2D Navier-Stokes Shape Design Using a Level Set Method

Antonio Baeza∗
Madrid Institute for Advanced Studies in Mathematics, Madrid, 28049, Spain

Carlos Castro†
Universidad Politécnica de Madrid, Madrid, 28040, Spain

Francisco Palacios‡
Madrid Institute for Advanced Studies in Mathematics, Madrid, 28049, Spain

Enrique Zuazua§
Universidad Autónoma de Madrid, Madrid, 28049, Spain

Optimal aerodynamic shape design aims at finding the minimum of a functional by controlling the PDE modeling the flow using surface deformation techniques. As a solution to the enormous computational resources required for classical shape optimization of functionals of aerodynamic interest, probably the best strategy is to use in a systematic way, methods inspired in control theory. One of the key ingredients relies on the use of the adjoint system techniques to simplify the computation of gradients.

In the present paper we will restrict our attention to optimal shape design in systems governed by the Navier-Stokes equations. We first review some standard facts on control theory applied to optimal shape design, and recall the Navier-Stokes equations in aerodynamical problems. We then study the adjoint formulation, providing a detailed exposition of how the derivatives of functionals may be obtained. Some relevant numerical issues will also be discussed. Finally, recent computational results applying a Level Set methodology are reviewed.

I. Introduction

I.A. Control theory applied to aerodynamical optimal shape design

In the last decades, Optimal Shape Design (OSD) has evolved very close to the CFD developments. In the early seventies, CFD began with some algorithms and software for computing flow past a given profile, modeling with potential flows. At the same time inverse design problems (finding the profile corresponding to a specified target pressure distribution) were also in front of the technology. Progress in CFD made a first attempt at using numerical optimization techniques in the late seventies. In those works first applications using very simplified approximations to the flow equations were combined with optimization algorithms.

In the early eighties the Dassault/INRIA group performed transonic potential flow calculations for a complete aircraft and posed the basis to optimal design of 3D configurations.

By the eighties, advances in computer hardware and algorithms had made feasible to solve the full Euler equations. Roe’s introduction of the concept of locally linearizing the equations through a mean value Jacobian had a great impact. In the middle eighties it was possible to calculate the Euler flow past a complete transport aircraft. At the same time, O. Pironneau had also investigated the problem of optimum shape design for elliptic equations using control theory. Some of the groundbreaking works in this field are

∗Research Scientist, iMath Project.
†Assistant Professor, Departamento de Matemáticas e Informática, ETSI Caminos, Canales y Puertos.
‡Director of Technological Innovation, IMDEA-Mathematics, AIAA Member.
§Professor, Departamento de Matemáticas, Facultad de Ciencias. Director, IMDEA-Mathematics.
due to J.-L. Lions. In the late eighties A. Jameson was the first to apply these techniques to the Euler and Navier-Stokes equations in the field of aeronautical applications.

Classical aeronautical applications of optimal shape design in systems governed by PDEs are formulated on a fluid domain $\Omega$, usually air, delimited by disconnected boundaries divided into a “far field” $\Gamma_\infty$ and several solid wall boundaries $S$, usually airplane surfaces (see Fig. 1).

This kind of design problems is aimed at minimizing a functional $J$ of the flow (e.g. drag, lift or a defined pressure distribution) defined on the boundaries $S$, $\Gamma_\infty$, or in a subdomain $\omega \subset \Omega$. From now on we will restrict ourselves to the analysis of optimization problems involving functionals defined on the solid wall $S$, whose value depends on the flow variables $U$ obtained solving the Navier-Stokes equations. In this context, the generic optimization problem can be succinctly stated as follows

$$J(S^{\text{min}}) = \min_{S \in S_{\text{ad}}} J(S),$$

where $S^{\text{min}}$ is the sought optimal surface belonging to the set $S_{\text{ad}}$ of admissible boundary geometries and

$$J(S) = \int_S j(U) \, ds,$$

is the objective function, whose evaluation is subject to the resolution of the flow equations to obtain $U$.

![Figure 1. Classical optimal design problem with 3 solid wall surfaces](image)

Let us consider an arbitrary (but small) perturbation of the boundary $S$ which, without loss of generality, can be parameterized by an infinitesimal deformation of size $\delta S$ along the normal direction

$$S' = \{ \bar{x} + \delta S (\bar{x}) \, \vec{n}_S (\bar{x}) , \, \bar{x} \in S \}.$$  

where $\vec{n}_S$ is the outward normal pointing vector to the surface $S$ defined by $\bar{x}$ In the continuous framework, assuming a regular flow solution $U$, the variation of the functional $J$ under the deformation can be evaluated as:

$$\delta J(S) = \int_{\delta S} j(U) \, ds + \int_S j'(U) \delta U \, ds,$$  

where the first term, which stems from the displacement of the boundary, takes the form

$$\int_{\delta S} j(U) \, ds = \int_S (\partial_n j - \kappa j) \delta S \, ds,$$

to lowest order in $\delta S$, where $\kappa$ is the curvature of $S$ (in the two-dimensional case). The second term in (4) is the contribution due to (infinitesimal) changes in the flow solution induced by the deformation.

