INSTITUTO SUPERIOR TÉCNICO

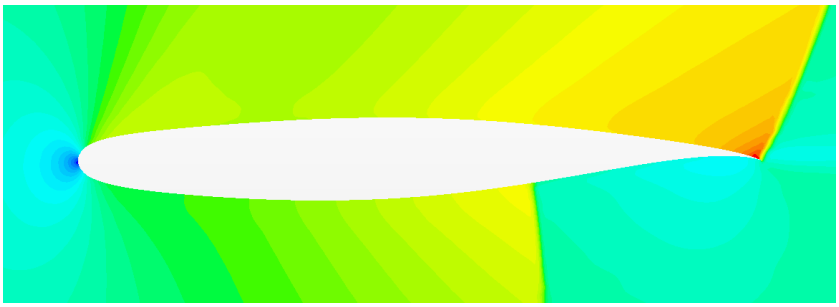
UNIVERSIDADE DE LISBOA

2nd Semester 2023/2024

Aerodynamics III

Transonic Flight Optimization with IGP Parameterization for Supercritical Airfoils and Shock Control Bump

---



GROUP NUMBER 4

Tiago Santos, 87290

João Gaspar, 96930

Guilherme Teixeira, 100210

Afonso Andrade, 108267

April 14, 2024

Contents

1 Introduction

In the pursuit of reducing flight times and enhancing efficiency in commercial aviation, the exploration of supersonic and transonic speeds has been pivotal. While supersonic flight, exemplified by the Concorde, offers remarkable speed, it also brings along significant challenges such as noise pollution along with environmental and economic concerns due to high fuel consumption.

Transonic flight presents an intriguing alternative. However, it introduces its own set of complexities, particularly regarding the airflow around an aircraft's airfoil. In these conditions, the interaction between subsonic and supersonic flows poses challenges that will degrade the airfoil's performance, namely a considerable increase in drag, demanding innovative solutions.

In an effort to mitigate the degradation of performance in this regime several solutions are often implemented such as increasing wing sweep angle, changing the wing's thickness and profile or even introducing active flow control techniques, such as boundary layer suction or blowing. This report will focus on improving the geometry of the 2D profile of the airfoil.

To improve its performance in these conditions, airfoil optimization constitutes an important role in the development sector. As a result of the competition between aircraft manufacturers, there is little available information about transonic optimised airfoils. This paper addresses an optimization of an airfoil for transonic flight at mach number $M = 0.87$ and angle of attack $\alpha = 1.5°$. The baseline airfoil is the NACA 4412. The goal is to maximize its lift to drag ratio, an important parameter that measures the efficiency of an aircraft´s wing in generating lift relative to the amount of drag produced.

2 Aerodynamic Analysis of NACA 4412

Before stating the optimization process, it is important to understand the flow proprieties around the NACA 4412 airfoil at the problem's conditions of $M = 0.87$ and $\alpha = 1.5°$. In this section an CFD analysis of the baseline airfoil is conducted in order to understande the optimization path.

2.1 Geometry

The considered airfoil is a NACA 4-digit airfoil, where the first digit describes the maximum camber as a percentage of the chord, the second defines decribes distance of maximum camber from the leading edge in tenths of chord. The last two digits define maximum thickness as a percent of the chord. For the NACA 4412 the maximum thickness is 12% at 30% chord, and the maximum camber is 4% at 40% chord

2.2 Simulation

To evaluate the airfoil's performance in the transonic regime, a CFD simulation was conducted considering the flow parameters in table 1.

Mach number	0.87
AoA	1.5º
Temperature	223.15
Pressure	2.64363E+4 Pa
Density	4.12707E-1

Table 1: Simulation flow parameters

The characteristic of transonic flow, the air is observed in Figure 1 to accelerate around the geometry of airfoil to supersonic speed and originating phenomena such as shockwaves and expansion fans. A relatively strong shockwave is formed in the upper surface close to the trailing edge, causing wave drag. A weaker shock is seen in the lower surface at about 80% of the chord.

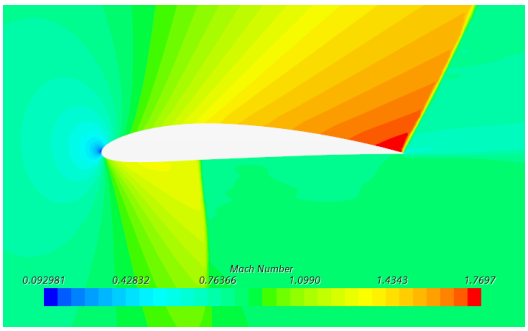


Figure 1: Mach field

The induced pressure field seen in Figure 2 is a result of the strength and location of the shocks and expansion waves. The curvature of the upper surface accelerates the flow beyond mach 1 and reduces the fluid's pressure, increasing overall lift. On the lower surface a smaller expansion fan also reduces pressure somewhat (detrimental to lift production), however the shock that slows down the air to subsonic speeds produces a advantageous pressure field when it comes to lift.

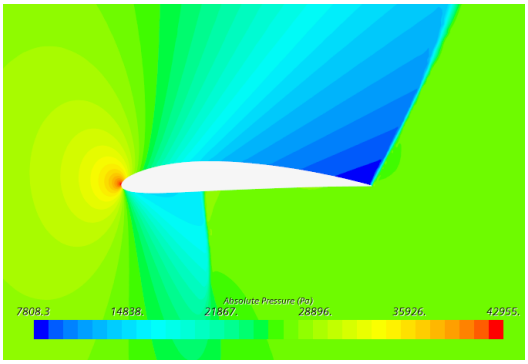


Figure 2: Pressure field

The following aerodynamic parameters were obtained as a result of the CFD simulation:

Lift	9373.2 N
Drag	2186.5 N
Lift to drag ratio	4.2869

Table 2: Aerodynamic performance of NACA 4412

3 Problem Statement

As seen in the previous section, the airfoil under analisys underperforms at the desired speed and angle of attack. This is due to transonic phenomena around its profile such as shock-waves and expansion waves that proved detrimental to lift production due to adverse pressure distributions, and incremental to drag mainly due to wave drag.

In order to improve its operatational performance at high subsonic mach numbers, the airfoil in question will be subject to a geometry modification, with the ultimate goal of increasing its lift to drag ratio. Its geometry will be parameterized so that it can be defined as a function of a discrete number of variables.

To explore the domain of possible solutions an optimization algorithm will be implemented to modify the parameters that define the geometry of the airfoil.

4 Parameterization

Over the years, numerous strategies have been developed to achieve optimal aerodynamic performance for both airfoils and wings, as well as fuselages. The work conducted by S.N. Skinner and H. Zare-Behtash [1] serves as a foundational baseline for implementation in the aircraft optimization domain. This work not only presents various optimization approaches but also underscores a fundamental mathematical requirement: parameterization.

A parameterization is a mathematical equation based model which describes the airfoil. This is a key aspect in the present strategy as will be discussed later. The choosen parameterization "Improved Geometric Parameterization (IGP)" was developed by Lu et al. [2] and is presented in next subsection.

