# Continuous adjoint approach for the Spalart–Allmaras model in aerodynamic optimization

Alfonso Bueno-Orovio*
*Oxford University, Oxford OX1 3QD, UK, and Polytechnic University of Madrid, Madrid 28040, Spain*
Carlos Castro†
*Polytechnic University of Madrid, Madrid 28040, Spain*
Francisco Palacios‡
*Stanford University, Stanford CA 94305, USA*
Enrique Zuazua§
*Basque Center for Applied Mathematics, Derio 48160, Spain*

**In this paper we describe and develop the continuous adjoint method to compute shape sensitivities in aerodynamic design with turbulence modeling. We focus on compressible flows described by the RANS equations and the classical Spalart–Allmaras model to describe turbulence. Turbulence modeling usually requires, in particular, to compute the distance to the surface. Here, this distance is incorporated to the system as a new variable solving the Eikonal equation.**

## I. Introduction

This paper contains a complete formulation of the adjoint approach for the shape optimization of an obstacle with boundary $S$ immersed in a fluid governed by the Reynolds Averaged Navier–Stokes (RANS) equations. These equations are the result of expressing flow quantities in a turbulent regime as the sum of random fluctuations over small times scales about a steady or slowly varying mean flow, and averaging the Navier–Stokes equations. This yields a set of equations for the mean flow that are formally identical to the standard Navier–Stokes equations, but incorporating two extra unknown terms, the Reynolds stress and the turbulent heat flux. Further averages of the RANS equations will lead to additional unknowns at each level, being unable to produce a balance between the number of unknowns and the number of equations. In order to close the system, the most extended approach is based on the coupling of a turbulent model,[33] which allows to approximate the Reynold stress and the turbulent heat flux terms. This gives rise to a rich variety of turbulence models fitting to different fluid regimes. Among these, we focus on the Spallart–Allmaras one,[29] due to its wide use in industrial applications of aerodynamics. As a result, a system of partial differential equations (PDEs), suitable for numerical simulation at a relatively low cost, but still retaining significant properties of the turbulent flow, is obtained.

Shape optimization methods have grown in importance in aerodynamic design within the last decade. In gradient-based optimization techniques, the goal is to minimize a suitable cost or objective function (drag coefficient, deviation from a prescribed surface pressure distribution, etc.) 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.

*Research Assistant, Computing Laboratory, Computational Biology Group, and Departamento de Matemáticas e Informática, ETSI Caminos, Canales y Puertos.
†Professor, Departamento de Matemáticas e Informática, ETSI Caminos, Canales y Puertos.
‡Engineering Research Associate, Department of Aeronautics and Astronautics, AIAA Member.
§Ikerbasque Research Professor & Scientific Director, Basque Center for Applied Mathematics (BCAM).

Gradients can be computed in a variety of ways, the most actively pursued one being the adjoint method,[2,4,13,24] due, among other factors, to their ability of computing these derivatives at a cost comparable to that of numerically solving the state PDEs. Adjoint methods are conventionally divided into continuous and discrete. In the continuous approach, the adjoint equations are derived from the governing PDE and then subsequently discretized, whereas in the discrete approach the adjoint equations are directly derived from the discretized governing equations.

The continuous adjoint approach allows understanding the physical significance of the adjoint equations and boundary conditions. It also has the advantage that the adjoint system has an unique form independent of the scheme used to solve the flow-field system, and hence offers flexibility in choosing the discretization scheme for the adjoint system. However, this may result in discrepancies in the gradient of the discretized objective function. On the other hand, the discrete adjoint provides the "exact" gradient of the discretized objective function, and should give gradients which are closer in value to those computed by finite differences. It also ensures that the optimization process can fully converge, although it suffers from increased memory requirements and development costs may increase noticeably, especially in case of high-order discretizations with extended stencils.

From a practical point of view, a nontrivial question appears when computing the distance to the surface, required by the model. In this work we solve this problem by characterizing this distance as a solution of the Eikonal equation in the domain. Thus, the final system of equations incorporates the Eikonal equation, taking part of the continuous formulation of the problem. The analysis with other turbulence models can be derived similarly.

To the best of authors' knowledge, the only previous contribution concerning such a continuous adjoint approach to the RANS equations in conjunction to a turbulence model is the work of Zymaris *et al.*[34] This work, however, is devoted to the incompressible version of the Navier–Stokes equations, and restricted to interior flows in duct geometries. Besides, only sensitivities regarding the total pressure loss functional in the duct are considered. Other few similar works take into account variations in turbulent viscosity, although they are all based on the discrete adjoint approach: Nielsen *et al*,[22] Dwight *et al*,[7] Anderson and Bonhaus,[1] Lee *et al*,[20] Mavriplis[21] and Kim *et al*.[18] We refer to the introduction in Zymaris *et al*[34] for a more detailed description on the previous literature on this subject.

The organization of the paper is as follows. In §II we describe the model and state the optimization problem. In §III the continuous adjoint method to compute sensitivities of the RANS equations, coupled with the Spalart–Allmaras turbulence model, is derived in a general framework. The practical implementation of the method is described in §IV. In §V some numerical experiments will be presented illustrating the relevance of the developments described in this work. Some details on the sensitivity calculus and the explicit formulas are given in the appendix §VII at the end of this work.

## II. Description of the problem

The Navier–Stokes equations[19,32] (see §VII.A) express the conservation of mass, momentum and energy in a viscous fluid. Classical aeronautical applications assume that the air is governed by these Navier–Stokes equations on a domain $\Omega \subset \mathbb{R}^3$, delimited by disconnected boundaries divided into a "far field" $\Gamma_\infty$, and adiabatic walls that we denote by $S$. Their steady-state formulation (without source terms) can be written in the following form:

$$\begin{cases} R_U(U,\hat{\nu}) = \nabla\cdot\vec{F}^c - \nabla\cdot\left(\mu^1_{tot}\vec{F}^{v1} + \mu^2_{tot}\vec{F}^{v2}\right) = 0 & \text{in } \Omega, \\ \vec{v} = 0 & \text{on } S, \\ \partial_n T = 0 & \text{on } S, \\ (W)_+ = W_\infty & \text{on } \Gamma_\infty, \end{cases} \tag{1}$$

where $U = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho E)^\intercal$ stands for the vector of conservative variables, $\rho$ is the density, $E$ is the energy, $\vec{v} = (v_1, v_2, v_3) \in \mathbb{R}^3$ is the flow speed in a Cartesian system of reference, and $T$ is the temperature. The last equation in (1) represents classical far field boundary conditions simulating the fluid behavior at infinity. The vectors $\vec{F}^c(U) = (\vec{F}^c_1, \vec{F}^c_2, ..., \vec{F}^c_5)^\intercal$ are the convective fluxes and $\vec{F}^{vk}(U) = (\vec{F}^{vk}_1, \vec{F}^{vk}_2, ..., \vec{F}^{vk}_5)^\intercal$, $k = 1, 2$ are the viscous fluxes, in where we have considered separately the contribution of the viscous forces and the heat flux transfer. Expressions for $\mu^1_{tot}$ and $\mu^2_{tot}$ are given in (3) below. Here $(\cdot)^\intercal$ denotes transposition.

As usual in turbulent modeling based upon the Boussinesq hypothesis, which states that the effect of turbulence can be represented as an increased viscosity, the viscosity is divided into a laminar $\mu_{dyn}$ and a turbulent $\mu_{tur}$ component. The laminar or dynamic viscosity is usually taken to be only dependent on the temperature, $\mu_{dyn} = \mu_{dyn}(T)$, whereas $\mu_{tur}$ is obtained from a suitable turbulence model involving the flow and a set of new variables $\hat{\nu}$, i.e., $\mu_{tur} = \mu_{tur}(U, \hat{\nu})$. Here we assume that $\hat{\nu}$ is a single scalar variable obtained from a one-equation turbulence model, which can be written, in a general form, as

$$\begin{cases} R_{\hat{\nu}}(U, \hat{\nu}, d_S) = \nabla \cdot \vec{T}^{cv} - T^s = 0 & \text{in } \Omega, \\ \hat{\nu} = 0 & \text{on } S, \\ \hat{\nu}_\infty = \sigma_\infty \nu_\infty & \text{on } \Gamma_\infty, \end{cases} \tag{2}$$

where $\vec{T}^{cv} = \vec{T}^{cv}(U, \hat{\nu}, d_S)$ stands for the convective and viscous terms, and $T^s = T^s(U, \hat{\nu}, d_S)$ is used for the source term, $d_S$ being the distance to the boundary of the obstacle $S$. In the particular case of the Spalart–Allmaras model the convective, viscous and source terms are given in §VII.B. For fully turbulent calculations, the far field boundary condition for the turbulent viscosity is to impose some fraction of the laminar viscosity at the far field,[29] where $\sigma_\infty$ is some turbulence model constant, usually ranging between 3 and 5. On viscous walls $\hat{\nu}$ is set to zero, corresponding to the absence of turbulent eddies very near to the wall.

Once the turbulent viscosity has been computed, turbulence is coupled to the main stream flow by replacing the dynamic viscosity in the momentum and energy equations with

$$\mu^1_{tot} = \mu_{dyn} + \mu_{tur}, \qquad \mu^2_{tot} = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{tur}}{Pr_t} \tag{3}$$

where $Pr_d$ and $Pr_t$ are respectively the dynamic and turbulent Prandtl numbers.

Note that in (2) we have incorporated the distance variable, $d_S$, which is common in turbulence modeling and, in particular, it appears in the considered Spalart–Allmaras model. The new variable, $d_S(S)$, solves the so-called Eikonal equation

$$\begin{cases} R_d(d_S) = |\nabla d_S|^2 - 1 = 0 & \text{in } \Omega, \\ d_S = 0 & \text{on } S. \end{cases} \tag{4}$$

Systems (1), (2) and (4), together with a suitable equation of state to describe the fluid thermodynamics, constitute a complete system of equations and boundary conditions for the flow variables.[25,33]

A key element for the definition of a optimal shape design problem is the objective function. In this case, we introduce a objective function which is assumed to only depend on the values of the flow variables at the boundary $S$. As it was shown in,[4] for the Navier–Stokes system only objective functions depending on $\vec{f}$ and the temperature $T$ are allowed for continuous adjoint optimization, with $\vec{f}$ given by

$$\vec{f} = (f_1, f_2, f_3) = P\vec{n} - \bar{\sigma} \cdot \vec{n}, \qquad \bar{\sigma} = \mu^1_{tot} \bar{\tau} \tag{5}$$

where $\vec{n}$ denotes the exterior normal to the surface $S$, $P$ is the pressure of the fluid and $\bar{\sigma}$ the second order tensor of viscous stresses, $\bar{\tau}$ given in §VII.A. Note that this includes, in particular, functionals depending only on the pressure $P$ since it can be written as a function of $\vec{f}$, using the fact that $\vec{n} \cdot \bar{\sigma} \cdot \vec{n} = 0$ on the boundary.[4] More precisely,

$$P = \vec{n} \cdot (P\vec{n} - \bar{\sigma} \cdot \vec{n}) = \vec{f} \cdot \vec{n} \qquad \text{on } S. \tag{6}$$

In the presence of turbulence modeling a similar situation appears, the only difference being that we can now add dependence on the new turbulence variable $\hat{\nu}$ in terms of its unknown value on the boundary, i.e. $\partial_n \hat{\nu}$, since $\hat{\nu} = 0$ on $S$. Thus, we will consider the following general choice of objective function

$$J(S) = \int_S j(\vec{f}, T, \partial_n \hat{\nu}, \vec{n})\, ds. \tag{7}$$

## III. Variation of the objective function: the adjoint approach

As usual in the adjoint approach, flow equations are incorporated to the cost functional as constraints by means of a Lagrange multiplier for each equation $\Psi_U^\intercal = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5), \psi_{\hat{\nu}}, \psi_d$. In this way, the

Lagrangian reads

$$\mathcal{J}(S) = \int_S j(\vec{f}, T, \partial_n\hat{\nu}, \vec{n})\, ds + \int_\Omega \left(\Psi_U^\mathsf{T} R_U(U,\hat{\nu}) + \psi_{\hat{\nu}} R_{\hat{\nu}}(U,\hat{\nu},d_S) + \psi_d R_d(d_S)\right) d\Omega. \tag{8}$$

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 to the surface $S$. The new surface obtained after the deformation is then given by

$$S' = \{\vec{x} + \delta S\, \vec{n},\ \vec{x} \in S\} \tag{9}$$

where, for small deformations, the following holds[28]

$$\begin{cases} \delta\vec{n} = -\nabla_S(\delta S) \\ \delta(ds) = -2H_m \delta S\, ds \end{cases} \tag{10}$$

where $H_m$ is the mean curvature of $S$ computed as $(\kappa_1+\kappa_2)/2$, and $(\kappa_1,\kappa_2)$ are curvatures in two orthogonal directions on the surface. Here $\nabla_S$ represents the tangential gradient operator on $S$, defined as a $\mathbb{R}^3$ vector with null normal component.