The most expensive terms (in terms of time and required computational resources) are those which involve the computation of $\delta U$. These can be obtained by solving the linearized flow equations for each independent deformation (design variable). If the design space is large, as is the case in real applications, the computational cost is prohibitive. It is then convenient to switch to the control theory approach, which reduces significantly the computational cost of getting the gradients, using the adjoint or dual formulation of the shape design problem.
II. 2D Navier-Stokes Equations

The Navier-Stokes equations\textsuperscript{7,8} provide a complete (dynamical) description of a viscous fluid and expresses the conservation of mass, momentum and energy. The complete system of equations (without source terms and assuming adiabatic boundary conditions at the solid wall) can be written in the following conservative form:

\[
\begin{aligned}
\frac{\partial U}{\partial t} + \nabla \cdot \left( \vec{F}(U) - \vec{F}^v(U, T) \right) &= 0, \\
\vec{v}|_{S} &= 0, \\
\partial_n T|_{S} &= 0,
\end{aligned}
\]

(6)

where, at the “far field” boundary \(\Gamma_\infty\), boundary conditions are specified for incoming waves, while outgoing waves are determined by the solution inside the fluid domain.

In (6), \(U^T = (\rho, \rho v_x, \rho v_y, \rho E)\) are the conservative variables, \(T\) is the temperature and \(\vec{F} = (F_x, F_y)\) is the convective flux vector

\[
F_x = \begin{pmatrix}
\rho v_x \\
\rho v_x^2 + \rho v_y v_y \\
\rho v_x v_y \\
\rho v_x H
\end{pmatrix}, \quad F_y = \begin{pmatrix}
\rho v_y \\
\rho v_x v_y \\
\rho v_y^2 + \rho v_x v_x \\
\rho v_y H
\end{pmatrix},
\]

(7)

where \(\rho\) is the fluid density, \(\vec{v} = (v_x, v_y)\) is the flow speed in a Cartesian system of reference, \(E\) is the total energy, \(P\) the system pressure and \(H\) the enthalpy. On the other hand the viscous flux \(\vec{F}^v = (F^v_x, F^v_y)\) is given by

\[
F^v_x = \begin{pmatrix}
0 \\
\sigma_{xx} \\
\sigma_{xy} \\
\sigma_{xx} v_x + \sigma_{xy} v_y + k \partial_x T
\end{pmatrix}, \quad F^v_y = \begin{pmatrix}
0 \\
\sigma_{yx} \\
\sigma_{yy} \\
\sigma_{yx} v_x + \sigma_{yy} v_y + k \partial_y T
\end{pmatrix},
\]

(8)

where \(\sigma_{ij} = \mu (\partial_i v_j + \partial_j v_i) - (2/3) \mu (\nabla \cdot \vec{v}) \delta_{ij}\) is the shear stress tensor, the dynamic viscosity \(\mu = \mu(T)\) and the thermal conductivity \(k = k(\mu)\) being obtained from empirical relations (Sutherland’s law and Eucken’s formula, respectively). The system of equations (6) must be completed by an equation of state which defines the thermodynamic properties of the fluid. For a perfect gas:

\[
P = (\gamma - 1) \rho \left[ E - \frac{1}{2} \left( v_x^2 + v_y^2 \right) \right], \quad H = E + \frac{P}{\rho},
\]

(9)

where \(\gamma = \frac{C_p}{C_v} \approx 1.4\) for standard air conditions.

III. Continuous adjoint formulation

In gradient-based optimization techniques, the goal is to minimize a suitable cost or objective function with respect to a set of design variables (defining, for example, an airfoil profile or aircraft surface). Minimization is achieved by means of an iterative process which requires the computation of the gradients or sensitivity derivatives of the cost function with respect to the design variables.

Gradients can be computed in a variety of ways, the simplest method is based in the linearization of the cost function completed with the linearization of the governing flow equations for evaluating flow variables variations. The main drawback of this classical approach is that one linearized flow equations must be solved per design variable, so if the design variables are numerous, the computation could be prohibitive.

A very efficient method to compute the cost function gradient is the adjoint method, which allows the solution of general sensitivity analysis problems governed by fluid dynamics models. At the computational level there are two approaches to the adjoint system, the continuous method and the discrete one: In the continuous approach, the adjoint equations are derived from the Navier-Stokes equations and then subsequently discretized, whereas in the discrete approach the adjoint equations are directly derived from the discretized governing equations.\textsuperscript{9,10}
There is a trade-off between the complexity of the adjoint discretization one uses and the accuracy in the estimation of the gradients that one obtains. However, in some cases, owing to the lack of differentiability of the numerical scheme, the discrete approach can not be applied directly.

III.A. Treatment of non-differentiable numerical schemes (continuous vs. discrete adjoint)

Given a field governed by the Navier-Stokes equations which can be discretized by using a differentiable method and which has regular solutions, both the continuous and discrete adjoint methods provide a way of computing the gradient of a functional which depends on the solution of the system for a set of design variables. Moreover, in simple problems involving scalar conservation laws, such as Burgers’ equation (both viscous and inviscid), it is possible to prove, using ideas of Γ-convergence, that the global minima of the discrete functional associated with convergent numerical approximation schemes converge to the minima of the continuous one, as the mesh-size tends to zero. In more complex scenarios, a general proof has not been given.

