encit 2020

Associação Brasileira de Engenharia e Ciências Mecânicas

18th Brazilian Congress of Thermal Sciences and Engineering
November 16-20, 2020 (Online)

ENC-2020-0589
A STUDY OF DISCRETE ADJOINT-BASED GRADIENT METHOD IMPLEMENTATION FOR AERODYNAMIC OPTIMIZATION

Gilberto Bueno Luque Filho
Marco Aurélio Leonel Matunaga
Instituto Tecnológico de Aeronáutica, DCTA/ITA, São José dos Campos, SP, Brazil
gbueno.luque@gmail.com, marco.matunaga@gmail.com

João Luiz F. Azevedo
Instituto de Aeronáutica e Espaço, DCTA/IAE/ALA, São José dos Campos, SP, Brazil
joaoluiz.azevedo@gmail.com

Abstract. *The present work aims to perform an application of the implemented discrete adjoint optimization method to a wing optimization using low-fidelity aerodynamics computed with lifting line theory and an airfoil optimization in a transonic flow regime governed by the 2-D Euler equations. The first application was chosen for solving the optimization problem in a considerable short time, allowing exploration of coupled systems, validations and optimization method comparisons. The second one was selected to apply this knowledge to a problem that involves geometry parametrization, mesh deformation and non-linearities as the solution of the flow field has shock waves. The aerodynamic calculation, geometry parametrization and mesh deformation are computed by a Fortran code, which is, then, imported to the Python optimization process through a Python script and optimized using Scipy Python library. The results obtained are quite encouraging and demonstrate the applicability of the present approach.*

Keywords: *Adjoint method, Lifting line theory, 2-D Euler equations, Algorithmic differentiation, Gradient-based optimization*

1. INTRODUCTION

The aeronautical industry has always been characterized by strong competition due to the constant addition of new competitors arriving at the market and the searching for the best performance under stringent requirements. The heightening in competition leads the conceptual design process to handle an interdisciplinary and highly-coupled system with a large number of design parameters and constraints. Moreover, all of these analyses have to be performed within a feasible time.

Numerical optimization algorithms are, basically, branched in gradient-free methods, also known as zero-th order methods, and gradient-based methods (Matunaga and Azevedo, 2019). The first class of methods does not rely on any information other than the objective function values. For such approaches, computational time tends to increases exponentially with the increment of new design variables. Grid searching, genetic algorithms, neural networks and simulated annealing are good algorithmic examples of this approach. On the other hand, gradient-based methods not only use the objective function values, but they also use their gradients with respect to the design parameters. The main advantage of gradient methods is that they converge to the optimum with a significantly smaller number of function evaluations and some of the algorithms have a mathematical proof of convergence. Conjugate gradient, Newtons's method and Sequential Quadratic Programming method are examples of these methods. Nonetheless, besides the fact that such methods converge to a local minimum, which thankfully most of the times is close enough to the global minimum, one considerable disadvantage is the dependency on the computation of gradients and, hence, on how efficient this computation can be.

For gradient computations, a commonly used approach to calculate the sensitivities of the objective function and constraints is the finite-difference approach, which can be easily implemented but with a truncation error associated. The foundation of this approach is based on Taylor series expansions. Consequently, the cost of calculating sensitivities is proportional to the number of design variables as the function must be computed for every design parameter perturbation. In order to avoid the truncation error, the complex-step derivative approximation is considered, in which the real part of the function carries the function value and the imaginary part carries the derivative value. However, its cost still is proportional to the number of input variables once the foundation idea is also Taylor series expansions. Therefore, algorithmic differentiation or automatic differentiation comes up based on a systemic application of the chain rule for differentiation of each operation in the program structure. This approach is divided in two groups, forward and backward derivative propagations. The first one applies the chain rule in each line according to the previously computed variables, and the derivative of the first line of the program is computed with respect to the input variables. Therefore, the forward algorithmic differentiation computes the derivatives of the output of the program, by the chain rule, with respect to the inputs. Consequently, even though there is no truncation error, the cost is proportional to the number of input variables. For the reverse differentiation mode, the chain rule is computed from the outputs to the inputs, which means that each line of the code must be created for every program variable in other to propagate the outputs dependence on the inputs. Thus, differently from the aforementioned methods, the reverse mode has the total number of operations independent of the number of input variables. On the other hand, memory requirements may be prohibitive, especially for the case of large iterative algorithms.

Other approaches to analyse sensitivities, with more accuracy and efficiency, are the adjoint and direct methods. Between those, adjoint methods are particularly attractive since the cost of computing the gradient of a given function is independent of the number of design variables. Therefore, the calculation of the derivatives by solving an adjoint system is cheaper than solving a linear system using a direct method. For instance, the computational cost is comparable to the cost of obtaining one flow solution (Martins, 2002).

The adjoint approach can be treated as a continuous or as a discrete adjoint approach. The first one calculates the variation of the cost function and the field equations with respect to the flow-field variables and design variables through the use of Lagrange multipliers. Hence, the field equations and the adjoint equations, with their boundary conditions, must be discretized to obtain numerical solutions and the gradient is exactly computed, using integral by parts to derive an adjoint set of PDE's that propagates sensitivities. The discrete adjoint approach applies the control theory directly to the set of discrete field equations and the resulting equations depend on the type of scheme used to solve the flow equations (Nadarajah and Jameson, 2000). Therefore, if the discrete adjoint equations are solved exactly, the resulting solution for the Lagrange multipliers produces an exact gradient of the inexact cost function and the derivatives are consistent with finite difference gradients independent of the mesh size, deriving the adjoint equations to the discretized PDE equations.

The present work performs an optimization analysis by gradient and derivative-free methods. The derivatives are computed by the adjoint method and by finite differences. The sensitivity analysis necessary to build the adjoint system is constructed by the reverse algorithmic differentiation (Secco *et al.*, 2017). This process is performed by the Tapenade software (Hascoët and Pascual, 2013). The applications cover aerodynamic and aerostructural optimization using both a low-fidelity numerical method and the 2-D Euler equations. The low-fidelity numerical method is based on the lifting line theory for the aerodynamic section and 1-D finite element method for the structural analysis. A more challenging application covers an aerodynamics optimization of an airfoil within a transonic flow condition governed by the two dimensional Euler equations which, then, includes the consideration of flow non-linearities. The adjoint method is an important methodology towards industrial applications of aerodynamic shape optimizations with CFD solvers. Therefore, the present work aims to scientifically contribute to the dissemination and applications of discrete adjoint approach in Brazil.

2. METHODOLOGY

The computational program used can be divided in four sections. The first and second sections, both computed in Fortran, are associated with the aerodynamics module of lifting-line theory and two-dimensional Euler code, respectively. The third and fourth sections regard the solution of the adjoint system and the optimization routine itself.

2.1 Lifting Line Theory

The classical lifting-line theory, developed by Prandtl's in 1918, is based on applying the two-dimensional Kutta-Joukowski law to a three-dimensional flow. A modern adaptation of Prandtl's classic lifting line theory is utilized, which considers a fully three-dimensional vortex lifting law. These modifications enable applications for wings with an arbitrary camber, sweep, and dihedral angles (Phillips and Snyder, 2000).

The physics of the problem is contained in the residual equation, given by

$$R = 2\left[\left(v_\infty + \sum_{j=1}^{N} v_{ij} G_j\right) \times \zeta_i\right] G_i - C_{l_i}(\alpha_i, \delta_i), \tag{1}$$

where $v_\infty = \mathbf{V}_\infty / V_\infty$, and $\mathbf{V}_\infty$ is the velocity of the uniform flow, or freestream, and V_∞ is the magnitude of $\mathbf{V}_\infty$. Moreover, v_{ij} is the dimensionless velocity induced at control point j by vortex i, having a unit strength. G_{ji} is the dimensionless vortex strength for the j-th wing section, ζ_i the dimensionless spanwise length vector, C_{l_i} the section lift coefficient for the i-th wing section, α_i is the local angle of attack for the i-th wing section and δ_i is the flap deflection for the i-th wing section. Basically, the first term uses the circulation values to compute aerodynamic forces on every bound vortex. The second term computes the magnitude of the force given by the 2-D section, and the residual is the difference between the forces for each horseshoe vortex.

2.2 2-D Euler Equations

The flow is governed by the two-dimensional Euler equations, which are a compressible, rotational, inviscid and non-linear flow representation, assuming the hypothesis of a perfect gas state. In this way, it is completely possible to capture the presence of a shock waves present in a transonic flow. Due to the use of unstructured meshes and the finite volume method, these equations are written in Cartesian form. In addition, as it is usual in CFD applications, algebraic flow vectors are used and the equations are made dimensionless. In this way, the governing equations are given by

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0\,, \tag{2}$$

where Q is the vector of conserved quantities, given by

$$Q = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ e \end{array} \right\}, \tag{3}$$

and E and F are the flow vectors in the x and y directions, respectively, defined by

$$E = \left\{ \begin{array}{c} \rho u \\ \rho u + p \\ \rho v u \\ (e+p)u \end{array} \right\}, \tag{4}$$

$$F = \left\{ \begin{array}{c} \rho v \\ \rho u v \\ \rho u v + p \\ (e+p)v \end{array} \right\}, \tag{5}$$

where ρ is the density, u and v are the Cartesian components of velocity and e is the total energy per unit of volume.

In the present work, the flux vector interface values are calculated by averaging the conserved properties of the neighboring volumes and, then, computing the fluxes as a function of the averaged conserved variables, as proposed by Jameson and Mavriplis (1986). Moreover, as described in Pulliam (1986) and Jameson *et al.* (1981), centered discretization schemes naturally present problems of unmatched decoupling of the mesh points and oscillations in regions of large gradients and discontinuities, such as shock waves. Therefore, it is necessary to use artificial dissipation models that are added to the finite volume scheme. The implemented artificial dissipation terms are a combination of second and fourth differences with coefficients that depend on the local pressure gradient. This approach yields a cell-centered, second-order, finite volume scheme, which has the residue equation given by

$$R_i = C(Q_i) - D(Q_i)\,, \tag{6}$$

where C is the convective operator, D is the artificial dissipation operator and the i subscript represents the i-th cell.

