AIAA JOURNAL
Vol. 45, No. 9, September 2007

Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids

Carlos Castro*
Universidad Politécnica de Madrid, 28040 Madrid, Spain
Carlos Lozano† and Francisco Palacios‡
Instituto Nacional de Técnica Aeroespacial (INTA),
Torrejón de Ardoz, 28850 Madrid, Spain
and
Enrique Zuazua§
Universidad Autónoma de Madrid, 28049 Madrid, Spain

DOI: 10.2514/1.24859

A continuous adjoint approach to aerodynamic design for viscous compressible flow on unstructured grids is developed. Sensitivity gradients are computed using tools of shape deformation of boundary integrals. The resulting expressions involve second-order derivatives of the flow variables that require numerical solvers with greater than second-order accuracy. A systematic way of reducing the order of these terms is presented. The accuracy of the sensitivity derivatives is assessed by comparison with finite-difference computations, and the validity of the overall methodology is illustrated with several design examples.

Nomenclature

$\boldsymbol{A}$	=	inviscid Jacobian vector
$\boldsymbol{A}^v$	=	viscous Jacobian vector
C_D	=	drag coefficient
C_L	=	lift coefficient
C_p	=	pressure coefficient
C_∞	=	$\gamma M_\infty^2 P_\infty/2$, reference dynamic pressure
$C_p^{(d)}$	=	target pressure coefficient
c_p	=	specific heat at constant pressure
D_{ij}	=	matrix of derivatives of the viscous flux with respect to the gradients of the primitive variables
$\boldsymbol{d}$	=	force direction vector
ds	=	curve/surface integration measure
E	=	total energy
$\boldsymbol{e}$	=	unit tangent vectors on a surface
$\boldsymbol{F}$	=	Cartesian inviscid flux vector
$\boldsymbol{F}^v$	=	Cartesian viscous flux vector
f, g, h	=	generic functions
$\boldsymbol{f}$	=	force vector
$\boldsymbol{f}^*$	=	nondimensional force vector
G	=	surface sensitivity (local gradient)
g_{ab}	=	metric tensor on a surface
g^{ab}	=	inverse metric tensor on a surface
H	=	total enthalpy
H_m	=	mean curvature of a surface
I_{eq}	=	common surface contribution to the gradient of the cost function
J	=	cost function
K	=	relative variation of integration measure
k	=	coefficient of thermal conductivity
L	=	matrix of left eigenvectors of inviscid Jacobian
L_{ab}	=	second fundamental form of a surface
M	=	$\partial U/\partial V$, transformation matrix between conservative and primitive flow variables
M_∞	=	freestream Mach number
$\boldsymbol{n}$	=	unit normal vector
P	=	static pressure
P_∞	=	freestream pressure
Pr	=	Prandtl number
R	=	gas constant
S	=	wall boundary
s	=	arc-length curve parameter
T	=	temperature
t	=	time
$\boldsymbol{t}$	=	unit tangent vector
U	=	conservative flow variables
u, v, v_i	=	Cartesian velocity components
V	=	primitive flow variables
$\boldsymbol{v}$	=	velocity vector
W	=	characteristic variables
$\boldsymbol{x}$	=	Cartesian coordinate vector
x, y, z	=	Cartesian coordinates
α	=	angle of attack; also normal boundary deformation
$\beta, \boldsymbol{\beta}$	=	tangent boundary deformation
Γ_∞	=	"far field"
Γ_{ca}^b	=	Christoffel symbols of a surface
γ	=	ratio of specific heats
Δ	=	first difference
δ	=	first variation
δ_{ij}	=	Kronecker delta function
η	=	curve parameter, also, second surface parameter
κ	=	curvature of a curve
Λ	=	diagonal matrix of inviscid eigenvalues
μ	=	dynamic viscosity
μ_t	=	turbulent viscosity

Presented as Paper 51 at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January 2006; received 28 April 2006; revision received 5 December 2006; accepted for publication 9 April 2007.

*Assistant Professor, Department of Mathematics and Computation, School of Civil Engineering.

†Research Scientist, Fluid Dynamics Branch, Department of Aerodynamics and Propulsion.

‡Research Scientist, Fluid Dynamics Branch, Department of Aerodynamics and Propulsion. Member AIAA.

§Professor, Department of Mathematics, Faculty of Sciences.

ξ, η	=	surface parameters
ρ	=	density
Σ	=	adjoint stress tensor
σ	=	Reynolds stress tensor
$\boldsymbol{\varphi}$	=	adjoint velocity vector
Ψ, ψ	=	adjoint variables
Ω	=	fluid domain

Subscripts

$a, b, \ldots$	=	ath, bth, ... parametric directions on a surface (covariant)
i, j, k	=	Cartesian ith, jth, kth directions
n	=	normal derivative
tg	=	tangent derivative
x, y	=	Cartesian x, y components of a vector
∞	=	freestream reference quantity
$\vert_S$	=	quantity evaluated on the boundary S

Superscripts

$a, b, \ldots$	=	ath, bth, ... parametric directions on a surface (contravariant)
T	=	transpose of a matrix
v	=	viscous flow quantities
-1	=	inverse of a matrix
$\cdot$	=	derivative with respect to the curve parameter

Mathematical symbols

$\boldsymbol{\nabla}$, ∇	=	nabla (gradient) operator
$\boldsymbol{\nabla}_{\mathrm{tg}}$	=	tangent derivative on a surface
∂	=	partial derivative
∂_a	=	$\partial/\partial\xi^a$
∂_i	=	$\partial/\partial x_i$
∂_n	=	normal derivative to curve/surface
∂_{tg}	=	tangent derivative to a curve
∂_x	=	$\partial/\partial x$
∂_y	=	$\partial/\partial y$
$\partial\Omega$	=	fluid domain boundary

I. Introduction

THE use of computational fluid dynamics (CFD) tools in aerodynamic design optimization has grown in importance 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.

Gradients can be computed in a variety of ways, the most actively pursued recently being adjoint methods [1–5], which allow the solution of general sensitivity analysis problems governed by fluid dynamics models ranging from the full potential equation to the full compressible Reynolds-averaged Navier–Stokes equations. Adjoint methods are conventionally divided into continuous and discrete. In the continuous approach, the adjoint equations are derived from the governing partial differential equations (PDEs) and then subsequently discretized, whereas in the discrete approach the adjoint equations are directly derived from the discretized governing equations.

Although the discrete adjoint method should give gradients which are closer in value to exact finite-difference gradients, the continuous adjoint method has the advantage that the adjoint system has a unique form independent of the scheme used to solve the flowfield system. Numerical studies have shown that in typical shape optimization problems in transonic flow the differences are small enough that they have no significant effect on the final result [6].

The present work focuses on the continuous adjoint approach on unstructured grids, for which several limitations have been uncovered in the past such as the apparent inability of the method to handle arbitrary cost functions and the need of flow solvers with greater than second-order accuracy. The first problem is inherent to the continuous approach (either inviscid or viscous, on structured as well as unstructured grids) [7–10], but has not been encountered so far in the discrete adjoint approach. As for the second problem, which was first pointed out in [7], it stems from the need to compute accurate second-order derivatives of the flow variables, which are required for obtaining sensitivity derivatives from the continuous adjoint approach for viscous flows on unstructured grids. As second-order derivatives computed using data from a spatially second-order accurate scheme (which are by far the most commonly encountered schemes in unstructured flow solvers) are not consistent, in general, this issue has been actually one of the major drawbacks to using the continuous adjoint method on unstructured grids. On structured grids, where mapping to a fixed computational space is possible, this problem can be avoided [9]. The mapping technique has been extended to optimization on unstructured grids for inviscid flows [11,12], but a generalization to viscous flows is still lacking.

The present work aims precisely at filling some of those gaps by presenting a systematic continuous adjoint formulation for design optimization for viscous flows which is suitable for unstructured as well as structured grids. The point of view adopted here is similar to that in [7] and solves some of the drawbacks presented there. Indeed, the need for accurate second-order derivatives of the flow variables required for computing sensitivity derivatives for viscous flows is solved with the development of a systematic way of reducing the order of the higher derivative terms, which essentially amounts to using the flow equations restricted to the boundary to convert normal to tangent derivatives, and integrate by parts the latter to reduce the overall order of derivatives. A few remarks concerning the class of admissible optimization functionals are also made. In particular, it is shown that, for steady, compressible viscous flows, arbitrary functions of the pressure alone can be naturally optimized, and that there is no need to formally include components from the viscous stress tensor to obtain suitable boundary conditions for the viscous adjoint equations.

The organization of the paper is as follows. The exposition begins with a brief introduction to the continuous adjoint approach and a detailed review of its application to aerodynamic design using the Euler and Navier–Stokes equations. The caveats of the approach are discussed and a proposal of resolution is put forward. Next, the practical implementation of the method is described and supporting numerical results are presented. Finally, an appendix contains a compilation of useful formulas.

II. Continuous Adjoint Approach to Aerodynamic Design Optimization

A. Formulation of the Problem

In what follows we will be interested in design optimization problems within the continuous adjoint approach. In aeronautic applications, the basic setup comprises a fluid domain Ω bounded by a typically disconnected boundary $\partial\Omega$ which is conventionally divided into a "far-field" component Γ_∞ and a wall boundary S. Aeronautic optimization problems seek the minimization of a certain cost function, such as the deviation of the pressure on S from a prescribed pressure distribution in the so-called inverse design problems, or integrated force coefficients (drag or lift) in force optimization problems. In these examples the cost function can be defined as an integral over the wall boundary S of a suitable function f of the flow variables (collectively referred to as U)

$$J = \int_S f(U)\,\mathrm{d}s \qquad (1)$$

where ds denotes the appropriate integration measure. Cost functions involving domain integrals are also possible, but those will not be considered in the present work.

Upon deformation of the control surface S, the cost function varies due to the variation of the geometry and the change in the solution induced by the deformation. Accordingly, the variation of the cost function comprises essentially two terms

$$\delta J = \int_{\delta S} f(U)\,\mathrm{d}s + \int_S \frac{\partial f}{\partial U}\delta U\,\mathrm{d}s \tag{2}$$

The first, "geometric," term can be expanded as follows [13]:

$$\int_{\delta S} f(U)\,\mathrm{d}s = \int_S \frac{\partial f}{\partial U}(\delta \boldsymbol{x}\cdot \boldsymbol{\nabla} U)\,\mathrm{d}s + \int_S f\delta\,\mathrm{d}s \tag{3}$$

where $\delta\boldsymbol{x}$ stands for the deformation of the points defining the boundary S and δ ds denotes the appropriate change in the measure. If f contains geometric quantities such as the unit normal to the boundary $\boldsymbol{n}$—as is the case, for example, in force optimization problems—Eq. (3) takes the form

$$\int_{\delta S} f(U,\boldsymbol{n})\,\mathrm{d}s = \int_S \frac{\partial f}{\partial U}(\delta \boldsymbol{x}\cdot \boldsymbol{\nabla} U)\,\mathrm{d}s + \int_S \frac{\partial f}{\partial \boldsymbol{n}}\cdot\delta\boldsymbol{n}\,\mathrm{d}s + \int_S f\delta\,\mathrm{d}s \tag{4}$$

where $\delta\boldsymbol{n}$ is the variation in the boundary normal induced by the deformation of the boundary. The contribution of Eq. (4) is readily computable once the boundary deformation as well as the solution to the flow equations in the unperturbed geometry is known.

As for the second term of Eq. (2), it involves the variation δU of the flow variables under the perturbation. These can be obtained a priori from the solution of the linearized flow equations (subject to the appropriate boundary conditions), but this requires a flow evaluation per independent perturbation. If the design space is large, the computational cost is prohibitive. A convenient shortcut can be found by resorting to the adjoint equations, which can be understood, in a variational context, as consistency conditions for the Lagrange multipliers (the adjoint variables) which enforce the flow equations [2,3].

B. Aerodynamic Design with the Euler Equations

For the sake of completeness, the case of steady inviscid, two-dimensional compressible flow will be addressed first. Although the results that will be presented are not new, the discussion will serve as an introduction to the method and to illustrate how the same approach can accommodate inviscid as well as viscous optimization problems within a fully systematic and unified viewpoint.

