Aerodynamic Shape Optimization Using "Turbulent" Adjoint And Robust Design in Fluid Mechanics

Kyriakos C. Giannakoglou, Dimitrios I. Papadimitriou, Evangelos M. Papoutsis-Kiachagias and Ioannis S. Kavvadias

Abstract This article presents adjoint methods for the computation of the first- and higher-order derivatives of objective functions F used in optimization problems governed by the Navier–Stokes equations in aero/hydrodynamics. The first part of the chapter summarizes developments and findings related to the application of the continuous adjoint method to turbulence models, such as the Spalart-Allmaras and k-ε ones, in either their low- or high-Reynolds number (with wall functions) variants. Differentiating the turbulence model, over and above to the differentiation of the mean–flow equations, leads to the computation of the exact gradient of F, by overcoming the frequently made assumption of neglecting turbulence variations. The second part deals with higher-order sensitivity analysis based on the combined use of the adjoint approach and the direct differentiation of the governing PDEs. In robust design problems, the so-called second-moment approach requires the computation of second-order derivatives of F with respect to (w.r.t.) the environmental or uncertain variables; in addition, any gradient-based optimization algorithm requires third-order mixed derivatives w.r.t. both the environmental and design variables; various ways to compute them are discussed and the most efficient is adopted. The equivalence of the continuous and discrete adjoint for this type of computations is demonstrated. In the last part, some other relevant recent achievements regarding the adjoint approach are discussed. Finally, using the aforementioned adjoint methods, industrial geometries are optimized. The application domain includes both incompressible or compressible fluid flow applications.

K.C. Giannakoglou (✉) · D.I. Papadimitriou · E.M. Papoutsis-Kiachagias · I.S. Kavvadias
National Technical University of Athens, Athens, Greece
e-mail: kgianna@central.ntua.gr

D.I. Papadimitriou
e-mail: dpapadim@mail.ntua.gr

E.M. Papoutsis-Kiachagias
e-mail: vaggelisp@gmail.com

I.S. Kavvadias
e-mail: kavvadiasj@hotmail.com

N.D. Lagaros and M. Papadrakakis (eds.), *Engineering and Applied Sciences Optimization*, Computational Methods in Applied Sciences 38,
DOI 10.1007/978-3-319-18320-6_16

1 Flow Equations and Objective Function

In this section, the equations governing the state (i.e. flow) problem for incompressible fluid flows, using either the one-equation Spalart-Allmaras [1] or the two equation Launder-Sharma $k-\varepsilon$ [2] turbulence models, are briefly presented. The mean–flow state equations are

$$R^p = -\frac{\partial \upsilon_j}{\partial x_j} = 0 \tag{1}$$

$$R^{\upsilon_i} = \upsilon_j \frac{\partial \upsilon_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[(\nu+\nu_t)\left(\frac{\partial \upsilon_i}{\partial x_j}+\frac{\partial \upsilon_j}{\partial x_i}\right)\right] = 0 \tag{2}$$

where υ_i are the velocity components, p the static pressure divided by the density, ν and ν_t the bulk and turbulent viscosities. The turbulence model (TM) equation(s) is/are

$$R^{\widetilde{\nu}} = \upsilon_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b_2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu}P\left(\widetilde{\nu}\right) + \widetilde{\nu}D\left(\widetilde{\nu}\right) = 0 \tag{3}$$

for the Spalart-Allmaras model (TM = SA) and

$$\begin{aligned} R^k &= \upsilon_j \frac{\partial k}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_t}{Pr_k}\right)\frac{\partial k}{\partial x_j}\right] - P_k + \varepsilon + D = 0 \\ R^\varepsilon &= \upsilon_j \frac{\partial \varepsilon}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_t}{Pr_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] - c_1 P_k \frac{\varepsilon}{k} + c_2 f_2 \frac{\varepsilon^2}{k} - E = 0 \end{aligned} \tag{4}$$

for the Launder-Sharma k-ε (TM = KE) one. $\widetilde{\nu}$ is the turbulence state variable if TM=SA ($\nu_t = \widetilde{\nu} f_{\upsilon_1}$) and k, ε are the corresponding quantities (turbulent kinetic energy and turbulent energy dissipation) if TM = KE ($\nu_t = c_\mu \frac{k^2}{\varepsilon}$). In both cases, the boundary conditions and the model constant values are omitted in the interest of space; see [1] and [2].

In general, the objective function may comprise both surface (S) and volume (Ω) integrals, as follows

$$F = \int_S F_S dS + \int_\Omega F_\Omega d\Omega = \int_S F_{S_i} n_i dS + \int_\Omega F_\Omega d\Omega \tag{5}$$

where n_i are the components of the normal to the boundary outward unit vector.

2 The Adjoint Method for Shape Optimization in Turbulent Flows

In discrete adjoint, the differentiation of the turbulence model equations is straightforward and can be found in several published works, [3, 4]. In contrast, the majority of existing continuous adjoint methods/codes rely on the so-called "frozen turbulence" assumption, in which the sensitivities of the turbulence quantities w.r.t. the design variables are neglected [5–9]. The first continuous adjoint to the Spalart-Allmaras model, for incompressible flows, was presented by the current group of authors in [10] and was extended to compressible flows in [11]. Regarding the adjoint approach to high-Reynolds turbulence models, the continuous adjoint to the k-ε model with wall functions has recently been published, [12], whereas the continuous adjoint to the low-Reynolds Launder-Sharma k-ε model can be found in [13]. All these adjoint approaches which rely upon the differentiated turbulence model will hereafter be referred to as "turbulent" adjoint, to distinguish it from the "frozen turbulence" approach.

2.1 Continuous Adjoint to Low-Re Turbulence Models

In the continuous adjoint approach for shape optimization problems, the total derivative (symbol δ) of any function Φ w.r.t. the design variables b_n must be distinguished from the corresponding partial sensitivity (symbol ∂) since

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_l}\frac{\delta x_l}{\delta b_n} \tag{6}$$

where $\frac{\delta x_l}{\delta b_n}$ are the sensitivities of nodal coordinates. In case Φ is defined along a surface, Eq. 6 becomes $\frac{\delta_s \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m$. Since any sufficiently small surface deformation can be seen as a normal perturbation, only the normal part of the surface deformation velocity $\delta x_m / \delta b_n n_m$ contributes to changes in Φ.

In order to formulate the adjoint method, the augmented objective function F_{aug} is defined as the sum of F and the field integrals of the products of the adjoint variable fields and the state equations, as follows

$$F_{aug} = F + \int_{\Omega} u_i R_i^{v} d\Omega + \int_{\Omega} q R^{p} d\Omega + E_{TM} \tag{7}$$

where u_i are the adjoint velocity components, q the adjoint pressure and the extra terms E_{TM} depend on the turbulence model (TM). If $TM = SA$,

$$E_{SA} = \int_{\Omega} \widetilde{\nu_a} R^{\widetilde{\nu}} d\Omega \tag{8}$$

whereas if $TM = KE$,

$$E_{KE} = \int_{\Omega} \left(k_a R^k + \varepsilon_a R^{\varepsilon} \right) d\Omega \tag{9}$$

where $\widetilde{\nu_a}$, k_a and ε_a are the adjoints to $\widetilde{\nu}$, k and ε, respectively.

