CAD-BASED AERODYNAMIC SHAPE DESIGN OPTIMIZATION WITH THE DLR TAU CODE

M. Martín[1], E. Andrés[2], M. Widhalm[3], P. Bitrián[4], C. Lozano[5]
[2, 4, 5] National Institute of Aerospace Technology – Fluid Dynamics Branch
[3] German Aerospace Center – Institute of Flow Technology
[1] cfd03@inta.es , [2] esther.andres@insa.es , [3] markus.widhalm@dlr.de

Keywords: *shape design, gradient-based optimization, adjoint methods*

Abstract

With the improving capabilities of Computational Fluids Dynamics (CFD) on predicting aerodynamic performance, these tools are now being increasingly used for aerodynamic design optimization in the aerospace industry. In this paper, an automated optimization framework is presented to address inviscid aerodynamic design problems. Key aspects of this framework include the use of the continuous adjoint methodology to make the computational requirements independent from the number of design variables and Computer Aided Design (CAD) based shape parameterization, which uses the flexibility of Non-Uniform Rational B-Splines (NURBS) to handle complex configurations.

1 Introduction

The Advisory Council for Aeronautics Research in Europe (ACARE) asked, in the document VISION 2020 [1], for a reduction of fuel consumption by 50% in 2020 compared to the data in 2000. For achieving this objective, several disciplines will have to deploy all their current capabilities and introduce innovative concepts and automatization into their processes.

In the aerodynamics field, the use of integrated tools for shape optimization has grown in importance for the preliminary design of an aircraft, where mainly inviscid flow computations are performed due to the severe computational time required to obtain more accuracy. Within a gradient-based optimization approach, the use of the adjoint methodology [2-7] has been introduced during the last decade and it has demonstrated to be an efficient method to compute the gradients of a cost function.

Using the adjoint methodology, the sensitivities are calculated on each grid point. One of the main problems of using the grid points directly as design variables is the occurrence of surface bumps during the optimization process. The effect of these bumps produces an optimized geometry for a specific flight condition, but behaves very poorly if that condition changes.
In order to solve these problems, a higher-level parameterization should be considered. The use of NURBS has been suggested [8-10] as an efficient and flexible parameterization, able to represent complex configurations while preventing the formation of local bumps.

The main advantages of the approach proposed in this paper for shape optimization are:

- The use of the Adjoint Methodology for fast gradient computation. It provides independence from the number of design variables.
- The use of Computer Aided Design (CAD) based parameterization; meaning NURBS.

The structure of the paper is as follows: First a brief introduction to the TAU code is presented. Next, the proposed CAD-based aerodynamic shape optimization process is described to introduce all the steps involved. In the third section, the NURBS parameterization and the approach employed for the so called inversion point problem are explained. The continuous

[1] PHD student from Universidad Politecnica Madrid (ETSIT), [2] INSA subcontracted, [4] technical assistance

adjoint formulation and how the gradients are referred to the control points of the NURBS are briefly described in the fourth section. Finally, the optimization framework is applied to both 2D and 3D configurations and the results are shown.

2 The unstructured TAU code

The fluid flow over the object of interest is simulated with the TAU code [11]. The unsteady TAU code solves the compressible, three-dimensional Reynolds-Averaged Navier-Stokes equations using a finite volume formulation and based on hybrid unstructured grid approach.

A dual-grid approach with an edge based data structure is used in order to make the flow solver independent from the cell types used in the initial grid. The TAU code consists of several different modules, including:

- The Grid Partitioner, which splits the primary grid into *n* number of subgrids for *n* processors.
- The Preprocessor module, which uses the information from the initial grid to create a dual-grid and secondly coarser grids for multi-grid.
- The Solver module, which performs the flow and adjoint calculations on the dual-grid.
- The Adaptation module, which refines and derefines the computational grid with different indicator functions.
- The Deformation module, which propagates the deformation of the surface grid points to the surrounding volume grid.
- The Postprocessing module, which is used to convert results into formats usable by popular visualization tools.

Together all modules are available with Python interfaces for computing complex application, e.g. unsteady cases, complete force polar curves or fluid-structure couplings in an automatic framework. Furthermore, it eases the usage on highly massive parallel computers to execute applications.

3 Introduction to the CAD-based aerodynamic shape optimization process

As it has been mentioned above, the proposed approach is a combination of the continuous adjoint methodology for gradient computation and the NURBS definition of the geometry for parameterization.

The CAD-based shape optimization process comprises the following steps:

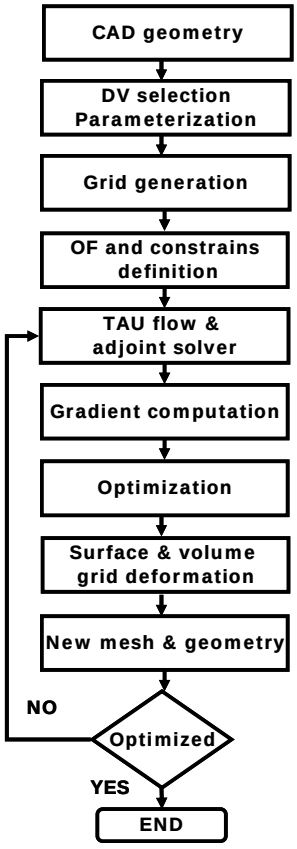


Fig. 1: CAD-based optimization process

3.1 CAD geometry

The original geometry is an input to the process, and it can be obtained directly from CAD applications, as for example, CATIA (in IGES format).

3.2 Selection of the Design Variables. Parameterization

The parameterization is represented with a NURBS which describes the surface of the desired geometry. The design variables are the

coordinates (x,y,z) and weights of the NURBS control points, so for each control point there are four design variables. The main advantage of using NURBS is that it provides a global parameterization with a smooth surface and a control of the curvature while still maintaining the locality in the deformation. In addition, the optimized surface at the end of the process has the correct format to feed directly the CAD and grid generation applications.

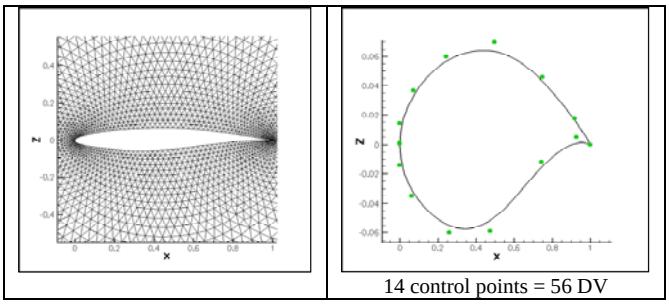


Fig. 2: RAE2822 unstructured grid (left) and NURBS parameterization (right)

With the possibility to handle different NURBS patches for each part of an aircraft, as for example the wing and fuselage, this parameterization is able to represent almost every kind of surface. This feature allows different levels of optimization; but working with several NURBS patches requires a correct treatment of the intersections and special considerations related to the continuity between patches.

In practice, the NURBS description provided directly by CAD tools is not usually appropriated for an optimization process and should be modified, for example, by modifying the order of the NURBS or setting an evenly distribution of the control points, to improve the chances for a good optimization.

3.3 Grid generation

The grids used for simulations in this paper were created with the hybrid grid generation software Centaur, developed by Centaur Soft [12].

3.4 Objective Functions and Constraints definition

The most common objective functions, as, for instance, minimize drag coefficient maintaining constant lift coefficient, are implemented in the optimization framework. An analysis of the efficiency of each objective function has been performed and the results will be displayed in section 6.

During the optimization, the following constraints can be applied:

- Fixed control points in all directions.
- Fixed control points in one direction (x,y,z).
- Fixed control points for weights.

For a wing section, the most common constraints are to fix the control point located on the trailing edge and to maintain the chord length constant. Volumetric and other geometric constraints can also be applied, but the objective function of such problems is strongly non-linear, so that the optimizer most likely converges towards a local minimum that is not too far from the original configuration.