Also in section 4.1, two modification to the airfoil are presented in order to improve the aerodynamic performance. Firstly, IGP parameterization does not ables to change the trailing edge height, so 4.2 shows how the parameterization is change to inclue it. Another change to the airfoil, 4.3 presents an approach done by Tian et al. [3], where a bump is added to a supercritical airfoil to reduce the shock wave.

4.1 Improved Geometric Parameterization (IGP)

The IGP method consists in defining a camber and thickness distribution that are used to build the airfoil surfaces. The camber is expressed based on the Bézier polynomial as shown on equation 1 and the thickness is expressed by the polynomial basis function represented on equation 2. With this method it is possible to parameterise an airfoil with only 8 input variables, described in table 3, that will conduct to 10 output variables, shown in table 4. The low number of control parameters will further increase the computational efficiency in the process of optimization. For instance a Parsec model has 12 variables [4].

c_1, c_2	coefficients of camber-line-abscissa parameter equation
c_3, c_4	coefficients of camber-line-ordinate parameter equation
X_T	chordwise location of maximum thickness
T	maximum thickness
ρ_0	leading edge radius
β_{TE}	trailing edge boat-tail angle

Table 3: IGP method inputs

Camber distribution equations:

$$\begin{cases} x_C = 3c_1k(1-k)^2 + 3c_2(1-k)k^2 + k^3 \\ y_C = 3c_3k(1-k)^2 + 3c_4(1-k)k^2 \end{cases} \quad (1)$$

C	maximum camber
k_c	k value on the location of maximum camber
X_C	chordwise location of maximum camber
α_{TE}	angle between camber line and chord line on trailing edge
b_{X_C}	camber line curvature on the location of maximum camber
t_1,..., t_5	coefficients of thickness equation

Table 4: IGP method outputs

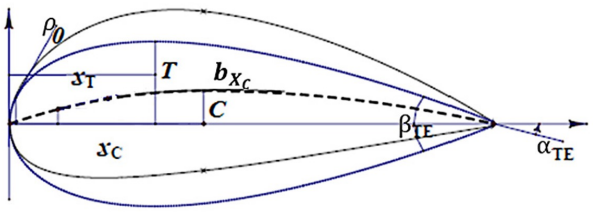


Figure 3: Geometric parameters

Thickness distribution equation:

$$t = t_1 x^{0.5} + t_2 x + t_3 x^2 + t_4 x^3 + t_5 x^4 \tag{2}$$

To compute the camber and thickness distribution it is necessary to solve their backstepping equation set.
Camber backstepping equation set:

$$\begin{cases} 3c_3(3{k_c}^2 - 4k_c + 1) + 3c_4(-3{k_c}^2 + 2k_c) = 0 \\ 3c_3 k_c (1-k_c)^2 + 3c_4(1-k_c){k_c}^2 = C \\ 3c_1 k_c (1-k_c)^2 + 3c_2(1-k_c){k_c}^2 + {k_c}^3 = C \\ \frac{c_4}{1-c_2} = \tan\alpha_{TE} \\ \left| \frac{6c_3(3k_c-2)+6c_4(-3k_c+2)}{(6c_1(3k_c-2)+6c_2(-3k_c+2)+3{k_c}^2)^2} \right| = b_{X_C} \end{cases} \tag{3}$$

Thickness backsteping equation set:

$$\begin{cases} t_1 {X_T}^{0.5} + t_2 X_T + t_3 {X_T}^2 + t_4 {X_T}^3 + t_5 {X_T}^4 = T \\ 0.5 t_1 {X_T}^{-0.5} + t_2 + 2t_3 X_T + 3t_4 {X_T}^2 + 4t_5 {X_T}^3 = 0 \\ 0.25t_1 + 0.5t_2 + t_3 + 1.5t_4 + 2t_5 = -\tan\frac{\overline{\beta_{TE}}}{2} \\ t_1 = \sqrt{2\overline{\rho_0}} \\ t_1 + t_2 + t_3 + t_4 + t_5 = 0 \end{cases} \tag{4}$$

Where, $\overline{\rho_0}$ and $\overline{\beta_{TE}}$ are the dimensionless quantities of ρ_0 and β_{TE}.

$$\overline{\rho_0} = \frac{{\rho_0}^2}{\frac{T}{X_T}} \tag{5}$$

$$\overline{\beta_{TE}} = \frac{\beta_{TE}}{\arctan\frac{T}{1-X_T}} \tag{6}$$

These equations sets represent the relation between curve function parameters and the geometric parameters shown in figure 3. Their solution makes possible to define the upper and lower surfaces of the airfoil. The correspondent equations are equations 7 and 8 respectively.

Upper surface:

$$\begin{cases} x_U = x_C \\ y_U = y_C + \frac{1}{2}t(x_C) \end{cases} \tag{7}$$

Lower surface:

$$\begin{cases} x_U = x_C \\ y_U = y_C - \frac{1}{2}t(x_C) \end{cases} \tag{8}$$

For instance, varying k from 0 to 1 in a step of 0.01, the camber line is described in 101 points, with the x_C values the thickness value is also known in those 101 points. Using these two values the upper and lower surfaces are defined by 101 points each. In later sections, there is described the k values chosen for the optimization.

When compared to other airfoil parameterizations such as the PARSEC, the IGP method reveals some advantages. The separation between camber and thickness is the main difference from the past efforts, this method only requires 4 parameters to construct the camber distribution while other need requires at least 10 parameters as a result of coupling of camber and thickness. Other advantage is the fact that the parameterization variables can be easily related to the geometric parameters commonly used in the general aerodynamic theory. This allows changing the airfoil characteristics in order to accomplish some specific goal following general aerodynamic theory. A good example of that is to increase the camber in order to reduce the zero-lift angle of attack. Besides this, the use of less control parameters reduces the compute of airfoil optimization.

4.2 Trailing edge modification

From the previous methodology, a parameterization is retrieved allowing manipulation of the airfoil. However, another important parameter modification lacking in the methodology is the height of the trailing edge. To accomplish the modification of this feature, a series of calculations are performed on

the airfoil to have a smooth change of the parameter.

