COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0204
A METHODOLOGY OF AIRFOIL AND WING OPTIMIZATION USING GENETIC ALGORITHMS

Giovana Weffort Fernandes
Manuel Nascimento Dias Barcelos Júnior
Universidade de Brasília - Faculdade UnB Gama - St. Leste Projeção A - Gama Leste - Brasília - Brasil
giwf015@gmail.com, manuelbarcelos@aerospace.unb.br

Abstract. *The present paper aims to present a methodology of optimization of airfoils and wings through genetic algorithms. The plan was to develop accessible and in-depth algorithms which may be configured and adapted according to the user's preference. The optimization algorithms allow for the configuration of the objective functions and constraints in terms of four distinct aerodynamic coefficients (lift coefficient, drag coefficient, L/D ratio and pitch moment coefficient), in various possible combinations, and they employ the CST as the airfoil parametrization method. The main version of the codes was written in GNU Octave, using XFOIL as the flow solver in the airfoil algorithm, and using APAME as the solver in the wing algorithm. The genetic algorithm is evaluated via standard test functions, and case studies are defined aiming to optimize specific geometries in given flight conditions established by a Reynolds number and an angle of attack. The results were successful within the stipulated objectives and allow for discussion of the behavior of the optimization process, and how the modified geometries compare to their original versions. In conclusion, the main advantages of this methodology are its reliability and ease of adaptations. Every element of the optimization process may be changed to suit the needs of the user, which include the object of study, the geometric parametrizations, the objective functions, constraints, the optimization algorithm itself, and other aspects. A basic demonstration of this is given at the end of the paper, where a given result from the Python version of the wing optimization algorithm, using VSPAERO as the solver, is shown.*

Keywords: *Airfoils, Wings, Genetic Algorithms, CST*

1. INTRODUCTION

Aeronautical engineering is a field that provides a plethora of situations which may be approached as optimization problems. Aerodynamics alone is a discipline recurrent as the subject to optimization algorithms due to the broad spectrum of parameters to be analyzed, which include the relevant aerodynamic geometries, non dimensional coefficients, flight conditions, flow properties, etc. Among these items, airfoil and wing optimization is a particularly common subject, a fact justified by the great relevance these components have in aircraft design. For instance, ZHU *et al.* (2004) performs the optimization of airfoils using a hybrid evolutionary algorithm, which improves some weaknesses of evolutionary algorithms. Another example is done by Koziel and Leifsson (2011), which uses surrogate-based optimization in order to reduce the computational costs of the CFD simulations. Wing optimization problems may also present themselves in a wide spectrum of different configurations. Körpe and Kanat (2019) optimize a UAV wing in terms of both aerodynamic and structural requirements, manipulating the airfoil shapes and the planform (in a manner similar as the one adopted in this study), and de L. S. M. Vilela and da Silva (2019) perform the optimization process without numerical simulations, instead opting for analytical equations relating to the performance of the aircraft. Airfoil optimization has been noted to be a topic slightly more common in research papers than wing optimization, and using a greater variety of methods and objectives. This tendency may be justified in that wing optimization is a more difficult problem to implement, due to the 3D nature of both the geometry and the airflow, which result in more design variables and longer run times.

The present study is a particular interpretation and implementation of these optimization problems, with the scope limited to subsonic airflows, employing genetic algorithms, panel methods, and using lift coefficients, drag coefficients, lift-to-drag ratios and moment coefficients as objective functions or restrictions. One algorithm is dedicated to airfoil optimization, using CST as the geometry parametrization method. The second algorithm is used for wing optimization, also using CST to parametrize the airfoils, along with additional parameters to describe the planform geometry. Planforms are limited to a few shapes (rectangular, trapezoidal and double trapezoidal) to simplify the algorithm in regards to both the actual programming of the genetic operators and the number of design variables. In both algorithms, the search space is limited to geometries whose design parameters are close to the parameters of a specified initial geometry. Case studies representing hypothetical design requirements are executed in order to demonstrate the methodology proposed here. These case studies consist of L/D maximization problems, some of which are executed without constraints, and others with constraints represented by the lift coefficient and the moment coefficient.

The algorithms are programmed in Octave, using XFOIL as the solver for the airfoil algorithm and APAME for the wing algorithm. A secondary version of the wing algorithm using VSPAERO as the solver was also developed. In this case, the algorithm was written in Python in order to take advantage of the VSPAERO Python API. A single example result of this version of the algorithm shall be demonstrated.

2. GENETIC ALGORITHMS

Some types of optimization problems are not possible to solve via deterministic means. On the other hand, taking a brute force approach is also not practical due to the considerable computational and time costs. In between both of these extremes are the meta-heuristic algorithms, characterized by two main features: selection of the best results and randomization (Yang, 2010). The randomization avoids dependence to the algorithm's starting points, and the selection of the best solutions ensures the algorithm actually converges to the desired results instead of roaming arbitrarily in the search space.

