



Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications

E. M. Papoutsis-Kiachagias · K. C. Giannakoglou

Received: 24 November 2014 / Accepted: 5 December 2014 / Published online: 16 December 2014
© CIMNE, Barcelona, Spain 2014

Abstract This article focuses on the formulation, validation and application of the continuous adjoint method for turbulent flows in aero/hydrodynamic optimization. Though discrete adjoint has been extensively used in the past to compute objective function gradients with respect to (w.r.t.) the design variables under turbulent flow conditions, the development of the continuous adjoint variant for these flows is not widespread in the literature, hindering, to an extend, the computation of exact sensitivity derivatives. The article initially presents a general formulation of the continuous adjoint method for incompressible flows, under the commonly used assumption of “frozen turbulence”. Then, the necessary addenda are presented in order to deal with the differentiation of both low- and high-Reynolds (with wall functions) number turbulence models; the latter requires the introduction of the so-called “adjoint wall functions”. An approach to dealing with distance variations is also presented. The developed methods are initially validated in 2D cases and then applied to industrial shape and topology optimization problems, originating from the automotive and hydraulic turbomachinery industries.

1 Introduction

The fast-paced development of high performance computer systems along with the maturity of computational methods has led to the widespread use of Computational Fluid

Dynamics (CFD) in a significant range of engineering applications, concerning the flow analysis around aircrafts and automotive vehicles, simulations of flows through thermal and hydraulic turbomachines and so on. The availability of CFD codes and the cost benefits resulting from using computational analysis rather than conducting experiments, has given rise to an intense academic and industrial interest in the use of computational methods for the design/optimization of aero/hydrodynamic components. The corresponding scientific area is entitled CFD-based optimization.

According to control theory, adapted to the needs of CFD-based optimization, the geometric shape under consideration is controlled by a number of variables, referred to as the design or optimization variables. For instance, these could be the control point coordinates of Bézier–Bernstein polynomials parameterizing the shape of an airfoil or a duct. The quality of the shape under consideration is evaluated by computing a quantity (usually an integral), known as the objective function. For instance, this could be either the drag or lift coefficient of an airfoil. The objective function value depends on the values of the design variables and the computed fields of flow variables, obtained through the numerical solution of the flow equations. The flow equations can be seen as constraints imposed on the optimization problem; these constraints may be the flow equations for inviscid or viscous flows of incompressible or compressible fluids. In all cases examined in this article, the Navier–Stokes equations for viscous, incompressible flows are solved. The goal of the optimization method is to compute the values of the design variables that minimize the objective function in hand. Maximization problems can easily be reformulated to minimization ones and, then, solved using the proposed methods.

CFD-based optimization methods can be classified into two main categories, according to the strategy used to compute the optimal set of design variables: stochastic [15,

E. M. Papoutsis-Kiachagias · K. C. Giannakoglou (✉)
Parallel CFD and Optimization Unit, School of Mechanical Engineering, National Technical University of Athens, Athens, Greece
e-mail: kgianna@central.ntua.gr

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

39, 104, 130] and deterministic [21, 22, 60, 98, 105, 115, 155] optimization methods.

Evolutionary algorithms (EAs) are the most popular representatives of stochastic population-based optimization methods; they mimic natural evolution operations, such as crossover and mutation, and apply them to a set of candidate solutions, organised in generations, until a convergence criterion is met. The randomized way candidate solutions are spawn makes entrapment to local minima highly unlikely, since almost the entire design space is explored. This leads to the computation of global minima, provided that enough generations have been performed. Since the evolution operators do not interfere with the flow solver, EAs may be used with various solvers without requiring access to their source code. Furthermore, EAs can compute Pareto fronts of non-dominated solutions in multi-objective optimization (MOO) problems, with a single run. On the other hand, a great number of candidate solutions must be evaluated before acquiring the optimal one(s), leading to a high optimization turn-around time, especially when the evaluation software is time-consuming. Practically, the number of evaluations required increases with the number of the design variables. A number of remedies have been proposed to decrease the turn-around time of an EA-based optimization. These include the use of Parallel EAs with the concurrent evaluation and evolution of candidate solutions [5, 69], the use of asynchronous EAs which are suitable for heterogeneous multiprocessor platforms [6, 14], the use of metamodel-assisted EAs where a surrogate evaluation model is used as often as possible to avoid time-consuming evaluations [47, 55], the use of hierarchical EAs [69, 95], including the hybridization with other (usually deterministic) search methods [89, 117].