The governing equations in this case are

$$\boldsymbol{\nabla}\cdot\boldsymbol{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = 0 \quad \text{in } \Omega \qquad U = \begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}$$
$$F_x = \begin{pmatrix}\rho u\\ \rho u^2 + P\\ \rho u v\\ \rho u H\end{pmatrix}, \qquad F_y = \begin{pmatrix}\rho v\\ \rho u v\\ \rho v^2 + P\\ \rho v H\end{pmatrix} \tag{5}$$

In these definitions, ρ is the density, u and v are the Cartesian velocity components, E is the total energy, and P and H are the pressure and enthalpy, given by the following relations:

$$P = (\gamma - 1)\rho\left[E - \frac{1}{2}(u^2 + v^2)\right], \qquad H = E + \frac{P}{\rho} \tag{6}$$

where γ is the ratio of specific heats. Equations (5) are subject to characteristic-type boundary conditions [14] on the far-field boundary Γ_∞, and to nontranspiration boundary conditions on solid wall boundaries

$$\boldsymbol{v}\cdot\boldsymbol{n} = u n_x + v n_y = 0 \quad \text{on } S, \qquad \boldsymbol{v} = (u, v), \qquad \boldsymbol{n} = (n_x, n_y) \tag{7}$$

The next step in the adjoint approach amounts to defining a suitable cost function. Conventional cost functions include specified pressure distributions (inverse design), force (drag or lift) or moment coefficients, efficiency (i.e., lift over drag), etc. For inverse design the appropriate definitions are

$$J = \frac{1}{2}\int_S (\Delta C_p)^2\,\mathrm{d}s \qquad \Delta C_p = C_p - C_p^{(d)}$$
$$C_p = \frac{P - P_\infty}{C_\infty}, \qquad C_\infty = \frac{1}{2}\gamma M_\infty^2 P_\infty \tag{8}$$

where $C_p^{(d)}$ is the target pressure coefficient distribution, and M_∞ and P_∞ are the freestream Mach number and pressure, respectively, whereas for force optimization

$$J = \int_S C_p(n_x\cos\alpha + n_y\sin\alpha)\,\mathrm{d}s = C_D \qquad \text{(drag coefficient)}$$
$$J = \int_S C_p(-n_x\sin\alpha + n_y\cos\alpha)\,\mathrm{d}s = C_L \qquad \text{(lift coefficient)} \tag{9}$$

or, in compact notation,

$$J = \int_S C_p(\boldsymbol{n}\cdot\boldsymbol{d})\,\mathrm{d}s, \qquad \boldsymbol{d} = \begin{cases}(\cos\alpha, \sin\alpha) & \text{(drag)}\\ (-\sin\alpha, \cos\alpha) & \text{(lift)}\end{cases} \tag{10}$$

where α is the angle of attack. S is a closed curve corresponding to the airfoil profile (or a disjoint union of several curves in the case of high-lift devices) which can be described by the parameterization $\boldsymbol{x}(\eta) = (x(\eta), y(\eta))$ with parameter η. Dot notation will be used to indicate differentiation with respect to η. If $\boldsymbol{t} = (t_x, t_y)$ denotes the unit tangent vector to the curve, the following holds:

$$\text{united tangent vector}: \boldsymbol{t} = \frac{\dot{\boldsymbol{x}}}{|\dot{\boldsymbol{x}}|}, \qquad |\dot{\boldsymbol{x}}| = \left|\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\eta}\right| = \sqrt{\dot{x}^2 + \dot{y}^2}$$
$$\text{unit normal vector: } \boldsymbol{n} = \frac{(-\dot{y}, \dot{x})}{|\dot{\boldsymbol{x}}|}$$
$$\text{integration measure: } \mathrm{d}s = |\dot{\boldsymbol{x}}|\mathrm{d}\eta$$
$$\text{curvature: } \kappa = \frac{\dot{x}\,\ddot{y} - \ddot{x}\,\dot{y}}{|\dot{\boldsymbol{x}}|^3} = \frac{\boldsymbol{n}\cdot\ddot{\boldsymbol{x}}}{|\dot{\boldsymbol{x}}|^2}$$
$$\text{tangent derivatives: } \partial_{\mathrm{tg}} f(\eta) = \boldsymbol{t}\cdot\boldsymbol{\nabla} f = \frac{\mathrm{d}f}{\mathrm{d}s} = \frac{1}{|\dot{\boldsymbol{x}}|}\frac{\mathrm{d}f}{\mathrm{d}\eta} = \frac{1}{|\dot{\boldsymbol{x}}|}\dot{f}$$
$$\text{differential relations: } \partial_{\mathrm{tg}}\boldsymbol{t} = \frac{1}{|\dot{\boldsymbol{x}}|}\dot{\boldsymbol{t}} = \kappa\boldsymbol{n}, \qquad \partial_{\mathrm{tg}}\boldsymbol{n} = \frac{1}{|\dot{\boldsymbol{x}}|}\dot{\boldsymbol{n}} = -\kappa\boldsymbol{t} \tag{11}$$

where a parameterization is picked such that $\boldsymbol{n}$ is the exterior unit normal. A generic deformation of the boundary can be described as follows:

$$\delta\boldsymbol{x}(\eta) = \alpha(\eta)\boldsymbol{n} + \beta(\eta)\boldsymbol{t} \tag{12}$$

where tangential and normal deformations have been explicitly separated, being quantified by $\beta(\eta)$ and $\alpha(\eta)$—not to be confused with the angle of attack—, respectively. [Even though it is a standard fact that every sufficiently small deformation of a curve can be described by a normal deformation alone (tangent deformations being equivalent to reparameterizations of the curve), the use of nonnormal deformations is nevertheless so widespread in the aeronautics literature that we prefer to keep the tangent component explicitly.]

For sufficiently small values of the deformation, the following holds:

$$
\begin{gathered}
\delta\dot{\boldsymbol{x}} = (\dot{\alpha} + \beta|\dot{\boldsymbol{x}}|\kappa)\boldsymbol{n} + (\dot{\beta} - \alpha|\dot{\boldsymbol{x}}|\kappa)\boldsymbol{t} \\
\delta\,\mathrm{d}s = \left(\frac{\dot{\boldsymbol{x}}\cdot\delta\dot{\boldsymbol{x}}}{\dot{x}^2+\dot{y}^2}\right)\mathrm{d}s = (\partial_{\mathrm{tg}}\beta - \alpha\kappa)\,\mathrm{d}s \qquad \delta\boldsymbol{t} = (\beta\kappa + \partial_{\mathrm{tg}}\alpha)\boldsymbol{n} \\
\delta\boldsymbol{n} = -(\beta\kappa + \partial_{\mathrm{tg}}\alpha)\boldsymbol{t}
\end{gathered}
\tag{13}
$$

By using Eqs. (2–4) and (11–13), the variation of the said cost functions (8) and (10) are

$$
\delta\left(\frac{1}{2}\int_S(\Delta C_p)^2\,\mathrm{d}s\right) = \int_S\left(\frac{\Delta C_p}{C_\infty}(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P) + K\frac{(\Delta C_p)^2}{2}\right)\mathrm{d}s + \frac{1}{C_\infty}\int_S(\Delta C_p\delta P)\,\mathrm{d}s \tag{14}
$$

where $K = (\dot{\boldsymbol{x}}\cdot\delta\dot{\boldsymbol{x}})/(\dot{x}^2+\dot{y}^2)$ for inverse design, and

$$
\delta\int_S C_p(\boldsymbol{n}\cdot\boldsymbol{d})\,\mathrm{d}s = \int_S\left(\frac{1}{C_\infty}(\boldsymbol{n}\cdot\boldsymbol{d})(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P) + KC_p(\boldsymbol{d}\cdot\boldsymbol{n}) + C_p(\boldsymbol{d}\cdot\delta\boldsymbol{n})\right)\mathrm{d}s + \frac{1}{C_\infty}\int_S((\boldsymbol{n}\cdot\boldsymbol{d})\delta P)\,\mathrm{d}s \tag{15}
$$

for force optimization. The terms in Eqs. (14) and (15) and which involve the pressure variation δP cannot be computed without explicitly solving the linearized flow equations. The alternate strategy, which has now become standard lore, is to resort to the adjoint equations, which give an elegant and computationally economical way to computing the unknown terms. The starting point is the linearized flow equations, which in the inviscid 2-D case are

$$
\begin{gathered}
\boldsymbol{\nabla}\cdot\delta\boldsymbol{F} = \partial_x((A_x)^TM^{-1}\delta U) + \partial_y((A_y)^TM^{-1}\delta U) \\
= \boldsymbol{\nabla}\cdot(\boldsymbol{A}^TM^{-1}\delta U) = 0 \quad \text{in } \Omega \\
\boldsymbol{A}^T = ((A_x)^T, (A_y)^T), \qquad (A_x)^T = \frac{\partial F_x}{\partial V}, \qquad (A_y)^T = \frac{\partial F_y}{\partial V} \\
M^{-1} = \frac{\partial V}{\partial U}, \qquad V = \begin{pmatrix}\rho\\ u\\ v\\ P\end{pmatrix}
\end{gathered}
\tag{16}
$$

where, for later convenience, the inviscid Jacobians have been written in terms of primitive variables V (see the Appendix for details). Boundary conditions for δU at the wall are obtained from the linearization of the nontranspiration boundary condition Eq. (7) on the wall

$$
\boldsymbol{n}\cdot\delta\boldsymbol{v} = -[(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla})\boldsymbol{v}]\cdot\boldsymbol{n} - \delta\boldsymbol{n}\cdot\boldsymbol{v} \quad \text{on } S \tag{17}
$$

On the far-field boundary, the appropriate boundary conditions are obtained from those of U as follows. Let L^{-1} be the matrix of left eigenvectors of the Jacobian $\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1}$ and Λ the diagonal matrix of eigenvalues. Therefore,

$$
\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1} = L\Lambda L^{-1}, \qquad W = L^{-1}U \tag{18}
$$

where W are characteristic variables. On the far-field boundary with characteristic-type boundary conditions on the flow variables, the propagation of information is based on the sign of the eigenvalues. Along incoming characteristics, that is, for negative eigenvalues, the corresponding characteristic variables are given in terms of freestream quantities such as the Mach number, the angle of attack, etc. If those are kept fixed by the perturbation, the corresponding variations vanish

$$
L^{-1}\delta U|_{\text{neg. eigenvalues}} = 0 \tag{19}
$$

The linearized flow equations are next multiplied by the vector of adjoint variables

$$
\Psi^T = \begin{pmatrix}\psi_1\\ \psi_2\\ \psi_3\\ \psi_4\end{pmatrix} \tag{20}
$$

and integrated over the flow domain Ω

$$
0 = \int_\Omega \Psi(\boldsymbol{\nabla}\cdot\delta\boldsymbol{F})\,\mathrm{d}\Omega = \int_\Omega \Psi(\partial_x((A_x)^TM^{-1}\delta U) + \partial_y((A_y)^TM^{-1}\delta U))\,\mathrm{d}\Omega \tag{21}
$$

Integrating by parts in Eq. (21) gives

$$
0 = -\int_\Omega \delta U^T(M^{-1})^T(\boldsymbol{A}\cdot\boldsymbol{\nabla}\Psi^T)\,\mathrm{d}\Omega + \int_{\Gamma_\infty}\Psi(\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1})\delta U\,\mathrm{d}s + \int_S \Psi(\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1})\delta U\,\mathrm{d}s \tag{22}
$$

Each of the three terms in Eq. (22) is forced to vanish independently. The first one, which is a domain integration, vanishes provided that Ψ satisfies the steady-state inviscid adjoint equation

$$
(M^{-1})^T\boldsymbol{A}\cdot\boldsymbol{\nabla}\Psi^T = 0 \tag{23}
$$

On the far-field boundary, incoming characteristics for Ψ correspond to outgoing characteristics for δU, and vice versa. Consequently, in view of Eq. (19) it is possible to choose boundary conditions for Ψ such that

$$
\Psi(\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1})\delta U = (\Psi L)\Lambda(L^{-1}\delta U) = 0 \tag{24}
$$

that is, by setting to zero the adjoint variables corresponding to outgoing characteristics [2,7] or positive Λ eigenvalues.

$$
\Psi L|_{\text{pos. eigenvalues}} = 0 \tag{25}
$$

Along incoming characteristics, the corresponding adjoint variables on the boundary are extrapolated from the interior of the domain.

All that is left from Eq. (22) is a boundary contribution at the solid wall S, which boils down to the following relation:

$$
0 = \int_S \Psi(\boldsymbol{n}\cdot\boldsymbol{A}^TM^{-1})\delta U\,\mathrm{d}s = \int_S(\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\,\mathrm{d}s + \int_S(\boldsymbol{n}\cdot\boldsymbol{\varphi})\delta P\,\mathrm{d}s \tag{26}
$$

where, for convenience, the vector $\boldsymbol{\varphi} = (\psi_2, \psi_3)$ has been defined, or, equivalently,

$$
\int_S(\boldsymbol{n}\cdot\boldsymbol{\varphi})\delta P\,\mathrm{d}s = -\int_S(\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\,\mathrm{d}s \tag{27}
$$

In view of Eqs. (14) and (15) it can be seen that if the adjoint variables satisfy the following boundary conditions

$$
\begin{gathered}
\boldsymbol{n}\cdot\boldsymbol{\varphi}|_S = \frac{\Delta C_p}{C_\infty} \qquad \text{(inverse design)} \\
\boldsymbol{n}\cdot\boldsymbol{\varphi}|_S = \frac{1}{C_\infty}(\boldsymbol{d}\cdot\boldsymbol{n}) \qquad \text{(force optimization)}
\end{gathered}
\tag{28}
$$

the complete variation of the cost functions for the inverse design and force optimization are

$$
\begin{aligned}
\delta\left(\frac{1}{2}\int_S (\Delta C_p)^2\,\mathrm{d}s\right) &= \int_S\left(\frac{\Delta C_p}{C_\infty}(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P) + K\frac{(\Delta C_p)^2}{2}\right)\mathrm{d}s - I_{\mathrm{eq}}\\
\delta\int_S C_p(\boldsymbol{n}\cdot\boldsymbol{d})\,\mathrm{d}s &= \int_S\left(\frac{1}{C_\infty}(\boldsymbol{n}\cdot\boldsymbol{d})(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P) + KC_p(\boldsymbol{d}\cdot\boldsymbol{n})\right.\\
&\quad \left. + C_p(\boldsymbol{d}\cdot\delta\boldsymbol{n})\right)\mathrm{d}s - I_{\mathrm{eq}}\\
\text{where } I_{\mathrm{eq}} &= \int_S (\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\,\mathrm{d}s\\
\boldsymbol{n}\cdot\delta\boldsymbol{v}|_S &= -[(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla})\boldsymbol{v}]\cdot\boldsymbol{n} - \delta\boldsymbol{n}\cdot\boldsymbol{v}
\end{aligned}
\tag{29}
$$

Notice that even though I_{eq} in Eq. (29) contains the variation of the velocity vector $\delta\boldsymbol{v}$, it is not necessary to solve the linearized flow equations (16) to actually compute its value. Indeed, I_{eq} only involves the normal component of $\delta\boldsymbol{v}$ on the wall, whose value in terms of geometric quantities and flowfield variables is given by the linearized boundary condition (17).

It is interesting to compare the results in Eq. (29) to previous works, such as, for example, Eqs. (27a)–(28b) in [7], and particularly to Eq. (27) for the variation of the inverse design cost function in Jameson–Kim's reduced gradient formulation [15], which, although presented in a seemingly different form, is exactly the same as in Eq. (29).