Meta-heuristic algorithms are usually based upon processes in nature, and such is the case of genetic algorithms, whose main philosophy of implementation is the survival of the fittest. The genetic algorithm carries out the optimization process as an analogy to the progression of natural selection, including representations of biological mechanisms such as reproduction, crossover and mutation (Rao, 2009). The basic steps in the operation of a genetic algorithm are

1. Generation of the initial population;

2. Evaluation of the individuals' fitnesses;

3. Reproduction and crossover;

4. Mutation.

The algorithm then returns to step 2, running in an iterative manner until a stop criterion is achieved. Another mechanism commonly added to genetic algorithms is elitism, which, in its most basic form, consists of copying the most fit individual of a population to the next iteration. This approach is the one adopted for the algorithms of this study. The benefit of implementing elitism is to avoid the oscillation of the best results due to the randomness in the genetic operators, which may happen when the algorithm converges to a specific set of genes. Therefore, even if the algorithm does not improve a particular solution, said solution is not lost in the following iterations.

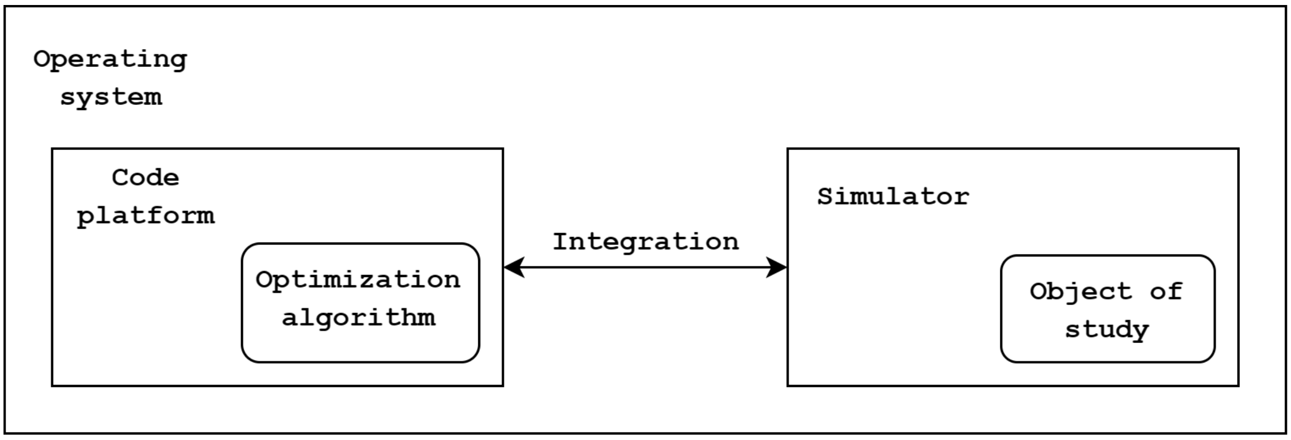


Figure 1: Representation of the methodology as a modular system.

The outlook of the methodology employed in this work may easily be understood as a modular system, as illustrated by Fig. 1. For example, for the airfoil optimization codes of this study, the operating system is Windows 10, the code platform is Octave, the optimization algorithm is the genetic algorithm, the simulator is XFOIL and the object of study are airfoils. Surely, if modifications are made to one of these elements, the rest of the components must be adapted if needed. So, for instance, when the object of study is changed to wings, the simulator is also changed to account for 3D flows (in this case, APAME), and the optimization algorithm is adjusted to properly deal with the geometric characteristics of wings. This philosophy is backed up by the ease of adaptation of the genetic algorithms, which may be used to solve multiple types of optimization problems without much hindrance.

2.1 Validations

The algorithms written for these studies were validated using standard test functions (Yang, 2010). Some of these validations are shown here. The first test was done with Ackley's function:

$$f(x) = -20 \exp \left[-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right] - \exp \left[\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right] + 20 + \exp[1] \tag{1}$$

Where $i = 1, 2, ..., n$ and the terms x_i are defined within $-32,768 \leq x_i \leq 32,768$. The global minimum is $f_* = 0$ at $\mathbf{x}_* = (0, 0, ..., 0)$. The results are shown in Tab. 1. It is clear how the population size may aid the algorithm in the search to the global optimum - with a wider genetic pool available from the start, the algorithm may therefore explore a larger portion of the search space and reach the most desired solutions. These results show that the algorithm's solutions proceed in direction of the global minimum, even if it is not reached.

Table 1: Results of the algorithm validation test with Ackley's function.