3.5 TAU Flow and Adjoint solver

The flow and adjoint solver modules are executed to obtain the sensitivities values for the selected cost function over the surface grid points. It is important to remark that the computation of the sensitivities, using a continuous adjoint approach, depends only on the surface values (how a change of a design variable affects the considered cost function).

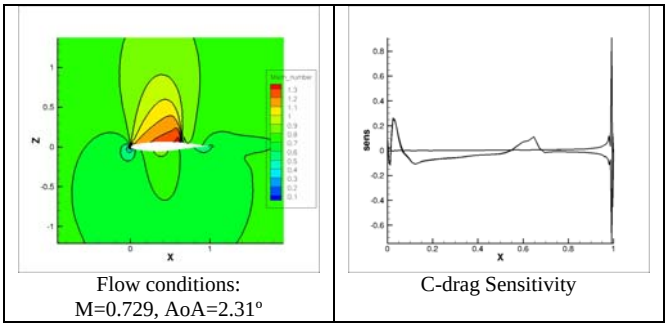


Fig. 3: Mach number (left) and sensitivity value (right) along the RAE2822 profile

As it will be shown later, the computation of the flow and adjoint solution consumes most of the required computational time in the whole process.

3.6 Gradient computation over the NURBS control points

The functional sensitivities over the surface grid points obtained from the continuous adjoint solution are combined now with the geometric sensitivities (how a movement of a design variable affects the surface grid) [18].

The gradients are then calculated on each control point of the NURBS using the following formulation

$$\delta J = \int_S (\delta j \cdot \delta x \cdot n) ds \qquad \textbf{(1)}$$

where δj is an adjoint solution calculated on each vertex of the computational grid, δx are the geometric sensitivities that are analytically calculated from the NURBS equations, n is the surface normal and ds is the element surface area. This formulation does not consider tangent deformations of the surface because its effect is considered negligible, in most cases, but the leading edge and the trailing edge could require a special treatment that includes tangential derivatives. Additionally, this formulation works if the normal is correctly calculated, which sometimes it is not the case at the trailing edge due to a discontinuity of the curvature.

3.7 Optimization

A simple steepest descent optimization algorithm has been implemented into the framework for testing purposes.

3.8 Surface and volume deformation

The surface deformation is performed by the modification of the NURBS control points and the regeneration of the surface grid using the parametric coordinates calculated from the inversion point algorithm. This guarantees that the vertices are always on the NURBS surface.

The same parametric values are used during the whole optimization; therefore the inversion point step has only to be done once and its computational time requirements do not affect to the process. The design variables are the control point coordinates, which can move freely in all x, y, z directions and the weight, although in practice the weight is not used. Variations of the knots distribution and basis function are not considered. Additionally, a good practice could involve a movement of the control points in an average normal direction.

Once the new surface grid is obtained, the volume grid is deformed with an advancing front algorithm, using the TAU deformation module.

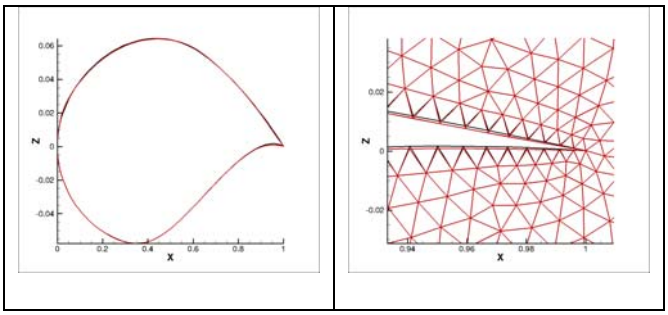


Fig. 4: RAE2822 surface (left) and volume deformation (right)

4 Shape parameterization

4.1 Introduction to shape parameterization into an optimization framework

The geometry parameterization is crucial in an automatic aerodynamic optimization problem. NURBS have demonstrated to be able to accurately represent a large family of airfoils; furthermore, they are also the standard of geometry format definition in IGES (International Graphics Exchange Specification). In addition, from a practical point of view, using the same CAD format significantly reduces the integration effort necessary to carry out multidisciplinary design of complex configurations.

This parameterization ensures good smoothness properties [10], reducing the risk of numerical noise, while the parameterization is still local, which means that when a control point is moved, only a portion of the surface is modified, leaving the rest intact, and gives enough freedom for the optimizer to converge to an optimal design.

4.2 Brief mathematical background of Non Uniform B-Splines (NURBS)

From a mathematical point of view, NURBS surfaces are defined as:

$$S(\xi,\eta)=\frac{\sum_{i}^{I}\sum_{j}^{J}U_{i,p}(\xi)V_{j,q}(\eta)w_{ij}C_{ij}}{\sum_{i}^{I}\sum_{j}^{J}U_{i,p}(\xi)V_{j,q}(\eta)w_{ij}} \qquad (2)$$

where C are the control points spatial coordinates, w are the control points weights, and U and V are the basis functions which are calculated using the following expression:

$$U_{i,1}(\xi)=\begin{cases}1 \; if \;\; u_i \le \xi < u_{i+1}\\ 0 \quad otherwise\end{cases}$$
$$U_{i,k}(\xi)=\frac{(\xi-u_i)U_{i,k-1}(\xi)}{u_{i+k-1}-u_i}+\frac{(u_{i+k}-\xi)U_{i+1,k-1}(\xi)}{u_{i+k}-u_{i+1}} \qquad (3)$$

The basis coefficients are calculated from the knots vectors $\overline{U}$ and $\overline{V}$ which is a sequence of real numbers that frequently have the form:

$$\overline{U}=\{\underbrace{0,...,0}_{p+1},u_{p+1},...u_i,...,u_I,\underbrace{1,...,1}_{p+1}\} \qquad (4)$$

Basis functions are equal to zero everywhere except for an interval delimited by the order of the NURBS, defining the area of influence of each control point. A more extended reference for NURBS can be found in [13].

4.3 Point Inversion Algorithm

Any vertex of the computational grid on the surface should be mapped onto the original CAD geometry which is stored as a set of NURBS. During the optimization process, using the parametric coordinates $\{\xi,\eta\}$, it is easy to map the vertices of the computational grid onto the deformed NURBS surface and obtain the new spatial coordinates. The process to transform space coordinates into the parametric coordinates of the NURBS is usually referred as the point inversion problem. That is $\Re^3\{x,y,z\}\rightarrow\Re^2\{\xi,\eta\}$

The parametric values are not provided by the grid generator application and therefore the point inversion is necessary. NURBS are rational polynomial expressions with normally middle or high degree, and therefore, the problem of finding the parametric coordinates analytically becomes from hard to impossible as the NURBS order increases.

Some iterative algorithms have been considered to obtain these parametric coordinates: Hu & Wallner [14], Ma & Hewitt [15], Selimovic [16], also grid scan and quads, Redondo [17], which is essentially a variant of the bisection method. All these methodologies suffer from the same problem: they perform the task well only if an appropriated interval or initial value is provided. For making the problem even harder, the vertices of the grid not always match the NURBS surface, and the presence of discontinuities, intersections and kinks are also frequent.

In order to obtain a valid initial value, the implemented method is based on three assumptions:

- the normal of the vertex is known or can be reasonably calculated;
- the vertex does not need to be on the NURBS surface but very close;
- the parametric coordinates calculated for a low order NURBS are reasonably accurate for a higher order NURBS.