Assuming a regular flow solution $U$ and a smooth boundary $S$, the variation of the functional $J$ under the deformation can be evaluated as

$$\delta\mathcal{J} = \int_S \delta j(\vec{f}, T, \partial_n\hat{\nu}, \vec{n})\, ds + \int_{\delta S} j(\vec{f}, T, \partial_n\hat{\nu}, \vec{n})\, ds + \int_\Omega \left(\Psi_U^\mathsf{T} \delta R_U(U,\hat{\nu}) + \psi_{\hat{\nu}} \delta R_{\hat{\nu}}(U,\hat{\nu},d_S) + \psi_d \delta R_d(d_S)\right) d\Omega \tag{11}$$

where, using the convention of summation of repeated indexes $i = 1,2,3$, we have

$$\begin{aligned} \delta j(\vec{f}, T, \partial_n\hat{\nu}, \vec{n}) &= \frac{\partial j}{\partial f_i}\delta f_i + \frac{\partial j}{\partial T}\delta T + \frac{\partial j}{\partial(\partial_n\hat{\nu})}\delta(\partial_n\hat{\nu}) - \frac{\partial j}{\partial\vec{n}}\cdot\nabla_S(\delta S) \\ &= \frac{\partial j}{\partial\vec{f}}\cdot(\delta P\vec{n} - \delta\bar{\sigma}\cdot\vec{n}) + \frac{\partial j}{\partial T}\delta T + \frac{\partial j}{\partial(\partial_n\hat{\nu})}\delta(\partial_n\hat{\nu}) \\ &\quad - \left(\frac{\partial j}{\partial\vec{n}} - \frac{\partial j}{\partial\vec{f}}P - \frac{\partial j}{\partial\vec{f}}\cdot\bar{\sigma}\right)\cdot\nabla_S(\delta S) \end{aligned} \tag{12}$$

$$\int_{\delta S} j(\vec{f}, T, \partial_n\hat{\nu}, \vec{n})\, ds = \int_S \left(\frac{\partial j}{\partial f_i}\partial_n f_i + \frac{\partial j}{\partial T}\partial_n T + \frac{\partial j}{\partial(\partial_n\hat{\nu})}\partial_n^2\hat{\nu} - 2H_m j\right)\delta S\, ds \tag{13}$$

and $\delta R_U$, $\delta R_{\hat{\nu}}$ and $\delta R_d$ represent the variations of $R_U$, $R_{\hat{\nu}}$ and $R_d$ respectively. Note that in (12) we have written the variation $\delta\vec{f}$ in terms of $\delta P$ and $\delta\bar{\sigma}$ and we have used formula (10) for $\delta\vec{n}$. The variations $\delta P\vec{n} - \delta\bar{\sigma}\cdot\vec{n}$, $\delta T$ and $\delta(\partial_n\hat{\nu})$ appearing in (12) can be computed from the following linearized system

$$\begin{cases} \delta R_U(U,\hat{\nu}) = \dfrac{\partial R_U}{\partial U}\delta U + \dfrac{\partial R_U}{\partial\hat{\nu}}\delta\hat{\nu} = 0 & \text{in } \Omega, \\ \delta\vec{v} = -\partial_n\vec{v}\,\delta S & \text{on } S, \\ \partial_n(\delta T) = (\nabla T)\cdot\nabla_S(\delta S) - \partial_n^2 T\delta S & \text{on } S, \\ (\delta W)_+ = 0 & \text{on } \Gamma_\infty. \end{cases} \tag{14}$$

$$\begin{cases} \delta R_{\hat{\nu}}(U,\hat{\nu},d_S) = \dfrac{\partial R_{\hat{\nu}}}{\partial U}\delta U + \dfrac{\partial R_{\hat{\nu}}}{\partial\hat{\nu}}\delta\tilde{\nu} + \dfrac{\partial R_{\hat{\nu}}}{\partial d_S}\delta d_S = 0 & \text{in } \Omega, \\ \delta\hat{\nu} = -\partial_n\hat{\nu}\,\delta S & \text{on } S, \\ \delta\hat{\nu} = \sigma_\infty\delta\nu & \text{on } \Gamma_\infty. \end{cases} \tag{15}$$

$$\begin{cases} \delta R_d(d_S) = \nabla d_S\cdot\nabla\delta d_S = 0 & \text{in } \Omega, \\ \delta d_S = -\delta S & \text{on } S. \end{cases} \tag{16}$$

where $(\delta W)_+$ represents the incoming characteristics on the "far field" boundary. In the last formula of (16) we have used the fact that $\partial_n d_S = 1$ and therefore, $\delta d_S = -\partial_n d_S \delta S = -\delta S$.

As usual, domain integrals in (11) are eliminated using integration by parts and introducing the associated adjoint operators. The integration by parts also provides some boundary terms. These boundary terms are combined with those boundary terms in (11), depending on $\delta P\vec{n} - \delta\sigma\cdot\vec{n}$, $\delta T$ and $\delta(\partial_n\hat{\nu})$, through the boundary conditions for the adjoint operators. We describe this process below.

From (14)-(16), the last three terms in (11) read

$$\int_\Omega \Psi_U^\mathsf{T}\left(\frac{\partial R_U}{\partial U}\delta U + \frac{\partial R_U}{\partial\hat{\nu}}\delta\hat{\nu}\right) + \int_\Omega \psi_{\hat{\nu}}\left(\frac{\partial R_{\tilde{\nu}}}{\partial U}\delta U + \frac{\partial R_{\hat{\nu}}}{\partial\hat{\nu}}\delta\hat{\nu} + \frac{\partial R_{\hat{\nu}}}{\partial d_S}\delta d_S\right) + \int_\Omega \psi_d\frac{\partial R_d}{\partial d_S}\delta d_S =$$
$$\int_\Omega \left(A_U^U\Psi_U + A_{\hat{\nu}}^U\psi_{\hat{\nu}}\right)^\mathsf{T}\delta U + \int_\Omega \left(A_U^{\hat{\nu}}\Psi_U + A_{\hat{\nu}}^{\hat{\nu}}\psi_{\hat{\nu}}\right)\delta\hat{\nu} + \int_\Omega \left(A_{\hat{\nu}}^d\psi_{\hat{\nu}} + A_d^d\psi_d\right)\delta d_S + \int_S B_S\,ds \quad (17)$$

where $A_U^U = \left(\frac{\partial R_U}{\partial U}\right)^\mathsf{T}$, $A_{\hat{\nu}}^U = \frac{\partial R_{\hat{\nu}}}{\partial U}$, $A_U^{\hat{\nu}} = \left(\frac{\partial R_U}{\partial\hat{\nu}}\right)^\mathsf{T}$, $A_{\hat{\nu}}^{\hat{\nu}} = \frac{\partial R_{\hat{\nu}}}{\partial\hat{\nu}}$, $A_{\hat{\nu}}^d = \frac{\partial R_{\hat{\nu}}}{\partial d_S}$ and $A_d^d = \frac{\partial R_d}{\partial d_S}$ are the adjoint operators and $B_S$ stands for the boundary terms coming from the integration by parts

$$\begin{aligned}\int_S B_S\,ds &= \int_S \vec{\varphi}\cdot(\delta P\vec{n} - \delta\bar{\sigma}\cdot\vec{n})\,ds + \int_S (g_1\cdot\vec{\varphi} + g_2\partial_n\psi_5 - \psi_{\hat{\nu}}g_5)\delta T\,ds + \int_S g_3\psi_{\hat{\nu}}\delta(\partial_n\hat{\nu})\,ds \\ &\quad + \int_S \psi_{\hat{\nu}}g_4\delta P\,ds - \int_S \psi_d\delta S\,ds + \int_S \hat{g}\delta S\,ds \end{aligned} \quad (18)$$

where $\vec{\varphi} = (\psi_2,\psi_3,\psi_4)$ and $\vec{g}_1$, $g_i$, $i = 2,3,4,5$ are some functions that do not depend on the adjoint variables $\Psi_U, \psi_{\hat{\nu}}, \psi_d$, and $\hat{g}$ which does not depend on $\psi_d$. The analytical expression of these terms and the adjoint operators above is given in detail in (99)-(108) in Appendix §VII.

We now observe that, due to the relation (6), the first and third terms on the right hand side of (18) can be written together. In fact, the linearization of (6) provides

$$\delta P = \delta(P\vec{n}\cdot\vec{n} - \vec{n}\cdot\bar{\sigma}\cdot\vec{n}) = \vec{n}\cdot(\delta P\vec{n} - \delta\bar{\sigma}\cdot\vec{n}) - 2(P\vec{n} - \vec{n}\cdot\bar{\sigma})\cdot\nabla_S(\delta S) \quad \text{on } S, \quad (19)$$

where we have used the symmetry of the tensor $\bar{\sigma}$ and the formula for $\delta\vec{n}$ in (10). Therefore, formula (18) can be written as

$$\begin{aligned}\int_S B_S\,ds &= \int_S (\vec{\varphi} + \psi_{\hat{\nu}}g_4\vec{n})\cdot(\delta P\vec{n} - \delta\bar{\sigma}\cdot\vec{n})\,ds + \int_S (g_1\cdot\vec{\varphi} + g_2\partial_n\psi_5 - \psi_{\hat{\nu}}g_5)\delta T\,ds + \int_S g_3\psi_{\hat{\nu}}\delta(\partial_n\hat{\nu})\,ds \\ &\quad - \int_S \psi_d\delta S\,ds + \int_S \hat{g}\delta S\,ds - 2\int_S \psi_{\hat{\nu}}g_4(P\vec{n} - \vec{n}\cdot\bar{\sigma})\cdot\nabla_S(\delta S)\,ds. \end{aligned} \quad (20)$$

In order to eliminate domain integrals in (11), when replacing the last three terms by using (17), we assume that the adjoint variables satisfy

$$0 = A_U^U\Psi_U + A_{\hat{\nu}}^U\psi_{\hat{\nu}} \quad (21)$$
$$0 = A_U^{\hat{\nu}}\Psi_U + A_{\hat{\nu}}^{\hat{\nu}}\psi_{\hat{\nu}} \quad (22)$$
$$0 = A_{\hat{\nu}}^d\psi_{\hat{\nu}} + A_d^d\psi_d. \quad (23)$$

Analogously, all boundary terms in (17) without explicit dependence on $\delta S$ can be eliminated by considering the following choice of boundary conditions for the adjoint variables

$$\psi_{\hat{\nu}} = -\frac{1}{g_3}\frac{\partial j}{\partial(\partial_n\hat{\nu})} \quad \text{on } S, \quad (24)$$
$$\varphi_i = -\frac{\partial j}{\partial f_i} - \psi_{\hat{\nu}}g_4 n_i \quad \text{on } S, \quad (25)$$
$$\partial_n\psi_5 = -\frac{1}{g_2}\left(\frac{\partial j}{\partial T} - \vec{g}_1\cdot\vec{\varphi} + \psi_{\hat{\nu}}g_5\right) \quad \text{on } S, \quad (26)$$
$$\psi_d = 0 \quad \text{on } S. \quad (27)$$

Note that this choice for the boundary conditions must be done in order, i.e. the value of $\psi_{\hat{\nu}}$ on the right hand side of (25) and (26) is obtained from (24) while the value of $\vec{\varphi}$ in (26) is computed from (25).

Combining (11)-(13), (17)-(20), the adjoint equations (21)-(23) and the boundary conditions in (24)-(27), we finally obtain

$$\begin{aligned}\delta\mathcal{J} &= \int_S \left(\frac{\partial j}{\partial f_i}\partial_n f_i + \frac{\partial j}{\partial T}\partial_n T + \frac{\partial j}{\partial\left(\partial_n\hat{\nu}\right)}\partial_n^2\hat{\nu}\right)\delta S\, ds \\ &\quad - \int_S \left(\frac{\partial j}{\partial \vec{n}} + \frac{\partial j}{\partial \vec{f}}P - \frac{\partial j}{\partial \vec{f}}\cdot\bar{\sigma}\right)\cdot\nabla_S(\delta S)\, ds - \int_S (\hat{g}+2H_m j)\delta S\, ds \\ &\quad -2\int_S \psi_{\hat{\nu}} g_4(P\vec{n}-\vec{n}\cdot\bar{\sigma})\cdot\nabla_S(\delta S)\, ds. \end{aligned} \tag{28}$$

In this expression, the adjoint variables are obtained by solving the closed system of partial differential equations and boundary conditions given by (21)-(27).

It is important to recall here that neither the Navier–Stokes (21) nor the Spalart–Allmaras (22) adjoint equations depend on the adjoint distance variable $\psi_d$. This is also the case of the functional sensibility (28), since typical objective functionals in aerodynamics do not depend explicitly on the distance to the surface. In this situation, it is therefore not necessary to solve (23), and the adjoint system simply reduces to (21)-(22). One could consider, however, more sophisticated situations that would require the solution of (23). This could be, for instance, the case of functionals with an explicit dependence on the distance to the surface by means of a domain integral.

Some particular but still interesting situations provide a simplified formula for the variation of $J$, as described in[4] for Navier–Stokes. Assume that the objective function depends only on $\vec{f}$ in the following way

$$j(\vec{f}) = \vec{f}\cdot\vec{d} \tag{29}$$

where $\vec{d}$ is a constant vector (the choice $\vec{d}=\vec{n}$ is also possible with some modifications, but for simplicity we focus on constant $\vec{d}$). Note that this is the case in drag or lift optimization problems. The adjoint boundary conditions in this situation simply become

$$\psi_{\hat{\nu}} = \psi_d = 0, \quad \vec{\varphi} = -\vec{d}, \quad \partial_n\psi_5 = \vec{g}_1\cdot\vec{d}/g_2 \tag{30}$$

and the variation of $J$ is given by

$$\delta\mathcal{J} = \int_S \vec{d}\cdot\partial_n f_i\delta S\, ds - \int_S \left(P\vec{d}-\vec{d}\cdot\bar{\sigma}\right)\cdot\nabla_S(\delta S)\, ds - \int_S (\hat{g}+2H_m j)\delta S\, ds. \tag{31}$$

Integrating now by parts, and assuming that either $S$ is smooth or $\delta S = 0$ at its singular points, one yields

$$\begin{aligned}\delta\mathcal{J} &= \int_S \partial_n(P\vec{d}\cdot\vec{n}-\vec{d}\cdot\bar{\sigma}\cdot\vec{n})\delta S\, ds + \int_S \nabla_S\cdot\left(P\vec{d}-\vec{d}\cdot\bar{\sigma}\right)\delta S\, ds - \int_S (\hat{g}+2H_m j)\delta S\, ds \\ &= \int_S \nabla\cdot(P\vec{d}-\vec{d}\cdot\bar{\sigma})\delta S\, ds - \int_S \hat{g}\delta S\, ds = -\int_S \hat{g}\delta S\, ds. \end{aligned} \tag{32}$$

Here we have used the fact that the divergence operator, on local coordinates of $S$, is given by

$$\nabla\cdot\vec{q} = \partial_n(\vec{q}\cdot\vec{n}) + \nabla_S\cdot\vec{q} - 2H_m\vec{q}\cdot\vec{n} \tag{33}$$

for a general vector field $\vec{q}$, and the identity

$$\nabla\cdot(P\vec{d}-\vec{d}\cdot\bar{\sigma}) = (\nabla P - \nabla\cdot\bar{\sigma})\cdot\vec{d} = 0 \qquad \text{on } S, \tag{34}$$

which is obtained assuming that the momentum equations in the Navier–Stokes system are satisfied on the boundary, i.e. $\nabla P = \nabla\cdot\bar{\sigma}$ on $S$.