Based on the Leibniz theorem, the derivative of F_{aug} w.r.t. b_n is

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_{\Omega} u_i \frac{\partial R_i^{\upsilon}}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega \tag{10a}$$
$$+ \int_{S_{W_p}} \left(u_i R_i^{\upsilon} + q R^p \right) \frac{\delta x_k}{\delta b_n} n_k dS + \frac{\delta (E_{TM})}{\delta b_n}$$

$$\frac{\delta (E_{SA})}{\delta b_n} = \int_{\Omega} \widetilde{\nu_a} \frac{\partial R^{\widetilde{\nu}}}{\partial b_n} d\Omega + \int_{S_{W_p}} \widetilde{\nu_a} R^{\widetilde{\nu}} \frac{\delta x_k}{\delta b_n} n_k dS \tag{10b}$$

$$\frac{\delta (E_{KE})}{\delta b_n} = \int_{\Omega} k_a \frac{\partial R^k}{\partial b_n} d\Omega + \int_{\Omega} \varepsilon_a \frac{\partial R^{\varepsilon_a}}{\partial b_n} d\Omega + \int_{S_{W_p}} \left(k_a R^k + \varepsilon_a R^{\varepsilon} \right) \frac{\delta x_k}{\delta b_n} n_k dS \tag{10c}$$

where S_{W_p} is the parameterized (in terms of b_n) part of the solid wall. The development of the volume integrals in Eqs. 10a, b, based on the Green-Gauss theorem and the elimination of terms depending on the variations of the mean–flow and turbulence model variables, lead to the adjoint mean–flow equations

$$R^q = \frac{\partial u_j}{\partial x_j} = 0 \tag{11}$$

$$R_i^u = u_j \frac{\partial \upsilon_j}{\partial x_i} - \frac{\partial (\upsilon_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \frac{\partial q}{\partial x_i} + AMS_i = 0 \tag{12}$$

The extra terms AMS_i arise from the differentiation of the turbulence model, see [10, 13]. The adjoint turbulence model variables fields $\widetilde{\nu_a}$, k_a and ε_a are governed by the "turbulent" adjoint PDEs, which are

$$R^{\widetilde{\nu_a}} = -\frac{\partial (\upsilon_j \widetilde{\nu_a})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \widetilde{\nu_a}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_j} \right)$$
$$+ \widetilde{\nu_a} \widetilde{\nu} C_{\widetilde{\nu}} + \frac{\partial \nu_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial \upsilon_i}{\partial x_j} + \frac{\partial \upsilon_j}{\partial x_i} \right) + (-P + D) \widetilde{\nu_a} = 0 \tag{13a}$$

$$R^{k_a} = -\frac{\partial (\upsilon_j k_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{Pr_k} \right) \frac{\partial k_a}{\partial x_j} \right]$$
$$+ \left(\frac{B_1}{Pr_k} - \frac{\nu}{k} \right) \frac{\partial k}{\partial x_j} \frac{\partial k_a}{\partial x_j} + \frac{B_1}{Pr_{\varepsilon}} \frac{\partial \varepsilon}{\partial x_j} \frac{\partial \varepsilon_a}{\partial x_j} + B_1 \left(\frac{\partial \upsilon_i}{\partial x_j} + \frac{\partial \upsilon_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j}$$

$$
+\left[\frac{\nu}{2k^2}\left(\frac{\partial k}{\partial x_j}\right)^2-\frac{\nu}{k}\frac{\partial^2 k}{\partial x_j^2}-PB_1\right]k_a
$$
$$
-\left[c_1\frac{\varepsilon}{k}PB_1+2\nu\left(\frac{\partial^2 \upsilon_k}{\partial x_i \partial x_j}\right)^2 B_1+c_2 f_2\frac{\varepsilon^2}{k^2}-1.2c_2\frac{k^2}{\nu^2}e^{-Re_t^2}-c_1 P_k\frac{\varepsilon}{k^2}\right]\varepsilon_a=0 \tag{13b}
$$

$$
R^{\varepsilon_a}=-\frac{\partial(\upsilon_j\varepsilon_a)}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_t}{Pr_\varepsilon}\right)\frac{\partial\varepsilon_a}{\partial x_j}\right]
$$
$$
+\frac{B_2}{Pr_\varepsilon}\frac{\partial\varepsilon}{\partial x_j}\frac{\partial\varepsilon_a}{\partial x_j}+\frac{B_2}{Pr_k}\frac{\partial k}{\partial x_j}\frac{\partial k_a}{\partial x_j}+B_2\left(\frac{\partial\upsilon_i}{\partial x_j}+\frac{\partial\upsilon_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j}+(1-PB_2)k_a
$$
$$
+\left[-2\nu\left(\frac{\partial^2\upsilon_k}{\partial x_i\partial x_j}\right)^2 B_2-c_1\frac{\varepsilon}{k}PB_2+2c_2 f_2\frac{\varepsilon}{k}-0.6c_2\frac{k^3}{\nu^2\varepsilon}e^{-Re_t^2}-c_1 P_k\frac{1}{k}\right]\varepsilon_a=0 \tag{13c}
$$

The detailed derivation of the adjoint PDEs, the various terms or constants in Eqs. 13a, b, c and the corresponding adjoint boundary conditions can be found in [10] or [13].

After satisfying the field adjoint equations, the sensitivity derivatives of F_{aug} are given by

$$
\frac{\delta F_{aug}}{\delta b_n}=\int_S BC_i^u\frac{\partial\upsilon_i}{\partial b_n}dS+\int_S(u_j n_j+\frac{\partial F_{S_i}}{\partial p}n_i)\frac{\partial p}{\partial b_n}dS+\int_S(-u_i n_j+\frac{\partial F_{S_k}}{\partial\tau_{ij}}n_k)\frac{\partial\tau_{ij}}{\partial b_n}dS
$$
$$
+\int_{S_{W_p}}n_i\frac{\partial F_{S_{W_p,i}}}{\partial x_m}n_m\frac{\delta x_k}{\delta b_n}n_k dS+\int_{S_{W_p}}F_{S_{W_p,i}}\frac{\delta n_i}{\delta b_n}dS+\int_{S_{W_p}}F_{S_{W_p,i}}n_i\frac{\delta(dS)}{\delta b_n}
$$
$$
+\int_{S_{W_p}}(u_i R_i^\upsilon+qR^p)\frac{\delta x_k}{\delta b_n}n_k dS+SD \tag{14}
$$

where, depending on the turbulence model, terms BC_i^u and SD can be found in [10] or [13]. The gain from using the “turbulent” adjoint approach and overcoming the “frozen turbulence” assumption, at the expense of additionally solving the adjoint to the turbulence model PDEs, is demonstrated below in a few selected cases. The “frozen turbulence” assumption may lead to wrongly signed sensitivities, misleading or delaying the optimization process. As an example, the optimization of a 90° elbow duct, targeting minimum total pressure losses, $minF=-\int_{S_I}\left(p+\frac{1}{2}\upsilon^2\right)\upsilon_i n_i dS-\int_{S_O}\left(p+\frac{1}{2}\upsilon^2\right)\upsilon_i n_i dS$, where S_I and S_O are the inlet to and outlet from the flow domain, with a Reynolds number equal to 3.5×10^4, modeled using TM=SA model is demonstrated in Fig. 1, [10]. Comments can be found in the caption.

The shape optimization of an S-shaped duct, with the same target as before, is demonstrated in Fig. 2. The flow Reynolds number based on the inlet height is $Re=1.2\times10^5$ and TM=KE is used. The upper and lower duct contours are parameterized using Bézier–Bernstein polynomials with 12 control points each. The Fletcher-Reeves Conjugate Gradient (CG) method is used. The gradients used by

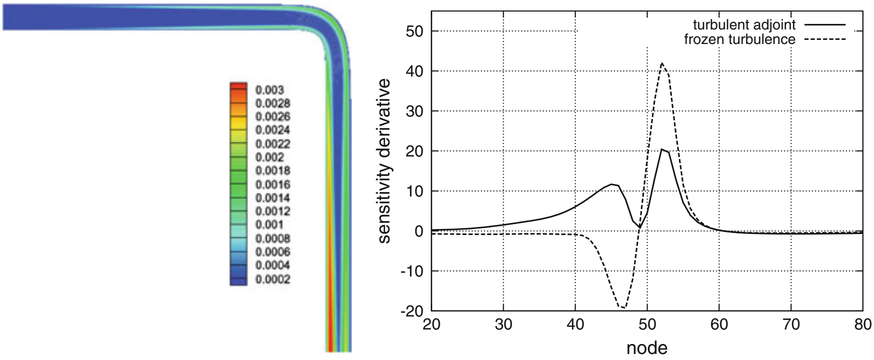

Fig. 1 Adjoint to the low-Re Spalart–Allmaras model: *Left* adjoint pressure field in a 90° elbow duct with constant cross-section. *Right* sensitivity derivatives of the total pressure losses function ($\delta F/\delta b_n$), where b_n are the normal displacements of the solid wall grid nodes. Two sensitivity distributions, close to the 90° bend, are compared. The abscissa stands for the nodal numbers of the wall nodes. By making the "frozen turbulence" assumption, wrongly signed sensitivities between nodes 20 and 50 are computed. Extensive validation of the adjoint solver against direct differentiation is conducted in [10]

