally short and normally relies on human intuition [1].

Then in the preliminary a concept from the previous considered is selected and the design is refined using better models and the design is further developed.

Finally, the detailed design is referred to the phase where the design is developed up to every small detail to be finally produced. However, this process is not linear but iterative as issues can require a step back to previous phases.

2.7.2 Optimization process

Compared to conventional design processes that require the presence of humans in all steps to make a decision to determine if the result is good enough or can be further improved, a design optimization process includes the use of an optimization algorithm that is automated a seen in Figure 15. This design optimization can be applied to any of Figure 14 phases.

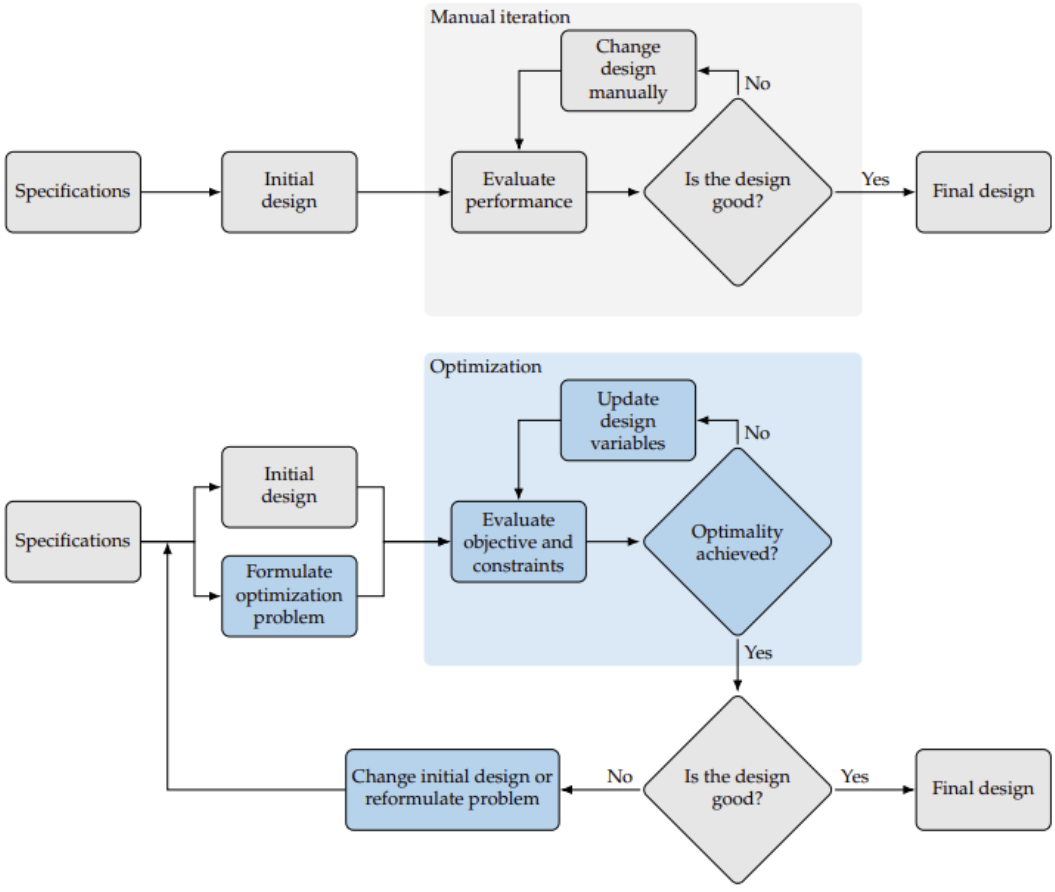

Figure 15. Comparison between conventional (top) versus design optimization process (bottom) [1].

The design optimization process requires a formal formulation of the problem by stating the design variables that should be modified, the objective to optimized (minimize for example) and the constraints to satisfy (limits imposed).

2.7.2.1 Design Variables

This part of the optimization problem will describe the elements that describe the system that should be adjusted to achieve the desired outcome. The choice and definition will depend on the specific problem and the objectives that are expected to achieve.

In order to approach an optimization problem, it is recommended to start with smaller sets of variables and then expanding the set for more extensive results as they will increase the computational time and the complexity of the optimization [1].

2.7.2.2 Objective function

This element is the one that will decide if one design is better or worse than another and it should be the quantity to either maximize or minimize depending on the goal to achieve. Examples of objective functions in aerospace engineering are structural weight, drag, lift, lift to drag ratio, range, fuel consumption or noise.

This function will be computed by a computational model that the engineer defines or uses for the problem, varying its complexity from a simple equation to coupled implicit models [1].

It is one of the most crucial elements to define (it could sometimes not be obvious), as if the engineer selects a function that does not represent the true intent (even if the mathematical model works perfectly) it will never produce an optimal solution from an engineering perspective.

2.7.2.3 Constrains

Another main element in the definition of the optimization problem consists in limiting the possible variation of the design variables in order to reach the objective function as the more freely the optimization can work the more complex and time consuming. However, over constraining can produce results that are not desirable or even create problems with no solution, specifically when these constraints are interdependent and affect to other constraints. Relaxing or removing some of them can be a solution in order to obtain a desired result [1].

2.7.2.4 Problem Statement

Once these elements have been defined, they can be put together in an optimization problem statement as it is conventionally done by the following wording: “minimize the objective function by varying the design variables within their bounds subject to the constraints” [1].

2.8 Aerodynamic Optimization

This concept is a specific case in the context of design optimization, which involves the systematic process of designing and refining objects shapes to improve how this object behaves as submerged in a gas such as air to perform in a desired way that can be either maximized or minimized. Normally, aerodynamic optimization problems are focused in minimizing drag, maximizing lift or achieving other characteristics with the objective of reducing fuel consumption, increasing speed capabilities, improving stability or other specific goals [18].

In order to reach these goals in an autonomous computational process (there is also the possibility to perform aerodynamic optimization manually or by experimental studies), the Aerodynamic Shape Optimization (ASO) process shown in Figure *16* is generally followed:

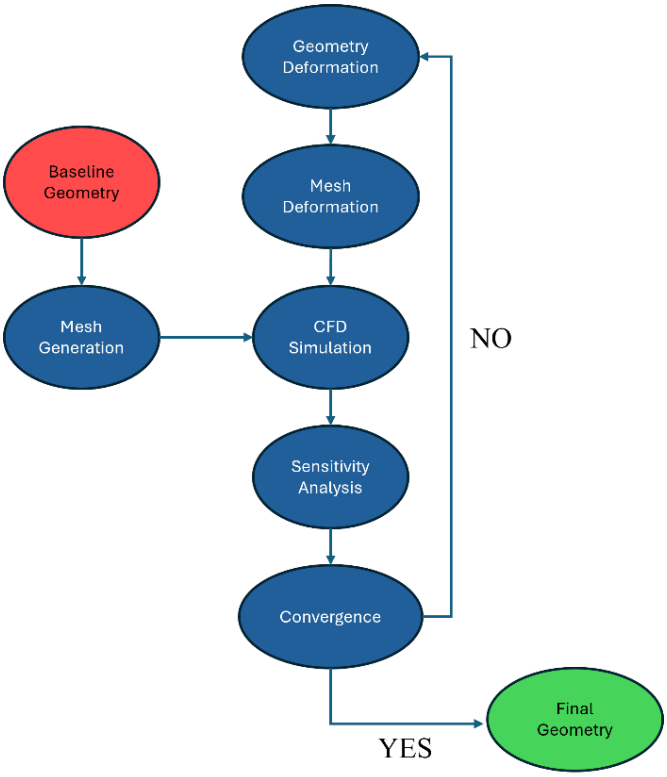

Figure 16. General Algorithm for Aerodynamic Shape Optimization (ASO).

2.8.1 Geometry

First, a geometry has to be created as the baseline point of the study (for this project this would represent the selection of a predefined airfoil that can be extracted from multiple databases). Also, a boundary has to be defined, where the airfoil geometry will be submerged into some fluid as air. The definition of this boundary space is important as some margins have to be accounted on each direction respect to the airfoil in order to reduce possible errors in the computation, as if the geometry is close to the boundary walls it could produce some kind of interference leading to errors.

2.8.2 Mesh Generation

Secondly, this predefined geometry has to be discretized, by the generation of a computational domain where the numerical problem will be solved computing a set of equations using CFD. This domain is called "mesh" and is a grid of cells built from nodes and (points where lines meet) elements (a set of nodes that form an area).

As a general rule, the finer these cells are the better results. However, the increase of the number of nodes an element comes from a computational cost, so a balance has to be found between grid size, accuracy, and computational time [19].

Also, there are different mesh types that should be chosen depending in the type of solver and the geometry to be analysed. Three types as seen in Figure 17 can be distinguished.

- **Structured mesh**: it is formed by regular grid arrangements (in 2D normally by quadrilaterals), in a regular manner which simplifies the discretization process, provides with uniform resolution through the domain and has a more efficient memory usage [19]. However, they require a more experienced user, can consume larger times (as days or months in some cases [20]) and it is limited for complex geometries as being unable to wrap around them. An important note is that it requires a prior knowledge of the flow to be oriented.

- **Unstructured mesh**: they are built from hexahedral, tetrahedral, pyramids or prisms without a regular alignment. It can adapt to complex geometries and irregular boundaries more easily. However, it is more expensive in time and memory solvers, is very sensitive to geometric imperfections and lacks precise control of the local refinement which implies greater overall number of cells [20].

- **Hybrid mesh**: it is a combination of both previous types with the aim to use structured meshes in regions where the flow gradients are significant (vortices, wakes or the boundary layer) to better capture these effects, while filling the far field with an unstructured mesh to reduce the number of elements [20].

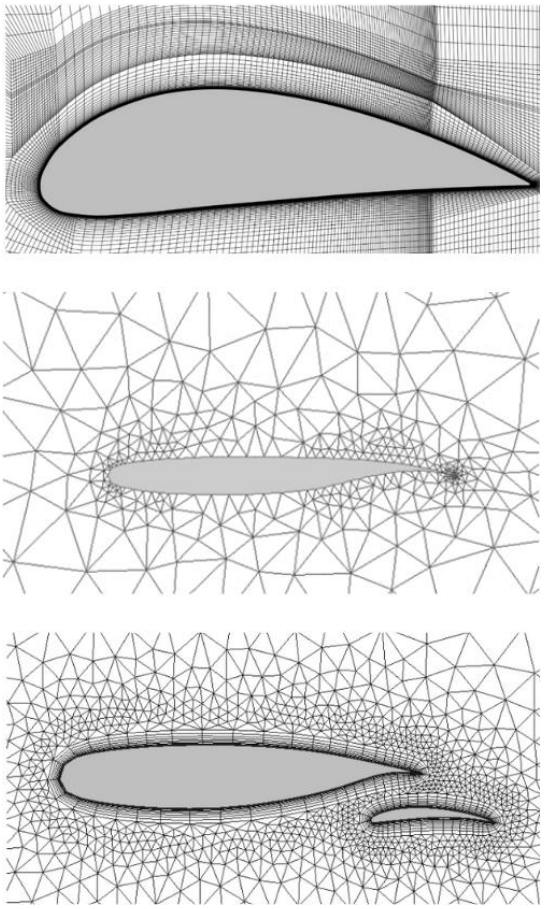

Figure 17. Mesh types: structured (upper), unstructured (middle) and hybrid (lower) [20].

This step is fundamental in order to obtain good results as a good mesh minimizes the numerical diffusion and it allows to find an accurate solution and/or faster convergence [20]. The main idea is to be able to generate mesh that makes the solution independent of this step, so a mesh sensitivity study is normally required.

2.8.3 CFD simulation

Computational Fluid Dynamics (CFD) appeared early in the twenty-first century as a third approach to the field of fluid dynamics [19] (apart from analytical and experimental approaches) for the need to predict the behaviour of fluid flows using computational methods. These three approaches are interconnected (as shown in Figure 18), as they do not work isolated but are

combined in order to provide with a strong and solid understanding of the flow around the object or geometry that wants to be studied.

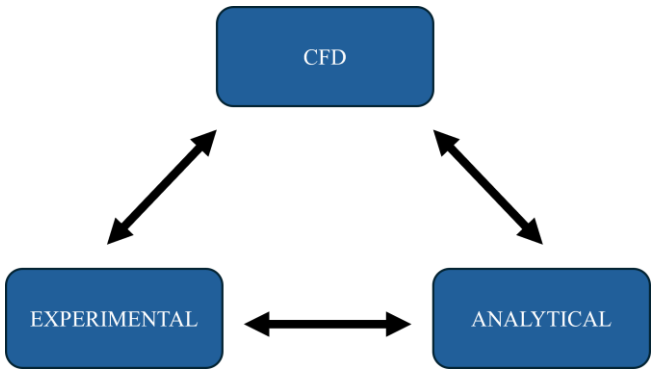

Figure 18. Fluid Dynamics three approaches.

CFD compared to experimental methods is more cost-effective on most cases, as it eliminates the need for physical prototypes and testing facilities by being able to simulate different flow conditions with the only resource of computers. However, for complex simulations CFD requires significant computational resources as high-performance clusters that also suppose elevated costs.

On the other hand, CFD needs to be validated by experimental studies in order to determine if the flow solution obtained has a true physical meaning and the numerical model is related to reality. Some of this validation is relies on the necessity to use simplifications of physical phenomena or assumptions (turbulence modelling, boundary conditions, etc) that can limit the results.

Compared to analytical approaches CFD allows to calculate for complex cases reaching to solutions that cannot be achieved analytically. However, for problems that are not well defined to give an initial "feel" of the problem by the simplification of models and better understand the future study to perform.

Each approach has its limitations and advantages so a combination of them is in most cases the better practice to follow.

2.8.3.1 Governing equations

CFD simulations involve solving the set of fundamental equations that govern fluid motion using numerical algorithms calculated by a computer [19]. For this project, two models will be used: Euler and RANS, that will be covered in section 2.9.2 for the SU2 specific CFD software.

2.8.3.2 Post processing.

After the flow solution has been obtained, some means of visualization are possible thanks to post-processing software. This step is essential in order to interpret the results obtained by visualizing data in a compressible manner. Some post processing techniques involve the creation of contour and vector plots that are able to represent pressure and velocity around the geometry as well as important flow structures such as vortices or shock waves.

Also, post processing can help in the validation process, by comparing between the flow representation that the computer is able to provide and the flow visualization of a wind tunnel to assess the correlation of the CFD approach as shown in Figure *19*.

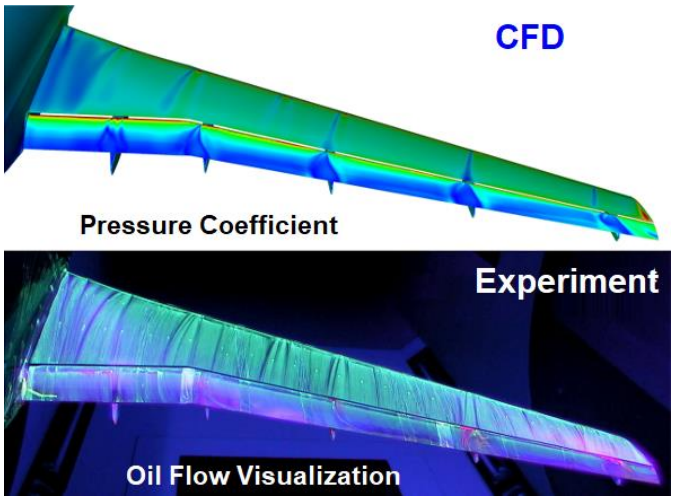

Figure 19. CFD post processing and experimental flow visualization in a wind tunnel for an aircraft wing [21].

2.8.4 Sensitivity Analysis

For a pure CFD simulation, the process would have been finished in the previous step. However, as being in an optimization problem there are still some important steps to define.

Once the computed flow solution has been obtained, the next step in shape optimization is determining the design sensitivities (for this project it will refer on how changes in the airfoil geometry affect the aerodynamic parameters of interest such as drag and lift and the pressure distribution to reduce the shockwave intensity). In a mathematical way, it involves determining the derivatives of one or more functions of interest with respect to the vector of variables [1]. In gradient-based optimization methods, sensitivity analysis refers to this process of computing derivatives.

More specifically, the surface sensitivity can be defined as a measure of the variation of the objective function respect to the possible variations of the geometry surface in the normal direction [18].

The way to compute these derivatives and the accuracy is key in the success of these methods, as the computational cost is directly dependent on the method used and is a fundamental way to improve the full optimization process.

2.8.5 Gradient-Based Optimization

Gradient-based optimization is a method widely used in aerodynamics, referring to the process that is followed in optimization problems by starting from an initial design and using gradient information to reach to an optimum [18].

Once surface sensitivities are obtained, gradient based optimization algorithms are employed to adjust the airfoil shape by iterations. The gradient information from the sensitivity analysis is used to determine how to modify the airfoil geometry and improve its performance.

A simple explanation of the gradient calculation is explained in [1] by the following simple example:

Defining $f(x)$ as an objective function with x being the design variables vector $x = [x_1, x_1, \dots, x_n]$, the gradient of the function would be a column vector of partial derivatives respect to each design variable as:

$$\nabla f(x) = [\frac{\partial f}{x_1}, \frac{\partial f}{x_2}, \dots, \frac{\partial f}{x_n}]$$

So, it is possible to build a map as shown in Figure 20 as the gradient, representing a vector in the direction of highest increase from that point showing the sensitivity (components in the gradient vector determine how fast the function changes with respect to the design variables as the slope of the vector on the different coordinates).

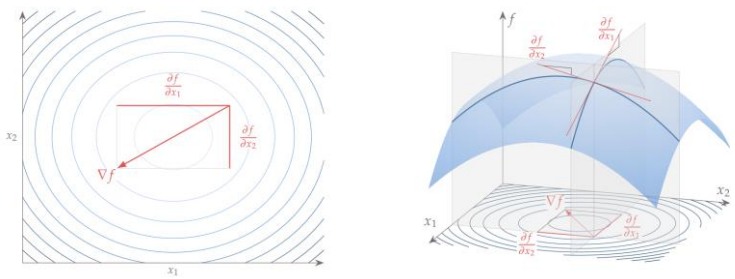

Figure 20. Gradient vector representation in a 2D case for two design variables [1].

On of the keys on these problems is therefore in how these derivatives are computed (there are partial but also total derivatives that need to be calculated) as for complex cases this process is not straight forward, and advanced differentiation methods were developed to improve design optimization processes.

2.8.6 Methods for Computing Partial Derivatives

This section addresses some of the characteristics of the principal methods to compute derivatives in order to introduce the need for Implicit Analytic methods, and more specifically, the Adjoint Method as the key concept of this project.

2.8.6.1 Symbolic Differentiation

Is widely used in calculus, however it is very limited in use as it can only be applied to straight forward cases where there is a small number of operations [1]. This method is computationally intensive for large-scale problems so is not a good choice for aerodynamic optimization problems.

2.8.6.2 Finite Differences

It is a simple method, popular to approach the computation of derivatives. It only requires function values so it is the only viable option for problems where we can only control the inputs and observe the outputs so there is no information about how the function is evaluated. As by default, most gradient-based optimization algorithms use this method to compute the gradients [1]. However, this method is inaccurate and inefficient as the main complication is choosing the step size as explained in “The Step-Size Dilemma” in [1]. Computational cost is dependent on the number of variables so is not an efficient way to approach problems with large numbers of them.

2.8.6.3 Complex Step

It computes derivatives of real functions using complex variables. It requires access to the source code and is more accurate than finite differences, however it is not more efficient as the computational cost also scales with the number of variables [1].

2.8.6.4 Algorithmic Differentiation

AD It is also known as “computational differentiation or automatic differentiation” and is a well-known approach based on the application of the chain rule to computer programs. The advantage of this method is that depending on the type of AD the computing derivative cost can be either dependent on the number of variables or on the number of functions, making it flexible for different optimization problems [1].

2.8.7 Methods to Compute Total Derivatives

The direct and the Adjoint approach they both refer to implicit analytic methods, that linearize the model of equations and obtain a system of equations with solution on the desired derivatives that we want to compute.

As explained in [1], the mathematical background of the Direct and the Adjoint methods will be described in a conceptual way in order to give some insight to the computation theory used for this project.

Given an objective function $f(x,u)$, where x represent the design variables and u state variables.

In order to compute the function of interest, we need to have values for x and u and then solve for f. However, as u is an implicit function of x, f also is. Therefore, to compute f for a given value of x, first a value of u that satisfies the following residual system has to be obtained:

$$r(u;x) = 0$$

This step is the most computationally costly as it requires the use of a solver. Residual equations could be non-linear, and in PDE models it is common to have millions of states. Once u and x are obtained, f is much cheaper to obtain. Also, computing the derivative $\frac{df}{dx}$ by finite-differences requires the solver to find u for each perturbation of x and therefore it means to run this solver n_x times, not scaling well for costly solutions.

Implicit Analytic Methods address the way to solve the nonlinear solution in the derivative computation and therefore are more computationally efficient solutions to obtain derivatives. Two methods can be found that address this problem and that will be further explained: the Direct and the Adjoint method.

2.8.7.1 Direct Method

Given the explicit and implicit dependence of f and x mentioned before, the chain rule can be applied to write the total derivative of the Jacobian of f as:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u}\frac{du}{dx}$$

So, the result to this equation would be an $(n_f \, x \, n_x)$ matrix. The partial derivatives represent the variation of $f(x,u)$ with their respective changes in x or u, without regard to satisfying the

governing equation, while the total derivative $\frac{df}{dx}$ accounts for the change in u to keep the residuals equal to zero.

The importance of using the chain rule relies on the differences on obtaining the total and the partial derivatives:

- **Computing the total derivative $\frac{df}{dx}$**: x is perturbed and the governing equations are re-solved to obtain u. Then, f is computed by already having x and u.
- **Computing partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$**: x or u is perturbed and f is recomputed but without the need to re-solve the governing equations. Numerically these derivatives are much easier to compute or can be obtained symbolically.

In a point where $r(u; x) = 0$, a perturbation of x has to also suppose a perturbation in u while satisfying the governing equations. The differential of residuals will be written as:

$$dr = \frac{\partial r}{\partial x} dx + \frac{\partial r}{\partial u} du = 0$$

Rearranging the previous equation and diving both terms by dx we can obtain the following linear system:

$$\frac{\partial r}{\partial u} \frac{du}{dx} = -\frac{\partial r}{\partial x}$$

Where $\frac{\partial r}{\partial x}$ and $\frac{du}{dx}$ are $(n_u \, x \, n_x)$ matrices and $\frac{\partial r}{\partial u}$ is a square matrix with $(n_u \, x \, n_u)$ size. This system is very useful as if the partial derivatives are obtained, the total derivative $\frac{du}{dx}$ can also be calculated. As the total derivative has n_x dimension, the system has to be solved for each value of x.

Now rearranging and assuming that the matrices can be inverted we obtain:

$$\frac{du}{dx} = -\frac{\partial r}{\partial x} \frac{\partial r}{\partial u}^{-1}$$

Which can be substituted into $\frac{df}{dx}$ expression as:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \frac{\partial f}{\partial u} \frac{\partial r}{\partial u}^{-1} \frac{\partial r}{\partial x}$$

By having performed these previous steps, we have been able to obtain an expression for the total derivative $\frac{df}{dx}$, which is only dependent on partial derivatives that can be computed by the methods previously described (symbolic differentiation, finite differences, complex step, or AD as some examples).

By expressing the total derivative $\frac{du}{dx}$ as $\phi = -\frac{du}{dx}$ it is possible to rewrite the linear system as:

$$\frac{\partial r}{\partial u}\phi = \frac{\partial r}{\partial x}$$

And solving for ϕ obtaining

$$\phi = \frac{\partial r}{\partial u}^{-1} \frac{\partial r}{\partial x}$$

Finally, the total derivative of the function f can be expressed as:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \frac{\partial f}{\partial u}\phi$$

This method has its larger computationally cost in the solution of the ϕ linear system and the cost is associated with the number of inputs n_x but it is almost independent on the number of outputs n_f as the matrix has $(n_u \, x \, n_x)$ dimensions. However, in the direct method the constant of proportionality is much smaller as the nonlinear equation of residuals is only solved once.

2.8.7.2 Adjoint Method

Following the previous development of the direct method equations, the adjoint approach aims to reduce the computational cost by changing the linear system to solve [1].

Instead of solving the linear system for $\frac{\partial r}{\partial x}$ it is solved for $\frac{\partial f}{\partial u}$ in the right-hand side. This step corresponds to replacing the Jacobians shown below by a new matrix of unknowns (called the adjoint vectors and denoted by ψ.

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \boxed{\frac{\partial f}{\partial u}\frac{\partial r}{\partial u}^{-1}} \frac{\partial r}{\partial x}$$

$$\psi^T = \frac{\partial f}{\partial u}\frac{\partial r}{\partial u}^{-1}$$

Multiplying in both sides by $\frac{\partial r}{\partial u}$ and using the transpose of the whole expression, the adjoint equation is obtained:

$$\frac{\partial r}{\partial u}^T \psi = \frac{\partial f}{\partial u}^T$$

The key idea is that now, this linear system is not dependent on x so, each adjoint vector is correlated to a function of interest f_i and can be found by solving the adjoint equation with the corresponding $\frac{\partial f}{\partial u}$. Once the solution for ψ is obtained, the total derivative is computed as follows:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} - \psi^T \frac{\partial r}{\partial u}$$

As in the Direct method, the solution of the adjoint linear system has the highest cost in computation. Even though the linear system has a similar size to the Direct method, in the adjoint approach the computational cost scales with the number of outputs n_f instead of the number of inputs n_x as ψ^T matrix dimensions are $(n_f \, x \, n_u)$ dimensions. The overall cost of computing derivatives for both approaches can be summarized in Table 1. With this table, it is possible to understand the preference in the use of the adjoint method in aerodynamic shape optimization problems, where objective functions are few, but the number of design variables is much larger.

Step	Direct	Adjoint
Partial derivative computation	Same	Same
Linear solution	n_x times	n_f times
Matrix multiplications	Same	Same

Table 1. Computational cost between Direct and Adjoint methods [1].

As Joaquim R. R. A. Martins outlines in [22], the Adjoint Method is the most important development in aerodynamic shape optimization as a contribution of Jameson (as he derived the compressible Euler equations and applied them to airfoil [23] and wing design [24]). This approach, jointly with the use of gradient-based optimization have made it possible to solve optimization problems that involve the use of many variables, which is crucial in aerospace engineering for example as the design of wings requires hundreds of design variables to fully take advantage of design optimization [22].

Therefore, in the development of design optimization, not only efficient gradient-based algorithms are necessary but also how the gradient are computed in an accurate and efficient

manner is fundamental (in a good balance between the implementation effort, accuracy, and efficiency).

Two types of adjoint methods can be identified: the continuous adjoint and the discrete adjoint.

- **Continuous Adjoint**: it linearizes the partial differential equations governing the flow to obtain the adjoint equations and then discretised them to solve for the desired derivatives [22].
- **Discrete Adjoint**: this approach starts with the discretization of the governing equations as a set of residual equations $r(u) = 0$ and then differentiates them to obtain a linear system that is solved to obtain the desired derivatives [22].

A comparison between these two methods was made in [25], concluding that the even though the preferred approach in the last decade was the discrete approach as it is more straightforward to develop and because the gradients are consistent with the discretization, this method had greater computational cost and there was no particular benefit in using the discrete approach over the continuous.

For these reasons, the positioning towards using continuous or discrete approaches is directed into the continuous adjoint for this project in order to reduce the computation cost and due to the availability of resources (just a personal computer with serial computation is used).