Population size	**Result**	**Number of iterations**
400	0	47
200	3.8609	64
100	6.2076	91

The second example of the validation tests performed was with Griewank's function:

$$f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{2}$$

Where $i = 1, 2, ..., n$ and the terms x_i are defined in within $-600 \leq x_i \leq 600$. The global minimum is $f_* = 0$ at $\mathbf{x}_* = (0, 0, ..., 0)$. The results are shown in Tab. 2. Again, the algorithm displays desirable behaviour in regards to its solutions, with the effects of the population size also evident.

Table 2: Results of the algorithm validation test with Ackley's function.

Population size	**Result**	**Number of iterations**
400	0,1094	22
200	1,2752	46
100	6,9458	85

The fundamental code structure used in these validation tests is the same as in the airfoil and wing algorithms, therefore, they also have the same properties proved by these validations. Tests related to configuration parameters such as population size and mutation chance were also performed, and the configurations chosen here reflect the lessons learned in those tests.

3. AIRFOILS

The analysis and design of airfoils is an important step in the design of wings. By inspecting the 2D flow over a given airfoil, it is possible to estimate the aerodynamic characteristics of a complete wing, even though said characteristics do not translate perfectly to the 3D realm (Abbott and Doenhoff, 1959). The data most commonly investigated to understand the performance of airfoils are its non-dimensional coefficients (lift, drag and pitching moment coefficients), along with the lift-to-drag ratio - all of which are used by the algorithms as objective functions or restrictions depending on the configuration.

For this work, the applicability of the algorithms is shown via example case studies, in which the goal is to optimize a NACA 2412 airfoil. For the airfoil optimization problem, the case studies are

1. Maximize L/D;
2. Maximize L/D and maintain C_l as close as possible to its original value;
3. Maximize L/D and maintain both C_l and C_m as close as possible to their respective original values.

In case studies 2 and 3, the original coefficients refer to the coefficients of the non optimized airfoil. The flight condition in all cases is defined by $Re = 1e6$ and $5°$ angle of attack, and the flow solver used for the airfoil optimization algorithm is XFOIL, a popular and robust panel method code. For all airfoil case studies, the population size is 150, mutation chance is 5% and the algorithm runs a set number of 5 iterations.

The optimization process requires that the geometry of the object of study be described by a parametrization method, so that these geometries may be systematically modified by the algorithm. In the present study, the geometric parametrization method chosen was the Class and Shape function Transformation (CST).

3.1 Class and Shape function Transformation (CST)

The CST is a versatile parametrization method which may be used to generate multiple types of 2D geometries applicable as cross sections and axisymmetric bodies (Kulfan and Bussoletti, 2006). The CST itself is comprised of a class function and a shape function, with the former defining the overall shape of the geometry, and the latter manipulating details of the contour. The general equation is

$$\frac{z}{c}(x/c) = C(x/c)S(x/c) + \frac{x}{c}\frac{\Delta Z_{TE}}{c} \tag{3}$$

Where x is the abscissa, z is the ordinate, c is the chord length and ΔZ_{TE} is the vertical distance of the trailing edge to the $z = 0$ line. The class and shape functions are, respectively, $C(x/c)$ and $S(x/c)$. For the case of airfoils, the class function is

$$C(x/c) = \sqrt{\frac{x}{c}}\left(1 - \frac{x}{c}\right) \tag{4}$$

The shape function is guided by a Bernstein polynomial of order n, given by

$$BPn = \sum_{r=0}^{n} K_{r,n} x^r (1 - x)^{n-r} \tag{5}$$

Where the factors $K_{r,n}$ are binomial coefficients:

$$K_{r,n} = \frac{n!}{r!(n - r)!} \tag{6}$$

The shape function for the airfoil is simply Eq. 5 multiplied by design variables A_r, which affect specific regions of the contour from the leading edge to the trailing edge. Variable A_0 corresponds to the leading edge and is given by

$$A_0 = \sqrt{\frac{2R_{LE}}{c}} \tag{7}$$

Where R_{LE} is the leading edge radius. Similarly, A_n is attributed to the trailing edge, and is given by

$$A_n = \tan\beta + \frac{\Delta Z_{TE}}{c} \tag{8}$$

Where β is the trailing edge boattail angle. Adopting $\zeta = z/c$, $\psi = x/c$ and $\Delta\xi = \Delta Z_{te}/c$, the complete equation of CST for airfoils is

$$\zeta(\psi) = \sqrt{\psi}(1 - \psi)\sum_{r=0}^{n}\left[A_r\left[K_{r,n}\psi^r(1 - \psi)^{n-r}\right]\right] + \psi\,\Delta\xi \tag{9}$$

Equation 9 describes only one surface of the airfoil at a time. To obtain the lower surface, it is necessary only to multiply the resulting ordinates by -1. By attributing a set of variables to the upper surface and another set to the lower surface, it is possible to control all regions of the airfoil's contour. Symmetric airfoils are generated when the variable set for the upper surface is equal to the one of the lower surface.