Finally, plugging in Eq. (29) the actual values of K, $\delta\boldsymbol{x}$, $\delta\boldsymbol{n}$ from Eqs. (11–13) and integrating by parts where appropriate, using $\partial_{\mathrm{tg}}\boldsymbol{t} = \kappa\boldsymbol{n}$, $\partial_{\mathrm{tg}}\boldsymbol{n} = -\kappa\boldsymbol{t}$, and $\int_S(\partial_{\mathrm{tg}}f(s))\,\mathrm{d}s = 0$, Eq. (29) can be cast in the form

$$
\begin{aligned}
\delta\left(\frac{1}{2}\int_S (\Delta C_p)^2\,\mathrm{d}s\right) &= \int_S\left(\frac{\Delta C_p}{C_\infty}\partial_n P - \kappa\frac{(\Delta C_p)^2}{2}\right)\alpha\,\mathrm{d}s - I_{\mathrm{eq}}\\
\delta\int_S C_p(\boldsymbol{n}\cdot\boldsymbol{d})\,\mathrm{d}s &= \int_S \frac{1}{C_\infty}(\boldsymbol{d}\cdot\boldsymbol{\nabla}P)\alpha\,\mathrm{d}s - I_{\mathrm{eq}}\\
\text{where } I_{\mathrm{eq}} &= -\int_S ((\boldsymbol{\nabla}\cdot\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\\
&\quad + (\boldsymbol{t}\cdot\boldsymbol{v})\partial_{\mathrm{tg}}(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4))\alpha\,\mathrm{d}s
\end{aligned}
\tag{30}
$$

where $\partial_n = \boldsymbol{n}\cdot\boldsymbol{\nabla}$ is the normal derivative to the surface S. As is clear from Eq. (30), the final expressions for the complete variations do not depend on the tangent component β of the deformation, as expected. Likewise, the (normal) deformation parameter has been isolated so that the variation can be written in the generic form

$$
\delta J = \int_S G\alpha\,\mathrm{d}s \tag{31}
$$

where, for example, the expression for the force optimization problems is

$$
\begin{aligned}
G &= \frac{1}{C_\infty}(\boldsymbol{d}\cdot\boldsymbol{\nabla}P) + (\boldsymbol{\nabla}\cdot\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\\
&\quad + (\boldsymbol{t}\cdot\boldsymbol{v})\partial_{\mathrm{tg}}(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)
\end{aligned}
\tag{32}
$$

Notice that G is essentially a local gradient. As such, it gives for each point on the surface the optimal deformation direction, that is to say, the size of the deformation in the normal direction which produces the largest variation in the cost function. This result opens the possibility to substitute the standard deformation functions (such as Hicks–Henne functions, Bézier polynomials, etc.) by G (with the appropriate modifications to account for possible geometric restrictions).

Repeating the computation for a more general pressure-dependent cost function such as

$$
\int_S g(P,\boldsymbol{n})\,\mathrm{d}s
$$

the variation is

$$
\begin{aligned}
&\delta\int_S g(P,\boldsymbol{n})\,\mathrm{d}s\\
&\quad = \int_S\left(\frac{\partial g}{\partial P}(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P) + \frac{\partial g}{\partial\boldsymbol{n}}\cdot\delta\boldsymbol{n} + Kg(P,\boldsymbol{n})\right)\mathrm{d}s - I_{\mathrm{eq}}
\end{aligned}
\tag{33}
$$

where I_{eq} is the same as in Eq. (29), and the following boundary conditions hold:

$$
\begin{aligned}
\boldsymbol{n}\cdot\boldsymbol{v}|_S &= 0\\
\boldsymbol{n}\cdot\delta\boldsymbol{v}|_S &= -[(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla})\boldsymbol{v}]\cdot\boldsymbol{n} - \delta\boldsymbol{n}\cdot\boldsymbol{v}\\
\boldsymbol{n}\cdot\boldsymbol{\varphi}|_S &= \frac{\partial g}{\partial P}
\end{aligned}
\tag{34}
$$

Proceeding with Eq. (33) as in the derivation of Eq. (30), the following expression is obtained:

$$
\begin{aligned}
\delta\int_S g(P,\boldsymbol{n})\,\mathrm{d}s &= \int_S\left[\frac{\partial g}{\partial P}\partial_n P + \left(\partial_{\mathrm{tg}}\frac{\partial g}{\partial\boldsymbol{n}}\right)\cdot\boldsymbol{t}\right.\\
&\quad \left. - \kappa\left(g - \frac{\partial g}{\partial\boldsymbol{n}}\cdot\boldsymbol{n}\right)\right]\alpha\,\mathrm{d}s - I_{\mathrm{eq}}
\end{aligned}
\tag{35}
$$

which gives

$$
\begin{aligned}
G &= \frac{\partial g}{\partial P}\partial_n P + \left(\partial_{\mathrm{tg}}\frac{\partial g}{\partial\boldsymbol{n}}\right)\cdot\boldsymbol{t} - \kappa\left(g - \frac{\partial g}{\partial\boldsymbol{n}}\cdot\boldsymbol{n}\right)\\
&\quad + (\boldsymbol{\nabla}\cdot\boldsymbol{v})(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)\\
&\quad + (\boldsymbol{t}\cdot\boldsymbol{v})\partial_{\mathrm{tg}}(\rho\psi_1 + \rho\boldsymbol{v}\cdot\boldsymbol{\varphi} + \rho H\psi_4)
\end{aligned}
\tag{36}
$$

in the general case.

In this section we have reviewed the formulation of the continuous adjoint approach for inviscid flows and we have applied it to the computation of sensitivities of cost functions which depend solely on the pressure. One could wonder as to the possibility of dealing with more general cost functions depending on flow variables other than the pressure. In this regard, it has been known for some time that cost functions which do not depend exclusively on the pressure do not lead a priori to well-posed *continuous, inviscid* adjoint systems [7–9]. However, it has been argued [10] that it is possible to lift the restriction on the allowed cost functions if the flow equations restricted to the boundary are explicitly taken into account.

Before moving on to consider viscous aerodynamic design, we will briefly comment on the issue of mesh sensitivities. Mesh sensitivities in the context of numerical approximations of PDE constitute a broad subject. In the present context it arises due to the contributions that the numerical mesh or grid introduces in the variations of the discrete functional that mimics the continuous functional to be optimized. In other words, mesh sensitivities reflect the effect of changing the grid, which is a contribution of purely numerical origin and, as such, appear naturally in the discrete adjoint approach, and also in the gradients obtained by finite differences.

The issue of mesh sensitivities in aeronautical optimal design has been previously treated in a number of articles. Jameson's original formulation for the continuous adjoint gradient [3] contained a field integral which incorporated the effect of the mesh variations throughout the domain, in the form of the variation of the mapping function. Numerical experiments conducted in [15] to assess the accuracy of the reduced gradient formulas for Euler flows showed that while the original adjoint gradients (which incorporate the effect of the mesh variation) are slightly better than the surface adjoint gradients (in terms of agreement with finite difference and complex-step gradients), the discrepancies decrease as the mesh is refined (in structured grids). On the other hand, it has been pointed out in [7] that mesh sensitivity terms are critical in obtaining accurate derivatives for geometries with singularities on unstructured grids. In particular, it is shown that although the effect of the grid sensitivities on the value of the gradients at generic points of the boundary decreases as the grid is refined, this is not the case in the vicinity of geometric singularities such as the trailing edge, where errors caused by ignoring the contribution of the mesh sensitivities do not vanish as the mesh is refined. Roughly speaking, one may expect mesh sensitivities to be more relevant for problems in which solutions are less regular and, accordingly, their numerical approximations more sensible to changing the mesh.

Mesh sensitivities have not been explicitly taken into account in this work, and no attempt has been made to assess their impact on the results presented here. The comparison of the gradients obtained with our continuous adjoint formulation and those obtained via finite differences show a very good agreement and this can be viewed as an indication of the correctness of the formulation. However, as mentioned above, geometries with singularities or solutions with shocks will probably require a more sophisticated optimization strategy involving a mesh sensitivity analysis (see [16] for a discussion of the impact of shock discontinuities on flow solutions). This can be done, for example, by including the particular methodology used to deform the mesh in the computation of the gradients (as in [3]).

In any case, any algorithm has to deal, in one way or another, with changing grids. In our numerical simulations this has been done in an automatic way by redefining the mesh in each new domain, after deformation, by simply using a spring analogy of the so-called matricial method [17]; see also Sec. V.C. Of course, more accurate simulation methods should include also the possibility of optimizing the mesh in each iteration of the optimization process, that is, after each deformation of the geometry. This has not been done so far and needs further work. But there is a solid theoretical background for developing such methods, based on adjoint techniques (see, for example [18] and references therein) which allow optimizing the mesh to guarantee a better approximation of the solution of the state equation. It is worth underlying that our approach allows incorporating such mesh sensitivity tools in a modular way so that, as shown in this paper, the method may also work by replacing it by a simpler and more systematic or automatic method for mesh adaptation. A complete analysis of the possible combination of the methods developed in this paper and mesh sensitivity techniques is still under development.

C. Aerodynamic Design with the Navier–Stokes Equations

The governing equations, for viscous laminar flows in two dimensions, are

$$\boldsymbol{\nabla} \cdot \boldsymbol{F} = \boldsymbol{\nabla} \cdot \boldsymbol{F}^v \quad \text{in } \Omega \tag{37}$$

where $\boldsymbol{F} = (F_x, F_y)$ has been defined in Eq. (5) and

$$F_x^v = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ u\sigma_{xx} + v\sigma_{xy} + k\frac{\partial T}{\partial x} \end{pmatrix}, \qquad F_y^v = \begin{pmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ u\sigma_{yx} + v\sigma_{yy} + k\frac{\partial T}{\partial y} \end{pmatrix} \tag{38}$$

The viscous stresses may be written as

$$\sigma_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \qquad \sigma_{xy} = \sigma_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$
$$\sigma_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \tag{39}$$

where μ is the laminar viscosity coefficient. The coefficient of thermal conductivity and the temperature are computed as

$$k = \frac{c_p}{Pr}\mu, \qquad T = \frac{P}{R\rho} \tag{40}$$

where c_p is the specific heat at constant pressure, Pr is the Prandtl number, and R is the gas constant. Turbulent flows can be incorporated by adding to μ the turbulent viscosity coefficient μ_t, whose value is computed by means of a suitable turbulence model, but this possibility will not be considered in this work. (The generalization of the continuous adjoint approach to include the Spalart–Allmaras turbulence model is currently under investigation.) Equation (37) is supplemented with characteristic-type boundary conditions on the far field, and nonslip conditions on solid walls

$$u = v = 0 \quad \text{on } S \tag{41}$$

An additional boundary condition has to be imposed on the temperature on solid walls, which can be either adiabatic or isothermal (constant temperature)

$$\begin{aligned} \partial_n T|_S = \boldsymbol{n} \cdot \boldsymbol{\nabla} T|_S = 0 \quad & \text{adiabatic} \\ T|_S = T_0 \quad & \text{constant temperature} \end{aligned} \tag{42}$$

The corresponding linearized flow equations are from Eqs. (37–42); see [8],

$$\boldsymbol{\nabla} \cdot ((\boldsymbol{A} + \boldsymbol{A}^v)^T M^{-1}\delta U) - \partial_x\left(D_{xx}^T M^{-1}\frac{\partial}{\partial x}\delta U + D_{xy}^T M^{-1}\frac{\partial}{\partial y}\delta U\right)$$
$$- \partial_y\left(D_{yx}^T M^{-1}\frac{\partial}{\partial x}\delta U + D_{yy}^T M^{-1}\frac{\partial}{\partial y}\delta U\right) = 0 \quad \text{in } \Omega \tag{43}$$

where the matrices are

$$\begin{aligned} \boldsymbol{A} &= \left(\frac{\partial \boldsymbol{F}}{\partial V}\right)^T, & \boldsymbol{A}^v &= -\left(\frac{\partial \boldsymbol{F}^v}{\partial V}\right)^T \\ D_{xx} &= \left(\frac{\partial F_x^v}{\partial(\partial_x V)}\right)^T, & D_{xy} &= \left(\frac{\partial F_x^v}{\partial(\partial_y V)}\right)^T \\ D_{yx} &= \left(\frac{\partial F_y^v}{\partial(\partial_x V)}\right)^T, & D_{yy} &= \left(\frac{\partial F_y^v}{\partial(\partial_y V)}\right)^T \end{aligned} \tag{44}$$

supplemented with the following boundary conditions:

$$\begin{aligned} & L^{-1}\delta U|_{\text{incoming characteristics}} = 0 \quad \text{on } \Gamma_\infty \\ & \delta u|_S = -\delta \boldsymbol{x} \cdot \boldsymbol{\nabla} u = -\alpha \partial_n u \\ & \delta v|_S = -\delta \boldsymbol{x} \cdot \boldsymbol{\nabla} v = -\alpha \partial_n v \\ & \boldsymbol{n} \cdot \boldsymbol{\nabla} \delta T|_S = -\delta \boldsymbol{n} \cdot \boldsymbol{\nabla} T - n_i \delta x_j \partial_j \partial_i T \qquad \text{adiabatic} \\ & \delta T|_S = -\delta \boldsymbol{x} \cdot \boldsymbol{\nabla} T \qquad \text{constant temperature} \end{aligned} \tag{45}$$

where the summation over repeated indices is understood, that is,

$$n_i \delta x_j \partial_j \partial_i T = n_x \delta x_x \partial_x \partial_x T + n_x \delta x_y \partial_y \partial_x T + n_y \delta x_x \partial_x \partial_y T$$
$$+ n_y \delta x_y \partial_y \partial_y T \tag{46}$$

The issue of defining relevant cost functions and determining their variations is considered next. The structure of the complete objective functions can be determined by examining the boundary terms which arise when the adjoint equation is derived from the linearized flow equations in the way discussed in the previous section.