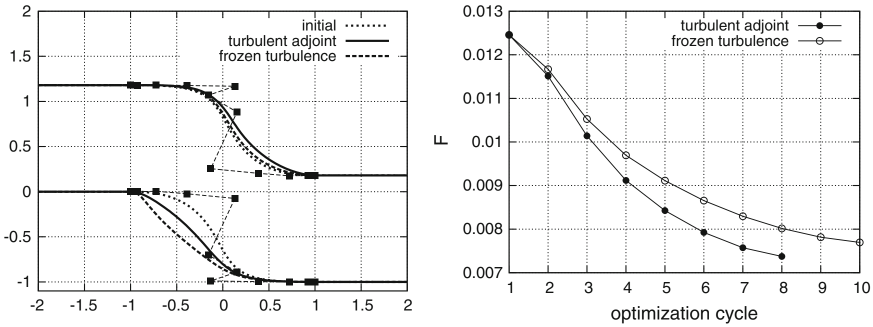

Fig. 2 Adjoint to the low-Re Launder–Sharma k–ε model: Shape optimization of an S-shaped duct targeting minimum total pressure losses. *Left* Starting duct shape compared to the optimal shapes resulting from **a** "turbulent" adjoint and **b** adjoint based on the "frozen turbulence" assumption; axes not in scale. *Right* Convergence history of the CG algorithm driven by the two different adjoint methods. From [13]

each method to update the design variables are based on (a) "turbulent" adjoint and (b) adjoint with the "frozen turbulence" assumption. The starting duct shape along with the optimal ones computed by CG, based on the two variants of the adjoint formulation, are presented in Fig. 2. The shape resulting from (a) has an F value by about 3 % lower than that of (b) and reaches the optimal solution after ~20 % less cycles.

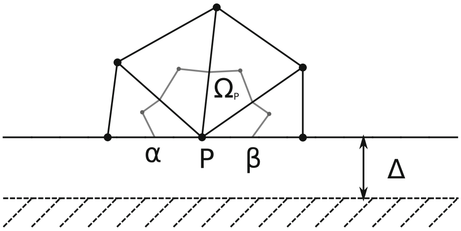

Fig. 3 The adjoint technique with wall functions: A vertex-centered finite volume Ω_P associated with the "wall" (*horizontal line*) at node P. The real solid wall lies underneath P, at a distance Δ

2.2 Continuous Adjoint to High-Re Turbulence Models

In industrial projects, many analysis codes rely on the use of the wall function (WF) technique, due to the less stretched and generally coarser meshes required close to the walls and the resulting economy in the overall CPU cost. The development of the adjoint approach to the wall function model is, thus, necessary. This is briefly presented below for the k-ε and the Spalart-Allmaras models. The two developments differ since the first was based on the in-house GPU-enabled RANS solver, [14], with slip velocity at the wall, [15], while the second on the OpenFOAM code with a no-slip condition at the wall. Note that these differences in the primal boundary conditions at the wall cause differences in the corresponding adjoint boundary conditions.

Regarding the k-ε model, the development, which was carried out by the authors' group [12] was based on vertex-centered finite volumes with non-zero slip velocity at the wall. The real solid wall is assumed to lie at a distance Δ underneath S_W. Integrating the state equations over the finite volume of Fig. 3, the diffusive flux across segment $\alpha\beta$ depends on the friction velocity υ_τ,

$$\upsilon_\tau^2 = (\nu + \nu_t)\left(\frac{\partial \upsilon_i}{\partial x_j} + \frac{\partial \upsilon_j}{\partial x_i}\right) n_j t_i \tag{15}$$

and $\upsilon_t = \upsilon_i t_i$; υ_t computed via the local application of the law of the wall. With known υ_τ, the k and ε values at P are

$$\begin{aligned} k_P &= \frac{\upsilon_\tau^2}{\sqrt{c_\mu}}, & \varepsilon_P &= \frac{\upsilon_\tau^3}{\kappa \Delta}, & &\text{if} \quad y^+ \geq y_c^+ \\ k_P &= \frac{\upsilon_\tau^2}{\sqrt{c_\mu}}\left(\frac{y^+}{y_c^+}\right)^2, & \varepsilon_P &= k_P^{\frac{3}{2}} \frac{1+\frac{5.3\nu}{\sqrt{k_P \Delta}}}{\kappa c_\mu^{-\frac{3}{4}} \Delta}, & &\text{if} \quad y^+ < y_c^+ \end{aligned} \tag{16}$$

where $y^+ = \frac{\upsilon_\tau \Delta}{\nu}$, $\upsilon^+ = \frac{\upsilon_t}{\upsilon_\tau}$, which result from the expressions $\upsilon^+ = \frac{1}{\kappa} lny^+ + B$, with $\kappa = 0.41$ and $B = 5.5$, if $y^+ \geq y_c^+$ or $\upsilon^+ = y^+$ if $y^+ < y_c^+$.

Similar to the definition of υ_τ, Eq. 15, the development of the adjoint equations introduces the adjoint friction velocity u_τ at each "wall" node (such as P) defined by, [12],

$$u_\tau^2 = (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j t_i \tag{17}$$

Attention should be paid to the close similarity of Eqs. 17 and 15. During the solution of the adjoint PDEs (TM=KE), the value of u_τ, which contributes to the adjoint viscous fluxes at the "wall" nodes, is expressed in terms of the gradients of k, k_a, ε and ε_a, as follows

$$u_\tau^2 = \frac{1}{c_\upsilon}\left[2u_k t_k \upsilon_\tau - \left(\nu + \frac{\nu_t}{Pr_k}\right)\frac{\partial k_a}{\partial x_j} n_j \frac{\delta k}{\delta \upsilon_\tau} - \left(\nu + \frac{\nu_t}{Pr_\varepsilon}\right)\frac{\partial \varepsilon_a}{\partial x_j} n_j \frac{\delta \varepsilon}{\delta \upsilon_\tau}\right] \tag{18}$$

On the other hand, if TM=SA (based on a cell-centered finite-volume scheme with a no-slip condition at the solid wall boundary faces), the wall function technique is based on a single formula modeling both the viscous sublayer and the logarithmic region of the boundary layer

$$f_{WF} = y^+ - \upsilon^+ - e^{-\kappa B}\left[e^{\kappa\upsilon^+} - 1 - \kappa\upsilon^+ - \frac{(\kappa\upsilon^+)^2}{2} - \frac{(\kappa\upsilon^+)^3}{6}\right] = 0 \tag{19}$$

In this case, the adjoint friction velocity must be zeroed. This is the major difference between the two finite–volume approaches (cell– and vertex–centered); despite this difference and any difference in the interpretation of the adjoint friction velocity, both will be referred to as "adjoint wall function" technique. Here, also, the role of (zero) u_τ is to complete the adjoint momentum equilibrium at the first cell adjacent to the wall. The development is omitted in the interest of space. Applications, including validation, of the "adjoint wall function" technique are shown in Figs. 4 and 5, with comments in the caption.

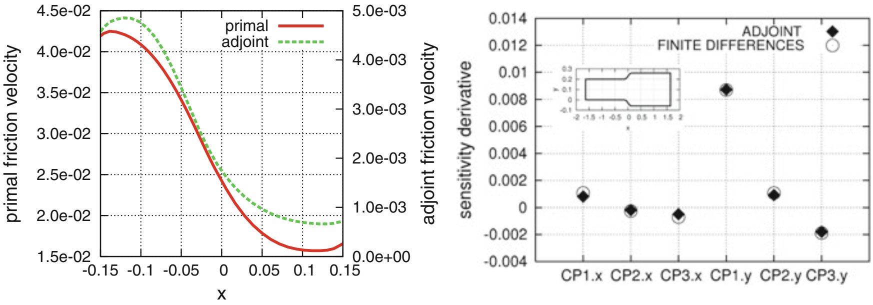

Fig. 4 Adjoint to the high-Re k-ε model: Optimization of an axial diffuser, for minimum total pressure losses, using the "adjoint wall function" technique. *Left* Friction velocity υ_τ and adjoint friction velocity u_τ distributions along its lower wall. *Right* Sensitivity derivatives of F w.r.t. the design variables, i.e. the coordinates of Bézier control points parameterizing the side walls. The "adjoint wall function" method perfectly matches the sensitivity derivatives computed by finite differences. From [12]

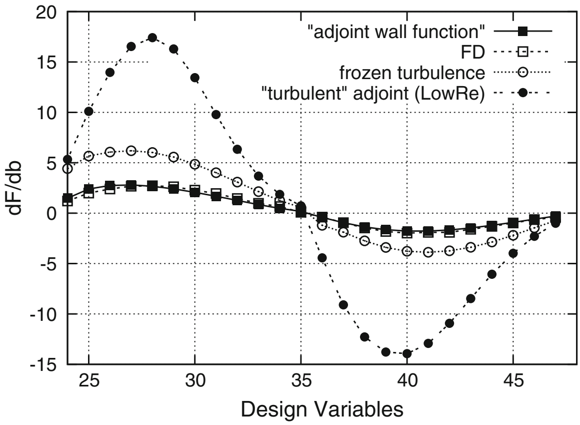

Fig. 5 Adjoint to the high-Re Spalart–Allmaras model, flow around the isolated NACA0012 airfoil, $\alpha_\infty = 3°$, $Re = 6 \times 10^6$: Drag sensitivities computed using the “adjoint wall function” method are compared to finite-differences (FD), the adjoint method using the “frozen turbulence” assumption and the adjoint method with the “low–Reynolds” approach. The latter implies that the turbulence model is differentiated but the differentiation of the wall functions is disregarded. Only the latest 24 design variables, namely the y coordinates of the control points, where the magnitude of the computed derivatives is greater, are considered; the first 12 correspond to the suction side and the other to the pressure side. It is interesting to note that the “low-Re” adjoint approach performs even worse than the “frozen turbulence” one. In other words, the incomplete differentiation of the turbulence model produces worse results than its complete omission!