The final expression in (32) involves the function $\hat{g}$, which as shown in §VII.E is reduced to $\hat{g}=h_3$ due to the boundary conditions in (30). In this way, the total variation of the functional finally comes given by the simplified expression

$$\delta\mathcal{J} = -\int_S h_3\delta S\, ds = \int_S (\vec{n}\cdot\bar{\Sigma}^{\varphi}\cdot\partial_n\vec{v} - \mu_{tot}^2 C_p\nabla_S\psi_5\cdot\nabla_S T)\,\delta S\, ds \tag{35}$$

with $\bar{\Sigma}^{\varphi}$ depending on the gradient of the adjoint variables $\vec{\varphi}$.

## IV. Numerical implementation of the adjoint equations

An appropriate rearrangement of terms in the vector-matrix multiplications in (21)-(23) leads to a more compact formulation for the evaluation of the residuals of the adjoint system. Such a formulation, more suitable for a practical implementation of the adjoint equations is presented as follows.

### A. Spatial discretization

A finite-volume discretization is used to solve both the direct and adjoint equations. As usual, the finite-volume discretization is obtained by applying the integral formulation of the governing equations to a control volume $\Omega_h$, consisting of a cell of the median-dual mesh surrounding each node. In order to avoid any possible confusion, mesh nodes will be numbered instead of denoted by standard indexes $i, j$, here reserved to indicate spatial coordinates.

#### 1. *RANS equations*

The solution of the flow (1) and the turbulence model (2) enter on the adjoint equations as the coefficients of the adjoint system. For the flow equations, a central scheme with Jameson-Schmidt-Turkel (JST)-type scalar artificial dissipation[9,16] is used for the discretization of the convective flux. The convection of the turbulent variable $\hat{\nu}$ is discretized using a fully upwinded scheme. Second order accuracy is easily achieved via reconstruction of variables on the cell interfaces by using a MUSCL approach with limitation of gradients.[30] In both cases, viscous fluxes are computed with a node-gradient-based approach due to Weiss *et al.*,[9,31] which, apart of reducing the truncation error of the scheme, avoids the odd-even decoupling of mesh nodes in the computation of residuals, resulting in second-order spatial accuracy. Source terms are approximated via piecewise reconstruction in the finite-volume cells.

The solution of the turbulence model (2) requires as well the numerical approximation of the Eikonal equation (4), in order to compute the distance field to the boundary of the obstacle. Our implementation makes use of an efficient fast-marching solver for unstructured grids,[8] based on a finite-element approximation to the Eikonal equation in each mesh element. Second order accuracy is recovered by using not only the information of the nodal values of the distance field, but incorporating the direction of the computed $\nabla d_S$ into the solver.[27]

#### 2. *Adjoint flow equation*

Convective residuals. A modified version of the JST scheme without low-order dissipation is used for the discretization of the convective term.[4] This is given by

$$\int_{\Omega_h} \nabla \Psi_U^{\mathsf{T}} \cdot \vec{A}^c \, d\Omega \tag{36}$$

which unfortunately is not written in conservative form, thus avoiding the direct application of the Green–Gauss theorem. However, this flux can be seen as a convection with non-constant coefficients, given by the evaluation of matrices $\vec{A}^c$ at the different mesh nodes. Hence, across the face of the control volume separating two mesh nodes with local indexes 0 and 1, the part of the convective residual for node 0 can be computed as

$$\mathcal{R}_U^{\text{conv}} = \begin{pmatrix} (\vec{v}\cdot\vec{S})\psi_1 + \frac{|\vec{v}|^2}{2}(\gamma-1)l_{1\psi} - (\vec{v}\cdot\vec{S})l_{2\psi} \\ (\vec{v}\cdot\vec{S})\psi_2 - v_1(\gamma-1)l_{1\psi} + S_1 l_{2\psi} \\ (\vec{v}\cdot\vec{S})\psi_3 - v_2(\gamma-1)l_{1\psi} + S_2 l_{2\psi} \\ (\vec{v}\cdot\vec{S})\psi_4 - v_3(\gamma-1)l_{1\psi} + S_3 l_{2\psi} \\ (\vec{v}\cdot\vec{S})\psi_5 + (\gamma-1)l_{1\psi} \end{pmatrix} + \mathcal{D}_{01} \tag{37}$$

where the adjoint variables are reconstructed at the cell face as $\Psi_U = \frac{1}{2}(\,\Psi_U|_0 + \Psi_U|_1)$, whereas flow variables are evaluated at node 0 (equivalently for node 1). In (37), $\vec{S} = S_i = (S_x, S_y, S_z)$ denotes the normal vector of the face located at the edge going from node 0 to node 1 such that its length coincides with the face area, and we have introduced the following notation

$$l_{1\psi} = (\vec{\varphi}\cdot\vec{S}) + (\vec{v}\cdot\vec{S})\psi_5 \tag{38}$$

$$l_{2\psi} = \psi_1 + (\vec{\varphi}\cdot\vec{v}) + H\psi_5. \tag{39}$$

The artificial dissipation between nodes 0 and 1 can be expressed as

$$\mathcal{D}_{01} = \kappa^{(4)}\varepsilon^{(4)}\left(\nabla^2\ \Psi_U|_0 - \nabla^2\ \Psi_U|_1\right)\omega_{01}\lambda_{01} \tag{40}$$

$$\varepsilon^{(4)} = \left(3\frac{N_0+N_1}{N_0N_1}\right)^2 \tag{41}$$

$$\lambda_0 = |\vec{v}_0\cdot\vec{S}| + c_0|\vec{S}| \tag{42}$$

$$\lambda_{01} = |\vec{v}_{01}\cdot\vec{S}| + c_{01}|\vec{S}| \tag{43}$$

$$\omega_0 = \left(\frac{\lambda_0}{4\lambda_{01}}\right)^{\frac{1}{2}} \tag{44}$$

$$\omega_{01} = \frac{\omega_0\omega_1}{\omega_0+\omega_1} \tag{45}$$

where $\nabla^2\ \Psi_U|_0 = \sum_{n\in\mathcal{N}_0}(\Psi_U|_n - \Psi_U|_0)$ denotes the undivided Laplacian operator, where $\mathcal{N}_0$ represents the set of neighboring points to node 0 and $N_0$ its size, $\vec{v}_{01} = \frac{1}{2}(\vec{v}|_0 + \vec{v}|_1)$ and $c_{01} = \frac{1}{2}(c|_0 + c|_1)$ are the fluid and sound speeds at the cell face, and $\kappa^{(4)}$ is an adjustable parameter. An artificial dissipation of upwind-type could have been used as well, for which we refer the reader to.[4]

Viscous residuals. The viscous fluxes of the adjoint flow equation are discretized using the Green–Gauss integral relation

$$\int_{\Omega_h}\nabla\left(\nabla\Psi_U^{\mathsf{T}}\cdot\mu_{tot}^k\mathbf{D}^{vk}\right)\,d\Omega = \int_{\partial\Omega_h}\nabla\Psi_U^{\mathsf{T}}\cdot\mu_{tot}^k\mathbf{D}^{vk}\,d\vec{S} \tag{46}$$

which yields the following form for the viscuous residuals

$$\mathcal{R}_U^{\text{visc}} = \frac{1}{\rho}\begin{pmatrix} -v_i\Sigma_{ij}S_j + \left(\frac{|\vec{v}|^2}{2} - \frac{P}{(\gamma-1)\rho}\right)\Sigma^5 \\ \Sigma_{1j}S_j - v_1\Sigma^5 \\ \Sigma_{2j}S_j - v_2\Sigma^5 \\ \Sigma_{3j}S_j - v_3\Sigma^5 \\ \Sigma^5 \end{pmatrix} \tag{47}$$

where the following abbreviations have been used

$$\Sigma_{ij} = \Sigma_{ij}^{\varphi} + \Sigma_{ij}^5 \tag{48}$$

$$\Sigma_{ij}^{\varphi} = \mu_{tot}^1\left(\partial_j\varphi_i + \partial_i\varphi_j - \frac{2}{3}\delta_{ij}\nabla\cdot\vec{\varphi}\right) \tag{49}$$

$$\Sigma_{ij}^5 = \mu_{tot}^1\left(v_j\partial_i\psi_5 + v_i\partial_j\psi_5 - \frac{2}{3}\delta_{ij}\vec{v}\cdot\nabla\psi_5\right) \tag{50}$$

$$\Sigma_i^5 = \gamma\mu_{tot}^2\,\partial_i\psi_5 \tag{51}$$

$$\Sigma^5 = \gamma\mu_{tot}^2\,\nabla\psi_5\cdot\vec{S}. \tag{52}$$

For coherence with the discretization of the convective residuals, the gradients of the adjoint variables are averaged (making use of the Weiss correction) at the cell face, whereas flow variables are evaluated at node 0.

Source terms residuals. The remaining terms in (99) are treated as source terms, and approximated via piecewise reconstruction of the solution in each finite volume cell. After some manipulations, the residual vector can be written as

$$\mathcal{R}_U^{\text{source}} = |\Omega_h|\begin{pmatrix} -\frac{1}{\rho}v_i\sigma_{ij}\partial_j\psi_5 - \Sigma_{ij}\partial_i\left(\frac{v_j}{\rho}\right) + \left(\frac{|\vec{v}|^2}{2}\alpha_j - \beta_j\right)\partial_j\psi_5 + \frac{1}{\rho}\Sigma_j^5 v_i\partial_j v_i + \theta(|\vec{v}|^2 - E) + \xi \\ \frac{1}{\rho}\sigma_{1j}\partial_j\psi_5 + \Sigma_{1j}\partial_j\left(\frac{1}{\rho}\right) - v_1\alpha_j\partial_j\psi_5 - \frac{1}{\rho}\Sigma_j^5\partial_j v_1 - \theta v_1 \\ \frac{1}{\rho}\sigma_{2j}\partial_j\psi_5 + \Sigma_{2j}\partial_j\left(\frac{1}{\rho}\right) - v_2\alpha_j\partial_j\psi_5 - \frac{1}{\rho}\Sigma_j^5\partial_j v_2 - \theta v_2 \\ \frac{1}{\rho}\sigma_{3j}\partial_j\psi_5 + \Sigma_{3j}\partial_j\left(\frac{1}{\rho}\right) - v_3\alpha_j\partial_j\psi_5 - \frac{1}{\rho}\Sigma_j^5\partial_j v_3 - \theta v_3 \\ \alpha_j\partial_j\psi_5 + \theta \end{pmatrix} \tag{53}$$

where $|\Omega_h|$ denotes the cell measure, and

$$\alpha_i = \gamma \mu_{tot}^2 \, \partial_i \left( \frac{1}{\rho} \right) \tag{54}$$

$$\beta_i = \frac{\gamma}{\gamma - 1} \mu_{tot}^2 \, \partial_i \left( \frac{P}{\rho^2} \right) \tag{55}$$

$$\kappa_\psi = \bar{\tau} \colon \nabla \vec{\varphi} + \vec{v} \cdot \bar{\tau} \cdot \nabla \psi_5 + \frac{C_p}{Pr_t} \nabla T \cdot \nabla \psi_5 \tag{56}$$

$$\theta = \left[ \kappa_\psi \left( 1 - \frac{\mu_{tur}}{\mu_{dyn}} \frac{3 c_{v1}^3}{\chi^3 + c_{v1}^3} \right) - \frac{C_p}{Pr_t} \nabla T \cdot \nabla \psi_5 \left( 1 - \frac{Pr_t}{Pr_d} \right) \right] \frac{(\gamma - 1)}{R \rho} \frac{\partial \mu_{dyn}}{\partial T} \tag{57}$$

$$\xi = \kappa_\psi \left( 1 + \frac{3 c_{v1}^3}{\chi^3 + c_{v1}^3} \right) \frac{\mu_{tur}}{\rho} \tag{58}$$

and $\bar{\tau} \colon \nabla \vec{\varphi} = \tau_{ij} \partial_i \varphi_j$, with $\frac{\partial \mu_{dyn}}{\partial T}$ given by (80).

Coupling residuals. Coupling between the Navier–Stokes and the Spalart–Allmaras adjoint equations is obtained by means of (100). Its two first terms are integrated using piecewise reconstruction, yielding a combined residual

$$\mathcal{R}_U^{\hat{\nu}}\big|_{p.w.} = |\Omega_h| \begin{pmatrix} -\frac{\Gamma}{\rho} \left( \widetilde{\alpha} \left( |\vec{v}|^2 - E \right) - \frac{\mu_{dyn}}{\rho} \right) + \frac{\hat{\nu}}{\rho} v_j \partial_j \psi_{\hat{\nu}} + \psi_{\hat{\nu}} \frac{\Lambda}{|\vec{\omega}|} \left( \partial_j v_i - \partial_i v_j \right) \partial_j \left( \frac{v_i}{\rho} \right) \\ \frac{\Gamma}{\rho} \widetilde{\alpha} v_1 - \frac{\hat{\nu}}{\rho} \partial_1 \psi_{\hat{\nu}} - \psi_{\hat{\nu}} \frac{\Lambda}{|\vec{\omega}|} \left( \partial_j v_1 - \partial_1 v_j \right) \partial_j \left( \frac{1}{\rho} \right) \\ \frac{\Gamma}{\rho} \widetilde{\alpha} v_2 - \frac{\hat{\nu}}{\rho} \partial_2 \psi_{\hat{\nu}} - \psi_{\hat{\nu}} \frac{\Lambda}{|\vec{\omega}|} \left( \partial_j v_2 - \partial_2 v_j \right) \partial_j \left( \frac{1}{\rho} \right) \\ \frac{\Gamma}{\rho} \widetilde{\alpha} v_3 - \frac{\hat{\nu}}{\rho} \partial_3 \psi_{\hat{\nu}} - \psi_{\hat{\nu}} \frac{\Lambda}{|\vec{\omega}|} \left( \partial_j v_3 - \partial_3 v_j \right) \partial_j \left( \frac{1}{\rho} \right) \\ -\frac{\Gamma}{\rho} \widetilde{\alpha} \end{pmatrix} \tag{59}$$

with the abbreviations

$$\widetilde{\alpha} = \frac{(\gamma - 1)}{R \rho} \frac{\partial \mu_{dyn}}{\partial T} \tag{60}$$

$$\Gamma = \psi_{\hat{\nu}} \Lambda \hat{S}^{f_{v2}} \left( f_{v2}^{\chi} + f_{v2}^{v1} f_{v1}^{\chi} \right) \chi^{\nu} - \frac{1}{\sigma} \nabla \hat{\nu} \cdot \nabla \psi_{\hat{\nu}} \tag{61}$$

and all partial derivatives appearing in these formulas given in §VII.D.