2.8.8 Geometry and Mesh deformation

After the previous steps have been completed, and the design sensitivities of the function of interest are obtained, the gradient-based optimization algorithm, guides the search for an optimal solution by defining in which direction and extent has the geometry to be modified in order to improve the design [26]. Once this is clear, the objective of geometry and mesh deformation is to transmit that information obtained to the model to perform the next iteration of the previous steps mentioned. For this goal, multiple parametric method, and modelling techniques such as, B-splines, Ferguson's Splines, Parsec, Hicks-Henne "bump functions" or Free-Form Deformation as some examples.

The key consideration on this step is to be able to perform changes in the geometry in a flexible way in order to explore as many design solutions as possible while maintaining the number of design parameters to a minimum in order to reduce the computational cost and increase the quality of the results obtained as the solution will be dependent on this selected method [26]. Thanks to the adjoint method, the solution is dependent on the ability of parametrization method in adapting

to geometries and computing its effect on the objective function but it's mainly independent on the number of variables.

However, for the material of this project only Hicks-Henne "bump" functions will be explained, and its choice will be discussed.

2.8.8.1 Hicks-Henne "Bump" Functions

This kind of functions were proposed by Hicks and Henne in 1978 as another alternative parametrization technique, being specifically developed in order to generate airfoil profiles [27].

This method consists in the use of a baseline airfoil to which a linear combination of m "bump" functions are added to perturb the initial geometry. The geometry is built as a superposition of analytical shape functions defined by sine functions as seen in Figure 21.

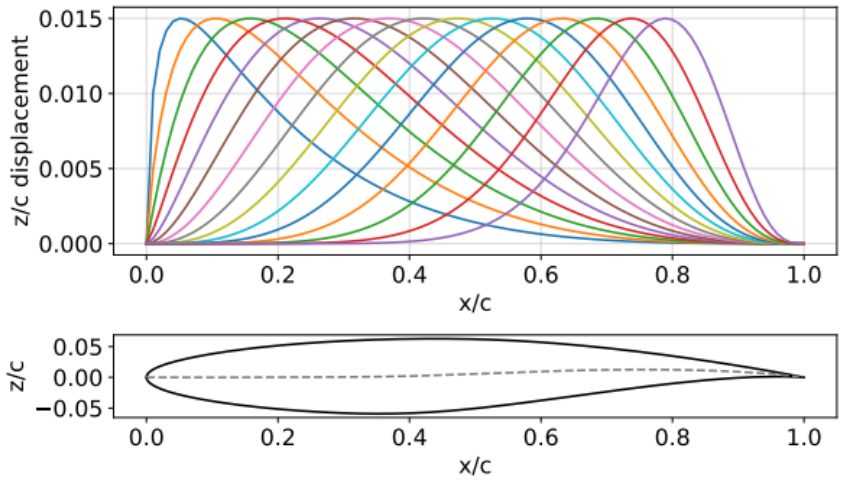

Figure 21. Example of airfoil parametrization using Hicks-Henne "Bump" functions for the RAE2822 airfoil [28].

In this method, an airfoil initial shape represented by $y_{initial}$ is deformed to give the airfoil shape y following the next equation:

$$y = y_{initial} + \sum_{i=1}^{m} \alpha_i f_i(x)$$

Where α_i are the coefficients employed to control the magnitude of the shape functions that act as weighting coefficient and as the design variables that multiply each Hicks-Henne bump functions. The function $f_i(x)$ is defined as follows:

$$f_i(x) = \left(\sin\left(\pi x^{log\frac{0.5}{log\, t_1}}\right)\right)^{t_2} \quad 0 \leq x \leq 1$$

Where t_1 locates the maximum point of bump and t_2 controls the width.

As seen, there are three variables that can be modified. So, to ensure that the optimization process only depends on the coefficients α_i, the rest of variables can be set to remain fixed. Also, this method allows to refine some specific region while keeping the remaining object undisturbed.

This approach has shown to be able to produce good approximations to aerodynamic profiles, being also effective for wing parametrization [28].

On the other hand, the main drawback of this method has been found to be related with the non-orthogonality of the sine functions used, which can result in a problem with no solution or at least not guaranteed for some problems [29].

2.8.9 Convergence

Finally, the means to check that not further iterations should be performed convergence criteria is used to identify this state. This step refers to the finished state of the iterative process when the optimization has reached a point where further iterations wouldn't significantly improve the results and where the objective function reaches a stable value. It is a fundamental aspect of CFD (which has its own convergence) and in the overall optimization process, so defining the adequate convergence criteria is essential.

Residuals represent an important consideration in CFD simulations as they represent the error between the computed and the exact solution, so convergence is achieved when these values reach a sufficient low level.

In order to achieve good convergence, the CFD solver robustness has to be high. The reason is that the CFD solver is more likely to fail as the optimizer does not have the same intuition as an engineer does, opening the possibility to generate bad design shapes at some points of the optimization [22]. In this cases it is important that the solver is able to produce results even when the design does not make much sense.

If the CFD fails to converge at some point during an iteration, it would interrupt the optimization process that should be restarted. There are some optimizers which allow to return a "flag" that

will prevent from exiting outright by taking a less extreme step. However, the optimization might also fail as the information is less rich than in a normal evaluation. As stated in [22], inaccurate information is still better than a "failure", so forcing the CFD to converge in a solution is worthwhile even if that solution does not have a real physical meaning.

2.9 SU2

SU2 is an open-source software as a computational analysis and design tool that has been developed to solve multiphysics analysis and also optimization problems. Its architecture is suited to enable the coupling of multiple physics and optimization packages using powerful tools to address complex problems in engineering. One of its most important characteristics and as one of the reasons of its choice for this project is the extensive development of optimization tools and gradient computation. It is highly focused on the management of large quantities of data in an efficient manner such as the capability to provide with high fidelity capability and sensitivity information for computing design optimization problems using a gradient based framework by allowing to use adjoint methodologies for this goal [30]. It is mainly used for CFD analysis and optimization framework as a free software to use, making it a very good choice for the completion of this project.

For gradient computation, SU2 includes the capability to perform continuous adjoint and discrete adjoint (this last with Algorithmic Differentiation support to compute partial derivatives), allowing it for multiple applications in the optimization framework as it potentially involves large numbers of design parameters.

It is continuous under development of the Aerospace Design lab (ADL) and by Standford University Department of Aeronautics while also collaborating with multiple individual developers and universities around the world that contribute to the advancement of this tool in other areas [30].

Its open-source and open access characteristics positions this software as an example to computational scientists in the world and as a baseline for future development in the computational fluid dynamics and aerodynamics community. These characteristics also allow for some advantages as the rapid and effective bidirectional transfer of information and technology improvements between developers and the community.

2.9.1 Software Architecture

2.9.1.1 C++ Modules

SU2 uses C++ executables as core tools, each of them developed for a specific function. These core individual modules are based on C++ code, with the aim to perform CFD analysis and optimization tasks with PDE constraints [26]. The main modules and their function are the following:

- **SU2_CFD**: it is the Computational Fluid Dynamics Code and is the main and most important module. Its function is to solve direct, adjoint (steady or unsteady) problems for multiple solver such as Euler, Reynolds-Averaged Navier-Stokes (RANS), plasma, free surface, electrostatic, etc [30].
- **SU2_CFD_AD**: is used to perform adjoint flow simulations using Automatic Differentiation (AD).
- **SU2_DEF:** it is the mesh deformation module, that computes the geometrical deformation of surfaces for the mesh and the surrounding nodes producing the volumetric grid. This section is the one that include parametrization techniques including free form deformation (FFD) and Hicks Henne bump functions [30].
- **SU2_DOT:** it is the gradient projection module. It is used to compute the partial derivatives of functions of interest respect to design variables coming from the geometry parametrization. It uses the sensitivities at each mesh node provided by the adjoint solution and projects it into the design space to evaluate derivatives and calculate gradients [30].
- **SU2_DOT_AD:** is has the same function as the previous module but when using Automatic Differentiation to compute derivatives.
- **SU2_GEO:** it is the geometry definition and evaluation module. It evaluates or constrains during the optimization geometric quantities such as volume, thickness, etc [30].
- **SU2_SOL:** it is the solution export module, which generates the output volume and surface solution files in the requested format from the configuration file [30].

Even though these modules can be used individually, the main advantage of SU2 is the possibility to interconnect them in an optimization algorithm to perform much complex activities such as the matter of this project in automatic airfoil shape aerodynamic optimization.

2.9.1.2 Python Framework

Apart from the C++ software used for independent modules, the way to couple them in order to compute complex analysis and design tasks is by the use of Python scripts.

SU2 is built to reduce the complexity in its use and with a high autonomous degree in the coupling of necessary tasks for analysis. There are five levels of components that build the overall architecture, mostly as Python scripts that perform tasks such as modifying the configuration input, executing the C++ modules or post-process resulting data [30]. For further simplification and to shorten the time to set up problems, all these levels start from a single ASCII configuration file. These levels as explained in [30] (from low to high) are the following:

- **Core tools**: it is the lowest level and contains the previously described binary executables. It inputs the information from the configuration file and outputs a history file with data related to forces, moments and other parameters that can be plotted.

- **Pre- and postprocessing**: this second level includes task related to the preparation of the calculation (restarting solution or parallel launch for example) and the management of the solution file output from the SU2_SOL.

- **Sensitivity Analysis**: It manages pre- and postprocessing related to calculating sensitivities with respect to the set of design variables. For adjoint calculations, it requires executions of SU2_CFD and the projection of surface sensitivities by SU2_DOT.

- **Design evaluation**: SU2 wraps with a black box (system that only considers inputs and outputs and not the internal architecture) the previous components to only take a vector of design variables as input. Therefore, the optimizer can drive the design by called the wrapped content. It receives the state vector and performs the set of steps to return performance data as objectives, constrains and sensitivities. It organizes in structured folders restart and plot data useful for secondary analyses or for debugging.

- **Design optimization**: it is the highest level of architecture, and its characterized by the use of gradient-based optimization (methodology explained in Section 2.8.5) using also adjoint approach for the sensitivity analysis (methodology explained in Section 2.8.4). SU2 default optimizer is the SLSQP (Sequential Least SQuares Programming) that is found in another open-source package called SciPy.

The most important python script used for this project is the **"shape_optimization.py"** which manages all the necessary modules to perform shape optimization taking the choices for design variables, solver, objective function, and other additional settings from the configuration file.

2.9.2 Solver Governing Equations

Another important part of SU2 is the physical models described by governing equations that are used by the CFD solver.

For this project, two set of governing equations for compressible flow are used: **Euler and RANS** (this last with Spalart-Allmaras turbulence model).

2.9.2.1 Compressible Euler

Euler approach can be characterised by the assumption of inviscid flow, meaning that there are no effects due to viscosity, which in reality is a fundamental effect in aircraft flight as without viscosity there is no boundary layer.

However, Euler compressible equations are widely used for aerodynamic analysis as they suppose less intensive computationally compared to Navier-Stokes equations while being able to capture shock wave effects, wave drag or pressure distributions as the matter of this project.

For this project, Euler equations will be used to perform simplified cases of aerodynamic optimization using adjoint methodology as it commonly applied to early design stages and analysis due to the balance between computational cost and accuracy of the solution.

The Euler governing equations are a simplification of the Navier-Stokes considering no viscosity and no thermal conductivity, being expressed as following [30]:

$$\mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - S = 0$$

Being U (conservative variables):

$$U = \{\rho, \rho\bar{v}, \rho E\}^T$$

With S as a generic term and the convective flux expressed as:

$$\bar{F}^c = \begin{Bmatrix} \rho\bar{v} \\ \rho\bar{v} \times \bar{v} + \bar{I}p \\ \rho E\bar{v} + p\bar{v} \end{Bmatrix}$$

being $\bar{v} = (u, v, w,)^T$ as the flow speed in three dimensions, ρ is the fluid density, E the total energy per unit mass, p the static pressure and T the temperature.

2.9.2.2 Reynolds-Averaged Navier-Stokes

On the other hand, RANS governing equations represent a much complex approach by the introduction of viscosity effects and turbulence modelling in order to capture boundary layer behaviour.

This approach will be used as a more comprehensive analysis for transonic flow in order to obtain more accurate drag coefficient (summing contributions of wave and viscous drag approximations and to capture detachment of the boundary layer due to shock wave formation).

Additional to those equations used for 2.9.2.1, a viscous flux is added as shown in [30]:

$$\bar{F}^v = \begin{Bmatrix} \dot{\bar{\tau}} \\ \bar{\tau} \cdot \bar{v} + \kappa \nabla \mathrm{T} \end{Bmatrix}$$

being $\bar{v}$ the same to that of 2.9.2.1, T temperature and κ the thermal conductivity and $\bar{\tau}$ the viscous stress tensor.

So, the differential form of the compressible Navier-Stokes equations is the following:

$$\mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) - S = 0$$

For turbulent flow, Spalart-Allmaras is used as a single-equation turbulence model as it is specifically designed for aerospace applications and supposes a good balance between computational efficiency and accuracy to predict turbulent flows [31].

2.10 ParaView

In order to visualize the data obtained from SU2 and to keep the open-source philosophy, ParaView will be used as the post processing visualization software. It is possible to use this tool as SU2 has the option to use ".vtk" as an output format if this is selected in the configuration file.

ParaView is an open-source data analysis and visualization software that allows to visualize large dataset numbers and provide to the user with means to represent this data by physical visualization means [32].

For this project this tool is used to show the flow field around the airfoil geometries for different parameters such as velocity, pressure, or Mach number as the most important.

Chapter 3. STUDY CASES

This section consists in the application of the previous theory to three different airfoil optimization cases for both governing set of equations (Euler and RANS).

3.1 Study Case 1: Unconstrained Drag minimization of NACA 0012 in transonic inviscid flow.

This first case represents an introduction to the two main cases of study of this project by performing the first tutorial provided by SU2 repository from [33].

This study case is a simplified optimization problem that will help to understand SU2 as a design optimization software using adjoint capabilities and allowing to familiarize with the main tools.

In this way, an unconstrained shape optimization problem will be set up for NACA 0012 airfoil under transonic conditions with the objective to minimize drag.

As previously mentioned, the Python scripts are the tools that manage the set of C++ executables, in this case, **"shape_optimization.py"** (that will drive the design cycle involving the flow solver, the adjoint solver and the mesh deformation tools) and the Python modules **"NumPy"** and **"SciPy"** (two python libraries used for mathematical and numerical analysis) are required to perform this study case.

The configuration and mesh files used are extracted from the git repository for this tutorial in [34]. So, the only three files required in the folder are those shown in Figure *22*.

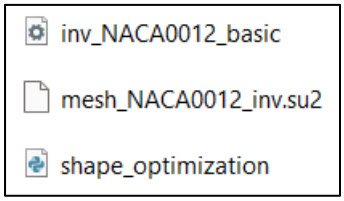

Figure 22 Folder with necessary files for Study Case 1.

3.1.1 Problem Statement

The formulation of this problem is very simple as there are no constrains and only requires the minimization of C_d. So, in a mathematical way this would be:

Minimize Cd

3.1.2 Freestream Conditions

As the intention of the optimization is reducing the drag of the airfoil due to transonic effects, the freestream flow conditions are defined so shock waves appear on the airfoil surface causing drag to rise but due to pressure drag and not to boundary layer effects as there is no viscosity.

Pressure	101325 Pa
Temperature	288.15 K
Mach Number	0.8

Table 2. Airflow conditions for Study Case 1.

3.1.3 Solver

For this case, the effects of viscosity are neglected so the **Euler** and **adjoint Euler** equations are used. In order to use the continuous adjoint this option is selected as shown in Figure *23*.

```
% ------------- DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, FEM_ELASTICITY,
%                               POISSON_EQUATION)
SOLVER= EULER
%
% Mathematical problem (DIRECT, CONTINUOUS_ADJOINT)
MATH_PROBLEM= CONTINUOUS_ADJOINT
%
% Restart solution (NO, YES)
RESTART_SOL= YES
```

Figure 23. Solver and mathematical problem definition in the configuration file.

3.1.4 Geometry and Mesh definition

The geometry that is to be optimized is the NACA 0012 airfoil. One of the reasons of this airfoil is that it has been widely studied in computational and numerical studies but also in experimental wind tunnel research, so it is easier to compare and discuss the results. NACA 0012 will be placed in a far-field boundary with a respective angle of attack of 1.25 degrees with the incoming flow of air.

The discretization of the domain is done by the use of an unstructured O-type mesh with 10,216 elements (see Figure *24*) as it will require less computational power and as an inviscid problem there is no need to capture boundary layer effects (requiring structured mesh refinement near the surface).

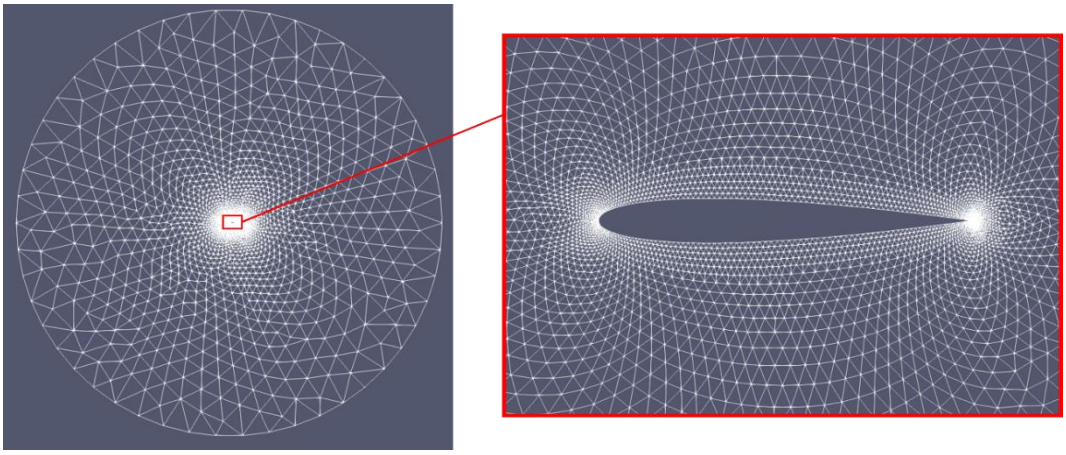

Figure 24. NACA 0012 unstructured O-type mesh in far-field boundary.

3.1.5 Airfoil Parametrization

In order to parameterize the airfoil surface, as stated in 2.8.8.1, Hicks-Henne bump functions will be used for this project. This set of functions can be used from the configuration file by defining multiple number of parenthesis members separated by semicolons with some specific nomenclature.

The first value of the parenthesis refers to the type of variable, in this case **30** for a Hicks-Henne function. The second, refers to the scale of the variable, typically left as **1.0**. Then, the geometry where the variable will be applied is written as follows: **| airfoil |**. Then, either **1 (upper) or 0 (lower)** is used to select between surfaces the airfoil. Finally, a number between 1 or 0 is selected for the x position of the bump, for this case between **1 and 0** as the chord of the airfoil is 1. The set of variables defined for this case are the following:

```
DEFINITION_DV = ( 30, 1.0 | airfoil | 0, 0.05 ); ( 30, 1.0 | airfoil | 0, 0.10 ); ( 30, 1.0 | airfoil | 0, 0.15 ); ( 30, 1.0 | airfoil | 0, 0.20 ); ( 30, 1.0 | airfoil | 0, 0.25 ); ( 30, 1.0 | airfoil | 0, 0.30 ); ( 30, 1.0 | airfoil | 0, 0.35 ); ( 30, 1.0 | airfoil | 0, 0.40 ); ( 30, 1.0 | airfoil | 0, 0.45 ); ( 30, 1.0 | airfoil | 0, 0.50 ); ( 30, 1.0 | airfoil | 0, 0.55 ); ( 30, 1.0 | airfoil | 0, 0.60 ); ( 30, 1.0 | airfoil | 0, 0.65 ); ( 30, 1.0 | airfoil | 0, 0.70 ); ( 30, 1.0 | airfoil | 0, 0.75 ); ( 30, 1.0 | airfoil | 0, 0.80 ); ( 30, 1.0 | airfoil | 0, 0.85 ); ( 30, 1.0 | airfoil | 0, 0.90 ); ( 30, 1.0 | airfoil | 0, 0.95 ); ( 30, 1.0 | airfoil | 1, 0.05 ); ( 30, 1.0 | airfoil | 1, 0.10 ); ( 30, 1.0 | airfoil | 1, 0.15 ); ( 30, 1.0 | airfoil | 1, 0.20 ); ( 30, 1.0 | airfoil | 1, 0.25 ); ( 30, 1.0 | airfoil | 1, 0.30 ); ( 30, 1.0 | airfoil | 1, 0.35 ); ( 30, 1.0 | airfoil | 1, 0.40 ); ( 30, 1.0 | airfoil | 1, 0.45 ); ( 30, 1.0 | airfoil | 1, 0.50 ); ( 30, 1.0 | airfoil | 1, 0.55 ); ( 30, 1.0 | airfoil | 1, 0.60 ); ( 30, 1.0 | airfoil | 1, 0.65 ); ( 30, 1.0 | airfoil | 1, 0.70 ); ( 30, 1.0 | airfoil | 1, 0.75 ); ( 30, 1.0 | airfoil | 1, 0.80 ); ( 30, 1.0 | airfoil | 1, 0.85 ); ( 30, 1.0 | airfoil | 1, 0.90 ); ( 30, 1.0 | airfoil | 1, 0.95 )
```

A physical representation of the points where the bump function is applied is shown in Figure *25*.

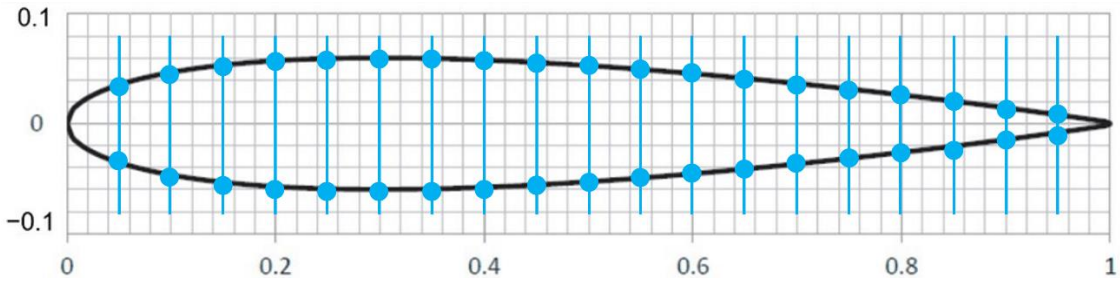

Figure 25. Hicks-Henne bump locations in NACA 0012.

3.1.6 Key configuration settings

An important parameter is the CFL (Courant-Friedrich-Levy), which is a stability criterion used in CFD to ensure numerical stability when solving PDEs governing fluid, which is an important consideration in order to accelerate the convergence searching for a steady state solution to improve the overall efficiency of the procedure [35]. In general terms, lower CFL values indicate smaller time steps relative to the spatial resolution of the flow (it can be said that the flow is being tracked more frequently in time compared to the size of the grid). However, the lowest this value the highest computational cost. This parameter is defined in the numerical method definition as shown in Figure 26.

```
% ------------- COMMON PARAMETERS DEFINING THE NUMERICAL METHOD ---------------%
%
% Numerical method for spatial gradients (GREEN_GAUSS, WEIGHTED_LEAST_SQUARES)
NUM_METHOD_GRAD= GREEN_GAUSS
%
% Courant-Friedrichs-Lewy condition of the finest grid
CFL_NUMBER= 10.0
%
% Adaptive CFL number (NO, YES)
CFL_ADAPT= NO
%
% Parameters of the adaptive CFL number (factor down, factor up, CFL min value,
%                                        CFL max value )
CFL_ADAPT_PARAM= ( 1.5, 0.5, 1.0, 100.0 )
%
% Runge-Kutta alpha coefficients
RK_ALPHA_COEFF= ( 0.66667, 0.66667, 1.000000 )
%
% Number of total iterations
ITER= 1000
```

Figure 26. Numerical method definition in configuration file for study case 1.

Also, the JST (Jameson-Schmidt-Turkel (JST) parameter (shown in Figure *27*) affects to the amount of numerical dissipation added to the solution, so higher JST values result in higher numerical dissipation which helps to stabilize the simulation, but which can reduce accuracy, particularly in regions away from shocks [36].

```
% ---------------------- FLOW NUMERICAL METHOD DEFINITION ----------------------%
% Convective numerical method (JST, LAX-FRIEDRICH, ROE-1ST_ORDER,
%                              ROE-2ND_ORDER)
CONV_NUM_METHOD_FLOW= JST
%
% 2nd and 4th order artificial dissipation coefficients
JST_SENSOR_COEFF= ( 0.5, 0.02 )
%
% Time discretization (RUNGE-KUTTA_EXPLICIT, EULER_IMPLICIT, EULER_EXPLICIT)
TIME_DISCRE_FLOW= EULER_IMPLICIT
```

Figure 27. JST definition in the configuration file for study case 1.

In order to define the adjoint numerical method, parameters shown in Figure 28 are selected in order to control JST and CFL definitions of the adjoint problem.

```
% ---------------- ADJOINT-FLOW NUMERICAL METHOD DEFINITION -------------------%
% Adjoint problem boundary condition (DRAG, LIFT, SIDEFORCE, MOMENT_X,
%                                     MOMENT_Y, MOMENT_Z, EFFICIENCY,
%                                     EQUIVALENT_AREA, NEARFIELD_PRESSURE,
%                                     FORCE_X, FORCE_Y, FORCE_Z, THRUST,
%                                     TORQUE)
OBJECTIVE_FUNCTION= DRAG
%
% Convective numerical method (JST, LAX-FRIEDRICH, ROE-1ST_ORDER,
%                              ROE-2ND_ORDER)
CONV_NUM_METHOD_ADJFLOW= JST
%
% 2nd, and 4th order artificial dissipation coefficients
ADJ_JST_SENSOR_COEFF= ( 0.0, 0.02 )
%
% Time discretization (RUNGE-KUTTA_EXPLICIT, EULER_IMPLICIT)
TIME_DISCRE_ADJFLOW= EULER_IMPLICIT
%
% Reduction factor of the CFL coefficient in the adjoint problem
CFL_REDUCTION_ADJFLOW= 0.8
%
% Limit value for the adjoint variable
LIMIT_ADJFLOW= 1E6
```

Figure 28. Adjoint-flow numerical definition in the configuration file for study case 1.

3.1.7 Running the program

In order to execute the optimization script, the following command line has to be written in the console inside the folder with the three files:

python shape_optimization.py -g CONTINUOUS_ADJOINT -o SLSQP -f inv_NACA0012_basic.cfg

3.1.8 Numerical Results

The solution for this problem is quite simple, so the optimization successfully finished after 9 design iterations, 9 gradient evaluations and with 18 function evaluations as shown in Figure 29. As seen from the objective function values, this value has been reduced after each step reaching a limit where it stabilizes showing the convergence point.

```
  NIT    FC           OBJFUN            GNORM
    1     2     8.183601E-09     3.847000E-06
    2     3     3.054350E-09     2.758859E-06
    3     4     2.031754E-09     9.230682E-07
    4     5     6.112871E-10     6.906011E-07
    5     6     5.968792E-10     1.259340E-07
    6     7     5.718434E-10     2.330871E-07
    7     8     5.583355E-10     1.122869E-07
    8     9     5.579660E-10     9.803238E-08
    9    10     5.657791E-10     1.082116E-07
Optimization terminated successfully    (Exit mode 0)
            Current function value: 5.579660375e-10
            Iterations: 9
            Function evaluations: 18
            Gradient evaluations: 9
```

Figure 29. Iterations display in console.

From the history file that is automatically built by SU2 optimization process, information about multiple parameters can be analysed for each function evaluation.