The first step is to find the maximum thickness of the profile. The points onwards of the maximum thickness will be subject to modifications. This is accomplished by defining a percentage (p) that the trailing edge will raise or lower in relation to the maximum height of the foil. After, at each location, it is decided how much the airfoil changes based on the previous percentage. The upper surface is modified by the percentage definition for each step and for the lower surface, depending on the position a squared percentage is added depending on the position to allow for a smoother transition as previously mentioned, inputting more change to the end of the profile than from the maximum thickness end. Below are the equations used in the code. In Figure 4 there are presented the comparison of an original airfoil and one modified in the trailing edge.

$$\Delta y_{up}(x) = (max[y] - y_{up}(x))(1-p) \tag{9}$$

$$y_{up2}(x) = max[y] - \Delta y_{up}(x) \tag{10}$$

$$\Delta y_{low}(x) = (max[y] - y_{low}(x))p \tag{11}$$

$$y_{low2}(x) = y_{low}(x) + \frac{(x - x_{max[y]})^2}{(x_{trail} - x_{max[y]})^2}\Delta y_{low}(x) \tag{12}$$

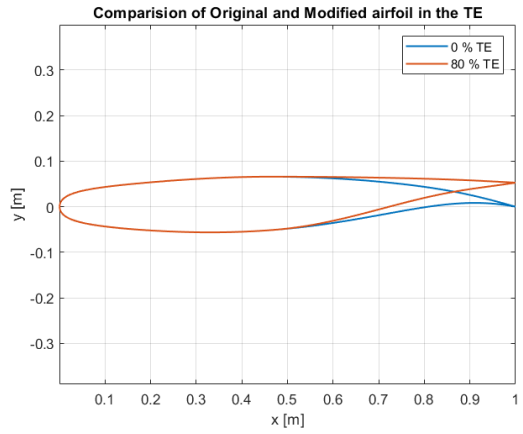


Figure 4: Comparision of Original and Modified airfoil in the TE.

4.3 Bump Control

The work developed by Tian et al [3] discusses the use of Shock Control Bumps (SCBs) to manage shock waves and improve aerodynamic performance. SCBs induce a λ-shock structure by splitting a shock wave into front and rear legs, reducing pressure loss compared to a single shock wave. However, improper SCB positioning or shape can lead to undesirable effects. If positioned too upstream or low , figure 5a, SCBs can cause re-acceleration, forming secondary shock systems and increasing drag. Conversely, positioning them too downstream or high can induce re-acceleration 5b, leading to boundary-layer separation and increased pressure losses. Optimal SCB placement, figure 5c, maximizes the λ-shock structure while minimizing additional pressure losses and secondary shock systems.

Research has shown significant drag reductions in configurations like modified wings with Hybrid Laminar Flow control and SCB techniques. Studies have focused on supercritical airfoils due to their widespread use in modern large aircraft. The paper provides insights into the shock-controlling mechanisms of SCBs and offers guidance for optimizing supercritical airfoil and wing designs under various conditions.

To implement the bump, [3] uses Hicks-Henne shape functions to describe the bump distribution as shown in figure 6.

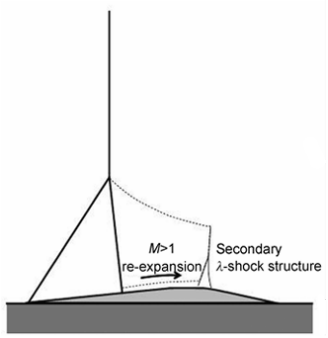
The primary parameters characterizing SCB includes the following parameters: the length l_B (where the chord length of the airfoil c is normalized to 1), the position of the crest $\frac{c_B}{l_B}$, the height h_B, and the distance between the crest of the bump and the location of the shock wave. To implement the bump distribution f_{B}

$$\begin{aligned} f_{\mathrm{B}}(x) &= h_{\mathrm{B}} H(x_{\mathrm{B}}), \\ H(x_{\mathrm{B}}) &= \sin^4(\pi x_{\mathrm{B}}^m), \\ m &= \ln(0.5)/\ln(c_{\mathrm{B}}/l_{\mathrm{B}}). \end{aligned} \tag{13}$$

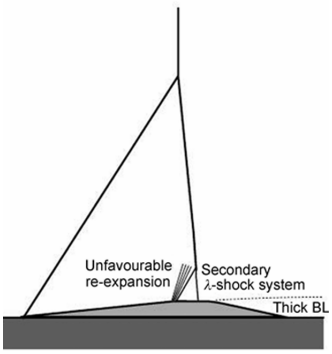
Hence, [3] improves Hicks- Henne shape functions and applied them to SCB. The improved expression is given by 14

$$H(x_{\mathrm{B}}) = \begin{cases} \sin^4(\pi x_{\mathrm{B}}^m), & m = \frac{\ln(0.5)}{\ln(c_{\mathrm{B}}/l_{\mathrm{B}})}, \quad c_{\mathrm{B}}/l_{\mathrm{B}} \geqslant 0.5, \\ \sin^4(\pi(1-x_{\mathrm{B}})^m), & m = \frac{\ln(0.5)}{\ln(1-c_{\mathrm{B}}/l_{\mathrm{B}})}, \quad c_{\mathrm{B}}/l_{\mathrm{B}} \leqslant 0.5. \end{cases} \tag{14}$$

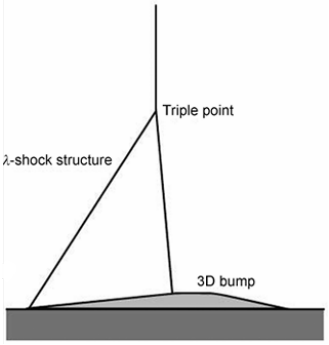
Also the following ranges range for the parameters are usually considered in the literature:



(a) Front shock postion



(b) Back shock position



(c) Optimum shock postion

Figure 5: Shock waves considering different bump ostions

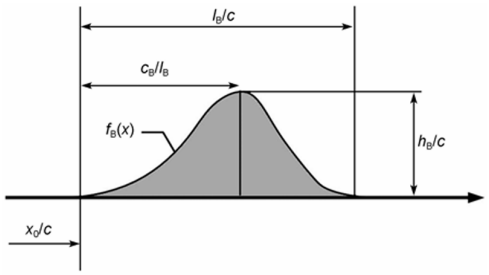


Figure 6: Definition of bump geometry and basic Hicks-Henne shape functions.

$$\begin{aligned} 0.1 &\leqslant l_{\mathrm{B}} \leqslant 0.4, \\ 0 &\leqslant c_{\mathrm{B}}/l_{\mathrm{B}} \leqslant 1, \\ 0 &\leqslant h_{\mathrm{B}} \leqslant 0.007, \end{aligned} \tag{15}$$

5 Optimization

The objective of this work, as stated is to optimize the NACA 4412 airfoil. The method chosen to optimize the airfoil is a genetic algorithm due to the easiness of implementation by using MATLAB built-in functions, and the possibility of arriving at local minimums not depend on the initial condition, as is the case for a gradient descent optimization. Also, because NACA airfoils are designed for subsonic mach numbers, for a supercritical airfoil, a simple change of parameters based on brute force was considered to not suffice for this work. A series of given constraints are set in the problem statement, and this will serve as a basis for airfoil acceptance and parameterization bounding. The constraints are presented below.

$$0 <= x <= 1 \tag{16}$$

$$-0.1 <= y <= 0.1 \tag{17}$$

$$0.11988 < tmax/c < 0.12012 \tag{18}$$

$$\alpha_{TE} < 30^{\circ} \tag{19}$$

So, T will be set as Equation 18, and a linear constraint to α_{TE} is applied, complying with Equation 19. This equation is retrieved from the methodologies section, noting $\beta = \alpha_{TE}$ and leading to Equation 20, retreived from the definition of the trailing edge angle equations of Equations 3 on the parameterization section. This equation is used as an upper bound by setting α_{TE} as 30°.

$$c_4 - c_2 tan\alpha_{TE} <= tan\alpha_{TE} \tag{20}$$

Also, genetic algorithms, figure 7 mimic natural selection to solve problems by iteratively evolving a population of potential solutions. Individuals in the population are scored based on their fitness, akin to an organism's effectiveness in resource competition. The fittest individuals have the chance to reproduce through crossover, combining genetic information to create offspring. Less fit individuals are less likely to be selected, simulating natural selection. The process involves selection, crossover (recombination), and mutation to introduce diversity. These mechanisms efficiently explore solution spaces, especially in complex, nonlinear problems where traditional methods struggle. They're commonly used in parameter optimization, feature selection, and machine learning model tuning.