The third term of (100) is however written in conservative form. Hence, this part of the residual is evaluated by means of the Green–Gauss theorem as

$$\mathcal{R}_U^{\hat{\nu}}\big|_{cons} = \frac{\psi_{\hat{\nu}}}{\rho} \frac{\Lambda}{|\vec{\omega}|} \begin{pmatrix} -v_i \left( \partial_j v_i - \partial_i v_j \right) S_j \\ \left( \partial_j v_1 - \partial_1 v_j \right) S_j \\ \left( \partial_j v_2 - \partial_2 v_j \right) S_j \\ \left( \partial_j v_3 - \partial_3 v_j \right) S_j \\ \cdot \end{pmatrix}. \tag{62}$$

### 3. *Adjoint turbulent equation*

Convective residuals. To preserve consistency with the direct solver, the turbulent adjoint variable $\psi_{\hat{\nu}}$ is discretized as well using a second-order upwind scheme with face reconstruction and limitation of gradients. This convection also comes given as a non-conservative flux in the form

$$\int_{\Omega_h} \nabla \psi_{\hat{\nu}} \cdot \vec{B}^{cv} \, d\Omega \tag{63}$$

where the analytical expression for the flux $\vec{B}^{cv}$ is given by (87). Supposing a regular solution of the adjoint equations, the convective residual for node 0 across the face of the control volume separating nodes 0 and 1 is approximated as

$$\mathcal{R}_{\hat{\nu}}^{conv} = \frac{1}{2} \left[ \vec{B}_0^{cv} \cdot \vec{S} \left( \psi_{\hat{\nu}}|_0 + \psi_{\hat{\nu}}|_1 \right) - |\vec{B}_{01}^{cv} \cdot \vec{S}| \left( \psi_{\hat{\nu}}|_1 - \psi_{\hat{\nu}}|_0 \right) \right]. \tag{64}$$

VISCOUS RESIDUALS. Viscous fluxes are newly discretized using the Green–Gauss integral relation

$$\int_{\Omega_h} \nabla \left( \nabla \psi_{\hat{\nu}} \cdot \mathbf{E}^{cv} \right) \, d\Omega = \int_{\partial \Omega_h} \nabla \psi_{\hat{\nu}} \cdot \mathbf{E}^{cv} \, d\vec{S} \tag{65}$$

yielding for node 0 the following residual

$$\mathcal{R}_{\hat{\nu}}^{\text{visc}} = -\frac{\nu + \hat{\nu}}{\sigma} \nabla \psi_{\hat{\nu}} \cdot \vec{S} \tag{66}$$

where gradients of the turbulent adjoint variable are newly averaged at the cell face, including the Weiss correction, and rest of variables are evaluated at node 0.

SOURCE TERMS RESIDUALS. The rest of terms in (102) constitute the source contributions to the adjoint turbulent equation. As for the coupling residuals of the adjoint Navier–Stokes equation, the two first terms are integrated using piecewise reconstruction, resulting in a residual in the form

$$\mathcal{R}_{\hat{\nu}}^{source}|_{p.w.} = |\Omega_h| \left( -\psi_{\hat{\nu}} B^s \right) \tag{67}$$

with $B^s$ given by (90). The $\mathbf{E}^s$ term is evaluated by the Green–Gauss integral relation, yielding

$$\mathcal{R}_{\hat{\nu}}^{source}|_{cons.} = \frac{2 c_{b2}}{\sigma} \psi_{\hat{\nu}} \nabla \hat{\nu} \cdot \vec{S} \tag{68}$$

COUPLING RESIDUALS. Finally, residuals coming from the coupling with the adjoint flow equation, given by (101), can be written as

$$\mathcal{R}_{\hat{\nu}}^{U} = |\Omega_h| \kappa_{\psi} \frac{\partial \mu_{tur}}{\partial \hat{\nu}} \tag{69}$$

with $\kappa_{\psi}$ and $\frac{\partial \mu_{tur}}{\partial \hat{\nu}}$ given by (56) and (83).

#### 4. *Adjoint Eikonal equation*

As previously stated, neither the adjoint flow nor the adjoint turbulent equations depend on the adjoint distance variable $\psi_d$. This adjoint variable is neither needed to compute the functional sensibility, since typical objective functionals in aerodynamics do not depend explicitly on the distance to the surface. Hereafter, and for the functionals considered below, there is no need to implement and numerically solve (23).

#### 5. *Boundary conditions*

Boundary conditions for a solid wall can be imposed in two ways, either by using a ghost cell scheme adapted to unstructured meshes, or by directly enforcing the boundary conditions on the analytical flux expressions. On the far-field, characteristic boundary conditions are used.

### B. Steady-state time integration

A time-marching strategy in pseudo-time[9,12] is used to obtain the steady solution of the flow equations and the adjoint system (21)-(22). Although the two sets of equations are coupled over the turbulent viscosity $\mu_{tur}$, and it would be more efficient in terms of operation counts to formulate and solve them simultaneously, decoupling is by far the most widely used strategy, both for simplicity and because of the different character of both types of equations. In addition, it also gives flexibility in order to introduce in a future other models of turbulence.

Due to the inherent stiffness of turbulence-transport equations, time integration of both the Navier–Stokes and the Spalart–Allmaras adjoint systems is tackled with an implicit backward-Euler scheme.[7] Direct inversion of the banded matrix that defines this system is impractical because of rapid increase of operation count with the number of mesh points and large storage requirements, especially in 3-D. The resulting linear system is therefore solved iteratively by means of a LU-SGS algorithm.[17] In order to speed up the rate of convergence, an overset multigrid scheme is used in conjunction with the solver.[14,15]

### C. Design variables

Different types of design variables are traditionally used in aerodynamic shape optimization, including deformation or bump functions such as Wagner polynomials, Hicks–Vanderplaats functions, Legendre polynomials, Hicks–Henne functions, Bézier polynomials, nonuniform rational B–splines (NURBS), as well as modifications in the thickness and camber line and also individual surface node movement.

In the present work, the shape functions introduced by Hicks–Henne[11] have been used

$$\Delta y = \sum_{n=1}^{N} \delta_n f_n(x), \quad f_n(x) = \sin^3\left(\pi x^{e_n}\right), \quad e_n = \frac{\log(0.5)}{\log(x_n)} \tag{70}$$

where each shape function has its maximum at the point $x_n$, being $N$ the number of bump functions and $\delta_n$ the design variable step. These functions are applied separately to the upper and lower surfaces.

In 3D configurations, a Free–Form Deformation[26] (FFD) strategy has been chosen. In FFD a initial box is parameterized as a Bézier solid, and the object (wing, airplane, etc.) that we want to design is inside this box. The control points of the box become design variables, as they control the form of the solid, and thus the shape of the surface grid inside (FFD wraps the object on this solid box). The overall procedure is as follow:

- Define the control points that determine the initial box, according with the part of the physical geometry we want to modify, i.e., encapsulating the part of the object that is going to be modeled. This way, the solid box is now parameterized by the following expression:

$$X(u,v,w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_i^l(u) B_j^m(v) B_k^n(w), \tag{71}$$

  where $u, v, w \in [0;1]$, and the $B$ are the Bernstein polynomials.

- Given a set of points on the object, which the cartesian coordinates are known, transform their coordinates to the parametric coordinates of the Bézier box.

- Modify the box by modifying its control points. Every point inside the box will inherit the deformation on the designed surface.

- Recover the new cartesian coordinates of the modified points of the initial object.

To sum up, in a optimization procedure the coordinates of the control points of the Bézier box, are the design variables. A change in those points inherits a smooth movement of the object surface.

### D. Mesh deformation

Once the boundary displacements have been computed, a torsional spring method is used in order to move the unstructured meshes. The method is based on the definition of a stiffness matrix, $k_{ij}$, that connects the two ends of a single bar (mesh edge), in addition to torsional springs around each mesh node which prevent nodes from crossing edges.[3,6] Equilibrium of forces is then imposed at each mesh node

$$\left( \sum_{j \in \mathcal{N}_i} k_{ij} \vec{e}_{ij} \vec{e}_{ij}^{\mathsf{T}} \right) \vec{u}_i = \sum_{j \in \mathcal{N}_i} k_{ij} \vec{e}_{ij} \vec{e}_{ij}^{\mathsf{T}} \vec{u}_j \tag{72}$$

where the displacement $\vec{u}_i$ is unknown and is computed as a function of the known current displacements $\vec{u}_j$, being $\mathcal{N}_i$ the set of neighboring points to node $i$ and $\vec{e}_{ij}$ the unitary vector in the direction connecting both points. The system of equations (72) is solved iteratively by a conjugate gradient algorithm with Jacobi preconditioning.

### E. Optimization framework

The continuous adjoint formulation allows the computation of a wide range of different objective functions: quadratic deviation from a target pressure (inverse design), drag minimization, lift maximization, pitching moment, aerodynamic efficiency, and linear combinations of those. Also, several constraints have been implemented: fixed non-dimensional flow parameters (minimum lift, maximum drag, etc.) and geometrical estimations (maximum and minimum thickness, curvature, volume, area, etc.).

The optimization results presented in this work make use of the *SciPy* library (`http://www.scipy.org`), a well-established open-source software for mathematics, science, and engineering. The SciPy library provides many user-friendly and efficient numerical routines for the solution of non-linear constrained optimization problems, such as conjugate gradient, Quasi–Newton or sequential least-squares programming algorithms. At each design iteration, the SciPy routines require as inputs the values and gradients of the objective functions as well as the chosen constraints.

## V. Summary of results, numerical experiments

All the final formulas of this paper have been implemented in the aerodynamic shape design suite Cades[23] (*Code for Aerodynamical DEsign and Simulation*, developed by the research group). Cades is a complete suite for aerodynamical shape design composed by five C++ programs (CFD solver, grid adaptation, grid deformation, domain partitioning, and gradients computation), and several Python scripts. In particular, the software Cades_CFD is a finite volume code that solves the direct (flow), adjoint (using the formulation of this paper), and linearized problems for the potential, Euler, Navier-Stokes, and RANS equations on either 2-D or 3-D unstructured meshes using an edge-based data structure.

All meshes used in the computations are unstructured grids, where appropriate resolution near the airfoil surfaces is achieved with a layered subgrid composed of quadrilateral elements. Since the computation of gradients using the Green–Gauss theorem is known to produce considerable errors in any mixed grid at interfaces from differing element types, gradients are evaluated by a weighted Least–Squares approximation. Standard values for the first, second and forth–order artificial dissipation coefficients of the flow JST solver are 0.1, 1/4, and 1/64, respectively. The value for the adjoint forth–order artificial dissipation coefficient $\kappa^{(4)}$ in (40) is taken to be 1/150.

All the final formulas of this paper have been implemented in the software suite with the exception of the terms involving the factor $\frac{\Lambda}{|\vec{\omega}|}$ in (59) and (62). Numerical inspection of the quantity $\Lambda$, given by (95), reveals very small values for this factor and only within the turbulent boundary layer. Precisely, the magnitude of the vorticity is maximal close to the airfoil surface, making the overall factor $\frac{\Lambda}{|\vec{\omega}|}$ negligible when compared to the rest of terms in the computation of the adjoint residuals. Neglecting these terms, however, helps to avoid numerical instabilities further from the obstacle surface, where the magnitude of the vorticity tends to zero. As proved in the following sections, very good results are still obtained by making this assumption, although other regularization approaches for these terms are under current development.

In this section some preliminary numerical test will be presented. Firstly, gradients computed with the adjoint method described in this paper are compared with those obtained with a forward finite difference (i.e., brute force) to demonstrate the quality of the gradients computed using this approach in comparison with those obtained using the so–called *frozen viscosity* hypothesis. Then, a transonic unconstrained drag minimization problem is shown to highlight the importance of using the complete adjoint methodology compared with the *frozen viscosity* strategy. Finally, an unconstrained inviscid 3D drag minimization problem using the Free–Form deformation technique will be presented to study the viability of the FFD for the definition of 3D design variables.

### A. Regulatory effects and improvements in convergence of the full turbulent adjoint

Our first case of study involves a transonic, turbulent flow over the RAE-2822 airfoil. The flow conditions correspond to AGARD-AR-138 case 9,[5] with corrections to account for wind tunnel effects,[10] namely $M_\infty = 0.734$, $\alpha = 2.54^o$, and $\mathrm{Re} = 6.5 \times 10^6$. Under these conditions the flow develops a shock wave on the upper surface (located at about 50% to 60% of the airfoil chord) and a very small (if present at all) shock-induced separation behind the shock. The computational grid is an unstructured, two-dimensional grid with 13937 nodes and 22842 elements, with 192 nodes on the airfoil surface and 40 nodes on the far-field boundary, with is located 100 chords away from the airfoil.