Recently, the “turbulent adjoint” method for the $k - \omega$ SST turbulence model with wall functions was published by the same group, [16].

2.3 Other Applications of the Continuous Adjoint Methods

The continuous adjoint method is a low-cost tool to derive information regarding the optimal location and type of steady suction/blowing jets, used to control flow separation, [13]. In unsteady flows, the adjoint method, [17], can also be used to compute the optimal characteristics of unsteady jets, such as pulsating or oscillating ones. Such an application, where the optimal amplitudes of pulsating jets have been computed using the unsteady continuous adjoint method is presented in Fig. 6.

An inherent difficulty of the adjoint method, applied to unsteady flows, is the need of having the primal solution field available for the solution of the adjoint equations in each time-step. Since the adjoint solution evolves backwards in time, the need to store every primal solution arises. Since storing everything is expensive memory-wise, some turnaround is often used instead. A very common approach is the checkpointing technique, [18], where selected primal flow fields are stored (checkpoints) and the rest are recomputed starting from the checkpoints. Checkpointing is much cheaper memory-wise, at the expense of extra CPU time, needed for the re-computations of the primal fields.

Slot	*Amplitude*
1	0.0160
2	0.0301
3	0.0315
4	0.0068
5	−0.0400
6	−0.0596
7	−0.0147
8	0.0776

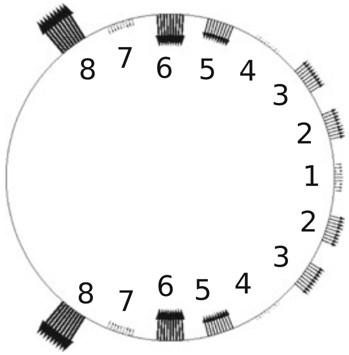

Fig. 6 Time-averaged drag minimization of the unsteady flow around a cylinder at $Re = 100$, using pulsating jets: Optimal amplitudes for the symmetrically placed jets computed by the continuous adjoint method

Another way to overcome this is to use an approximation to the time evolution of the primal fields, such as the proper orthogonal decomposition (POD) technique, [19]. Approximating the primal field of each iteration bears no extra CPU cost.

On the other hand, topology optimization in fluid mechanics exclusively relies upon the adjoint method. In these problems, a real-valued porosity (α) dependent term is introduced into the flow equations. Based on the local porosity values, domain areas corresponding to the fluid flow are identified as those with nodal values $\alpha \leq \varepsilon$, where ε is a user-defined infinitesimally small positive number. All the remaining areas where $\alpha > \varepsilon$ define the part of the domain to be solidified. The goal of topology optimization is to compute the optimal α field in order to minimize the objective function under consideration. Since the number of the design variables is equal to the number of mesh cells (and thus, very high), the adjoint method is the perfect choice for computing $\delta F/\delta\alpha$, as its cost is independent of the number of design variables. Continuous adjoint methods for solving topology optimization problems for laminar and turbulent ducted flows, with or without heat transfer, are described in [20]. For turbulent flows, the adjoint approach is exact, i.e. includes the differentiation of the turbulence model ("turbulent" adjoint), (Fig. 7).

3 Robust Design Using High-Order Sensitivity Analysis

In aerodynamics, robust design methods aim at optimizing a shape in a range of possible operating conditions or by considering environmental uncertainties, such as manufacturing imprecisions or fluctuations of flow conditions, etc. The latter depend on the so-called environmental variables $\mathbf{c}$ (c_i, $i \in [1, M]$). In robust design problems, the function to be minimized can be expressed as $\widehat{F} = \widehat{F}\,(\mathbf{b}, \mathbf{c}, \mathbf{U}(\mathbf{b}, \mathbf{c}))$, to denote the dependency of $\widehat{F}$ on the flow variables $\mathbf{U}$, the design variables $\mathbf{b}$ (b_l, $l \in [1, N]$)

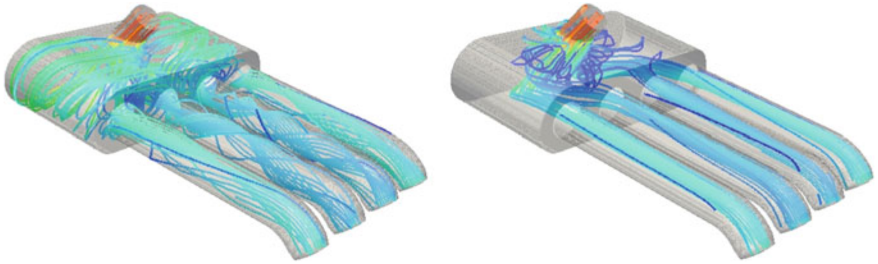

Fig. 7 Topology optimization of a plenum chamber targeting minimum fluid power losses (F), subject to a constraint requiring half of the plenum chamber volume to be flled by fluid. Primal velocity streamlines computed in the starting (*left*) and optimized geometries (*right*). Streamlines are colored based on the the primal velocity magnitude. A 29 % reduction in F was achieved after a 12 hour computation on 40 cores of 5 Intel Xeon E5620 CPUs (2.40 GHz)

which parameterize the aerodynamic shape and the uncertain environmental variables $\mathbf{c}$. A probability density function $g(\mathbf{c})$ can be associated with $\mathbf{c}$. In the so-called Second-Order Second-Moment (SOSM) approach, $\widehat{F}$ combines the mean value μ_F and variance ${\sigma_F}^2$ of F

$$\mu_F(\mathbf{b},\mathbf{c}) = \int F g(\mathbf{c}) d\mathbf{c} \simeq F + \frac{1}{2}\left[\frac{\delta^2 F}{\delta c_i^2}\right]_{\bar{\mathbf{c}}} \sigma_i^2 \tag{20}$$

$${\sigma_F}^2(\mathbf{b},\mathbf{c}) = \int (F-\mu_F)^2 g(\mathbf{c}) d\mathbf{c} \simeq \left[\frac{\delta F}{\delta c_i}\right]_{\bar{\mathbf{c}}}^2 \sigma_i^2 + \frac{1}{2}\left[\frac{\delta^2 F}{\delta c_i \delta c_j}\right]_{\bar{\mathbf{c}}}^2 \sigma_i^2 \sigma_j^2 \tag{21}$$

where the gradients are evaluated at the mean values $\bar{\mathbf{c}}$ of the environmental variables.

Based on the previous definitions, in robust design, $\widehat{F}$ becomes

$$\widehat{F}(\mathbf{b},\mathbf{c}) = w\mu_F + (1-w)\sigma_F^2 \tag{22}$$

where w is a user-defined weight. To compute $\widehat{F}$, efficient and accurate methods for first- and second-order derivatives of F w.r.t. the environmental variables are needed.