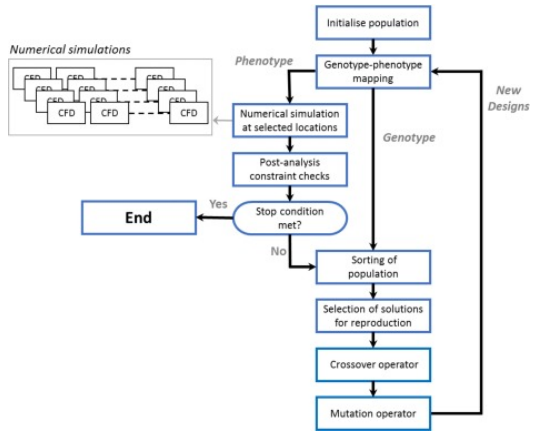


Figure 7: Schematic diagram of typical genetic algorithm structure for aerodynamic optimisation, from [1]

6 Simulation Setup

In order to accomplish the main goal of the paper, an simulation setup is necessary to implement the strategy previously mentioned. This section explains in detail all the simulations.

6.1 *STAR-CCM+* configuration

To validate any given airfoil geometry, a computational fluid dynamics (CFD) simulation was executed in *STAR-CCM+*, a comercial CFD software. The configuration of this software was achieved by loading a java file containing information of the airfoil geometry and CFD parameters. With the geometry being previously determined as a new individual of a generation from the genetic algorithm. Others parameters of the software were kept the same for every simulation, and are as follows:

6.1.1 Load airfoil coordinates

However a macro for running a *STAR-CCM+* simulation is given, it was found usefull to load the coordinates in the macro instead of direct define variables with the x and y coordinates.

So, the java function in section A was implemented to add this feature to the main macro.

6.1.2 Flow Physics and solver

For the flow setting, the inviscid model was considered, with parameters such as ideal gas behavior, temperature and pressure gradients, and two-dimensional analysis, all carefully defined within the *STAR-CCM+* environment. Additionally, solvers like steady-state, partitioning, and coupled implicit schemes were selected to ensure accurate representation of the flow physics involved.

6.1.3 Airfoil discretization

In order to generate the coordinates considering all the airfoil modifications some considerations are important to make when discretize de airfoil domain. for generating the air foil coordinates, it werre defined 4 regions in the airfoil discretization with different number of points. This will increase the sample foil points aiming to improve the geometry.

The first two regions are defined for the leading edge. The leading edge has the biggest gradient of curvature, then it was stated that more points are needed. Also, in the literature is a common technique to a well defined airfoil. The second region of the discretization technique, helps to a have a smoother transition from the leading edge to the upper and lower surfaces.

The 3 region can also be seen in figure 8, and is the bump discretization. This region needs to be refined has it is a very tiny size when compared with the foil. So to have a good definition as seen in figure 6, the number of point in this region was increased in the upper surface.

Finally, the last region considered was the trailing edge region, once some considerations and modifications are made to the trailing edge. Other regions where considered with a reasonable number of points but no special needs are considered.

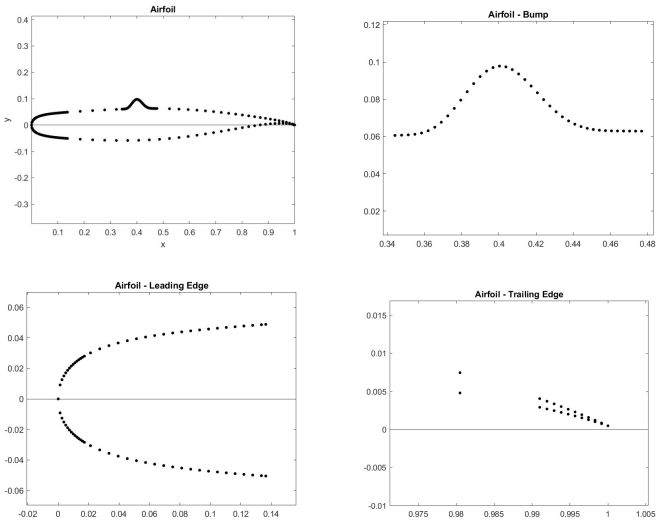


Figure 8: Foil discretization by sections

Defining the airfoil discretization is firstly done considering the k values as an input to the IGP parameterization function. anda define the four regions: 2 for the leading edge, one for

upper and lower surfaces and one for the trailing edge. For each of these regions a different $x-size$ was set along with different number of points. For implementing the bump, it is firstly need the baseline airfoil from IGP. After a refinement in the bump region is done and then the bump distribution is computed for this region as presented in 4.3.

6.1.4 Convergence criteria

Once considering a Genetic Algorithm, a good convergence criteria is necessary to a fast and accurate simulation. So, it was observed that considering $^{L}/_{D}$ as an asymptotic limit, was a good choice. Then, the macro was change to consider the asymptotic limit as 0.001 for the past 10 iterations results.

6.2 GA configuration

Genetic optimization algorithms is used to optimize the design of the airfoil by iteratively improving its shape to achieve improved aerodynamic efficiency. Since maximizing lift to drag ratio is the goal, then will D/L will be evaluate as the objective function, since the GA will minimize this value.

In order to run the genetic algorithm, a toolbox function for a generic run. The *Global Optimization Toolbox* provides a **ga** function in which it is simple to implement a good algorithm with simple configurations. For the flow solver, the *STAR-CCM+* configuration was presented in section 6.1

Further info of the **ga** cand be found in `https://www.mathworks.com/help/gads/ga.html`

The following sub-sections present variations in the **ga** function inputs for running different ways of generating the airfoil. Three airfoils were generated and named Airfoil 1, Airfoil 2, and Airfoil 3. Airfoil 1 was generated using a genetic algorithm, in Airfoil 2 a mutation was implemented, and in Airfoil 3 a bump was added to Airfoil 1 or 2, depending on the highest $^{L}/_{D}$ value. The simulations were run by utilizing 8 cores.