On the other hand, deterministic optimization algorithms start with a given geometry and improve it based on information related to the derivatives (sensitivity derivatives) of the objective function w.r.t. the design variables. Since the design variables are not updated randomly but based on the direction dictated by the computed sensitivities, each cycle of the algorithm produces an improved geometry (provided that the sensitivity derivative values have been computed correctly). This gives rise to a class of algorithms that require much less candidate evaluations than those required by EAs. On the other hand, since the values of the sensitivity derivatives are zeroed at local minima (in non-constrained optimization), it is not unlikely for a gradient-based method (GBM) to be trapped in a local optimum. In such a case, the designer gets an optimized rather than an optimal solution. In addition, entrapment to local minima can be avoided by changing the starting point and repeating the optimization algorithm, increasing though the overall optimization cost. In addition, deterministic optimization algorithms can not easily deal with MOO problems. Usually, different objectives are concatenated into a single one using weight factors and, then, a single objective

optimization (SOO) problem is solved. Different solutions can be obtained by choosing different sets of weight factors, only in convex optimization problems. Pareto-front computation techniques utilizing GBMs have also been proposed [144].

The efficiency of GBMs greatly depends on the method used to compute the necessary objective gradients. A straightforward way to do so is by using finite differences (FD). According to FD, each of the design variables, $b_n, n \in [1, N]$, is perturbed by an infinitesimally small quantity, ϵ , and the objective function is re-evaluated using the perturbed design. For a second-order FD scheme, the derivatives of an objective function F w.r.t. to $\mathbf{b} \in \mathbb{R}^N$ are given by

$$\frac{\delta F(\mathbf{b})}{\delta b_n} = \frac{F(b_1, \dots, b_n + \epsilon, \dots, b_N) - F(b_1, \dots, b_n - \epsilon, \dots, b_N)}{2\epsilon} \quad (1)$$

Though straightforward to implement, since it only requires the re-computation of the value of the objective function, the cost of the FD method scales linearly with the number of the design variables, N , making it practically infeasible for large scale optimization problems. In addition, the computed derivative values are sensitive to ϵ and, usually, a trial and error process must be employed to ensure that ϵ -independent derivatives have been computed. Choosing a “too” small ϵ value from the beginning is not a wise decision, since this can lead to round-off errors. When working with FD, the flow equations must be fully converged, since the sensitivity values are computed by subtracting two very close F values [70, 87].

The ϵ -dependence and the need to fully converge the flow equations can be circumvented by using the complex variable method [110, 112], according to which the sensitivities are computed by

$$\frac{\delta F(\mathbf{b})}{\delta b_n} = \frac{Im [F(b_1, \dots, b_n + i\epsilon, \dots, b_N)]}{\epsilon} \quad (2)$$

where $i = \sqrt{-1}$ and Im is the imaginary part of the complex function F . Since the sensitivity derivatives are no longer computed by subtracting two very close values, round-off errors cease to exist, circumventing the ϵ -dependency and the need to fully converge the flow equations. Nevertheless, the cost of the complex variable method stills scales linearly with N , making it impractical for large scale optimization problems. Furthermore, access to the source code of the flow equations solver is needed, which must be re-written to handle complex instead of real variables.

A third alternative for computing sensitivity derivatives is the direct differentiation (DD) method [159]. According to DD, the flow equations are differentiated w.r.t. to \mathbf{b} and N linear systems are solved for the derivatives of the flow

variables w.r.t. the design ones. Then, the sensitivities are expressed in terms of these fields. DD does not suffer from ϵ -dependency but is harder to implement than FD (since a new flow solver has to be written) and its cost still scales with N . Nevertheless, DD is an indispensable part of algorithms used to compute high-order sensitivity derivatives [122, 123].

Among the methods used to compute the sensitivity derivatives required by GBMs, the adjoint method [73, 134] has been receiving a lot of attention, due to the fact that the cost of computing the necessary derivatives is, practically, independent from the number of the design variables. This makes the method an excellent choice for large scale optimization problems. In order to achieve this independence, an augmented objective function is formulated by adding the sum of the residuals of the flow equations (also referred to as the primal or state equations), multiplied by the adjoint (or co-state or dual) variables to F . The above-mentioned procedure is equivalent to formulating a problem for the minimization of F subject to the (equality) constraint of satisfying the flow equations. The latter has been incorporated into the optimization problem by means of Lagrange multipliers (the adjoint variables). Then, after differentiating the augmented function, taking into consideration the derivative of the flow equations w.r.t. the design variables and re-arranging the resulting terms, the adjoint equations and their boundary conditions are formulated in order to make the cost of computing the sensitivity derivatives independent from N .