3.1 Computation of Second-Order Moments

In aerodynamic optimization, the computation of the Hessian of F, subject to the constraint of satisfying the flow equations, can be conducted in at least four different ways. All of them can be set up in either discrete or continuous form [21–23]. The presentation is always much more synoptic in the discrete sense. In this case, the first-order variation rate of F w.r.t. c_i, $i=1,\ldots,M$ is given by

$$\frac{dF}{dc_i} = \frac{\partial F}{\partial c_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{dc_i} \tag{23}$$

whereas the sensitivities of the discretized residuals R_m of the flow equations w.r.t. c_i are given by

$$\frac{dR_m}{dc_i} = \frac{\partial R_m}{\partial c_i} + \frac{\partial R_m}{\partial U_k}\frac{dU_k}{dc_i} = 0 \tag{24}$$

where U_k are the discretized field of the flow variables. Solving Eq. 24 for $\frac{dU_k}{dc_i}$, at the cost of M equivalent flow solutions (EFS; this is approximately the cost of solving the primal equations) and, then, computing $\frac{dF}{dc_i}$ from Eq. 23 is straightforward but costly and will be referred to as Direct Differentiation (DD). Since the cost to compute the gradient of F using DD scales with M, the Adjoint Variable (AV) method can be used instead. The adjoint equations to be solved for the adjoint variables Ψ_m are

$$R_k^{\Psi} = \frac{\partial F}{\partial U_k} + \Psi_m \frac{\partial R_m}{\partial U_k} = 0 \tag{25}$$

and $\frac{dF}{dc_i}$ are computed as

$$\frac{dF}{dc_i} = \frac{\partial F}{\partial c_i} + \Psi_m \frac{\partial R_m}{\partial c_i} \tag{26}$$

To compute the Hessian of F, starting from Eq. 23, the so-called DD-DD approach is set up, so that

$$\begin{aligned}\frac{d^2F}{dc_i dc_j} = {} & \frac{\partial^2 F}{\partial c_i \partial c_j} + \frac{\partial^2 F}{\partial c_i \partial U_k}\frac{dU_k}{dc_j} + \frac{\partial^2 F}{\partial U_k \partial c_j}\frac{dU_k}{dc_i} \\ & + \frac{\partial^2 F}{\partial U_k \partial U_m}\frac{dU_k}{dc_i}\frac{dU_m}{dc_j} + \frac{\partial F}{\partial U_k}\frac{d^2 U_k}{dc_i dc_j}\end{aligned} \tag{27}$$

where $\frac{d^2U_k}{dc_i dc_j}$ is computed by first solving the following DD equations

$$\begin{aligned}\frac{d^2R_n}{dc_i dc_j} = {} & \frac{\partial^2 R_n}{\partial c_i \partial c_j} + \frac{\partial^2 R_n}{\partial c_i \partial U_k}\frac{dU_k}{dc_j} + \frac{\partial^2 R_n}{\partial U_k \partial c_j}\frac{dU_k}{dc_i} \\ & + \frac{\partial^2 R_n}{\partial U_k \partial U_m}\frac{dU_k}{dc_i}\frac{dU_m}{dc_j} + \frac{\partial R_n}{\partial U_k}\frac{d^2 U_k}{dc_i dc_j} = 0\end{aligned} \tag{28}$$

Note that $\frac{dU_k}{dc_i}$ are already known from the solution of Eq. 24.

The DD-DD approach requires upon the computation of $\frac{dU_k}{dc_i}$ and $\frac{d^2U_k}{dc_i dc_j}$ and its CPU cost is $M+\frac{M(M+1)}{2}$ EFS in total (excluding the cost of solving the flow equations). So, the overall CPU cost scales with M^2 and becomes too expensive for use in real–world optimization.

Two less expensive approaches to compute the Hessian of F are the AV-DD (AV for the gradient and DD for the Hessian) and AV-AV ones. As shown in [23], both cost an many as $2M+1$ EFS. It can be shown that the fourth alternative way, i.e. the DD-AV approach (DD for the gradient and AV for the Hessian), is the most efficient one to compute the Hessian matrix. In DD-AV, the Hessian matrix is computed by

$$\begin{aligned}\frac{d^2F}{dc_i dc_j} = & \frac{\partial^2 F}{\partial c_i \partial c_j} + \Psi_n \frac{\partial^2 R_n}{\partial c_i \partial c_j} + \left(\frac{\partial^2 F}{\partial U_k \partial U_m} + \Psi_n \frac{\partial^2 R_n}{\partial U_k \partial U_m} \right) \frac{dU_k}{dc_i} \frac{dU_m}{dc_j} \\ & + \left(\frac{\partial^2 F}{\partial c_i \partial U_k} + \Psi_n \frac{\partial^2 R_n}{\partial c_i \partial U_k} \right) \frac{dU_k}{dc_j} + \left(\frac{\partial^2 F}{\partial U_k \partial c_j} + \Psi_n \frac{\partial^2 R_n}{\partial U_k \partial c_j} \right) \frac{dU_k}{dc_i} \end{aligned} \tag{29}$$

where $\frac{dU_k}{dc_i}$ result from Eq. 24 and Ψ_m is computed by solving the adjoint equation, Eq. 25 (same as before). The total CPU cost of DD-AV is equal to $M+1$ EFS being, thus, the most economical approach.

3.2 Robust Shape Optimization Using Third-Order Sensitivities

If the problem of minimizing the combination of the two first statistical moments is to be solved using a stochastic method such as an evolutionary algorithm, the methods presented in Sect. 3.1 serve to provide μ_F and ${\sigma_F}^2$; no other derivation is required. However, if a gradient-based method is used, the gradient of $\widehat{F}$ Eq. 22 must be differentiated w.r.t. b_q,

$$\frac{\delta \widehat{F}}{\delta b_q} = w\left(\frac{\delta F}{\delta b_q} + \frac{1}{2} \frac{\delta^3 F}{\delta c_i^2 \delta b_q} \sigma_i^2 \right) + (1-w) \frac{2 \frac{\delta F}{\delta c_i} \frac{\delta^2 F}{\delta c_i \delta b_q} \sigma_i^2 + \frac{\delta^2 F}{\delta c_i \delta c_j} \frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q} \sigma_i^2 \sigma_j^2}{2 \sqrt{\left[\frac{\delta F}{\delta c_i} \right]^2 \sigma_i^2 + \frac{1}{2} \left[\frac{\delta^2 F}{\delta c_i \delta c_j} \right]^2 \sigma_i^2 \sigma_j^2}} \tag{30}$$

From Eq. 30, the computation of $\frac{\delta \widehat{F}}{\delta b_q}$ involves up to third-order mixed sensitivities w.r.t. c_i and b_q, such as $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$. The computation of the second and third-order sensitivity derivatives is presented in detail in [24–26]. For instance, in the discrete sense, the highest-order derivative $\frac{d^2 F}{dc_i dc_j db_q}$ is computed using the expression

$$\begin{aligned}\frac{d^3F}{dc_i dc_j db_l} &= \frac{\partial^3 F}{\partial c_i \partial c_j \partial b_l} + \frac{\partial^3 F}{\partial c_i \partial b_l \partial U_k} \cdot \frac{dU_k}{dc_j} + \frac{\partial^3 F}{\partial c_j \partial b_l \partial U_k} \cdot \frac{dU_k}{dc_i} \\ &+ \frac{\partial^3 F}{\partial b_l \partial U_k \partial U_m} \cdot \frac{dU_k}{dc_i} \cdot \frac{dU_m}{dc_j} + \frac{\partial^2 F}{\partial b_l \partial U_k} \cdot \frac{d^2 U_k}{dc_i dc_j} \\ &+ \mathscr{K}_n^{i,j} \frac{\partial R_n}{\partial b_l} + \mathscr{L}_n^j \left(\frac{\partial^2 R_n}{\partial c_i \partial b_l} + \frac{\partial^2 R_n}{\partial b_l \partial U_k} \cdot \frac{dU_k}{dc_i} \right) \\ &+ \mathscr{M}_n^i \left(\frac{\partial^2 R_n}{\partial c_j \partial b_l} + \frac{\partial^2 R_n}{\partial b_l \partial U_k} \cdot \frac{dU_k}{dc_j} \right) \\ &+ \mathscr{N}_n \left(\frac{\partial^3 R_n}{\partial c_i \partial c_j \partial b_l} + \frac{\partial^3 R_n}{\partial c_i \partial b_l \partial U_k} \cdot \frac{dU_k}{dc_j} + \frac{\partial^3 R_n}{\partial c_j \partial b_l \partial U_k} \cdot \frac{dU_k}{dc_i} \right. \\ &+ \left. \frac{\partial^3 R_n}{\partial b_l \partial U_k \partial U_m} \cdot \frac{dU_k}{dc_i} \cdot \frac{dU_m}{dc_j} + \frac{\partial^2 R_n}{\partial b_l \partial U_k} \cdot \frac{d^2 U_k}{dc_i dc_j} \right) \end{aligned} \quad (31)$$

where the adjoint variables $\mathscr{N}_n$ satisfy the equation

$$\frac{\partial F}{\partial U_k} + \mathscr{N}_n \frac{\partial R_n}{\partial U_k} = 0 \quad (32)$$

and the equations to be solved for $\mathscr{L}_n^j$ and $\mathscr{M}_n^i$ can be found in [24].