6.2.1 Airfoil optimization through Genetic algorithm - Airfoil 1

The analysis starts by defining the number of variables. The variables are defined as 9, from IGP and tail parameters. Linear constrains are defined as Equation 20 in the form of A and b matrices presented in the next equations.

$$A = [0,0,0,0,0,tand(30),0,1,0] \tag{21}$$

$$b = [tand(30)] \tag{22}$$

Also, the upper and lower boundaries are of the parameters can be defined. Present in [2] the IGP parameters present a feasible space for typical airfoil parameters. These parameters are utilized in this work and present at Table 5. Also, this table presents the allowable range of p, the tail parameter. Parameters are presented as lower bound lb, and upper bound ub. The maximum thickness was defined following Equation 18.

	lb	**ub**
T	0.11988	0.12012
Xt	0.2002	0.4813
p0	0.1750	1.4944
beta	0.1452	4.8724
c1	0.010	0.960
c2	0.020	0.970
c3	-0.074	0.247
c4	-0.102	0.206
p	0	0.3

Table 5: Upper and lower bounds used in the optimization.

After the definition of constrains, the options were set. The population size was defined as 10 to allow for genetic variability and the maximum generations as 1000 to allow the algorithm to run until convergence.

The genetic algorithm has an objective function that starts by discretizing the k vector, in accordance with section 6.1.3. After this step, a function related to a parameterization function is called that will compute the geometry based on the IGP and p variables. This function outputs the x and y coordinates of the airfoil and these values are saved in an Excel. This Excel in turn will be read by the macro after the *STAR-CCM+* solver is called on Matlab. The solver outputs the L and D values. If lift is negative, the value outputted by the objective function is set as 10 in order to discard this value. If lift is positive, the output of the objective function is set as $^{D}/_{L}$.

6.2.2 Airfoil optimization through Genetic algorithm with mutation - Airfoil 2

The objective of this analysis is to observe if inserting mutations allows the genetic algorithm to converge to other local

minimums with lower values of $^{D}/_{L}$. So, the simulation setup is the same as sub-section 6.2.1, with the addition of an option related to the mutation. For the mutation function, one chose *@mutationadaptfeasible*. The mutation chooses a direction and step length that satisfies bounds and linear constraints for the mutation. The default value of the mutated individuals is 10%, so, by choosing a population of 10 individuals, 1 is mutated in each generation. While crossovers mix genetic material between parents to create offspring, mutations make random tweaks directly to individual genes. Crossovers blend parental traits, while mutations add new genetic diversity. This diversity lets explore different solution paths and avoid getting stuck in less-than-ideal solutions too soon.

6.2.3 Shock control bump optimization - Airfoil 3

After retrieving the airfoil of higher $^{L}/_{D}$ from the previous optimizations, Airfoil 1 or 2 are subject to the addition of a bump.

The upper and lower bounds of the bump utilized in this work were set as the following Table 6.

	lb	ub
xb	0.2	0.8
lb	0.1	0.35
cb	0.3	0.7
hb	0.001	0.01

Table 6: Upper and lower bounds used in the bump optimization.

The populations size and generations were set as 10, and 1000, respectively.

The objective function after receiving the k values, loads the parameterization of the optimised airfoil and computes the baseline geometry. Based on the bump parameters, the bump is added to the airfoil, and the solver is run. Mutations were also added in this analysis.

7 Results

In this section one present the results for the optimization of the NACA 4412 airfoil. One starts by presenting the results, and in sub-section 7.1, 7.2, and 7.3 several plots are presented for Airfoil 1, 2, and 3, respectively.

The geometry results of the Genetic Algorithm for the 3 analysis are present in Table 8. In Table 8 the performance is present. From these results, Airfoil 2 has the higher Lift-to-drag ratio and the airfoil that is chosen as the output of the optimization proposed in this work.

	IGP							Tail		Bump		
	T	Xt	$\overline{p_0}$	$\overline{\beta_{TE}}$	c1	c2	c3	c4	p	xb	lb	cb	hb
Airfoil 1	0.1201	0.3862	1.0197	0.4891	0.6905	0.7931	-0.0030	0.0490	0	-	-	-	-
Airfoil 2	0.1201	0.3980	1.1025	0.5943	0.5760	0.9072	-0.0072	0.0360	0.008	-	-	-	-
Airfoil 3	0.1201	0.3980	1.1025	0.5943	0.5760	0.9072	-0.0072	0.0360	0.008	0.7731	0.2981	0.4574	0.0035

Table 7: GA results for the retrieved airfoils.

	Lift	Drag	Lift to drag ratio
Airfoil 1	9033.8 N	1427.8 N	6.3273
Airfoil 2	8821.3 N	1355.3 N	6.5089
Airfoil 3	8843.1 N	1364.2 N	6.4819

Table 8: Results for the retrieved airfoils.

Running the genetic algorithm considering the optimization, as explained in section 6.2.1, leads to the airfoil shape as presented in figure 9. As one can see, the airfoils presents a

shape format similiar to a supercritical airfoil. This result was expected as has some previous works shown [5].

Also, it was expected to keep along the the upper surface the higher mach number and consequently the expansion wave so the pressure reduces. However, the obtain airfoil shape as more flat upper surface meaning that the normal shock wave, for the NACA 4412 figure 2, is weaker as can be seen in figure 10.

Considering both the lower mach number throw the expansion wave and the weaker shock wave, comparing tables 2 and 8, this flow changes lead to reduction in lift but a huge reduction in drag. In conclusion, the $^{L}/_{D}$ ratio increased 47.6%.

Airfoil 2 is an relative improvement of 2.87% of the ratio $^{L}/_{D}$ from Airfoil 1 meaning that the mutations are an important parameter in genetic parameters once improving $^{L}/_{D}$ is a key business in Aeronautical Industry. One also notices the trailing edge angle is higher in Airfoil 2, contributing to the positive component of drag and better results.

Both Airfoil 1 and Airfoil 2 tool fuffill the thickness relative maximum $12\% \pm 0.1\%$ as can be stated in figures 13 and 25 Another geometry limitation was the trailing edge angle. Figures 14 and 20 shows that a 30° limit was achieved. All this geometry limits were implemented directly in the airfoil parameterization as explained in a previous section.

In the case of the addition of the bump, one may observe in Table 8 that when the bump is added to Airfoil 2 (Airfoil 3, whose geometry is presented in Figure 21) the Lift to drag ratio diminishes. This is not an excepted result and due to time constraints, this result is still kept on the report for the analysis of the failure of the added bump. Table 8 results indicate that both the lift and drag increase but the drag increases most. To provide an intuition of these results one first resort to Figures 16, and 22 which present the cp distribution for Airfoil 2 and 3, respectively. The distributions present to be quite similar. However, in the bump region, an expansion fan appears, reducing pressure on the top of the airfoil. This can be observed also from the pressure distribution of Airfoil 3 in Figure 23. This explains the lift augmentation. However, the decrease of pressure will lead to an increase in speed before the shock at the end of the airfoil. This in turn leads to a stronger shock, increasing the drag. This speed increase can be observed, by comparing Figures 18, and 24 presenting the mach distribution for the airfoils. So, the shock control bump outputted by the GA does not force the shock at the bump, and the benefits of this feature are not observed in the results of this report. To note that although the bump is added, the maximum thickness and camber angle limits are fulfilled and their distribution through the airfoil chord is present in Figure 25 and 26 respectively.