One more issue is of interest. The adjoint equations are linear equations, so they can be solved by techniques which are simpler than those used for the direct problem, which is non-linear. However, in aeronautical applications, the use of the same numerical schemes for solving both equations is widespread.

On the other hand, efficient aeronautical fluid dynamics numerical schemes (Roe, JST, ...) are often not differentiable and, consequently, the functional to be optimized depends on the solution and on the design parameters in a non-differentiable way. In a strict sense, it is therefore impossible to build gradient-based descent algorithms upon such schemes, since the gradient of the functional actually does not exist.

In practice, dealing with non-differentiable numerical schemes, several tricks have been proposed in the literature to make things work at the computational level:

- To freeze the non-differentiable terms of central schemes with high-order artificial viscosity when using the discrete adjoint approximation.
- To develop pseudo-linearized models for Godunov schemes.
- To use the continuous approach as a shortcut. Note, however, that one should always be careful when doing that since the goal is to minimize a discrete functional. At this respect it is also important to note that comparing values of “gradients” of different functionals, or of different discrete approximations of the same continuous functional, is a delicate matter, since very different functionals may have very close minima, their gradients being significantly different.

III.B. 2D steady adjoint Navier-Stokes equations

We are going to limit our study to steady Navier-Stokes equations. We consider an optimization problem defined by the cost function \( J(U) \) subject to the steady Navier-Stokes equations

\[
\begin{cases}
\partial_x F_x(U) + \partial_y F_y(U) & - (\partial_x F_{x}^{y}(U, \partial_x U, \partial_y U) + \partial_y F_{y}^{y}(U, \partial_x U, \partial_y U)) = 0 \quad \text{in } \Omega, \\
\vec{v} = 0, & \partial_n T = 0 \quad \text{on } S.
\end{cases}
\]  

(10)

The objective is to evaluate the variation of the functional \( J(U) \) under shape changes of the surface \( S \). Since the reference flow is smooth (viscous flows do develop shock waves, but these are not singularities of the flow as in the Euler case), the variation \( \delta J \) of the functional is computed as in (4).

Let us introduce the notation

\( \vec{n}_S \) and \( \vec{n}_{\Gamma_{\infty}} \) are the inward-pointing unit vectors normal to \( S \) and \( \Gamma_{\infty} \) (respectively),

\( \vec{t}_S \) and \( \vec{t}_{\Gamma_{\infty}} \) are the \( \pi/2 \) counter-clock-wise rotations of \( \vec{n}_S \) and \( \vec{n}_{\Gamma_{\infty}} \) (respectively),

\( \partial_n = \vec{n} \cdot \nabla \) is the normal derivative,

\( \partial_{tg} = \vec{t} \cdot \nabla \) is the tangential derivative,

Linearizing the Navier-Stokes equations (10) about a given steady-state solution \( U \)

\[
\partial_x (A_{x} \delta U) + \partial_y (A_{y} \delta U) \\
- (\partial_x (A_{x}^{y} \delta U + D_{xx} \partial_x (\delta U)) + \partial_y (A_{y}^{y} \delta U + D_{yy} \partial_y (\delta U)) = 0
\]  

(11)
where $\delta U$ is the linear perturbation, and the spatially varying matrices are defined by

\[
\begin{align*}
A_x &= \frac{\partial F_x}{\partial U} \bigg|_{U(x,y)} \quad A_y = \frac{\partial F_y}{\partial U} \bigg|_{U(x,y)} \quad A_v = \frac{\partial F^v}{\partial U} \bigg|_{U(x,y)} \quad A^v = \frac{\partial F^v}{\partial U} \bigg|_{U(x,y)}, \\
D_{xx} &= \frac{\partial F^v}{\partial (\partial_x U)} \bigg|_{U(x,y)} \quad D_{xy} = \frac{\partial F^v}{\partial (\partial_y U)} \bigg|_{U(x,y)} \quad D_{yy} = \frac{\partial F^v}{\partial (\partial_y U)} \bigg|_{U(x,y)} \quad D_{yy} = \frac{\partial F^v}{\partial (\partial_y U)} \bigg|_{U(x,y)}.
\end{align*}
\]

Using the linearized Navier-Stokes equations, it is possible to compute the value of $\delta U$ for each design variable (surface modification):

\[
\begin{align*}
\mathbf{\nabla} \cdot (\mathbf{A} \delta U) \\
&= \nabla \cdot (\mathbf{A} \delta U) \\
&= -\partial_x (D_{xx} \delta U) + D_{xy} \delta y (\delta U)) \\
&- \partial_y (D_{yx} \delta U) + D_{yy} \delta y (\delta U)) = 0, \quad \text{in } Q, \\
\delta u|_S &= -\delta S \partial_u, \quad \delta v|_S = -\delta S \partial_v, \\
\bar{n}_S \cdot \nabla T|_S &= -\delta \bar{n} \cdot \nabla T - n_1 \delta x \partial_j \partial_j T, \quad \text{on } S, \\
(\delta W)_+ &= 0, \quad \text{on } \Gamma_\infty.
\end{align*}
\]

where $(\delta W)_+$ represents the incoming characteristics on the “far field” boundary (we suppose that the far field boundary conditions are constants). $\delta S$, which parameterizes infinitesimal deformations along the normal direction —see (3)—, is an input datum to the design problem. In practice, $\delta S$ has to be directly realized by means of the admissible design variables thus making impossible arbitrary deformations.