As seen in Figure *30*, drag is rapidly minimized and then it is stabilized for a long number of evaluations. This large number of evaluations happens because the convergence criteria for the objective function was set by a requested accuracy of $1E^{-10}$.

As being a Euler problem with no constrains, the drag that is expected an obtain is very close to zero (with a reduction of in a reduction of 97.4 %, mainly because of the spatial integration of the equation [33]).

On the other hand, as a consequence of unconstrained optimization, lift is significantly reduced as shown in Figure *31*. For this parameter, more fluctuations are appreciated but it also stabilizes after the 12-function evaluation.

Because there is no restriction on the lift capabilities of the airfoil and the angle of attack is fixed, the optimizer tends to reduce drag by any means, affecting to the Cl value with a 80.48 % reduction from the initial value.

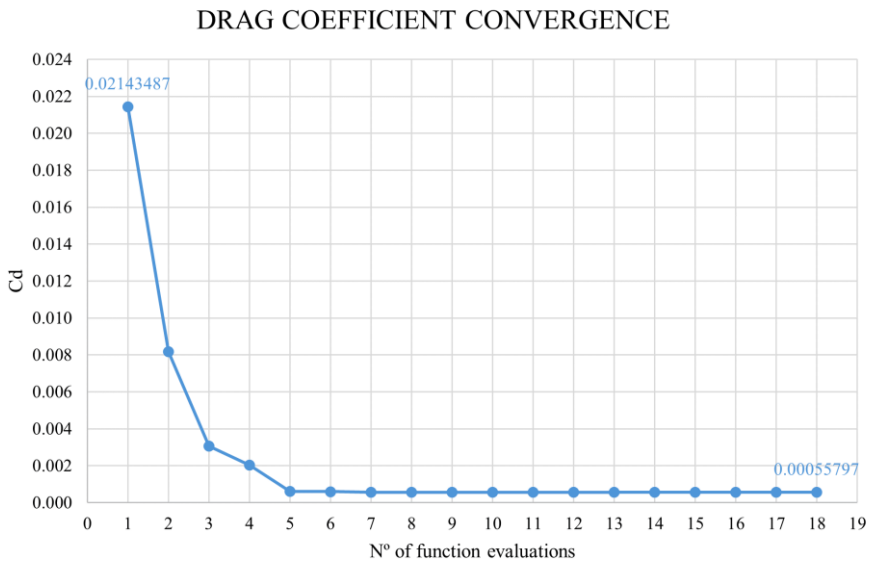

Figure 30. Drag coefficient as a function of the number of function evaluations during the optimization process for Study Case 1.

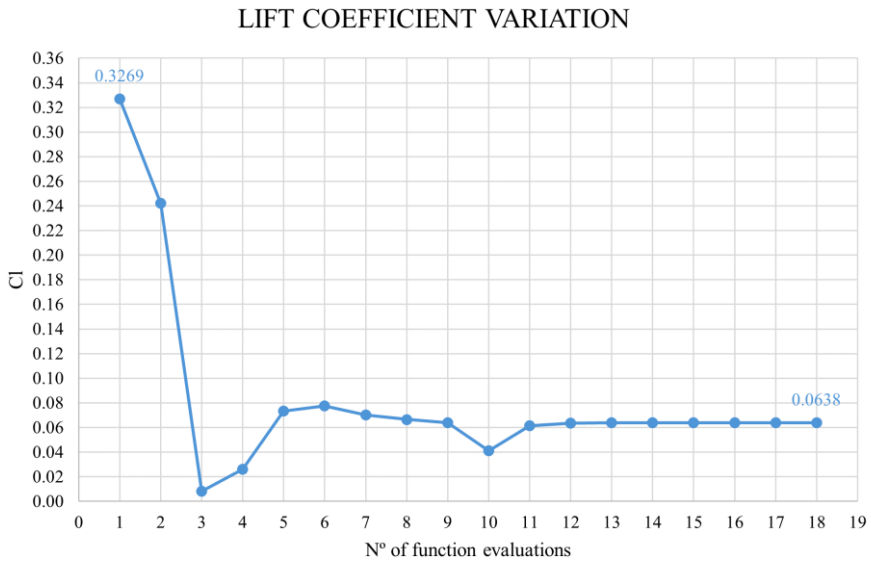

Figure 31. Lift coefficient as a function of the number of function evaluations during the optimization process for Study Case 1.

Finally, a comparison between both coefficients can be made by relating the lift to drag (Cl/Cd) ratio to the number of function evaluations as shown in Figure 32.

As this is not the objective of the optimization, the final lift to drag ratio is not the most optimal from the design evaluations that were made, as in evaluation 6 there is a maximum (129.9) that is

then reduced to the final value of 114.4. However, compared to the initial value, there is a 647.7% improvement in the lift to drag capabilities of this airfoil.

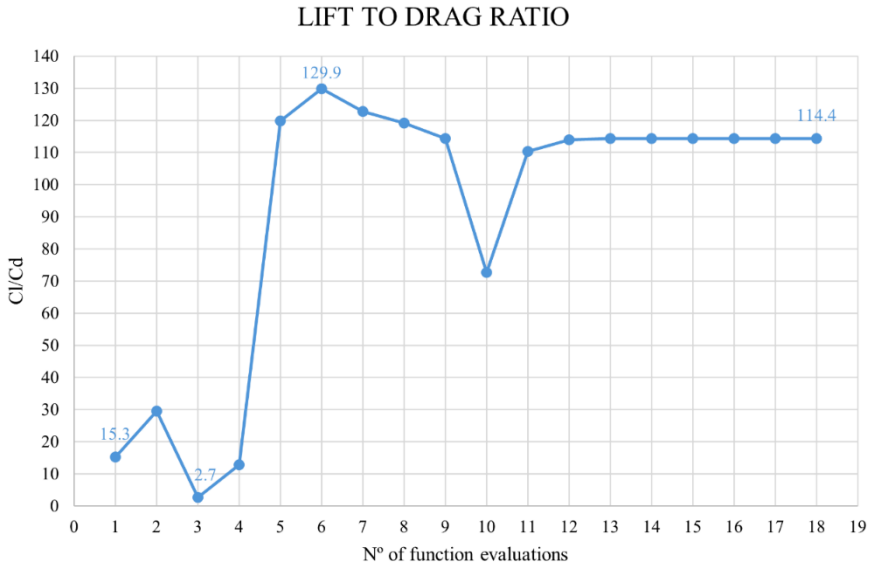

Figure 32. Lift to drag ratio as a function of the number of function evaluations during the optimization process for Study Case 1.

3.1.9 Post processing results

Apart from numerical data obtained from the configuration file, SU2 provides with post processing data from each design evaluation. A total of 18 designs were used in order to reach to the optimized airfoil. However, the most important designs (first and optimized) will be discussed in this section by the use of ParaView post processing visualization tool.

3.1.9.1 Contour plots

As the first step of the optimization process, a solution is obtained for the baseline NACA 0012 geometry. As seen in Figure 33, the flow around the baseline geometry has significant shock wave effects as the flow accelerates over the upper surface from Mach 0.8 up to Mach 1.4 where the hard shock wave is created. On the other hand, in the lower surface there is also a weak shock wave where flow reaches a value around Mach 1.

As explained in section 2.5.1, the expected behaviour of the properties of air before and after the shock wave are fulfilled by this CFD analysis as shown from density (Figure 75), pressure (Figure 76) and temperature (Figure 77) from the APPENDICES section.

In comparison to the baseline airfoil, the optimized geometry clearly shows the work of the optimization algorithm in reducing the shock wave formation on both, upper and lower surfaces, where these effects have been minimized (no shocks appear on the optimized airfoil).

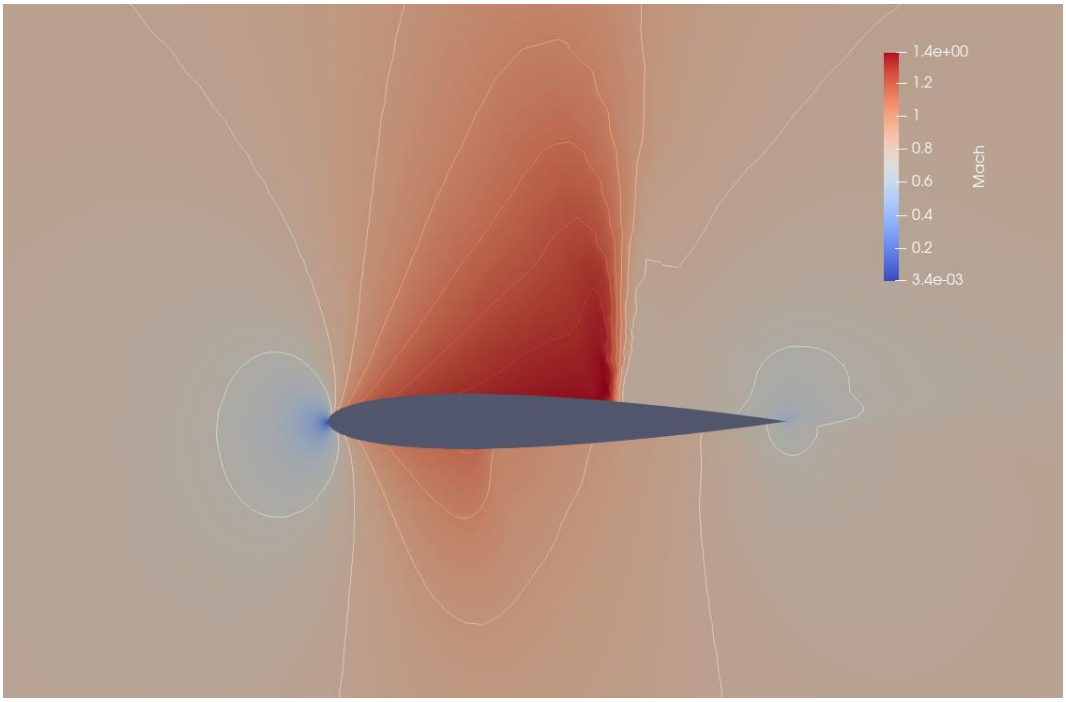

Figure 33. Mach contour over NACA 0012 airfoil (baseline) for Study Case 1.

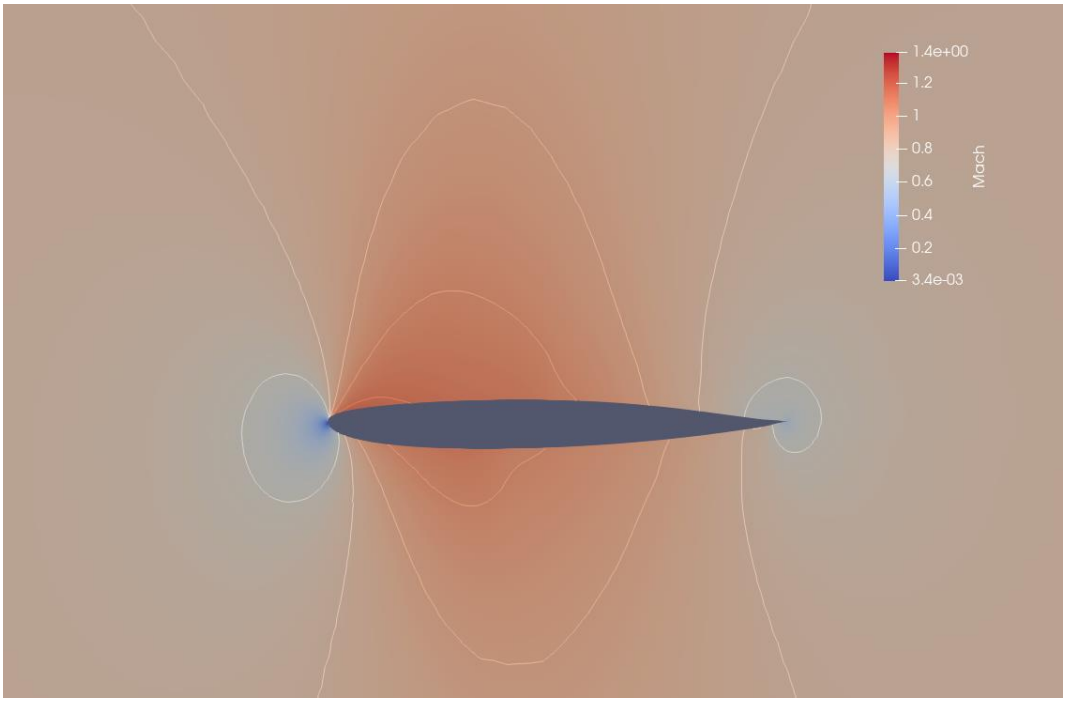

Figure 34. Mach contour over NACA 0012 airfoil (optimized) for Study Case 1.

3.1.9.2 Pressure distribution

As explained in Section 2.4.1 lift can be identified as a cause and effect between downwash and pressure difference between the upper and lower surfaces of the airfoil. Also, as it was introduced in 2.6, the pressure distribution over an airfoil is one of the main conditionals for the creation of shock waves, due to the local peaks of low pressure and high velocity. Figure *35* shows a comparison between the baseline airfoil and the optimized one, where pressure distribution along the surface can be clearly distinguished for both cases. From this graph, the low pressure on the upper surface can be distinguished, however differences are very low due to the small lift generation, especially for the optimized geometry. On the other hand, the theory introduced in 2.6 is demonstrated, as the optimized geometry has reduced the low-pressure peaks that accelerate the flow compared to that high peak seen at 60% of the chord for the baseline airfoil resulting in a more homogeneous pressure distribution.

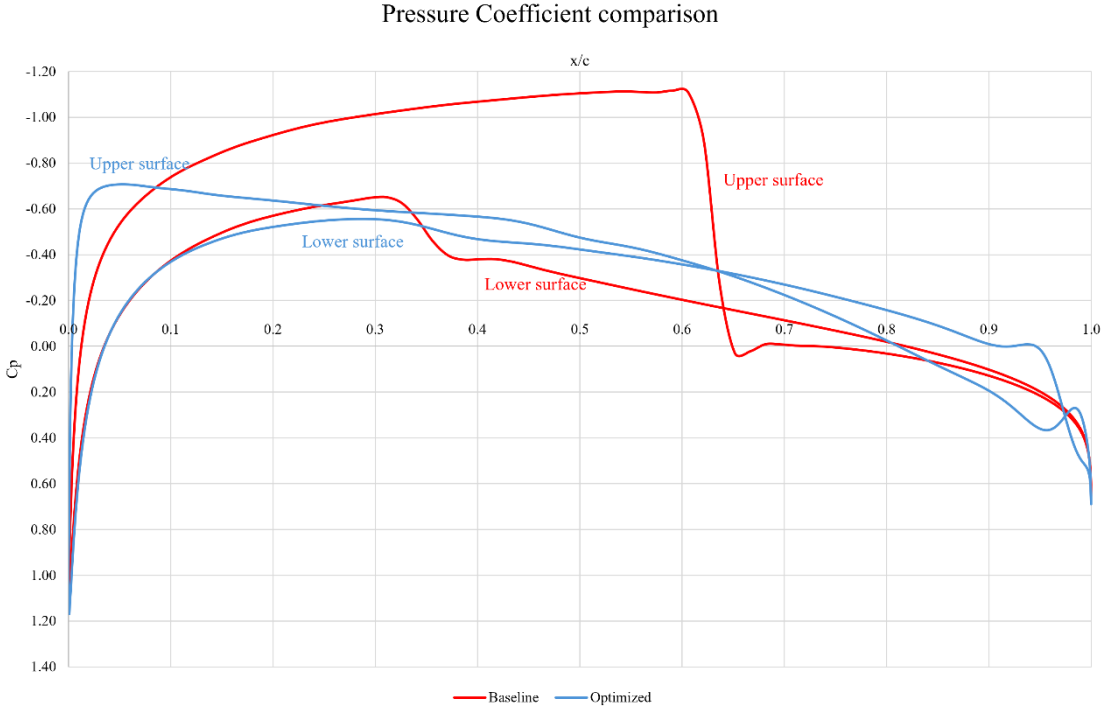

Figure 35. Pressure coefficient comparison between baseline and modified geometry for Study Case 1.

3.1.9.3 Geometric Comparison

Finally, both geometries can be compared by looking at Figure 36. The unconstrained design problem has resulted in the reduction of the thickness (initial thickness 12%) of the optimized airfoil (final thickness 10.5 %) in order to reduce shocks.

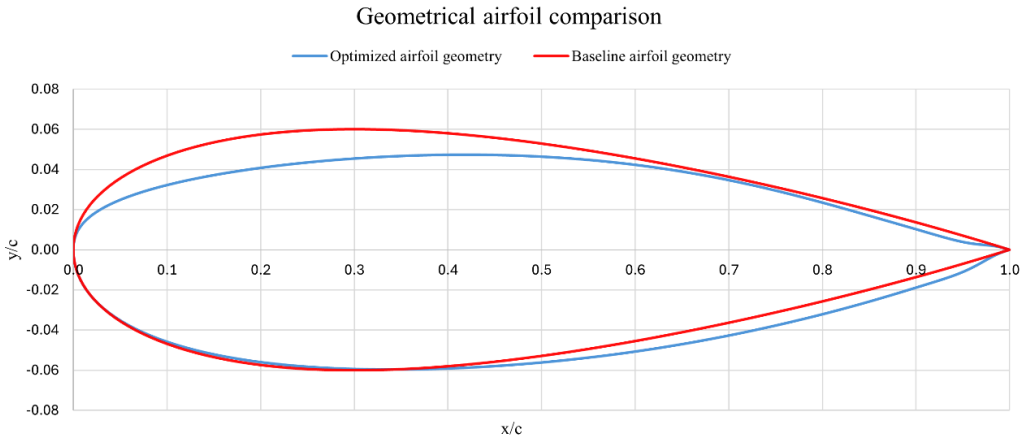

Figure 36. Geometrical airfoil comparison between optimized and baseline airfoils for Study Case 1.

3.1.9.4 Adjoint sensitivities

One of the keys of this project is the use of the adjoint method to compute sensitivities that are used by the gradient-optimizer to search for the optimal geometric deformation. The solutions to the adjoint equations which are derived from the flow equations are able to give density contours which give insights of the objective function sensitivity respect to changes in the airfoil surface as seen in Figure *37*. This contour plot indicates the regions around the airfoil surface that would have higher influence in the objective function (drag for this case) and therefore indicate the direction in which the airfoil should be modified to improve performance.

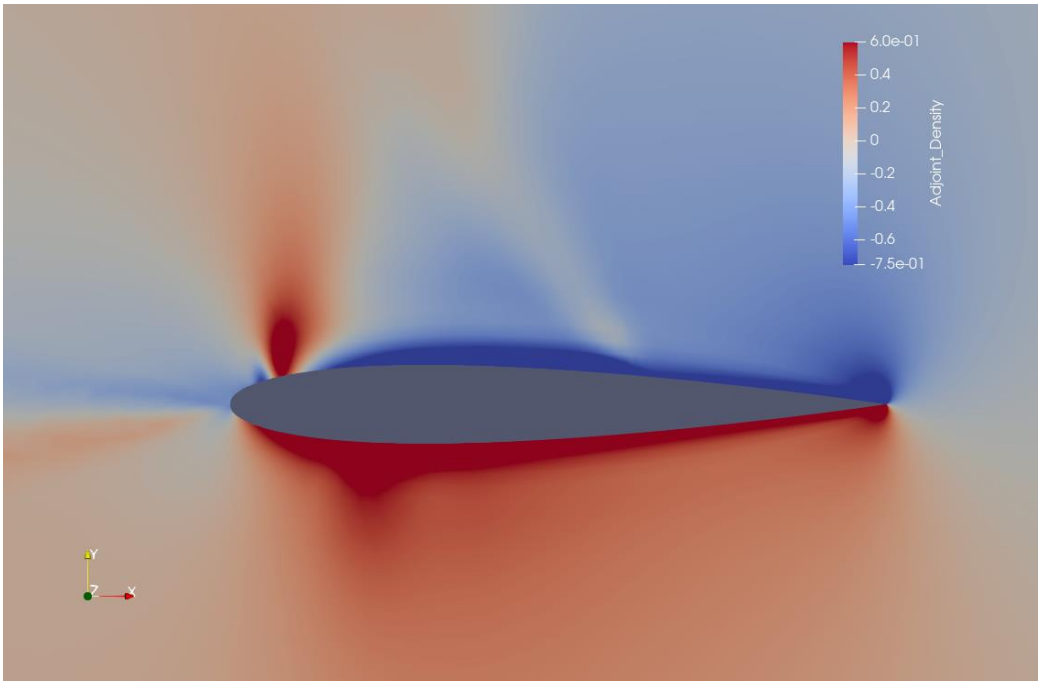

Figure 37. Adjoint-Drag density contours.

3.2 Study Case 2: Constrained Drag minimization of NACA 0012 in transonic inviscid flow for different Mach Numbers.

The previous introductory study to the optimization framework of SU2 shows one of the main fundamentals in autonomous optimization, the need to identify the issues that give the problem a real physical utility to produce solutions that can adapt to real cases. This concept is fundamental and requires an engineer to limit the algorithm capabilities as otherwise it would take advantage of any characteristic of the problem to improve and minimize the objective function.

From the unconstrained case, two main issues that have important aerodynamic meaning but not from the engineering point of view can be identified:

- **Cl reduction**: as the optimizer has identified, reducing drag for transonic problems has its main consideration in reducing the local peaks of low pressure and acceleration of the flow which are on the other hand important considerations on the lift generation, so a significant reduction was produced as a result of the minimization of drag. From the engineering point of view, this supposes a big problematic, as the main function of an airfoil is to be capable to generate the right amount of lift for some specific conditions to allow the aircraft to fly, which if not satisfied results in a poor design. For this reason, this study case will be run for a constant lift coefficient (allowing to vary the angle of attack) by setting the objective Cl to 0.6, which is a value chosen as the higher bound which can be found in commercial flight (typical cruise Cl range is 0.4 – 0.6) [37].

- **Thickness reduction**: drag for transonic flight is also influenced by the relative thickness that the airflow sees, so the lower the thickness/chord ratio of the airfoil the slenderer the wing will be and the lower acceleration due to the curvature [38]. In this way, the optimizer has come with the same conclusion as the final thickness was reduced compared to the baseline. This is also an important concern which has to be limited, as the thickness of a wing has structural and volumetric capacity considerations (volume to carry fuel). For this reason, another constrain to the problem will consist of maintaining a minimum airfoil thickness which is defined to be greater than the initial one, knowing that close decimal values are accepted so thickness will be greater or almost equal to the initial.

One of the main drawbacks of constraining the problem is the higher computational time and cost, as the optimizer has to consider these effects, which can lead to poor convergence or even result in problems with small improvement or not solution if the constrains are very demanding.

For this study case the same mesh will be used but the configuration file will be modified to introduce these two aspects.

On the other hand, this case study has the objective to better understand the Mach effects on the optimization framework of SU2 so the same configuration file for three different Mach numbers will be used.

The required files are therefore those from Figure *22* but changing the configuration file with the settings that will be later explained and being renamed "inv_NACA0012_constrained". SciPy and NumPy are also required but as they were compiled in the previous example, they should be available for this case.

3.2.1 Problem Statement

The formulation of this optimization problem from a mathematical point of view is:

$$\begin{aligned} &\textit{Minimize } Cd \\ &\textit{Subject to} \\ &\qquad t/c \geq 0.12 \\ &\qquad Cl = 0.6 \end{aligned}$$

3.2.2 Freestream Conditions

For this study case same airflow conditions will be used to ensure shock wave creation, but a wider range of flight speeds defined by three different Mach numbers will be tested.

Pressure	101325 Pa
Temperature	288.15 K
Mach Number	0.6 0.7 0.8

Table 3. Airflow conditions for Study Case 2.

3.2.3 Solver

Euler equation and continuous adjoint methodology (see Figure *23*) is also used for this case.

3.2.4 Geometry and Mesh definition

The same geometry and mesh as in the previous example is used for this case as the convergence was shown to be good and also as a way to make the process unaffected by this issue.

3.2.5 Airfoil Parametrization

As demonstrated by the previous example and to fix an important part of the problem (to maintain the same number and way to define design variables), the parametrization will be done by the same means, using Hicks Henne bump functions with locations as shown in Figure *25*.

3.2.6 Key configuration settings

The first key setting is related to the limitation of the lift coefficient value, by fixing this parameter using the "Cl driver definition" (shown in Figure *38*), This option allows to run at a constant lift value, by updating the angle of attack (therefore as another design variable) of the simulation such as the resultant Cl matches the one specified as the target Cl. "DCL_DAPLHA" is the proportional constat used to compute the change in AoA, the "UPDATE_AOA_ITER_LIMIT" refers to the number of iterations between angle of attack updates and the "ITER_DCL_DALPHA" is used to define the number of iterations to calculate dCl/dAlpha at the end of the simulation, which is used by the adjoint to calculate more accurate gradients for this fixed CL mode.

```
% ------------------------- CL DRIVER DEFINITION ----------------------------%
%
% Activate fixed lift mode (specify a CL instead of AoA, NO/YES)
FIXED_CL_MODE= YES
%
% Target coefficient of lift for fixed lift mode (0.80 by default)
TARGET_CL= 0.6
%
% Estimation of dCL/dAlpha (0.2 per degree by default)
DCL_DALPHA= 0.2
%
% Maximum number of iterations between AoA updates
UPDATE_AOA_ITER_LIMIT= 100

% Number of iterations to evaluate dCL/dAlpha at the end of the simulation
ITER_DCL_DALPHA= 500
%
% Evaluate dObjFunc/dCL during runtime (YES) or use the value stored in the
% direct solution file (NO).
EVAL_DOF_DCX= NO
```

Figure 38. Cl driver definition setting in the configuration file.

In order to constrain the airfoil geometry, the configuration template provided by SU2 gives the options shown in Figure *39*. For this case, the "AIRFOIL_THICKNESS" the preferred option, being this at least as thick as NACA 0012 as shown in Figure *40*.

```
% Available geometrical based objective functions or constraint functions
%    AIRFOIL_AREA, AIRFOIL_THICKNESS, AIRFOIL_CHORD, AIRFOIL_TOC, AIRFOIL_AOA,
%    WING_VOLUME, WING_MIN_THICKNESS, WING_MAX_THICKNESS, WING_MAX_CHORD,
%    WING_MIN_TOC, WING_MAX_TWIST, WING_MAX_CURVATURE, WING_MAX_DIHEDRAL
%    STATION#_WIDTH, STATION#_AREA, STATION#_THICKNESS, STATION#_CHORD, STATION#_TOC,
%    STATION#_TWIST (where # is the index of the station defined in GEO_LOCATION_STATIONS)
```

Figure 39. Set of available geometric objective and constraint functions for SU2 configuration file.

```
% Optimization constraint functions with pushing factors (affects its value, not the gradient
% in the python scripts), separated by semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= ( AIRFOIL_THICKNESS  > 0.12 ) * 0.001
```

Figure 40. Thickness constraint definition in the configuration file.

3.2.7 Numerical Results

For each case, numerical results will be shown and compared to understand how different Mach numbers can affect the convergence of the solution, the thickness and the drag obtained as a result.

3.2.7.1 MACH 0.6

This airspeed represents the initiation of transonic flow in the airfoil, where shock waves can start to appear, but their effect on drag is much smaller. As seen from Figure 41, the initial drag of this airfoil is much lower than for Case Study as the drag is much less dominated by the shock effects, so it can be considered normal that the optimization margin has been significantly reduced to a 24.66 % of drag reduction.