7.1 Airfoil 1 Plots

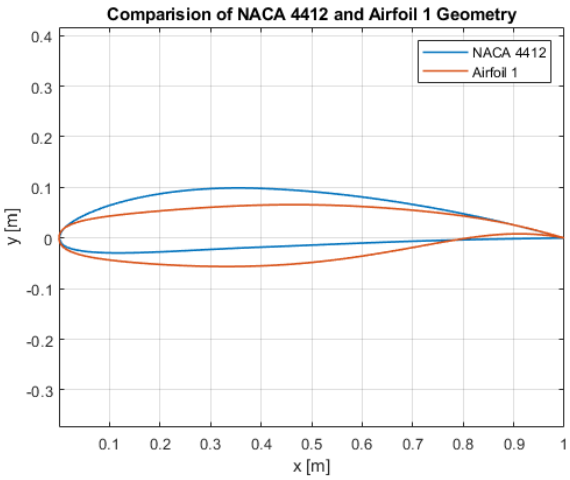


Figure 9: Airfoil 1 Geometry comparison.

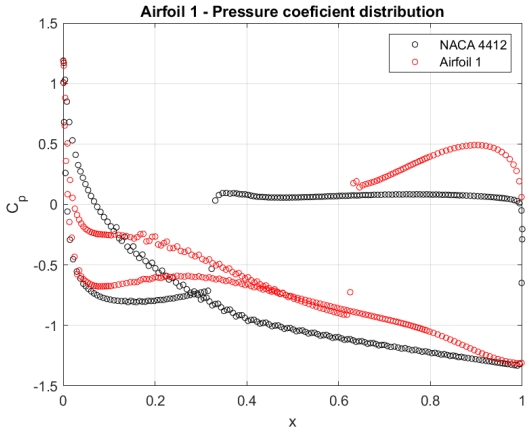


Figure 10: Airfoil 1 Cp comparison.

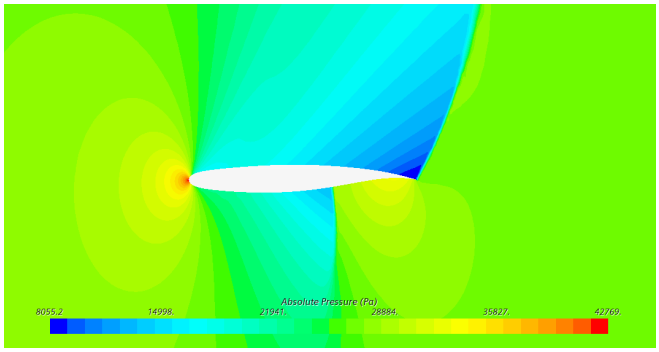


Figure 11: Airfoil 1 Pressure Field.

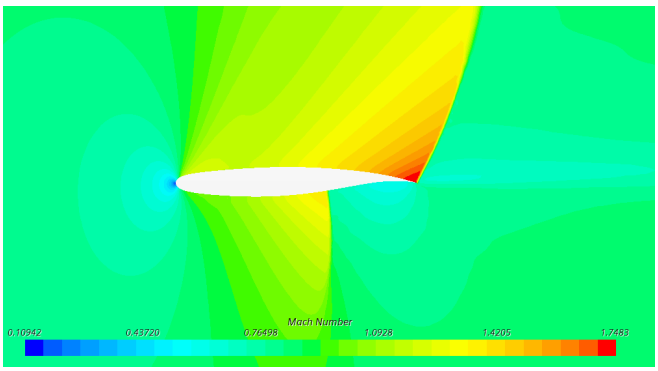


Figure 12: Airfoil 1 Mach Field.

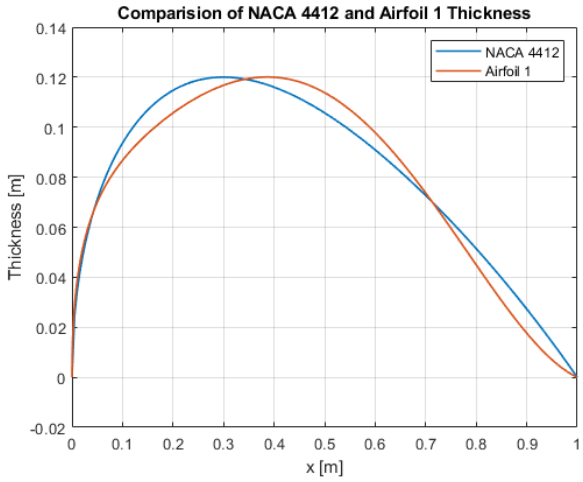


Figure 13: Airfoil 1 Thickness comparison.

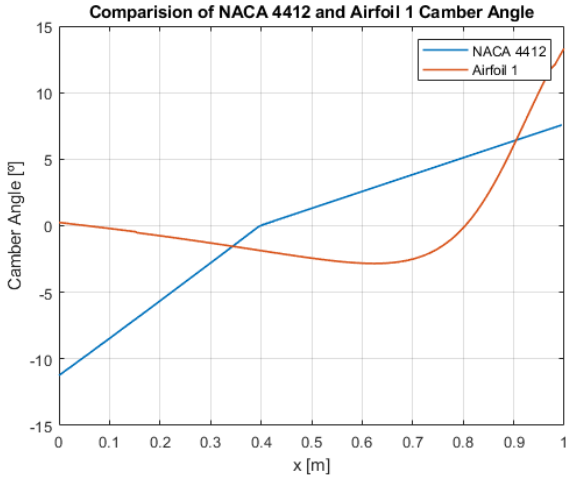


Figure 14: Airfoil 1 Camber Angle comparison.

7.2 Airfoil 2 Plots

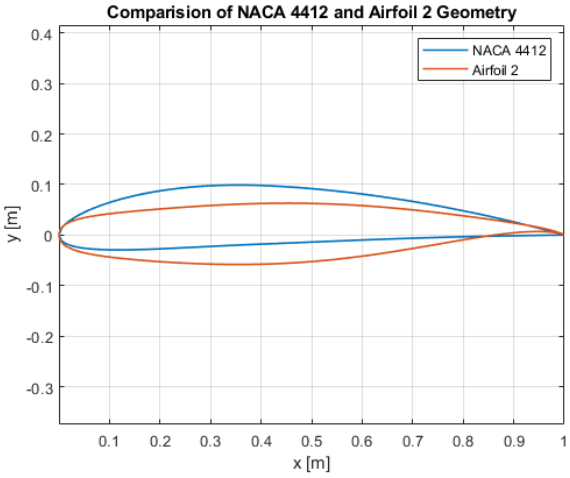


Figure 15: Airfoil 2 Geometry comparison.