According to Eq. 31, $\frac{dU_k}{dc_j}$ and $\frac{d^2U_k}{dc_i dc_j}$ must be available. These are computed by twice applying the DD technique, practically by solving Eqs. 24 and 28. This is the costly part of the algorithm, since it costs as many as $M + \frac{M(M+1)}{2}$ EFS. However, in the majority of cases, the environmental variables are much less than the design ones, $M \ll N$. The computation of $\mathscr{K}_N^{i,j}$, $i, j \in [1, M]$ costs $M + \frac{M(M+1)}{2}$ EFS and that of $\mathscr{M}_n^i$, $i \in [1, M]$ M EFS. The overall cost per optimization cycle becomes $M^2 + 3M + 2$ EFS; where the last two EFS correspond to the solution of the primal and adjoint (i.e. Eq. 32 for $\mathscr{N}$) equations. The aforementioned technique, which is referred to as DD$_c$-DD$_c$-AV$_b$ (subscripts denote whether the differentiation is made w.r.t. **c** or **b**) has the minimum computational cost, provided that $M < N$.

3.3 Robust Design Using Continuous Adjoint

This section aims at briefly demonstrating that the material presented in Sects. 3.1 and 3.2 can also be based on the continuous, rather that the discrete, adjoint. Without loss in generality, this will be demonstrated in an inverse design problem, by assuming inviscid flow of a compressible fluid.

The steady–state 2D Euler equations of a compressible fluid are given by

$$\frac{\partial f_{nk}}{\partial x_k} = 0 \quad (33)$$

where $k = 1, 2$ (for the Cartesian components) and $n = 1, \dots, 4$ (four equations in 2D). The inviscid fluxes f_{nk} are

$$[f_{1k}, f_{2k}, f_{3k}, f_{4k}] = [\rho \upsilon_k, \rho \upsilon_k \upsilon_1 + p\delta_{k1}, \rho \upsilon_k \upsilon_2 + p\delta_{k2}, \upsilon_k(E + p)]$$

where ρ, p, υ_k and E stand for the density, pressure, Cartesian velocity components and total energy per unit volume, respectively. The array of conservative flow variables is $[U_1, U_2, U_3, U_4] = [\rho, \rho\upsilon_1, \rho\upsilon_2, E]$. For the inverse design problem, the objective function is

$$F = \frac{1}{2} \int_{S_w} (p - p_{tar})^2 dS \tag{34}$$

where p_{tar} is the target pressure distribution along the solid wall.

In this problem, it is straightforward to derive the continuous adjoint PDEs which take the form

$$-A_{nmk} \frac{\partial \mathscr{N}_n}{\partial x_k} = 0, \quad m = 1, \dots, 4 \tag{35}$$

where $A_{nmk} = \frac{\partial f_{nk}}{\partial U_m}$ $(n = 1, 4, \quad m = 1, 4, \quad k = 1, 2)$ are the Jacobian matrices of the inviscid fluxes. Eq. 35 is equivalent to Eq. 32 in the continuous sense, considering that, in continuous adjoint, $\frac{\partial F}{\partial U_k}$ appears in the application of boundary conditions.

In the continuous approach, the DD$_c$-DD$_c$ approach can also be formulated by setting up, discretizing and numerically solving PDEs for $\frac{\delta U_m}{\delta c_i}$ and $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$. The M systems of PDEs, to be solved for $\frac{\delta U_m}{\delta c_i}$, result from the first-order sensitivities of the Euler equations w.r.t. the environmental variables,

$$\frac{\partial}{\partial x_k} \left(A_{nmk} \frac{\delta U_m}{\delta c_i} \right) = 0 \ , \quad n = 1, \dots, 4 \quad i = 1, \dots, M \tag{36}$$

along with appropriate boundary conditions. For the $\frac{M(M+1)}{2}$ systems of equations, to be solved for $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$, $i = 1, M$, $j = 1, M$, Eq. 36 are differentiated once more to give

$$\frac{\partial}{\partial x_k} \left(A_{nmk} \frac{\delta^2 U_m}{\delta c_i \delta c_j} + \frac{\delta A_{nmk}}{\delta c_j} \frac{\delta U_m}{\delta c_i} \right) = 0 \ , \quad n = 1, \dots, 4; \quad i, j = 1, \dots, M \tag{37}$$

With known $\frac{\delta U_m}{\delta c_i}$ and $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$ fields, the first- and second-order sensitivities of F w.r.t. the environmental variables are given by

$$\frac{\delta F}{\delta c_i} = \int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta c_i} dS \tag{38}$$

and

$$\frac{\delta^2 F}{\delta c_i \delta c_j} = \int_{S_w} \left[\frac{\delta p}{\delta c_i} \frac{\delta p}{\delta c_j} + (p - p_{tar}) \frac{\delta^2 p}{\delta c_i \delta c_j} \right] dS \tag{39}$$

where $\frac{\delta p}{\delta c_i}$ and $\frac{\delta^2 p}{\delta c_i \delta c_j}$ can be expressed in terms of the corresponding derivatives of the conservative flow variables U_m.

The highest-order mixed derivatives are computed through the solution of additional adjoint PDEs, similar to the corresponding discrete equations. For instance, the third-order mixed sensitivity derivatives of F, required in Eq. 30, are given by

$$\begin{aligned}
\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q} = & \int_{S_w} \frac{\delta p}{\delta c_i} \frac{\delta p}{\delta c_j} \frac{\delta (dS)}{\delta b_q} + \int_{S_w} (p - p_{tar}) \frac{\delta^2 p}{\delta c_i \delta c_j} \frac{\delta (dS)}{\delta b_q} \\
& + \int_{S_w} \left(\mathscr{K}^{i,j}_{k+1} p - \mathscr{K}^{i,j}_{n} f_{nk} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} \\
& + \int_{S_w} \left(\mathscr{L}^{i}_{k+1} \frac{\delta p}{\delta c_j} - \mathscr{L}^{i}_{n} \frac{\delta f_{nk}}{\delta c_j} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} \\
& + \int_{S_w} \left(\mathscr{L}^{j}_{k+1} \frac{\delta p}{\delta c_i} - \mathscr{L}^{j}_{n} \frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} \\
& + \int_{S_w} \left(\mathscr{N}_{k+1} \frac{\delta^2 p}{\delta c_i \delta c_j} - \mathscr{N}_{n} \frac{\delta^2 f_{nk}}{\delta c_i \delta c_j} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} \\
& - \int_{S_w} \mathscr{K}^{i,j}_{n} \frac{\partial f_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_q} n_k dS - \int_{S_w} \mathscr{L}^{i}_{n} \frac{\partial}{\partial x_l} \left(\frac{\delta f_{nk}}{\delta c_j} \right) \frac{\delta x_l}{\delta b_q} n_k dS \\
& - \int_{S_w} \mathscr{L}^{j}_{n} \frac{\partial}{\partial x_l} \left(\frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta x_l}{\delta b_q} n_k dS - \int_{S_w} \mathscr{N}_{n} \frac{\partial}{\partial x_l} \left(\frac{\delta^2 f_{nk}}{\delta c_i \delta c_j} \right) \frac{\delta x_l}{\delta b_q} n_k dS \\
& + \int_{S_w} \mathscr{L}^{j}_{n} \frac{\partial}{\partial x_k} \left(\frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta x_l}{\delta b_q} n_l dS + \int_{S_w} \mathscr{L}^{i}_{n} \frac{\partial}{\partial x_k} \left(\frac{\delta f_{nk}}{\delta c_j} \right) \frac{\delta x_l}{\delta b_q} n_l dS \\
& + \int_{S_w} \mathscr{K}^{i,j}_{n} \frac{\partial f_{nk}}{\partial x_k} \frac{\delta x_l}{\delta b_q} n_l dS + \int_{S_w} \mathscr{N}_{n} \frac{\partial}{\partial x_k} \left(\frac{\delta^2 f_{nk}}{\delta c_i \delta c_j} \right) \frac{\delta x_l}{\delta b_q} n_l dS
\end{aligned} \tag{40}$$

where the additional adjoint fields $\mathscr{L}^i_n = \frac{\delta \mathscr{N}_n}{\delta c_i}$ and $\mathscr{K}^{i,j}_n = \frac{\delta \mathscr{L}^i_n}{\delta c_j}$ are computed by solving the adjoint equations

$$-A_{nmk} \frac{\partial \mathscr{L}^i_n}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_i} \frac{\partial \mathscr{N}_n}{\partial x_k} = 0 \tag{41}$$