On the other hand, Figure 41 also shows that the convergence is not fully linear and has a very dispersed drag evaluation consequence of the big change in geometry between two iterations. However, good convergence is achieved with a very stable value. It is also important to notice that the final drag coefficient is significantly smaller than for Study Case 1 optimization probably due to the constrained problem.

Figure *42* and Figure *43* show the variation of the two constrained parameters as a function of the number of evaluations, showing a very stable performance for lift coefficient but showing some variation for the thickness with a big step that matches that of the high drag evaluated, which is then stabilized to a value of 0.14 compared to the initial 0.12.

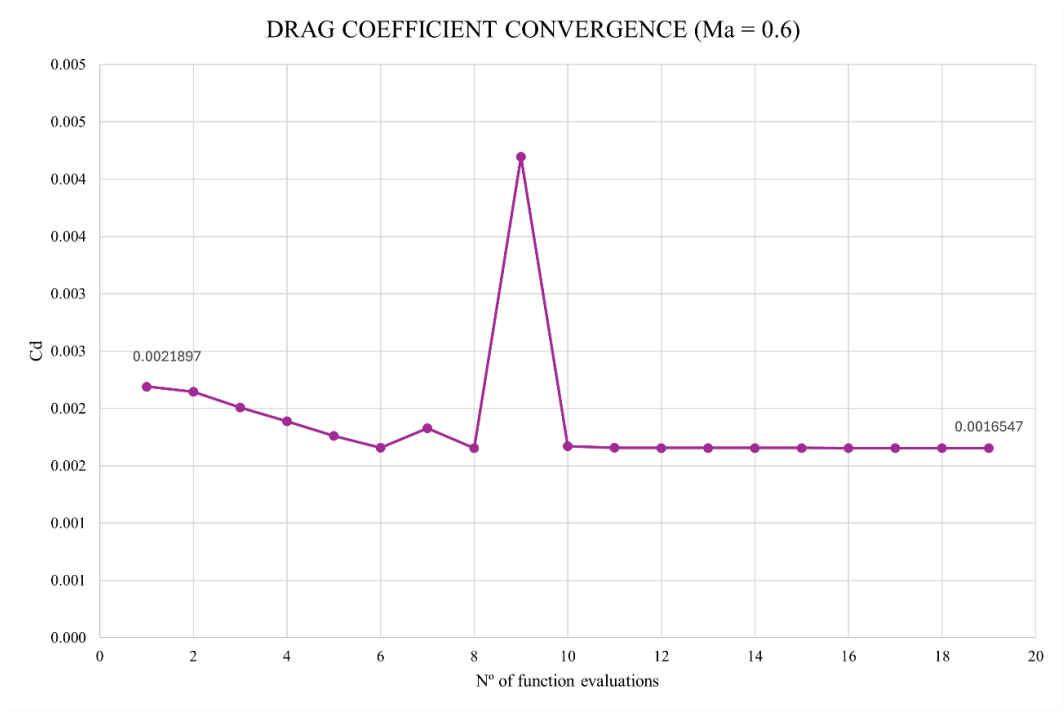

Figure 41. Drag coefficient convergence for Study Case 2 (Ma = 0.6).

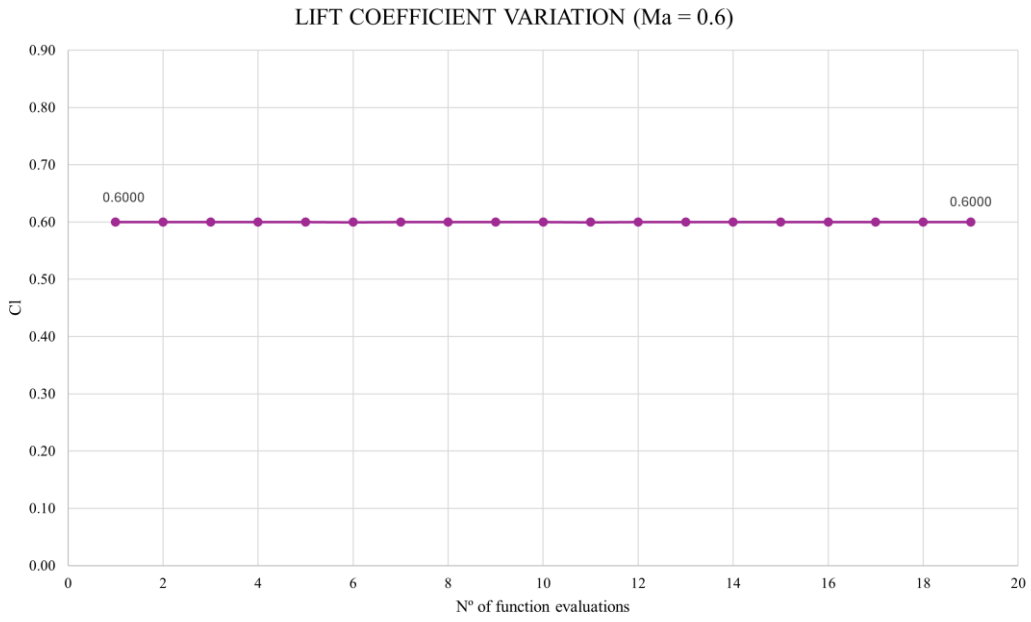

Figure 42. Lift coefficient convergence for Study Case 2 (Ma = 0.6).

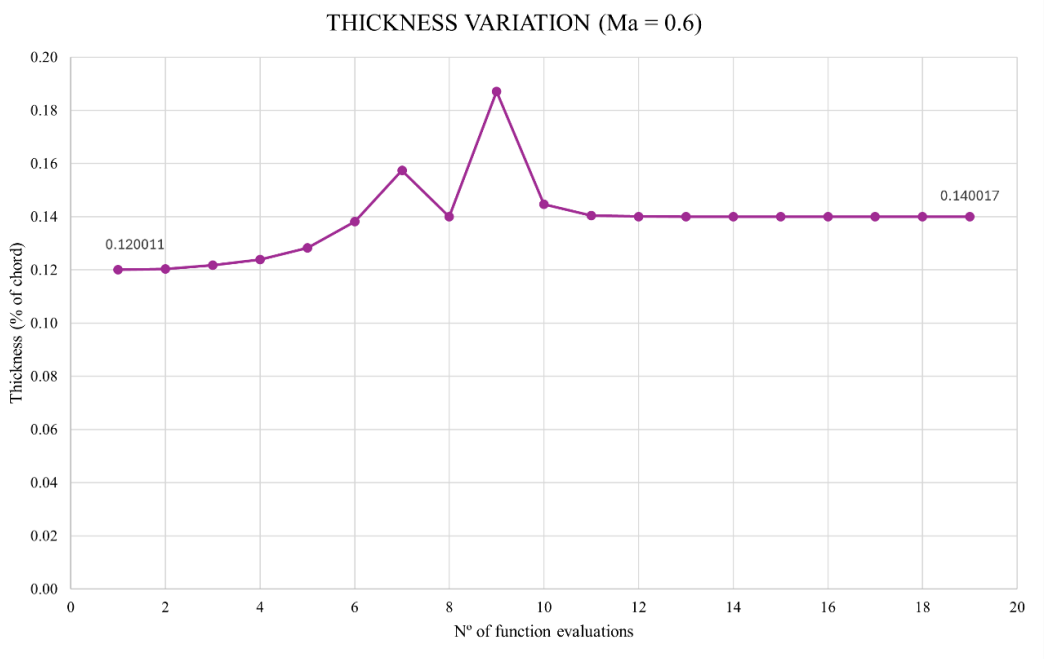

Figure 43. Thickness variation as a function of the number of function evaluations for Study Case 2 (Ma = 0.6).

3.2.7.2 MACH 0.7

For this second Mach evaluation, results are significantly different as expected. At this Mach, shock effects are much more significant, so the optimization makes much more sense as a transonic inviscid case. This has allowed for better improvement of the objective function corresponding to an 86.32 % from the initial value starting form a much higher drag coefficient (see Figure 44).

As observed from Mach 0.6 optimization, there is a point where drag evaluations show fluctuations that can be attributed to higher changes in thickness as shown from Figure *46*, which for this high speed produce undesired shock appearance that significantly affect the drag coefficient. However, the optimizer is also able to reach convergence after these fluctuations and give as a solution a drag value slightly above that from Mach 0.6 case.

Lift coefficient as seen from Figure 45 is very stable in the total number of evaluations and does not seem to be affected by drag fluctuations.

The final thickness at which convergence is found corresponds to a 4.3 % increase, which can be attributed to the need to reduce the total curvature on the upper surface which increases the total thickness.

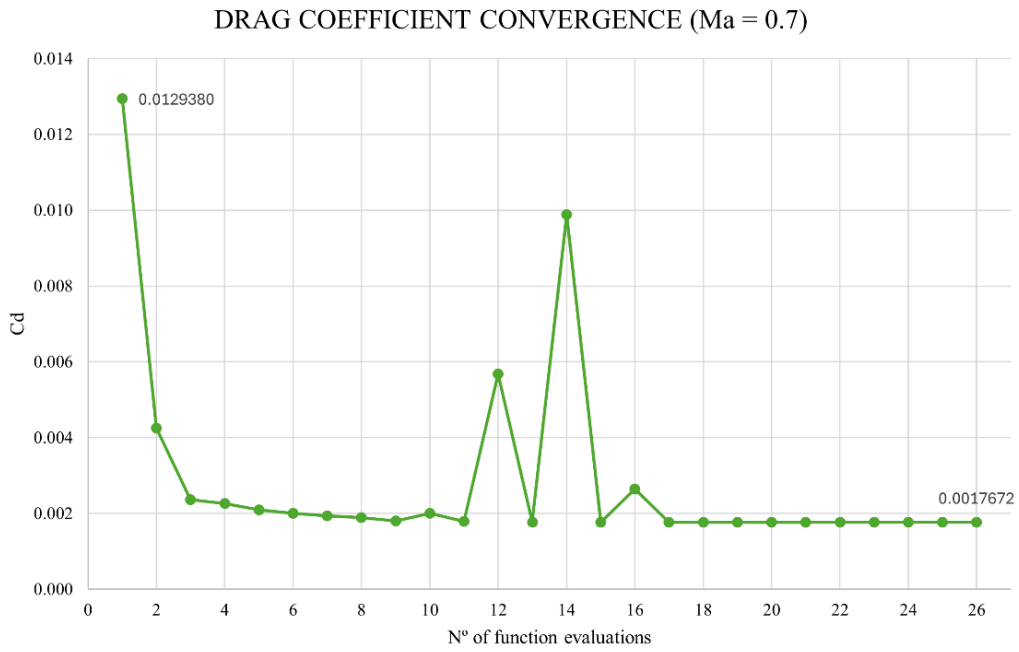

Figure 44. Drag coefficient convergence for Study Case 2 (Ma = 0.7).

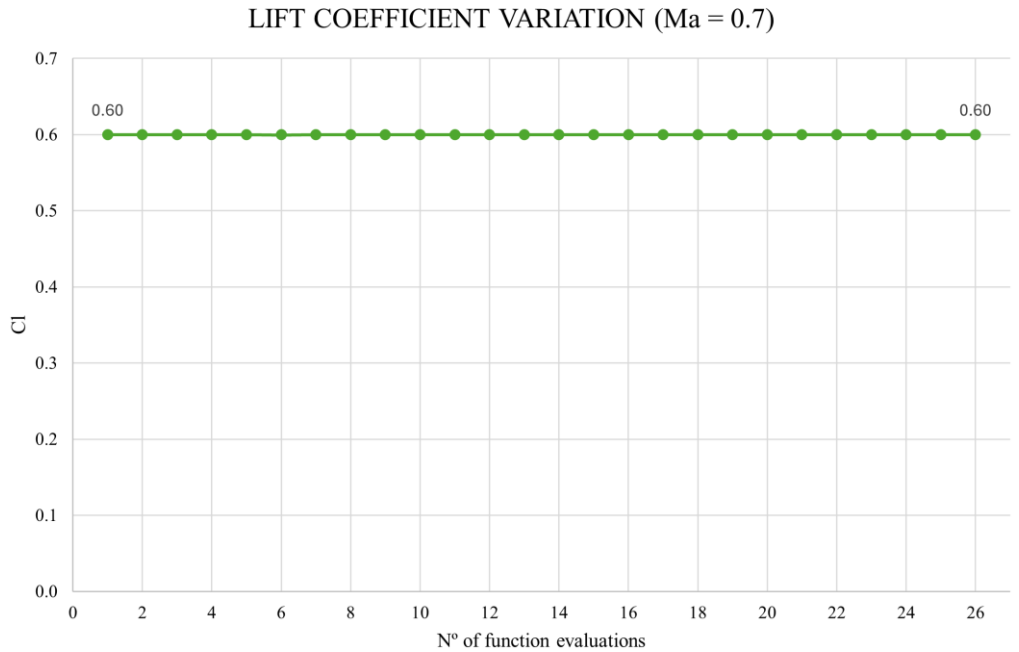

Figure 45. Lift coefficient variation as a function of evaluations for Study Case 2 (Ma = 0.7).

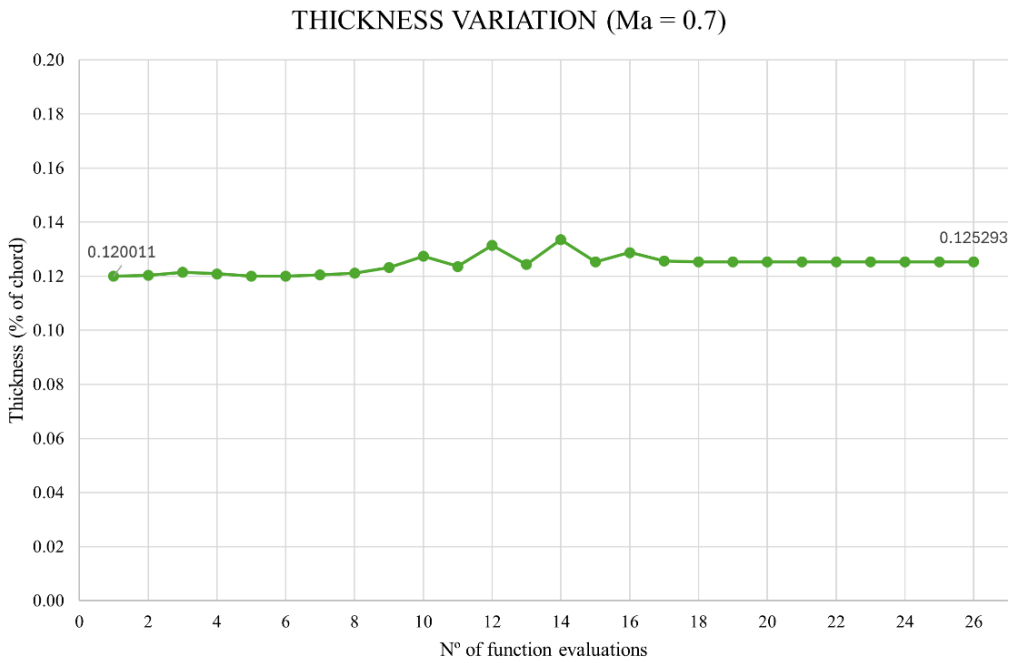

Figure 46. Thickness variation as a function of the number of evaluations for Case Study 2 and (Ma = 0.7).

3.2.7.3 MACH 0.8

For this case, the convergence process shows even higher fluctuations (see Figure 47) than for the previous, as any change in the geometry has higher impact and the possibility to create shocks than for lower speeds. In this way, the sensitivity analysis and the gradient based optimization is more complex in general. However, initial fluctuations reduce drag coefficient continuously after each two evaluations reaching a point close to the minimum, then a very high fluctuation is observed but then the value stabilizes and finally converges.

As defined to run simulations at constant lift coefficient with variations in the angle of attack, the drag coefficient for the baseline Naca 0012 is higher than for the baseline of Case Study 1. The improvement for this optimization is similar to the previous case and much higher than for first Mach, resulting in an 87.03 % drag coefficient reduction. Due to the constrained problem, this final converged value is significantly higher than for the unconstrained optimization, which shows an expected behaviour.

In relation to the variations of lift coefficient, fluctuations appear as seen in Figure 48, which weren't present in the two previous cases. These fluctuations in general match the drag coefficient variations but aren't directly related in the same magnitude change.

Finally, thickness is almost not varied for the whole set of function evaluations except for a single point that matches the high drag coefficient peak (see Figure 49). The final thickness for this airfoil almost does not vary from the initial, showing a clear tendency to approximate to the minimum permitted by the problem definition in order to find the optimum solution.

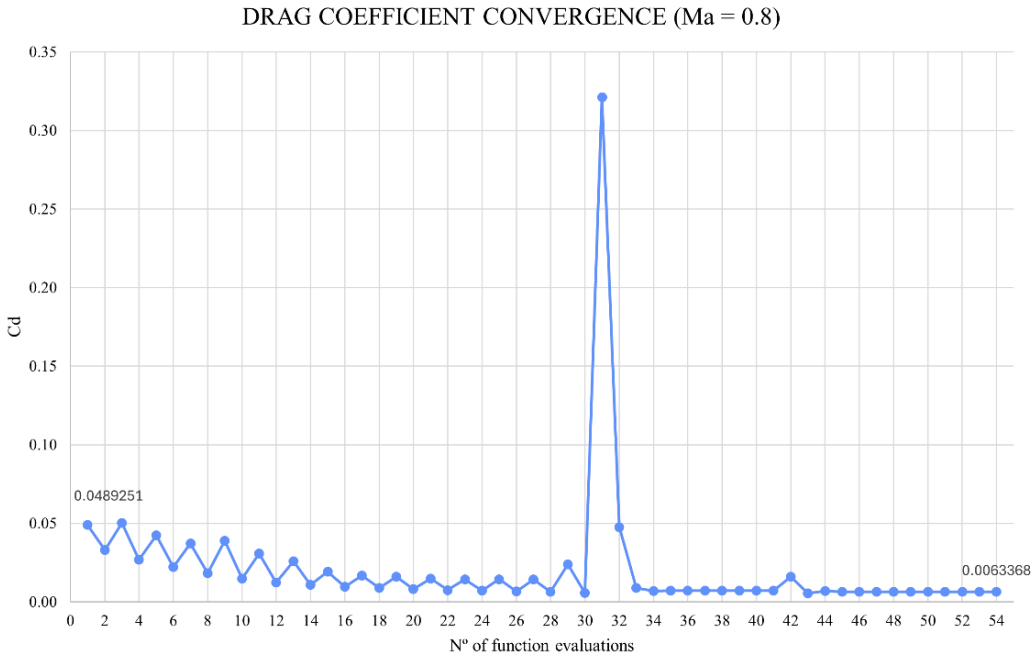

Figure 47. Drag coefficient convergence for Study Case 2 (Ma = 0.8).

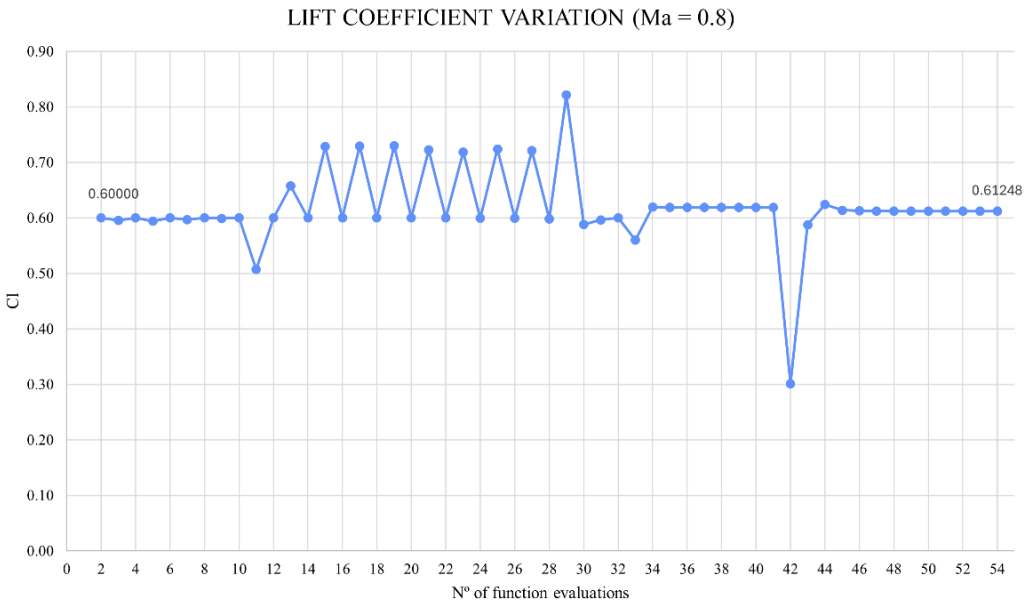

Figure 48. Lift coefficient variation as a function of number of function evaluations for Study Case 2 (Ma = 0.8).

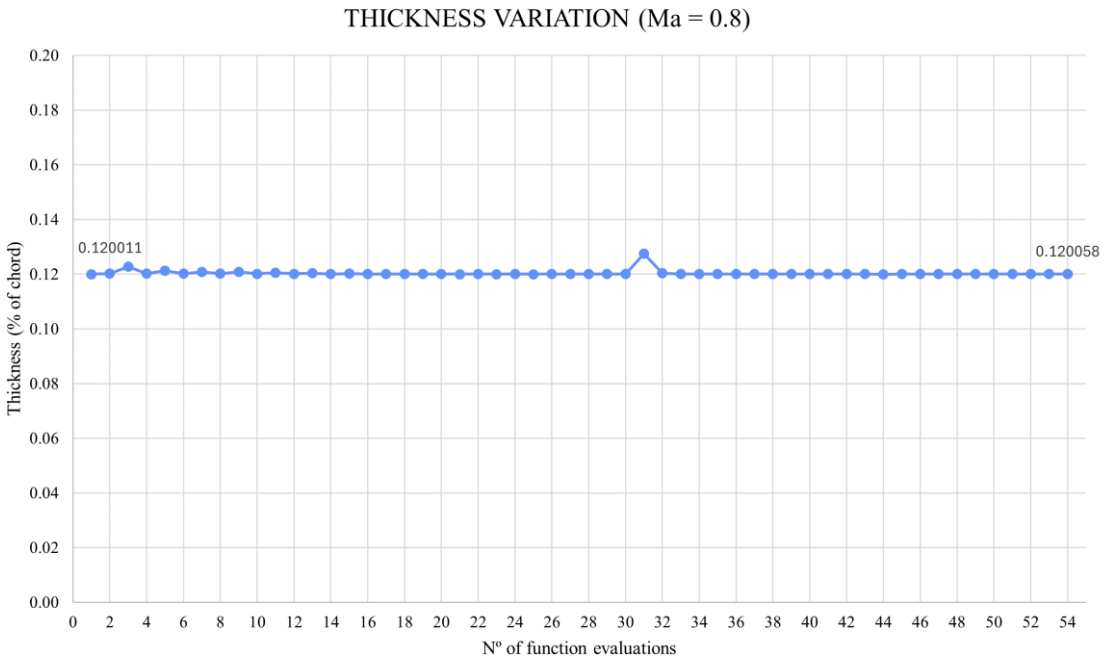

Figure 49. Thickness variation as a function of the number of function evaluations for Study Case 2 (Ma = 0.8).

3.2.8 Post Processing Results

Also, post processing visualization of Mach contours and pressure distribution around the airfoils will be discussed for each Mach number and compared to its baseline geometry under the same flow conditions.

3.2.8.1 MACH 0.6

Thanks to the visualization of the flow, it is possible to identify the behaviour of the optimization process for Ma = 0.6 nearly as it was expected from Figure 11. At this airflow conditions, the baseline airfoil shows a small transonic behaviour. However, the initiation of a supersonic region near the leading edge at (5 % of the chord approximately), which can be appreciated from Figure 54. This is a result a small local region of low pressure or acceleration that is reflected in the upper surface in the pressure coefficient contour from Figure 52. The rest of the pressure distribution is homogeneous reducing the difference of pressures between surfaces between the section comprising this local peak and the trailing edge.

For the optimized airfoil, the result is interesting in the way that it is highly influenced by the lift coefficient desired as for the optimal geometry, thickness and curvature increase (as seen in Figure 53) but flow is less accelerated. This behaviour can be explained because the simulation is stated for a fixed lift coefficient but with a varying angle of attack, so this angle reduced for the

optimized case than for the baseline, resulting in lower acceleration and lower peak of low pressure (see Figure 52). For the optimum design, pressure distribution is more homogenous reducing the leading-edge supersonic region but with a larger region of lower pressure in the upper surface.

On the trailing edge of the optimized airfoil, there is a small kick, so the pressure has a different distribution than for the rest of the chord.

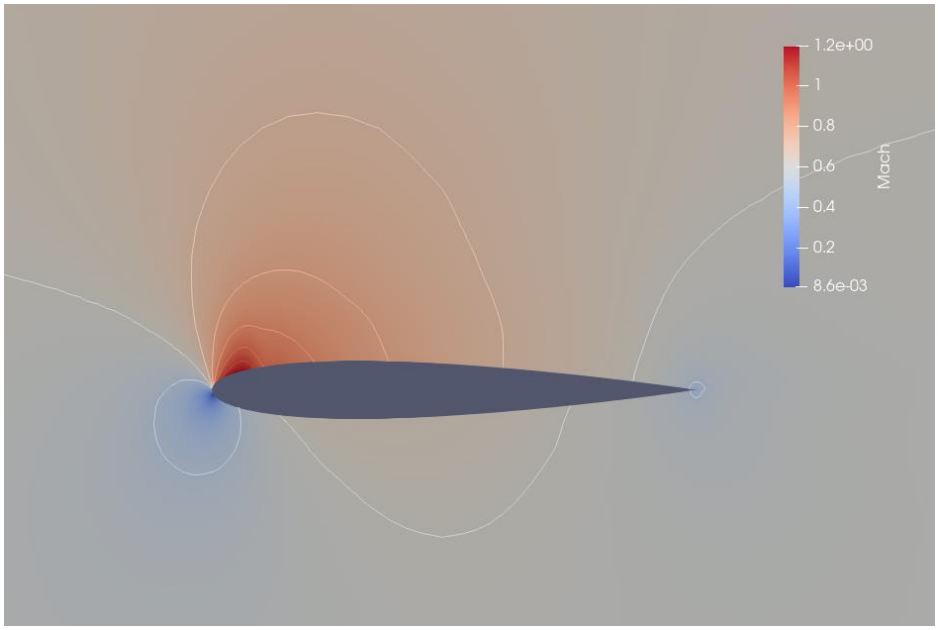

Figure 50. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.6).

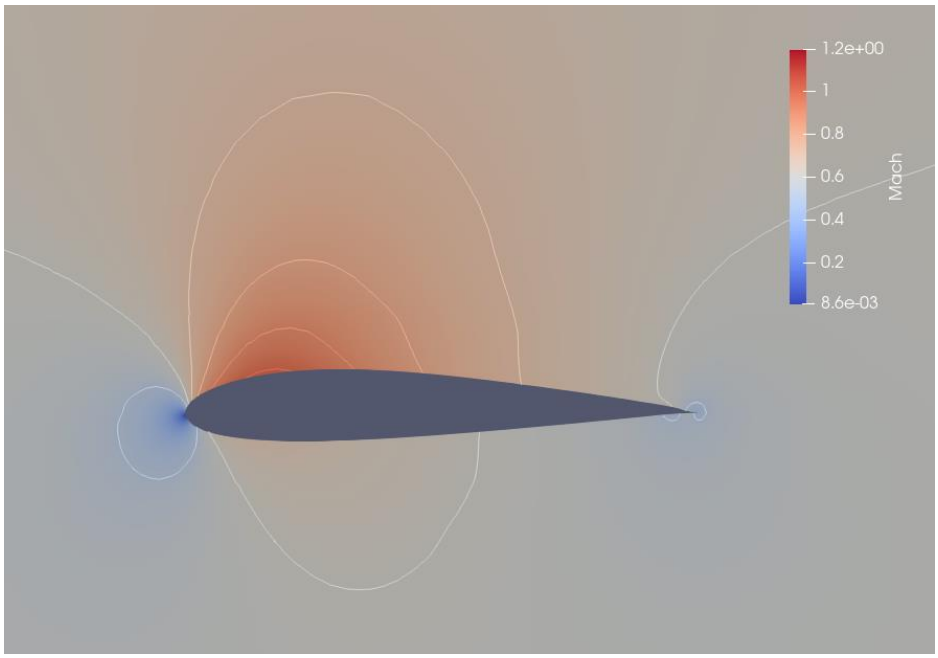

Figure 51. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.6).