2.3 Geometry Parametrization

The success of an optimization of the model problem depends on two critical steps, the choice of design variables and the cost function, also called merit function, f. In this section, the airfoil geometry itself is chosen to be the design variable. This can be translated in a computational domain as the contour grid points. Even though the cost of the adjoint approach is independent of the number of design variables, it is convenient to define a new set of design variables, based on continuous functions in order to gradient based optimization to work well, avoiding unsmooth geometry profiles due to high frequency modes contained in point-wise gradients computed. One such method of parameterizing the design space was presented by Hicks and Henne (1978), which is based on a formulation of a set of smooth functions, commonly referred to as "Hicks-Henne Bump Functions", that perturb the geometry profile. These bump functions are sinusoidal perturbations applied at different locations along the airfoil. A commonly used form is

$$b(x) = a\left[sin\left(\pi x^{\frac{log(0.5)}{log(t_1)}}\right)\right]^{t_2} \quad \text{for } 0 \leq x \leq 1\,, \tag{7}$$

where a is the bump amplitude, t_1 locates the maximum of the bump in $0 \leq x \leq 1$ and t_2 controls the width of the bump. Therefore, each bump is parameterized by three variables and for a complete parameterization, bumps must me specified for both upper and lower surfaces of the airfoil. From Eq. (7) it is possible to notice that indeed, there is no geometry parameterization as in NURBS, Becker *et al.* (2011) approach, instead, the perturbation is parameterized according to the sinusoidal functions, to the geometry.

2.4 Grid Perturbation

As aforementioned, the cost function choice, in a optimization process, is a crucial step. The cost function, generally, can be written as a function of state variables Q, the design variables, x and all grid points location χ. Therefore, the variation of the cost function can be expressed as

$$\delta f = \frac{\partial f^T}{\partial Q}\delta Q + \frac{\partial f^T}{\partial \chi}\delta\chi \; + \frac{\partial f^T}{\partial x}\delta x \; . \tag{8}$$

The solution of the adjoint equation removes the dependence of the gradient on the flow solution, so that only the variations of the grid point locations and the variation of the surface shape remain. Hence, in order to couple the variation of design variables on the surface and its contribution to the variation of the grid points location $\delta\chi$, the grid regeneration is needed for every surface perturbation. This procedure would have to be repeated a number of times proportional to the number of design variables, which has an enormous computational cost, mainly for three-dimensional grids.

In order to avoid these drawbacks, the adopted approach for this work is based on the grid perturbation method introduced by Jameson (1990), that modifies the current location of the grid points based on perturbations at the geometry surface, result from a change on design variables.

The method modifies the grid points along each grid index line projecting from the surface. Then the grid points at each location along the grid line is proportionally moved based on the arc length size of each point at the surface in relation to the arc length between the surface and the far-field point along the grid line. The algorithm can be described as

$$\left.\begin{aligned} {\chi_x}_{i,j}^{new} &= {\chi_x}_{i,j}^{old} + C_j \left({\chi_x}_{i,1}^{new} - {\chi_x}_{i,1}^{old}\right) \\ {\chi_y}_{i,j}^{new} &= {\chi_y}_{i,j}^{old} + C_j \left({\chi_y}_{i,1}^{new} - {\chi_y}_{i,1}^{old}\right) \end{aligned}\right\} \quad \text{for} \qquad i = I,\; j = 2, ..., j_{max} \; , \tag{9}$$

where χ_x and χ_y are the x and y grid coordinates, respectively. I is the current grid index. The vector C_j can be defined as follows

$$C_j = 1 - (3 - 2N_j)\, N_j^2 \; , \tag{10}$$

where N is the ratio of the arc length from the surface to the current grid point and the total arc length from the surface to the far-field along the grid line as

$$N_j = \frac{\sum_{l=2}^{j} \sqrt{({\chi_x}_{i,l} - {\chi_x}_{i,l-1})^2 + ({\chi_y}_{i,l} - {\chi_y}_{i,l-1})^2}}{\sum_{l=2}^{j_{max}} \sqrt{({\chi_y}_{i,l} - {\chi_y}_{i,l-1})^2 + ({\chi_y}_{i,l} - {\chi_y}_{i,l-1})^2}} \; . \tag{11}$$

From Eq. (9) the variation of the grid point location can be expressed as a function of the variation of the surface points as

$$\delta\chi = C_j \delta x \; . \tag{12}$$

Therefore, after a variation of a design variables, all surface grid points are also changed by the Hicks-Henne Bump Functions, and consequently this allows the variation of the grid point location in the equation for gradient evaluation, to be substituted with the variation of the surface points. The presented grid perturbation method described in this subsection has been found to be very robust Reuther (1996) and its robustness was verified by modifying two-dimensional viscous meshes Kim (2002). Moreover, even in regions of high non-linearity with large surface perturbations, the grid perturbation method was successful in producing smooth meshes without grid point cross-overs.

2.5 Aerostructural Analysis

2.5.1 General Formulation

For aerostructural analysis, one important metric regarding the performance of an airplane that can be considered as an objective function, is the fuel burn, which is how much fuel it uses for a given mission. It is possible to compute the fuel burn of an airplane flying for a time *T*, via the Breguet range equation which is given by:

$$FB = W_0 \left(1 - exp\left(-\frac{T \cdot SFC}{L/D}\right)\right) \; , \tag{13}$$

where W_0 is the initial weight of the airplane, SFC is the thrust-specific fuel consumption of the engine and L/D is the aerodynamic efficiency of the airplane during this flight phase. The aerodynamic efficiency (L/D) depends on which lift coefficient (C_L) the airplane is flying with. If we enforce level flight, which has a load factor $n_f = 1$ and $nf = \frac{L}{W_0}$. However, if lift is available but not used, there is a lift excess in the flight, given by:

$$\Delta L = \frac{L}{W_0} - 1\,. \tag{14}$$

The initial weight of an airplane can, basically, be separated into three contributions, which are the wing structure weight W_s, the fixed weight, W_f, of payload, systems, and the remaining structures and the fuel burn. Therefore,

$$W_0 = W_s + W_f + FB\,. \tag{15}$$

Consequently, Eq. (15) becomes:

$$FB = (W_s + W_f) \cdot \left(exp\left(\frac{T \cdot SFC}{L/D}\right) - 1 \right)\,. \tag{16}$$

Even though Eq. (16) consider a level flight condition, the structure should also support maneuvering conditions, which results in a load factor higher than one, and for commercial aircraft, regulatory agencies define the limit load factors that the structure should stand, and this value is usually around 2.5. One way to consider this load factor for structural analysis, is taking into account, n_f, in the safety margin, m, calculation of every single $i-th$ section of the wing, according to the following:

$$m_i = 1 - \frac{n_f \cdot \sigma_i}{\sigma_Y}\,, \tag{17}$$

where σ_Y is the yield stress of the beam material and σ_i is the local yield stress. The structure fails if $m_i < 0$, then, the smallest value of σ of the structure must be positive. Therefore, the aerostructural optimization should have a constraint on σ for every single wing element, which would increase the adjoint sensitivity computation cost and allow possible discontinuities. In order to concatenate these multiple constraints in a single composite function, the Kreisselmeier–Steinhauser (KS) function, by Kreisselmeier and Steinhauser (1979), is considered to be the aggregation constraint and, applied to Eq. (17), the (KS) margin function can be written as:

$$m_{KS} = -n_f \cdot \frac{1}{\rho_{KS}} ln\left(\sum_{i}^{n_p} e^{-\rho_{KS} \cdot m_i}\right) + 1 - n_f\,, \tag{18}$$

where ρ_{KS} controls the quality of the estimate, and improves the estimate of the minimum value by increasing its value.

The implemented aerostructural optimization analysis consists of four main blocks, the aerodynamics analysis, computed by the lifting line theory, described in section 2.1, the load transfer block, displacement transfer block and structural analysis block, which are briefly described in the next three subsections.

2.5.2 FORCES Block

The load transfer, or *FORCES* block, is a mapping of the aerodynamic load distribution over the span to structural forces on the wing. For simplicity, the aerodynamic and structural discretizations coincide, and the lift distribution along each bound vortex is converted into consistent forces and moments applied at the end nodes of each panel.

2.5.3 FEM Block

For the structural block, a 2 DOF per node finite element method (FEM) approach is implemented, assuming a beam only subject to vertical (bending) loads. The beam is represented by a circular beam and each element stiffness matrix can be descried as:

$$K_i = \frac{EI_i}{L_i^3} \begin{bmatrix} 12 & 6L_i & -12 & 6L_i \\ 6L_i & 4L_i^2 & -6L_i & 2L_i^2 \\ -12 & -6L_i & 12 & -6L_i \\ 6L & 2L_i^2 & -6L_i & 4L_i^2 \end{bmatrix}\,,$$

where E is the Young's modulus, I_i is moment of inertia, and the L is the length of a specific element i. For a circular beam, the moment of inertia can be computed as:

$$I_i = \pi t_i r_i^3\,, \tag{19}$$

where t_i is the thickness of the circular beam in the $i-th$ section and r_i is the radius of the beam, fixed by the airfoil section thickness.

2.5.4 DISPS Block

The displacements, computed by the finite elements analysis for a given input load distribution, must be transferred to the aerodynamics mesh in order to update the geometry and close the cycle of the fluid-structural problem. As previously discussed, the displacement is simply a Z coordinate update value, because only bending moments are taken into consideration and the aerodynamic and structural mesh discretizations coincide.