In order to eliminate $\delta U$, the adjoint problem is introduced through the Lagrange multipliers $\Psi^T = (\psi_1, \psi_2, \psi_3, \psi_4)$. The method of Lagrange multipliers is a very powerful technique for calculating the minimum of a constrained multi-variate function, the constraints being in this case the Euler equations. Then, the linearized Navier-Stokes equations multiplying by the Lagrange multipliers $\Psi$ and integrating over the whole domain $\Omega$ obtaining, after appropriate simplifications, the following expression

\[
\begin{align*}
- \int_\Omega \mathbf{\nabla} \psi^T \cdot (\mathbf{A} - \mathbf{A}^v) \delta U \, d\Omega \\
&+ \int_\Omega \mathbf{\nabla} \cdot (\mathbf{\partial x} \psi^T, \mathbf{\partial y} \psi^T) \cdot \begin{pmatrix} D_{xx} & D_{xy} \\
D_{yx} & D_{yy} \end{pmatrix} \delta U \, d\Omega \\
&+ \int_{\partial \Omega} \psi^T \left( \mathbf{A} - \mathbf{A}^v \right) \delta U \, dS - \int_{\partial \Omega} \psi^T \begin{pmatrix} D_{xx} & D_{xy} \\
D_{yx} & D_{yy} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{\partial x} (\delta U) \\
\mathbf{\partial y} (\delta U) \end{pmatrix} \, dS \\
&+ \int_{\partial \Omega} (\mathbf{\partial x} \psi^T, \mathbf{\partial y} \psi^T) \cdot \begin{pmatrix} D_{xx} & D_{xy} \\
D_{yx} & D_{yy} \end{pmatrix} \delta U \, dS = 0.
\end{align*}
\]

Let us now analyze each term in (13) separately, in order to simplify this expression and to obtain the adjoint system that allows the direct calculation of the variation of the functional.

- The adjoint equations are extracted from the requirement the first two terms of (13) vanish. This yields

\[
- \left( \mathbf{A} - \mathbf{A}^v \right)^T \cdot \mathbf{\nabla} \psi + \mathbf{\nabla} \cdot \left( \begin{pmatrix} D_{xx}^T & D_{xy}^T \\
D_{yx}^T & D_{yy}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{\partial x} \psi \\
\mathbf{\partial y} \psi \end{pmatrix} \right) = 0.
\]

- The boundary terms supported on the “far field” can be eliminated with the appropriate boundary conditions as in the Euler case, while the contribution on the solid wall, after expanding the matrices in (13) gives

\[
\begin{align*}
\int_S (\bar{n}_S \cdot \delta \bar{v}) (\rho \psi_1 + \rho H \psi_4) \, ds \\
- \int_S (\psi_4 \bar{n}_S \cdot \nabla \delta \bar{v} - \bar{n}_S \cdot \nabla \delta \bar{v}) \, ds \\
+ \int_S (k (\partial_n \psi_4) \delta T - k \psi_4 \partial_n (\delta T)) \, ds + \int_S (\bar{n}_S \cdot \nabla (\delta P - \bar{n}_S \cdot \delta \sigma \cdot \delta \varphi)) \, ds = 0,
\end{align*}
\]
where
\[ \Sigma_{ij} = \mu (\partial_i \varphi_j + \partial_j \varphi_i) - \frac{2}{3} \mu (\nabla \cdot \varphi) \delta_{ij}, \quad \delta k = \frac{C_p \delta \mu}{Pr}, \]
and no-slip (zero velocity) and adiabatic wall boundary conditions have been assumed on \( \Gamma \).

From the expression (13) it is possible to determine which objective functions are admissible for the computation of gradients with the adjoint method.\(^{16}\) In particular, \( j = j(\vec{f}, T) \) is an allowed functional, where \( \vec{f} = \rho \vec{u} - \vec{n}_{\Sigma} \cdot \sigma \), which requires the following boundary conditions on the solid wall:
\[
\begin{align*}
\Psi_2 &= \frac{\partial j}{\partial \Gamma_{eq}}, \\
\Psi_3 &= \frac{\partial j}{\partial T}, \\
\kappa \partial_n \Psi_4 &= \frac{\partial j}{\partial T}.
\end{align*}
\] (17)
The corresponding variation of the functional is:
\[ \delta J = \delta \int_{\Gamma} j \left( \vec{f}, T \right) dS = \int_{\Gamma} j \left( \vec{f}, T \right) dS - I_{eq}, \] (18)
where
\[ I_{eq} = \int_S (\vec{n}_S \cdot \delta \vec{v}) (\rho \psi_1 + \rho H \psi_4) - \psi_4 \vec{n}_S \cdot \sigma + \delta \vec{v} + k \psi_4 \partial_n (\delta T)) dS, \]
\[ \delta \vec{v} \big|_S = -\delta S \partial_n \vec{v}, \]
\[ \vec{n}_S \cdot \nabla \delta T \big|_S = -\delta S \partial_n \left( \nabla T \right) \cdot \vec{n} + (\partial_n \delta S) \left( \nabla T \right) \cdot \vec{t}_S. \] (19)