As was done in the inviscid case, the linearized flow equations are multiplied by the vector of adjoint variables and integrated over the domain Ω. The resulting expression is then integrated by parts to produce a domain term and boundary terms supported on the solid wall and far-field boundaries. Vanishing of the domain term is tantamount to the adjoint flow equations

$$(M^{-1})^T(\boldsymbol{A} + \boldsymbol{A}^v) \cdot \boldsymbol{\nabla}\Psi^T + \frac{\partial}{\partial x}\left((M^{-1})^T\left[D_{xx}\frac{\partial \Psi^T}{\partial x} + D_{yx}\frac{\partial \Psi^T}{\partial y}\right]\right)$$
$$+ \frac{\partial}{\partial y}\left((M^{-1})^T\left[D_{xy}\frac{\partial \Psi^T}{\partial x} + D_{yy}\frac{\partial \Psi^T}{\partial y}\right]\right) = 0 \tag{47}$$

The terms supported on the far-field boundary can be eliminated with appropriate boundary conditions for the adjoint variables. Finally, the contribution at the solid wall boundary yields the following relation:

$$
\begin{aligned}
&\int_S (\boldsymbol{n}\cdot\delta\boldsymbol{v}(\rho\psi_1+\rho H\psi_4)-\psi_4\boldsymbol{n}\cdot\sigma\cdot\delta\boldsymbol{v}+\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v})\,\mathrm{d}s\\
&\quad+\int_S (k(\partial_n\psi_4)\delta T-\psi_4(\delta k)\partial_n T-k\psi_4\partial_n\delta T)\,\mathrm{d}s\\
&\quad+\int_S (\boldsymbol{n}\cdot\boldsymbol{\varphi}\delta P-\boldsymbol{n}\cdot\delta\sigma\cdot\boldsymbol{\varphi})\,\mathrm{d}s=0 \qquad (48)
\end{aligned}
$$

where nonslip boundary conditions have already been enforced on the velocity components, and

$$
\begin{aligned}
&\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v}=n_x\Sigma_{xx}\delta u+n_x\Sigma_{xy}\delta v+n_y\Sigma_{yx}\delta u+n_y\Sigma_{yy}\delta v\\
&\Sigma_{xx}=\frac{2}{3}\mu\left(2\frac{\partial\psi_2}{\partial x}-\frac{\partial\psi_3}{\partial y}\right),\qquad \Sigma_{xy}=\Sigma_{yx}=\mu\left(\frac{\partial\psi_2}{\partial y}+\frac{\partial\psi_3}{\partial x}\right)\\
&\Sigma_{yy}=\frac{2}{3}\mu\left(2\frac{\partial\psi_3}{\partial y}-\frac{\partial\psi_2}{\partial x}\right),\qquad \delta k=\frac{c_p}{Pr}\delta\mu \qquad (49)
\end{aligned}
$$

In view of Eqs. (45) and (48), the following combination of cost functions and adjoint boundary conditions are possible on adiabatic walls

$$
\begin{aligned}
&\int_S g(\boldsymbol{f},T)\,\mathrm{d}s,\qquad \boldsymbol{f}=P\boldsymbol{n}-\boldsymbol{n}\cdot\sigma\\
&(\psi_2,\psi_3)=\left(\frac{\partial g}{\partial f_x},\frac{\partial g}{\partial f_y}\right),\qquad k\partial_n\psi_4=\frac{\partial g}{\partial T}
\end{aligned}
\qquad (50)
$$

with the corresponding variation being

$$
\begin{aligned}
&\delta\int_S g(\boldsymbol{f},T)\,\mathrm{d}s=\int_{\delta S} g\,\mathrm{d}s-I_{\mathrm{eq}}\\
&\quad\text{where } I_{\mathrm{eq}}=\int_S((\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1+\rho H\psi_4)\\
&\quad-\psi_4\boldsymbol{n}\cdot\sigma\cdot\delta\boldsymbol{v}+\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v}-k\psi_4\partial_n\delta T)\,\mathrm{d}s\\
&\delta\boldsymbol{v}=-\alpha\partial_n\boldsymbol{v},\qquad \partial_n\delta T=-\delta\boldsymbol{n}\cdot\boldsymbol{\nabla}T-n_i\delta x_j\partial_j\partial_i T
\end{aligned}
\qquad (51)
$$

whereas on constant temperature walls one finds

$$
\begin{aligned}
&\int_S h(\boldsymbol{f},k\partial_n T)\,\mathrm{d}s,\qquad \boldsymbol{f}=P\boldsymbol{n}-\boldsymbol{n}\cdot\sigma\\
&(\psi_2,\psi_3)=\left(\frac{\partial h}{\partial f_x},\frac{\partial h}{\partial f_y}\right),\qquad \psi_4=-\frac{\partial h}{\partial(k\partial_n T)}
\end{aligned}
\qquad (52)
$$

with variation

$$
\begin{aligned}
&\delta\int_S h(\boldsymbol{f},k\partial_n T)\,\mathrm{d}s=\int_{\delta S} h\,\mathrm{d}s-I_{\mathrm{eq}}\\
&\quad\text{where } I_{\mathrm{eq}}=\int_S((\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1+\rho H\psi_4)-\psi_4\boldsymbol{n}\cdot\sigma\cdot\delta\boldsymbol{v}\\
&\quad+\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v}+k(\partial_n\psi_4)\delta T)\,\mathrm{d}s\\
&\delta\boldsymbol{v}=-\alpha\partial_n\boldsymbol{v},\qquad \delta T=-\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}T
\end{aligned}
\qquad (53)
$$

Therefore, the Navier–Stokes equations allow a priori optimization with respect to any of the components of the total force exerted by the fluid on the wall (including both the pressure and the viscous stress terms), as well as with respect to surface temperature distributions—for adiabatic boundary conditions—or surface heat flux—for constant temperature boundary conditions; see [8], where the same conclusions were obtained. It should be noticed, however, that from the above expressions it is clear that functions that depend solely on the pressure are allowed. This is an issue that has not been sufficiently clarified in the literature, so it is worthwhile to discuss it in some detail. It has been argued (see, for example, [7,9]) that in allowable cost functions for viscous flows, the pressure must be accompanied by viscous terms to obtain a consistent set of boundary conditions for the adjoint variables, even in those cases where it is possible to show that the viscous terms actually vanish. We will show below that, for steady compressible viscous flows, as well as for general, possibly unsteady, incompressible viscous flows, no viscous terms are required, not even formally, for *any* pressure-dependent cost function. (In [10] it is shown, with a different reasoning, that functions that depend on the pressure alone are allowed.)

The result follows from noticing that functions such as those in either Eq. (50) or Eq. (52) with the following structure:

$$
\int_S g(\boldsymbol{f}\cdot\boldsymbol{n})\,\mathrm{d}s \qquad (54)
$$

where the possible dependence on the temperature has been ignored, actually depend on the pressure alone, because

$$
\boldsymbol{f}\cdot\boldsymbol{n}|_S=P-\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n}|_S=P \qquad (55)
$$

where it has been used that for steady flows $(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n})|_S=0$. We will now offer a proof of this identity, which also establishes its range of validity. Let us then consider a general, unsteady, viscous flow. For such a flow, on a solid wall with no-slip boundary conditions the following relation holds:

$$
(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n})|_S=\frac{4}{3}\mu\boldsymbol{\nabla}\cdot\boldsymbol{v}|_S=-\frac{4}{3}\mu\frac{1}{\rho}\frac{\partial\rho}{\partial t} \qquad (56)
$$

(where the last equality follows from the restriction of the continuity equation to the wall). From Eq. (56) it follows that $(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n})|_S$ is zero for steady flows (compressible or incompressible), such as those considered in the present work, as well as for general, possibly time-dependent, *incompressible* flows.

Having established that cost functions of the form (54) depend only on the pressure, the procedure for obtaining their gradients and adjoint boundary conditions will be explained. The shape variation is as follows:

$$
\delta\int_S g(P)\,\mathrm{d}s=\int_S\left(\frac{\mathrm{d}g}{\mathrm{d}P}(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P)+Kg(P)\right)\mathrm{d}s+\int_S\left(\frac{\mathrm{d}g}{\mathrm{d}P}\delta P\right)\mathrm{d}s \qquad (57)
$$

The last term can be computed from Eq. (48) as

$$
\int_S\left(\frac{\mathrm{d}g}{\mathrm{d}P}\delta P\right)\mathrm{d}s=\int_S\frac{\mathrm{d}g}{\mathrm{d}P}(\boldsymbol{n}\cdot\delta\sigma\cdot\boldsymbol{n})\,\mathrm{d}s-I_{\mathrm{eq}} \qquad (58)
$$

provided that the adjoint variables satisfy the following boundary conditions:

$$
\boldsymbol{\varphi}|_S=\frac{\mathrm{d}g}{\mathrm{d}P}\boldsymbol{n} \qquad (59)
$$

The a priori unknown term in Eq. (58) containing the variation of the stress tensor can be obtained from the linearization of relation (56)

$$
\boldsymbol{n}\cdot\delta\sigma\cdot\boldsymbol{n}|_S+2\delta\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n}|_S+n_in_j\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}\sigma_{ij}=0 \qquad (60)
$$

from where it finally follows that

$$
\begin{aligned}
\delta\int_S g(P)\,\mathrm{d}s&=\int_S\left(\frac{\mathrm{d}g}{\mathrm{d}P}(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P)+Kg(P)\right)\mathrm{d}s\\
&-\int_S\frac{\mathrm{d}g}{\mathrm{d}P}(2\delta\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n}+n_in_j\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}\sigma_{ij})\,\mathrm{d}s-I_{\mathrm{eq}} \qquad (61)
\end{aligned}
$$

Equations (59–61) show that for steady, compressible viscous flows, as well as for general, possibly unsteady, incompressible viscous flows, arbitrary functions of the pressure alone allow one to obtain a consistent set of boundary conditions for the adjoint equations without the need to include viscous terms, not even formally. The key point in the argument is the explicit utilization of Eq. (60).

The result of Eq. (61) can be illustrated by the case of prescribed surface pressure. (In this case, it has been shown in [7] that for steady incompressible viscous flows, the specification of a pressure distribution is allowable at the cost of introducing additional viscous terms, which actually vanish, but that are nevertheless formally required for the derivation of the boundary conditions for the adjoint equations.) For such a case the cost function is customarily defined as [7,9]

$$J = \frac{1}{2}\int_S (\widetilde{\Delta C_p})^2\, \mathrm{d}s, \qquad \widetilde{\Delta C_p} = \Delta C_p - \frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n}) \tag{62}$$

which is actually

$$J = \frac{1}{2}\int_S (\Delta C_p)^2\, \mathrm{d}s$$

once Eq. (56) is taken into account. According to Eq. (61), this function has a variation

$$\begin{aligned}\delta J = \int_S \bigg(& \frac{\Delta C_p}{C_\infty}\delta\boldsymbol{x}\cdot(\boldsymbol{\nabla} P - n_j n_i \boldsymbol{\nabla}\sigma_{ij}) + \frac{1}{2}K(\Delta C_p)^2 \\ & - 2\frac{\Delta C_p}{C_\infty}(\delta\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{n})\bigg)\, \mathrm{d}s - I_{\mathrm{eq}}\end{aligned} \tag{63}$$

and requires the following adjoint boundary conditions:

$$\begin{gathered}(\psi_2,\psi_3)|_S = \frac{\Delta C_p}{C_\infty}(n_x, n_y) \\ \partial_n\psi_4|_S = 0 \quad \text{(adiabatic)} \\ \psi_4|_S = 0 \quad \text{(constant temperature)}\end{gathered} \tag{64}$$

To end this section, the cost functions for force optimization problems will be spelled out in detail. The function for total (i.e., including viscous effects) force optimization is

$$\begin{gathered}J = \int_S (\boldsymbol{f}^*\cdot\boldsymbol{d})\, \mathrm{d}s, \qquad \boldsymbol{f}^* = C_p\boldsymbol{n} - \frac{1}{C_\infty}\boldsymbol{n}\cdot\sigma \\ \boldsymbol{d} = \begin{cases}(\cos\alpha, \sin\alpha) & \text{(drag)} \\ (-\sin\alpha, \cos\alpha) & \text{(lift)}\end{cases}\end{gathered} \tag{65}$$

which has a variation

$$\begin{aligned}\delta J = \int_S \bigg(& \frac{1}{C_\infty}((\delta\boldsymbol{x}\cdot\boldsymbol{\nabla} P)(\boldsymbol{n}\cdot\boldsymbol{d}) - n_i d_j \delta\boldsymbol{x}\cdot\boldsymbol{\nabla}\sigma_{ij}) + K(\boldsymbol{f}^*\cdot\boldsymbol{d}) \\ & + \delta\boldsymbol{n}\cdot\left(C_p\boldsymbol{d} - \frac{1}{C_\infty}\sigma\cdot\boldsymbol{d}\right)\bigg)\, \mathrm{d}s - I_{\mathrm{eq}}\end{aligned} \tag{66}$$

and requires the adjoint boundary conditions

$$\begin{gathered}(\psi_2,\psi_3)|_S = \frac{1}{C_\infty}(d_x, d_y), \qquad \partial_n\psi_4|_S = 0 \quad \text{(adiabatic)} \\ \psi_4|_S = 0 \quad \text{(constant temperature)}\end{gathered} \tag{67}$$

From the above discussion it is clear that it is also possible to consider optimization problems involving the pressure force alone. The resulting formulas follow from Eqs. (59) and (61).