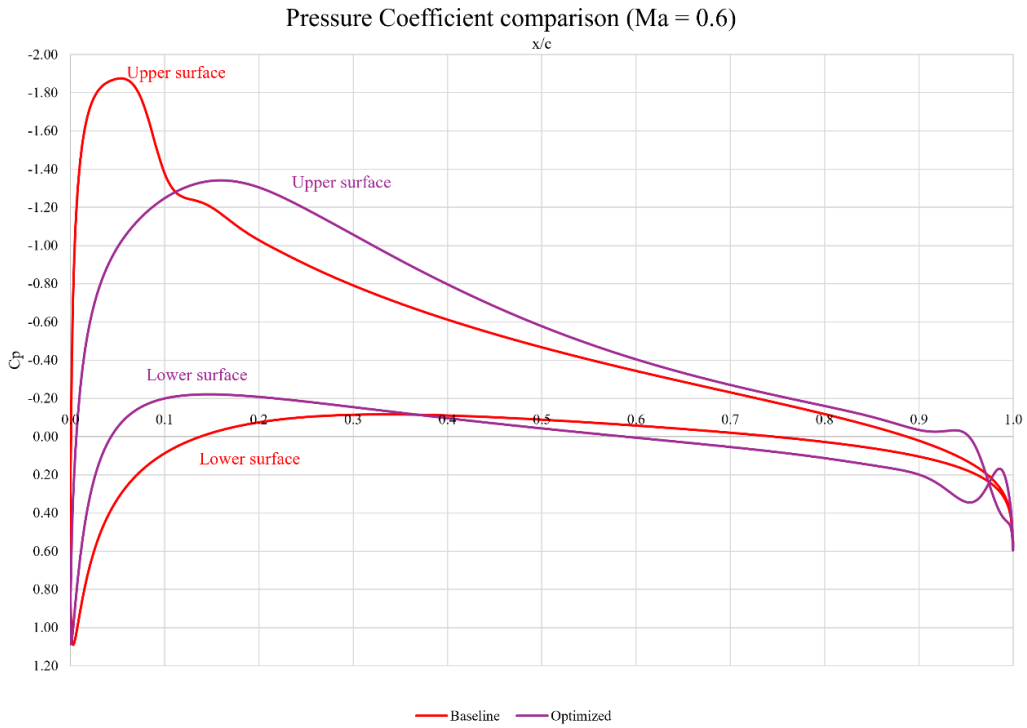

Figure 52. Pressure coefficient comparison between baseline and modified geometry for Study Case 2 (Ma = 0.6).

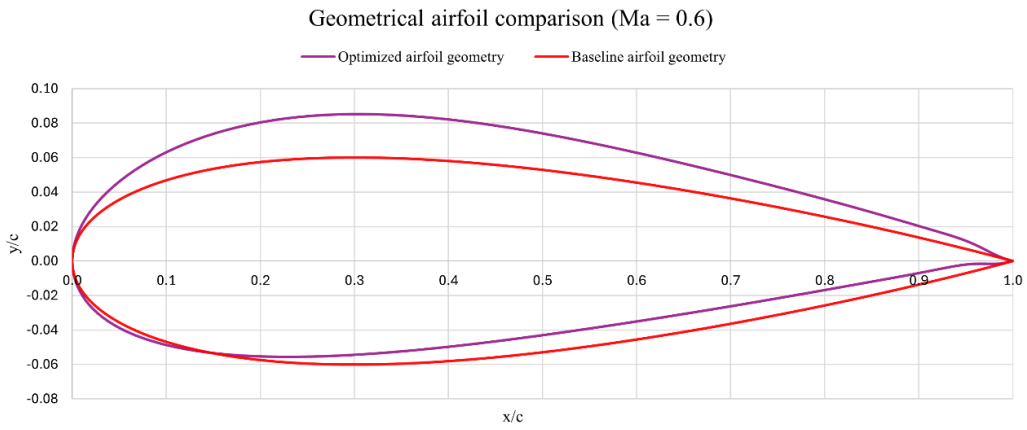

Figure 53. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Mach = 0.6).

3.2.8.2 MACH 0.7

From this second optimization, the higher Mach number influences the shock wave formation by extending backwards (see Figure 54) and increasing the strength of the shock as expected and previously introduced in Figure 11. This effect can also be appreciated from Figure *56*, where the upper surface pressure distributed is characterized by a larger low-pressure region up to 0.35 % of the chord and then reduced after the shock.

Instead, for the optimized airfoil, this stronger shock wave is divided into a first very weak shock and a secondary small acceleration that are a result of two local peaks of lower pressure as seen in Figure 56. The geometry optimization has been done by reducing the upper surface curvature at the beginning with a much flatter section, and then to comply with the lift coefficient imposed condition a higher curvature is used (while having similar thickness) in a downwards direction after approximately the 55 % of the chord (see figure Figure 57).

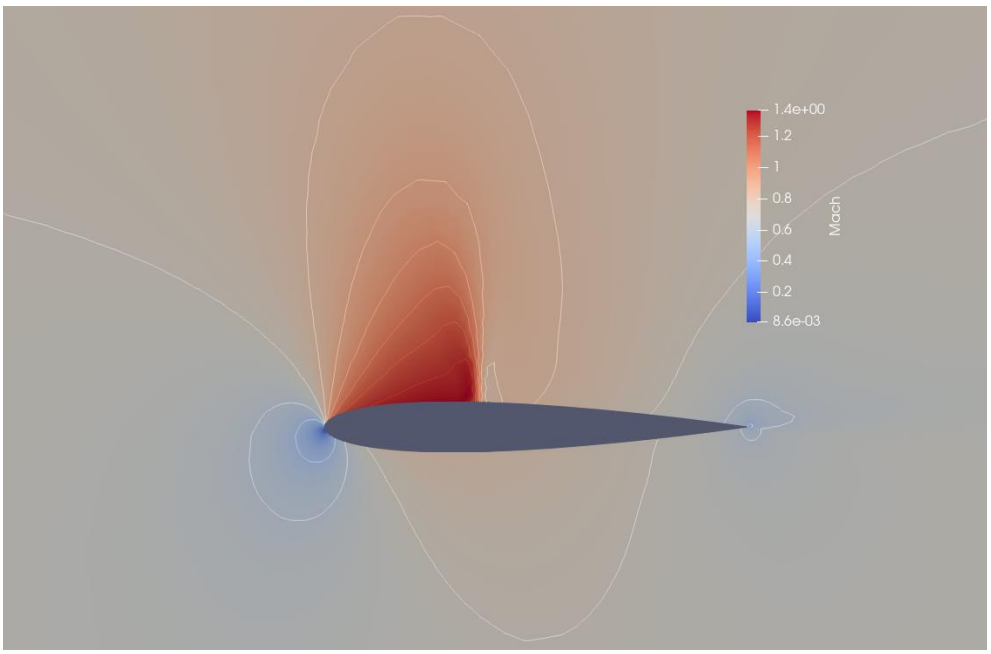

Figure 54. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.7).

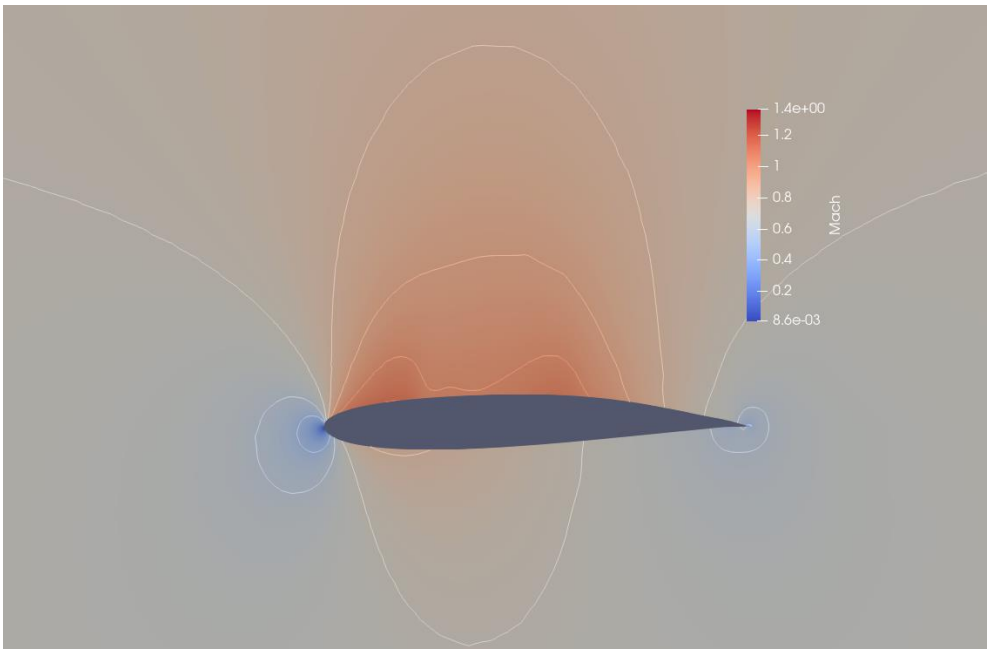

Figure 55. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.7).

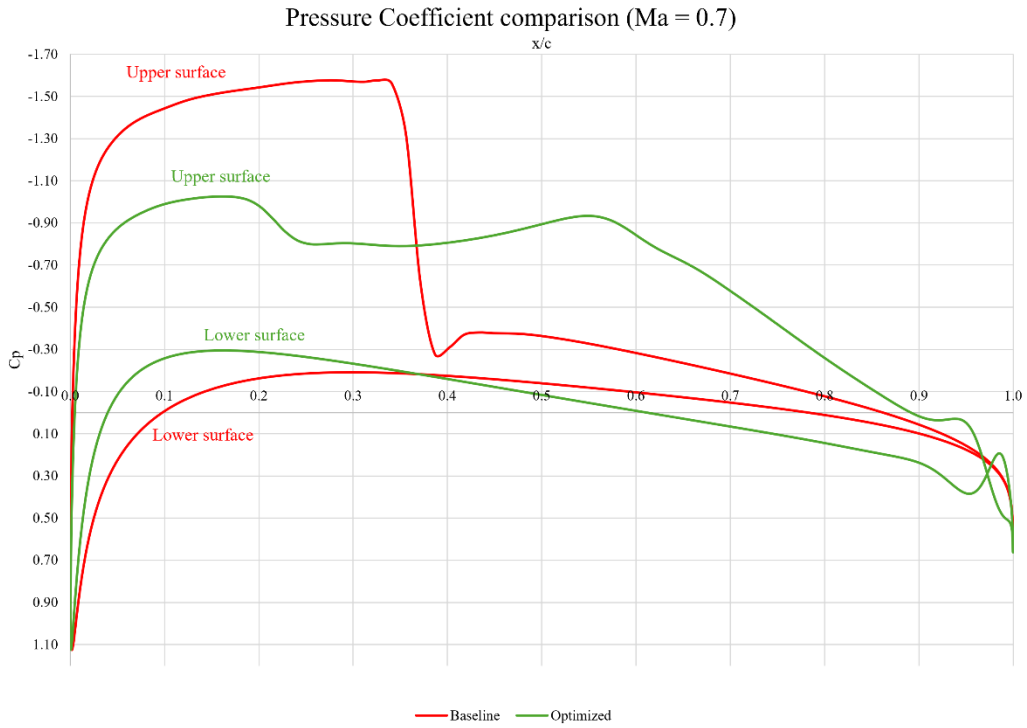

Figure 56. Pressure coefficient comparison between baseline and modified geometry for Study Case 2 (Ma = 0.7).

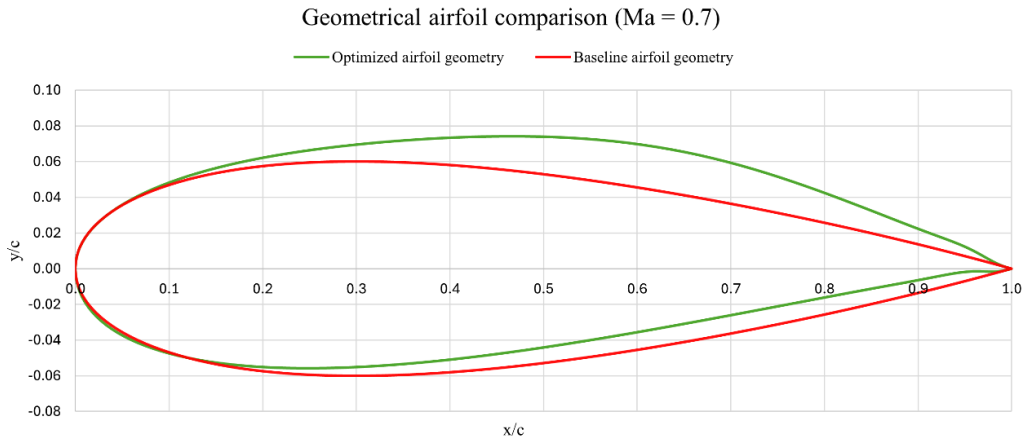

Figure 57. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.7).

3.2.8.3 MACH 0.8

This third case represents a fully transonic flight regime, where a strong shock is originated further aft (see Figure 58 and Figure 60) compared to the previous analysed cases. Airflow is highly accelerated in the upper surface reaching to speeds further that of sound up to the point located approximately at 70 % of the chord where the shock appears.

As compared with the Study Case 1, there is no lower surface weak shock, as for the first iteration the simulation is set to run the specified lift coefficient but with a possible angle of attack variation and therefore the flow is higher accelerated in the upper side and less in the lower surface. However, compared to the two previous Mach conditions, the lower surface exhibits lower pressure and higher speeds as expected.

For the optimized geometry, it is clearly shown that the tendency has been to reduce the upper curvature by using a long flat upper section, ending in a downward kick (see Figure 61) that is contributing in a big part to the downwash of the air, the low pressure region and therefore the lift generation. However, this flat region also exhibits acceleration of the flow creating a high Mach region that extends up to a weak shock at the back.

Due to the thickness geometrical constraints, the flatter upper side has to be compensated in some manner, which results in a lower surface belly with high curvature that is the cause of the weak shock originated on this side and that can be clearly appreciated from the sudden drop of pressure in the surface pressure distribution at 40 % of the airfoil chord.

Results from the optimized airfoil are very interesting as the resultant airfoil is very similar to that used in Figure 13 to illustrate the general philosophy of supercritical airfoils, producing the expected same shock wave effects and the belly and kicks in similar ways.

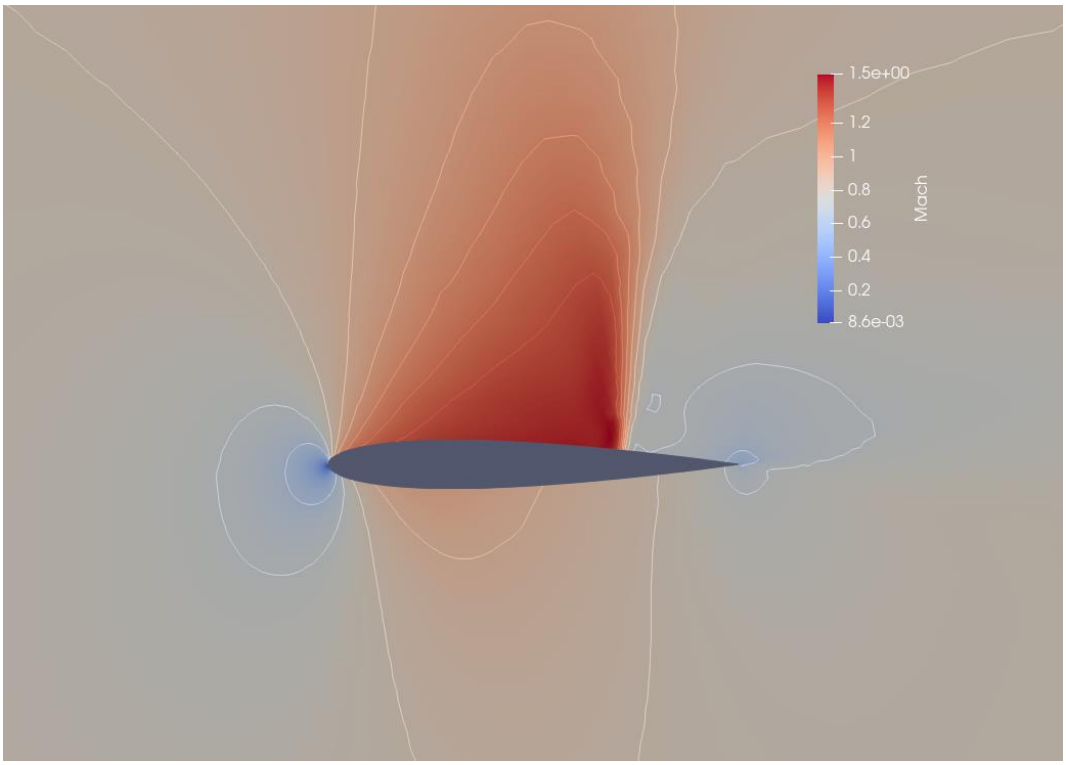

Figure 58. Mach number contour for baseline airfoil for Study Case 2 (Ma = 0.8).

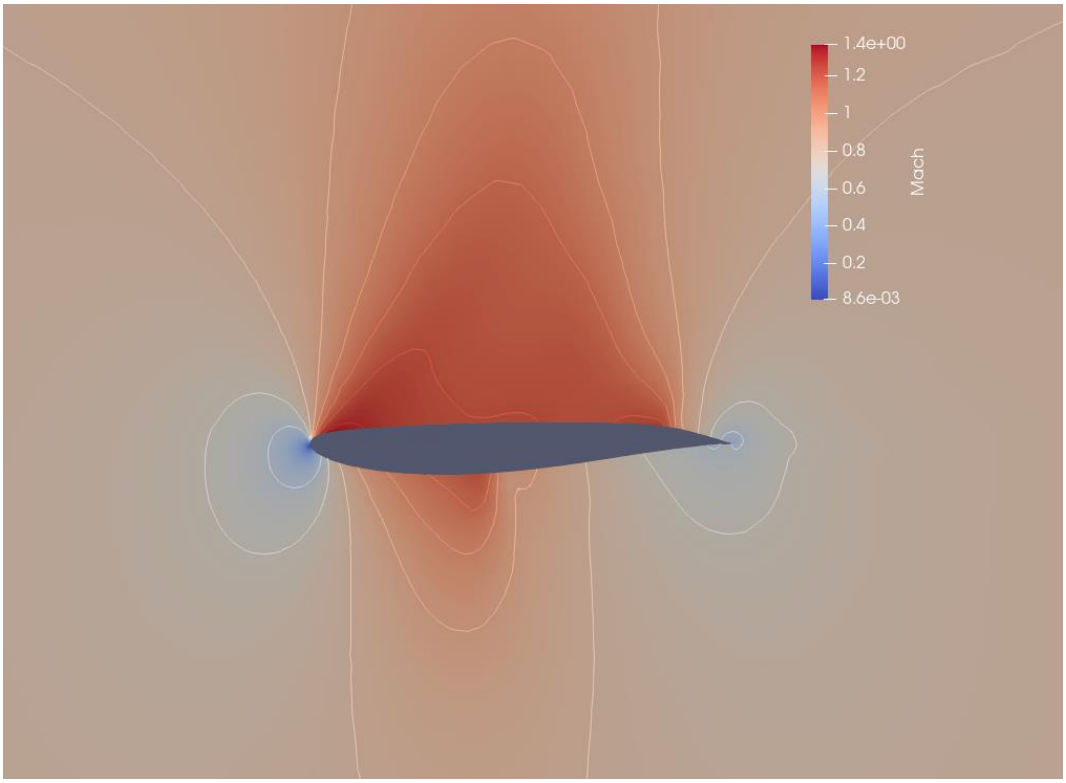

Figure 59. Mach number contour for optimized airfoil for Study Case 2 (Ma = 0.8).

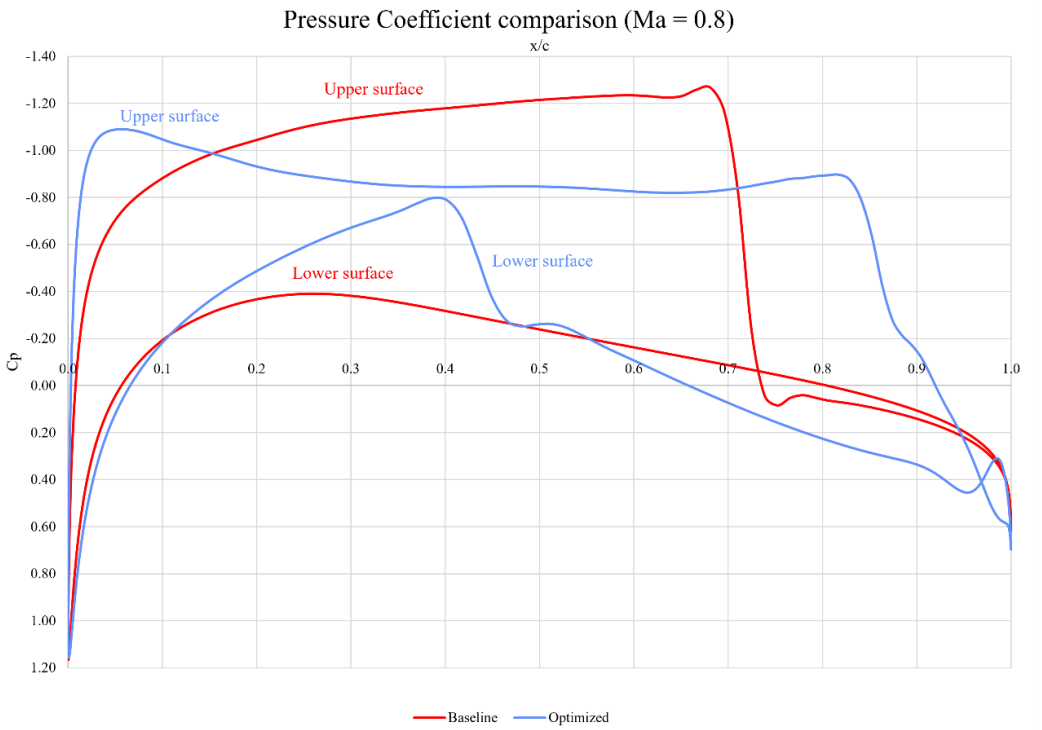

Figure 60. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.8).

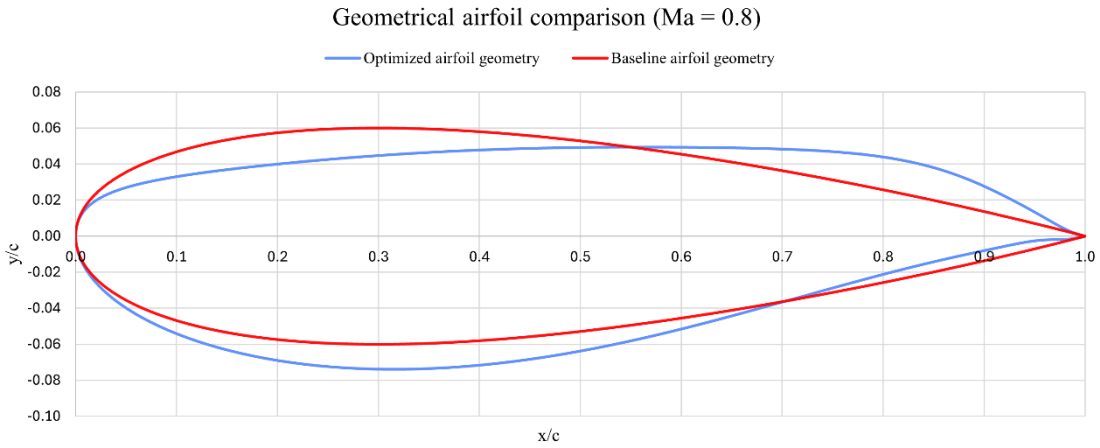

Figure 61. Geometrical comparison between baseline and optimized airfoils for Study Case 2 (Ma = 0.8).

3.2.9 Results Comparison

Table 4 shows a comparison for drag coefficient optimization for the three previous Mach number cases, allowing to determine the effects of Mach number in the optimizer behaviour.

Ma	Initial Cd	Opitmized Cd	Reduction Cd (%)	Function Evaluated
0.6	0.00219	0.00165	24.66	19
0.7	0.01294	0.00177	86.32	26
0.8	0.04892	0.00634	87.03	54

Table 4. Comparison for drag coefficient optimization and number of functions evaluated for Ma = 0.6, Ma = 0.7 and Ma = 0.8 for Study Case 2.

From this table it can be appreciated that the lower the Mach number is, the lower the initial drag coefficient. This behaviour is reasonable as the shock wave intensity is reduced. However, an important remark can be seen from the Ma = 0.6 case as there is no shock wave formed, the initial Cd is much lower than for the two other Mach cases.

In terms of optimized drag coefficient, values for Ma = 0.6 and Ma = 0.7 are very similar as the imposed conditions are less restrictive for the freestream conditions, however for the third case, this value is much greater as shock strength is higher even appearing in the lower surface.

For the minimisation of Cd (expressed as reduction of Cd in %) there is a clear benefit of applying this methodology to transonic airfoil behaviour. For the case of Ma = 0.6, as there is no shock

wave, reduction in Cd is much lower between baseline and optimized airfoils as the improvement comes from a more uniformly pressure distribution, but without dissipating any shock wave. However, for Ma = 0.7 and Ma = 0.8, as corresponding more to transonic flight regimes, the optimizer is able to produce larger improvements, being very similar for both cases independently of the variations in the initial and optimized Cd.

Finally, for the number of design evaluations it can be observed that the larger the Mach number the larger the number of evaluations needed to reach convergence due to larger number of fluctuations as a result of a higher sensitivity of the design to the generation of shock waves.

3.3 Study Case 3: Constrained Drag minimization of RAE 2822 in transonic viscous flow.

By the two previous study cases, it has been possible to understand the effects of the general optimization process, including constrains, convergence, and the Mach effects for supercritical airfoils. However, these studies were performed for inviscid considerations which can't capture the real effects of airfoils under these conditions. Therefore, this study case was performed for transonic viscous flow in order to capture boundary layer effects and obtain drag coefficient values more accurately.

This third case study correspond to the benchmark optimization problem that was proposed by the AIAA Aerodynamic Design Optimization Discussion Group (ADODG), which was founded as research collaborating community with some of the most important personalities in the aerodynamic design optimization framework to provide foundations for design optimization methods [39].

This group defined some test cases in order to provide with a problem formulation open for researchers from all over the world in order to find the solution with freedom to use optimization and parametrization methods. In this way, the community would have some clear problem statements to compare and contrast.

The third case for this project corresponds to the Case 2 called "Drag Minimization of the RAE 2822 in Transonic Viscous Flow", as the most complex case that was defined in [40] by the group for airfoil optimization.