With these assumptions, the intersection of a second order NURBS with the normal associated to the vertex can be calculated and the parametric values obtained.
The expression of a second order NURBS is:

$$S(\xi,\eta)=\sum_{i=I-1}^{I}\sum_{j=J-1}^{J}U_{i,2}V_{i,2}G_{i,j}=\phi(r\cdot N+P) \quad (5)$$

where U and V are the basis functions and G are the control point coordinates multiplied by its correspondent weight to drop it out of the equation.

$$G_{i,j}=C_{i,j}w_{i,j} \quad (6)$$

In equation (5), the right term is the parametric equation of a line defined by the normal N and the vertex P which is multiplied by the term φ; this term is the denominator that always appears in the NURBS equation which is basically constant if the weights are the same.

$$\phi=\sum_{i}^{I}\sum_{j}^{J}U_{i,p}(\xi)V_{j,q}(\eta)w_{ij} \quad (7)$$

After expanding the above equation, a second order equation system is obtained in the form:

$$\bar{a}\xi\eta+\bar{b}\,\xi+\bar{c}\,\eta+\bar{d}r+\bar{e}=0 \quad (8)$$

where

$$\bar{a}=G_{i-1,j-1}+G_{i,j}-G_{i-1,j}-G_{1,j-1} \quad (9)$$

$$\bar{b}=v_jG_{i-1,j}+v_{j+1}G_{i,j-1}-v_{j+1}G_{i-1,j-1}-v_jG_{i,j} \quad (10)$$

$$\bar{c}=u_{i+1}G_{i-1,j}+u_iG_{i,j-1}-u_{i+1}G_{i-1,j-1}-u_iG_{i,j} \quad (11)$$

$$\bar{d}=-\phi(u_{i+1}-u_i)(v_{i+1}-v_i)N \quad (12)$$

$$\bar{e}=u_{i+1}v_{j+1}G_{i-1,j-1}+u_iv_jG_{i,j}-u_{i+1}v_jG_{i-1,j}-\\-u_iv_{j+1}G_{1,j-1}-\phi(u_{i+1}-u_i)(v_{i+1}-v_i)P \quad (13)$$

Solving this system for all possible combination of pairs $\{i,j\}$, several candidates of parametric values ξ_{ij} and η_{ij} are obtained. Between all those possible candidates, the final ones are selected by using a search of minimum distance and fulfil a condition to be valid parametric values:

$$\{\xi^*_{ij},\eta^*_{ij}\}=Min\{[S(\xi_{ij},\eta_{ij})-P]^2\}$$
$$with$$
$$U_i-\sigma_u^*\le\xi^*_{ij}<U_{i+1}+\sigma_u^*$$
$$V_i-\sigma_v^*\le\eta^*_{ij}<V_{i+1}+\sigma_v^* \quad (14)$$

where σ_u^* and σ_v^* are relaxation factors, which virtually extends the coverage of the NURBS out of its boundaries and kinks; e.g., $\sigma_u^*=0.5(U_{i+1}-U_i)$. The parametric values calculated for a second order NURBS are close to the real values, or at least close enough to be valid initial values to be used in iterative methods. Additionally it provides several candidates if the first one fails to converge.

4.4 NURBS for the optimization process

In general, NURBS provided directly by the CAD applications are designed to accurately describe the geometry, but they are not the most appropriate ones as the initial parameterization in an optimization process. For example, a large density of control points is frequently found on zones with large curvature, while very sparse control points are found on flat zones.

In order to improve the chances of a good optimization, it could be necessary to transform these NURBS into more suitable ones that fulfil some requirements. For example, the optimizer works better with a reduced number and evenly distributed control points with a low order NURBS. By increasing the number of control points, the optimization could be slightly improved, as increases the possible shapes; but in this case if the order of the NURBS is still low, it could lead to a loss of smoothness and undesired bumps could appear during the optimization. So, there is a trade-off between the number of control points and the order of the NURBS.

5 Fast computation of the gradients

5.1 Continuous adjoint methodology

The computation of the gradients or sensitivity derivatives of the cost function, such as drag

and aerodynamic efficiency, are calculated using the adjoint formulation [18] for normal deformations.

The total derivative of a cost function J due to a design variable D can be written as:

$$\frac{dJ}{dD} = \frac{\partial J}{\partial D} + \frac{\partial J}{\partial W}\frac{\partial W}{\partial D} \qquad \textbf{(15)}$$

The first term is often referred as the sensitivities of a functional J. Upon deformation of the surface, the cost function varies due to a variation of the geometry. In aerodynamic optimization problems, suitable cost function examples include aerodynamic coefficients like drag and lift, which are directly calculated from the pressure distribution over the aerodynamic surface S.

$$J = \int_s f(W)ds \qquad \textbf{(16)}$$

It is assumed that the cost function f is differentiable, although this assumption may not be valid in the presence of shock waves or other discontinuities, numerical dissipation in the implementation can mitigate this effect and provide an approximate solution.

In this case, f has no dependency from the time or from the time derivatives of the flow variables W. So the expression can be written as:

$$\frac{dJ}{dD} = \int_{\partial s} f(W)ds + \int_s \frac{\partial f}{\partial W}\frac{\partial W}{\partial D}ds \qquad \textbf{(17)}$$

The equation above includes a term related to the geometric variation and a second term related to the flow variation. Cost functions as drag and lift are composed of terms that involve non geometric and geometric quantities such as the surface normal n. So the first term can be expanded containing three terms [18][19]:

$$\int_{\partial s} f(W)ds = \int_s \frac{\partial f}{\partial W}(\delta X \cdot \nabla W)ds + \\ + \int_S \frac{\partial f}{\partial n}\delta n dS + \int_S f\delta dS \qquad \textbf{(18)}$$

The notation δ stands for the deformation, with respect to the design variables D. The term δX denotes the deformation of the points that define the aerodynamic surface S, δn is the variation of the normal and δdS is the variation of the surface element upon deformation.

The term $\int_S f\delta dS$ is usually referred as the curvature term [20] with the form:

$$\int_S f\delta dS = \int_S fK dS$$
$$K = \begin{cases} \partial_{tg}\delta x_{tg} - k\delta x_n, & (2D) \\ \nabla_{tg}\cdot\delta\vec{x}_{tg} - 2H_m\delta\vec{x}_n & (3D) \end{cases} \qquad \textbf{(19)}$$

where k and H_m are the curvature profile and mean curvature of the surface respectively.

For the flow variation term, the adjoint equations are defined by introducing the Lagrange multiplier Λ, also referred as the adjoint vector or state, to the corresponding linearized Euler equations subject to the adjoint equations:

$$\left(\frac{\partial F}{\partial W}\right)^T \cdot \nabla\Lambda^T = 0 \qquad \textbf{(20)}$$

where F is the inviscid flux vector. With the appropriate adjoint boundary conditions on S so as to allow the computation of the flow variation term [4][6][18]. Optimization is possible in this case with respect to any functional that depends on the distribution of the pressure on the aerodynamic surface.

For drag and lift optimization problems the functional is defined with the following expression:

$$J = \int_S C_p(n\cdot d)ds;\ \vec{d} = \begin{cases} \{\cos\alpha, \sin\alpha\}\ (drag) \\ \{-\sin\alpha, \cos\alpha\}\ (lift) \end{cases} \qquad \textbf{(21)}$$

where C_p is the pressure coefficient, d is the force direction vector and α is the angle of attack. By defining

$$\Psi = \rho\psi_1 + \rho u\psi_2 + \rho v\psi_3 + \rho w\psi_4 + \rho H\psi_5 \qquad \textbf{(22)}$$

where ρ is the density, u, v, w are the Cartesian velocities of the fluid, H is the enthalpy, ψ_i are the adjoint variables, and by separating the normal and tangent parts of the deformation, compact expressions result for the drag and lift coefficients [18]:

$$\delta J = \int_S (d \cdot \nabla C_p + (\nabla \cdot v)\Psi + (v \cdot t)\partial_{tg}\Psi)\alpha ds \qquad \textbf{(23)}$$

where n and t are the normal and tangent vectors to the surface element respectively, and v is the velocity vector of the fluid.

The TAU code provides the flow solution and the surface sensitivities. By knowing the parametric coordinates of the computational grid, it is possible to translate the surface sensitivities onto the control points of the NURBS using equation (1).