In this section, a viscous continuous adjoint method for evaluating sensitivity derivatives has been described, which can be seen as an extension to the case of compressible flows of the methodology introduced in [7]. However, as was pointed out in [7] and can be seen from the above results, the evaluation of sensitivity derivatives for viscous flows requires the computation of second-order derivatives of the flow variables. [See, for example, Eq. (51), which involves the Hessian of the temperature through the linearization of the adiabatic boundary condition, and Eq. (66), which involves the gradient of the stress tensor.] But this is, in fact, one of the major drawbacks to using the continuous adjoint for unstructured meshes, because the accurate numerical evaluation of such derivatives would require in general a spatially third-order accurate scheme [second derivatives computed from second-order accurate numerical data are generically inconsistent as both the second derivatives and their truncation error are generically of the same order $\mathcal{O}(h^0)$ in terms of the grid spacing h], which is actually beyond the capabilities of most unstructured flow solvers. A natural solution seems to be to increase the accuracy of the numerical scheme, which not only represents a significant level of effort, but is actually out of reach for those users which work with commercial solvers as flow-analysis tools. Another possible solution is to use the mapping techniques advocated by Jameson [9], but those are more suited to structured grids, and their implementation in viscous unstructured solvers is to some extent unnatural and computationally costly; recall that a surface formulation of the viscous adjoint gradients employing mapping techniques is not available, and that the evaluation of field integrals involving mesh movement terms for general unstructured grids can incur significant computational costs [15].

Using different, alternative ways to compute second derivatives does not provide a generic solution either. While it would be possible, though improbable, that for a particular second-order accurate numerical scheme of the flow solver, with a particular grid and a particular way of computing the second derivative, the leading $\mathcal{O}(h^0)$ term of the truncation error of the second derivatives (and hence of the gradient of the cost function) vanishes, this is definitely not a generic situation. The aim of this work is precisely to provide a systematic solution which, without the need for further numerical developments in the flow solver, yields a consistent expression for the gradients within the continuous adjoint approach.

Some of these considerations were already made in [7], where the proposed solution was to abandon the purely continuous adjoint approach in favor of a hybrid discrete approach.

It turns out that there exists a simpler alternative which allows maintaining the continuous adjoint framework while avoiding the cost of increasing the accuracy of the entire flowfield. The basic idea, which will be shown in the next section, is that it is possible to reduce the order of derivatives by using the restriction to the boundary of the (steady-state) flow equations

$$\begin{gathered}\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0, \qquad \boldsymbol{\nabla}\cdot\sigma = \boldsymbol{\nabla} P, \qquad \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla} T) = -\sigma{:}\nabla v \\ \text{where } \sigma{:}\nabla v \doteq \sigma_{ij}\partial_i v_j = \sigma_{xx}\partial_x u + \sigma_{xy}\partial_x v + \sigma_{yx}\partial_y u + \sigma_{yy}\partial_y v\end{gathered} \tag{68}$$

In this way, the resulting gradients can be written in terms of first derivatives of the flow and adjoint variables only. But first derivatives computed from second-order accurate data with a first-order accurate method are generically first-order accurate and therefore consistent, which makes the technique suitable for application with most flow solvers.

III. Reduction of the Higher Derivative Terms

The first term to be considered, which involves second derivatives of the temperature, appears in I_{eq} on solid walls with adiabatic boundary conditions [see Eq. (51)] and thus affects the computation of every cost function on such walls. It has the form

$$I_{\mathrm{eq}} = \cdots - \int_S k\psi_4(\partial_n \delta T)\, \mathrm{d}s = \cdots + \int_S k\psi_4 n_i \delta x_j (\partial_j\partial_i T)\, \mathrm{d}s \tag{69}$$

The Hessian operator $\partial_i\partial_j$ on S can be expressed in terms of tangent and normal derivatives. Tangent derivatives pose no problem as they can be readily integrated by parts, thereby reducing the number of derivatives. Normal derivatives, on the other hand, cannot be integrated by parts along S, but can be converted into tangent derivatives by using the flow equations and boundary conditions on S. The idea is as follows. Taking into account Eq. (12), Eq. (69) can be recast in the form

$$\begin{aligned}\int_S k\psi_4 n_i\delta x_j(\partial_i\partial_j T)\,\mathrm{d}s &= \int_S k\psi_4\alpha n_i n_j(\partial_i\partial_j T)\,\mathrm{d}s\\ &+ \int_S k\psi_4\beta n_i t_j(\partial_i\partial_j T)\,\mathrm{d}s = \int_S \alpha k\psi_4\left(\partial_n^2 T\right)\mathrm{d}s\\ &+ \int_S k\psi_4\beta(\partial_{\mathrm{tg}}\partial_n T)\,\mathrm{d}s + \int_S k\psi_4\beta\kappa(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\\ &= \int_S \alpha k\psi_4\left(\partial_n^2 T\right)\mathrm{d}s + \int_S k\psi_4\beta\kappa(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\end{aligned} \tag{70}$$

where the following identities have been used:

$$\begin{gathered}\partial_{\mathrm{tg}} \doteq \boldsymbol{t}\cdot\boldsymbol{\nabla} = t_i\partial_i, \qquad \partial_n \doteq \boldsymbol{n}\cdot\boldsymbol{\nabla} = n_i\partial_i\\ n_i n_j\partial_i\partial_j = \partial_n^2, \qquad t_i t_j\partial_i\partial_j = \partial_{\mathrm{tg}}^2 - \kappa\partial_n\\ n_i t_j\partial_i\partial_j = \partial_n\partial_{\mathrm{tg}} = \partial_{\mathrm{tg}}\partial_n + \kappa\partial_{\mathrm{tg}}\end{gathered} \tag{71}$$

As for the term containing two normal derivatives, it can be rewritten as follows:

$$k\partial_n^2 T|_S = \boldsymbol{\nabla}(k\boldsymbol{\nabla}T)|_S - \partial_{\mathrm{tg}}(k\partial_{\mathrm{tg}}T) = -\sigma{:}\nabla v - \partial_{\mathrm{tg}}(k\partial_{\mathrm{tg}}T) \tag{72}$$

where use has been made of the energy equation on the boundary [see Eq. (68)] as well as of the Neumann boundary condition for the temperature. Introducing Eq. (72) into Eq. (70) the following expression results:

$$\begin{aligned}\int_S k\psi_4 n_i\delta x_j(\partial_i\partial_j T)\,\mathrm{d}s &= -\int_S \alpha\psi_4(\sigma{:}\nabla v)\,\mathrm{d}s\\ &- \int_S \alpha\psi_4\partial_{\mathrm{tg}}(k\partial_{\mathrm{tg}}T)\,\mathrm{d}s + \int_S k\psi_4\beta\kappa(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\end{aligned} \tag{73}$$

There still remains a term with two tangent derivatives, which can be tackled by integration by parts

$$\begin{aligned}\int_S k\psi_4 n_i\delta x_j(\partial_i\partial_j T)\,\mathrm{d}s &= -\int_S \alpha\psi_4(\sigma{:}\nabla v)\,\mathrm{d}s\\ &+ \int_S k\partial_{\mathrm{tg}}(\alpha\psi_4)(\partial_{\mathrm{tg}}T)\,\mathrm{d}s + \int_S k\psi_4\beta\kappa(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\end{aligned} \tag{74}$$

Hence, the final expression results for I_{eq} on adiabatic solid walls

$$\begin{aligned}I_{\mathrm{eq}} &= \int_S(\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1 + \rho H\psi_4)\,\mathrm{d}s + \int_S(\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v})\,\mathrm{d}s\\ &- \int_S \psi_4(\boldsymbol{n}\cdot\sigma\cdot\delta\boldsymbol{v})\,\mathrm{d}s + \int_S k\psi_4(\delta\boldsymbol{n}\cdot\boldsymbol{\nabla}T)\,\mathrm{d}s\\ &- \int_S \alpha\psi_4(\sigma{:}\nabla v)\,\mathrm{d}s + \int_S k\partial_{\mathrm{tg}}(\alpha\psi_4)(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\\ &+ \int_S k\psi_4\beta\kappa(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\end{aligned} \tag{75}$$

Equation (75) can be further reduced by noting that, from Eq. (13),

$$\delta\boldsymbol{n} = -(\beta\kappa + \partial_{\mathrm{tg}}\alpha)\boldsymbol{t} \tag{76}$$

which, when substituted into Eq. (75) yields

$$\begin{aligned}I_{\mathrm{eq}} &= \int_S(\boldsymbol{n}\cdot\delta\boldsymbol{v})(\rho\psi_1 + \rho H\psi_4)\,\mathrm{d}s + \int_S(\boldsymbol{n}\cdot\Sigma\cdot\delta\boldsymbol{v})\,\mathrm{d}s\\ &- \int_S \psi_4(\boldsymbol{n}\cdot\sigma\cdot\delta\boldsymbol{v})\,\mathrm{d}s - \int_S \alpha\psi_4(\sigma{:}\nabla v)\,\mathrm{d}s\\ &+ \int_S k\alpha(\partial_{\mathrm{tg}}\psi_4)(\partial_{\mathrm{tg}}T)\,\mathrm{d}s\end{aligned} \tag{77}$$

Taking into account that $\delta\boldsymbol{v} = -\alpha\partial_n\boldsymbol{v}$, Eq. (77) can be written as

$$\begin{aligned}I_{\mathrm{eq}} &= \int_S(-(\boldsymbol{n}\cdot\partial_n\boldsymbol{v})(\rho\psi_1 + \rho H\psi_4) - \boldsymbol{n}\cdot\Sigma\cdot\partial_n\boldsymbol{v}\\ &+ \psi_4(\boldsymbol{n}\cdot\sigma\cdot\partial_n\boldsymbol{v}) - \psi_4(\sigma{:}\nabla v) + k(\partial_{\mathrm{tg}}\psi_4)(\partial_{\mathrm{tg}}T))\alpha\,\mathrm{d}s\end{aligned} \tag{78}$$

Additional terms containing second derivatives of the velocity field appear for cost functions involving the stress tensor; see Eqs. (63) and (66). For force optimization problems the disturbing term is of the form

$$\begin{aligned}-\int_S n_i d_j(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}(\sigma_{ij}))\,\mathrm{d}s &= -\int_S n_i d_j\delta x_k\partial_k(\sigma_{ij})\,\mathrm{d}s\\ &= -\int_S \alpha d_j n_i n_k(\partial_k\sigma_{ij})\,\mathrm{d}s - \int_S \beta d_j n_i t_k(\partial_k\sigma_{ij})\,\mathrm{d}s\\ &= -\int_S \alpha d_j n_i(\partial_n\sigma_{ij})\,\mathrm{d}s - \int_S \beta d_j n_i(\partial_{\mathrm{tg}}\sigma_{ij})\,\mathrm{d}s\end{aligned} \tag{79}$$

As before, the trick is to convert the normal derivatives to tangent derivatives by resorting to the flow equations. It follows from the identity

$$n_i\partial_n\sigma_{ij} \equiv \boldsymbol{n}\cdot\partial_n\sigma = \boldsymbol{\nabla}\cdot\sigma - \boldsymbol{t}\cdot\partial_{\mathrm{tg}}\sigma \tag{80}$$

and the momentum equation, $\boldsymbol{\nabla}\cdot\sigma|_S = \boldsymbol{\nabla}P$, that Eq. (79) can be cast in the form

$$\begin{aligned}-\int_S n_i d_j(\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}(\sigma_{ij}))\,\mathrm{d}s &= -\int_S \alpha d_j(\partial_i\sigma_{ij})\,\mathrm{d}s\\ &+ \int_S \alpha t_i d_j(\partial_{\mathrm{tg}}\sigma_{ij})\,\mathrm{d}s - \int_S \beta n_i d_j(\partial_{\mathrm{tg}}\sigma_{ij})\,\mathrm{d}s\\ &= -\int_S \alpha(\boldsymbol{d}\cdot\boldsymbol{\nabla}P)\,\mathrm{d}s - \int_S \delta x_i^{\perp} d_j(\partial_{\mathrm{tg}}\sigma_{ij})\,\mathrm{d}s\\ &= -\int_S(\delta\boldsymbol{x}\cdot\boldsymbol{n})(\boldsymbol{d}\cdot\boldsymbol{\nabla}P)\,\mathrm{d}s + \int_S \boldsymbol{d}\cdot\sigma\cdot(\partial_{\mathrm{tg}}\delta\boldsymbol{x}^{\perp})\,\mathrm{d}s\end{aligned} \tag{81}$$

where

$$\delta\boldsymbol{x}^{\perp} = \beta\boldsymbol{n} - \alpha\boldsymbol{t} \tag{82}$$

Therefore, from Eqs. (66) and (81) it follows that

$$\begin{aligned}&\delta\int_S\left(C_p(\boldsymbol{n}\cdot\boldsymbol{d}) - \frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{d})\right)\mathrm{d}s\\ &= \frac{1}{C_\infty}\int_S((\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P)(\boldsymbol{n}\cdot\boldsymbol{d}) - n_i d_j\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}\sigma_{ij})\,\mathrm{d}s\\ &+ \int_S\left(K\left(C_p(\boldsymbol{n}\cdot\boldsymbol{d}) - \frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{d})\right)\right.\\ &\left.+ \delta\boldsymbol{n}\cdot\left(C_p\boldsymbol{d} - \frac{1}{C_\infty}\sigma\cdot\boldsymbol{d}\right)\right)\mathrm{d}s - I_{\mathrm{eq}}\\ &= \frac{1}{C_\infty}\int_S((\delta\boldsymbol{x}\cdot\boldsymbol{\nabla}P)(\boldsymbol{n}\cdot\boldsymbol{d}) - (\delta\boldsymbol{x}\cdot\boldsymbol{n})(\boldsymbol{d}\cdot\boldsymbol{\nabla}P))\,\mathrm{d}s\\ &+ \int_S(C_p\boldsymbol{d}\cdot(\delta\boldsymbol{n} + \mathrm{K}\boldsymbol{n}))ds\\ &+ \frac{1}{C_\infty}\int_S(\boldsymbol{d}\cdot\sigma\cdot(\partial_{\mathrm{tg}}\delta\boldsymbol{x}^{\perp} - K\boldsymbol{n} - \delta\boldsymbol{n}))\,\mathrm{d}s - I_{\mathrm{eq}}\end{aligned} \tag{83}$$

where like terms have been grouped for later convenience. It is possible to further reduce Eq. (83) by using Eqs. (11–13) and (82), which can be combined in the identity

$$K\boldsymbol{n} + \delta\boldsymbol{n} = (\partial_{\mathrm{tg}}\beta - \alpha\kappa)\boldsymbol{n} - (\beta\kappa + \partial_{\mathrm{tg}}\alpha)\boldsymbol{t} = \partial_{\mathrm{tg}}\delta\boldsymbol{x}^{\perp} \tag{84}$$