The objective of this case study will be to approach the problem using the previously studied and defined methodology based on the continuous adjoint for gradient computation and Hicks-Henne bump functions as the parametrization method.

3.3.1 Problem Statement

As defined in [40], this benchmark problem has the same objective as in the previous two Study Cases of this project which is in minimizing drag. However, apart from lift, new restrictions for this problem are used as the intention is to constrain other important parameters such as the airfoil area or the pitching moment:

- **Moment coefficient**: airfoils with camber, apart from having lift, also have a moment which is a result of considering the lift force to act in the aerodynamic centre. It is important to consider this moment from the aircraft design point of view as this moment

and the one generated by the lift force to the centre of gravity will need to be compensated by a larger tail and therefore larger drag. Normally, pitching down moments are obtained for normal flight conditions so the restriction of this problem is to reduce nose-down pitching moment at 25 % of the chord to be higher or equal than -0.092.

- **Area**: in the previous Study Cases, thickness was selected as the geometrical restriction to match the structural needs of the baseline airfoil. However, for this case, the goal is to fix the are inside the airfoil, which would correspond to the volume of the wing when extended in 3 dimensions as shown in Figure 5, ensuring maintaining the same fuel capacity and also similar structural properties. The area of the optimized airfoil is therefore set to be higher or equal to that of the initial.

Using mathematical formulation, the problem statement is:

$$\begin{aligned} &\textit{Minimize } Cd \\ &\textit{Subject to} \\ &\qquad Cl = 0.824 \\ &\qquad Cm \geq -0.092 \\ &\qquad Area \geq Area_{initial} \end{aligned}$$

3.3.2 Freestream Conditions

Also, freestream conditions are given in [40] to set common problem conditions for researchers around the world. Mach number, temperature and Reynolds number are given as shown in Table 5 as well as the use of 1.4 as the ratio of specific heats and 0.7 for the Prandtl Number (notice that the way to define the freestream conditions is different from the two previous test cases due to the variation of the flow solver used):

Temperature	288.15 K (assumed as it is not defined)
Mach Number	0.734
Reynolds Number	6.5 $x10^6$

Table 5. Freestream conditions for Study Case 3.

3.3.3 Solver

As previously stated, RANS solver with Spalart-Allmaras turbulence model is used for this viscous case, being defined as shown in Figure 62.

```
% ------------ DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, FEM_ELASTICITY,
%                               POISSON_EQUATION)
SOLVER= RANS
%
% Specify turbulent model (NONE, SA, SA_NEG, SST)
KIND_TURB_MODEL= SA
%
% Mathematical problem (DIRECT, CONTINUOUS_ADJOINT)
MATH_PROBLEM= CONTINUOUS_ADJOINT
%
% Restart solution (NO, YES)
RESTART_SOL= YES
```

Figure 62 Solver and mathematical problem definition for Study Case 3.

3.3.4 Geometry and Mesh definition

The baseline Airfoil used is the RAE 2822, a supercritical airfoil widely used for CFD software validation for multiple research studies.

Instead of using an unstructured mesh, as outlined in 2.8.2 capturing boundary layer effects requires the higher local refinement near the airfoil surface, and the use of structured grids in these regions highly increases the accuracy where complex physical phenomena needs to be captured.

As seen in Figure 63, hybrid mesh obtained from [41] is used for a balance between computational cost and accuracy in a O-type far field boundary. The mesh is composed of 22,842 elements.

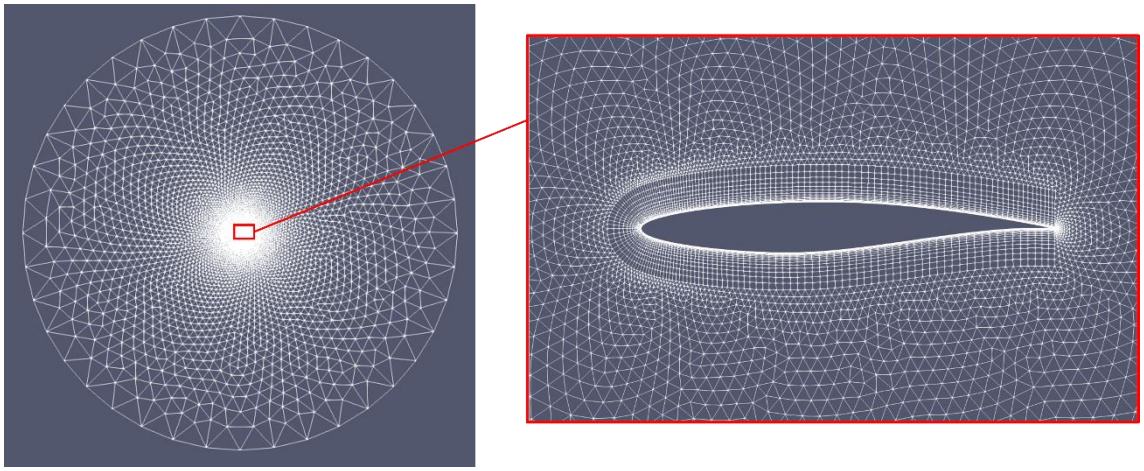

Figure 63. Hybrid mesh used for Study Case 3.

3.3.5 Airfoil Parametrization

The parametrization method that will be used is the same that for Study Case 2 (see Figure 25), as it has resulted to give proper results in smoothness and flexibility to deform the geometry, but in this case applied to RAE 2822 as seen in Figure 64.

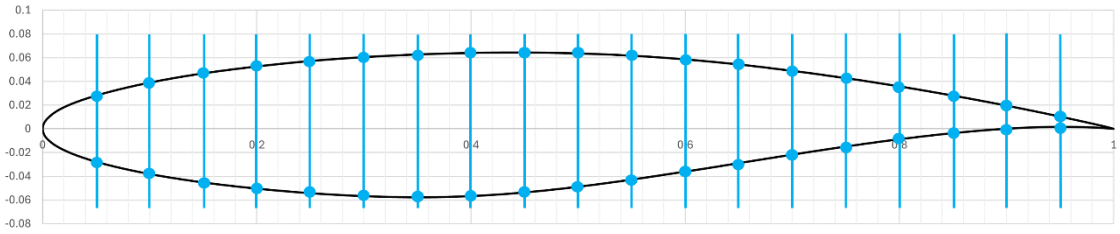

Figure 64. RAE 2822 Hicks-Henne bump locations.

3.3.6 Key configuration settings

The key configuration settings which are different from the previous two study cases are related to the RANS solver such as “Slope Limiter and Dissipation Sensor definition”. “Flow Numerical Definition” or “Turbulent Numerical Method Definition” choosing values for CFL and JST as those used from the configuration file used for SU2 tutorial found in [42].

The Cl driver definition is used as in Study Case 2 but for the specified value for this case (Cl = 0.824).

On the other hand, new definition for constrains taken from the configuration template options are specified as shown in Figure 65. Notice that the airfoil area is set to a value which is equal or higher to the initial airfoil area but SU2 does not allow for this specific definition. Instead, higher to the initial area (previously obtained) is selected, knowing that very close decimals are accepted by SU2 and can give the result as equal).

For the pitching moment coefficient (moment around the z axis for the 2 D space in x and y coordinates) SU2 uses pitching down as positive and not the general convention of negative for pitching down used in flight mechanics (see discussion of this choice in Annex 4.4). For this reason, the moment is constrained to be less than 0.092 as this corresponds to a smaller pitch-down moment as required by the problem statement.

```
% Optimization constraint functions with pushing factors
% (affects its value, not the gradient  in the python scripts), separated by semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= ( MOMENT_Z < 0.092 ) * 0.001; ( AIRFOIL_AREA  > 0.078 ) * 0.001
```

Figure 65. Constrains definition for Case Study 3.

The boundary and freestream conditions are defined as shown in Figure 66 (notice that the angle of attack used is obtained from the wind tunnel correction from [42] but is set as a design variable due to the use of Cl driver).

```
% ------------------- COMPRESSIBLE FREE-STREAM DEFINITION -------------------%
%
% Mach number (non-dimensional, based on the free-stream values)
MACH_NUMBER= 0.734
%
% Angle of attack (degrees, only for compressible flows)
AOA= 2.79
%
% Free-stream temperature (288.15 K by default)
FREESTREAM_TEMPERATURE= 288.8
%
% Reynolds number (non-dimensional, based on the free-stream values)
REYNOLDS_NUMBER= 6.5E6
%
% Reynolds length (1 m by default)
REYNOLDS_LENGTH= 1.0
```

Figure 66. Compressible free-stream definition for Study Case 3.

3.3.7 Numerical Results

Convergence for Study Case 3 (see Figure 67) produces a much lower improvement in drag minimisation, which is an expected behaviour due to the RANS type simulation compared to the previous cases. Compared to that of Study Case 2, there are large differences drag coefficients improvement (87.03 % improvement for the Mach 0.7 case) but not so different in comparison of the baseline airfoils (as the initial Cd obtained is between Mach 0.7 and Mach 0.8 inviscid cases).

For this case, there is also some fluctuations mainly for initial design evaluations, reaching values close to the final corresponding to the optimized airfoil, but which are further improve in subsequent design evaluations that have some peaks that correspond to design steps which are erroneous that are then recovered following the initial tendency of the optimization. This peaks in drag coefficient coincide with the corresponding variations of airfoil area or moment coefficient (see Figure 68 and Figure 70). These two constrains have fluctuations in some evaluations but both converge satisfying the problem statement with 0.2 % increase of area and a reduction of the nose down pitching moment of the 10 % compared to baseline airfoil.

Lift coefficient on the other hand, is constant throughout the process (see Figure 69) and does not seem to have significant variations as those observers for Mach 0.8 for Study Case 2.

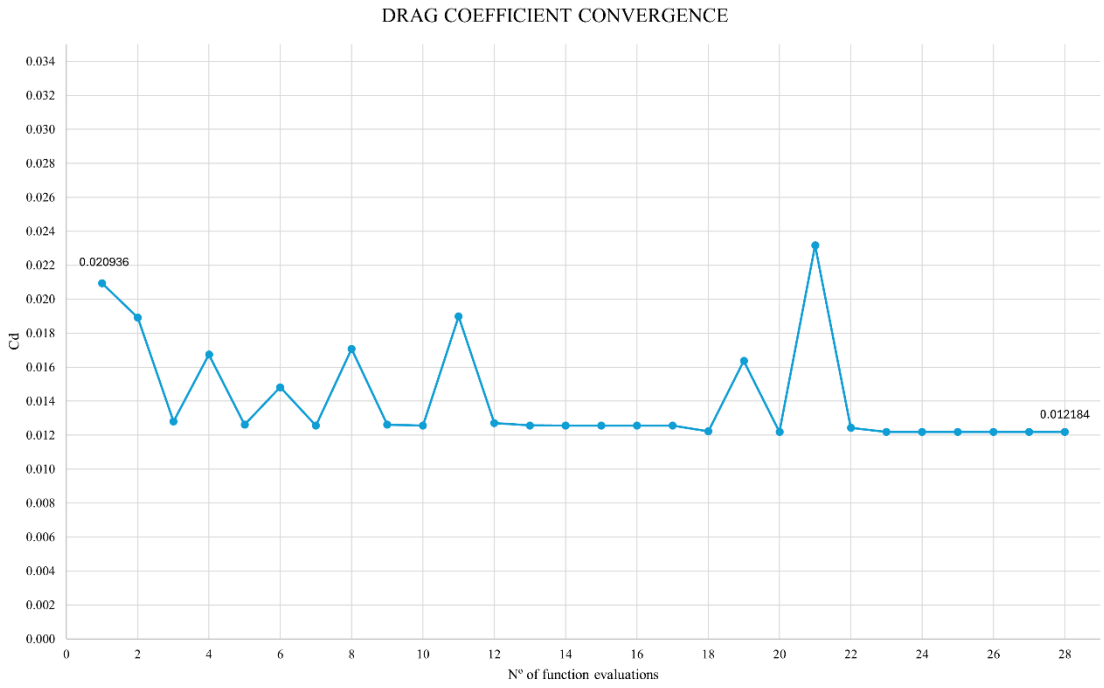

Figure 67. Drag coefficient convergence for Study Case 3.

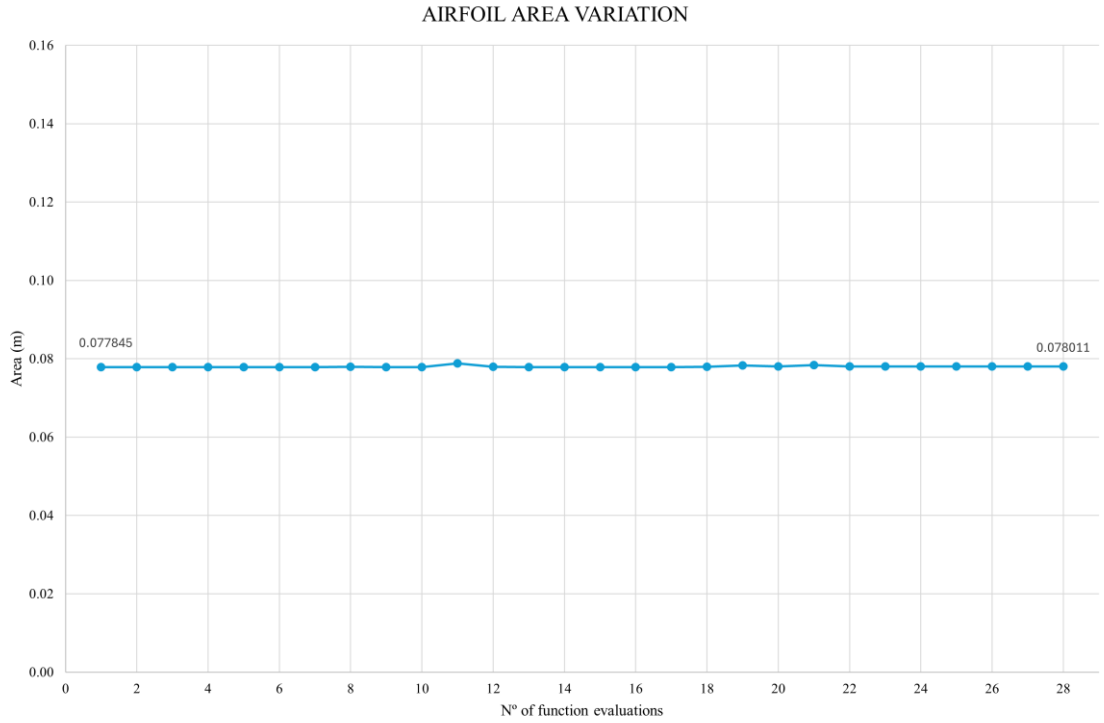

Figure 68. Airfoil area variation as a function of the number of function evaluations for Study Case 3.

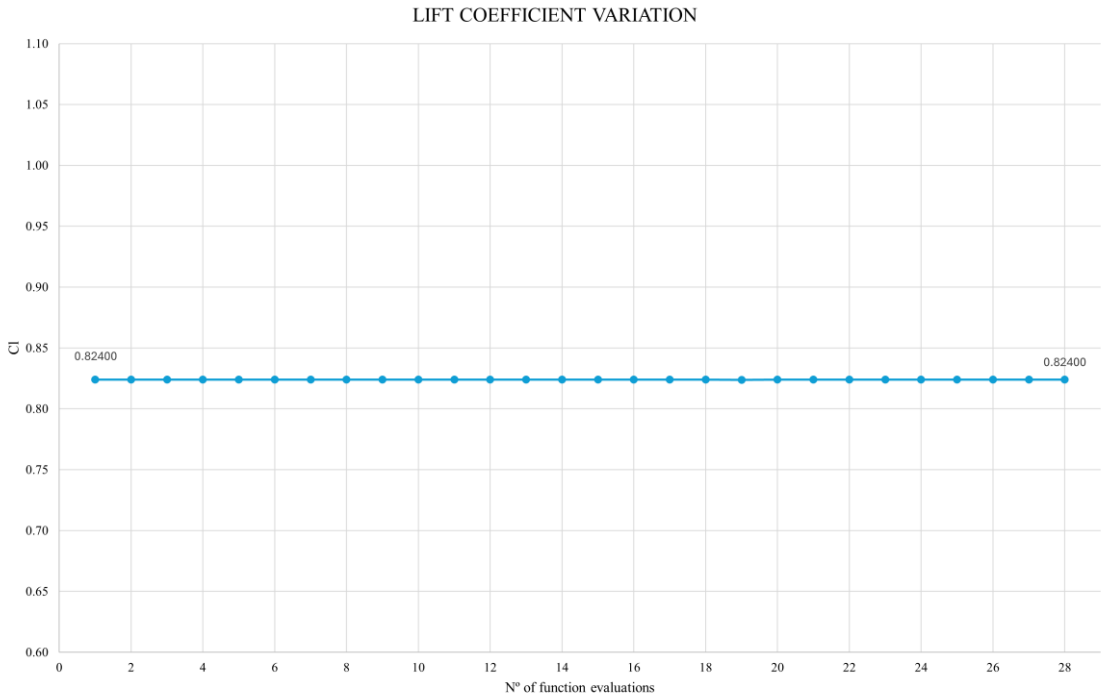

Figure 69. Lift coefficient variation as a function of number of function evaluations for Study Case 3.

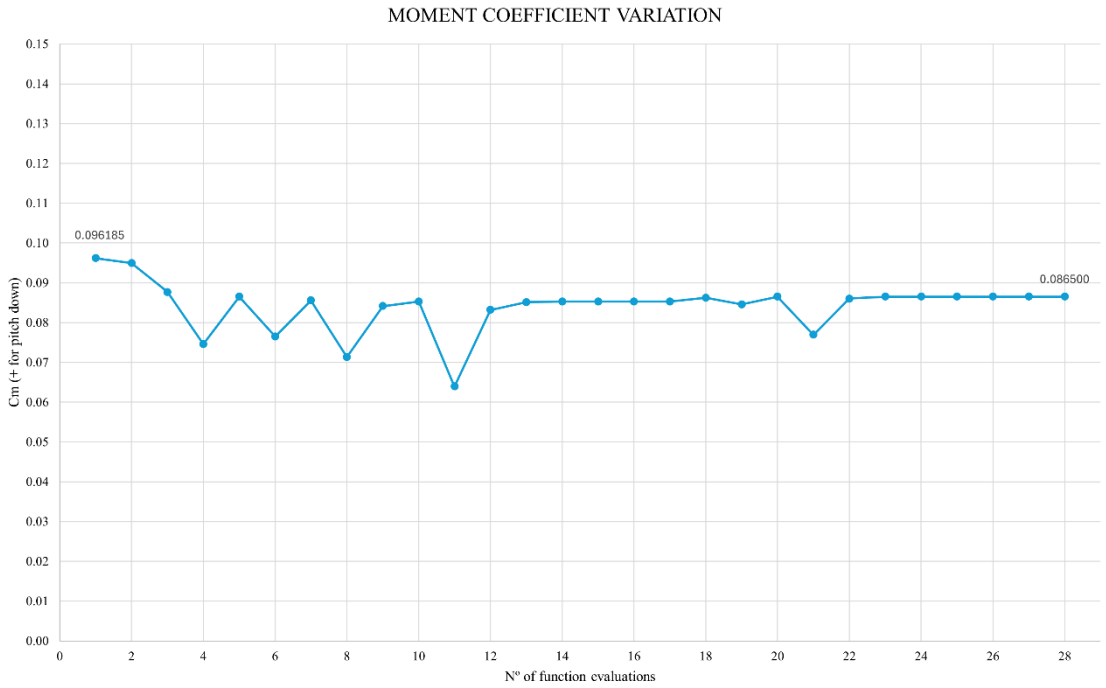

Figure 70. Moment coefficient variation as a function of number of function evaluations for Study Case 3.

3.3.8 Comparison with other research studies

Table *6* shows the comparison between this project approach towards the ADODG case 2 and other research studies, where the remarked blue line represents this project results as "Present". Drag improvement in percentage, also showing initial and optimized values can be compared, being associated to the parametrization method and optimizer type. Also, the number of design variables and number of function evaluations are shown.

The "Present" study results show very similar values as those from multiple other research studies, being inside the bounds for initial, final and reduction in percentage of drag coefficient in comparison as shown in Table 6.

Reference Article	Initial Cd	Optimized Cd	Reduction Cd (%)	Parametrization Method	Optimizer Type	Dv number	Function Evaluated
Carrier et al. [43]	0.0202	0.0111	45.0	Bézier and B-Spline	Gradient-based (SLSQP)	10	25
Poole et al [44]	0.0174	0.0090	48.3	RBF with SVD for selection of DV	Gradient-based (FSQP)	7	40
Lee et al. [45]	0.0234	0.0132	43.6	B-Splines and FFD	Gradient-based (SNOPT)	34	250
Bisson and Nadarajah [46]	0.0178	0.0107	42.7	Third-order B-spline	Gradient-based (SNOPT)	16	25
Yang and Ronch [47]	0.0241	0.0151	38.0	Hicks-Henne bump functions	Gradient-based (SNOPT)	30	-
Lourenço [26]	0.0196	0.0128	34.9	FFD	Gradient-based (SLSQP)	26	47
Present	**0.0209**	**0.0122**	**41.6**	**Hicks-Henne bump functions**	**Gradient-based (SLSQP)**	**38**	**28**

Table 6. ADODG benchmark Case 2 comparison with other research studies [27].

Compared to study [47] with the same parametrization method but different gradient-based optimizer and number of design variables, the present study represents better improvements with 3.6 % difference.

On the other hand, larger improvement (6.7 % improved) is obtained compared to research from [26] that uses also SU2 with the same optimizer type but different parametrization method and using the discrete adjoint instead of the continuous.

3.3.9 Post Processing Results

The baseline RAE 2822 Mach behaviour under Study Case 3 freestream conditions can be seen in Figure 71. This contour plot shows the effects of RANS solver applied to transonic airfoil as after the shock generated approximately at 55 % of the chord (see Figure 73) the boundary layer can be appreciated to be slightly detached as the expected behaviour explained in 2.5.2. However, this Mach Number does not seem to produce drag divergence as the separation and drag coefficient would be significantly higher.

For the optimized geometry, the upper surface has been flattened on the region between leading edge and 55 % percent of the chord, coinciding with the region of Mach increase and shock wave generation for the baseline airfoil. This geometrical correction dissipates the initial shock wave into an acceleration region but without producing any shock as the pressure coefficient of Figure 72 shows a much uniform distribution to that of the baseline RAE 2822. The dissipation of the shock wave reduces the slightly detachment of the boundary layer and thus reduces the drag coefficient as the objective of the problem.

Because of this flatten upper surface, the airfoil must recover the reduction in area by increasing the "belly" on the lower surface (see Figure 74), but not significant enough to produce a weak shock in the lower side given this freestream conditions. After this flattened airfoil region, the curvature is similar between the two geometries up to the trailing edge where for the optimized airfoil it is has lower thickness and larger downward curvature.

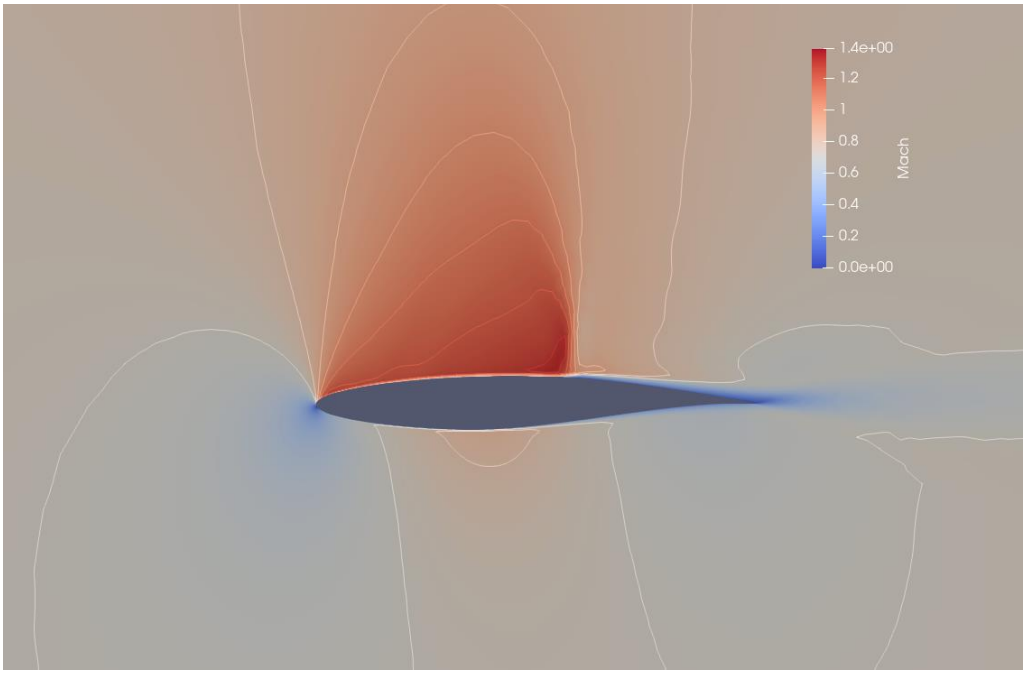

Figure 71. RAE 2822 baseline airfoil Mach contour for Study Case 3.

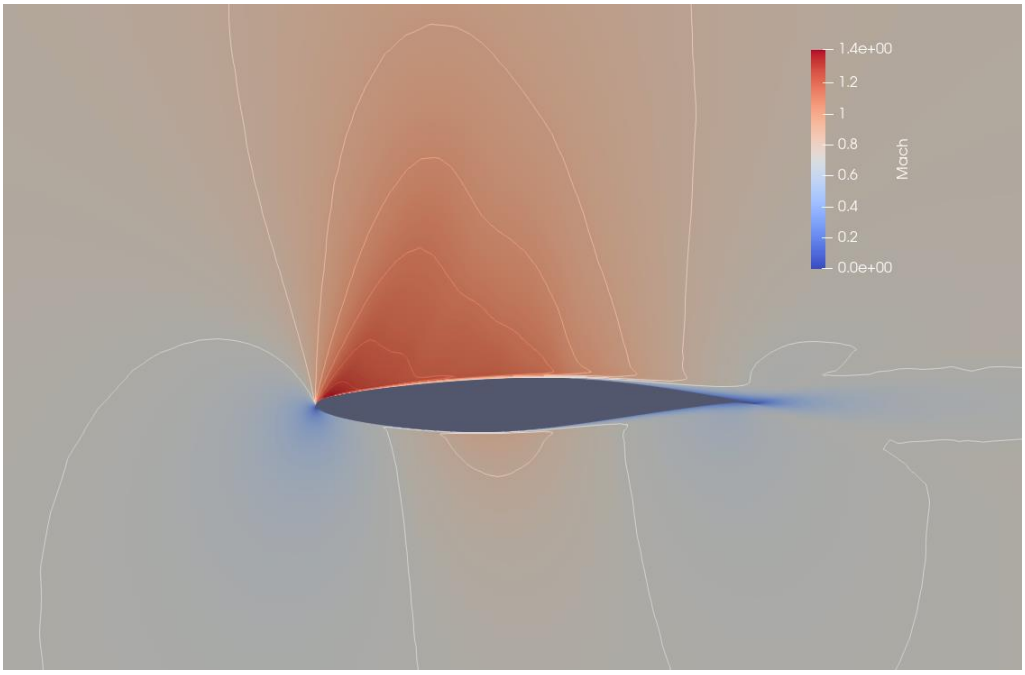

Figure 72. Optimized airfoil Mach contour for Study Case 3.