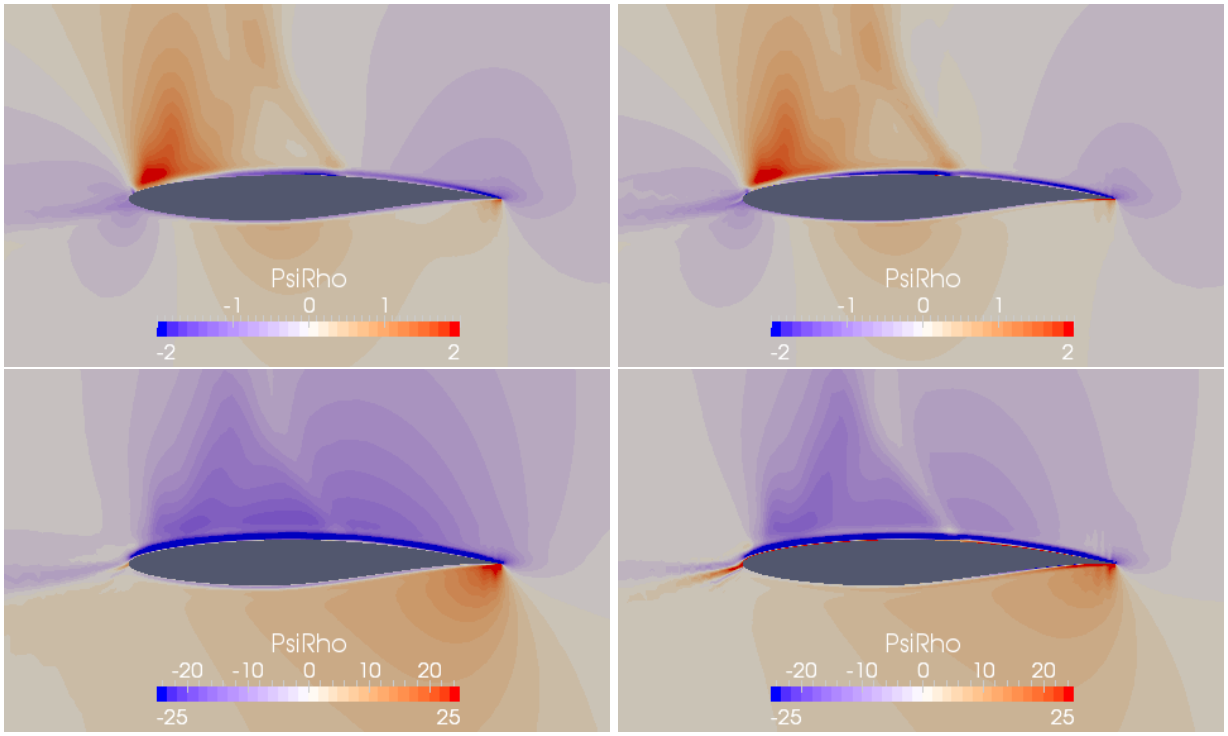


**Figure 1. Comparison of computed adjoint density variables $\psi_1$ for drag (top row of the panel) and lift (bottom row) objective functions, using the proposed fully turbulent adjoint approach (left column) or under the frozen viscosity hypothesis (right column).**

| | Drag coefficient | | Lift coefficient | |
|---|---|---|---|---|
| | Fully Turbulent | Frozen Viscosity | Fully Turbulent | Frozen Viscosity |
| $\psi_1$ | [-4.80, 2.50] | [-6.61, 17.37] | [-125.62, 77.44] | [-157.88, 458.59] |
| $\psi_2$ | [-8.34, 14.21] | [-28.43, 31.18] | [-260.59, 362.68] | [-756.58, 774.01] |
| $\psi_3$ | [-18.67, 1.06] | [-57.71, 0.88] | [-522.61, 28.92] | [-1478.69, 34.67] |
| $\psi_5$ | [-1.55, 0.74] | [-1.54, 1.26] | [-38.18, 22.41] | [-33.06, 35.76] |
| $\psi_{\tilde{\nu}}$ | $[-1.65, 0.65] \times 10^5$ | N/A | $[-4.55, 1.38] \times 10^6$ | N/A |

**Table 1. Range of variation of the different adjoint variables for drag and lift objective functions, using the proposed fully turbulent adjoint approach or under the frozen viscosity hypothesis. N/A $\equiv$ Not Assigned.**

A comparison between the computed solutions for the adjoint density variable $\psi_1$, using either the proposed fully turbulent adjoint method or under the generally used frozen viscosity hypothesis, can be found in Fig. 1 for both the drag and lift objective functions. Although similar, one can realize that contour curves for the fully turbulent adjoint are in general smoother than those for the adjoint method with frozen viscosity, and much less affected by the presence of the shock wave. This is due to the coupling terms between the flow and the turbulent equations introduced by the new formulation, given by (100). The same applies for the rest of the adjoint flow variables. In fact, and due to visualization purposes, all the smoothing effects of the fully turbulent adjoint can not be clearly appreciated in Fig. 1. This becomes more evident in Table 1, where the minimum and maximum values of all the adjoint variables are listed. The range of variation of all the adjoint variables is always smaller using the proposed method than the standard adjoint with frozen viscosity. The values of the adjoint turbulent variable $\psi_{\tilde{\nu}}$ are however much larger than those of the adjoint flow variables, although concentrated in the boundary layer and neither affected by the shock wave.

The convergence properties of the adjoint method are also improved by considering the full coupling between the flow and the turbulent equations. This is shown in Fig. 3, where it is plotted the convergence of

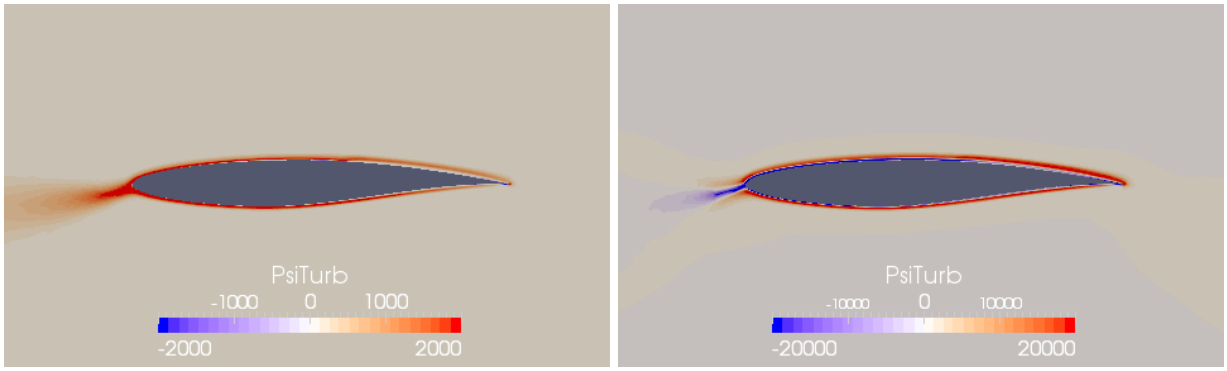


**Figure 2. Computed adjoint turbulent variables $\psi_{\tilde{\nu}}$ for drag (leftl) and lift (right) functionals.**

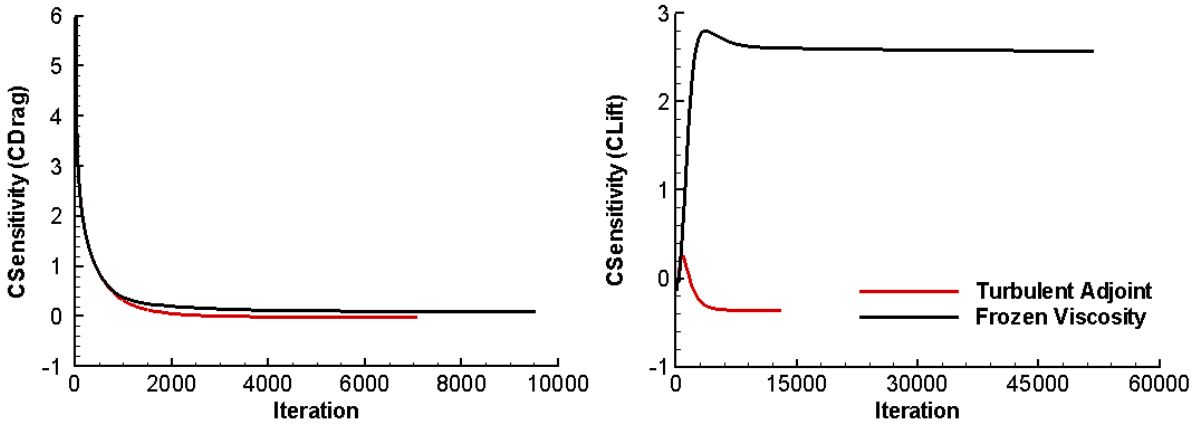


**Figure 3. Convergence of the geometrical sensitivity in the computation of the gradient of the drag (left) and lift (right) coefficients.**

the sensitivity parameter given by (35) for the computation of the gradient of the drag and lift coefficients. The two adjoint approaches are compared using the same stopping criteria of an accumulated error of $10^{-5}$ in the last 100 elements of the Cauchy series. The convergence is better in the computation of the gradient of the drag coefficient under deformations of the wing profile, where 7091 iterations are needed for the fully turbulent approach instead of the 9535 of the adjoint method with frozen viscosity, but especially relevant for the case of the lift gradient, where only 13128 iterations are now required instead of the respective 51897. Note that, since the final values of the adjoint flow variables are different, the two approaches do not have to converge to the same value of surface sensitivity.

## B. Numerical comparison of gradients in drag and lift optimization problems

We now present comparisons of the gradients computed by the two adjoint methods for the previous case of study. The Hicks–Henne functions have been used as design variables. The first design variable has its maximum close to the trailing edge on the lower side of the airfoil, and subsequent variables displace the maximum in the clockwise direction. A total number of 38 bump functions were used, spanning the complete surface of the airfoil.

The gradients computed with the fully turbulent adjoint method and the continuous adjoint with frozen viscosity are compared with those obtained with a forward finite-difference (i.e., *brute force*) method, where the finite step of the design variable must be carefully selected depending on the flow regime.

Figure 4 summarizes the main results on the comparison of the gradients. Assuming the finite-differences approximation as the reference solution, it is clear that the gradients computed by the adjoint method

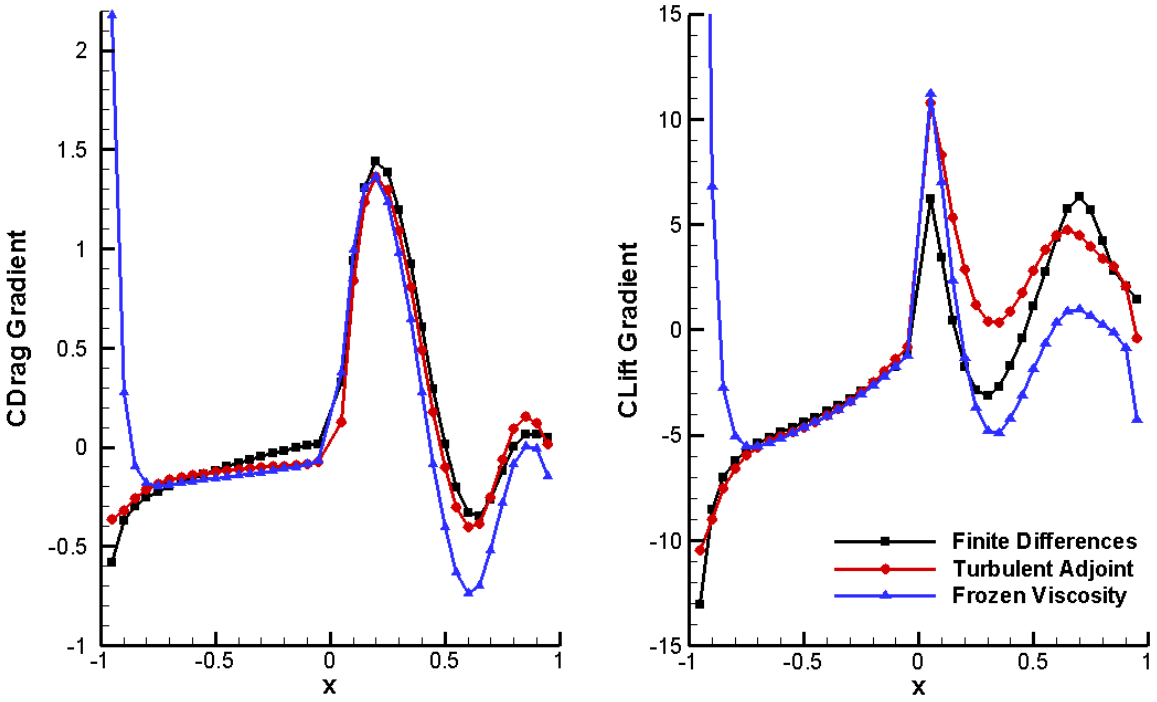


**Figure 4. Comparison of gradients computed by the fully turbulent adjoint approach and frozen viscosity adjoint of the drag (left) and lift (right) coefficients. Forward finite-differences approximations of the gradients are given as reference solutions. In these figures, the negative part of the $x$ axis denotes design variables located on the lower side of the airfoil, and positive $x$ those located on the upper side.**

described in this paper are in much better agreement to the exact solution than those computed using the frozen viscosity simplification. Moreover, as it can be seen in both examples close to the trailing edge on the lower surface of the airfoil, the frozen viscosity adjoint can indeed predict a gradient not only wrong in magnitude but also in its direction, thus seriously compromising the efficiency of the gradient-based optimization solver during the minimization process.

Although in general much better, especially for the approximation of the derivatives of the drag coefficient, the gradients computed by the fully turbulent continuous approach are not in total agreement with those obtained by the finite-differences method. The reason of these discrepancies are: Firstly, a continuous adjoint can not incorporate the influence of the mesh sensitivities on the calculations. Secondly, the flow sensitivity depends on the quality of the flow variable gradients that are used to build the source terms of the continuous adjoint formulation. And finally, the solution to the RANS equations is strongly affected by the computation of the distance field to the boundary surface. The very small deformations required to approximate the gradient by finite-differences might not be properly seen by either an Eikonal or even a brute-force distance solver. On the other hand, the adjoint method does not need to compute a new distance field for each surface deformation, since it only uses the distance field computed in the original grid with much better quality. So, apart from the fact of being just a first-order approximation to the exact gradient, it remains uncertain the degree of accuracy that can be attributed to the finite-differences solver in these situations. This may serve as an explanation to the bigger discrepancies between the turbulent adjoint and the finite-differences for the computation of the lift derivatives on the upper surface, where the lift coefficient is extremely sensible to the position of the shock.