Likewise,

$$\frac{1}{C_\infty}((\delta\boldsymbol{x}\cdot\nabla P)(\boldsymbol{n}\cdot\boldsymbol{d})-(\delta\boldsymbol{x}\cdot\boldsymbol{n})(\boldsymbol{d}\cdot\nabla P))$$

$$=\frac{1}{C_\infty}(\boldsymbol{d}\cdot\delta\boldsymbol{x}^{\perp})\partial_{\mathrm{tg}}P=(\boldsymbol{d}\cdot\delta\boldsymbol{x}^{\perp})\partial_{\mathrm{tg}}C_p \tag{85}$$

Together, Eqs. (83–85) yield finally

$$\delta\int_S\left(C_p(\boldsymbol{n}\cdot\boldsymbol{d})-\frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{d})\right)\mathrm{d}s$$

$$=\int_S((\partial_{\mathrm{tg}}C_p)(\boldsymbol{d}\cdot\delta\boldsymbol{x}^{\perp}))\,\mathrm{d}s+\int_S(C_p\boldsymbol{d}\cdot(\partial_{\mathrm{tg}}\delta\boldsymbol{x}^{\perp}))\,\mathrm{d}s-I_{\mathrm{eq}}$$

$$=\int_S\partial_{\mathrm{tg}}(C_p\boldsymbol{d}\cdot\delta\boldsymbol{x}^{\perp})\,\mathrm{d}s-I_{\mathrm{eq}}=-I_{\mathrm{eq}} \tag{86}$$

This result leaves for the local gradient G in this case the final expression [see Eq. (78)]:

$$\delta\int_S\left(C_p(\boldsymbol{n}\cdot\boldsymbol{d})-\frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{d})\right)\mathrm{d}s=\int_S G\alpha\,\mathrm{d}s$$

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

$$+\psi_4(\sigma{:}\nabla v)-k(\partial_{\mathrm{tg}}\psi_4)(\partial_{\mathrm{tg}}T) \tag{87}$$

As for inverse design problems, the reduction of the higher derivative terms makes it possible to rewrite the expression (63) as

$$\delta\frac{1}{2}\int_S(\Delta C_p)^2\,\mathrm{d}s=\int_S\left[-\frac{1}{2}\alpha\kappa(\Delta C_p)^2+\frac{\Delta C_p}{C_\infty}(\boldsymbol{t}\cdot\sigma\cdot\boldsymbol{n})\partial_{\mathrm{tg}}\alpha\right.$$

$$\left.-\frac{\alpha}{C_\infty}(\boldsymbol{t}\cdot\sigma\cdot\boldsymbol{n})\partial_{\mathrm{tg}}\Delta C_p\right]\mathrm{d}s-I_{\mathrm{eq}} \tag{88}$$

IV. Extension to 3-D

The previous results can be readily extended to the case of 3-D flows. The governing equations are a direct generalization of Eq. (37)

$$\frac{\partial F_i}{\partial x_i}=\frac{\partial F_i^v}{\partial x_i}\quad\text{in }\Omega \tag{89}$$

with

$$U=\begin{pmatrix}\rho\\ \rho v_1\\ \rho v_2\\ \rho v_3\\ \rho E\end{pmatrix},\qquad F_i=\begin{pmatrix}\rho v_i\\ \rho v_i v_1+P\delta_{i1}\\ \rho v_i v_2+P\delta_{i2}\\ \rho v_i v_3+P\delta_{i3}\\ \rho v_i H\end{pmatrix}$$

$$F_i^v=\begin{pmatrix}0\\ \sigma_{ij}\delta_{j1}\\ \sigma_{ij}\delta_{j2}\\ \sigma_{ij}\delta_{j3}\\ v_j\sigma_{ij}+k\frac{\partial T}{\partial x_i}\end{pmatrix} \tag{90}$$

where v_1, v_2, and v_3 are the Cartesian velocity components, δ_{ij} is the Kronecker delta function, and

$$\sigma_{ij}=\mu\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}-\frac{2}{3}\delta_{ij}\frac{\partial v_k}{\partial x_k}\right) \tag{91}$$

are the viscous stresses. In what follows, Latin indices from the middle of the alphabet $i, j, k, \ldots, =1, 2, 3$ will denote 3-D Cartesian coordinates, $x_i=(x, y, z)$. The repeated index i implies summation over $i=1$–3.

The solid walls will be represented in 3-D by a closed surface S (or union thereof) described by a parameterization $\boldsymbol{x}(\xi,\eta)=(x(\xi,\eta), y(\xi,\eta), z(\xi,\eta))$ with parameters (ξ,η) which we shall refer collectively to as ξ^a, with Latin indices from the beginning of the alphabet $a, b, c, \ldots, =1, 2$ denoting parametric directions on the surface. Let $\boldsymbol{e}_a$ denote the tangent vectors to the surface corresponding to the given (ξ,η) parameterization and assume that the parameterization is picked such that $\boldsymbol{n}=(\boldsymbol{e}_1\times\boldsymbol{e}_2)/|\boldsymbol{e}_1\times\boldsymbol{e}_2|$ is the exterior unit normal to the surface. The surface integration measure is now $\mathrm{d}s=|\boldsymbol{e}_1\times\boldsymbol{e}_2|\mathrm{d}\xi\,\mathrm{d}\eta$. A generic deformation of the boundary can be described as follows:

$$\delta\boldsymbol{x}(\xi,\eta)=\alpha(\xi,\eta)\boldsymbol{n}+\beta^a(\xi,\eta)\boldsymbol{e}_a \tag{92}$$

For sufficiently small values of the deformation, the following holds:

$$\delta\boldsymbol{e}_a=(\partial_a\alpha+\beta^b L_{ab})\boldsymbol{n}+\left(\partial_a\beta^b+\beta^c\Gamma^b_{ca}-\alpha L_{ac}g^{cb}\right)\boldsymbol{e}_b$$

$$\delta\boldsymbol{n}=-g^{ab}(\partial_a\alpha+\beta^c L_{ac})\boldsymbol{e}_b$$

$$\delta\,\mathrm{d}s=\left(\partial_a\beta^a+\beta^c\Gamma^a_{ca}-\alpha L_{ab}g^{ab}\right)\mathrm{d}s=(\boldsymbol{\nabla}_{\mathrm{tg}}\cdot\boldsymbol{\beta}-2H_m\alpha)\,\mathrm{d}s \tag{93}$$

where g^{ab} is the inverse metric tensor, L_{ab} is the second fundamental form, Γ^c_{ab} are the Christoffel symbols, and $H_m=L_{ab}g^{ab}/2$ is the mean curvature of the surface; hence, in passing from 2-D to 3-D the replacement

$$K_{2\text{-D}}=\partial_{\mathrm{tg}}\beta-\kappa\alpha\to K_{3\text{-D}}=\boldsymbol{\nabla}_{\mathrm{tg}}\cdot\boldsymbol{\beta}-2H_m\alpha$$

needs to be done in the curvature term of the geometric part of the variation described in Eq. (3).

As in 2-D, the analysis of the variation of the usual objective functions (lift/drag, inverse design) unveils terms containing second derivatives of flow variables. Proceeding as above, these terms can be reduced, resulting in

$$\delta\int_S\left(C_p(\boldsymbol{n}\cdot\boldsymbol{d})-\frac{1}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{d})\right)\mathrm{d}s=-I_{\mathrm{eq}}=\int_S G\alpha\,\mathrm{d}s$$

$$G=(\boldsymbol{n}\cdot\partial_n\boldsymbol{v})(\rho\psi_1+\rho H\psi_5)+\boldsymbol{n}\cdot\Sigma\cdot\partial_n\boldsymbol{v}-\psi_5(\boldsymbol{n}\cdot\sigma\cdot\partial_n\boldsymbol{v})$$

$$+\psi_5(\sigma_{ij}\partial_i v_j)-k(\boldsymbol{\nabla}_{\mathrm{tg}}\psi_5)\cdot(\boldsymbol{\nabla}_{\mathrm{tg}}T) \tag{94}$$

for lift/drag optimization problems. Also, for prescribed surface pressure the following expression results:

$$\delta\frac{1}{2}\int_S(\Delta C_p)^2\,\mathrm{d}s=\int_S\left[-\alpha H_m(\Delta C_p)^2+\frac{\Delta C_p}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{\nabla}_{\mathrm{tg}}\alpha)\right.$$

$$\left.-\frac{\alpha}{C_\infty}(\boldsymbol{n}\cdot\sigma\cdot\boldsymbol{\nabla}_{\mathrm{tg}}\Delta C_p)\right]\mathrm{d}s-I_{\mathrm{eq}} \tag{95}$$

V. Brief Description of the Numerical Implementation and Summary of Results

All the final formulas of this paper have been implemented in the 2-D cell-vertex finite-volume code NENS (no estructurado Navier–Stokes) (developed by INTA) that solves the Navier–Stokes equations on unstructured meshes using an edge-based data structure. To simplify the implementation of the adjoint equations, most of the original subroutines of the direct (flow) solver have been used (edge-based data structure, time integration scheme, multigrid scheme, etc).

The analytical expressions developed in this paper have been tested and some results will be shown to demonstrate the quality of the gradients calculated using this approach both in Euler and laminar Navier–Stokes problems. Also, a full optimization problem is exhibited with the purpose of demonstrating the potentiality of the developed software.

A. Numerical Implementation

1. *Spatial Discretization*

A finite-volume discretization is used to solve both the direct and adjoint equations. The finite-volume discretization is obtained by applying the integral formulation of the governing equations to a control volume consisting of a cell of the median-dual mesh surrounding each node. A time-marching strategy is used to obtain the steady solution [19,20]. For the flow equations, a central scheme with Jameson–Schmidt–Turkel (JST)-type scalar artificial dissipation [14,19] or a Roe-type upwind scheme with linear reconstruction [21] is used for the discretization of the convective flux, while viscous fluxes are computed with a node-gradient-based approach due to Weiss et al. [19,22], resulting in second-order spatial accuracy. For the adjoint equation

$$\frac{\partial}{\partial t}\int_{c_i}\Psi\,\mathrm{d}\Omega-\int_{\partial c_i}\boldsymbol{F}\boldsymbol{n}\,\mathrm{d}l+\left[\int_{\partial c_i}\boldsymbol{F}^v\boldsymbol{n}\,\mathrm{d}l+\int_{c_i}Q^v\,\mathrm{d}\Omega\right]=0$$
$$\rightarrow\frac{\partial}{\partial t}(\Psi_i\Omega_i)=\sum_{k=1}^{\text{neighbor}}f_{ik}S_{ik}-\left[\sum_{k=1}^{\text{neighbor}}f_{ik}^vS_{ik}+Q_i^vV_i\right]\tag{96}$$

where Ψ is the vector of adjoint variables, $\boldsymbol{F}$ is the adjoint convective flux, $\boldsymbol{F}^v$ is the adjoint viscous flux, Q^v is an adjoint viscous source term, and the surfaces S_{ik} (edges in 2-D) form the control volume of the dual mesh for the node i. The scheme for the adjoint convective flux is based on a central discretization with dissipation terms of either artificial dissipation type or upwind flux. The artificial dissipation between two nodes 0 and 1 can be expressed as follows:

$$d_{01}=\varepsilon(\nabla^2\Psi_0-\nabla^2\Psi_1)\varphi_{01}\lambda_{01}$$
$$\text{where }\nabla^2\Psi_0=\sum_{k=1}^{\text{neighbor}}(\Psi_k-\Psi_0)=-n_0\Psi_0+\sum_{k=1}^{\text{neighbor}}(\Psi_k)$$
$$\lambda_{01}=(|\boldsymbol{v}_{01}\cdot\boldsymbol{n}_{01}|+c_{01})S_{01},\qquad \boldsymbol{v}_{01}=\frac{\boldsymbol{v}_0+\boldsymbol{v}_1}{2}$$
$$c_{01}=\frac{c_0+c_1}{2}\tag{97}$$

where ε is an adjustable parameter, λ_{01} is the local spectral radius, ∇^2 denotes the undivided Laplacian operator, $\boldsymbol{v}_{01}$ and c_{01} denote the fluid speed and sound speed at the cell face, $\boldsymbol{n}_{01}$ denotes the unit vector normal to the face of the control volume associated to the edge connecting nodes (0,1), and S_{01} is the size of the said face.

If an upwind scheme is used, the system of adjoint equations can be expressed (in one dimension) as

$$\frac{\partial\Psi}{\partial t}-\frac{\partial F}{\partial x}=0\rightarrow\frac{\partial\Psi}{\partial t}-\frac{\partial F}{\partial\Psi}\frac{\partial\Psi}{\partial x}=\frac{\partial\Psi}{\partial t}-A^T\frac{\partial\Psi}{\partial x}=0\tag{98}$$

where A^T is the transpose of the Jacobian inviscid matrix flux. Using a 1-D finite-volume method, the discretization around the node i has the following form:

$$\Psi_i^{n+1}=\Psi_i^n-\frac{\Delta t}{\Delta x}[F_{i+1/2}^n(\Psi)-F_{i-1/2}^n(\Psi)]$$
$$F_{i+1/2}=F_i^++F_{i+1}^-=1/2A^T(\Psi_{i+1}-\Psi_i)-1/2|A^T|(\Psi_{i+1}-\Psi_i)$$
$$F_{i-1/2}=F_{i-1}^++F_i^-=1/2A^T(\Psi_i-\Psi_{i-1})-1/2|A^T|(\Psi_i-\Psi_{i-1})\tag{99}$$

2. Steady-State Time Integration

To speed up the rate of convergence an overset multigrid scheme is used in which the Jacobian matrices are linearly interpolated between different mesh levels. Time integration is tackled with an explicit q-stage Runge–Kutta scheme.