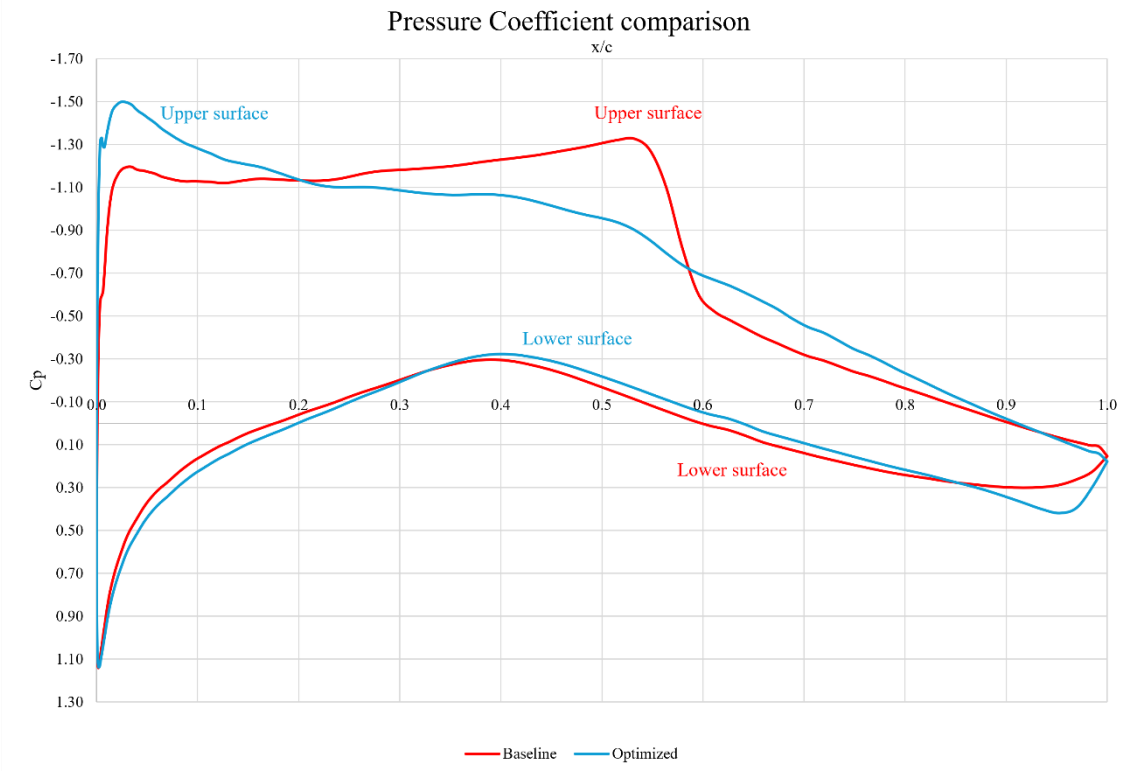

Figure 73. Pressure coefficient comparison between baseline and modified geometry for Study Case 3.

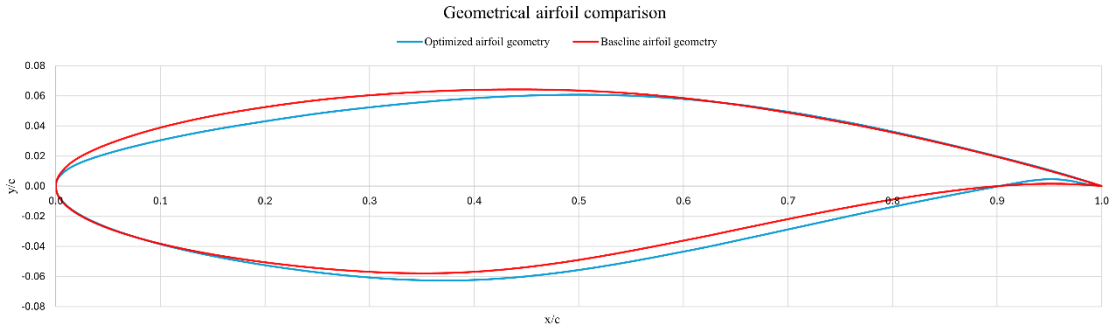

Figure 74. Geometrical comparison between baseline and optimized airfoils for Study Case 3.

Chapter 4. CONCLUSIONS AND FUTURE WORK

4.1 Concluding remarks

The development of this project has allowed to enrich and complete knowledge acquired during university studies through the introduction, study and application of an important field in the present and future of aerospace engineering.

The insights of automatic aerodynamic shape optimization are very valuable from the educational point of view as they allow to better understand flow behaviour and increase the intuition of the engineer as solutions obtained are not always trivial and obvious, allowing to think out of the box for future problems.

Trought the completion of the three Study Cases of this project, it has been possible to understand the requirements and complexity of designing supercritical airfoils to significantly increase the high-speed cruise capabilities of aircraft, which is an important consideration for the continuous evolution of aviation.

The importance of reducing the Mach divergence and critical Mach Number has been demonstrated by obtaining large improvements in airfoil performance through the reduction of drag coefficient for the different Study Cases. However, it is important to understand that not all these cases can be considered as a complete real approximation to the real world, as there are clear differences between Euler and RANS optimization.

For the Euler (inviscid) cases (Study Case 1 and Study Case 2), large improvements are obtained between baseline and optimized airfoils, due to two main reasons:

- The initial airfoil is not designed for transonic flight and produces strong shock wave formation that when dissipated or reduced for the optimal design producing extremely large minimisations in drag coefficient (up to 97.4 % reduction for the unconstrained case). Comparing Study Case 2 and 3, it can be appreciated how the optimized geometry is closer to that of the baseline airfoil, as NACA 0012 is not an airfoil designed for transonic flight regime while 2822 is a supercritical airfoil itself.

- The assumption of inviscid flow and the non-consideration of boundary layer effects produces optimization to be mainly restricted by the constraints imposed for the problem and not mainly on flow separation or friction. Constrains have been demonstrated to be

one of the main limitations to the reduction of drag as for inviscid cases, the lower the constrains the further the airfoil can be optimized up to results where the drag coefficient value is mainly given by errors in the spatial integration (as that of the unconstrained study).

On the other hand, RANS simulations provide with numerical solutions that are closer to reality, in terms of optimization percentage (41.6 % in drag coefficient reduction for Study Case 3), where shock wave dissipation have physical impact in the reduction of the detachment of the boundary layer as one of the key objectives in supercritical airfoil design.

Differences between both approaches should be considered for different design phases (Euler for preliminary approaches to the problem and RANS for detailed design and validation) as the computational cost and time is highly dependent on this selection due to the necessity of using refined meshes for RANS simulations and larger complexity in the solver equations.

Each of the Study cases has contributed to remark different characteristics of the adjoint-based optimization of supercritical airfoils.

- Trought Study Case 1, it has been possible to be familiarized with SU2 working process, understanding configuration settings that are then applied to the following Study Cases, determining that Hicks-Henne bump functions constitute a good parametrization method that was also later applied and also allowing to understand how the solutions to the adjoint equations are used to determine design sensitivities and how they are projected into SU2 workspace.

- Study Case 2 allowed to understand how the definition of constrains and freestream conditions affect the solution. Showing how the optimizer is able to find the appropriate solution even if it is non-intuitive (as the increase in thickness for Ma = 0.6 which is not a general solution for transonic flight) but that can be explained understanding that the benefit is obtained from the reduction in angle of attack and therefore reduction in upper surface flow. On the other hand, for Ma = 0.7 and Ma = 0.8, that surpass the critical Mach number, the tendency is to dissipate or reduce the intensity of the shock wave by using a flattened upper surface, compensating thickness with a bigger lower surface curvature and using an aft kick to produce lift. The highest the Mach number the larger the percentage of improvement and the higher Cd values obtained for baseline and optimized.

- Finally, Study Case 3, gives insight into closer to reality supercritical airfoil design and optimization as the flattened surfaces are less abrupt due to the need to generate lift by

producing downwash, which has to be more smoothly done instead of larger abrupt aft downwards kicks as those cases from Study Case. Also, this case showed the importance of well defining constrains according to sign convention but outlining that results obtained can still produce interesting results to analyse.

The selection of Hicks-Henne bump functions with 38 variables including the angle of attack and use of the continuous adjoint method have resulted in good optimization results in comparison to other research studies (shown in Table 6) with an improvement of 41.6 % as drag coefficient reduction, resulting in a lift to drag ratio of 67.63.

One of the key aspects of automatic design optimization that was identified is the importance of constrains, as the optimizer algorithm is able to exploit the airfoil geometry and find the weaknesses of the problem statement in order to improve optimization as shown in examples such as the reduction of lift coefficient for Study Case 1 or the increase in thickness (with a subsequent decrease in the angle of attack) for Study Case 2; Ma = 0.6. Therefore, the engineer knowledge is fundamental to set the appropriate restrictions for the problem and obtain solutions with not only aerodynamic validity but also from the engineering point of view. In this way, the multiple use of geometrical restrictions has been demonstrated to be fundamental, as thickness restrictions can allow to increase airfoil area and area restrictions can allow to reduce thickness. A cautious selection (even imposing multiple conditions) should be the correct approach towards these problems considering that some of them may not find an appropriate solution.

On the other hand, it has been identified that convergence is one of the main possible improvements in this project, as mainly in every drag convergence large peaks result from erroneous evaluations have appeared and also larger fluctuations are observed, which can be an effect of errors in the evaluation of sensitivities due to the use of the continuous adjoint, (as the main disadvantage to discrete adjoint is the lower consistency with gradients). Also, better convergence could be improved by the use of higher level of refinement in meshes, considering the associated exchange in computational cost.

The approach towards an open-source philosophy (reflected in the selection of software as SU2 and ParaView) can be considered as a fundamental approach that should be followed in the scientific community as this allows to and accelerate technical development of new innovative designs. However, this philosophy has also associated disadvantages, as this can lead to fractured code development with lower coordination, as a problem that was faced during due to the lack of information in some crucial aspects (as that of the Cm sign convention) for the most recent versions of SU2 increasing the complexity of this project.

Therefore, through the completion of this project, a successful application of the continuous adjoint has been demonstrated for the design and optimization of transonic airfoils in an effective computational manner satisfying the goals of the project.

4.2 Future development

This project represents a small contribution to aircraft aerodynamic design and optimization, that could be largely extended and completed with the following fields of study:

- **Experimental validation**: CFD is a great tool for aircraft design, however it only acts as a simulation of real-world conditions in a computational way. For this reason, substantiation of the work performed in this project should be done, validating with experimental studies (such as wind tunnel testing) the airfoil geometries obtained determining if these results can be applied to the real world and ensuring the CFD model is accurate to reality.

- **Wing design**: as a simplified case, 2D aerodynamic shape optimization is important but the real flight conditions have additional considerations due to the addition of another dimensions. A three-dimensional extension of an airfoil is represented by wings, to which adjoint based optimization can also be applied for spanwise considerations. This would allow to consider design variables such as chord or twist angle in order to optimize three-dimensional pressure distributions to reduce also wing tip vortices and therefore achieve much more aerodynamic efficient flight.

- **Application to novel aircraft concepts**: adjoint-based optimization can also be applied to new aircraft concepts taking advantage of large number of variables to obtain aerodynamic efficient geometries for blended wing bodies (as that of Figure 3) or to facilitate the implementation of truss braced wings reducing interference drag and optimizing truss geometries.

- **Multi-objective optimization**: apart from drag minimization, other objective functions could be introduced into the problem formulation to improve other aspects of aircraft design such as maximizing lift to drag ratio, reducing noise or even coupling CFD with topological optimization considering both, structures and aerodynamics in the search of optimal solutions.

- **Surrogate-Based Optimization**: this methodology could be investigated for additional increase in the efficiency of the optimization process as surrogate modelling techniques can help to find faster optimal designs. This novel optimization technique could be coupled with adjoint-based sub-optimizations for much more efficient automatic design processes [48].

APPENDICES

4.3 Study Case 1 Appendix

Figure 75, Figure 76 and Figure 77 show the flow properties discontinuities as a consequences of a strong shock wave in the upper surface and a weak shock in the lower surface for NACA 0012 and freestream condition of Ma = 0.8.

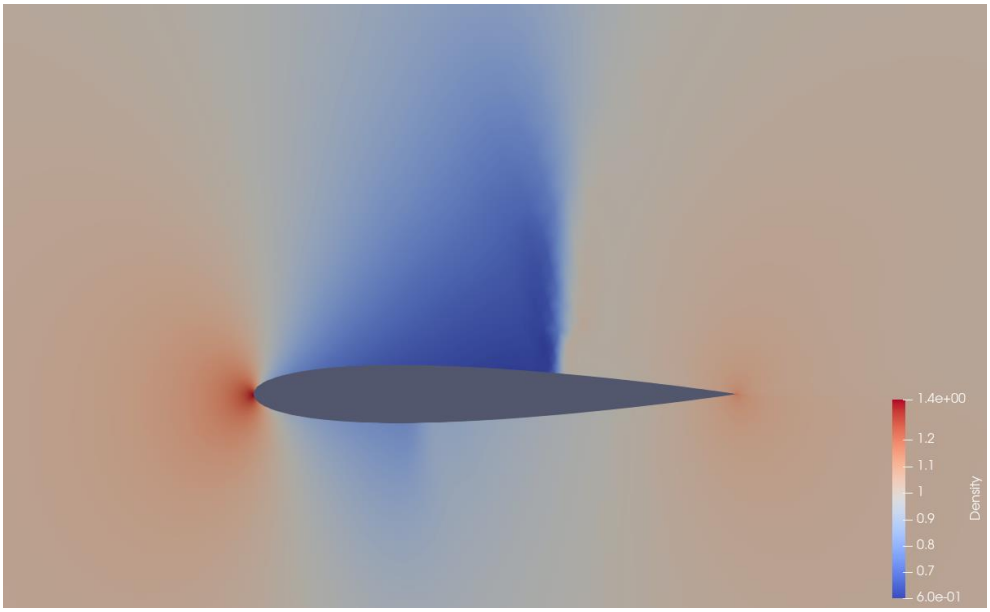

Figure 75. Density contour for NACA 0012 (baseline airfoil) in Case study 1.

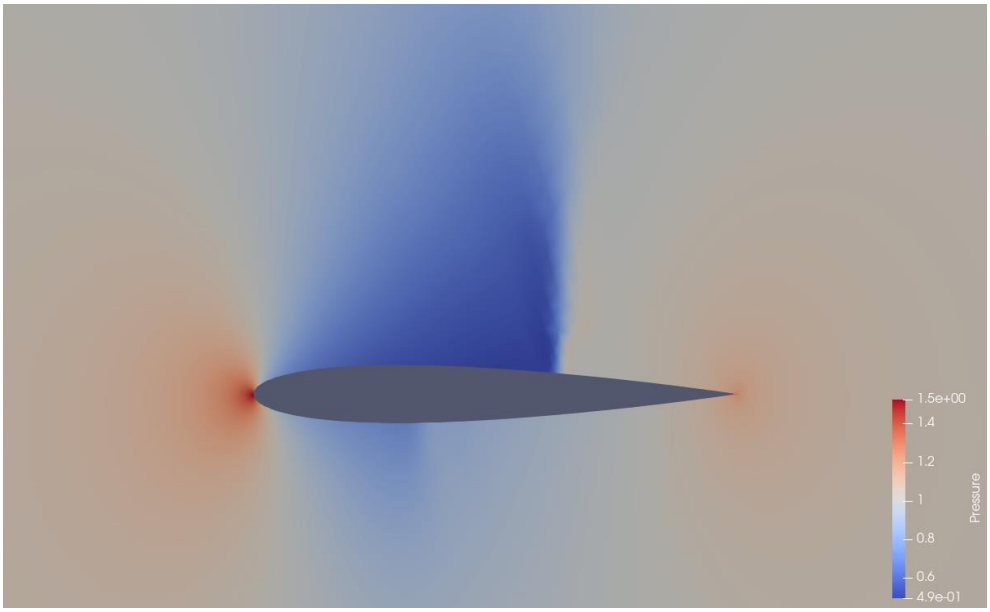

Figure 76. Pressure contour for NACA 0012 (baseline airfoil) in Case study 1.

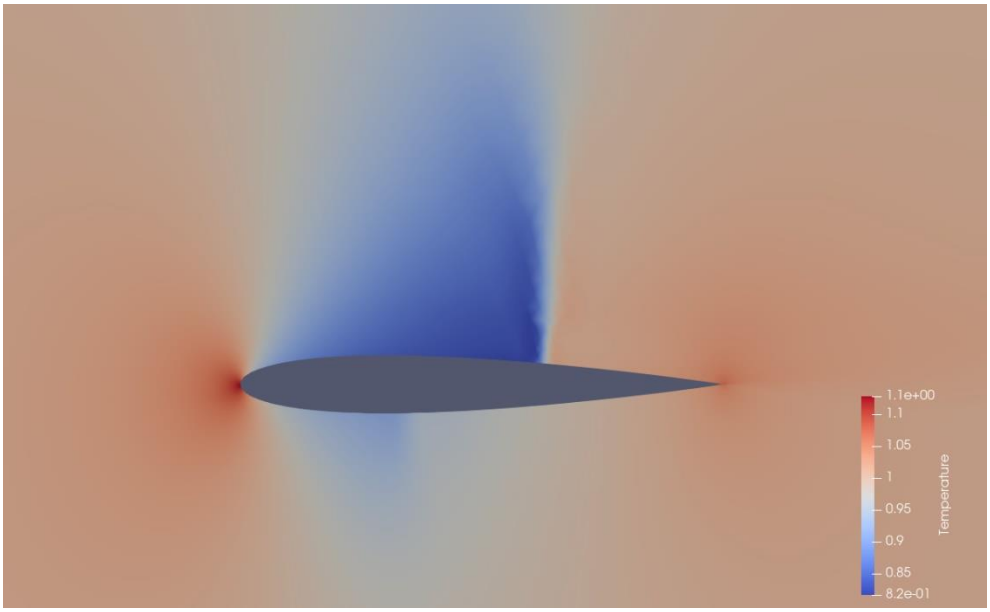

Figure 77. Temperature contour for NACA 0012 in Case study 1.

4.4 Study Case 3 Appendix

Initially, the moment coefficient was defined as shown in Figure 78 as stated in the optimization problem formulation (as SU2 sign convention was not considered).

```
% Optimization constraint functions with pushing factors
% (affects its value, not the gradient  in the python scripts), separated by semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= ( MOMENT_Z > -0.092 ) * 0.001; ( AIRFOIL_AREA  > 0.078 ) * 0.001
```

Figure 78. Constrains definition for Study Case 3 Annex.

Even though this definition was misleading due to the sign convention used in SU2 (as explained in 3.2.6), interesting optimization results were achieved, being similar (numerically) as those previously presented for Study Case 3 but different in post processing visualization. This annex has the intention to compare results from both moment coefficient definitions and act as a base to avoid committing this error for future research studies.

4.4.1 Numerical Results

Convergence for this way to constrain the moment coefficient is significantly different as the number of evaluations is much larger. For this case, there is also some fluctuations mainly for initial design evaluations, which are then reduced but reaching to a high peak due to an erroneous evaluation (evaluation nº 24) that shows a large geometry deformation and subsequent extreme flow separation (see right top corner of Figure 79) which is then corrected. This peak coincides

with the corresponding peak of airfoil area for this case which seems larger variation than for the correct Study Case 3 (see Figure 68), but which is stabilised with 0.12 % increase to the initial area which can be considered a negligible difference. Also, there is a peak of moment coefficient at this point which is then corrected.

Lift coefficient on the other hand, is constant throughout the process (see Figure 69) and does not seem to have significant variations as for Study Case 3.

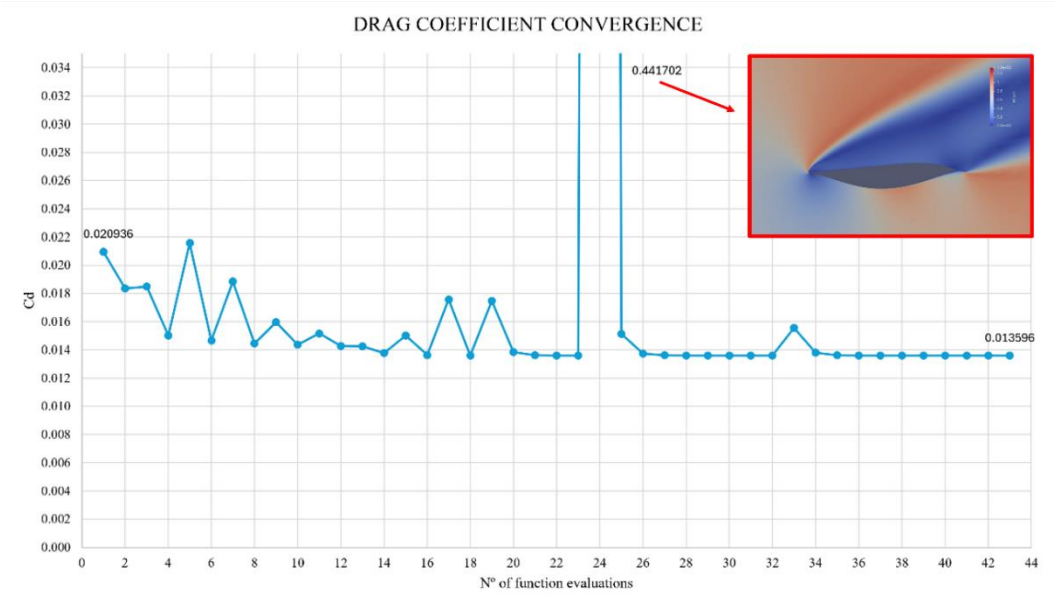

Figure 79. Drag coefficient convergence for Case study 3.

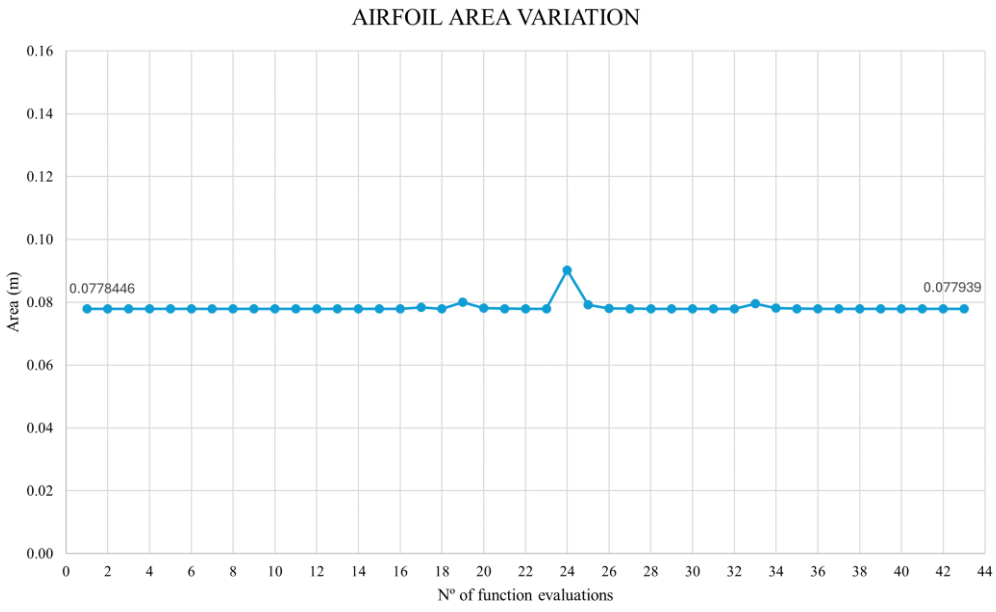

Figure 80. Airfoil area variation as a function of the number of function evaluations for Case Study 3.

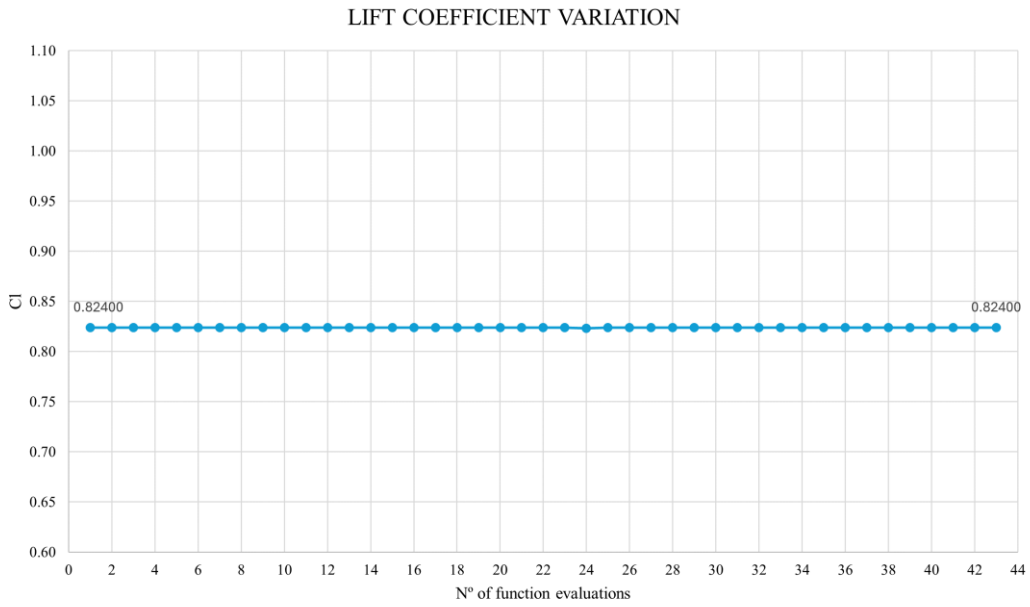

Figure 81. Lift coefficient variation as a function of number of function evaluations for Case Study 3.

The erroneous Cm sign assumption comes from Figure 82, as positive values with similar magnitude are obtained to that of -0.092 (constrains are generally imposed knowing that the obtained values would probably be close to the value of the constrain, in this case -0.092). The constrain is satisfied but the behaviour of the airfoil would be very different to the baseline if the sign convention was using negative values for pitching down.

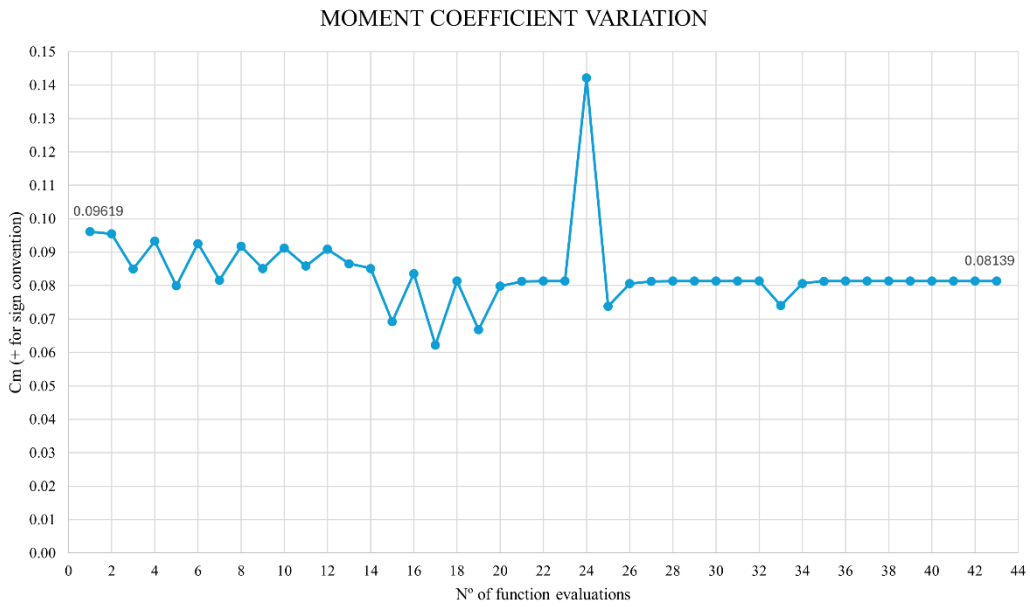

Figure 82. Moment coefficient variation as a function of number of function evaluations for Case Study 3.