In the present study, the CST parametrization was configured with 6th order Bernstein polynomials.

3.2 Reverse CST using the least squares method

Optimizing a specific airfoil shape requires first that the airfoil be described in the parametrization method used by the algorithm. Because the number of coordinates is bigger than the number of design variables, the least squares method (LSM) is most adequate to better fit the parametrized airfoil shape to the coordinates, and is also the method chosen by Kulfan and Bussoletti (2006). The general formulation for the LSM, as given by da R. Lopes (1996), is

$$\begin{cases} \left[\sum_{k=1}^{m} g_1(x_k)g_1(x_k)\right]\alpha_1 + \ldots + \left[\sum_{k=1}^{m} g_n(x_k)g_1(x_k)\right]\alpha_n = \sum_{k=1}^{m} f(x_k)g_1(x_k) \\ \left[\sum_{k=1}^{m} g_1(x_k)g_2(x_k)\right]\alpha_1 + \ldots + \left[\sum_{k=1}^{m} g_n(x_k)g_2(x_k)\right]\alpha_n = \sum_{k=1}^{m} f(x_k)g_2(x_k) \\ \vdots \\ \left[\sum_{k=1}^{m} g_1(x_k)g_n(x_k)\right]\alpha_1 + \ldots + \left[\sum_{k=1}^{m} g_n(x_k)g_n(x_k)\right]\alpha_n = \sum_{k=1}^{m} f(x_k)g_n(x_k) \end{cases} \tag{10}$$

Where x_1, x_2, …, x_m and $f(x_1)$, $f(x_2)$, …, $f(x_m)$ is the data for which a correlation is to be obtained, $g_1(x_1)$, $g_2(x_2)$, …, $g_m(x_m)$ are functions chosen to correlate the data and α_1, α_2, …, α_m are the unknowns. In the present problem, the points x and $f(x)$ represent the coordinates of the airfoil and the α values are the design variables that must be calculated. The functions $g(x)$ are obtained from Equation 9: by expanding its summation term, it is possible to build a linear system in the molds of Equations 10 to solve for the design variables. For a single airfoil, this process must be done twice - once for each surface.

Although the trailing edge thickness is already known from the airfoil coordinates, its inclusion as a variable in the LSM is required so that the other variables are correctly estimated. After obtaining all parameters, the trailing edge thickness values obtained from the LSM are neglected in favor of those encoded in the airfoil coordinates. Care must also be taken with the leading edge, because the absolute value of the slope of the curve must not exceed $90°$. The solution implemented in this study to avoid this problem was to move the complete set of coordinates in the x-z plane to ensure the leftmost point in the airfoil contour is positioned at the origin of the plane.

4. WINGS

Due to the multitude of different types of performance a given aircraft may have, wing design, in turn, also is a discipline with a wide range of possibilities regarding the geometry. In the case of the present work, the algorithms deal with wings of rectangular, trapezoidal and double trapezoidal planforms. These geometries were adopted for two reasons: first, it simplifies the parametrization of the geometries in the algorithm, and second, these planforms are very traditional in aircraft design, with or without modifications.

The parametrization of the wings establishes the planform dimensions and the airfoils. For trapezoidal wings, airfoils are defined in the root and in the tip of the wing, and the planform is defined by the span, chord lengths (root and tip), washout angle and wing sweep. For double trapezoidal wings, an additional station ("middle") between the root and the tip is created, with the wing being defined with two parts: the inner part, from the root station to the middle station, and the outer part, from the middle station to the tip station. The airfoil at the middle station may be specified in the same manner as done in the root an tip stations. The chord length at the middle station may be set separately from the other stations, and different values of wing sweep, span and washout may be attributed to each part of the wing. In any case, rectangular planform wings are simply those of constant chord length, no matter the type, and sweep angles are defined in terms of the leading edge. In all wings, the airfoils specified at the stations are parametrized using CST, also using 6th order Bernstein polynomials.

The solver used for the wing optimization algorithm is APAME (Aircraft Panel Method), a potential flow solver which implements the 3D panel method to perform aerodynamic simulations. It should be noted that APAME does not estimate viscous drag (Filkovic, 2008), a fact that must be taken into account when analyzing the results of the algorithm. The case studies for the wing optimization algorithm are the same as those for the airfoil algorithm, with the same flight condition - disregarding only the Reynolds number, which is irrelevant due to the flow modeling in APAME. The wing to be optimized has 14m span, 1.5m root chord, 0.5m tip chord, zero leading edge sweep, and zero washout. The root and tip airfoils are, respectively, a NACA 2412 and a NACA 0010. This wing shall be encoded into the algorithm allowing it to be changed to a double trapezoidal planform if needed. To achieve this, the middle station is positioned at halfway the semi-span, with a chord length equal to the mean of the other chords. The initial airfoil of the middle station is a linear interpolation of the other two airfoils. For the wing case studies, the population size is 300, mutation chance is 5% and the algorithm runs a set number of 5 iterations.