The geometric derivatives δx to a displacement of the position of the control point through a Cartesian direction e_k are exactly the basis coefficients.

$$\delta x = U_{i,p} V_{j,q} \vec{e}_k \qquad \textbf{(24)}$$

while the geometric derivative related to the weight of the control point is calculated with the following expression:

$$\delta w = U_{i,p} V_{j,q} (P - C_{ij}) \qquad \textbf{(25)}$$

where P and C_{ij} are the spatial coordinates of the surface element and control point respectively.

5.2 Validation using finite differences

For validation purposes, the gradients have been also calculated using the finite differences approach. The gradients obtained should be similar to those computed with the adjoint methodology. However, insufficient grid resolution and inaccurate normal calculation may degrade the solution leading to a fail in the optimization process.

As it can be observed in the figures 6, 7 and 8, the gradients are similar, although there are important discrepancies in the gradients of the design variables located on the trailing edge (design variable number 46), where the definition of the normal vector is not clearly calculated, because of a discontinuity of the curvature, and the tangential component of the sensitivity is not considered. The deformation of this design variable is displayed in figure 5.

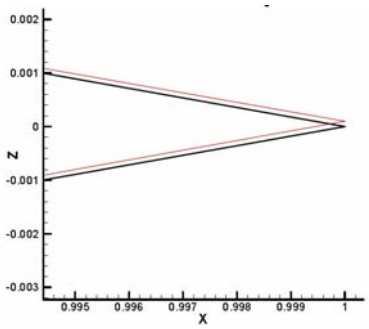


Fig. 5: Surface deformation for design variable number 46

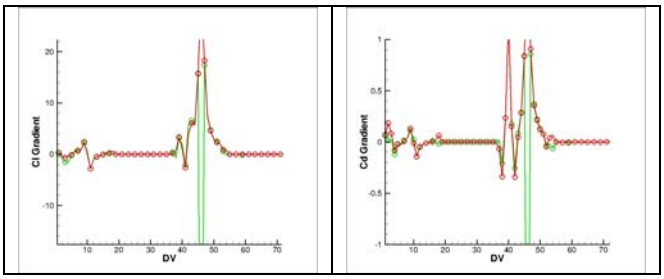


Fig. 6: NACA0012 Adjoint (red) vs. finite differences gradients (green) for lift (left) and drag (right) coefficients

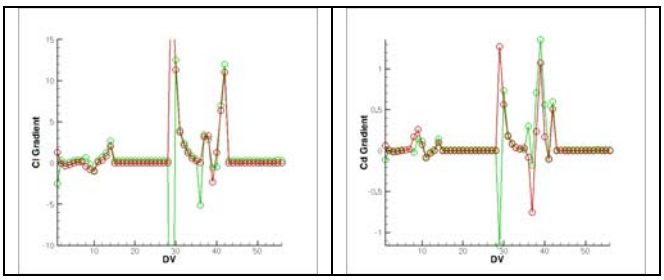


Fig. 7: RAE2822 Adjoint (red) vs. finite differences gradients (green) for lift (left) and drag (right) coefficients

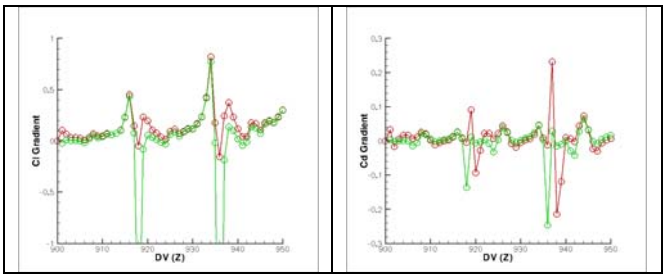


Fig. 8: ONERAM6 adjoint (red) vs. finite differences gradients (green)

These discrepancies on the trailing edge are not of much concern, because during the optimization process the control point located there is fixed. However, an alternative method for calculating normal vectors in areas with discontinuities of the curvature and a formulation that considers the tangential component of the derivatives should be considered in the near future.

6 Application to aerodynamic shape optimization

In this section, the methodology is applied to the inviscid aerodynamic optimization of 2D NACA0012 and RAE2822 profiles and both 3D ONERAM6 wing and DLR F6 wing-fuselage configurations. The results are displayed and commented.

6.1 Optimization of a 2D NACA0012 profile

The design optimization is applied to a NACA0012 airfoil as initial design at M=0.8, AoA=1.25º. The NURBS curve, figure 9, has 18 control points and is capable to represent the shape with reasonable accuracy.

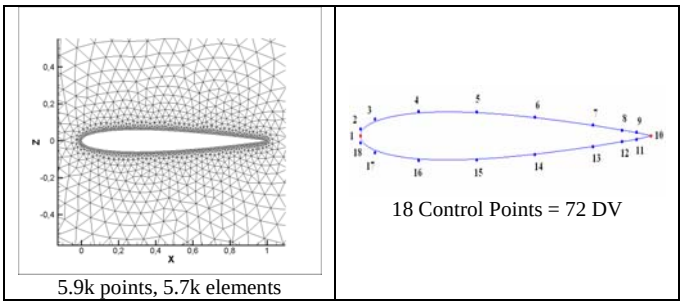


Fig. 9: NACA0012 grid (left) and NURBS parameterization (right)

The current study is an inviscid case, which accurately predicts wave shocks in a transonic flow regime. The design optimization problem selected is the minimization of the drag coefficient maintaining the lift coefficient into a range of 10% from the original.

$$Min\{C_d + a(C_L - C_L{}^o)^4\}$$
$$\delta = -\{\dot{c}_d + 4{\cdot}10{\cdot}\dot{c}_L(C_L - C_L{}^o)^3\} \qquad \textbf{(26)}$$

where C_D and C_L are the drag and lift coefficients respectively, c_d and c_l are the normalized values of drag and lift coefficients and $C_L{}^T$ is the initial value of the lift coefficient.

The optimization algorithm employed is a simple descent method in which small steps are taken in the gradient direction [21]. During the whole optimization process the maximum deformation is maintained constant, approximately 1% of the camber.

During the process, the following constraints are applied:

- The control points situated on the leading and trailing edge are fixed in order to maintain the chord line and angle of attack.
- All control points are fixed in weight.
- The chord is maintained constant.

We can observe that the optimization process has reduced the wave drag by preventing rapid acceleration of the flow at the leading edge. As it can be observed in the following figures the shock has almost disappeared.

The efficiency, in figure 10, means the lift coefficient divided by the drag coefficient. All the coefficients values are normalized with their initial value.

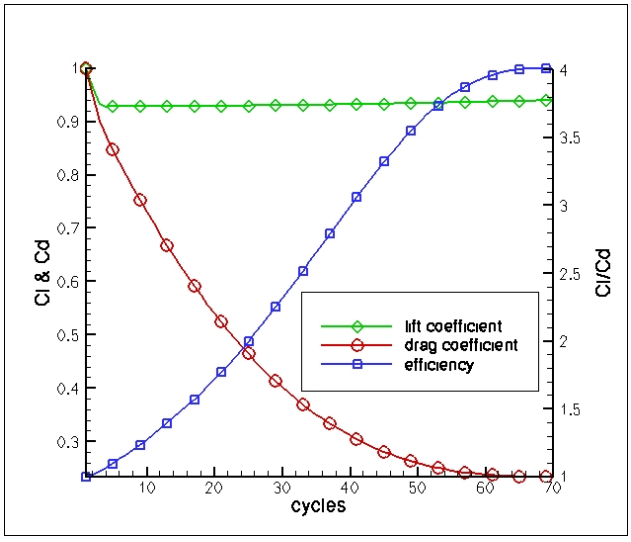


Fig. 10: Convergence history of the optimization using a NACA0012 as initial design

After 70 optimization cycles the drag coefficient has been reduced by more than 70% of its initial value, while the lift coefficient is kept constant in the desired range.