3. Boundary Conditions

Boundary conditions for a solid wall can be imposed in two ways: 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 boundary characteristic boundary conditions are used.

B. Design Variables

Concerning the design variables, different possibilities have been tested: deformation or bump functions such as Wagner polynomials, Hicks–Vanderplaats functions, Legendre polynomials, Hicks–-Henne functions, Bézier polynomials, nonuniform rational B spline (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 [23] have been used:

$$\Delta y=\sum_{k=1}^{N}\delta_kf_k(x)$$
$$f_k(x)=\sin^3(\pi x^{e_k}),\quad\text{where }e_k=\frac{\log(0.5)}{\log(x_k)}\tag{100}$$

where N is the number of bump functions and δ_k is the design variable step. Each shape function has its maximum at the point x_k and these functions are separately applied to the upper and lower surfaces.

C. Mesh Deformation

A matricial method is used to move 2-D meshes. The matricial method is based on the definition of a stiffness matrix that connects the two ends of a single bar (mesh edge). All the quantities must be stored using a sparse method and a conjugate gradient algorithm is used to solve the linear system.

$$\begin{bmatrix}F_{X_i}\\F_{Y_i}\\M_{\theta_i}\\F_{X_j}\\F_{Y_j}\\M_{\theta_j}\end{bmatrix}=[R]^T\cdot[k_L]\cdot[R]\cdot\begin{bmatrix}X_i\\Y_i\\\theta_i\\X_j\\Y_j\\\theta_j\end{bmatrix}\tag{101}$$

where $[R]$ is the rotational matrix, and F and M are forces and torques, respectively. Finally the stiffness matrix has the form

$$[k_L]=\begin{bmatrix}\frac{AE}{L}&0&0&\frac{-AE}{L}&0&0\\0&\frac{12EI}{L^3}&\frac{6EI}{L^2}&0&\frac{-12EI}{L^3}&\frac{6EI}{L^2}\\0&\frac{6EI}{L^2}&\frac{4EI}{L}&0&\frac{-6EI}{L^2}&\frac{2EI}{L}\\\frac{-AE}{L}&0&0&\frac{AE}{L}&0&0\\0&\frac{-12EI}{L^3}&\frac{-6EI}{L^2}&0&\frac{12EI}{L^3}&\frac{-6EI}{L^2}\\0&\frac{6EI}{L^2}&\frac{2EI}{L}&0&\frac{-6EI}{L^2}&\frac{4EI}{L}\end{bmatrix}\tag{102}$$

where L is the bar length and the values of A (bar area), E (elasticity modulus), and I (moment of inertia) are free for the designer to choose.

D. 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 nondimensional flow parameters (minimum lift, maximum drag, etc.) and geometrical estimations (maximum and minimum thickness, curvature, volume, area, etc.).

The modularity must be a fundamental characteristic in optimal design software: The program ACTIV [24] is used to compute the objective function values and gradients using control theory, and the program Optimizer is used to find an optimum using either gradient or nongradient-based strategies. In our case, where a gradient-based optimization method has been used, the program Optimizer uses an INTA's version of the program CONMIN [25], a well-known software tool for the solution of nonlinear constrained optimization problems. The CONMIN uses a feasible search direction obtained from a compromise between the gradients of objective functions and the imposed constraints. At each design iteration, the CONMIN program requires as inputs the values and gradients of the objective functions as well as the chosen constraints.

E. Summary of Results

In this section some relevant results are given. The 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.

The gradients computed with the adjoint method described in this paper 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 adequately selected depending on the characteristics of the flow.

1. *Euler Transonic Redesign of a NACA 0012 Airfoil*

A single-point optimization case is used to show the accuracy of the developed continuous adjoint method for inviscid flows. The flow conditions are Mach number 0.8 with angle of attack 1.25 deg. The governing equations are the Euler equations, so drag improvement in this case means wave drag decrease.

In Fig. 1, a comparison between the gradient of the drag coefficient computed by finite-difference and adjoint methods is shown. The agreement is very good, with some minor discrepancies, likely owing to the finite step in finite-difference computations, becoming noticeable on the upper side of the airfoil downstream of the shock wave.

In the proposed transonic design problem, the objective function is the minimization of the wave drag, increasing the lift to 0.34 and with a minimum thickness of 10%, keeping the angle of attack fixed. The results are shown in Fig. 2; the shock wave has almost disappeared in few iterations.

After the optimization process the new airfoil has a drag coefficient of 0.0012, which is 5% of the original NACA 0012 drag (overall reduction of 200 counts). Also, the final lift coefficient is 105% greater than the original one.

2. *Viscous Subsonic Gradients of a Cylinder*

The main objective of this example is to evaluate the accuracy of the viscous drag gradients computed by means of the proposed continuous adjoint method for laminar viscous flows in a well-known subsonic problem. We are using a simple configuration of a cylinder facing a low velocity (Mach number 0.1) and low Reynolds number (equal to 50) flow that leads to a steady flow solution of the problem.

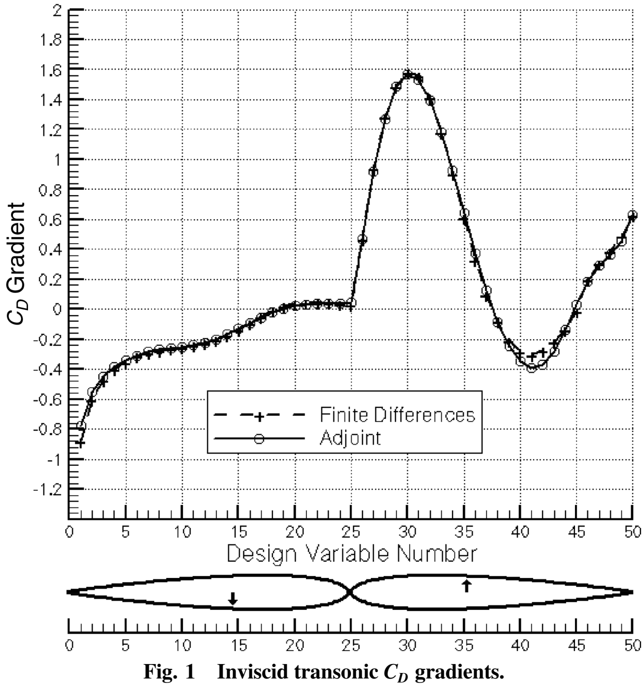

Fig. 1 Inviscid transonic C_D gradients.

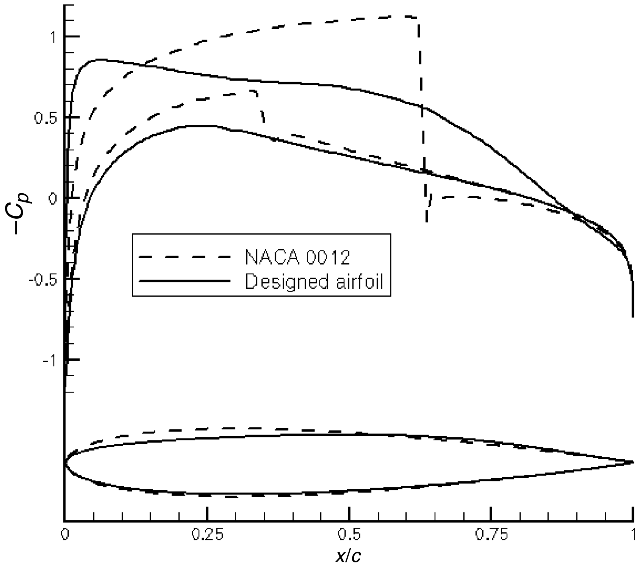

Fig. 2 Initial and designed C_p and geometry.

This test case clearly illustrates the necessity of the reduction of the second-order derivative terms. In Fig. 3, the finite-difference gradients are compared with the gradients computed with the adjoint method (with and without the reduction of the second-order derivative terms) on two different meshes (containing 200 and 400 nodes on the cylinder, respectively). In the case without reduction of second derivatives, first and second derivatives of the variables have been computed with both a first-order accurate Green–Gauss scheme and a least-squares scheme (see [26,27] for details). In the case with reduction, derivatives have been computed either with a Green–Gauss scheme or directly on the surface mesh (with a first-order central-difference scheme) when only tangent derivatives were required. The discrepancy between the results with and without the reduction of the second derivative terms and the improvement when the reduction is made are dramatic. In particular, it is noticeable how well the gradients computed with the reduction of the second derivatives agree with finite-difference computations.

In conclusion, the continuous adjoint approach provides very accurate gradients in a highly sensitive problem like this. Also, the

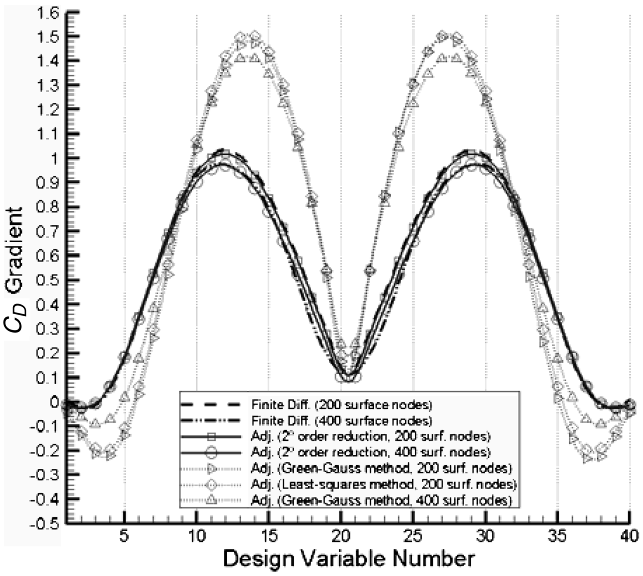

Fig. 3 Viscous subsonic C_D gradients for the cylinder.

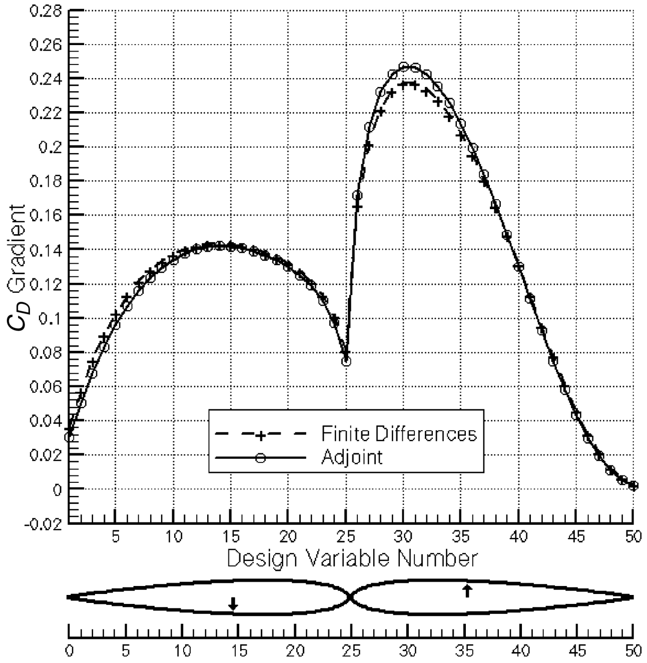

Fig. 4 Viscous laminar subsonic C_D gradients.

importance of the reduction of second-order derivative terms for obtaining numerical results of quality should be emphasized.

3. Viscous Subsonic Redesign of a NACA 0012 Airfoil

To study the accuracy of the developed methodology, a case of viscous laminar flow is selected. The flow conditions are Mach number equal to 0.3, angle of attack of 2.50 deg, and low Reynolds number of 1000 to keep the laminar flow along the airfoil. The proposed design problem starts with the flow conditions described above, the objective is drag minimization, increasing the lift to 0.15, using three geometrical constraints: minimum value for the greatest thickness (12%), frozen curvature at the leading edge, and minimum thickness at 75% of the chord.

In Fig. 4 a comparison between the gradients computed by finite-difference and adjoint methods is shown. The agreement is excellent. To obtain these results a hybrid mesh is used with 25 points on the viscous layer that has a thickness of 0.031 (in chord units).

The results of the optimization are shown in Fig. 5. After nine design cycles the new airfoil based on a NACA 0012 has a drag of 0.1225 that is 97% of the original NACA 0012 drag (reduction of 36 counts), while the final lift is 111% greater than the original one.

VI. Conclusions

In this work a systematic continuous adjoint approach to aerodynamic design optimization has been presented. The resulting expressions are suitable for optimization under viscous as well as inviscid flow conditions on unstructured as well as structured grids.

In the past, several drawbacks of the continuous adjoint approach for viscous flows on unstructured grids have been pointed out. One of the upshots of this work has been the resolution of some of those issues. The most significant is the need to compute second-order derivatives of the flow variables, which are required for computing sensitivity derivatives from the adjoint variables, which can be circumvented using the systematic procedure described in the paper. The procedure essentially amounts to using the flow equations restricted to the boundary to convert normal to tangent derivatives, and integrate by parts the later to reduce the overall order of derivatives. This is important because second-order derivatives computed using data from a spatially second-order accurate scheme (such as the one used in the present work, and the ones implemented in an ample majority of the unstructured flow solvers currently available) are not consistent, in general. This has been one of the major obstacles to using the continuous adjoint approach on unstructured meshes. After the reduction, the resulting gradients can be written in terms of first derivatives of the flow and adjoint variables only. But first derivatives can be consistently computed from second-order accurate data, which makes the technique suitable for application with most flow solvers.