### III.C. Discretization of the adjoint equations

The adjoint equations have been discretized with a standard edge-based finite volume formulation on the dual grid,\(^{18-20}\) obtained by applying the integral formulation of the adjoint equations to a dual grid control volume \( \Omega_j \) surrounding any given node \( j \) of the grid and performing exact integration around the outer boundary of this control volume. The viscous part is discretized using linear, centered approximations, as explained below, see section III.C.1. We focus on the discretization of the convective part. Using the divergence theorem
\[ \int_{\Omega_j} \frac{d\Psi_j}{dt} = \int_{\Gamma_j} \Psi \vec{n}_S dS = \int_{\Omega_j} \frac{d\Psi_j}{dt} - \sum_{k=1}^{m_j} \vec{f}_{jk} \cdot \vec{n}_{jk} S_{jk} = 0, \] (20)
where \( \vec{A}_S^T \) is the (transposed) Euler Jacobian evaluated at the node \( j \), \( \Gamma_j \) is the boundary of \( \Omega_j \), \( |\Omega_j| \) is the area (volume in 3D) of \( \Omega_j \) and, for every neighbor node \( k \) of \( j \), \( \vec{n}_{jk} \) is the outward unit vector normal to the face of \( \Gamma_j \) associated with the grid edge connecting \( j \) and \( k \) and \( S_{jk} \) is its area (or length in 2D), \( \vec{f}_{jk} \) is the numerical flux vector at the said face, \( \Psi_j \) is the value of \( \Psi \) at the node \( j \) (it has been assumed that \( \Psi_j \) is equal to its volume average over \( \Omega_j \)), and \( m_j \) is the number of neighbors of the node \( j \).

The solution is advanced in time using a first order (forward Euler) scheme:
\[ \Psi^{n+1}_j = \Psi^n_j - \frac{\Delta t}{|\Omega_j|} \sum_{k=1}^{m_j} \vec{f}_{jk} \cdot \vec{n}_{jk} S_{jk} = 0, \] (21)
and a multistage Runge-Kutta was used in the computations.

Next, we will review several alternative schemes for defining the numerical flux vector.

#### III.C.1. Central scheme with artificial dissipation

In the present work, we have developed a central scheme inspired by the standard JST scheme,\(^{13}\) following the adaptation to unstructured flow solvers presented in.\(^{21}\) In our scheme, the numerical (convective) flux is computed as
\[ f_{jk}^{cent} = f_{jk} \cdot \vec{n}_{jk} = A_{jk}^T \left( \frac{\Psi_j + \Psi_k}{2} \right) + d_{jk}, \] (22)
The adjoint Navier-Stokes equations for compressible flows can be formulated as

\[ III.C.3. \text{ Adjoint viscous terms} \]

\[ f = \epsilon^{(4)}_j (\nabla^2 \Psi_j - \nabla^2 \Psi_k^s) \Phi_{jk} \lambda_{jk}, \]  

(23)

where \( \nabla^2 \) denotes the undivided Laplacian

\[ \nabla^2 \Psi_j = -m_j \Psi_j + \sum_{k=1}^{m_j} \Psi_k, \]  

(24)

\( \epsilon^{(4)}_j \) are user-defined constants, and \( \lambda_{jk} \) is the local spectral radius

\[ \lambda_{jk} = (|\vec{u}_{jk} \cdot \vec{n}_{jk}| + c_{jk}) S_{jk}, \]  

(25)

where \( \vec{u}_{jk} = \frac{\vec{u}_j + \vec{u}_k}{2} \) and \( c_{jk} = \frac{c_j + c_k}{2} \) denote the flow speed and sound speed at the face, respectively. \( \Phi_{jk} \) is introduced to account for the stretching of the mesh cells and is defined as \( \Phi_{jk} = 4 \frac{\Phi_i \Phi_k}{\Phi_i + \Phi_k} \) where \( \Phi_j = \frac{\lambda_j}{4 \lambda_{jk}} \).

### III.C.2. Roe’s upwind scheme

In addition to the central scheme presented above, an upwind scheme based upon Roe’s flux difference splitting scheme\textsuperscript{12,22} has been developed for the adjoint equations.

In our case, the aim is to use an upwind type formulas to evaluate a flow of the form \( \vec{A}^T \cdot \vec{\nabla} \Psi \). Taking into account that \( A^T = -(P^T)^{-1} \Lambda P^T \), where \( A^T = \vec{A}^T \cdot \vec{n} \) is the projected Jacobian matrix, \( \Lambda \) is diagonal matrix of eigenvalues and \( P \) is the corresponding eigenvector matrix, the upwind flux is computed as

\[ f^{\text{upw}}_{jk} = \frac{1}{2} \left( \vec{A}^T \cdot \vec{n}_{jk} (\Psi_j + \Psi_k) - |\vec{A}^T_{jk}| (\Psi_k - \Psi_j) \right) = \frac{1}{2} \left( \vec{A}^T_k (\Psi_j + \Psi_k) + (P^T)^{-1} |\Lambda| P^T \delta \Psi \right), \]  

(26)

where \( |\vec{A}^T_{jk}| = -(P^T)^{-1} |\Lambda| P^T \), the subindex \( jk \) denotes an average value in the interface. Note that \( f^{\text{upw}}_{jk} \neq f^{\text{upw}}_{kj} \).