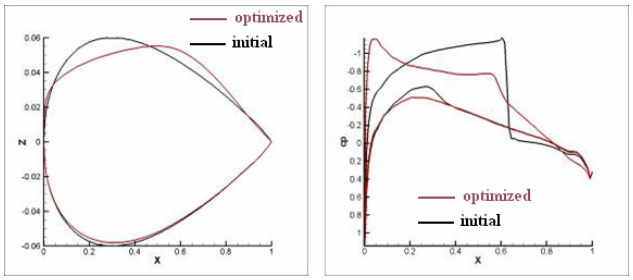


Fig. 11: Change of airfoil shape (left) and pressure distribution (right) after the optimization process

The optimization significantly reduces the strong shock on the upper side of the profile.

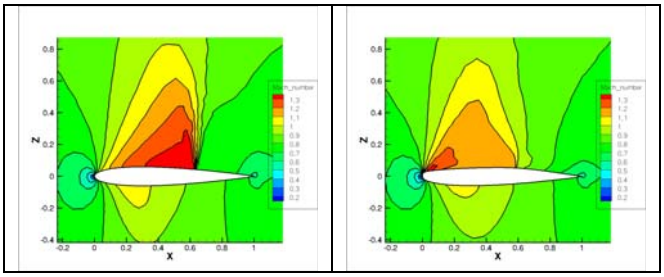


Fig. 12: Distribution of the Mach number before (left) and after (right) the optimization

The weak shock that appears near the leading edge, after the optimization, could be a consequence of the sensitivity formulation used. This fact will be further studied, and the tangential component of the derivatives will be included in the formulation to improve the solution.

6.2 Optimization of a 2D RAE 2822 profile

The methodology is also applied to a RAE 2822 at M=0.729, AoA=2.31º airfoil as initial design, described as a NURBS curve with 14 control points. The grid and NURBS used for this optimization can be observed in figure 13.

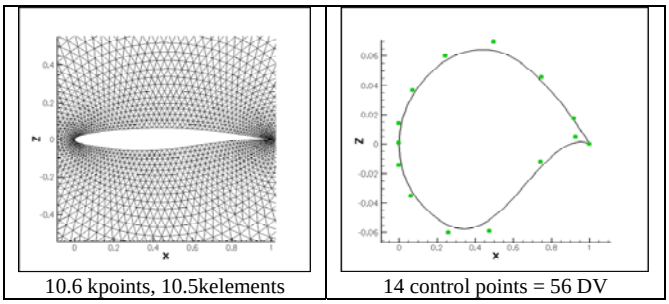


Fig. 13: RAE2822 unstructured grid (left) and NURBS (right)

The objective function selected is the same as before: minimization of the drag coefficient maintaining the lift coefficient into a range of 10% from the original.

During the optimization, the following constraints are applied:

- The control point situated on the trailing edge is fixed in all directions.
- All control points are fixed in X and Y directions, only Z movements are allowed.
- All control points are fixed in weight.
- The chord is maintained constant.

The convergence of lift and drag coefficients over the optimization steps are displayed in picture 14.

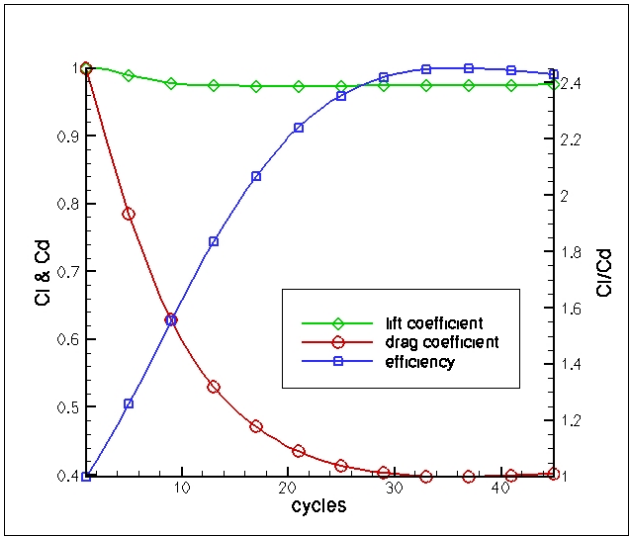


Fig. 14: Convergence history of the optimization using a RAE 2822 as initial design

After 35 optimization cycles the drag coefficient has been reduced by 60% of its initial value, while the lift coefficient is kept constant in the desired range.

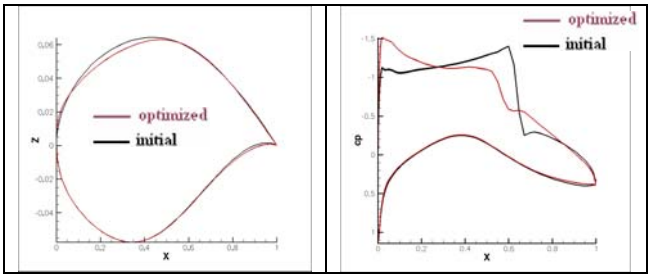


Fig. 15: Change of airfoil shape (left) and pressure distribution (right) after the optimization process

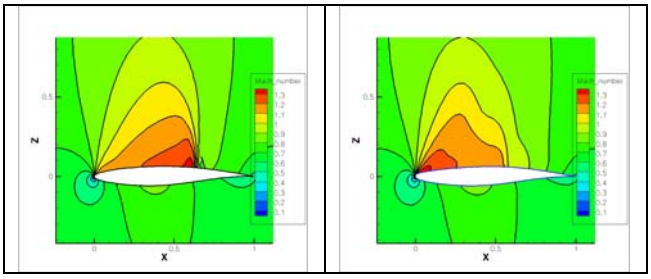


Fig. 16: Distribution of the Mach number before (left) and after (right) the optimization

As it can be observed in figures 15 and 16, the shock has almost disappeared after the optimization process; however a new weak shock has appeared near the leading edge.

6.3 Optimization of a 3D ONERA M6 wing

The methodology is also applied to optimize a three dimensional inviscid ONERA M6 wing at M=0.8395 and AoA=3.06º, described as a NURBS surface (for both upper and lower surface) using 25 x 18 control points.

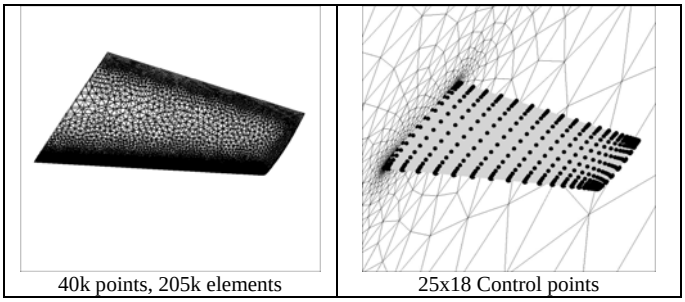


Fig. 17: Grid (left) and NURBS (right)

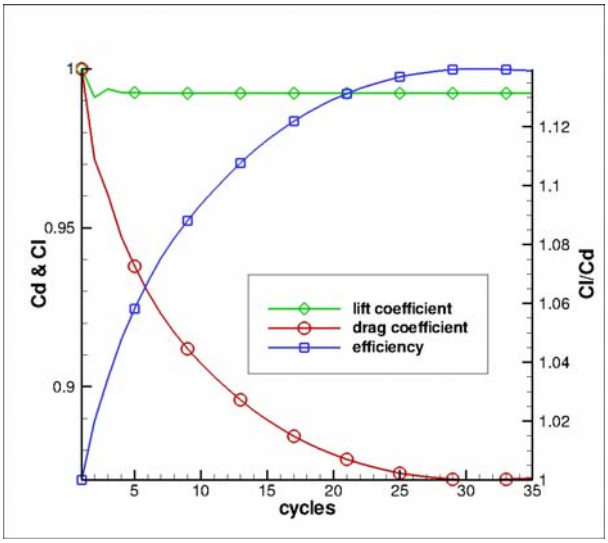


Fig. 18: Convergence history of optimization using an ONERA M6 wing as initial design

As constraint, the control points located on the trailing edge are kept fixed in all directions.