### C. 2D unconstrained drag minimization using adjoint RANS

The goal of this academic problem is to reduce the drag of a RAE-2822 profile, by means of modifications of its surface. The angle of attack, Mach number, and Reynolds number are fixed so that the flow remains

transonic ($M_\infty = 0.734$, $\alpha = 2.54^o$, and $\mathrm{Re} = 6.5 \times 10^6$). A total of 38 Hicks–Henne bump functions have been used as design variables. The first design variable has its maximum close to the trailing edge on the lower side of the airfoil, and subsequent variables displace the maximum in the clockwise direction.

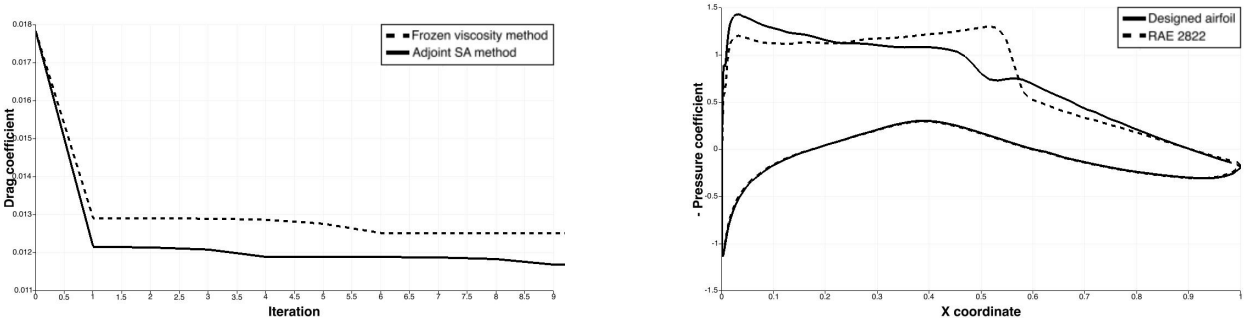


**Figure 5. Optimization convergence history, adjoint method vs. frozen viscosity (left). Pressure coefficient distribution, original configuration and final design (right).**

In Fig. 5 (left), the optimization convergence history is presented for the frozen viscosity method (without adjoint turbulence model), and for the complete adjoint Spalart–Allmaras problem (using the methodology described in this paper). As we expected, gradients computed with the complete adjoint Spalart–Allmaras formulation provide an airfoil with a lower value of the drag coefficient than using the frozen viscosity gradients. On the other hand, in Fig. 5 (right), the initial pressure distribution, and the optimized one are shown for the adjoint Spalart–Allmaras case. In particular, the original drag coefficient was 0.0178, and the final one is 0.0116.

### D. 3D design using Free–Form deformation technique, and adjoint surface formulation

The objective of this numerical test is to study the viability of the Free–Form technique for the definition of 3D design variables. A single–point minimization case is used to show the accuracy of the developed continuous adjoint method for inviscid flows. The flow conditions are Mach number 0.8395 with angle of attack 3.06 deg, and the selected wing is an ONERA-M6, only the upper surface will be resigned, and a total of 30 design variable will be used. The governing equations are the Euler equations, so drag improvement in this case means wave drag decrease.

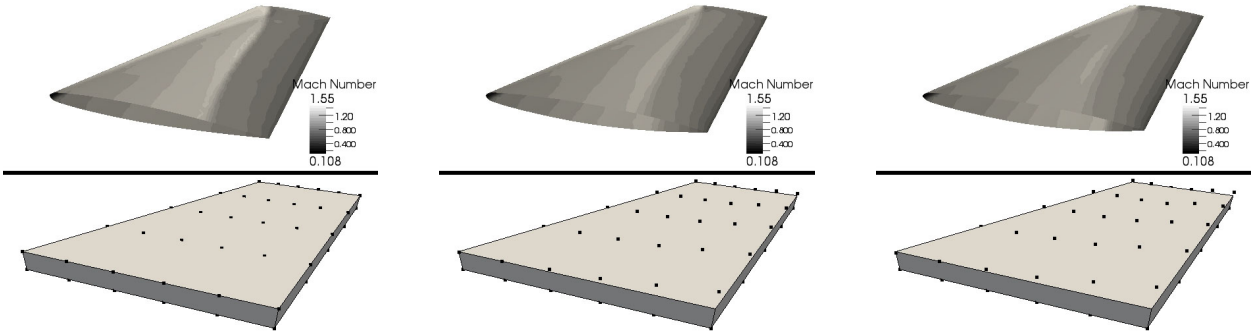


**Figure 6. Mach number distribution at iterations 0, 3, and 6 (upper), and control point position (FFD) at iterations 0, 3, 6 (lower).**

In Fig. 6 (upper) the Mach distribution for several optimization iterations is shown. At the last iteration, the shock wave has disappeared. In Fig. 6 (lower) it is possible to appreciate the movement of the control points of the FFD box.

After the minimization process, the new wing has a drag coefficient of 0.0029, which is a 25% of the original ONERA M6 drag (inviscid computation). Finally, in Fig. 7, the convergence history of the optimization

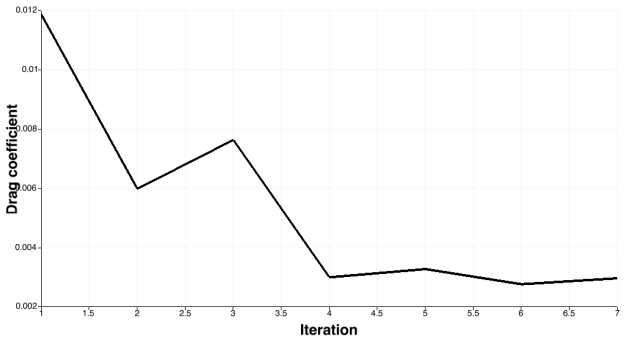


**Figure 7. Optimization history for the redesign of an ONERA–M6 using Free–Form deformation, and adjoint surface formulation.**

problem is shown.

## VI. Conclusions

In this work the continuous adjoint Spalart–Allmaras approach to aerodynamic design optimization has been presented. The formulation that we have derived do not need the computation of second–order derivatives of the flow variables, and once the adjoint variable has been computed, it only requires an integration on the surface of the aerodynamical body.

The accuracy of the sensitivity derivatives that result from the application of the method developed in this work has been assessed by comparison with finite-difference computations, which illustrate the importance of the complete formulation instead of the Frozen viscosity strategy. Finally, an unconstrained minimization case is shown, and a brief introduction to 3D design using Free–Form Deformation technique is also described.

The results presented here are very promising, but further numerical tests are necessary. In particular, complex 3D configurations, and convergence issues are open research topics to aboard the industrialisation of this continuous adjoint methodology.

## VII. Appendix

In this appendix we give the main formulas required to compute the sensitivities discussed above.

### A. Navier–Stokes equations

As usual in Navier–Stokes equations, system (1) consider separately convective terms, denoted by $\vec{F}^c$, and viscous ones, denoted by $\vec{F}^{v1}$ and $\vec{F}^{v2}$. They are given by

$$\vec{F}_i^c = \begin{pmatrix} \rho v_i \\ \rho v_i v_1 + P\delta_{i1} \\ \rho v_i v_2 + P\delta_{i2} \\ \rho v_i v_3 + P\delta_{i3} \\ \rho v_i H \end{pmatrix}, \quad \vec{F}_i^{v1} = \begin{pmatrix} \cdot \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ v_j \tau_{ij} \end{pmatrix}, \quad \vec{F}_i^{v2} = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ C_p \partial_i T \end{pmatrix}, \quad i = 1, \ldots, 3 \tag{73}$$

where $v_1, v_2$, and $v_3$ are the Cartesian velocity components, $H$ is the fluid enthalpy, $\delta_{ij}$ is the Kronecker delta function (i.e. $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ otherwise), and

$$\tau_{ij} = \partial_j v_i + \partial_i v_j - \frac{2}{3}\delta_{ij} \nabla \cdot \vec{v}$$

Recall that latin indexes $i, j$ denote 3-D Cartesian coordinates, $x_i = (x, y, z)$, with repeated indexes implying summation. In these formulas, $C_p$ is the specific heat at constant pressure, $T$ is the temperature, defined as

$$T = \frac{P}{R\rho}$$

and $R$ is the gas constant, so that for an ideal gas $\frac{C_p}{R} = \frac{\gamma}{(\gamma-1)}$ with $\gamma$ constant. In order to close the system of equations, the dynamic viscosity is assumed to satisfy the Sutherland's law

$$\mu_{dyn} = \frac{\mu_1 T^{3/2}}{T + \mu_2}$$

where $\mu_1$ and $\mu_2$ are also specified constants.

### B. Spalart–Allmaras turbulence model

In (2) we have considered a general framework for a one-equation turbulence modeling. In this section we focus on the Spallart–Allmaras model for which explicit formulas for the adjoint formulation are obtained below. In this case, the turbulent viscosity is computed from

$$\mu_{tur} = \rho \hat{\nu} f_{v1}, \tag{74}$$

where the function $f_{v1}$ is given by

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\hat{\nu}}{\nu}, \qquad \nu = \frac{\mu_{dyn}}{\rho}. \tag{75}$$

Equations (74) and (75) require a new variable $\hat{\nu}$ which is obtained by solving (2) with

$$\vec{T}^{cv}(U, \hat{\nu}) = -\frac{\nu + \hat{\nu}}{\sigma}\nabla\hat{\nu} + \vec{v}\hat{\nu} \tag{76}$$

$$T^s(U, \hat{\nu}, d_S) = c_{b1}\hat{S}\hat{\nu} - c_{w1} f_w \left(\frac{\hat{\nu}}{d_S}\right)^2 + \frac{c_{b2}}{\sigma}|\nabla\hat{\nu}|^2. \tag{77}$$

A number of constants required above and below are listed together

$$\sigma = 2/3 \qquad c_{b1} = 0.1355 \qquad c_{b2} = 0.622 \qquad \kappa = 0.41$$
$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma} \qquad c_{w2} = 0.3 \qquad c_{w3} = 2 \qquad c_{v1} = 7.1.$$

The production term $\hat{S}$ in (77) is defined as

$$\hat{S} = |\vec{\omega}| + \frac{\hat{\nu}}{\kappa^2 d_S^2} f_{v2}, \qquad \vec{\omega} = \nabla \times \vec{v}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

where $\vec{\omega}$ is the fluid vorticity and $d_S$ is the distance to the nearest wall.

Finally, the function $f_w$ in (77) is computed as

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \qquad g = r + c_{w2}(r^6 - r), \qquad r = \frac{\hat{\nu}}{\hat{S}\kappa^2 d_S^2}.$$

### C. Linearized Navier–Stokes equations

In this section we compute $\frac{\partial R_U}{\partial U}\delta U$ and $\frac{\partial R_U}{\partial \hat{\nu}}\delta\hat{\nu}$ in (14).

$$\frac{\partial R_U}{\partial U}\delta U = \nabla(\vec{A}^c \delta U) - \nabla \cdot \left(\vec{F}^{vk}\frac{\partial \mu_{tot}^k}{\partial U}\delta U + \mu_{tot}^k \vec{A}^{vk}\delta U + \mu_{tot}^k \mathbf{D}^{vk}\nabla\delta U\right) \tag{78}$$

$$\frac{\partial R_U}{\partial \hat{\nu}}\delta\hat{\nu} = -\nabla \cdot \left(\vec{F}^{vk}\frac{\partial \mu_{tot}^k}{\partial \hat{\nu}}\delta\hat{\nu}\right) \tag{79}$$

where

$$\frac{\partial \mu_{tot}^1}{\partial U} = \frac{\partial \mu_{dyn}}{\partial U} + \frac{\partial \mu_{tur}}{\partial U}, \qquad \frac{\partial \mu_{tot}^2}{\partial U} = \frac{1}{\mathrm{Pr}_d}\frac{\partial \mu_{dyn}}{\partial U} + \frac{1}{\mathrm{Pr}_t}\frac{\partial \mu_{tur}}{\partial U}$$
$$\frac{\partial \mu_{tot}^1}{\partial \hat{\nu}} = \frac{\partial \mu_{tur}}{\partial \hat{\nu}}, \qquad \frac{\partial \mu_{tot}^2}{\partial \hat{\nu}} = \frac{1}{\mathrm{Pr}_t}\frac{\partial \mu_{tur}}{\partial \hat{\nu}}$$

and

$$\frac{\partial \mu_{dyn}}{\partial T} = \mu_{dyn} \frac{T + 3\mu_2}{2T(T + \mu_2)} \tag{80}$$

$$\frac{\partial \mu_{dyn}}{\partial U} = \frac{\partial \mu_{dyn}}{\partial T} \frac{\partial T}{\partial U} \tag{81}$$

$$\frac{\partial \mu_{tur}}{\partial U} = \hat{\nu} f_{v1} \left(1 + \frac{3c_{v1}^3}{\chi^3 + c_{v1}^3}\right) \frac{\partial \rho}{\partial U} - \chi f_{v1} \frac{3c_{v1}^3}{\chi^3 + c_{v1}^3} \frac{\partial \mu_{dyn}}{\partial U} \tag{82}$$

$$\frac{\partial \mu_{tur}}{\partial \hat{\nu}} = \rho f_{v1} \left(1 + \frac{3c_{v1}^3}{\chi^3 + c_{v1}^3}\right) \tag{83}$$

with

$$\frac{\partial T}{\partial U} = \frac{(\gamma - 1)}{R\rho} \left(|\vec{v}|^2 - E, -v_1, -v_2, -v_3, 1\right), \quad \frac{\partial \rho}{\partial U} = (1, 0, 0, 0, 0).$$