### III.C.3. Adjoint viscous terms

The adjoint Navier-Stokes equations for compressible flows can be formulated as

\[ \partial_t \Psi - \vec{A}^T \cdot \vec{\nabla} \Psi + \left[ (\vec{A}^v)^T \cdot \vec{\nabla} \Psi - \vec{\nabla} \cdot \vec{F}^v \right] = 0, \]  

(27)

where \( \vec{A} \) and \( \vec{A}^v \) are, respectively, the Jacobian matrices of the convective and viscous fluxes. On the other hand, the adjoint viscous fluxes have the following expression:

\[ F^v_x = D^T_{xx} \partial_x \Psi + D^T_{yx} \partial_y \Psi, \]
\[ F^v_y = D^T_{xy} \partial_x \Psi + D^T_{yy} \partial_y \Psi. \]  

(28)

To discretize the viscous part we proceed as follows:

- The numerical evaluation of the term \( (\vec{A}^v)^T \cdot \vec{\nabla} \Psi \) is made by adding a source term, as follows:

\[ \int_{\Omega_j} (\vec{A}^v)^T \cdot \vec{\nabla} \Psi \, d\Omega = \int_{\partial \Omega_j} \vec{\nabla} \Psi \vec{n} \, dS \equiv \frac{1}{|\Omega_j|} \sum_{k=1}^{m_j} \frac{1}{2} (\Psi_k + \Psi_j) \vec{n}_{jk} S_{jk}, \]  

(29)

where the gradient \( \vec{\nabla} \Psi_j \) is computed with a suitable numerical method such as the Green-Gauss approach\textsuperscript{20}

- The numerical computation of the viscous adjoint flux \( \vec{F}^v \) requires the evaluation of the gradient of the adjoint variables on the dual grid cell faces. For a face associated to an edge \( ij \) which connects nodes \( i \) and \( j \), the gradient is computed as follows

\[ \vec{\nabla} \Psi_{ij} = \frac{1}{2} \left( \vec{\nabla} \Psi_i + \vec{\nabla} \Psi_j \right). \]  

(31)
III.D. Control stabilization in optimal shape design

The shape design procedure consists in an iterative optimization process that, in each step, requires the solution of an adjoint equation in order to find the steepest descent direction and the subsequent modification of the surface shape according to the obtained result. However, if the discrete dynamics or numerical approximation scheme generates spurious solutions that do not exist at the continuous level, the optimization iterative process may diverge. Various remedies have been developed to avoid those instabilities. In particular the efficiency of the two-grid algorithm proposed by Glowinski\textsuperscript{23} has been recently proved in.\textsuperscript{24} The method consists in, when computing the descent direction, considering only the solutions of the adjoint system corresponding to slowly oscillating data coming from a coarser mesh. In practice this has the effect of relaxing the control criteria or the functional to be minimized.

The bigrid strategy is a powerful numerical tool oriented to damping or filtering the spurious numerics in controls, therefore variations in it practical implementation are possible:

1. Solving the adjoint system in a coarse grid and then, extrapolating the numerical result to a finest grid.
2. Solving the adjoint system in a fine grid but using as inputs, results that have been previously filtered in a coarse grid level.
3. And Finally, by computing the control in the finest grid and making a bigrid filtering postprocess over the results.

In this section we have tested the need and efficiency of applying this two-grid technique in the shape design using Euler equations. We have selected the adimensional lift force $C_L$ as the cost function to be optimized, and the $xy$ position of each discrete node over the surface as design variables.

![Figure 2. Control of $\min C_L$ using different mesh sizes (left). Flow velocity over the designed surface (right)](image)

We first immerse, into a inviscid flow, a 2D shape, whose $C_L$ surface sensitivity is computed by solving the adjoint equation. This allows performing the first optimization iteration. In Fig. 2 different controls are shown for different numerical discretizations. Three different meshes have been used, the finest one having 400 nodes over the control surface, while the coarsest mesh has 100 nodes. It is possible to check that for the finest mesh the control presents very important high-frequency oscillations that may cause the divergence of the control process.

In Fig. 2 the flow velocity over the control surface is represented. It is possible to check that the discontinuity in Fig. 2 is located at the stagnation point, where bigger gradients are obtained in the coarsest mesh due to the lack of precision in the gradients computed for the adjoint variables.

In Fig. 3 two convergence strategies are shown, the first one consists in using a bigrid strategy to eliminate the high frequency spurious error in the control by filtering, over the solid surface, the computed functional...
sensitivity. The second one consists in solving the adjoint equation in a mesh coarser than the one used for solving the direct problem. In view of the obtained results, the direct application of a bigrid filtering strategy applied to the control is probably the best solution for controlling the whole problem.

As a further example, we consider an optimization problem in which the goal is the maximization of the lift force coefficient of a profile NACA 0012 defined by two different mesh discretizations (Fig. 4).

The aim is to control the Euler system by varying the shape of the airfoil in order to optimize the lift functional. First a coarse mesh is used; in Fig. 5 the controls computed at three controlling/optimization steps are shown. The behavior is smooth and the system is in the process of converging.