When generating the mesh for APAME, the wing sections in between the defined stations are interpolated linearly from their respective airfoils. This is necessary to improve the resolution of the mesh. If not for this interpolation, the only nodes available for the mesh would be those in the airfoils defined at the stations.

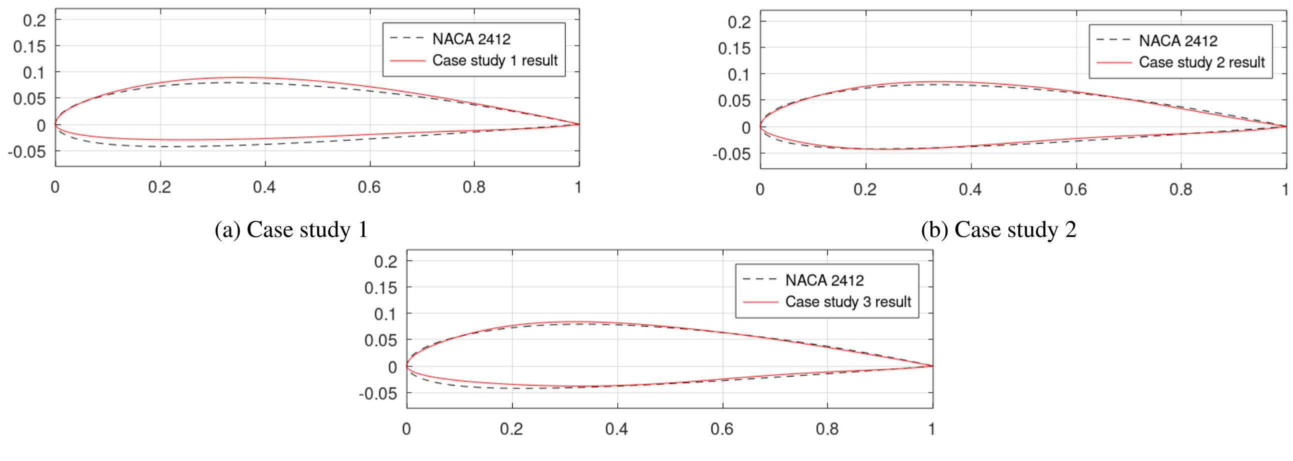
As a demonstration of the modular nature of the methodology, a final example shall be given by using the Python version of the wing optimization algorithm, which employs VSPAERO as the solver via the OpenVSP Python API. VSPAERO shall be set to VLM (Vortex Lattice Method) configuration to contrast and compare its results to APAME's 3D panel method results. The case study chosen for this final example shall be the second, as defined previously. OpenVSP has memory leaks which also manifest through its API (Swait and McDonald, 2023). These memory leaks accumulate throughout the code's execution, gradually slowing down the API's performance and, in turn, the algorithm's performance as well. Also, geometry generation errors in OpenVSP related to the usage of airfoil coordinate files may occasionally appear, halting the algorithm completely. These geometry errors occur more commonly with larger population sizes. Due to both of these factors, to keep the algorithm's execution time within practical values, the population size was reduced to 30 in this demonstration with VSPAERO. All other configurations are kept as before. Finally, it should be noted that the structure of the algorithm in both versions of the code is the same. The only differences from the Octave version to the Python version of the algorithm are the adaptations necessary to run the OpenVSP Python API.

5. DISCUSSIONS AND RESULTS

The case studies (referred to as CS in the tables) proposed previously were executed and their respective results are presented in the following tables and figures. First, the results from the airfoil optimization algorithm are demonstrated. Table 3 gathers the aerodynamic data of the original airfoil and the optimized ones, and Fig. 2 shows each of the optimized airfoils in comparison to the original.

Table 3: Results of the airfoil case studies (coefficients).

Airfoil	$\mathbf{C_l}$	$\mathbf{C_d}$	$\mathbf{L/D}$	$\mathbf{C_m}$
Original (NACA 2412)	0.8615	0.0084	102.0734	-0.0639
CS 1 result	0.9729	0.0078	124.7307	-0.0829
CS 2 result	0.8920	0.0075	117.8335	-0.0668
CS 3 result	0.8809	0.0075	116.5211	-0.0635



(a) Case study 1

(b) Case study 2

(c) Case study 3

Figure 2: Results of the airfoil case studies (geometries).