In this case, after 35 optimization cycles the drag coefficient has been reduced by 15% of its initial value, while the lift coefficient is only reduced by 1% (figures 18, 19 and 20).

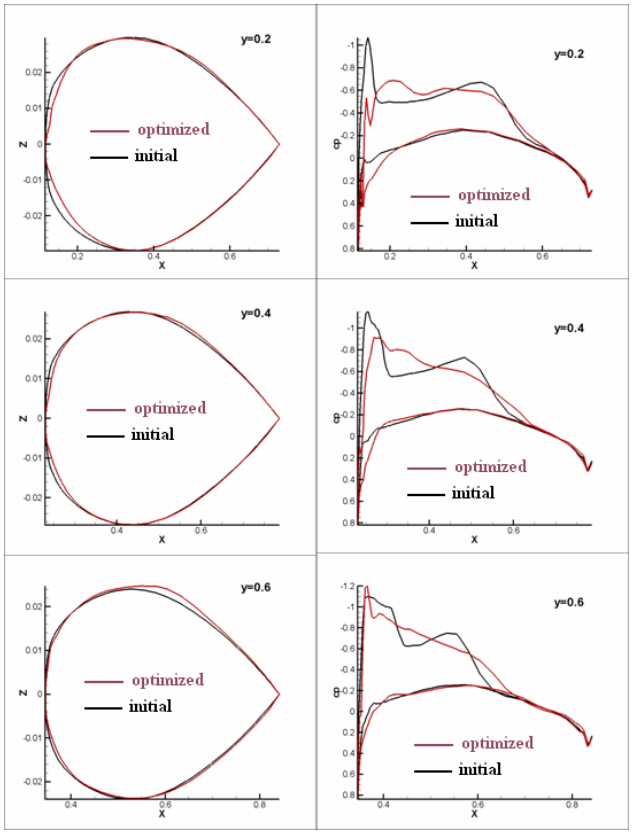


Fig. 19: Airfoil shape (left) and pressure distributions (right) for an ONERA M6 wing

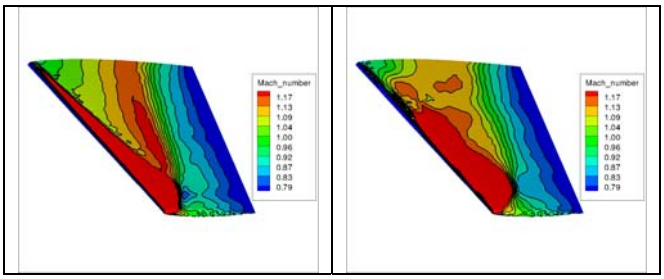


Fig. 20: Distribution of the Mach number before (left) and after (right) the optimization.

The lambda shock has been considerably weakened after the optimization process.

6.4 Optimization of a 3D DLR F6 wing-fuselage configuration

To complete the study a more complex test is chosen for optimization. The selected one is the DLR F6 configuration, which is a simplified wing-fuselage geometry that has been used in the past for validation of CFD codes at the second [22] and third [23] AIAA sponsored Drag Prediction Workshops. The computational grid has 218 kpoints and 1.2 mill. elements. The flow conditions are a free-stream Mach number of 0.8 and a fixed angle of attack at 0 degree.

For the wing optimization problem, a NURBS surface is defined over the F6 wing using 33 x 21 control points (figure 21).

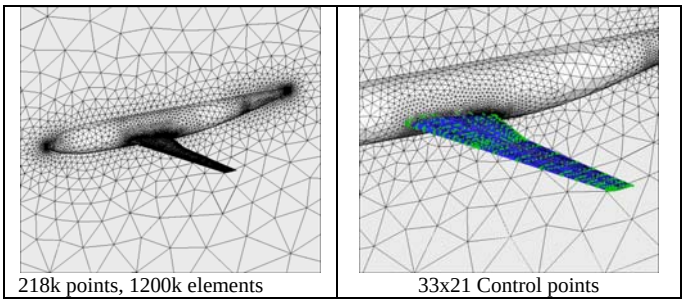


Fig. 21: DLR F6 Euler grid (left) and NURBS parameterization of the wing (right)

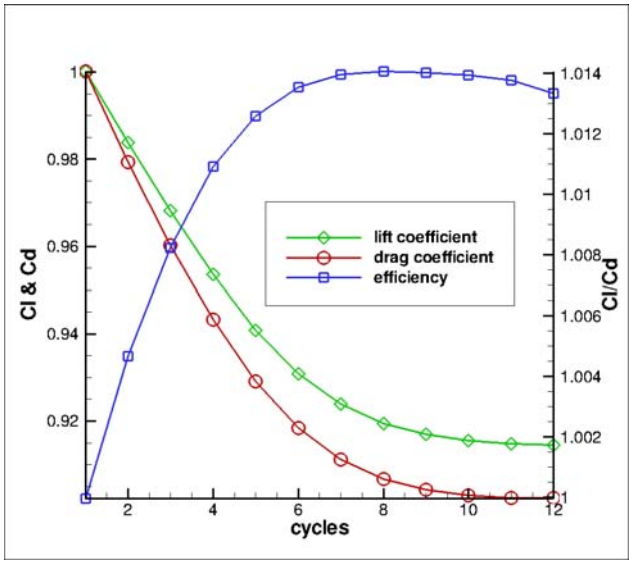


Fig. 22: Convergence history of the optimization using a DLR F6 wing fuselage as initial design

The control points located on the trailing and leading edge are kept fixed in all directions. In addition, the rows of control points close to the fuselage are kept constant too, in order to handle the intersection.

In this case, we used a high optimization step in order to reduce the required computational time. After 12 optimization cycles the drag coefficient has been reduced by 10% of its initial value, but the lift coefficient is also reduced by 8% (figures 22, 23 and 24).

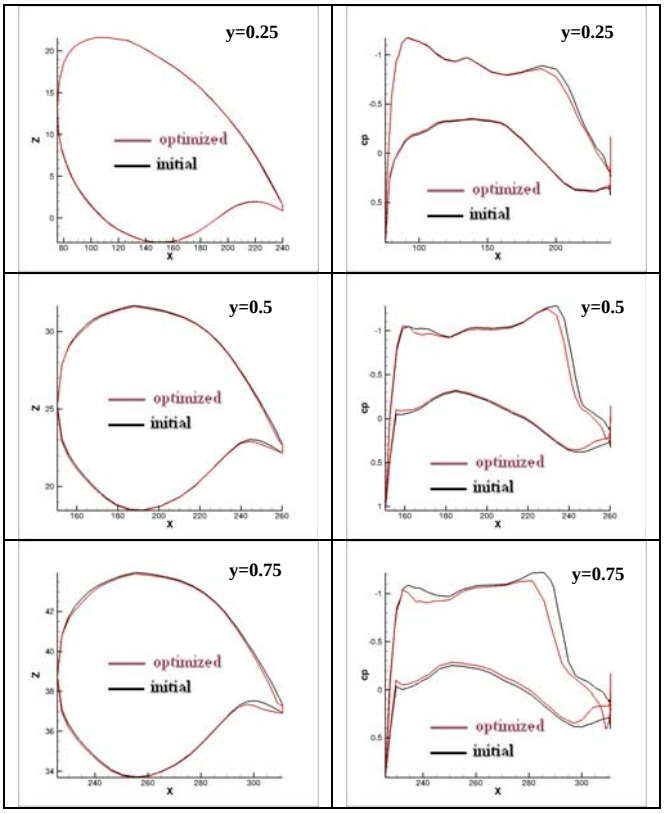


Fig. 23: Sections of the airfoil shape (left) and pressure distributions (right) after the optimization process

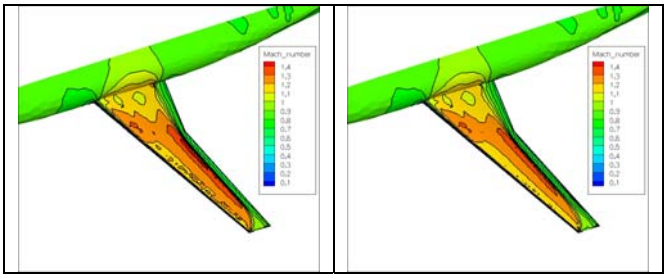


Fig. 24: Distribution of the Mach number before (left) and after 10 optimization cycles (right).