2.6 Adjoint System Solution

In the present discussion, the R variable refers to the residue of the corresponding problem for a matter of consistency with the previous discussion. The residue equations, *i.e.*, Eqs. (1) and (6), carry the physics of the problem and, in both cases, the residue depends on the design variables and on the state variables. For the lifting line theory, the state variable is the circulation Γ and, for the 2-D Euler equations, it is the Q array, represented in Eq. (3). In order to obtain the solution of the system, the main objective is to find a state variable in which the residue value is roughly zero. For the optimization, a merit function or cost function must be defined, which is also dependent on the design variables and state variables. If one considers the Euler equations as an example, the following can be written

$$\frac{d\vec{R}}{d\vec{x}} = \frac{\partial \vec{R}}{\partial \vec{x}} + \frac{\partial \vec{R}}{\partial \vec{Q}}\frac{d\vec{Q}}{d\vec{x}} = 0 \, , \tag{20}$$

$$\frac{df}{d\vec{x}} = \frac{\partial f}{\partial \vec{x}} + \frac{\partial f}{\partial \vec{Q}}\frac{d\vec{Q}}{d\vec{x}} \, , \tag{21}$$

where $\vec{Q}$ represents the state variables, $\vec{x}$ the design variables and f is the cost function. If one works with Eqs. (20) and (21), it is possible to write

$$\frac{df}{d\vec{x}} = \frac{\partial f}{\partial \vec{x}} \underbrace{- \frac{\partial f}{\partial \vec{Q}} \left[\frac{\partial \vec{R}}{\partial \vec{Q}}\right]^{-1}}_{\vec{\psi}} \frac{\partial \vec{R}}{\partial \vec{x}} \, , \tag{22}$$

where $\vec{\psi}$ is the Lagrangian multiplier, which can also be called the adjoint vector. Therefore, Eqs. (20) and (21) can be rewritten and the adjoint system can be constructed as

$$\left[\frac{\partial \vec{R}}{\partial \vec{Q}}\right]^{T} \cdot \vec{\psi} = - \left[\frac{\partial f}{\partial \vec{Q}}\right]^{T} \, , \tag{23}$$

$$\frac{df}{d\vec{x}} = \frac{\partial f}{\partial \vec{x}} + \psi^{T}\frac{\partial \vec{R}}{\partial \vec{x}} \, . \tag{24}$$

Assuming that a certain code routine, named here $main$, for the sake of discussion, has the design variables, $\vec{x}$, and state variables, $\vec{Q}$, as inputs and the cost function, f, and residual, $\vec{R}$, as outputs, the reverse algorithmic differentiation can be computed by Tapenade, resulting in a routine called $main_b$. The b underscore refers to backward differentiation or reverse differentiation. The following system is the result of this process:

$$\vec{Q}_b = \left[\frac{\partial \vec{R}}{\partial \vec{Q}}\right]^{T} \cdot \vec{R}_b + \left[\frac{\partial f}{\partial \vec{Q}}\right]^{T} \cdot f_b \, , \tag{25}$$

$$\vec{x}_b = \left[\frac{\partial \vec{R}}{\partial \vec{x}}\right]^{T} \cdot \vec{R}_b + \left[\frac{\partial f}{\partial \vec{x}}\right]^{T} \cdot f_b \, , \tag{26}$$

where the variables with the b underscore are the inputs of the created routine, $main_b$. Therefore, in order to obtain the adjoint system, Eqs. (23) and (24), the following relations must be satisfied

$$\vec{Q}_b = 0 \, , \tag{27}$$

$$\vec{R}_b = \vec{\psi} \, , \tag{28}$$

$$f_b = 1 \, . \tag{29}$$

Therefore, $\vec{Q}_b$ can be called the residue of the adjoint equation due to the fact that the system solution is the adjoint array, given by Eq. (28).

2.7 Scipy Minimization

The sequential least-squares quadratic programming (SLSQP), developed by Kraft (1994), is an iterative method for solving nonlinear optimal control problems subjects to a group of constraints, where both objective and constraint functions are twice continuously differentiable. On the other hand, the SLSQP method, which solves a sequence of optimization sub-problems at each time step, is suitable for large-scale optimization problems (Nocedal and Wright, 2006). Therefore, in the present work, the SLSQP minimization algorithm from the Scipy (Scientific Computing in Python) Python library (Virtanen *et al.*, 2020) is used. The selection of the Scipy Python library is related to its robustness and capability of dealing with constrained functions. Moreover, the use of this library requires that the design variables are bounded and that the minimization of Eq. (24) is actually being computed, which is precisely the case in the present effort.

3. RESULTS

3.1 3D Aerodynamic Optimization

The lifting line theory (LLT) flow calculation is based on potential flow. Consequently, for a 3-D wing, the obtained drag is only the induced drag since no viscous effects are taken into consideration. However, if the aerodynamic optimization aims to minimize drag, without constraints or limitations on the design variables, the optimizer would return an unrealistic wing geometry. Therefore, in order to avoid such problem, some form of constraint must be imposed in the optimization process. In the present case, a given value for the lift coefficient is imposed as a constraint and, in the optimization process, this constraint must be respected. Therefore, with the twist angle of each panel as the design variables and the lift coefficient as a constraint, the result will be a realistic geometry. The optimization problem statement is summarized in Tab. 1.

Table 1: Aerodynamic optimization problem using LLT; n_p is the number of panels.

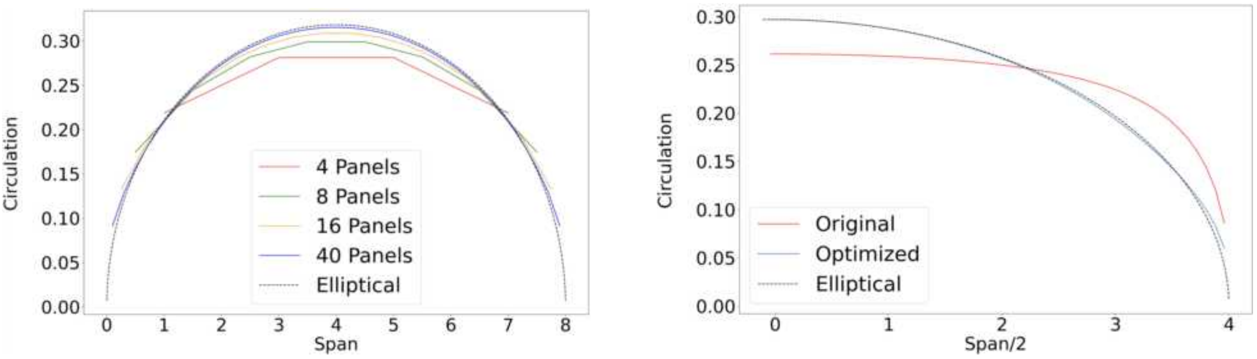
	Variable/function	**Description**	**Quantity**
minimize	C_D	Induced Drag	
with respect to	$\alpha_{0,i}$	Twist angle of panels [deg]	n_p
subject to	$C_L = C_{L_{target}}$	Lift coefficient constraint	1

The optimization is obtained for 4, 8, 16, 40 and 100 wing panels, for mesh refinement purpose and in order to have a better comparison between the gradient-free and the gradient approach. For the latter case, the comparison is also envisioned between the adjoint and finite differences methods for derivative calculations. Moreover, this analysis can show that, the cost of computing the gradients, by the adjoint method, is independent of the number of design variables. Table 2 gives some important comparisons between the initial and optimized wing geometries. It should be clear that all drag coefficient results, reported in Tab. 2, that is, for both the original and the optimized wings, have satisfied the lift coefficient constraint. The results in the table also indicate that the optimized wing has a 6% decrease in the drag coefficient for this flight condition, when compared to the original wing. Moreover, as the number of panels in the wing is increased, the section lift distribution along the wing, or the circulation distribution along the wing, tends to the elliptical distribution and, hence, the computed drag coefficient approaches the value of the theoretical drag coefficient for the elliptical wing, which, in theory, the minimum induced drag possible. Although the calculations are not shown here, the value of C_D shown in Tab. 2 for the 100-panel wing discretization is very close to the drag value for the elliptical wing.

Table 2: Comparison between initial and optimized wing parameters for 100 panels.

Parameter	**Initial Wing**	**Optimized Wing**
Planform	Rectangular	Rectangular
Γ distribution	non elliptical	elliptical
C_L	0.467	0.467
C_D [drag counts]	91.66	86.00
C_L/C_D	50.94	54.30

Figure 1 presents the circulation distribution along the wing span. The effect of the number of panels used in the calculation with the LLT method is indicated in Fig. 1a, which also shows the elliptical circulation distribution for comparison purposes. Clearly, the numerical errors are reduced with the increasing number of panels and, hence, the optimized wing, with the appropriate variation of the section twist angles, moves towards the elliptical one. Figure 1b presents the circulation distribution along the semi-span for the original wing in comparison with the distribution obtained for the optimized wing. Calculations in this case have used 100 panels. Again, for comparison purposes, the circulation distribution for the elliptical wing is also shown. As already pointed out, the differences between the elliptical circulation distribution and that of the optimized wing are minimal.

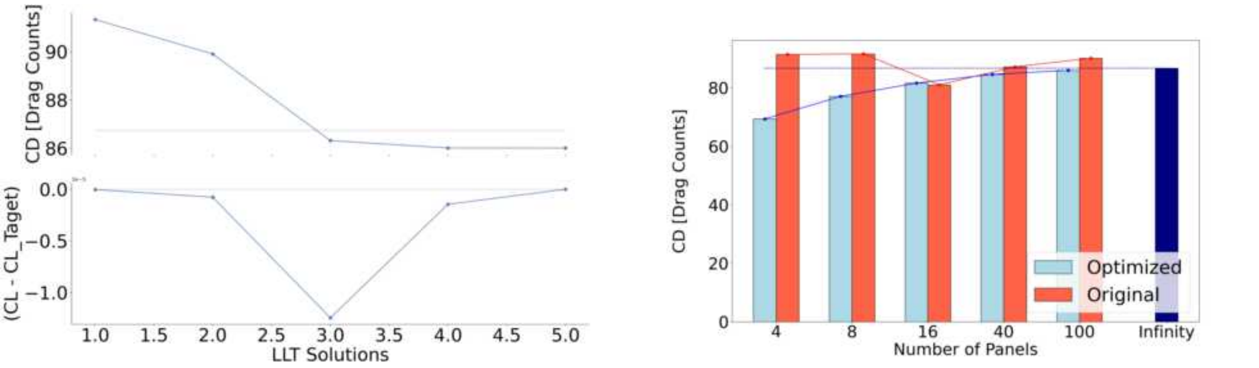


(a) Effect of the number of panels in the calculation.

(b) Comparison of original, optimized and elliptic wings.

Figure 1: Comparisons of the optimized circulation distributions as a function of the number of panels and with respect to the elliptical distribution.