The first conclusion that may be taken from the results in Tab. 3 is that the maximization of lift-to-drag ratio was successful in all case studies, with the highest increase obtained in the first case study, resulting in an airfoil with an L/D 22.2% higher than that of the NACA 2412 profile. The superiority of the the first case study's result in regards to the value of the lift-to-drag ratio was already expected, since the addition of constraints to the optimization processes inhibits the algorithm from exploring the full extent of the design space. With the C_l constraint included, the algorithm must instead, as much as possible, minimize the drag coefficient, which it achieves even further in case studies 2 and 3 compared to case study 1's results. It should also be noted that the moment coefficient in case study 2's result lies close to the original value, with case study 3's result bringing it even closer due to the explicit use of the C_m constraint. Both of these airfoils are more conservative in design in contrast to case study 1's result, which has a 29.7% increase in the absolute value of the moment coefficient. Figure 2 confirms the observations in regards to the magnitude of modifications allowed by the constraints (and their absence): the airfoil from case study 1 is the one most different from the original foil.

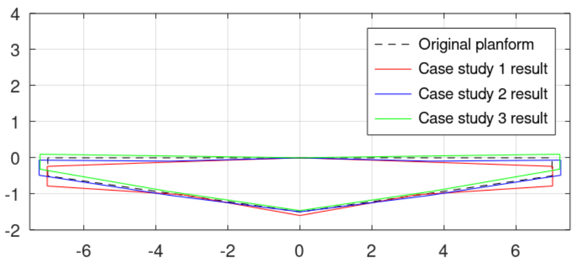
Next, results from the wing optimization case studies are presented. Table 4 shows the aerodynamic data of the relevant wings, and Tab. 5 shows the physical dimensions of these wings, where b is the complete span, b_m is the span between the middle stations in both sides of the wing, c_r is the root station chord length, c_m s the middle station chord length, c_t is the tip station chord length, λ_1 is the sweep of the leading edge of the inner part of the wing, λ_2 is the sweep of the leading edge of the outer part of the wing, and ϕ is the geometric wing twist angle (or washout, for negative values of ϕ). Lastly, Fig. 3 displays the optimized shapes of the planforms and airfoils. For the figures displaying the airfoils, the chords are set to unity instead of the chord lengths at their respective stations.

Table 4: Results of the wing case studies (coefficients).

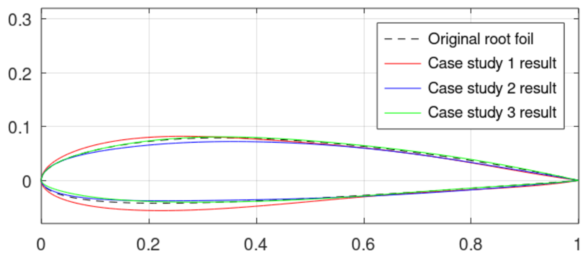
Wing	$\mathbf{C_l}$	$\mathbf{C_d}$	$\mathbf{L/D}$	$\mathbf{C_m}$
Original	0.5373	0.0735	7.3102	-0.1302
CS 1 result	0.3249	0.0285	11.4000	-0.1133
CS 2 result	0.4938	0.0532	9.2819	-0.1731
CS 3 result	0.5484	0.0664	8.2590	-0.1361

Table 5: Results of the wing case studies (dimensions).

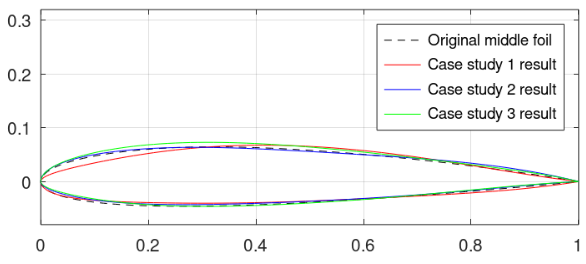
Wing	$\mathbf{b[m]}$	$\mathbf{b_m[m]}$	$\mathbf{c_r[m]}$	$\mathbf{c_m[m]}$	$\mathbf{c_t[m]}$	$\boldsymbol{\lambda_1}[^\circ]$	$\boldsymbol{\lambda_2}[^\circ]$	$\boldsymbol{\phi}[^\circ]$
Original	14.0000	7.0000	1.5000	1.0000	0.5000	0.0000	0.0000	0.0000
CS 1 result	14.0311	6.5020	1.5989	0.9184	0.5464	1.8704	1.9873	-4.9346
CS 2 result	14.4837	7.1892	1.4972	0.9489	0.4198	1.4789	-0.4551	-2.5627
CS 3 result	14.4343	7.3510	1.4625	0.9793	0.4142	-0.8186	-0.7473	-1.4223



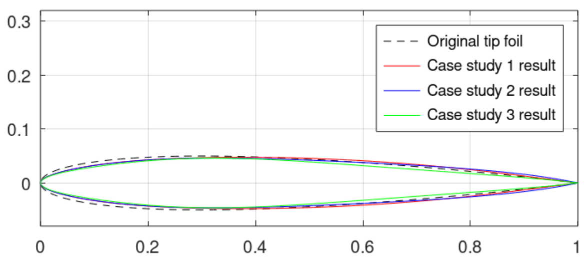
(a) Planforms



(b) Root airfoils



(c) Middle airfoils



(d) Tip airfoils

Figure 3: Results of the wing case studies (geometries).