However, if the finest mesh is used (see Fig. 6 (left)), the control diverges almost immediately. In the second iteration, the control has completely diverged and the optimization process completely fails. In Fig. 6 (right) a bigrid filtering strategy is applied to the computed functional sensitivity over the surface (control), it is important point out that the bigrid algorithm has been applied to the functional sensitivity computed in the fine mesh.

In this case, the control process has become stable, and a good solution for the shape optimization problem can be found.

IV. Practical shape design

With the aim of finding the surface shape which minimizes an aeronautical functional, the most used algorithms are those based on the classical gradient methods (steepest descent, conjugate gradient, etc.).
Figure 5. Euler Equations controlled in a coarse discretization

Figure 6. Euler Equations controlled in a fine discretization (left). Euler Equations controlled in a fine discretization (bigrand strategy) (right)
In recent years there has been intensive work done in adapting Level Set methods to deal with optimal shape design problems. The Level Set method\textsuperscript{25–28} has been originally developed to describe/predict the behavior of interfaces. In this section we sketch its possible application in the context of aeronautical optimal design. The method consists in viewing shapes as level sets of a moving function that evolves searching the minimizer of the functional under consideration. Obviously one of the main ingredients in developing a correct Level Set methodology is determining the interface velocity that necessarily depends on the adjoint and flow equations as the most classical methods we have discussed above.

This section is also devoted to connect the cost function gradient (computed via adjoint methodology) with the aeronautical shape optimization through the introduction of the Level Set strategy to move the solid boundary which is being designed.

IV.A. Evolution of shapes using a Level Set method

Traditionally, the interface between a solid shape and the surrounding air was described from a Lagrangian point of view, in which the reference frame surrounds the interface in movement and the coordinate system moves along the interface. But the Level Set method is based on the description of the interfaces through an Eulerian point of view in which the reference frame (coordinate system) remains fixed in space and the interface travels through the control volumes.

Within the scope of optimization involving fluid dynamics problems, this methodology is used in such a way that the interface is the separation between the solid body and the surrounding fluid. This interface will move with a normal speed that comes from the computed gradients on the surface.

The main drawback of Lagrangian procedures is the following: if in the course of the optimization process any two given surfaces come into contact at some point, the Lagrangian approach does not provide a solution (the new geometry in which the two geometric objects melt in one), while the Eulerian approach is able to do it.

In Level Set theory, we embed the initial position of the front in which we are interested, as the zero level set of a higher-dimensional function \( \phi \), and we define an evolution equation for \( \phi \), namely

\[
\phi_t + F |\nabla \phi| = 0, \quad t > 0,
\]

where \( F \) is the level set velocity in the whole domain \( \Omega \) and for certain forms of the speed function \( F \), one obtains a standard Hamilton-Jacobi equation. This point of view introduces two great advantages with respect to the classical treatment of interfaces:

- Topological changes in the front \( \Gamma \) are developed in a natural way. The front position at time \( t \) is given by \( \phi (x, y, t) \). So, due to the interface evolution, it is possible that more than one curve appears as time advances.

- Another important aspect is that, in industrial applications, this method has the same structure for 2D and 3D cases.

In Short, the Level Set method applied to an optimization problem describes the movement of interfaces from an Eulerian point of view through the definition of a Hamilton-Jacobi support equation (32) in an evolution problem. The speed of the interface (zero level set) is derived from the gradient of the functional and it is extrapolated to the whole domain using different techniques.

The practical/numerical aspects of an optimization using Level Sets is delicate due to the great amount of computational tools that must be used and the way in which they must interrelate.

The basic optimization algorithms involved in the Level Set methodology are schematically shown in Fig. 7. We will now outline the main steps of the developed algorithm:

1. On the initial geometric configuration, the fluid flow equations must be solved using the appropriate discretization.

2. After the resolution of the direct problem, the adjoint problem is posed and solved. As a result of the resolution of the adjoint problem, the sensitivity of the functional under geometric variations is obtained.
Figure 7. Level Set optimization procedure
3. The gradient of the functional is combined with the geometric restrictions and a non-dimensionalization technique is used to obtain the speed of advance of the interface which is independent of the computational mesh.

4. The Level Set structure is generated and the Level Set is advanced in time according to the calculated speeds using a support mesh which characteristics are best suited for solving the Level Set problem.

5. Lastly, the final geometry of the Level Set is recovered and is used to build a new mesh with adapted properties for the solution of the fluid flow equations, such as anisotropic mesh zone, boundary layer, etc.

These five steps are to be repeated in a closed loop until the desired convergence level is obtained. Combining all these methods (meshing, equation solution, Level Set evolution) makes the robustness of the Level Set application complex. However, as will be seen later on, its many possible applications compensate certain initial difficulties in the implementation.

IV.B. Aerodynamical application

In this section we will present the most relevant expressions for optimization problems with aeronautical applications governed by the Navier-Stokes equations.