Depending on the sequence of the differentiation of the objective function and the discretization of the flow equations, adjoint methods can be classified into two categories. In the continuous adjoint approach [11, 73, 74, 78], the objective function is augmented using the flow equations in their continuous form (PDEs, prior to their discretization) and the resulting adjoint equations are in the form of PDEs (together with their boundary conditions), to be discretized and numerically solved in order to compute the adjoint variables. On the other hand, according to the discrete adjoint approach [9, 10, 48, 58], the objective function is augmented using the discretized residuals of the primal equations. After differentiating the augmented objective function and rearranging, the adjoint variables are computed by numerically solving the resulting, already discretized, system of adjoint equations. Further discussion about discrete and continuous adjoint methods follows in Sect. 1.

Though adjoint-based sensitivity derivatives are mainly computed for use in GBMs, they are also met in the form of sensitivity maps. A sensitivity map illustrates the variation of the objective function w.r.t. the normal displacement of the wall boundary nodes of an aero/hydrodynamic shape. Typical examples can be found in [119], where sensitivity maps are plotted over the surface of passenger cars. Sensitivity maps are used to highlight the areas where aerodynamic improvement has the greatest potential and are a valuable

tool for the designer, even if an automatic optimization loop is not applied.

2 Literature Survey and Article Outline

2.1 The Adjoint Method in Aerodynamic Shape Optimization

Optimization problems were initially tackled using control theory by Lions in 1971 [96]. According to control theory, an objective function quantifies the deviation from the optimal solution. Typical examples of objective functions in the area of aerodynamics are the deviation from a desired pressure distribution along an airfoil, a blade or a wing, or the drag/lift force exerted on it. In 1984, Pironneau [134] was the first to apply adjoint methods in the area of aerodynamics, for potential flows governed by elliptic differential equations. In 1988, Jameson [73] was the first to mathematically develop the adjoint method for the compressible Euler equations, even without presenting applications.

Applications were published later by Jameson's group in [136], concerning the optimization of 2D airfoils based on the potential flow equations. In [78], the optimization of 2D airfoils targeting predetermined pressure distributions and shock-wave induced drag minimization are presented, based on the compressible Euler equations. The method was extended to cover 3D flows and was applied to the optimization of wing-body airplane configurations [137], using the Euler equations. A complete aircraft optimization using the Euler equations is presented in [135, 138], by implementing the adjoint method on distributed memory parallel computers. Extensions of the method to cover problems governed by the compressible Navier–Stokes equations are presented in [75, 77], while a gradient accuracy study of this approach is discussed in [87]. All the above-mentioned papers from Jameson's group implemented the continuous adjoint method on structured grids and the sensitivities were computed using information from the entire computational domain, utilizing volume integrals.

A continuous adjoint approach for inviscid compressible and laminar incompressible flows, solved on unstructured grids was presented by Anderson in [11, 12], with the sensitivity derivatives expression depending only on surface integrals. There, the method was also compared to discrete adjoint and FD. It was concluded that the results of the continuous adjoint approach tend to coincide with those computed by discrete adjoint and FD as the mesh becomes finer, while deviations from them occur in coarser grids. In addition, an investigation was conducted about the importance of mesh sensitivities, in discrete adjoint. The outcome of this investigation states that mesh sensitivities, i.e. contributions from the displacement of the interior mesh nodes due to variations

in the design variables, are important in areas of geometric singularities, like the trailing edge of an airfoil. Later on, Jameson presented a continuous adjoint approach with sensitivity derivatives depending solely on surface integrals, initially for structured [76] meshes. The so-called reduction of the sensitivity formula allowed Jameson's group to present the optimization of a full aircraft using unstructured grids [79, 80].