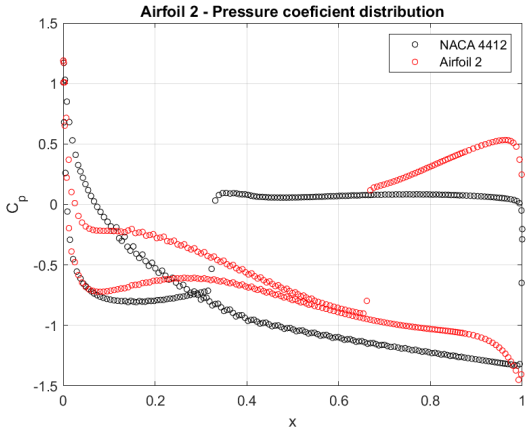


Figure 16: Airfoil 2 Cp comparison.

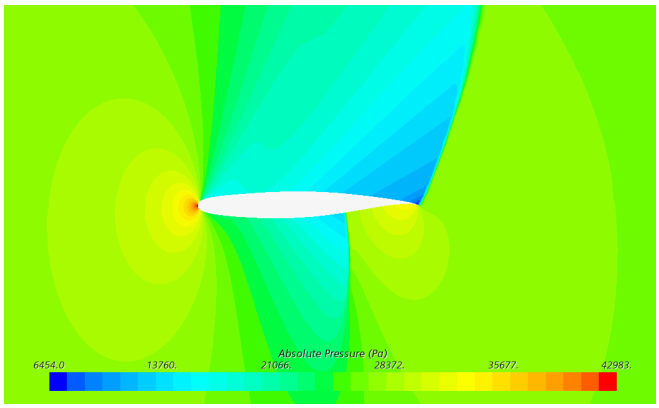


Figure 17: Airfoil 2 Pressure Field.

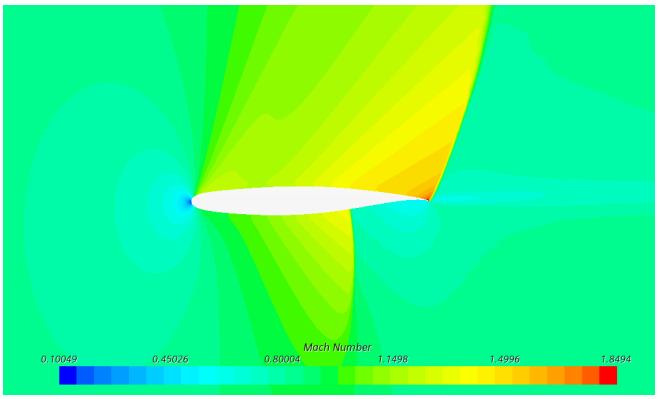


Figure 18: Airfoil 2 Mach Field.

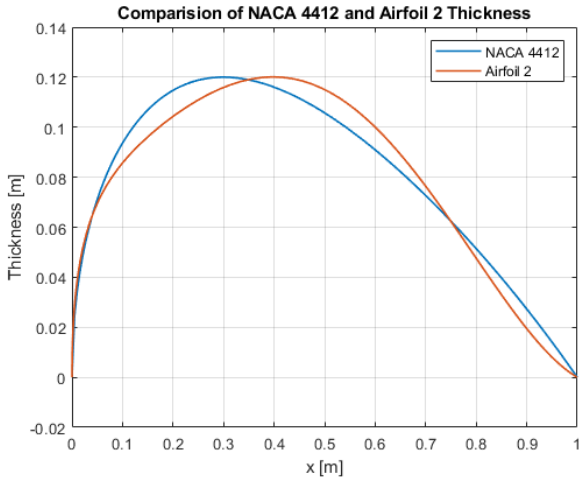


Figure 19: Airfoil 2 Thickness comparison.

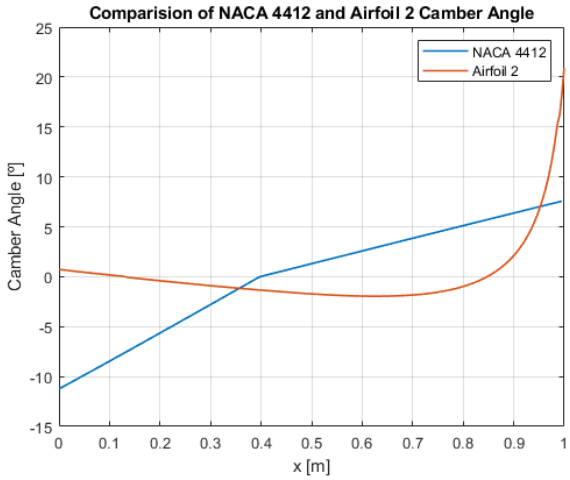


Figure 20: Airfoil 2 Camber Angle comparison.

7.3 Airfoil 3 Plots

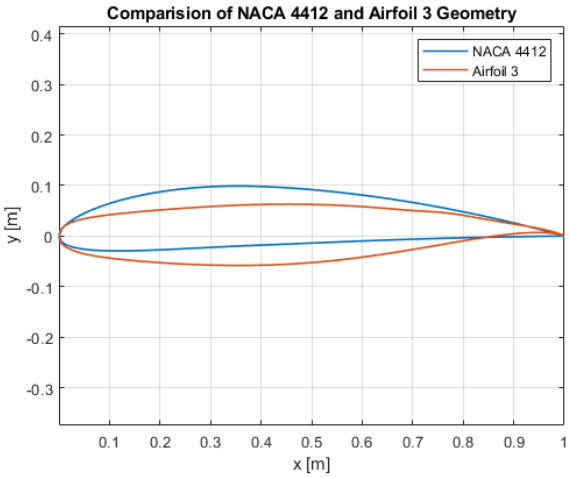


Figure 21: Airfoil 3 Geometry comparison.

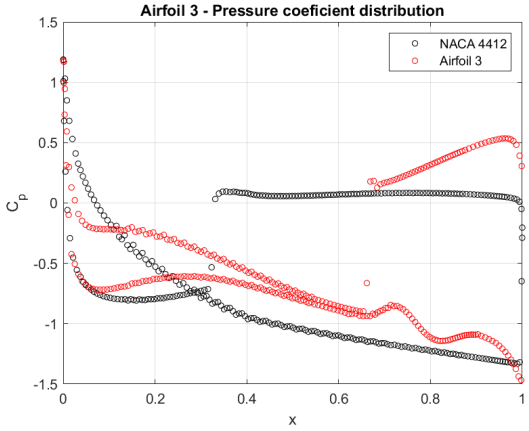


Figure 22: Airfoil 3 Cp comparison.