Figure 2 attempts to demonstrate some aspects of the computational result convergence as the design cycles progress and, also, as a function of the discretization used to represent the wing. Hence, Fig. 2a presents the optimization convergence history for both the objective and constraint functions. For the drag convergence graph, the light gray line represents the elliptic wing drag value and the blue line indicates the induce drag coefficient values calculated at each iteration step of the design cycle. The lower curves in this plot are indicating whether the lift coefficient constraint is being satisfied at each design cycle iteration. The light gray indicates the situation in which the constraint is satisfied and the blue line is the actual difference between the current lift coefficient and the target one. One can see that, during the iteration process, the lift constraint may not be satisfied but, as the iterations converge to the optimized wing, the constraint function is again satisfied. Figure 2b shows the evolution of the drag coefficient values as the spatial discretization of the wing is refined. The figure includes results for both the original and the optimized wing. One can see that, for coarse meshes, numerical errors might lead to some inconsistencies in the comparison of drag coefficients for original and optimized wings. However, as the number of wing panels is increased, the optimized solution yields drag coefficient values which are consistently lower than those provided by the original wing.



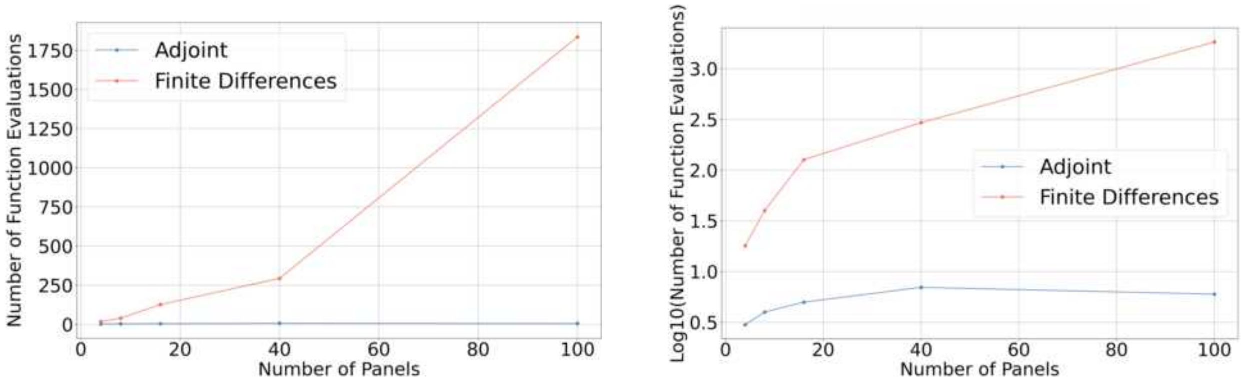
(a) Convergence history of objective and constraint functions, 100 panels.

(b) Drag evolution with mesh refinement: 4, 8, 16, 40 and 100 panels.

Figure 2: Convergence indicators for the optimization process.

The computational cost of the optimization process can be measured by the number of times the objective function is evaluated, which is known as the number of function evaluations. Therefore, this quantity is used to compare the computational costs of both adjoint and finite difference approaches. This comparison is shown in Fig. 3. The results in this figure are presented in two different ways, namely, with a linear plot and with a logarithmic plot. Figure 3 is essentially demonstrating that, for the finite difference approach, the cost of calculating sensitivities is proportional to the number of design variables and, hence, for a more complex problem, the costs associated with such sensitivity evaluation could be impracticable. On the other hand, the strongest benefit of the adjoint approach is also depicted in Fig. 3, which has roughly 4 orders of magnitude less computational cost when compared to the finite difference approach for the 100-panel discretization in the present problem.

Figure 4 depicts how the optimizer decided, based on the computed gradients, to change the design variables in order to satisfy the lift coefficient constraint and minimize the drag coefficient. In order to facilitate the visualization of the

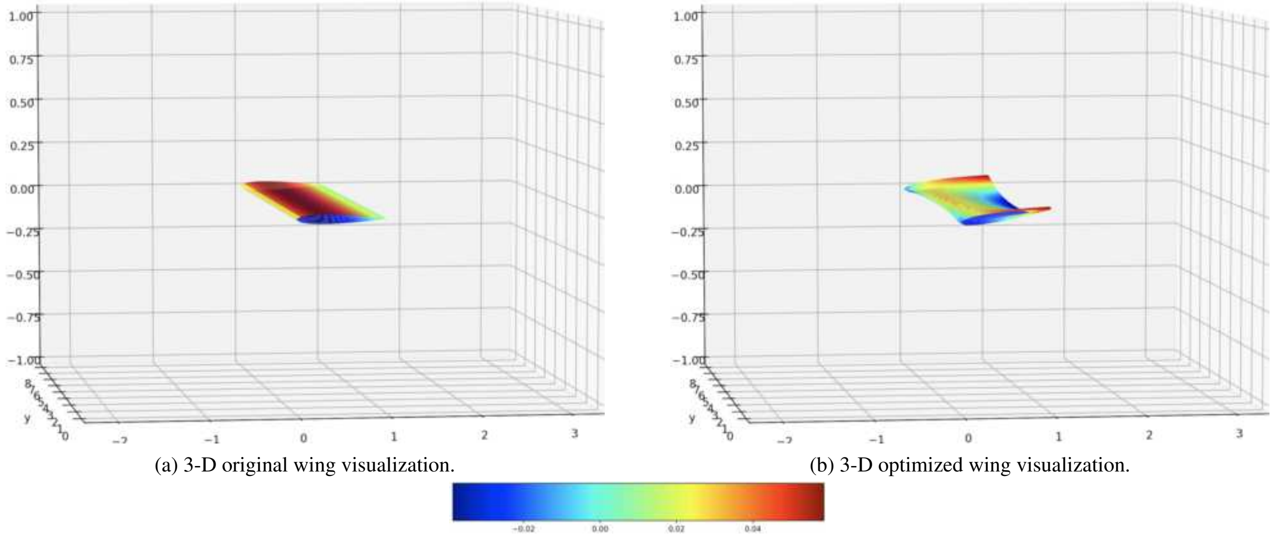


(a) Linear representation of the number of function evaluations.

(b) Logarithmic visualization of the number of function evaluations.

Figure 3: Comparison of the number of function evaluations for the adjoint and finite difference approaches.

results, a symmetric airfoil with small thickness is considered for the visualization of the outputs, even though the method does not take into account the thickness of the airfoil. The flight angle of attack is imposed as 5 deg. and, in order to reduce the induced drag, lift must be reduced mainly at the wing tips. Therefore, Fig. 4 presents the z-displacements of the 3-D wing in a surface plot in order to attempt to simplify the understanding of the twist angle changes due to the optimization process.



(a) 3-D original wing visualization.

(b) 3-D optimized wing visualization.

(c) Z coordinate color bar.

Figure 4: Visualization of the twist angle in the original and optimized wings.

3.2 3D Aerostructural Optimization

The aerostructural optimization is a great example of a multidisciplinary design optimization (MDO) problem, because it is possible to combine, aerodynamics, structural and performance design variables to minimize a specified merit function. In this present work, as previously point out, the fuel burn equation Eq. (16), concatenates those disciplines and is chosen to be minimized, with some structural and aerodynamics constraints. The aerostructural optimization problem statement is summarized in Tab. 3.

For the optimization, the fuel burn of a non-electric unmanned aerial vehicle (UAV) is considered for minimization. The requirements and problem restrictions are in Tab. 4.

Figure 5 depicts the change on weight components comparison between the original and optimized wing. Obviously, there is no change in the fixed weight component. On the other hand, the total weight of the wing is reduced in 1.90%, resulting from a reduction of the structural weight and 12.61% less fuel burn.

The structural point of view of this optimization problem regards the static margin calculation. From Fig. 6 is possible to see that the original wing was not a feasible one, once the aggregation spare failure condition was violated at the middle of the wing where highest loads are acting. After 83 adjoint iterations of the *AeroStructural Analysis* (ASA) framework,

Table 3: Aerostructural optimization problem using LLT and FEM; n_p is the number of panels.

	Variable/function	Description	Quantity
minimize	FB	Fue burn	
with respect to	$\alpha_{0,i}$	Twist angle of panels [deg]	n_p
	$t_i > 0.001$	Beam thickness distribution [m]	n_p
		Total design variables	$2n_p$
			.
subject to	$liftExcess = 0$	Lift constraint	1
	$KSmargin \geq 0$	Structural feasibility	1
		Total constraints	$2n_p$

Table 4: Flight conditions, performance, structural and aerodynamics characteristics

Parameter	Description	Value
AR	Aspect Ratio	10.0
S_{ref}	Wing reference area	16.0 m^2
α	Flight angle of attack	5.0 deg
V_∞	Cruise velocity	40.0 m/s
ρ_{air}	Air density	1.225 kg/m^3
E	Young Modulus	73.1e9 Pa
σ_Y	yield stress	324.0e6 Pa
ρ_{mat}	Material density	2780.0 kg/m^3
CD_0	zero-lift Drag	0.0270
T	Endurance	4.0 hours
SFC	thrust-specific fuel consumption	1.36e-4 s/m
n_f	Load Factor	4.5

the optimizer obtained a feasible wing where each section has a positive structural margin, with a lower total weight, 12.61% less fuel burn and 21.67% more lift than the original configuration.

In order to obtain the aforementioned results, the optimizer changed the design variable, twist angle and beam thickness of the spar as shown in Fig. 7. Aerodynamic considerations dominate the design of this specific wing, so the fuel burn minimization problem, results in a closer to elliptical lift distribution. The optimized beam thickness is greatest at the root and gradually decreases as it approaches the tip, as expected. Furthermore, to avoid structural failure, it twists down the outboard section of the wing to unload the tip . Consequently, the optimized wing twist is positive for most of the wing, except at the tip to produce a more aerodynamically efficient wing, also reducing the local angle of attack to minimize induced drag. As the structures does not failure, the banding of the optimized wing is considerably less severe and its behavior is smoother. A clear geometry comparison between the original and the optimized wing can be seen in Figs. 8 and 9.