The shock wave presented in the initial configuration has been slightly decreased as can be observed in the C_p distribution along the wing.

However, the results obtained in this configuration are not as good as expected, and further work will be necessary to improve different steps. In particular, the location of the control points over the wing can considerably affect the optimization process, and therefore, different distributions of control points over the wing surface will be further investigated.

6.5 Execution time considerations

In order to measure the efficiency of the proposed methodology in terms of computational time, a profiling has been performed for an optimization cycle in the ONERA M6 wing configuration.

The results are displayed in the following table:

STEP	TIME (seconds)	%
Point Inversion (only needed 1 time in the whole process)	*142*	
Preprocessing	*1*	*0.18 %*
Flow Solver	***314***	***58.97%***
Adjoint solver for drag coefficient	***110***	***20.66%***
Adjoint solver for lift coefficient	***106***	***19.90%***
Gradient computation	*0.5*	*0.1%*
Surface deformation	*0.01*	*<0.01%*
Volume deformation	*1*	*0.18 %*
TOTAL per cycle	*532.51*	

As it can be concluded from the table above, the most time consuming steps are the flow and adjoint solver execution. The rest of the process is almost negligible in terms of computational requirements.

For the flow solver the residual density magnitude has been decreased until 10^{-11}, while

for the adjoint solver it has been decreased only until 10^{-3}, because it has been observed that it is enough for a good quality of the gradients [24].

The total execution time using 50 cycles for the optimization of the 2D NACA0012 profile was 3000 seconds (50 min.), while the optimization of the 3D ONERA M6 wing took 26.000 seconds (~7h), using a sequential run on a Linux x86 computer.

6.6 Comparison of three different objective functions for the 2D NACA0012 profile optimization

The objective function employed in equation (26) is very specific for the optimization problems performed in this paper, and the arbitrary parameter *a* has been designed to be very restrictive with the lift. Having strong constraints, there is a possibility that the optimization stagnates without converging to any optimal profile. For completeness purposes, different objective functions have been selected to analyze how the optimization performs.

These objective functions have the form:

$$Min\left\{\left(\frac{C_d}{C_d^{\,o}}\right)^N + a\left(\frac{C_L - C_L^{\,T}}{C_L^{\,o}}\right)^M\right\} \qquad \textbf{(27)}$$

and

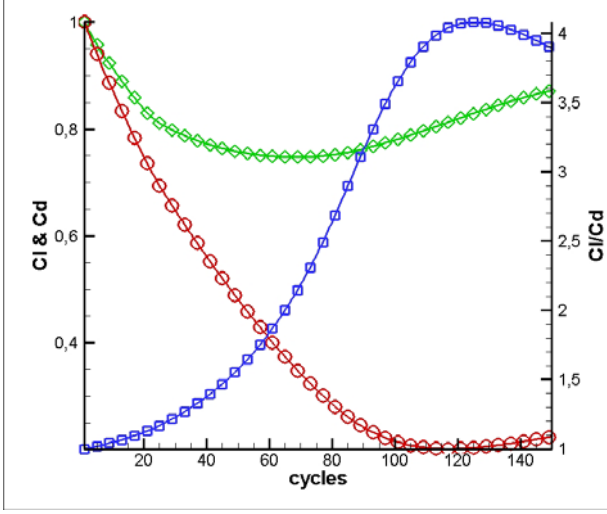
$$Min\left\{\left(\frac{C_L^{\,o}}{C_d^{\,o}}\cdot\frac{C_d}{C_L}\right)^N + a\left(\frac{C_L - C_L^{\,T}}{C_L^{\,o}}\right)^M\right\} \qquad \textbf{(28)}$$

The objective function of equation (27) is in fact a generalization of the objective function of equation (26). On the other hand, equation (28) means the maximization of the efficiency with a target lift $C_L^{\,T}$. In these objective functions, the drag and lift coefficients are normalized with the initial values $C_L^{\,o}$ and $C_D^{\,o}$ to create more general functions. In the following cases, the arbitrary parameter *a* is taken as 1. By increasing *a* the function will be more restrictive with the lift.

The first objective function is taking *N=1*, and *M=2*.

$$Min\left\{\frac{C_d}{C_d^{\,o}} + \left(\frac{C_L - C_L^{\,T}}{C_L^{\,o}}\right)^2\right\}$$
$$\delta = -\left\{\frac{\dot{c}_d}{C_d^{\,o}} + \frac{2\dot{c}_L\left(C_L - C_L^{\,T}\right)}{\left(C_L^{\,o}\right)^2}\right\} \qquad \textbf{(29)}$$

The evolution of the drag, lift and efficiency during the optimization process is shown in figure 25.



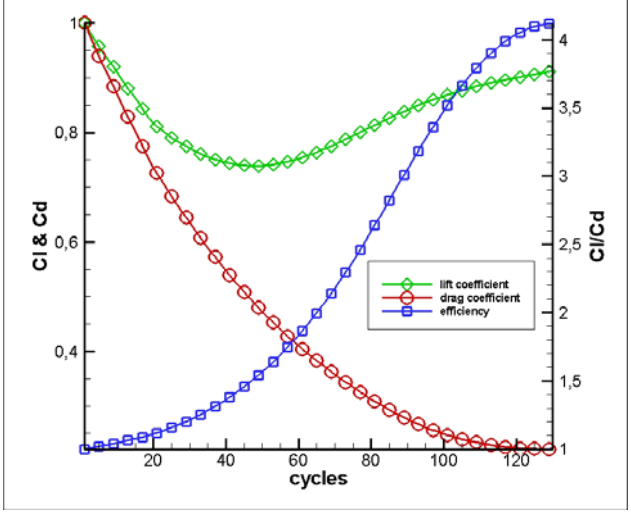
Evolution of the lift (green) and drag (red) coefficients and efficiency (blue) during the optimization process

Fig. 25: Optimization results for the objective function (29)

The second objective function is taking *N=2*, *M=2*.

$$Min\left\{\left(\frac{C_d}{C_d^{\,o}}\right)^2 + \left(\frac{C_L - C_L^{\,T}}{C_L^{\,o}}\right)^2\right\}$$
$$\delta = -2\cdot\left\{\frac{\dot{c}_d C_d}{\left(C_d^{\,o}\right)^2} + \frac{\dot{c}_L\left(C_L - C_L^{\,T}\right)}{\left(C_L^{\,o}\right)^2}\right\} \qquad \textbf{(30)}$$

The evolution of the drag and lift coefficients is displayed in figure 26.