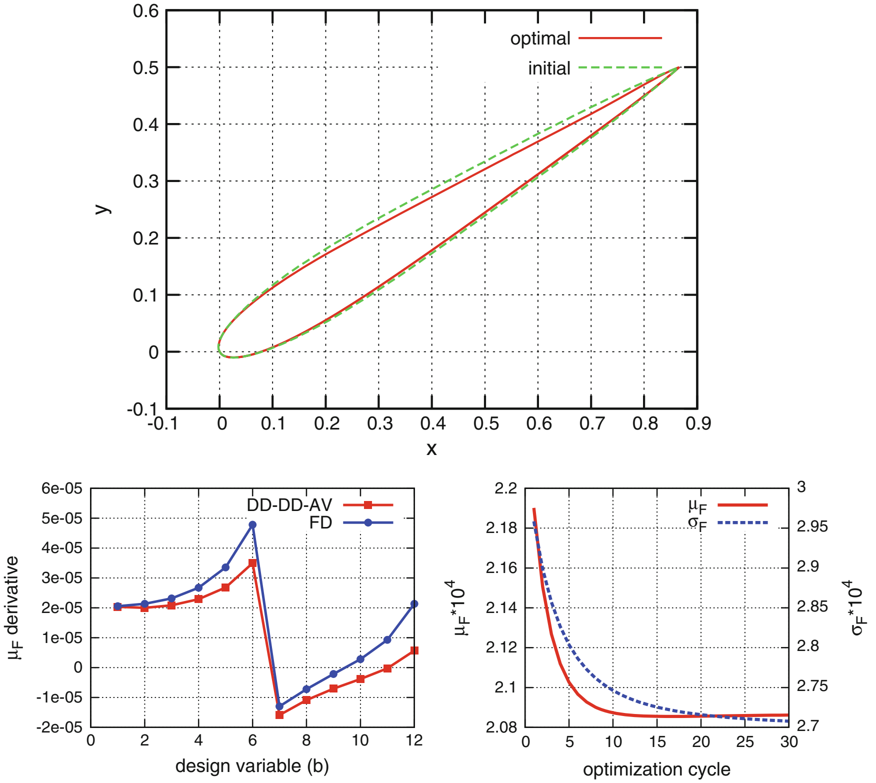

Fig. 8 Robust inverse design of a 2D cascade for a given target pressure distribution. *Top* Optimal cascade geometry compared to the initial one; the initial airfoil is symmetric. *Bottom-left* Comparison of sensitivities $\frac{\delta\mu_F}{\delta b_q}$ (b_q are the coordinates of the Bézier control points) computed using the DD$_c$-DD$_c$-AV$_b$ method and FD. The proposed method matches the third derivatives captured by FD. *Bottom-right* Convergence of the mean value and standard deviation of F using $w = 0.7$, see Eq. 22. From [26]

and

$$-A_{nmk}\frac{\partial \mathscr{K}_n^{i,j}}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_j}\frac{\partial \mathscr{L}_n^{i}}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_i}\frac{\partial \mathscr{L}_n^{j}}{\partial x_k} - \frac{\delta^2 A_{nmk}}{\delta c_i \delta c_j}\frac{\partial \mathscr{N}_n}{\partial x_k} = 0 \quad (42)$$

as explained in [25]. Similarities between the discrete and continuous variants of the DD$_c$-DD$_c$-AV$_b$ method can easily be identified.

An application of the robust design algorithm is illustrated in Fig. 8; it is related to the inverse design of a 2D cascade, [25, 26]. The airfoil shape controlling parameters are the design variables and the inlet/outlet flow conditions are the environmental ones.

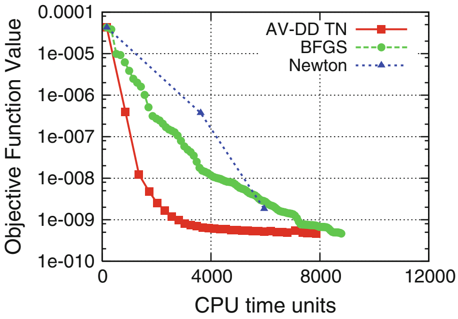

Fig. 9 Inverse design of a 2D airfoil cascade (42 design variablees) using the truncated Newton method: Comparison of the convergence rates of the AV-DD truncated Newton method (with 4 CG steps per cycle) with other second-order methods (BFGS and exact Newton). From [27]

3.4 Other Usage of the DD and AV Method

Apart from robust design applications, the developed methods for the computation of higher-order derivatives of F can also be used to support more efficient optimization methods, such as the (exact) Newton method. In such a case, however, the cost per optimization cycle depends on N and this may seriously hinder the use of such a method in industry. To cope with large scale optimization problems, the Newton equations can be solved through the CG method with truncation. By doing so, the Hessian matrix itself is not needed anymore, [27]. The adjoint approach followed by the DD of both the flow and adjoint equations (AV-DD) is the most efficient way to compute the product of the Hessian matrix with any vector required by the truncated Newton algorithm. The cost per Newton iteration scales linearly with the number of CG steps, rather than the much higher number of the design variables (if the Hessian itself was computed in the "exact" Newton method). The efficiency of the truncated Newton method is demonstrated in Fig. 9, in a problem with 42 design variables.

4 Industrial Applications

In Fig. 10, the application of the developed adjoint-based software to four industrial problems is presented. The first case deals with the blade optimization of a $3D$ peripheral compressor cascade in which the objective is the minimization of entropy losses within the flow passage, [28]. The second case is concerned with the shape optimization of a Francis turbine runner targeting cavitation suppression, [29], the third one with an air-conditioning duct targeting minimum total pressure losses and the last one with the shape optimization of a Volkswagen concept car, targeting minimum drag force, [30]. More comments can be found in the caption [31].