The optimization, as mentioned before, was obtained in 83 *ASA* iterations and the convergence history of the objective function, and lift excess and aggregated static margin, *KSmargin*, constraints functions can be seen in Fig. 10.

3.3 2D Transonic Airfoil Optimization

3.3.1 Code Validation

For the transonic code validation, experimental data Kroll and Jain (1987) is used as reference for three different mesh refinements for the NACA0012 profile, which are detailed in Tab. 5. The validation cases cover Mach 0.80 with 0.00 angle of attack and Mach 0.85 with 1.00 angle of attack.

Table 5: Mesh refinement details.

Mesh	Number of ξ nodes	Number of η nodes	Total number of nodes
$1n$	101	33	3333
$2n$	201	66	13266
$3n$	301	99	29799

The results agree well with the literature data. Moreover, even for the coarse mesh, the shock capture was adequate, showing that the implemented solver is suitable for the optimization process and inverse design problem.

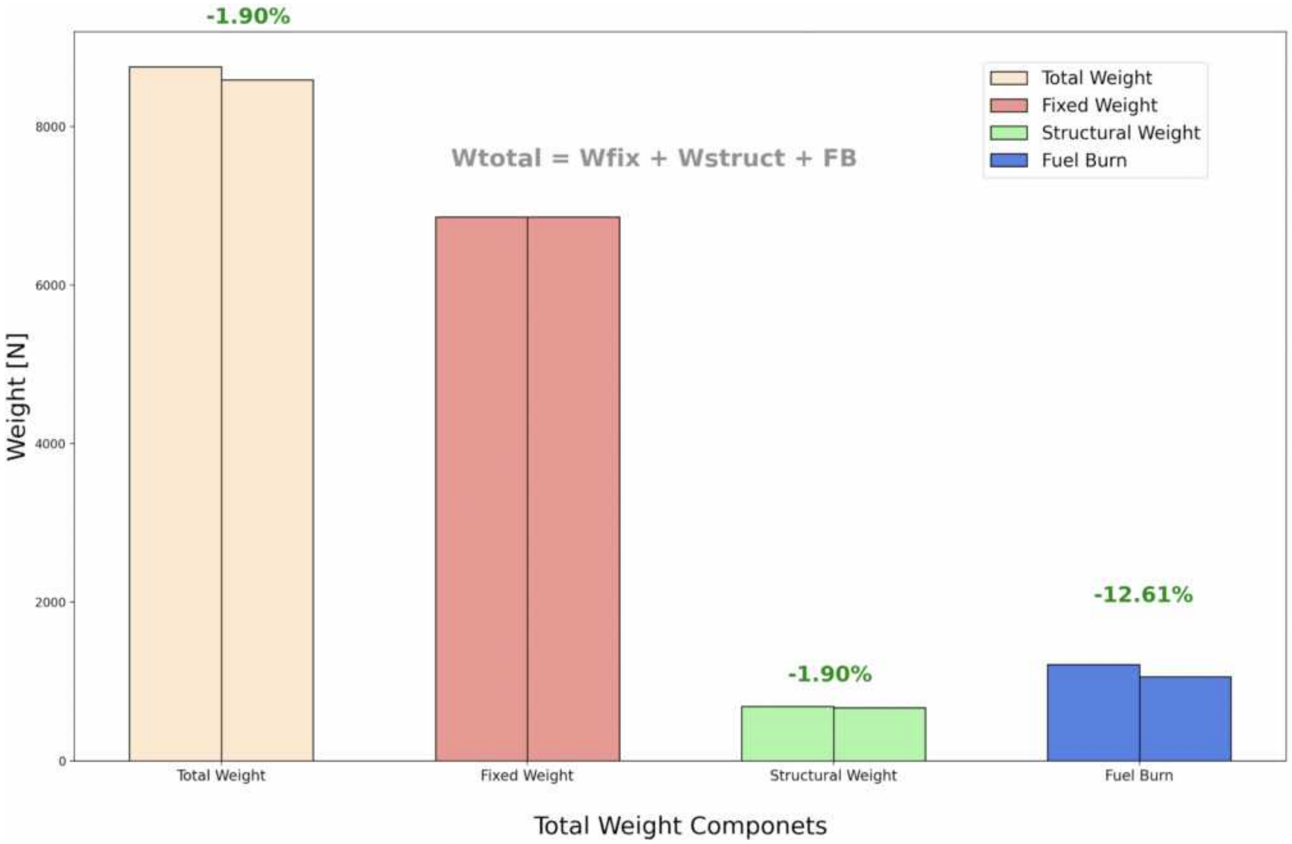


Figure 5: Weights components percentage reduction.

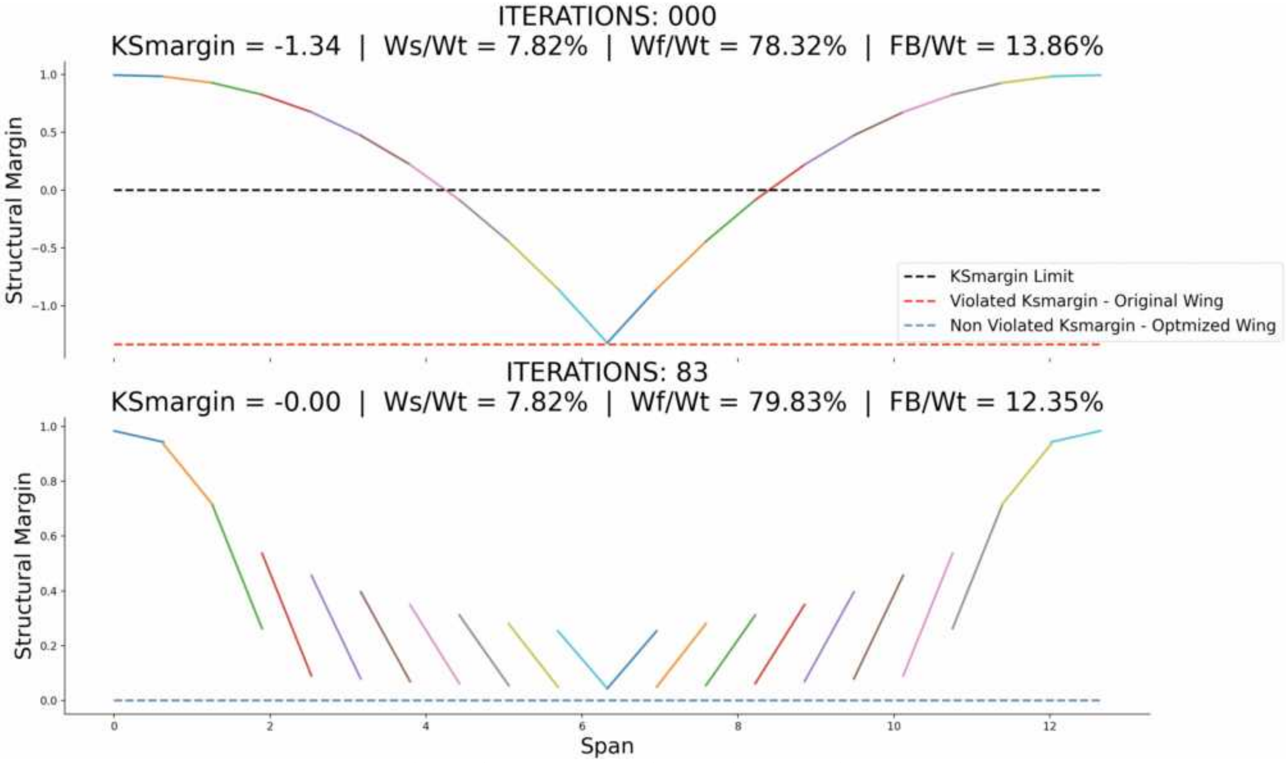


Figure 6: Structural static margin comparison.

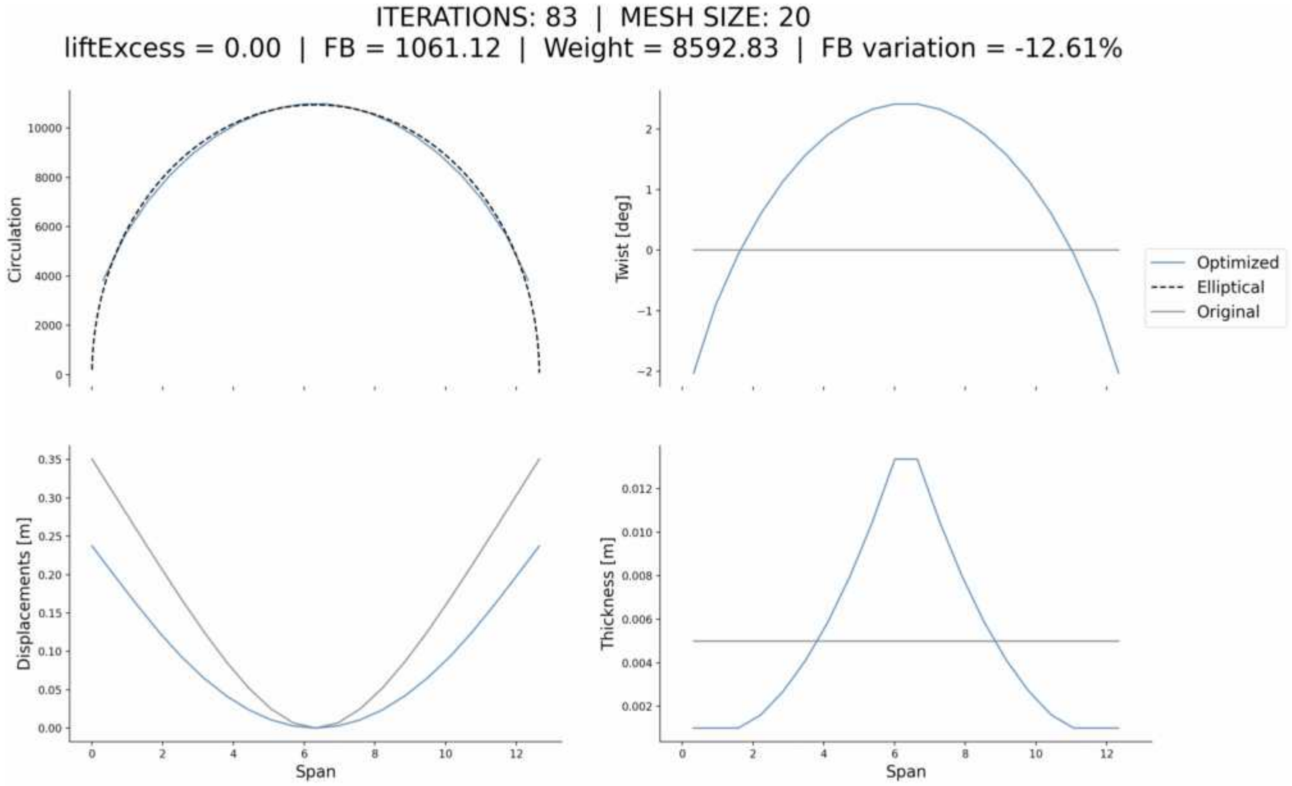


Figure 7: Design variables, aerodynamics and geometry characteristics comparison.