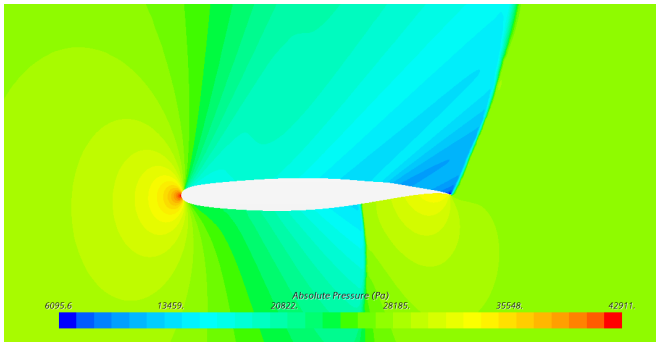


Figure 23: Airfoil 3 Pressure Field.

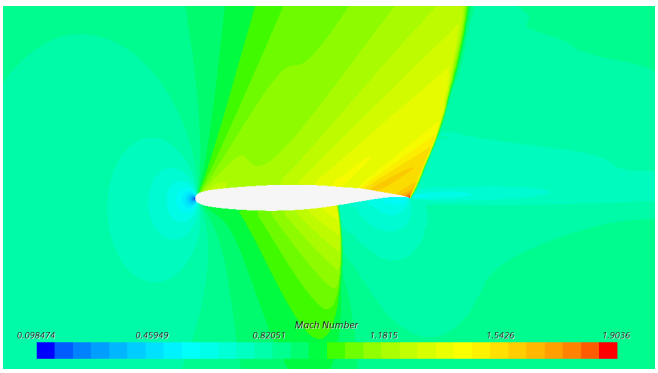


Figure 24: Airfoil 3 Mach Field.

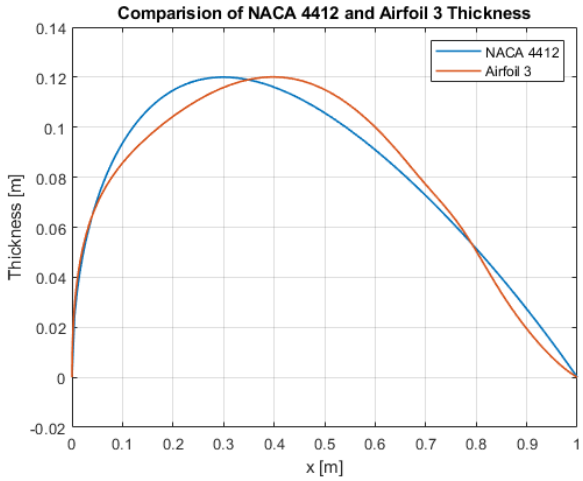


Figure 25: Airfoil 3 Thickness comparison.

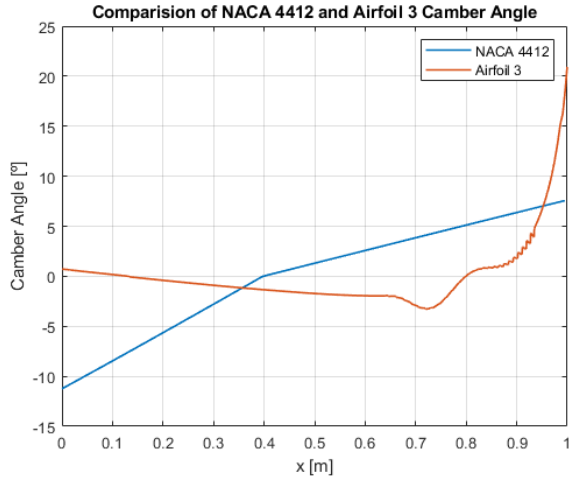


Figure 26: Airfoil 3 Camber Angle comparison.

8 Conclusion

In conclusion, a GA for the improvement of NACA4412 $^{L}/_{D}$ was implemented with recourse to IGP parameterization, trailing edge modification, and bump addition. The parameterization IGP was well set, and the ga could achieve a Genetic Algorithm output optimized solutions. It was clear that the GA was doing its job once over generations the obtained airfoil showed an improvement in $^{L}/_{D}$. The trailing edge modification was performed, but no significant change was encountered either aerodynamically or in $^{L}/_{D}$, meaning that the p values tended to 0.

When mutations were added, the GA found a minimum with other parameters, stating the importance of this feature in the genetic algorithm. The bump addition in this work's context revealed not achieving the desired results, by not forcing the shock to occur at the bump, and increasing the shock at the trailing edge.

Over the optimisation process some challenges were found, however the work done accomplished the final goal of optimising the NACA 4412 airfoil , outputting a supercritical similar airfoil with $^{L}/_{D} = 6.5089$, meaning a 51.8% of increase.

Bibliography

[1] S. Skinner and H. Zare-Behtash, "State-of-the-art in aerodynamic shape optimisation methods," en, *Applied Soft Computing*, vol. 62, pp. 933–962, Jan. 2018, ISSN: 15684946. DOI: 10.1016/j.asoc.2017.09.030. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494617305690 (visited on 03/21/2024).

[2] X. Lu, J. Huang, L. Song, and J. Li, "An improved geometric parameter airfoil parameterization method," en, *Aerospace Science and Technology*, vol. 78, pp. 241–247, Jul. 2018, ISSN: 12709638. DOI: 10.1016/j.ast.2018.04.025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963817304686 (visited on 03/28/2024).

[3] Y. Tian, P. Liu, and P. Feng, "Shock control bump parametric research on supercritical airfoil," en, *Science China Technological Sciences*, vol. 54, no. 11, pp. 2935–2944, Nov. 2011, ISSN: 1674-7321, 1862-281X. DOI: 10.1007/s11431-011-4582-y. [Online]. Available: http://link.springer.com/10.1007/s11431-011-4582-y (visited on 03/30/2024).

[4] P. Della Vecchia, E. Daniele, and E. DAmato, "An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm," en, *Aerospace Science and Technology*, vol. 32, no. 1, pp. 103–110, Jan. 2014, ISSN: 12709638. DOI: 10.1016/j.ast.2013.11.006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963813002046 (visited on 03/24/2024).

[5] N. Salunke, J. Ahamad, and S. Channiwala, "Airfoil Parameterization Techniques: A Review," *American Journal of Mechanical Engineering*, vol. 2, pp. 99–102, Jan. 2014. DOI: 10.12691/ajme-2-4-1.

Appendix

A Java function for importing foil coordinates in *STAR-CCM+*

```
private double[][] readCSV(String csvFile) {

    List<double[]> rows = new ArrayList<>();

    try (BufferedReader br = new BufferedReader(new FileReader(csvFile))) {
        String line;
        while ((line = br.readLine()) != null) {
            String[] values = line.split(";");
            double[] row = new double[values.length];
            for (int i = 0; i < values.length; i++) {
                row[i] = Double.parseDouble(values[i]);
            }
            rows.add(row);
        }
    }
    catch (IOException | NumberFormatException e) {
        e.printStackTrace();
    }

    // Convert List<double[]> to double[][]
    int numRows = rows.size();
    int numCols = numRows > 0 ? rows.get(0).length : 0;
    double[][] points_out = new double[numRows][numCols];
    for (int i = 0; i < numRows; i++) {
        points_out[i] = rows.get(i);
    }

    return points_out;
}
```