Regarding the discrete adjoint approach, the first second-order solution of the adjoint to the 2D and 3D Euler equations was presented by Elliot and Peraire in [48]. Along with the parallelization of the software, a complete aircraft configuration was optimized in [49], targeting a predefined pressure distribution. In [57, 58], Giles and his co-workers proposed an iterative solution scheme for the discrete adjoint equations which guarantees duality between the linearized primal equations and the adjoint equations. As a result, the same objective function value is computed using the two sets of primal and adjoint variables in each iteration, meaning that the primal and adjoint equations have the same asymptotic rate of convergence. This is a very useful conclusion since it guarantees that the adjoint equations will converge, as long as the primal problem has converged as well. Due to its practical impact, the method has then been adopted by many researchers [45, 103, 113]. However, due to flow phenomena such as vortex shedding, it is very common for steady state flow solvers to converge up to a point and, then, oscillate around a solution for complex, 3D turbulent flows. Due to the duality scheme proposed in [57, 58], these oscillations of the primal problem lead to the divergence of the adjoint equations. To circumvent this problem, in [32, 33], Campobasso and Giles proposed to wrap the linear adjoint solver with a Krylov subspace method, as a stabilizer. The GMRES [142] and RPM [83, 145] Krylov subspace methods were proposed to identify and specially treat the diverging modes, stabilizing thus the adjoint simulations. The method was then adopted by other groups as well [46, 102]. Discrete adjoint codes can be implemented by either differentiating the discretized primal residuals "by hand" or by using Algorithmic Differentiation (AD) [38, 61, 67]. According to AD, a specialized software (AD tool) "automatically" generates the adjoint to the primal code. This can facilitate the implementation of discrete adjoint codes, especially in cases where the primal code includes complicated functions like those of turbulence models or high-order convection schemes. Even though AD can speed up the implementation of discrete adjoint codes, the resulting software tends to have greater memory requirements in comparison with codes that are "hand-differentiated". Popular AD tools are TAPENADE [128], ADIFOR [23], OpenAD [160], ADOL-C [62], etc. Though ADOL-C can be used along with programs written in C++, many of the object-oriented features are not supported, hindering, to an extent, the application of AD to

modern CFD solvers. A possible remedy to this problem is the operator overloading approach [143].

The comparison of the continuous and discrete adjoint approaches has been presented in [59, 106, 107, 131]. While in the early years of the application of adjoint methods to aerodynamic optimization, a dispute about which method is more appropriate took place, it is nowadays a general belief that both adjoint variants can provide sufficiently accurate derivatives for use in practical optimization problems. Nevertheless, the following observations can be made: (a) since the discrete adjoint equations take into consideration the primal discretization schemes, the sensitivities computed using discrete adjoint tend to match the reference values (computed either with FD or the method of complex variables) more accurately than those computed by continuous adjoint, even for coarse meshes. However, the difference diminishes as the mesh size increases and does not seem to greatly affect the convergence of the optimization algorithm [106], (b) the implementation of continuous adjoint is easier, since the discretization schemes applied to the adjoint PDEs do not have to match those used for the primal ones. On the other hand, the discrete adjoint approach has to follow the primal discretization schemes in order to compute accurate sensitivities, a process that can become cumbersome to implement, especially for high-order schemes. Nevertheless, it is beneficial for the continuous adjoint approach to replicate the discretization schemes used by discrete adjoint. Such a process is described in [56, 154], for laminar, incompressible flows, (c) the continuous adjoint variant is less expensive in terms of CPU cost and memory requirements per iteration [59, 106, 131], (d) while the initial continuous adjoint formulations [75, 77, 78, 87, 135–138], resulted to a gradient expression that included both surface and volume integrals, modifications of the method allowed the computation of sensitivities, based exclusively on surface integrals [11, 121]; this reduces the cost and complexity of the continuous adjoint approach. A study regarding the accuracy of each method is presented in [97] for inviscid compressible flows. There, it is concluded that the surface-based method may compute sensitivities of the wrong sign in areas with geometrical singularities (such as the trailing edge of an airfoil), an issue also partially discussed in [11]. This behaviour is not observed in laminar flows [97]. The discrete adjoint gradient expression also includes the mesh sensitivities. However, the need to compute mesh sensitivities in discrete adjoint can be circumvented by formulating and solving the adjoint to the mesh deformation equation [72, 102, 114], (e) the physical understanding of the adjoint system is clearer using the continuous adjoint approach, since closed-form expressions exist for the field adjoint equations, their boundary conditions and the sensitivity derivatives expression. This way, the impact of each term can easily be quantified. If some terms are identified to have small or no impact at all, their

computation can be avoided, further simplifying the method [173].