As previously mentioned, even though the constrain is erroneous defined, the optimization is similar but with a lower improvement in drag reduction compared the correct Cm definition as seen from Table 7. This table compares the Cd optimization between correct Cm constrain and the erroneous one.

Reference Article	Initial Cd	Optimized Cd	Reduction Cd (%)	Parametrization Method	Optimizer Type	Dv number	Function Evaluated
Present (correct Cm constrain)	**0.0209**	**0.0122**	**41.6**	**Hicks-Henne bump functions**	**Gradient-based (SLSQP)**	**38**	**28**
Present (erroneous Cm constrain)	**0.0209**	**0.0136**	**34.9**	**Hicks-Henne bump functions**	**Gradient-based (SLSQP)**	**38**	**43**

Table 7. Study Case 3 drag coefficient results comparison between correct Cm constrain (green) and erroneous Cm constrain (red).

4.4.2 Post Processing Results

The baseline RAE 2822 Mach behaviour is identical (Figure 83) to that previously defined in Study Case 3 as there aren't any variations in the freestream definition.

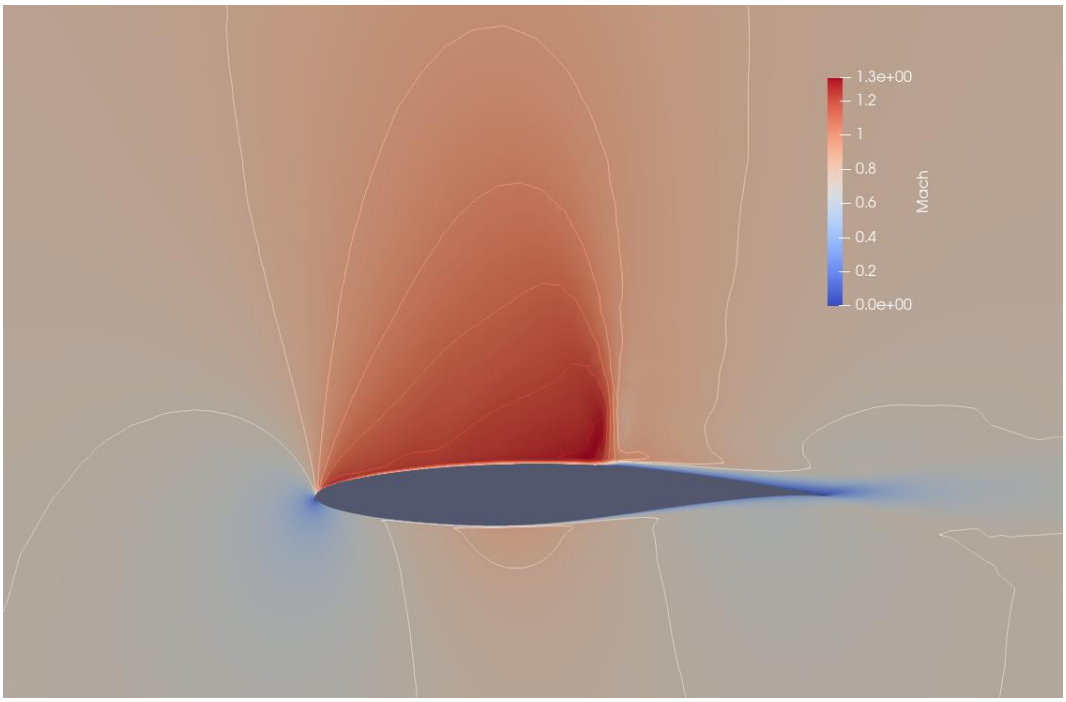

Figure 83. RAE 2822 baseline airfoil Mach contour for Study Case 3 (wrong Cm definition).

For the optimized airfoil, the geometry is slightly different than for that of the correct definition of Cm. The upper surface has been flattened on the region between leading edge and 55 % percent of the chord in a similar way (see Figure 85), coinciding with the region of Mach increase and

shock wave generation for the baseline airfoil. This geometrical correction dissipates the initial shock wave into a first weak shock wave producing slightly flow separation which is then reattached, and another supersonic region is developed leading to a secondary weak shock (see Figure 84). However, this second shock is displaced aft (now produced at 60% of the chord) and its strength is lowered, reducing the boundary layer separation, and therefore reducing drag.

After this flattened airfoil section, the curvature is similar between the baseline and optimized geometries, as by the formulation of the problem both must have the same lift capability. On the other hand, smoother trailing edge can be observed from Figure 86 in comparison to the optimized airfoil with correct Cm constrain from Figure 74.

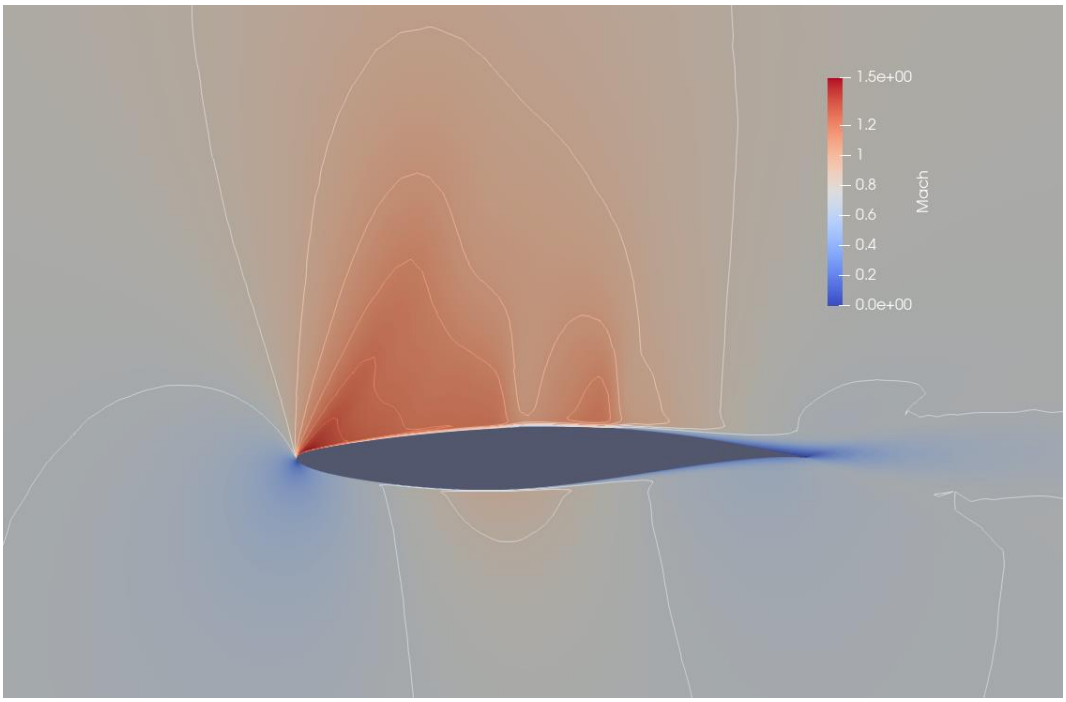

Figure 84. Optimized airfoil Mach contour for Study Case 3 (wrong Cm definition).

Even though this pressure distribution and double high Mach formation can seem contra intuitive, there are multiple supercritical airfoil studies that show similar behaviours:

- This solution was also observed in [47], obtaining different Mach contours for two consecutive function evaluations, and leading to the conclusion of a non-unique solution of the optimised airfoil (see Figure 87).
- Also, NASA studies on supercritical airfoils showed similar results for analytical design on 14-percent-thick supercritical airfoils at M = 0.73 and Cl= 0.7 and $Re = 30x10^6$ as seen in Figure 88, indicating a secondary velocity peak producing a secondary shock wave formation that could produce drag penalties if not well-managed [49].

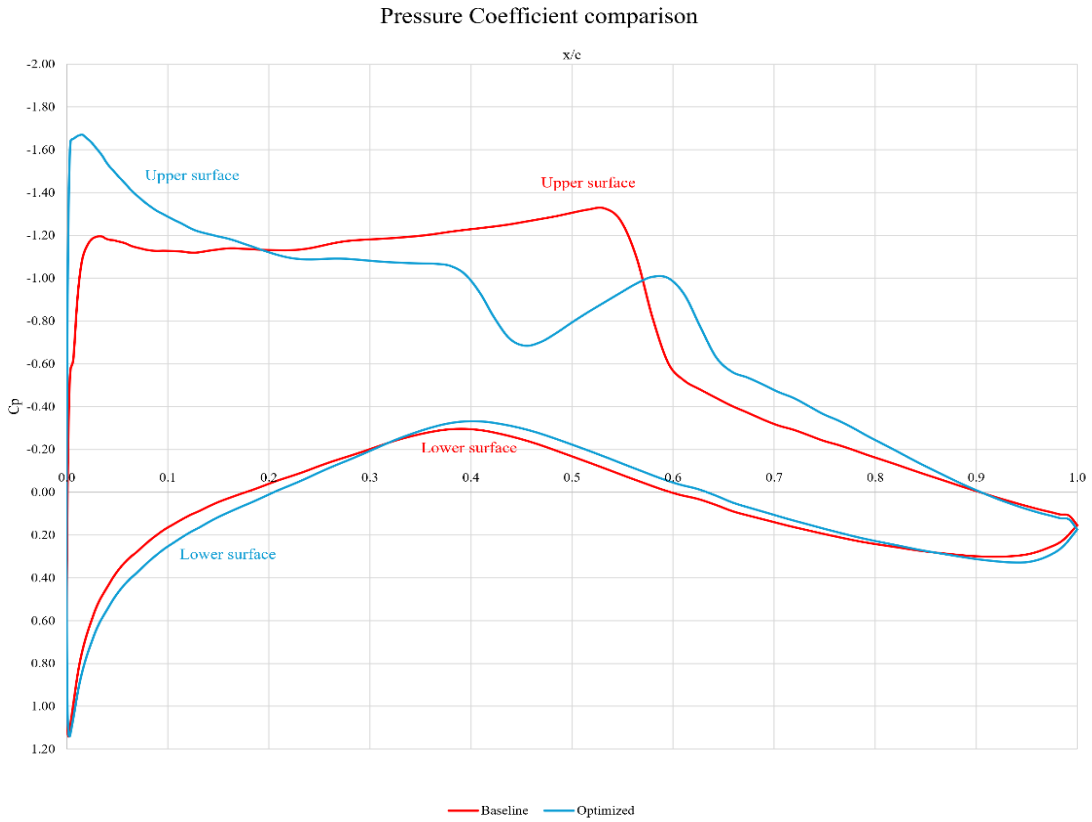

Figure 85. Pressure coefficient comparison between baseline and modified geometry for Study Case 3 (wrong Cm definition).

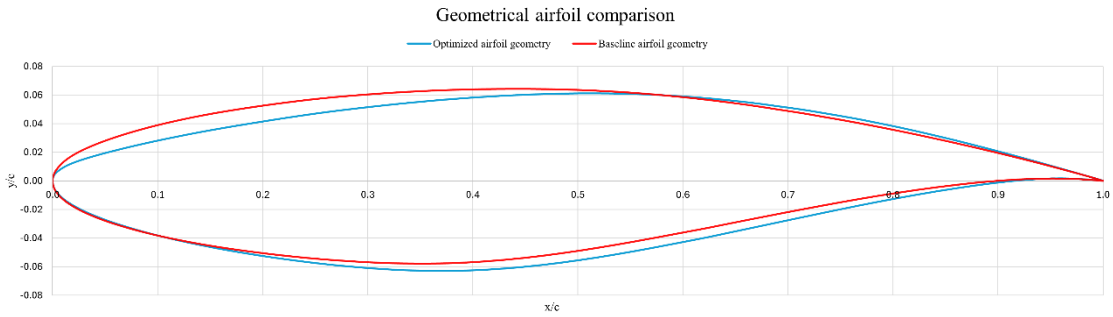

Figure 86. Geometrical comparison between baseline and optimized airfoils for Study Case 3 (wrong Cm definition).

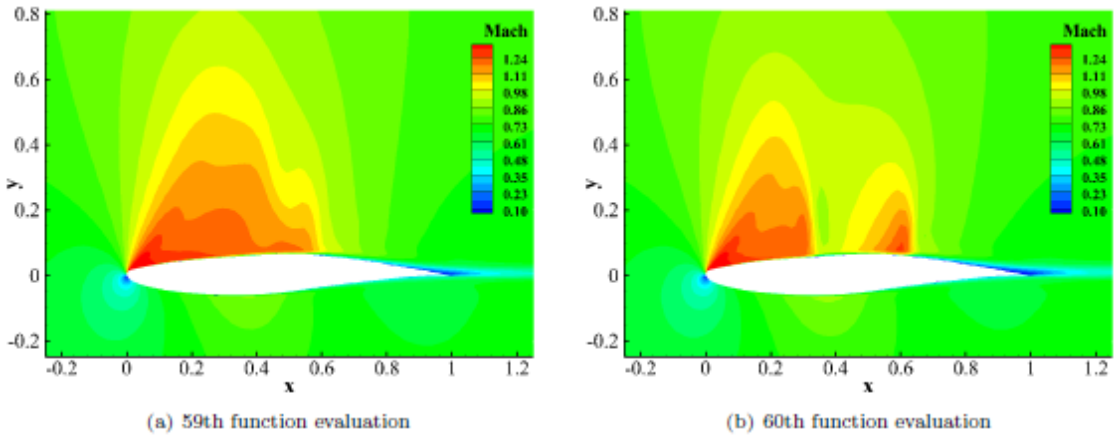

Figure 87. Comparison between two evaluations for ADODG case 2 obtaining non-unique flow solutions [47].

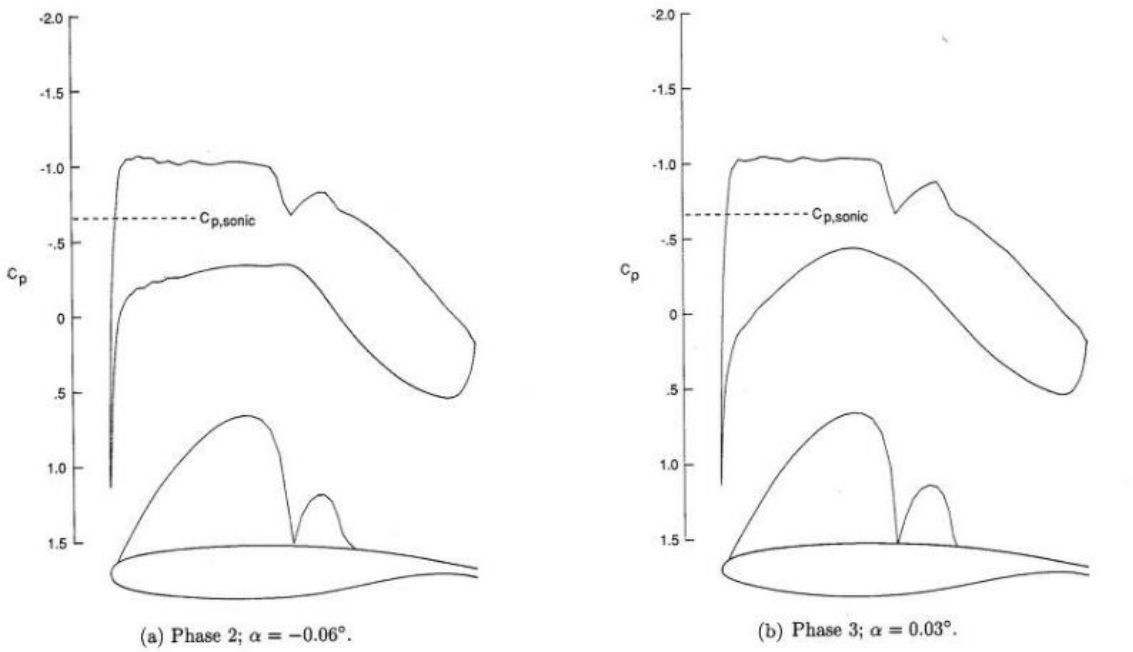

Figure 88. Analytical design pressure distributions for 14-percent-thick phase 2 and 3 supercritical airfoils (M = 0.73; Cl= 0.7; Re = 30E6 [49].

BIBLIOGRAPHY

[1] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*, 1st ed. Cambridge University Press, 2021. doi: 10.1017/9781108980647.

[2] 'Our Commitment to Fly Net Zero by 2050'. Accessed: Feb. 15, 2024. [Online]. Available: https://www.iata.org/en/programs/environment/flynetzero/

[3] 'Forecasted evolution of air transport apssenger traffic', IATA. Accessed: Feb. 15, 2024. [Online]. Available: https://www.iata.org/contentassets/1d7b998cda0a46a1ab31d4ee3cce5eaf/chart-forecasted-evolution-of-air-transport-pax-traffic.pdf

[4] R. McCausland, 'Net zero 2050: new aircraft', Jun. 2024.

[5] C. Gelzer, 'X-48B Blended Wing Body', NASA. Accessed: Mar. 15, 2024. [Online]. Available: https://www.nasa.gov/wp-content/uploads/2016/08/253983main_ED06-0201-2_full.jpg

[6] J. Hemmerdinger, 'PICTURE: Boeing unveils trussed wing capable of jet speeds', Flight Global. Accessed: Mar. 15, 2024. [Online]. Available: https://www.flightglobal.com/airframers/picture-boeing-unveils-trussed-wing-capable-of-jet-speeds/130914.article

[7] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, Second Edition. in AIAA Education Series. Washington, DC: Education Series, 1992.

[8] I. H. Abbot and A. E. Von Doenhoff, *Theory of wing sections. Including a summary of Airfoil Data*. New York, NY: Dover Plublications, INC, 1959.

[9] J. D. Anderson, *Introduction to flight*, 3rd ed. in McGraw-Hill series in aeronautical and aerospace engineering. New York: McGraw-Hill, 1989.

[10] J. D. Anderson, *Fundamentals of aerodynamics*, Sixth edition. in McGraw-Hill series in aeronautical and aerospace engineering. New York, NY: McGraw-Hill Education, 2017.

[11] O. Cleynen, *English: Selected airfoils in nature and various vehicles, with their approximate chord length indicated.* 2011. Accessed: Mar. 20, 2024. [Online]. Available: https://commons.wikimedia.org/wiki/File:Examples_of_Airfoils.svg

[12] O. Cleynen, *English: Vocabulary terms used in the aeronautical domain. Notes:* 2011. Accessed: Mar. 20, 2024. [Online]. Available: https://commons.wikimedia.org/wiki/File:Wing_profile_nomenclature.svg

[13] Y. A. Çengel and J. M. Cimbala, *Fluid mechanics: fundamentals and applications*, Third edition. New York: McGraw Hill, 2014.

[14] V. by ר. ה. McLean Original by J. Doug, *English: A diagram of the force vectors acting on a lifting airfoil.* 2019. Accessed: Mar. 21, 2024. [Online]. Available: https://commons.wikimedia.org/wiki/File:Airfoil_lift_and_drag.svg

[15] E. L. Houghton, Ed., *Aerodynamics for engineering students*, Sixth Edition. Amsterdam Heidelberg: Elsevier, 2013.

[16] W. Yamazaki, K. Matsushima, and K. Nakahashi, 'Unstructured Mesh Drag Prediction Based on Drag Decomposition', *Eur. Conf. Comput. Fluid Dyn.*, Jan. 2006.

[17] Cadence CFD, 'Understanding the Aerodynamic Characteristics of Supersonic Airfoils'. Accessed: Mar. 14, 2024. [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2022-understanding-the-aerodynamic-characteristics-of-supersonic-airfoils

[18] I. Gibert Martínez, F. Afonso, S. Rodrigues, and F. Lau, 'A Sequential Approach for Aerodynamic Shape Optimization with Topology Optimization of Airfoils', *Math. Comput. Appl.*, vol. 26, no. 2, p. 34, Apr. 2021, doi: 10.3390/mca26020034.

[19] J. D. Anderson, *Computational Fluid Dynamics. The basics with applications*. in McGraw-Hill series in aeronautical and aerospace engineering. New York, NY: McGraw Hill, 1995.

[20] J.-E. Lombard, 'Introduction to Structured Grid Generation for Aeronautics', *Swiss Institute of Technology Lausanne*, Sep. 2011.

[21] D. S. Pachpute, 'Flow Visualization Techniques in Experiment and CFD · CFD Flow Engineering', CFD Flow Engineering. Accessed: Apr. 13, 2024. [Online]. Available: https://cfdflowengineering.com/flow-visualization-techniques-in-experiment-and-cfd/

[22] J. R. R. A. Martins, 'Aerodynamic design optimization: Challenges and perspectives', *Comput. Fluids*, vol. 239, p. 105391, May 2022, doi: 10.1016/j.compfluid.2022.105391.

[23] A. Jameson, 'Automatic Design of Transonic Airfoils to Reduce Reduce the Shock Induced Pressure Drag'.

[24] A. Jameson, 'Aerodynamic Shape Optimization Using the Adjoint Method', Feb. 2003.

[25] S. Nadarajah and A. Jameson, 'A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization', in *38th Aerospace Sciences Meeting and Exhibit*,

Reno,NV,U.S.A.: American Institute of Aeronautics and Astronautics, Jan. 2000. doi: 10.2514/6.2000-667.

[26] J. P. B. Lourenço, 'Supersonic and Transonic Adjoint-based Optimization of Airfoils', *Thesis Aerosp. Eng.*, Nov. 2018.

[27] R. M. Hicks and P. A. Henne, 'Wing Design by Numerical Optimization', *J. Aircr.*, vol. 15, no. 7, pp. 407–412, 1978, doi: 10.2514/3.58379.

[28] C. Sabater and S. Görtz, 'Gradient-Based Aerodynamic Robust Optimization Using the Adjoint Method and Gaussian Processes', in *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, vol. 55, A. Gaspar-Cunha, J. Periaux, K. C. Giannakoglou, N. R. Gauger, D. Quagliarella, and D. Greiner, Eds., in Computational Methods in Applied Sciences, vol. 55. , Cham: Springer International Publishing, 2021, pp. 211–226. doi: 10.1007/978-3-030-57422-2_14.

[29] E. Sharifi, A. Akhavan Taheri Borojeni, and M. H. Hekmat, 'Investigation of the Adjoint Method in Aerodynamic Optimization Using Various Shape Parameterization Techniques', *J. Braz. Soc. Mech. Sci. Eng.*, vol. 32, pp. 176–186, Jun. 2010, doi: 10.1590/S1678-58782010000200012.

[30] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, 'SU2: An Open-Source Suite for Multiphysics Simulation and Design', *AIAA J.*, vol. 54, no. 3, pp. 829–846, Mar. 2016, doi: 10.2514/1.J053813.

[31] P. R. Spalart and S. R. Allmaras, 'A one-equation turbulence model for aerodynamic flows', *Rech. Aéroespatiale*, vol. 1, 1994.

[32] U. Ayachit, 'Updated for ParaView version 5.6', *Community Ed.*, Jan. 2019.

[33] 'Unconstrained shape design of a transonic inviscid airfoil at a cte. AoA'. Accessed: Apr. 15, 2024. [Online]. Available: https://su2code.github.io/tutorials/Inviscid_2D_Unconstrained_NACA0012/

[34] 'Tutorials/design/Inviscid_2D_Unconstrained_NACA0012 at master · su2code/Tutorials', GitHub. Accessed: Apr. 25, 2024. [Online]. Available: https://github.com/su2code/Tutorials/tree/master/design/Inviscid_2D_Unconstrained_NACA0012

[35] A. Hujeirat, 'A method for relaxing the Courant-Friedrich-Levy condition in time-explicit schemes', *Astron. Astrophys.*, vol. 430, no. 3, pp. 893–903, Feb. 2005, doi: 10.1051/0004-6361:20041530.

[36] A. Jameson, 'The Origins and Further Development of the Jameson-Schmidt-Turkel (JST) Scheme'.

[37] J. Jeon, Y. Ren, and G. Zha, 'Toward Ultra-High Cruise Lift Coefficient Using Flapped Coflow Jet Airfoil', in *AIAA SCITECH 2023 Forum*, National Harbor, MD & Online: American Institute of Aeronautics and Astronautics, Jan. 2023. doi: 10.2514/6.2023-1008.

[38] S. Ciornei, 'Mach number, relative thickness, sweep and lift coefficient of the wing', *Hambg. Univ. Appl. Sci.*, May 2005.

[39] 'Computational Aerodynamics Group - ADODG'. Accessed: May 12, 2024. [Online]. Available: https://sites.google.com/view/mcgill-computational-aerogroup/adodg

[40] S. Nadarajah, 'Drag Minimization of the RAE 2822 in Transonic Viscous Flow', Jul. 2013.

[41] 'Tutorials/design/Turbulent_2D_Constrained_RAE2822/mesh_RAE2822_turb.su2 at master · su2code/Tutorials', GitHub. Accessed: May 15, 2024. [Online]. Available: https://github.com/su2code/Tutorials/blob/master/design/Turbulent_2D_Constrained_RAE2822/mesh_RAE2822_turb.su2

[42] 'Tutorials/design/Turbulent_2D_Constrained_RAE2822/turb_SA_RAE2822.cfg at master · su2code/Tutorials', GitHub. Accessed: May 15, 2024. [Online]. Available: https://github.com/su2code/Tutorials/blob/master/design/Turbulent_2D_Constrained_RAE2822/turb_SA_RAE2822.cfg

[43] G. Carrier *et al.*, 'Gradient-Based Aerodynamic Optimization with the elsA Software', presented at the 52nd AIAA Aerospace Sciences Meeting - AIAA Science and Technology Forum and Exposition, SciTech 2014, Jan. 2014. doi: 10.2514/6.2014-0568.

[44] D. J. Poole, C. B. Allen, and T. Rendall, 'Control Point-Based Aerodynamic Shape Optimization Applied to AIAA ADODG Test Cases', in *53rd AIAA Aerospace Sciences Meeting*, in AIAA SciTech Forum. , American Institute of Aeronautics and Astronautics, 2015. doi: 10.2514/6.2015-1947.

[45] C. Lee, D. Koo, K. Telidetzki, H. Buckley, H. Gagnon, and D. W. Zingg, 'Aerodynamic Shape Optimization of Benchmark Problems Using Jetstream', in *53rd AIAA Aerospace Sciences Meeting*, in AIAA SciTech Forum. , American Institute of Aeronautics and Astronautics, 2015. doi: 10.2514/6.2015-0262.

[46] S. Nadarajah, 'Adjoint-Based Aerodynamic Optimization of Benchmark Problems', in *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2015. doi: 10.2514/6.2015-1948.

[47] G. Yang and A. Da Ronch, 'Aerodynamic Shape Optimisation of Benchmark Problems Using SU2', Jan. 2018. doi: 10.2514/6.2018-0412.

[48] Z.-H. Han and K.-S. Zhang, 'Surrogate-Based Optimization', in *Real-World Applications of Genetic Algorithms*, O. Roeva, Ed., InTech, 2012. doi: 10.5772/36125.

[49] NASA Technical Reports Server (NTRS), *NASA Technical Reports Server (NTRS) 19900007394: NASA supercritical airfoils: A matrix of family-related airfoils*. 1990. Accessed: May. 25, 2024. [Online]. Available: http://archive.org/details/NASA_NTRS_Archive_19900007394