In (78) we have

$$\left.\begin{array}{ll} \vec{A}^c = \left(A_x^c, A_y^c, A_z^c\right), & A_i^c = \left.\frac{\partial \vec{F}_i^c}{\partial U}\right|_{U(x,y,z)} \\ \vec{A}^{vk} = \left(A_x^{vk}, A_y^{vk}, A_z^{vk}\right), & A_i^{vk} = \left.\frac{\partial \vec{F}_i^{vk}}{\partial U}\right|_{U(x,y,z)} \\ \mathbf{D}^{vk} = \begin{pmatrix} D_{xx}^{vk} & D_{xy}^{vk} & D_{xz}^{vk} \\ D_{yx}^{vk} & D_{yy}^{vk} & D_{yz}^{vk} \\ D_{zx}^{vk} & D_{zy}^{vk} & D_{zz}^{vk} \end{pmatrix}, & D_{ij}^{vk} = \left.\frac{\partial \vec{F}_i^{vk}}{\partial (\partial_j U)}\right|_{U(x,y,z)} \end{array}\right\} \quad i, j = 1 \ldots 3, \quad k = 1, 2$$

Defining for convenience $a_0 = (\gamma - 1)$, $\phi = (\gamma - 1)\frac{|\vec{v}|^2}{2}$, then we have

$$A_i^c = \begin{pmatrix} \cdot & \delta_{i1} & \delta_{i2} & \delta_{i3} & \cdot \\ -v_i v_1 + \delta_{i1}\phi & v_i - (a_0 - 1)v_i\delta_{i1} & v_1\delta_{i2} - a_0 v_2 \delta_{i1} & v_1\delta_{i3} - a_0 v_3 \delta_{i1} & a_0\delta_{i1} \\ -v_i v_2 + \delta_{i2}\phi & v_2\delta_{i1} - a_0 v_1 \delta_{i2} & v_i - (a_0 - 1)v_i\delta_{i2} & v_2\delta_{i3} - a_0 v_3 \delta_{i2} & a_0\delta_{i2} \\ -v_i v_3 + \delta_{i3}\phi & v_3\delta_{i1} - a_0 v_1 \delta_{i3} & v_3\delta_{i2} - a_0 v_2 \delta_{i3} & v_i - (a_0 - 1)v_i\delta_{i3} & a_0\delta_{i3} \\ v_i\left(\phi - H\right) & -a_0 v_i v_1 + H\delta_{i1} & -a_0 v_i v_2 + H\delta_{i2} & -a_0 v_i v_3 + H\delta_{i3} & \gamma v_i \end{pmatrix}$$

$$A_i^{v1} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\eta_{i1} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i1} & \partial_1\left(\frac{1}{\rho}\right)\delta_{i2} - \frac{2}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i1} & \partial_1\left(\frac{1}{\rho}\right)\delta_{i3} - \frac{2}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i1} & \cdot \\ -\eta_{i2} & \partial_2\left(\frac{1}{\rho}\right)\delta_{i1} - \frac{2}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i2} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i2} & \partial_2\left(\frac{1}{\rho}\right)\delta_{i3} - \frac{2}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i2} & \cdot \\ -\eta_{i3} & \partial_3\left(\frac{1}{\rho}\right)\delta_{i1} - \frac{2}{3}\partial_1\left(\frac{1}{\rho}\right)\delta_{i3} & \partial_3\left(\frac{1}{\rho}\right)\delta_{i2} - \frac{2}{3}\partial_2\left(\frac{1}{\rho}\right)\delta_{i3} & \partial_i\left(\frac{1}{\rho}\right) + \frac{1}{3}\partial_3\left(\frac{1}{\rho}\right)\delta_{i3} & \cdot \\ v_j\pi_{ij} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i1} + \zeta_{i1} + \frac{1}{\rho}\tau_{i1} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i2} + \zeta_{i2} + \frac{1}{\rho}\tau_{i2} & v_j\partial_j\left(\frac{1}{\rho}\right)\delta_{i3} + \zeta_{i3} + \frac{1}{\rho}\tau_{i3} & \cdot \end{pmatrix}$$

$$A_i^{v2} = \gamma \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0}\partial_i\left(\frac{\phi}{\rho} - \frac{P}{\rho^2}\right) & -\partial_i\left(\frac{v_1}{\rho}\right) & -\partial_i\left(\frac{v_2}{\rho}\right) & -\partial_i\left(\frac{v_3}{\rho}\right) & \partial_i\left(\frac{1}{\rho}\right) \end{pmatrix}$$

$$D_{ii}^{v1} = \frac{1}{\rho} \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i1}\right)v_1 & \left(1 + \frac{1}{3}\delta_{i1}\right) & \cdot & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i2}\right)v_2 & \cdot & \left(1 + \frac{1}{3}\delta_{i2}\right) & \cdot & \cdot \\ -\left(1 + \frac{1}{3}\delta_{i3}\right)v_3 & \cdot & \cdot & \left(1 + \frac{1}{3}\delta_{i3}\right) & \cdot \\ -|\vec{v}|^2 - \frac{1}{3}v_i^2 & \left(1 + \frac{1}{3}\delta_{i1}\right)v_1 & \left(1 + \frac{1}{3}\delta_{i2}\right)v_2 & \left(1 + \frac{1}{3}\delta_{i3}\right)v_3 & \cdot \end{pmatrix}$$

$$D_{ij}^{v1} = \frac{1}{\rho}\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ -v_i\delta_{j1} + \frac{2}{3}v_j\delta_{i1} & \delta_{j1}\delta_{i1} - \frac{2}{3}\delta_{i1}\delta_{j1} & \delta_{j1}\delta_{i2} - \frac{2}{3}\delta_{i1}\delta_{j2} & \delta_{j1}\delta_{i3} - \frac{2}{3}\delta_{i1}\delta_{j3} & \cdot \\ -v_i\delta_{j2} + \frac{2}{3}v_j\delta_{i2} & \delta_{j2}\delta_{i1} - \frac{2}{3}\delta_{i2}\delta_{j1} & \delta_{j2}\delta_{i2} - \frac{2}{3}\delta_{i2}\delta_{j2} & \delta_{j2}\delta_{i3} - \frac{2}{3}\delta_{i2}\delta_{j3} & \cdot \\ -v_i\delta_{j3} + \frac{2}{3}v_j\delta_{i3} & \delta_{j3}\delta_{i1} - \frac{2}{3}\delta_{i3}\delta_{j1} & \delta_{j3}\delta_{i2} - \frac{2}{3}\delta_{i3}\delta_{j2} & \delta_{j3}\delta_{i3} - \frac{2}{3}\delta_{i3}\delta_{j3} & \cdot \\ -\frac{1}{3}v_iv_j & v_j\delta_{i1} - \frac{2}{3}v_i\delta_{j1} & v_j\delta_{i2} - \frac{2}{3}v_i\delta_{j2} & v_j\delta_{i3} - \frac{2}{3}v_i\delta_{j3} & \cdot \end{pmatrix} \quad (i \neq j)$$

$$D_{ii}^{v2} = \frac{\gamma}{\rho}\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{a_0}\left(\phi - \frac{P}{\rho}\right) & -v_1 & -v_2 & -v_3 & 1 \end{pmatrix}$$

$$D_{ij}^{v2} = \mathbf{0}_{5\times 5} \quad (i \neq j)$$

where tensors $\bar{\eta}$, $\bar{\pi}$ and $\bar{\zeta}$ in the definition of $A_i^{v1}$ are given by

$$\begin{aligned} \eta_{ij} &= \partial_i\left(\frac{v_j}{\rho}\right) + \partial_j\left(\frac{v_i}{\rho}\right) - \frac{2}{3}\delta_{ij}\nabla\cdot\left(\frac{\vec{v}}{\rho}\right) \\ \pi_{ij} &= v_j\partial_i\left(\frac{1}{\rho}\right) + v_i\partial_j\left(\frac{1}{\rho}\right) - \frac{2}{3}\delta_{ij}\,\vec{v}\cdot\nabla\left(\frac{1}{\rho}\right) = \eta_{ij} - \frac{1}{\rho}\tau_{ij} \\ \zeta_{ij} &= v_j\partial_i\left(\frac{1}{\rho}\right) - v_i\partial_j\left(\frac{1}{\rho}\right) + \frac{1}{3}v_i\partial_j\left(\frac{1}{\rho}\right). \end{aligned}$$

### D. Linearized Spalart–Allmaras turbulence model

Here we compute the terms corresponding to the linearized turbulence equation (15). Note that

$$\frac{\partial R_{\hat{\nu}}}{\partial U}\delta U = \nabla\cdot(\vec{F}^{cv}\delta U) - F^s\delta U - \mathbf{M}^s\nabla\delta U \tag{84}$$

$$\frac{\partial R_{\hat{\nu}}}{\partial \hat{\nu}}\delta\hat{\nu} = \nabla\cdot\left(\vec{B}^{cv}\delta\hat{\nu} + \mathbf{E}^{cv}\nabla\delta\hat{\nu}\right) - B^s\delta\hat{\nu} - \mathbf{E}^s\nabla\delta\hat{\nu} \tag{85}$$

$$\frac{\partial R_{\hat{\nu}}}{\partial d_S}\delta d_S = -K^s\delta d_S \tag{86}$$

where $\mathbf{M}^s = \left(M_x^s, M_y^s, M_z^s\right)$, $\mathbf{E}^{cv} = \left(E_x^{cv}, E_y^{cv}, E_z^{cv}\right)$, $\mathbf{E}^s = \left(E_x^s, E_y^s, E_z^s\right)$.

The Jacobian matrices associated to the convective/viscous flux are given by

$$\vec{B}^{cv} = \frac{\partial\vec{T}^{cv}}{\partial\hat{\nu}} = -\frac{\nabla\hat{\nu}}{\sigma} + \vec{v} \tag{87}$$

$$E_i^{cv} = \frac{\partial\vec{T}^{cv}}{\partial(\partial_i\hat{\nu})} = -\frac{\nu+\hat{\nu}}{\sigma} \tag{88}$$

$$\vec{F}^{cv} = \frac{\partial\vec{T}^{cv}}{\partial U} = \alpha\left(\frac{\partial T}{\partial U}\right)^{\mathsf{T}}\nabla\hat{\nu} + \begin{pmatrix} \frac{\mu_{dyn}}{\sigma\rho^2}\nabla\hat{\nu} - \frac{\hat{\nu}}{\rho}\vec{v} \\ \frac{\hat{\nu}}{\rho}\mathbf{I}_3 \\ \cdot \end{pmatrix}, \qquad \alpha = -\frac{1}{\sigma\rho}\frac{\partial\mu_{dyn}}{\partial T}. \tag{89}$$

Here, $\frac{\partial\mu_{dyn}}{\partial T}$ and $\frac{\partial T}{\partial U}$ are given in §VII.C, and $\mathbf{I}_3$ is the $3\times 3$ identity matrix.

Concerning the derivatives of the source term $T^s$ we have

$$B^s = \frac{\partial T^s}{\partial\hat{\nu}} = \left(c_{b1}\hat{S} - 2c_{w1}f_w\frac{\hat{\nu}}{d_S^2}\right) - c_{w1}\left[\frac{\hat{\nu}}{d_S}\right]^2 f_w^g g^r r^{\hat{\nu}} + \Lambda\left(\hat{S}^{f_{v2}}\left(f_{v2}^{\chi} + f_{v2}^{v1}f_{v1}^{\chi}\right)\chi^{\hat{\nu}} + \hat{S}^{\hat{\nu}}\right) \tag{90}$$

$$F^s = \frac{\partial T^s}{\partial U} = \Lambda\left[\hat{S}^{f_{v2}}\left(f_{v2}^{\chi} + f_{v2}^{v1}f_{v1}^{\chi}\right)\chi^{\nu}\left(\nu^T\frac{\partial T}{\partial U} + \nu^{\rho}\frac{\partial\rho}{\partial U}\right)^{\mathsf{T}} + \frac{1}{|\vec{\omega}|}\left(\partial_i\vec{v} - \nabla v_i\right)\cdot\partial_i N\right] \tag{91}$$

$$K^s = \frac{\partial T^s}{\partial d_S} = \frac{2c_{w1}f_w\hat{\nu}^2}{d_S^3} + \Lambda\hat{S}^{d_S} \tag{92}$$

$$E_i^s = \frac{\partial T^s}{\partial(\partial_i \hat{\nu})} = 2\frac{c_{b2}}{\sigma}\partial_i \hat{\nu} \tag{93}$$

$$M_i^s = \frac{\partial T^s}{\partial(\partial_i U)} = \frac{1}{|\vec{\omega}|}\Lambda\left(\partial_i \vec{v} - \nabla v_i\right)\cdot N \tag{94}$$

with

$$\Lambda = \frac{\partial T^s}{\partial \hat{S}} = c_{b1}\hat{\nu} - c_{w1}\left[\frac{\hat{\nu}}{d_S}\right]^2 f_w^g g^r r^{\hat{S}} \tag{95}$$

and where $N$ in (91) and (94) is the $3\times 5$ matrix defined by $\delta\vec{v} = N\delta U$, i.e.

$$N = \frac{1}{\rho}\begin{pmatrix} -v_1 & 1 & \cdot & \cdot & \cdot \\ -v_2 & \cdot & 1 & \cdot & \cdot \\ -v_3 & \cdot & \cdot & 1 & \cdot \end{pmatrix}.$$