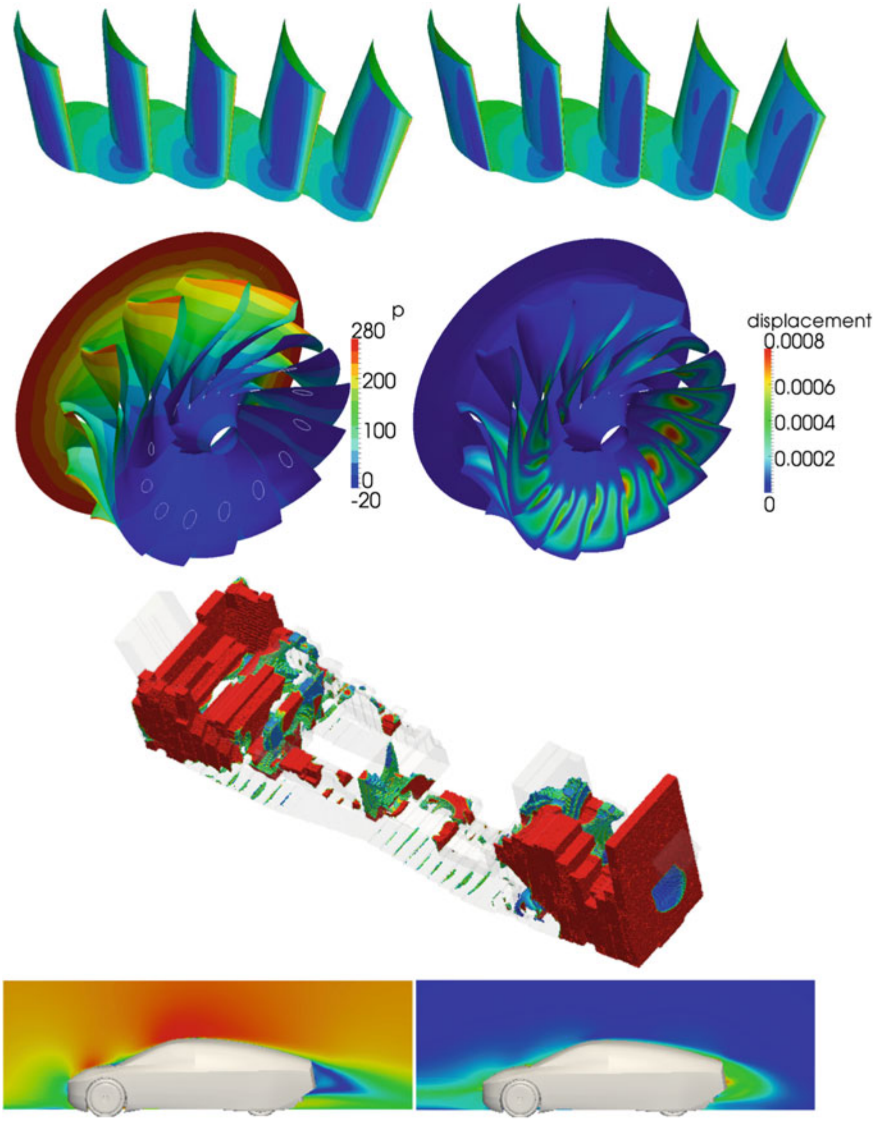

Fig. 10 *Row 1* Shape optimization of a 3D peripheral compressor cascade, targeting minimum entropy generation rate within the flow passage with constraints on the blade thickness. Pressure distributions over the initial (*left*) and optimal (*right*) blade geometries; from [28]. *Row 2* Optimization of a Francis runner blade for cavitation suppression. Pressure distribution over the initial blading (*left*); areas within the white isolines are considered to be cavitated; surface deformation magnitude over the optimized blading (*right*), after eliminating cavitation; from [29]. *Row 3* Topology optimization of an air-conditioning duct, used in a passenger car, targeting minimum total pressure losses. Porosity field at the last optimization cycle. The topology optimization led to the solidification of areas (in *red*) where, in the starting geometry, intense flow recirculation appeared. *Row 4* Optimization of the VW L1 concept car targeting minimum drag force. Primal velocity field calculated using the RANS equations along with the low-Re Spalart–Allmaras model (*left*) and adjoint velocity field calculated by using the "turbulent" adjoint method (*right*); from [30]

Acknowledgments Parts of the research related to the exact differentiation of the turbulence models were funded by Volkswagen AG (Group Research, K-EFFG/V, Wolfsburg, Germany) and Icon Technology and Process Consulting Ltd. The support of Volkswagen AG for the development of the adjoint method for flow control applications is also acknowledged.
Research related to topology optimization was partially supported by a Basic Research Project funded by the National Technical University of Athens.
The authors would like to acknowledge contributions to topology optimization from Dr. E Kontoleontos and to "turbulent" adjoint by Dr. A. Zymaris. They are also indebted to Dr. Carsten Othmer, Volkswagen AG (Group Research, K-EFFG/V), for his support, some interesting discussions on the continuous adjoint method and his contributions in several parts of this work.

References

1. P. Spalart and S. Allmaras: A one-equation turbulence model for aerodynamic flows. AIAA Paper, 04(39), 1992
2. Launder BE, Sharma BI (1974) Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. Lett Heat Mass Transf 1:131–137
3. Lee BJ, Kim C (2007) Automated design methodology of turbulent internal flow using discrete adjoint formulation. Aerosp Sci Technol 11:163–173
4. Mavriplis DJ (2007) Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. AIAA J 45:740–750
5. Pironneau O (1974) On optimum design in fluid mechanics. J Fluid Mech 64:97–110
6. Jameson A (1988) Aerodynamic design via control theory. J Sci Comput 3:233–260
7. Anderson WK, Venkatakrishnan V (1997) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA Paper 97–0643
8. Papadimitriou DI, Giannakoglou KC (2007) A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. J Comput Fluids 36:325–341
9. Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. Int J Numer Meth Fluids 58:861–877
10. Zymaris AS, Papadimitriou DI, Giannakoglou KC, Othmer C (2009) Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. Comput Fluids 38:1528–1538
11. Bueno-Orovio A, Castro C, Palacios F, Zuazua E (2012) Continuous adjoint approach for the SpalartAllmaras model in aerodynamic optimization. AIAA J 50:631–646
12. Zymaris AS, Papadimitriou DI, Giannakoglou KC, Othmer C (2010) Adjoint wall functions: a new concept for use in aerodynamic shape optimization. J Comput Phys 229:5228–5245
13. Papoutsis-Kiachagias EM, Zymaris AS, Kavvadias IS, Papadimitriou DI, Giannakoglou KC (2014) The continuous adjoint approach to the k-ε turbulence model for shape optimization and optimal active control of turbulent flows. Eng Optim. doi:10.1080/0305215X.2014.892595
14. Asouti VG, Trompoukis XS, Kampolis IC, Giannakoglou C (2011) Unsteady CFD computations using vertex-centered finite volumes for unstructured grids on graphics processing units. Int J Numer Meth Fluids 67(2):232–246
15. Trompoukis XS, Tsiakas KT, Ghavami Nejad M, Asouti VG Giannakoglou KC (2014) The continuous adjoint method on graphics processing units for compressible flows. In: OPT-i, international conference on engineering and applied sciences optimization, Kos Island, Greece, 4–6 June 2014
16. Kavvadias IS, Papoutsis-Kiachagias EM, Dimitrakopoulos G, Giannakoglou KC (2014) The continuous adjoint approach to the kω SST turbulence model with applications in shape optimization. Eng Optim. doi:10.1080/0305215X.2014.979816

17. Kavvadias IS, Karpouzas GK, Papoutsis-Kiachagias EM, Papadimitriou DI, Giannakoglou KC (2013) Optimal flow control and topology optimization using the continuous adjoint method in unsteady flows. In: EUROGEN conference 2013, Las Palmas de Gran Canaria, Spain
18. Griewank A, Walther A (2000) Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Trans Math Softw 26(1):19–45
19. Vezyris CK, Kavvadias IS, Papoutsis-Kiachagias EM, Giannakoglou KC (2014) Unsteady continuous adjoint method using POD for jet-based flow control. In: ECCOMAS conference 2014, Barcelona, Spain
20. Papoutsis-Kiachagias EM, Kontoleontos EA, Zymaris AS, Papadimitriou DI, Giannakoglou KC (2011) Constrained topology optimization for laminar and turbulent flows, including heat transfer. In: EUROGEN Conference 2011, Capua, Italy
21. Papadimitriou DI, Giannakoglou KC (2008) Computation of the Hessian matrix in aerodynamic inverse design using continuous adjoint formulations. Comput Fluids 37:1029–1039
22. Papadimitriou DI, Giannakoglou KC (2008) The continuous direct-adjoint approach for second-order sensitivities in viscous aerodynamic inverse design problems. Comput Fluids 38:1539–1548
23. Papadimitriou DI, Giannakoglou KC (2008) Aerodynamic shape optimization using adjoint and direct approaches. Arch Comput Meth Eng 15:447–488
24. Papoutsis-Kiachagias EM, Papadimitriou DI, Giannakoglou KC (2012) Robust design in aerodynamics using third-order sensitivity analysis based on discrete adjoint. Application to quasi-1D flows. Int J Numer Meth Fluids 69:691–709
25. Papadimitriou DI, Giannakoglou KC (2013) Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint. Int J Numer Meth Fluids 71:652–670
26. Papoutsis-Kiachagias EM, Papadimitriou DI, Giannakoglou KC (2011) On the optimal use of adjoint methods in aerodynamic robust design problems. In: CFD and OPTIMIZATION, (2011) ECCOMAS thematic conference. Antalya, Turkey 2011
27. Papadimitriou DI, Giannakoglou KC (2012) Aerodynamic design using the truncated Newton algorithm and the continuous adjoint approach. Int J NumerMeth Fluids 68:724–739
28. Papadimitriou DI, Giannakoglou KC (2006) Compressor blade optimization using a continuous adjoint formulation. In: ASME TURBO EXPO, GT2006/90466, Barcelona, Spain, 8–11 May 2006
29. Papoutsis-Kiachagias EM, Kyriacou SA, Giannakoglou KC (2014) The continuous adjoint method for the design of hydraulic turbomachines. Comput Meth Appl Mech Eng 276:621–639
30. Othmer C, Papoutsis-Kiachagias E, Haliskos K (2011) CFD Optimization via sensitivity-based shape morphing. In: 4th ANSA & μETA international conference, Thessaloniki, Greece, 2011
31. Menter FR (1994) Two-equation eddy-viscosity turbulence models for engineering applications. AIAA 32(8):269–289