Similarly to the airfoil optimization cases, the wing to obtain the highest lift-to-drag ratio was the one from case study 1, precisely due to the absence of constraints, which also results in the geometry which is most different from the original wing, especially in regards to the airfoils. In contrast to the respective result from the airfoil algorithm, however, the lift-to-drag ratio was obtained via a considerable reduction in the drag coefficient despite being accompanied by a reduction of 39.5% of the lift coefficient. With the activation of the lift coefficient constraint, the algorithm is not able to reduce the drag coefficient as was achieved in case study 1. This may be explained by the fact that the lift coefficient is kept at a higher value than that of case study 1's wing, which results in more production of lift and, in turn, more induced drag, which manifests through the drag coefficient. The absolute value of the moment coefficient also experiences a reduction in case study 1, which may be explained by the smaller value of the lift coefficient. Case study 3 resulted in the wing possessing the smallest increase of the lift-to-drag ratio, but also conforming closely to the constraints of the lift and moment coefficients - a behaviour also seen in the airfoil algorithm's result.

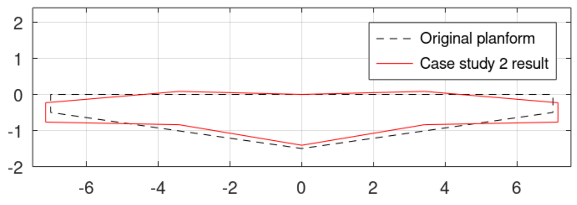
Finally, the following tables and figures present results from the python version of the wing optimization algorithm, operating with VSPAERO's VLM solver.

Table 6: Results of the second wing case study using VSPAERO (coefficients).

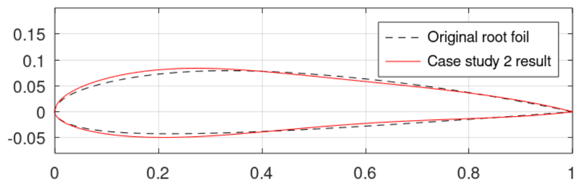
Wing	C_l	C_d	L/D	C_m
Original	0.5896	0.0148	39.7870	-0.3051
CS 2 result	0.5892	0.0142	41.3613	-0.3009

Table 7: Results of the second wing case study using VPSAERO (dimensions).

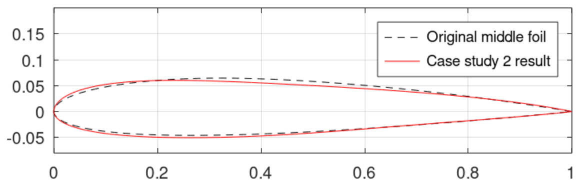
Wing	$b[m]$	$b_h[m]$	$c_r[m]$	$c_m[m]$	$c_t[m]$	$\lambda_1[°]$	$\lambda_2[°]$	$\phi[°]$
Original	14.0000	7.0000	1.5000	1.0000	0.5000	0.0000	0.0000	0.0000
CS 2 result	14.2791	6.8278	1.4093	0.9242	0.5371	-1.4107	4.8201	0.0000



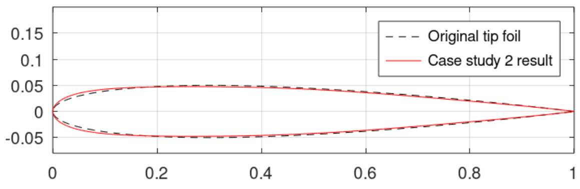
(a) Planforms



(b) Root airfoils



(c) Middle airfoils



(d) Tip airfoils

Figure 4: Results of the wing case studies using VSPAERO (geometries).

The optimized wing faithfully fulfills the lift coefficient constraint, demonstrating a 4.1% increase in the lift-to-drag ratio. The decrease in drag coefficient may be attributed to the slight increase in the aspect ratio of the wing, as seen in Fig. 4a. The moment coefficient also suffers very little change.

It is important to take note of changes in the estimation of the aerodynamic properties due to the differences in the solvers. Because the algorithm directly depends on this data to conduct the optimization process, the solver, therefore, influences the behavior of the algorithm and its solutions. An example of this is seen by comparing the case study 2 result from Tables 4 and 5 to Tables 6 and 7, and comparing Fig. 3 to Fig. 4. The most notable difference in the case study 2 results is the wing planform. The root airfoil is also modified differently. When using APAME, the optimized root airfoil is slightly thinner than the original, while, with VSPAERO, the optimized root airfoil is slightly thicker. Similar observations may be deducted from the middle and tip airfoils. These changes in behavior may be attributed mainly to the different ways the drag is estimated in each of the solvers employed.