Concerning the adjoint to turbulent flows simulated by PDE models, the first discrete adjoint method was published as early as in 1997 [9], based on the “hand-differentiation” of the Spalart–Allmaras model for compressible flows, see also [10]. Latter on [111], the impact of neglecting the differentiation of the turbulent viscosity (the so-called “frozen turbulence” assumption) was investigated, leading to the conclusion that, depending on the problem, the computed sensitivities can differ from FD, even in the computed sign. However, the impact of these sensitivity discrepancies on the optimization process was not quantified at the time. Discrete adjoint codes were developed by other research groups as well, regarding the differentiation of the one-equation Spalart–Allmaras model [27, 109], two-equation models like the Wilcox $k - \omega$ [99, 132], $k - \omega$ SST [84] and the $k - \epsilon$ [84, 85], ones. A detailed investigation concerning the impact of various approximations made during the development of a discrete adjoint code, including that of the “frozen turbulence” assumption, is presented in [45]. There, for the Spalart–Allmaras model, it is observed that, depending on the problem, the “frozen turbulence” assumption may lead to the computation of sensitivities of wrong magnitude and in some, more rare, cases even wrong sign. The impact of these inexact sensitivities in the optimization of 2D airfoils is investigated and it is shown that when the steepest descent and conjugate gradient methods are used to update the design variables, the final outcome of the optimization is not affected much by gradient inaccuracies; nevertheless, the final result might be obtained after more optimization cycles than those required by an adjoint method that takes into consideration the adjoint to the turbulence model. On the contrary, more sophisticated optimization methods like BFGS, tend to get more affected by inexact sensitivities. Similar conclusions can be drawn from [85, 86], for the $k - \omega$ SST, $k - \omega$ and $k - \epsilon$ models, especially in the presence of strong shock waves. However, according to Marta and Shankaran [99], very small differences in the sensitivities computed with or without the “frozen turbulence” assumption are observed in turbomachinery applications, using the $k - \omega$ Wilcox model. These rather conflicting observations indicate that the applications in which the turbulence model differentiation can significantly affect the sensitivities, are still unclear in the literature. Nevertheless, the fact that, depending on the problem, wrongly signed sensitivities can be computed using the “frozen turbulence” assumption, strongly suggests that the differentiation of the turbulence model should be taken into consideration. All the above-mentioned adjoint approaches to turbulent flows were based on the discrete adjoint method. Regarding the exact continuous adjoint approach to turbulent flows, the first relevant papers were published recently. In [173], the continuous adjoint approach to the low-Re num-

ber Spalart–Allmaras model for incompressible flows was presented. The importance of differentiating the turbulence model was clearly highlighted, since in some of the examined cases, the wrong sensitivity sign was computed when the “frozen turbulence” assumption was used. The low-Re number Launder–Sharma $k - \epsilon$ model was also differentiated by the same group using continuous adjoint [127], presenting applications to shape and flow control optimization. The method was then extended to flows simulated using high-Re number turbulence models [174], and particularly the $k - \epsilon$ high-Re number model, by introducing the notion of adjoint wall functions. Recently, the continuous adjoint to the high-Re number $k - \omega$ SST model for incompressible flows was also presented [82], overcoming difficulties related to non-differentiable model functions such as the \min , \max operators. Concerning compressible flows, in [29] Bueno-Orovio et al. formulated the continuous adjoint to the low-Re number Spalart–Allmaras model. In order to simplify the final gradient expression and make it independent of the distance variation w.r.t. the design variables, the eikonal equation for computing distances was introduced as an extra constraint to the optimization problem [29].

2.2 Topology Optimization in Fluid Mechanics

The second part of this article evolves around topology optimization. Since an extensive literature survey of topology optimization methods in fluid mechanics is presented in the introduction of Sect. 7, only basic references are presented herein. The notion of topology optimization was initially introduced in structural mechanics by Bendsoe and Kikuchi [17]. There, instead of modifying the boundary of a structure in order to design its optimal shape, the values of the material density in each node, ranging from zero (indicating the presence of void/absence of material) to unity (indicating solid), were sought, in order to identify areas in which material should be added so as to increase the structural stiffness under certain load. A few years passed before Borrvall and Peterson [25], adapted the idea to fluid dynamics, for problems governed by the Stokes equations. Topology optimization in fluids was achieved by introducing a real-valued porosity dependent term to the flow equations. In areas where the porosity variable takes on high values, the flow encounters intense resistance, resulting to zero (or, practically, very small) velocity values. These areas correspond to the solidified part of the domain. On the other hand, in areas where porosity values are equal to zero, the flow encounters no resistance. These areas correspond to the flow passage. Topology optimization seeks the optimal porosity values at each node (for vertex-centered codes) or cell (cell-centered codes) in order to minimize the objective function. Hence, the number of design variables coincides with the number of grid nodes or cells, making the adjoint method the perfect choice for com-