When using the Navier-Stokes equations, the optimizing functional in the case of calculating drag or lift forces must include the effect of the viscous forces and has the following form

$$\int_S \left( C_p\vec{n} - \frac{1}{C_\infty}\vec{n} \cdot \sigma \right) \cdot \vec{d} ds,$$

(33)

where the adiabatic wall boundary condition is assumed on $S$. In these circumstances, the adjoint variables must verify the following conditions on the solid boundary

$$\begin{cases} 
\varphi|S = \frac{1}{C_\infty}\vec{d}, \\
\partial_n\psi_4|S = 0.
\end{cases}$$

(34)

Finally, the complete variation of the drag or lift functionals is

$$\delta \left( \int_S \left( C_p\vec{n} - \frac{1}{C_\infty}\vec{n} \cdot \sigma \right) \cdot \vec{d} ds \right) = \int_S G\delta S \, ds,$$

(35)

and

$$G = (\vec{n} \cdot \partial_n\vec{v})(\rho\psi_1 + \rho H\psi_4) + \vec{n} \cdot \Sigma \cdot \partial_n\vec{v}$$

$$-\psi_4(\vec{n} \cdot \sigma \cdot \partial_n\vec{v}) + \psi_4(\sigma : \vec{\nabla} \vec{v}) - k(\partial_{tg}\psi_4)(\partial_{tg}T).$$

(36)

In the derivation of (36), the reduction of the higher order terms (involving second derivatives of quantities such as the temperature $T$ that appear in the original formulation), along the lines explained in, has been performed. This procedure is necessary in order to avoid the need to use a flow solver with higher than second order accuracy.

V. Level Set numerical optimization experiments

In this section, a shape control problem using a Level Set strategy is shown. The optimization of a non-conventional configuration will be made using the Level Set strategy. In Fig. 8 the most important optimization cases in which the Level Set approach suits best are presented:

- First, this technique is appropriate in those circumstances in which the parameterization of the surface is complex and hard to perform.

- The Level Set strategy is also valid in those geometries with edges or intersections that with great probability will cause the crossing of surfaces in the course of the optimization process.
Finally, this technique is very convenient in optimization problems in which there exist numerous of immersed bodies, each of which will probably evolve until touching or merging with all or a part of the others.

With the aim of testing the Level Set functionality, an unconventional configuration consisting of 7 small squares immersed in a fluid is presented. The flow conditions are Mach number 0.2, angle of attack 2.5° and a Reynolds number of 10 which guarantees a steady laminar flow around the bodies surrounded by the fluid. The objective function to optimize is the quotient between the lift $C_L$ and drag $C_D$ coefficients. In Fig. 9 the evolution of the optimization process is shown.

This optimization process is an interesting application of the Level Set methodology to external aerodynamic shape design and illustrates the capacity of designing bodies whose geometry is complex and difficult to parameterize.

The bigrid filtering strategy has been used to stabilize the control problem. Some interesting features are the merging of the two outermost bodies on both sides, the sharpness of the resulting first and last shapes and the formation of “S”-shaped channels between each of the blocks. These are only some of the interesting geometric phenomena that can be observed in this non-conventional optimization approach.

In Fig. 10 the evolution of the Level Set is displayed. In the first 25 iterations the efficiency has improved from 0.0722 to 0.2875. With 10 additional iterations the efficiency reaches a value of 0.3421 with a highest value of 0.7411 for the lift coefficient.

Another optimization example is shown in Fig. 12. In this case, the initial configuration is a rectangle, the objective function is the drag minimization, and two constraints are used: lift coefficient greater than 0.5, and the pitch moment must be between −0.02 and 0.02. Finally, the speed of the interface is computed
Figure 10. Visual evolution of the Level Set optimization

Figure 11. Evolution of adimensional coefficients
as:

\[
speed = C_d' - 4.0\epsilon_1 C_l' (\text{MAX}(C_{lmin} - C_l, 0.0))^3 \\
+ 4.0\epsilon_2 C_m' \left( (\text{MAX}(-C_{mmax} + C_m, 0.0))^3 - (\text{MAX}(C_{mmin} - C_m, 0.0))^3 \right),
\]

where in order to fulfill all the constrains, instead of choosing the exact value for \(C_{lmin}, C_{mmin}\) and \(C_{mmax}\) slightly greater values have been selected. On the other hand \(\prime\) is used to denote the derivative with respect to an infinitesimal movement of the surface.

The selected flow conditions are Mach number equal to 0.25, zero angle of attack and the Reynolds number is 500. In Fig. 11 the history of the adimensional parameters convergence is shown. In 20 iterations all the constraints are fulfilled and the drag has been diminish from 0.2404 to 0.2239

**VI. Conclusions**

In this work the continuous adjoint methodology for the calculation of gradients of functionals of the flow (defined on the surface) has been developed for the Navier-Stokes equations. Also, several issues concerning optimal design problems have been quantitatively established and applications of the Level Set method have been provided.

Since, in practice, one has to work with discrete numerical approximation schemes of PDEs, the most natural way to obtain the gradients of the objective function would appear to be through a discrete adjoint method in which adjoint equations are directly derived from the linearized discrete flow equations. Nevertheless, the use of this methodology requires making certain assumptions concerning the differentiability of the
numerical schemes used, that are not generally verified, because the best-suited methods for the resolution of the Euler equations are non-differentiable.

On the other hand, the continuous adjoint methodology derives the adjoint problem from the continuous formulation of the flow equations, and as such it constitutes a shortcut that allows to maintain the rigor throughout the whole procedure.

Finally, in this work we have also prove the utility of Level Set methods applied to optimal shape design problems in external flows.

References