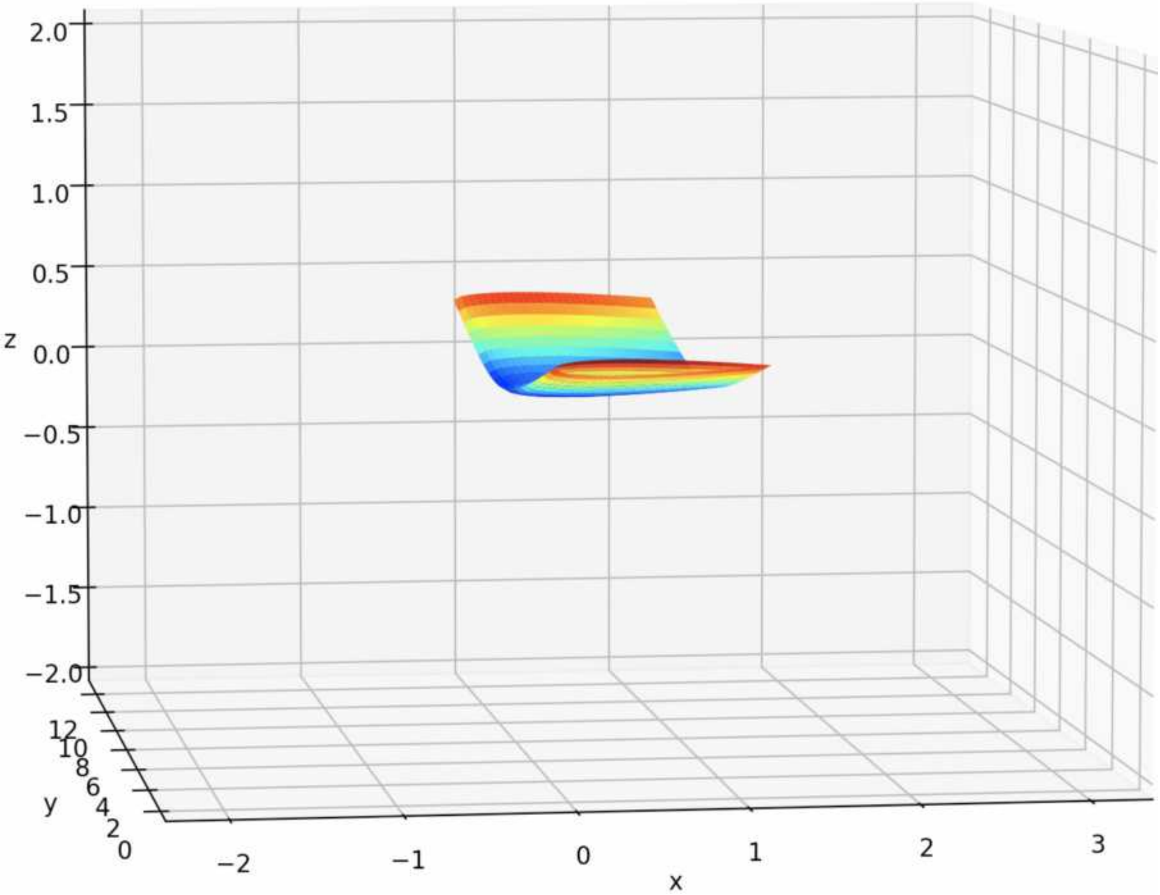


Figure 8: Original wing.

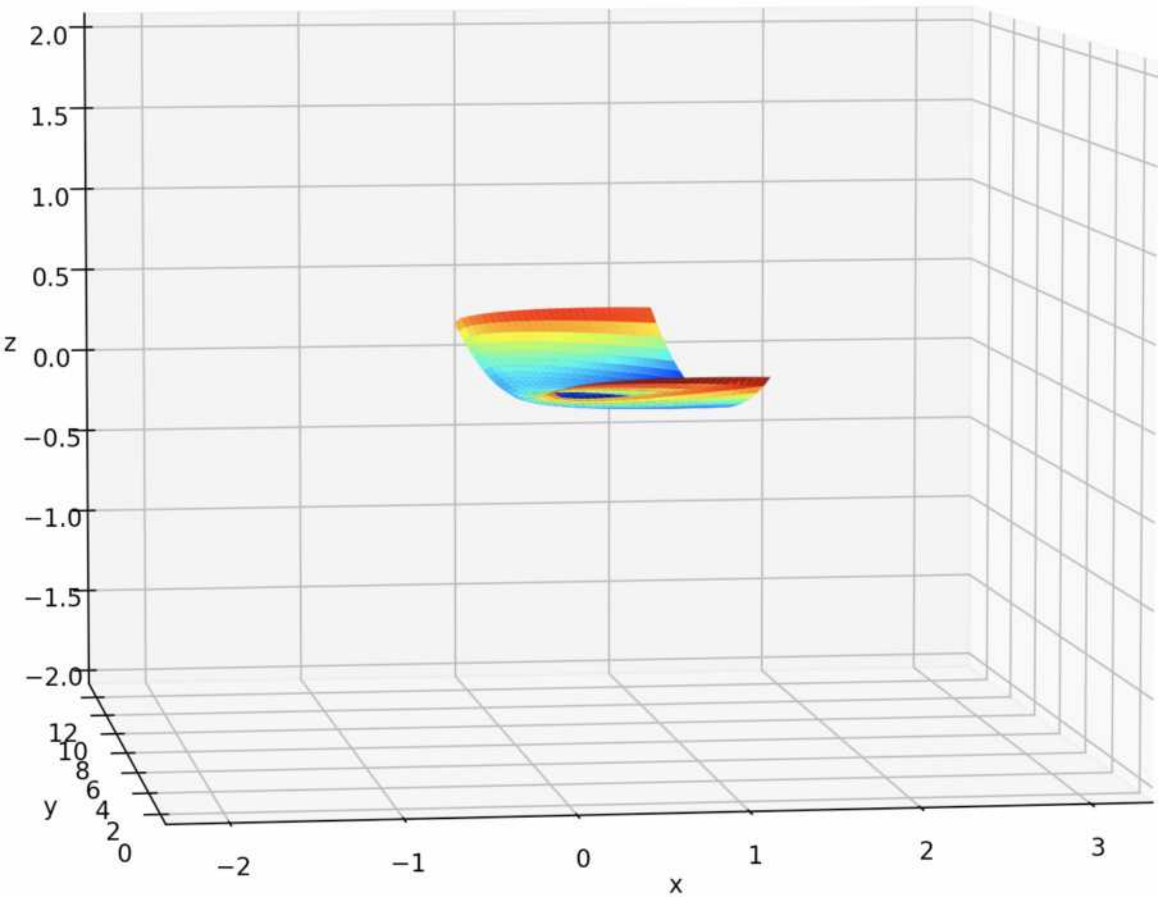


Figure 9: Optimized wing.