puting sensitivities of the objective function w.r.t. the porosity values. Among the advantages of topology optimization are the absence of parameterization and the ability to design unconventional shapes. It is ideal for the preliminary design of ducts or duct systems where an initial design is not trivial and only the inlet to and the outlet from the duct are known. On the other hand, since no parameterization is used and the walls are formed as the interface between the solidified and fluid parts of the domain, some kind of post-processing is needed to obtain the boundaries of the final geometry. Based on [17], a number of papers were published about topology optimization for Stokes flows [3, 50, 65, 66, 165], while the concept of topology optimization was extended to laminar flows by Gersborg-Hansen et al. [54] and Olesen et al. [116]. Regarding turbulent flows, the continuous adjoint approach was presented in [118] by making, however, the “frozen turbulence” assumption. Later on [90], an exact continuous adjoint formulation for topology optimization, regarding 2D incompressible flows, that takes into consideration the differentiation of the low-Re number Spalart–Allmaras model was presented.

2.3 Article Outline

In Sect. 3, the continuous adjoint formulation based on the “frozen turbulence” assumption is initially presented in order to introduce the basic concept of the adjoint method. Then, in Sect. 4, the exact adjoint to flows governed by the low-Re number Spalart–Allmaras and Launder–Sharma $k - \epsilon$ turbulence models is incrementally built on the material presented in the previous section. The continuous adjoint formulation to the low-Re variant of the Spalart–Allmaras model presented in [173] is revisited in order to account for distance variations w.r.t. the design variables. Inspired from the work presented in [29], the eikonal equation, used to compute the distance field, is added as an extra constraint to the optimization problem. By formulating and numerically solving the adjoint to the eikonal equation, the final gradient derivatives expression becomes independent of distance variations. Moreover, the boundary conditions for the adjoint eikonal equation, presented in [29] are revisited. The sensitivity derivatives computed using the proposed method are validated by comparing their values to those obtained using FD in 2D shape optimization problems. In addition, the impact of making the “frozen turbulence” assumption on the convergence of optimization algorithms is investigated, in an attempt to quantify the acceleration of GBMs which rely upon exactly computed gradients.

In Sect. 5, the continuous adjoint to the high-Re number variant of the Spalart–Allmaras turbulence model is formulated and validated. The notion of adjoint wall functions, initially introduced in [174] is extended to cover cell-centered codes that impose no-slip velocity conditions at the wall

boundaries. The proposed method is validated against FD in external aerodynamics, concerning the flow around isolated airfoils. Then, the drag sensitivity maps computed over the surface of a glider using the proposed approach and the “frozen turbulence” assumption are compared.

In Sect. 6, the methods developed in Sects. 4 and 5 are applied to the computation of sensitivity maps and the optimization of 3D car geometries. The objective functions to be minimized include the drag or lift force exerted on the rear axis of the car. In addition, an approximation of the adjoint to flows governed by the DES Spalart–Allmaras model [152], is presented. Finally, an application of the continuous adjoint method for the design/optimization of hydraulic turbomachines is presented, targeting cavitation suppression. The latter is especially challenging since, in its original form, is a max./min. problem and, thus, non-differentiable.

In Sect. 7, the continuous adjoint formulation for use in topology optimization problems governed by the high-Re number variant of the Spalart–Allmaras model is presented. Constraints are imposed on the optimization problems using the Augmented Lagrange Multipliers method. Applications include the topology optimization of an air-conditioning duct of a passenger car targeting minimum total pressure losses and the design of a plenum chamber for a school formula car, targeting minimum fluid power dissipation.

The continuous adjoint methods presented in Sects. 4–7 are implemented in the open-source CFD toolbox OpenFOAM[©].

3 Continuous Adjoint for Shape Optimization: The “Frozen Turbulence” Assumption

3.1 State Equations

The governing equations of the flow problems used in this section are the steady-state, Navier–Stokes equations for an incompressible fluid. Excluding heat transfer, these are written as [92],

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \quad (3a)$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{\partial p}{\partial x_i} = 0 \\ i = 1, 2, (3) \quad (3b)$$

where v_i are the velocity components, p is the static pressure divided by the constant density, v is the constant bulk viscosity and v_t is the turbulent viscosity, computed by simultaneously solving turbulence model PDEs. In what follows, the Einstein convention, where repeated indices imply summation, is used unless stated otherwise. The above-mentioned

mean-flow equations along with the turbulence model equations and their boundary conditions are referred to as the primal (or state) equations. The primal variables array \mathbf{U} comprises v_i , p and the turbulence model variables.