Finally, the whole set of partial derivatives needed to compute (90)-(95) is the following

$$f_w^g = \frac{\partial f_w}{\partial g} = \frac{c_{w3}^6}{g^6 + c_{w3}^6}\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6} \qquad g^r = \frac{\partial g}{\partial r} = 1 + c_{w2}\left(6r^5 - 1\right)$$

$$r^{\hat{\nu}} = \frac{\partial r}{\partial \hat{\nu}} = \frac{1}{\hat{S}\kappa^2 d_S^2} \qquad r^{\hat{S}} = \frac{\partial r}{\partial \hat{S}} = -\frac{\hat{\nu}}{\hat{S}^2\kappa^2 d_S^2}$$

$$r^{d_S} = \frac{\partial r}{\partial d_S} = -\frac{2\hat{\nu}}{\hat{S}\kappa^2 d_S^3} \qquad \hat{S}^{f_{v2}} = \frac{\partial \hat{S}}{\partial f_{v2}} = \frac{\hat{\nu}}{\kappa^2 d_S^2}$$

$$\hat{S}^{\hat{\nu}} = \frac{\partial \hat{S}}{\partial \hat{\nu}} = \frac{f_{v2}}{\kappa^2 d_S^2} \qquad \hat{S}^{d_S} = \frac{\partial \hat{S}}{\partial d_S} = -\frac{2\hat{\nu} f_{v2}}{\kappa^2 d_S^3}$$

$$f_{v2}^{\chi} = \frac{\partial f_{v2}}{\partial \chi} = -\frac{1}{\left(1 + \chi f_{v1}\right)^2} \qquad f_{v2}^{v1} = \frac{\partial f_{v2}}{\partial f_{v1}} = \frac{\chi^2}{\left(1 + \chi f_{v1}\right)^2}$$

$$f_{v1}^{\chi} = \frac{\partial f_{v1}}{\partial \chi} = \frac{3\chi^2 c_{v1}^3}{\left(\chi^3 + c_{v1}^3\right)^2} \qquad \chi^{\hat{\nu}} = \frac{\partial \chi}{\partial \hat{\nu}} = \frac{1}{\nu}$$

$$\chi^{\nu} = \frac{\partial \chi}{\partial \nu} = -\frac{\chi}{\nu} \qquad \nu^{\rho} = \frac{\partial \nu}{\partial \rho} = -\frac{\mu_{dyn}}{\rho^2}$$

$$\nu^T = \frac{\partial \nu}{\partial T} = \frac{1}{\rho}\frac{\partial \mu_{dyn}}{\partial T}.$$

## E. Adjoint formulas

In this section we give explicit formulas for the adjoint operators and boundary conditions. These are obtained from the identity (17), which is deduced from the following integration by parts

$$\begin{aligned}\int_\Omega \Psi_U^{\mathsf{T}}\left(\frac{\partial R_U}{\partial U}\delta U + \frac{\partial R_U}{\partial \hat{\nu}}\delta\hat{\nu}\right) &= \int_\Omega (A_U^U \Psi_U)^{\mathsf{T}}\delta U + \int_\Omega A_U^{\hat{\nu}}\Psi_U\delta\hat{\nu} \\ &+ \int_S \vec{\varphi}\cdot(\delta P\,\vec{n} - \delta\bar{\sigma}\cdot\vec{n}) + \int_S (\vec{g}_1\cdot\vec{\varphi} + g_2\partial_n\psi_5)\delta T + \int_S h_3\delta S \end{aligned} \tag{96}$$

$$\begin{aligned}\int_\Omega \psi_{\hat{\nu}}\left(\frac{\partial R_{\hat{\nu}}}{\partial U}\delta U + \frac{\partial R_{\hat{\nu}}}{\partial \hat{\nu}}\delta\hat{\nu} + \frac{\partial R_{\hat{\nu}}}{\partial d_S}\delta d_S\right) &= \int_\Omega A_{\hat{\nu}}^U\psi_{\hat{\nu}}\delta U + \int_\Omega A_{\hat{\nu}}^{\hat{\nu}}\psi_{\hat{\nu}}\delta\hat{\nu} + \int_\Omega A_{\hat{\nu}}^d\psi_d\delta\hat{\nu} \\ &+ \int_S \psi_{\hat{\nu}}g_3\partial_n(\delta\hat{\nu}) + \int_S \psi_{\hat{\nu}}g_4\delta P - \int_S \psi_{\hat{\nu}}g_5\delta T + \int_S \frac{1 + 2c_{b2}}{\sigma}\left(\partial_n\hat{\nu}\right)^2\psi_{\hat{\nu}}\delta S \end{aligned} \tag{97}$$

$$\int_\Omega \psi_d\nabla d_S\cdot\nabla\delta d_S = -\int_\Omega \nabla\cdot\left(\psi_d\nabla d_S\right)\delta d_S - \int_S \psi_d\delta S \tag{98}$$

where we have used $\partial_n d_S = 1$. Here, domain integrals on the right hand side contain the adjoint operators given by

$$A_U^U\Psi_U = -\nabla\Psi_U^{\mathsf{T}}\cdot\vec{A}^c - \nabla\cdot\left(\nabla\Psi_U^{\mathsf{T}}\cdot\mu_{tot}^k\mathbf{D}^{vk}\right) + \nabla\Psi_U^{\mathsf{T}}\cdot\mu_{tot}^k\vec{A}^{vk} + \nabla\Psi_U^{\mathsf{T}}\cdot\vec{F}^{vk}\frac{\partial\mu_{tot}^k}{\partial U} \tag{99}$$

$$A_{\tilde{\nu}}^{U}\psi_{\hat{\nu}} = -\nabla\psi_{\tilde{\nu}}\cdot\vec{F}^{cv} - \psi_{\hat{\nu}}F^{s} + \nabla\cdot(\psi_{\hat{\nu}}\mathbf{M}^{s}) \tag{100}$$

$$A_{U}^{\hat{\nu}}\Psi_{U} = \nabla\Psi_{U}^{\mathsf{T}}\cdot\vec{F}^{vk}\frac{\partial\mu_{tot}^{k}}{\partial\hat{\nu}} \tag{101}$$

$$A_{\hat{\nu}}^{\hat{\nu}}\psi_{\hat{\nu}} = -\nabla\psi_{\hat{\nu}}\cdot\vec{B}^{cv} + \nabla\cdot(\nabla\psi_{\hat{\nu}}\cdot\mathbf{E}^{cv}) - \psi_{\hat{\nu}}B^{s} + \nabla\cdot(\psi_{\hat{\nu}}\mathbf{E}^{s}) \tag{102}$$

$$A_{\hat{\nu}}^{d}\psi_{\hat{\nu}} = -K^{s}\psi_{\hat{\nu}} \tag{103}$$

$$A_{d}^{d}\psi_{d} = -\nabla\cdot(\psi_{d}\nabla d_{S})\,. \tag{104}$$

On the other hand, the terms $\vec{g}_1$, $g_2$, $g_3$, $g_4$, $g_5$ and $h_3$ appearing in the boundary integrals in (96)-(98) are given by

$$\vec{g}_1 = -\frac{\partial\mu_{dyn}}{\partial T}\vec{n}\cdot\bar{\tau}, \quad g_2 = C_p\mu_{tot}^2, \quad g_3 = -\frac{\nu}{\sigma} \tag{105}$$

$$g_4 = \partial_n\hat{\nu}\,\frac{\mu_{dyn}}{\sigma\rho P}, \quad g_5 = \partial_n\hat{\nu}\frac{1}{\sigma\rho}\left(\frac{\partial\mu_{dyn}}{\partial T} + \frac{\mu_{dyn}}{T}\right)$$

$$\begin{aligned} h_3 = {} & -(\rho\psi_1 + \rho H\psi_5)(\partial_n\vec{v}\cdot\vec{n}) + \psi_5\vec{n}\cdot\bar{\sigma}\cdot\partial_n\vec{v} - \vec{n}\cdot\bar{\Sigma}^{\varphi}\cdot\partial_n\vec{v} \\ & -\psi_5\bar{\sigma}\colon\nabla\vec{v} + \mu_{tot}^2 C_p\nabla_S\psi_5\cdot\nabla_S T + \partial_n\hat{\nu}(\vec{n}\cdot\bar{\tau}\cdot\vec{\varphi})\frac{\partial\mu_{tur}}{\partial\hat{\nu}} \end{aligned} \tag{106}$$

where $\bar{\sigma}\colon\nabla\vec{v} = \sigma_{ij}\partial_i v_j$, with

$$\bar{\Sigma}^{\varphi} = \mu_{tot}^{1}\left(\nabla\vec{\varphi} + \nabla\vec{\varphi}^{\mathsf{T}} - \mathbf{I}_d\frac{2}{3}\nabla\cdot\vec{\varphi}\right).$$

Some of the terms in $h_3$ above can be simplified. In particular, taking into account that $\vec{v} = 0$ and $\nabla_S\vec{v} = 0$ on the obstacle surface, we have $\nabla\vec{v} = \partial_j v_i = \partial_n v_i n_j$. Therefore

$$\bar{\sigma}\colon\nabla\vec{v} = \sigma_{ij}\partial_n v_i n_j = \vec{n}\cdot\bar{\sigma}\cdot\partial_n\vec{v} \qquad \text{on } S,$$

and the second and fourth terms in (106) cancel.

On the other hand, the continuity equation yields $\nabla\cdot\vec{v} = 0$ on $S$. Hence

$$0 = \nabla\cdot\vec{v} = \partial_i v_i = \partial_n v_i n_i = \partial_n\vec{v}\cdot\vec{n} \qquad \text{on } S,$$

and the first term in (106) also cancels.

In the same way, one may notice that also $\frac{\partial\mu_{tur}}{\partial\hat{\nu}} = 0$ on $S$, since $f_{v1} = 0$ on the obstacle surface. Thus, the term $h_3$ reads

$$h_3 = -\vec{n}\cdot\bar{\Sigma}^{\varphi}\cdot\partial_n\vec{v} + \mu_{tot}^2 C_p\nabla_S\psi_5\cdot\nabla_S T. \tag{107}$$

Finally, adding the three terms in (96)-(98) we easily obtain (17) with

$$\hat{g} = h_3 + \frac{1+2c_{b2}}{\sigma}\left(\partial_n\hat{\nu}\right)^2\psi_{\hat{\nu}}. \tag{108}$$

## VIII. Acknowledgements

The research described in this paper has been supported under the FuSim–E Programme, funded by Airbus Spain, and by Grant MTM2008-03541 of the MICINN (Spain). The fourth author is also supported by project PI2010-04 of the Basque Government and the ERC Advanced Grant FP7-246775 NUMERIWAVES.

## References

[1]W.K. Anderson and D.L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.

[2]W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA Paper*, 97-0643, 1997.

[3]C.L. Bottasso, D. Detomi, and R. Serra. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Comput. Methods Appl. Mech. Engrg.*, 194:4244–4264, 2005.

[4]C. Castro, C. Lozano, F. Palacios, and E. Zuazua. A systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA Journal*, 45(9):2125–2139, 2007.

[5]P. Cook, M. McDonald, and M. Firmin. Aerofoil rae2822 pressure distributions, and boundary layer and wake measurements. Technical Report AGARD 138, 1979.

[6]C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Comput. Struct.*, 80:305–316, 2002.

[7]R.P. Dwight and J. Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA Journal*, 44 (12):3022–3031, 2006.

[8]R.N. Elias, M.A.D. Martins, and A.L.G.A. Coutinho. Simple finite element-based computation of distance functions in unstructured grids. *Int. J. Numer. Meth. Engng.*, 72:1095–1110, 2007.

[9]P. Eliasson. Edge, a Navier-Stokes solver for unstructured grids. Technical Report FOI-R-0298-SE, FOI Scientific Report, 2002.

[10]W. Haase, F. Bradsma, E. Elsholz, M. Leschziner, and D. Schwamborn, editors. *EUROVAL – An European initiative on Validation of CFD codes*, volume 42 of *Notes on Numerical Fluid Mechanics*. John Wiley and Sons, Inc., 1993.

[11]R.M. Hicks and P.A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[12]C. Hirsch. *Numerical Computation of Internal and External Flows*. Wiley, New York, 1984.

[13]A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[14]A. Jameson, J. Alonso, and L. Martinelli. Multigrid unsteady Navier-Stokes calculations with aeroelastic applications. *AIAA Paper*, 95-0048, 1995.

[15]A. Jameson, L. Martinelli, and F. Grasso. A multigrid method for the Navier-Stokes equations. *AIAA Paper*, 86-0208, 1986.

[16]A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the euler equations by finite volume methods using runge-kutta time stepping schemes. *AIAA Paper*, 81-1259, 1981.

[17]A. Jameson and E. Turkel. Implicit schemes and lu-decompositions. *Mathematics of Computation*, 37:385–397, 1981.

[18]C.S. Kim, C. Kim, and O.H. Rho. Feasibility study of constant eddy-viscosity assumption in gradient-based design optimization. *J. Aircraft*, 40 (6):1168–1176, 2003.

[19]L.D. Landau and E.M. Lifshitz. *Fluid Mechanics (2nd Edition)*. Pergamon Press, 1993.

[20]B.J. Lee and C. Kim. Automated design methodology of turbulent internal flow using discrete adjoint formulation. *Aerospace Science and Technology*, 11:163–173, 2007.

[21]D.J. Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):740–750, 2007.

[22]E. Nielsen, J. Lu, M.A. Park, and D.L. Darmofal. An implicit exact dual adjoint solution method for turbulent flows on unstructured frids. *Computers and Fluids*, 33:1131–1155, 2004.

[23]F. Palacios, A. Bueno-Orovio, C. Castro, and E. Zuazua. Cades: An open-source arquitecture for flow simulation and optimal shape design in aerodynamics. *In preparation*.

[24]O. Pironneau. On optimum design in fluid mechanics. *J. Fluid Mech.*, 64:97–110, 1974.

[25]O. Pironneau. *Finite element methods for fluids*. John Wiley and Sons, Inc., New York, 1990.

[26]J. A. Samareh. Aerodynamic shape optimization based on free–form deformation. *AIAA Paper*, 2004-4630, 2004.

[27]J.A. Sethian and A. Vladimirsky. Fast methods for the eikonal and related hamilton-jacobi equations on unstructured meshes. *Proc. Natl. Acad. Sci. USA*, 97(11):5699–5703, 2000.

[28]J.-P. Sokolowski, J. Zolesio. *Introduction to shape optimization*. Springer Verlag, 1991.

[29]P Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA Paper*, 92-0439, 1992.

[30]V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. *AIAA Journal*, 93-0880, 1993.

[31]J.M. Weiss, J.P. Maruszewski, and A.S. Wayne. Implicit solution of the Navier-Stokes equation on unstructured meshes. *AIAA Journal*, 97-2103, 1997.

[32]F.M. White. *Viscous Fluid Flow*. McGraw Hill Inc., 1974.

[33]D.C. Wilcox. *Turbulence Modeling for CFD*. 2nd Ed., DCW Industries, Inc., 1998.

[34]A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and Othmer C. Continuous adjoint approach to the spallart-allmaras turbulence model for incompressible flows. *Computers and Fluids*, 38:1528–1538, 2009.