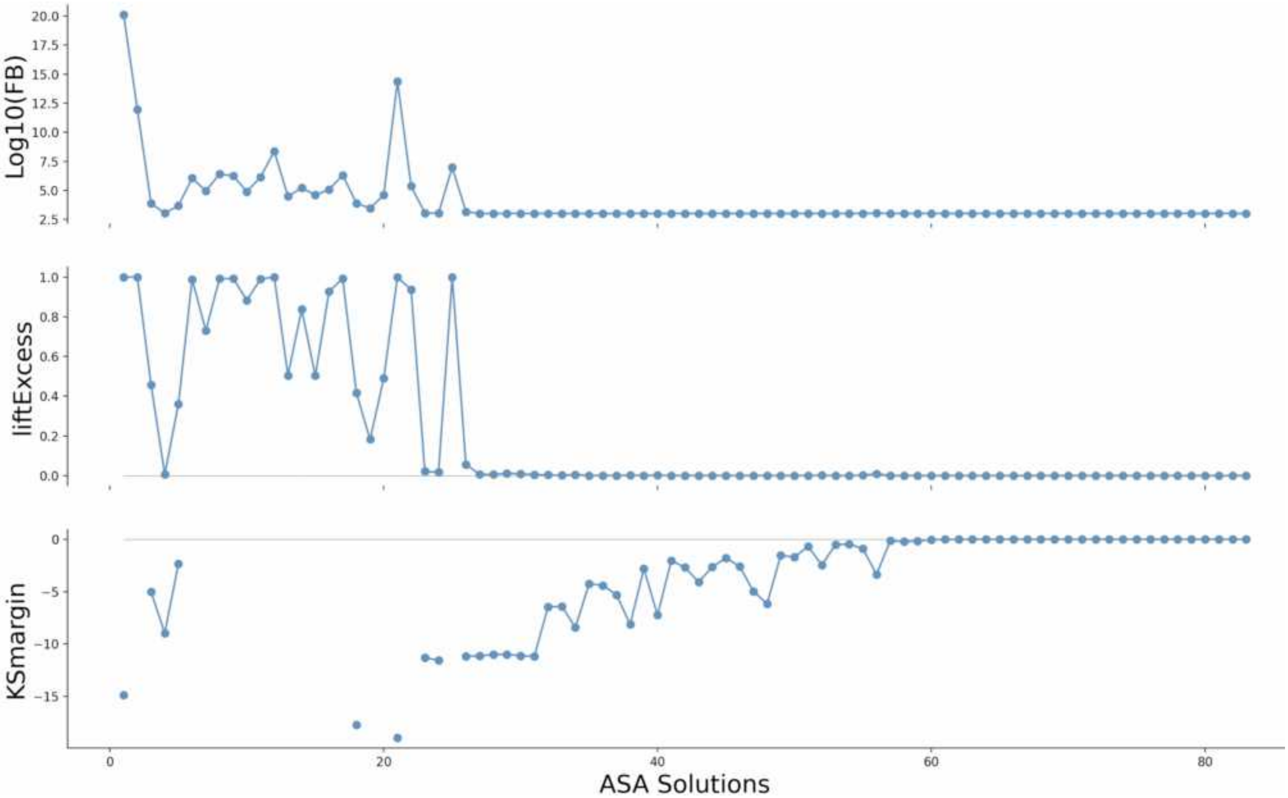
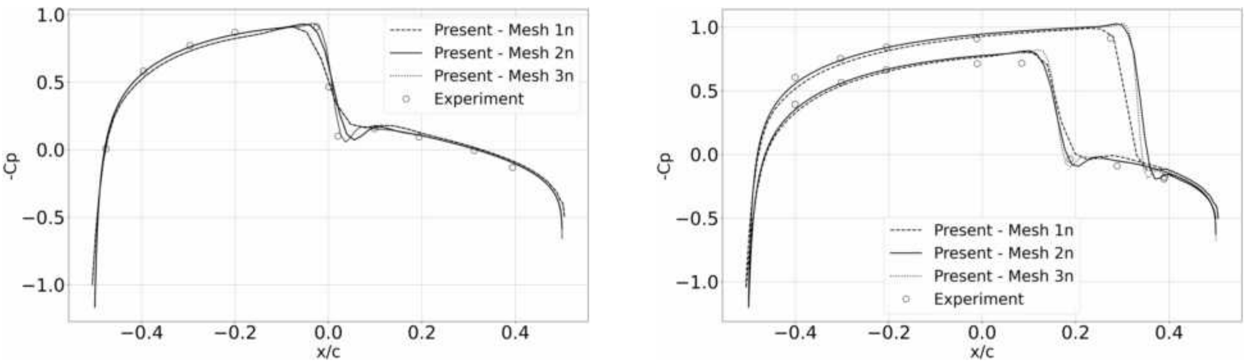


Figure 10: Convergence history of objective and constraint functions.



(a) Mach 0.80 and 0.00 of angle of attack.

(b) Mach 0.85 and 1.00 of angle of attack.

Figure 11: Validation cases and mesh refinement evaluation.

3.3.2 Inverse Design

The inverse design methods compare the distribution of a physical quantity of interest to a desired distribution, previously specified by the user as the pressure coefficient, for example. Based on this comparison, these methods estimate changes in geometry that would produce the desired effect. In this way, it is possible, among other things, to avoid adverse pressure gradients that would induce premature separation of the boundary layer in flows around airfoils.

In inverse problems, there is a possibility that the desired pressure distribution may not be achievable. This difficulty can be overcome by treating the problem as a particular case of an optimization problem, with a merit function that measures the error in solving the inverse problem. Thus, the process must lead to the realizable solution closest to the desired one, in the sense of least squares. Thus, a problem that is possibly non well-posed becomes a well-posed problem Jameson (2003). For example, if Cp_d is the desired surface pressure coefficient, a mathematical formulation of a cost function f, can be obtained as the integral:

$$f(x) = \oint_{airfoil} \frac{1}{2}(Cp - Cp_d)^2 \, ds \, , \tag{30}$$

where Cp is the resulting pressure coefficient distribution along the airfoil. For an airfoil defined by a set of N discrete grid points over the surface, this would result in a pressure for each X-coordinate, which, for convenience, forced to be the same by a spline function. Therefore, the cost function can be written as:

$$f(x) = \sum_{i=0}^{N} \frac{1}{2}(Cp_i - Cp_{d,i})^2 \, ds \, , \tag{31}$$

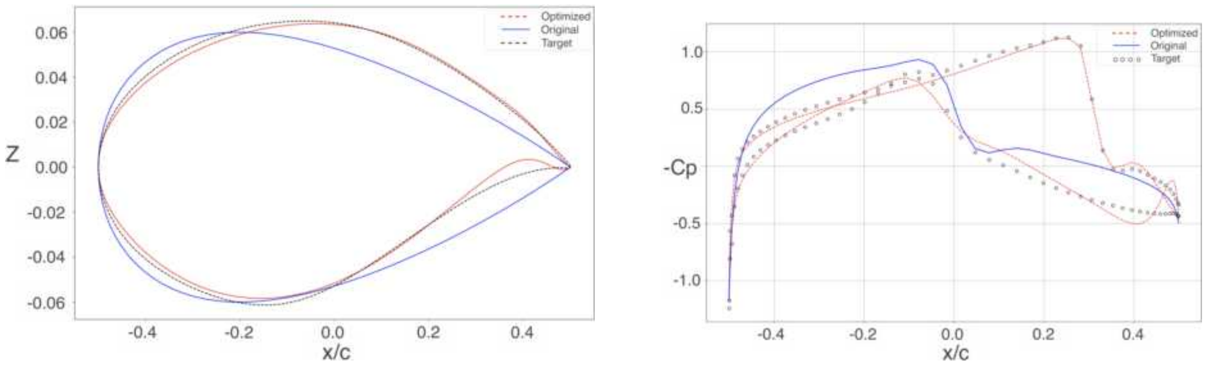
Therefore, for the inverse design problem, the target pressure coefficient distribution is obtained by the known RAE 2822 airfoil. The main objective is to minimize the cost function, Eq. (31), starting with a NACA 0012 airfoil, and obtain as output, both geometry and pressure coefficient distribution of RAE 2822.

A preliminary design case was chosen with 8 design variables, with 4 Hicks-Henne bumps on both the upper and lower surface of the airfoil, in 0.20%, 0.40%, 0.60% and 0.80% chord, along the airfoil from leading edge to trailing edge. The only varying parameter is the amplitude of each bump, resulting in 8 design variables. The initial, desired and optimized airfoils are represented by a continuous blue line, dashed black line and continuous red line, respectively, in Fig. 12a. Similarly, The initial, desired and optimized pressure coefficient distributions are depicted in continuous blue line, dotted black points and dashed red line, respectively, in Fig. 12b. Moreover, Fig. 13 shows a comparison between the adjoint and Nelder-Mead approach (Nelder and Mead, 1965), on the convergence result plotted against design iteration.

Figure 12 indicates that, the sensitivities computed with adjoint method, points the solution towards the desired geometry and Cp_d distribution, highlighting for the correct shock movement capture. However, the Hicks-Henne bump functions has limitations due to the fact that at the edge of each bump, the design variable movement is severally smoothed. Consequently, the trailing edge of the airfoil does not parametrize correctly the design variable displacement.

Figure 13 depicts one of the main benefits of gradient methods. The same output were obtained with 9 and 1000 iterations, with adjoint and Nelder-Mead, respectively. As aforementioned, gradient-free methods increase exponentially with the number of design variables, while the cost of computing the derivatives through the adjoint method are independent of the number of design variables.

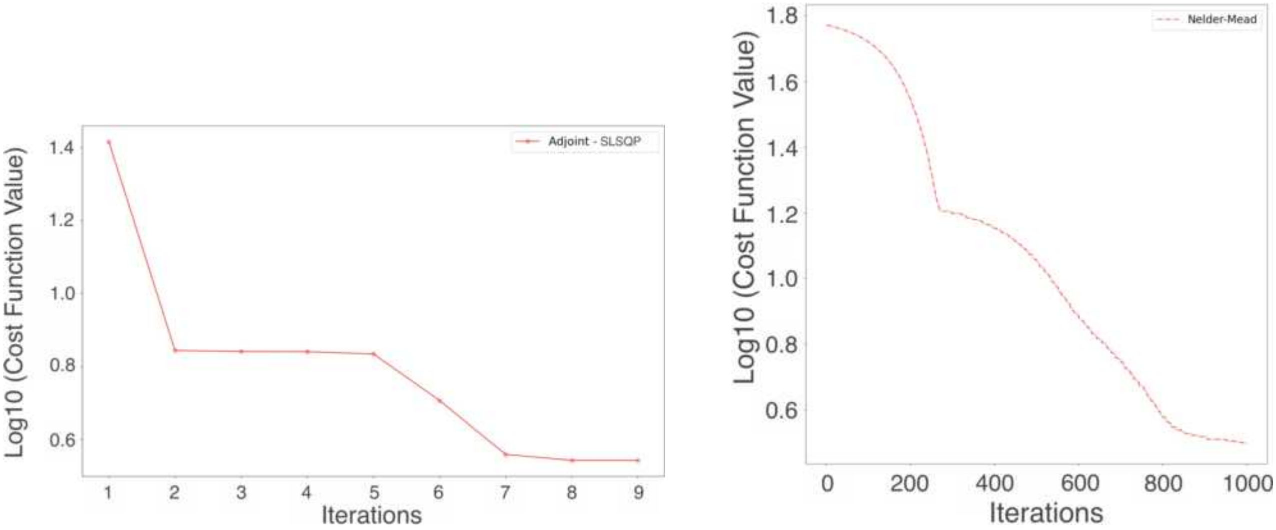
Figures 14b and 14c present the same inverse design problem as in Fig. 13, but with 30 design variables. A better matching with the desired geometry and pressure distribution was obtained in 46 iterations. Besides the number of iterations increased, the time spent by the gradient algorithm for each cycle of computation is roughly the same as with 8 design variables, as expected.



(a) Airfoil geometry evolution.

(b) Pressure coefficient evolution.

Figure 12: Airfoil optimization evolution with 8 design variables.



(a) Convergence history of adjoint optimization.

(b) Convergence history of Nelder-Mead, gradient free, approach.

Figure 13: Convergence history of cost function.