In internal aerodynamics, the boundary conditions used to “close” the primal problem are (a) Dirichlet conditions for v_i and the turbulence model variables along with a zero Neumann condition for p at the inlet and along the solid wall boundaries of the computational domain and (b) a zero Dirichlet condition for p along with zero Neumann conditions for v_i and the turbulence model variables at the outlet.

3.2 Introduction of the Adjoint Variables

An optimization problem aims at the minimization of F in the design space defined by the vector of design variables \mathbf{b} . Typical objective functions are the total pressure losses in internal aerodynamics and the drag/lift forces exerted on the surface of an aerodynamic body in external aerodynamics. Without loss in generality, F is considered to be a scalar function. If more than one objective functions are to be minimized, the most straightforward approach for convex problems is to aggregate them into a single objective function by multiplying them with user-defined weight factors. Typical examples of design variables are the control point coordinates for parameterizations using Bézier–Bernstein polynomials or NURBS [133] in shape optimization, the jet velocities in flow control optimization [94] and the artificial porosity field values in topology optimization [90].

In general, F depends on \mathbf{U} and $\mathbf{b} = (b_1, b_2, \dots, b_N)$. F may directly depend on \mathbf{b} . For example, in shape optimization problems, F usually includes the normal unit vector to the surface on which F is defined, which depends on \mathbf{b} . F also depends on \mathbf{U} which, in turn, depends on \mathbf{b} . By varying the design variables, \mathbf{U} varies too in order to satisfy the primal equations. So, $F = F(\mathbf{b}, \mathbf{U}(\mathbf{b}))$.

GBMs make use of the derivatives of F w.r.t. the design variables, i.e. $\delta F / \delta b_n$, in order to minimize F . Typical examples of GMBs are steepest descent, conjugate gradient [51], quasi-Newton methods like BFGS [115] and SR1 [30] and Newton’s method [124]. The key factor determining the computational cost of the optimization algorithm is the way $\delta F / \delta b_n$ is computed. As mentioned in the introductions, a simple, though costly, method to compute this gradient is through finite differences (FD). Alternatively, in the direct differentiation (DD) approach, the derivatives of the flow variables \mathbf{U} w.r.t. the design variables b_n are firstly computed and, through them, the $\delta F / \delta b_n$ values are derived. As stated before, DD costs as many as N equivalent flow solutions (EFS), i.e. as if the flow equations were solved N times instead. Both FD or DD are used to compute “reference” gradient values, against which the results of the proposed adjoint methods are compared to.

The great advantage of the adjoint method is that it computes $\delta F / \delta b_n$ at a cost that does not depend on N . Starting point for the formulation of the adjoint problem is the augmented function, F_{aug} , which is defined by adding the volume integrals of the state PDEs, multiplied by the adjoint variable fields, to F . So,

$$F_{aug} = F + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega \quad (4)$$

where Ω is the computational domain. In Eq. 4, u_i stand for the adjoint velocity components and q is the adjoint pressure. Since this section presents the basics of the continuous adjoint formulation by making use of the “frozen turbulence” assumption, no extra integrals containing the turbulence model equations are added to the r.h.s. of Eq. 4. Considering that the residuals of the primal equations must be zero, $F \equiv F_{aug}$.

By employing the Leibniz theorem for the differentiation of volume integrals with variable boundaries to Eq. 4, we get

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} &= \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n} \int_{\Omega} u_i R_i^v d\Omega + \frac{\delta}{\delta b_n} \int_{\Omega} q R^p d\Omega \\ &= \frac{\delta F}{\delta b_n} + \int_{\Omega} u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega \\ &\quad + \int_S (u_i R_i^v + q R^p) n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \quad (5)$$

where the boundary S of Ω are decomposed as $S = S_I \cup S_O \cup S_W \cup S_{W_p}$; its constituents S_I , S_O , S_W and $S_{W_p} = S_{W_p}(\mathbf{b})$ correspond to the inlet, outlet, fixed and controlled (i.e. parameterized) wall boundaries of Ω , respectively. Also, n_k stand for the components of the unit vector which is normal to the surface. Since S_{W_p} is the only controlled part of S , Eq. 5 can be written as

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} &= \frac{\delta F}{\delta b_n} + \int_{\Omega} u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega \\ &\quad + \int_{S_{W_p}} (u_i R_i^v + q R^p) n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \quad (6)$$

Since $\delta F_{aug} / \delta b_n \equiv \delta F / \delta b_n$, these will be used indifferently hereafter.