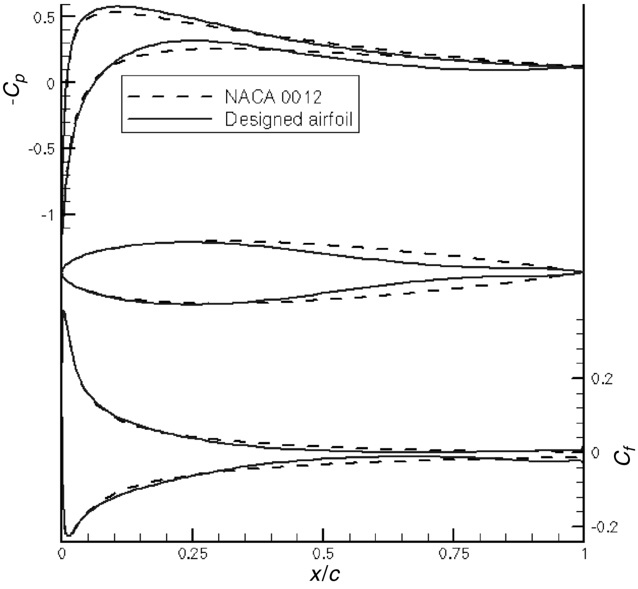

Fig. 5 Initial and designed C_p, C_f, and geometry.

Also, concerning the class of admissible optimization functionals, it has been shown that, for steady, compressible viscous flows, cost functions that depend solely on the pressure are admissible, and that consistent boundary conditions can be obtained without the need to formally include terms involving the stress tensor.

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 clearly illustrate the need to perform the reduction of the second derivative terms to obtain accurate gradients. Finally, the validity of the overall methodology has been illustrated with several design examples.

The results presented here are promising, but further numerical tests are necessary. In particular, a detailed study of the influence of the mesh sensitivities on the formulation, and of a possible strategy for incorporating them, should be carried out. Also, work to extend the methodology to deal with general turbulent three-dimensional flows (including the continuous adjoint formulation of the Spalart–Allmaras turbulence model) is currently in progress. We expect to report on these and related issues in the near future.

Appendix: Jacobian Matrices

Next, the definition of the Euler and Navier–Stokes Jacobian matrices is presented. The matrices are written in terms of primitive variables $V=(\rho,u,v,P)^T$. Switching to conservative variables $U=(\rho,\rho u,\rho v,\rho E)^T$ is accomplished with the aid of the transformation matrices

$$M=\frac{\partial U}{\partial V}=\begin{pmatrix}1 & \cdot & \cdot & \cdot\\ u & \rho & \cdot & \cdot\\ v & \cdot & \rho & \cdot\\ \frac{\boldsymbol{v}^2}{2} & \rho u & \rho u & \frac{1}{(\gamma-1)}\end{pmatrix}$$

$$M^{-1}=\begin{pmatrix}1 & \cdot & \cdot & \cdot\\ -\frac{u}{\rho} & \frac{1}{\rho} & \cdot & \cdot\\ -\frac{v}{\rho} & \cdot & \frac{1}{\rho} & \cdot\\ \frac{(\gamma-1)\boldsymbol{v}^2}{2} & (1-\gamma)u & (1-\gamma)v & (\gamma-1)\end{pmatrix} \tag{A1}$$

The Euler Jacobian matrices take the form

$$
\begin{aligned}
A_x &= \left(\frac{\partial F_x}{\partial V}\right)^T = \begin{pmatrix} u & u^2 & uv & \frac{1}{2}u\boldsymbol{v}^2 \\ \rho & 2\rho u & \rho v & \rho(H+u^2) \\ \cdot & \cdot & \rho u & \rho uv \\ \cdot & 1 & \cdot & \frac{u\gamma}{(\gamma-1)} \end{pmatrix} \\
A_y &= \left(\frac{\partial F_y}{\partial V}\right)^T = \begin{pmatrix} v & uv & v^2 & \frac{1}{2}v\boldsymbol{v}^2 \\ \cdot & \rho v & \cdot & \rho uv \\ \rho & \rho u & 2\rho v & \rho(H+v^2) \\ \cdot & \cdot & 1 & \frac{v\gamma}{(\gamma-1)} \end{pmatrix}
\end{aligned}
\tag{A2}
$$

whereas for viscous flux the appropriate matrices are

$$
A_x^v = -\left(\frac{\partial F_x^v}{\partial V}\right)^T = \begin{pmatrix} \cdot & -\frac{\partial\mu}{\partial\rho}\hat{\sigma}_{xx} & -\frac{\partial\mu}{\partial\rho}\hat{\sigma}_{xy} & -\frac{\partial\mu}{\partial\rho}\left(\hat{\sigma}_{xx}u + \hat{\sigma}_{xy}v + \frac{\gamma}{(\gamma-1)Pr}\frac{\partial}{\partial x}\left(\frac{P}{\rho}\right)\right) + \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial x}\left(\frac{P}{\rho^2}\right) \\ \cdot & \cdot & \cdot & -\sigma_{xx} \\ \cdot & \cdot & \cdot & -\sigma_{xy} \\ \cdot & -\frac{\partial\mu}{\partial P}\hat{\sigma}_{xx} & -\frac{\partial\mu}{\partial P}\hat{\sigma}_{xy} & -\frac{\partial\mu}{\partial P}\left(\hat{\sigma}_{xx}u + \hat{\sigma}_{xy}v + \frac{\gamma}{(\gamma-1)Pr}\frac{\partial}{\partial x}\left(\frac{P}{\rho}\right)\right) - \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial x}\left(\frac{1}{\rho}\right) \end{pmatrix}
\tag{A3}
$$

$$
A_y^v = -\left(\frac{\partial F_y^v}{\partial V}\right)^T = \begin{pmatrix} \cdot & -\frac{\partial\mu}{\partial\rho}\hat{\sigma}_{yx} & -\frac{\partial\mu}{\partial\rho}\hat{\sigma}_{yy} & -\frac{\partial\mu}{\partial\rho}\left(\hat{\sigma}_{yx}u + \hat{\sigma}_{yy}v + \frac{\gamma}{(\gamma-1)Pr}\frac{\partial}{\partial y}\left(\frac{P}{\rho}\right)\right) + \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial y}\left(\frac{P}{\rho^2}\right) \\ \cdot & \cdot & \cdot & -\sigma_{yx} \\ \cdot & \cdot & \cdot & -\sigma_{yy} \\ \cdot & -\frac{\partial\mu}{\partial P}\hat{\sigma}_{yx} & -\frac{\partial\mu}{\partial P}\hat{\sigma}_{yy} & -\frac{\partial\mu}{\partial P}\left(\hat{\sigma}_{yx}u + \hat{\sigma}_{yy}v + \frac{\gamma}{(\gamma-1)Pr}\frac{\partial}{\partial y}\left(\frac{P}{\rho}\right)\right) - \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial y}\left(\frac{1}{\rho}\right) \end{pmatrix}
$$

where $\hat{\sigma}_{ij} = \partial_i v_j + \partial_j v_i - \frac{2}{3}\delta_{ij}\boldsymbol{\nabla}\cdot\boldsymbol{v}$, and

$$
\begin{aligned}
D_{xx} &= \left(\frac{\partial F_x^v}{\partial(\partial_x V)}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & -\frac{\gamma\mu}{(\gamma-1)Pr}\frac{P}{\rho^2} \\ \cdot & \frac{4}{3}\mu & \cdot & \frac{4}{3}u\mu \\ \cdot & \cdot & \mu & v\mu \\ \cdot & \cdot & \cdot & \frac{\gamma}{(\gamma-1)Pr}\frac{\mu}{\rho} \end{pmatrix} \\
D_{yx} &= \left(\frac{\partial F_y^v}{\partial(\partial_x V)}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & -\frac{2}{3}\mu & -\frac{2}{3}v\mu \\ \cdot & \mu & \cdot & u\mu \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \\
D_{xy} &= \left(\frac{\partial F_x^v}{\partial(\partial_y V)}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \mu & v\mu \\ \cdot & -\frac{2}{3}\mu & \cdot & -\frac{2}{3}u\mu \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \\
D_{yy} &= \left(\frac{\partial F_y^v}{\partial(\partial_y V)}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & -\frac{\gamma\mu}{(\gamma-1)Pr}\frac{P}{\rho^2} \\ \cdot & \mu & \cdot & u\mu \\ \cdot & \cdot & \frac{4}{3}\mu & \frac{4}{3}v\mu \\ \cdot & \cdot & \cdot & \frac{\gamma}{(\gamma-1)Pr}\frac{\mu}{\rho} \end{pmatrix}
\end{aligned}
\tag{A4}
$$

In the derivation of the viscous Jacobians in Eq. (A3) the dependence of the laminar viscosity and heat conduction coefficients μ and k on the flow has been explicitly taken into account. If boundary deformations result in large variations of those coefficients (that is to say, if $\delta\mu$ and δk are not negligible), then the corresponding terms in Eq. (A3) must be taken into account when solving the adjoint equations. Otherwise, these terms can be dropped, which notably simplifies the resulting expressions

$$
\begin{aligned}
A_x^v &= -\left(\frac{\partial F_x^v}{\partial V}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial x}\left(\frac{P}{\rho^2}\right) \\ \cdot & \cdot & \cdot & -\sigma_{xx} \\ \cdot & \cdot & \cdot & -\sigma_{xy} \\ \cdot & \cdot & \cdot & -\frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial x}\left(\frac{1}{\rho}\right) \end{pmatrix} \\
A_y^v &= -\left(\frac{\partial F_y^v}{\partial V}\right)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial y}\left(\frac{P}{\rho^2}\right) \\ \cdot & \cdot & \cdot & -\sigma_{yx} \\ \cdot & \cdot & \cdot & -\sigma_{yy} \\ \cdot & \cdot & \cdot & -\frac{\gamma}{(\gamma-1)Pr}\mu\frac{\partial}{\partial y}\left(\frac{1}{\rho}\right) \end{pmatrix}
\end{aligned}
\tag{A5}
$$

Acknowledgments

This work has been supported in part by the Spanish Ministry of Education and Science (MEC) under Project DOMINO (CIT-370200-2005-10) and grants BFM2002-03345 and MTM2005-00714, by the Comunidad Autónoma de Madrid (CAM) under Project SIMUMAT (S-0505/ESP-0158), and by the Instituto Nacional de Técnica Aeroespacial (INTA) under the activity "Termofluidodinámica" (IGB4400903). The authors would like to thank the associate editor and reviewers of the AIAA Journal for their valuable comments.

References

[1] Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 1, June 1974, pp. 97–110.

[2] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, Sept. 1988, pp. 233–260.

[3] Jameson, A., "Optimum Aerodynamic Design Using CFD and Control Theory," AIAA Paper 95-1729, 1995.

[4] Monge, F., and Palacios, F., "Multipoint Airfoil Optimisation Using Control Theory," *Fourth European Congress on Computational Methods in Applied Sciences and Engineering* CD ISBN 951-39-1868-8, 2004.

[5] Zuazua, E., "Propagation, Observation, and Control of Waves Approximated by Finite Difference Methods," *SIAM Review*, Vol. 47, No. 2, 2005, pp. 197–243.

[6] Nadarajah, S. K., and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," AIAA Paper 2000-0667, 2000.

[7] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," *Computers and Fluids*, Vol. 28, Nos. 4–5, 1999, pp. 443–480.

[8] Giles, M. B., and Pierce, N. A., "Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behavior," AIAA Paper 97-1850, 1997.

[9] Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," AIAA Paper 97-0101, 1997.

[10] Arian, E., and Salas, M. D., "Admitting the Inadmissible: Adjoint Formulation for Incomplete Cost Functionals in Aerodynamic Optimization," *AIAA Journal*, Vol. 37, No. 1, 1999, pp. 37–44.

[11] Jameson, A., Sriram, S., and Martinelli, L., "A Continuous Adjoint Method for Unstructured Grids," AIAA Paper 2003-3955, 2003.

[12] Jameson, A., Sriram, S., Martinelli, L., and Haimes, B., "Aerodynamic Shape Optimization of Complete Aircraft Configurations Using Unstructured Grids," AIAA Paper 2004-533, 2004.

[13] Hadamard, J., *Leçons sur le Calcul des Variations*, Gauthier–Villars, Paris, 1910.

[14] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[15] Jameson, A., and Kim, S., "Reduction of the Adjoint Gradient Formula in the Continuous Limit," AIAA Paper 2003-0040, Jan. 2003.

[16] Castro, C., Lozano, C., Palacios, F., and Zuazua, E., "Control of Euler and Navier-Stokes Equations: Applications to Optimal Shape Design in Aeronautics," *Proceedings of the 9th European Control Conference (ECC'07)* [CD-ROM], Paper ThC05.5, July 2007.

[17] Albiges, M., Coin, A., and Journet, H., *Études des Structures par les Méthodes Matricielles*, Eyrolles Editeur, Paris, 1970.

[18] Giles, M. B., Pierce, N. A., and Süli, E., "Progress in Adjoint Error Correction for Integral Functionals," *Computing and Visualization in Science*, Vol. 6, Nos. 2–3, 2004, pp. 113–121.

[19] Eliasson, P., "EDGE, A Navier-Stokes Solver for Unstructured Grids," FOI Scientific Rept. FOI-R-0298-SE, 2002.

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

[21] Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper AIAA-89-0366, 1989.

[22] Weiss, J. M., Maruszewski, J. P., and Wayne, A. S., "Implicit Solution of the Navier-Stokes Equation on Unstructured Meshes," AIAA Paper 1997-2103, 1997.

[23] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," AIAA Paper 77-1247, Aug. 1977.

[24] Monge, F., and Palacios, F., "Inviscid Multipoint Airfoil Optimisation Using Control Theory," *ERCOFTAC Design Optimisation: Methods & Applications. International Conference & Advanced Course Program*, CD conference paper, 2004.

[25] Vanderplaats, G. N., "CONMIN—A FORTRAN Program for Constrained Function Minimization: User's Manual," NASA Rept. TM-X-62282, 1973.

[26] Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," AIAA Paper AIAA-93-0668, 1993.

[27] Ashford, G., "An Unstructured Grid Generation and Adaptive Solution Technique for High-Reynolds-Number Compressible Flows," Ph.D. Thesis, University of Michigan, Ann Arbor, MI, 1996.

Z. Wang
Associate Editor