6. CONCLUSIONS

Under the proposed methodology, case studies were executed and the results are satisfactory within the stipulated goals, in regards to both the objective function of maximization of lift-to-drag ratio and the constraints related to the lift coefficient and the moment coefficient. Such analyses were made to standalone airfoils and also wings along with their respective wing sections despite the evident differences between both types of geometry, which require specialized solvers and adaptations to the code regarding the genetic operators. Such adaptations are made easier thanks to the versatility of the genetic algorithm, which, even in its base form, shows great reliability in its performance and results. That said, this type of algorithm has its issues, such as the dependency on large populations to keep the consistency of its results, and the inherent random nature of the genetic operators, which generally prevents the algorithm from reaching the same exact solutions in different runs - instead, the genetic algorithm, after a set of iterations, tends to form a population of high fitness individuals with slightly different characteristics. A popular way of improving these issues is hybridism, which switches between the genetic algorithm and a direct search algorithm according to the location in the search space.

Because so many characteristics of a given geometry can be changed, the scope of the optimization must be adequately established within the needs of the project in hands so that the algorithm does not introduce geometries which may be undesirable. Wing optimization, specially, needs more justification in regards to its design variables and the parameters used as objective functions and restrictions. This is because the goals represented by the objective functions and constraints may be reached via different means by the algorithm. For instance, a reduction in drag may be achieved by a increase of the aspect ratio, or by a reduction of the airfoils' thicknesses. The exact manner in which a geometry is changed must be controlled by choosing which geometric parameters are introduced to the algorithm as design variables.

When a given airfoil optimization algorithm is written, the choice to use XFOIL as the flow solver is the most common occurrence due to the undeniable excellence of the software and the ease of its linking to the optimization algorithm. Such a common, prevalent choice does not seem to exist for wing optimization algorithms in the same manner as in the case of airfoils. This observation may be attributed, again, to the increased complexity of simulating 3D flows. XFOIL is popular because its panel method model is accurate in a plethora of cases, it is easily integrated to the optimization algorithm, and the geometry of the airfoils is simple to describe. For the simulation of wings, there are different solvers with varying physics models, in which assumptions and simplifications must be taken into account when choosing the proper software, given the intentions in the project. The geometry of a wing is naturally more difficult to describe due to its three dimensional nature, and different solvers may require different methods of describing said geometry. For instance, in APAME, a mesh must be generated respecting the guidelines of the software regarding the organization of the nodes and panels, and the installation of the OpenVSP Pyhton API is recommended (if not obligatory) to use VSPAERO with an optimizer. Also, the integration of the solver to the algorithm may also vary - because integration refers not only to the automated execution of the solver, but also to the return of its results back to the algorithm. The solvers chosen for this study were satisfactory for the present purposes, but their respective drawbacks must be recognized: APAME does not estimate viscous drag, and the memory leaks of the OpenVSP API hinder its usefulness in optimization algorithms. Again, it is a matter of choosing the solver most appropriate for a given context.

7. REFERENCES

Abbott, I.H. and Doenhoff, A.E.V., 1959. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Dover Publications, INC, New York.

da R. Lopes, M.A.G.R..V.L., 1996. *Cálculo Numérico: Aspectos teóricos e computacionais*. MAKRON Books, São Paulo, 2nd edition.

de L. S. M. Vilela, R. and da Silva, E.C., 2019. "Aerodesign aircraft wing optimization using genetic algorithm". In *IEEE Sympoium Series on Computational Intelligence*. Xiamen, China.

Filkovic, D., 2008. "Graduate work".

Körpe, D.S. and Kanat, O.O., 2019. "Aerodynamic optimization of a uav wing subject to weight, geometric, root bending moment, and performance constraints". *International Journal of Aerospace Engineering*, Vol. 2019.

Koziel, S. and Leifsson, L., 2011. "Transonic airfoil shape optimization using variableresolution models and pressure distribution alignment". In *AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, Honolulu, Hawaii.

Kulfan, B.M. and Bussoletti, J.E., 2006. ""fundamental" parametric geometry representations for aircraft component shapes". Seattle, Washington.

Rao, S.S., 2009. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, INC, New Jersey, 4th edition.

Swait, T. and McDonald, R., 2023. "Memory usage". URL `https://groups.google.com/g/openvsp/c/BMzuD3Is02s/m/RWu3Wd2FBQAJ`. Access on 14/10/2023.

Yang, X., 2010. *Engineering Optimization: An Introduction with Metaheuristic Applications*. John Wiley & Sons, Inc, New Jersey.

ZHU, Z., FU, H., YU, R. and LIU, J., 2004. "Multi-objective optimization design of airfoil and wing". *Science in China Technological Sciences*, Vol. 47, pp. 15–25.

8. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.