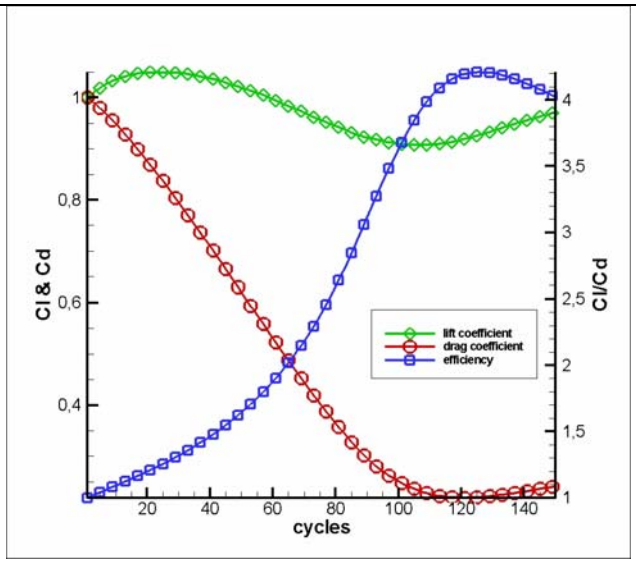


Evolution of the lift (green) and drag (red) coefficients and efficiency (blue) during the optimization process

Fig. 26: Optimization results for the objective function (30)

The third objective function is taken from equation (28) with *N=1*, and *M=2*.

$$Min\left\{\frac{C_L^{\,o}}{C_d^{\,o}}\cdot\frac{C_d}{C_L}+\left(\frac{C_L-C_L^{\,T}}{C_L^{\,o}}\right)^2\right\}$$
$$\delta=-\left\{\frac{C_L^{\,o}}{C_d^{\,o}}\cdot\frac{\dot{c}_d C_L-\dot{c}_L C_d}{C_L^{\,2}}+\frac{2\dot{c}_L\left(C_L-C_L^{\,T}\right)}{\left(C_L^{\,o}\right)^2}\right\} \qquad \textbf{(31)}$$



Evolution of the lift (green) and drag (red) coefficients and efficiency (blue) during the optimization process

Fig. 27: Optimization results for the objective function (31)

In these examples, the optimization does not stop when a maximum efficiency is achieved, because typically the lift should also be into a range, which could be 10% from the original lift.

All optimizations lead to a similar profile and there are no noticeable differences in the optimized geometry obtained using the described objective functions. All simulations converge to a maximum efficiency at similar times, and the drag is quickly reduced to about 80%. But the behavior of the optimizations with respect the lift is different.

In the optimization performed by equation (29) the drag is quickly reduced but it took some cycles to recover the lift into the required range. On the other hand, using the deformation function in equation (30), the drag sensitivities are multiplied by the actual drag coefficient, so there is potentially a better correction of the lift coefficient as the drag value decreases.

Nevertheless, it seems that in these objective functions the arbitrary parameter should be manipulated to force the lift into a range. Finally the optimization performed by equation (31) behaves better than the previous ones, obtaining a higher efficiency while the final lift coefficient is close to its original value.

7 Conclusions and future work

This work has presented an efficient tool for CAD based shape aerodynamic design in inviscid configurations. This tool is fully integrated with the TAU code and is capable to optimize by considerably weakening the shock wave. Next short-term step is to include the tangential components to the sensitivity formulation expecting to increase the accuracy of the gradients.

Future work will extend the framework to viscous configurations and analyze the influence of the distribution of the NURBS control points in order to improve the convergence on the pursuit of finding a global optimal design. In addition, the point inversion algorithm will be

further studied to improve its efficiency. On the other hand, aspects related to the curvature, volume and mechanical constraints will be applied to avoid unrealistic profiles.

Acknowledgments

The research described in this paper made by INTA and DLR researchers has been supported under the INTA activity Termofluidodinámica (INTA's code: IGB99001), by the Spanish Ministry of Education and Science (MEC) under the DOMINO Project (CIT-370200-2007-22) and the DOVRES Project (partially funded also by AIRBUS Spain), and under the high performance activity for the TAU code at $C^2A^2S^2E$ at the Institute of Aerodynamics and Flow Technology in Braunschweig. In particular, this work has been possible due to the close cooperation between the two organizations.

Copyright Statement

References

[1] The Advisory Council for Aeronautics Research in Europe (ACARE). *A vision for 2020.* January 2001.

[2] Jameson A., *Aerodynamic Shape optimization using Adjoint Method*, VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamic, Rhode St Genese, Belgium.

[3] Elliot J., and Peraire J. *Practical three-dimensional aerodynamic design by optimization*. AIAA Journal, Vol. 35, No. 9, 1997, pp. 1479-1485.

[4] Jameson A., Alonso J. J., Reuther J. J., Martinelli L., and Vassberg J. C. *Aerodynamic shape optimization echniques based on control theory*. AIAA Paper 98-2538.

[5] Mavriplis, D. *Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Meshes.* AIAA Journal, Vol. 44, Jan. 2006, pp. 42–50.

[6] Anderson W.K., Venkatakrishnan. *Aerodynamic design optimization on unstructured grids with a continuous adjoint formularion*. AIAA, Aerospace Sciences Meeting & Exhibit, 35th, Reno, NV, Jan. 6-9, 1997.

[7] Nadarajah S. K., Jameson A.. *Optimun Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjont Methods*. AIAA Journal Vol. 45, No. 7, July, 2007.

[8] Lepine J., Trepanier J. *Wing Aerodynamic Design Using an Optimized NURBS Geometrical Representation.* 38th Aerospace Science Meeting and Exibit, 10-13 January 2000, Reno, NV.

[9] Bentamy A., Guibault F., and Trepanier J. *Aerodynamic Optimization of a Realistic Aircraft Wing*, AIAA 2005-332.

[10] Mousavi A., Castonguay P. and Nadarajah S.. *Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization.* 18th AIAA Computational Fluid Dynamics. 25 - 28 June 2007, Miami.

[11] *Technical Documentation of the DLR TAU-Code.* Tech. rep., Institut of Aerodynamics and Flow Technology, 1994.

[12] CentaurSoft, *http://www.centaursoft.com.*

[13] Piegl L., Tiller W. *The NURBS book*. Springer-Verlag Berlin Heidelberg New York, 1995 and 1997, ISBN 3-540-61545-8.

[14] Hu S.M., Wallner J. *A second order algorithm for ortogonal projection onto curves and surfaces.* Computer Aided Geometric Design 22 pp. 251-260 (2005)

[15] Liang Y.M., Hewitt W.T. *Point inversion and projection for NURBS curve and surface: Control polygon approach.* Computer Aided Geometric Design 20 pp- 79-99 (2003).

[16] Selimovic I., *Improved algorithms for the projection of points on NURBS curves and surfaces.* Computer Aided Geometric Design, pp. 439-445 (2006).

[17] Redondo D. *Inversion de punto sobre un panel NURBS.* Universidad Carlos III de Madrid. Feb. 2009.

[18] Castro C., Lozano C., Palacios F. and Zuazua E.. *A Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids.* 44th AIAA Aerospace Sciences Meeting and Exhibit 9 - 12 January 2006, Reno, Nevada.

[19] W.K. Anderson, V. Venkatakrishnan. *Aerodynamic Design Optimization on Unstructured Grids with Continuous Adjoint Formulation.* AIAA 97-0643, (1997).

[20] Castro C., Lozano C., Palacios F., and Zuazua E. *A Systematic Formulation of the Continuous Adjoint Method Applied to Viscous Aerodynamic Design.* ERCOFACT 2006 Design Optimisation: Methods & Applications, (2006).

[21] Gerhold T., Galle M., Friedrichs O., and Evans J.., *Calculation of Complex Three-Dimensional Configurations employing the DLR TAU-Code*. Aiaa-97-0167, AIAA, 1997.

[22] Lain, K. R., Klausmeyer, S. M., Zickuhr, T., Vassberg, J. C., Wahls, R. A., Morrison, J. H., Brodersen, O., Rakowitz, M. E., Tinoco, E. N., and Godard, J.-L., *Data Summary from Second AIAA Computational Fluid Dynamics Drag Prediction Workshop.* Journal of Aircraft, Vol. 42, No. 5, 2005, pp. 1165.

[23] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Lain, K. R., and Mavriplis, D. J.,, *Summary of the Third AIAA CFD Drag Prediction Workshop.* AIAA paper 2007-0260, AIAA, Reno, Nevada, January 2007.

[24] Lozano C. *DOVRES Task 2.1/INTA3 Final Report: Assessment of Mesh Sensitivities in Continuous Adjoint Gradients for Aerodynamic Shape Optimization in the TAU Code.* INTA technical report December 2009.