From both figures, it is also possible to observe that there is still a small difference between the optimized and the desired profile. It is understood that this fact is a result of the limitations of the parameterization itself, which mathematically may not contain the desired geometry as a solution. Therefore, adding a new parameterization in the present optimization process could improve the present work and bring certain gains for more challenging applications.

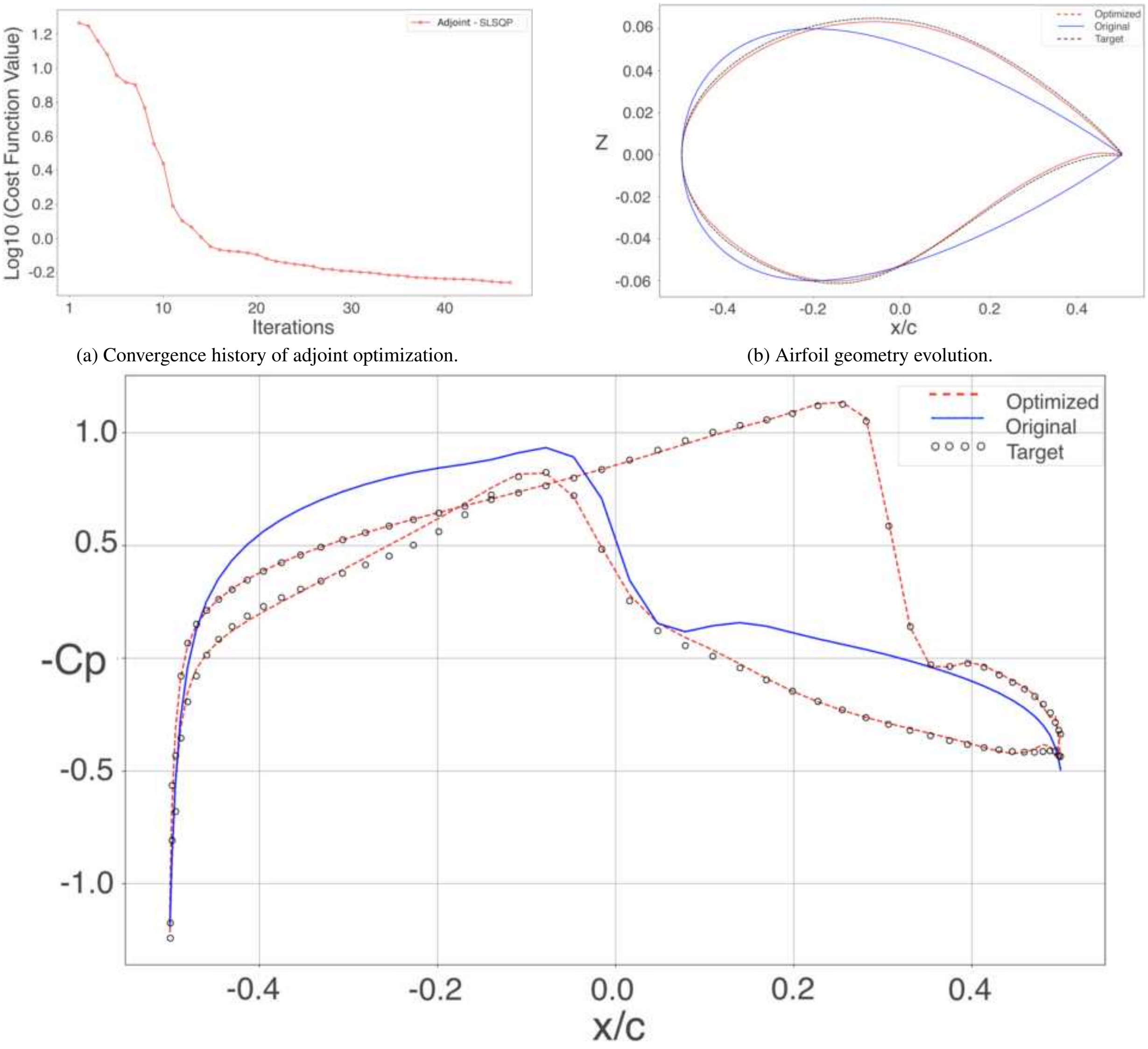
4. CONCLUDING REMARKS

The results presented in this work have addressed the optimization studies based on the lifting-line theory (LLT) for aerodynamic and aerostructural optimization, and inverse design problems based on the two-dimensional Euler equations. All the gradient computations necessary for the optimizations were carried out by the discrete adjoint method.

For both aerodynamic and aerostructural optimization cases, the results are consistent with the problem physics and optimal solutions found corroborate the functionality of the process. These results were obtained with a considerable lower computational cost, when compared to the calculation of gradients by finite difference as can be seen by the aerodynamic optimization results in Fig. 3. In the case of the aerostructural problem, even with low fidelity physics and also low computational cost, with 40 design variables and 2 constraints, the evaluation by finite differences appears to be prohibitive in terms of computational costs. Therefore, due to the implementation of the adjoint method, the aerostructural optimization process could be carried out in a personal computer within a short period of time. Consequently, a better understanding of the critical design points in a fluid-structure problem could be studied.

In order to evaluate the application of the adjoint method in problems with inherent nonlinearities, the inverse design of an airfoil in transonic flow regime, governed by the Euler equations, was adopted. The use of the inverse design allows the elimination of systemic errors from the solver and enables the identification of problems in the parameterization, mesh deformation and adjoint calculation packages.

The results reinforced the large advantage, in terms of computational cost, of gradient methods, compared to gradient-free methods. The effort here undertaken has indicated that the parameterization has some limitations that would be



(a) Convergence history of adjoint optimization.

(b) Airfoil geometry evolution.

(c) Pressure coefficient evolution with 30 design variables.

Figure 14: Airfoil optimization evolution with 30 design variables.

undesirable for more challenging applications and these should be addressed in future work. The methodology presented in this study was able to provide results in problems which involve nonlinearities such as shock waves. Moreover, the objectives of the studies here presented have been fully achieved.

The calculations have used a Fortran code for aerodynamics analysis, the reverse algorithmic differentiation and the adjoint module. The optimization itself has been treated with a Python script. In future work, more parameterization schemes will be implemented for the transonic adjoint framework. Moreover, the ultimate objective is to extend the adjoint knowledge to a high-fidelity adjoint-based aerodynamic and aerostructural optimization governed by the three-dimensional Navier-Stokes equations.

5. ACKNOWLEDGMENTS

The authors wish to express their gratitude to Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, which has supported the present research through a graduate scholarship for the second author. The authors also gratefully acknowledge the support for the present research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grant No. 309985/2013-7. The work is further supported by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under the Research Grant No. 2013/07375-0.

6. REFERENCES

Becker, G., Schäfer, M. and Jameson, A., 2011. "An Advanced NURBS Fitting Procedure for Post-Processing of Grid-Based Shape Optimizations". *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*.

Hascoët, L. and Pascual, V., 2013. "The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification". *ACM Trans. Math. Softw.*, Vol. 39, pp. 20:1–20:43.

Hicks, R.M. and Henne, P.A., 1978. "Wing Design by Numerical Optimization". *Journal of Aircraft*, Vol. 15, No. 7, pp. 407–412.

Jameson, A., 1990. "Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag". In *In Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics*. Uberlandia, MG, Brazil, pp. 5–17.

Jameson, A., 2003. "Lectures at the Von Kármán Institute, Brussels: Aerodynamic Shape Optimization Using the Adjoint Method".

Jameson, A. and Mavriplis, D., 1986. "Finite-Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh". *AIAA Journal*, Vol. 24, No. 4, pp. 611–618.

Jameson, A., Schmidt, W. and Turkel, E., 1981. "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes". In *AIAA 14th Fluid and Plasma Dynamics Conference*. AIAA Paper No. 81-1259, Palo Alto, CA.

Kim, S., 2002. *Design Optimization of High-Lift Configurations Using a Viscous Adjoint-Based Method*. Ph.D. thesis, Stanford University, Stanford, CA.

Kraft, D., 1994. "Algorithm 733: TOMP–Fortran Modules for Optimal Control Calculations". *ACM Transactions on Mathematical Software (TOMS)*, Vol. 20, No. 3, pp. 262–281.

Kreisselmeier, G. and Steinhauser, R., 1979. "Systematic Control Design by Optimizing a Vector Performance Index". *In International Federation of Active Controls Syposium on Computer-Aided Design of Control Systems*.

Kroll, N. and Jain, R.K., 1987. "Solution of Two-Dimensional Euler Equations: Experience with a Finite Volume Code".

Martins, J.R.R.A., 2002. *A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization*. Ph.D. thesis, Stanford University, Stanford, CA.

Matunaga, M.A.L. and Azevedo, J.L.F., 2019. "Wing Optimization Using the OPENMDAO Framework and Low-Fidelity Aerodynamics". In *25th ABCM International Congress of Mechanical Engineering*. Uberlandia, MG, Brazil.

Nadarajah, S. and Jameson, A., 2000. "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization". In *38th Aerospace Sciences Meeting and Exhibit*. Reno, NV.

Nelder, J.A. and Mead, R., 1965. "A Simplex Method for Function Minimization". *The Computer Journal*, Vol. 7, No. 4, pp. 308–313.

Nocedal, J. and Wright, S., 2006. *Numerical Optimization*. Springer, New York.

Phillips, W. and Snyder, D., 2000. "Modern Adaptation of Prandtl's Classic Lifting-Line Theory". *Journal of Aircraft*, Vol. 37, pp. 662–670.

Pulliam, T.H., 1986. "Artificial Dissipation Models for the Euler Equations". *AIAA Journal*, Vol. 24, No. 12, pp. 1931–1940.

Reuther, J.J., 1996. *Aerodynamic Shape Optimization Using Control Theory*. Ph.D. thesis, University of California, Davis, Davis, CA.

Secco, N., Jasa, J., Kenway, G. and Martins, J., 2017. "Component-Based Geometry Manipulation for Aerodynamic Shape Optimization with Overset Meshes". In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA Paper No. 2017-3327, Reston, VA.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E.W., van der Plas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F. and van Mulbregt, P., 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods*, Vol. 17, pp. 261–272.

7. RESPONSIBILITY NOTICE

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