A sharp distinction between $\delta() / \delta b_n$ and $\partial() / \partial b_n$ in Eq. 5 must be made. $\delta \Phi / \delta b_n$ denotes the total derivative of an arbitrary quantity Φ (which can be any flow variable or even the residual of the state equations) and represents the total change in Φ by varying b_n . The partial derivative $\partial \Phi / \partial b_n$ represents the variation in Φ caused due to changes in the flow variables (in turn, caused by the geometry deformation) and does not include contributions from the space deformation (still in its continuous form). From the discrete point of view, partial derivatives represent the variations in Φ if the grid nodes of

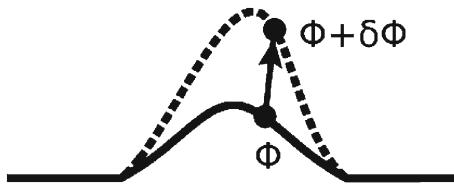


Fig. 1 The deformation of the starting geometry of the aerodynamic shape (solid line) to the new one (dashed line) causes the total variation in the quantity Φ , which depends on the flow variables, by $\delta\Phi = \frac{\partial\Phi}{\partial b}\delta b + \frac{\partial\Phi}{\partial x_k}\delta x_k$. The partial variation $\frac{\partial\Phi}{\partial b}\delta b$ is caused by changes in the flow variables over the entire domain due to the geometry change, without including changes due to the displacement of grid nodes. The contribution of $\frac{\partial\Phi}{\partial x_k}\delta x_k$ to the total variation $\delta\Phi$ depends on the displacement of the node under consideration, without accounting for any change in the flow variables

Ω remained unchanged. The total and partial derivatives are related through (Fig. 1)

$$\frac{\delta\Phi}{\delta b_n} = \frac{\partial\Phi}{\partial b_n} + \frac{\partial\Phi}{\partial x_k} \frac{\delta x_k}{\delta b_n} \quad (7)$$

For an arbitrary quantity Φ computed on a surface, such as the pressure on the airfoil surface, Eq. 7 takes a more specific form. Since any sufficiently small surface deformation can be seen as a normal perturbation, only the normal part of the surface deformation velocity $\delta x_k/\delta b_n$ causes a change in Φ [29, 64], hence

$$\frac{\delta_s\Phi}{\delta b_n} = \frac{\partial\Phi}{\partial b_n} + \frac{\partial\Phi}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m \quad (8)$$

Deformation in the tangential direction can be seen as an alternative parameterization of the surface and is of no value, hereafter.

In Eq. 8, $\delta_s/\delta b_n$ stands for the surface constrained material derivative, to be distinguished from the unconstrained material derivative expressed in Eq. 7. Since the surface deformation velocity can be decomposed into its normal and tangential components

$$\frac{\delta x_k}{\delta b_n} = \frac{\delta x_m}{\delta b_n} n_m n_k + \frac{\delta x_m}{\delta b_n} t_m t_k$$

and only the normal component is considered to affect the shape of the surface, Eq. 7 degenerates to Eq. 8 for surface constrained derivatives. Hence, Eq. 7 will be used in the interior of Ω while Eq. 8 for its boundaries.

For optimization problems with a fixed computational domain such as flow control or topology optimization problems, total and partial derivatives coincide since $\delta x_m/\delta b_n = 0$. The development presented in this section focuses on shape optimization, which poses the greatest mathematical challenge.

3.3 Objective Function Expression and Its Differentiation

An objective function comprising both surface and volume integrals can be generally expressed as

$$F = \int_S F_S dS + \int_\Omega F_\Omega d\Omega = \int_S F_{S_i} n_i dS + \int_\Omega F_\Omega d\Omega \quad (9)$$

where F_S and F_Ω are the integrands on the boundary and volume of the domain, respectively. It is assumed that objective functions defined along S include the normal unit vector \mathbf{n} . Differentiating F w.r.t. the design variables gives

$$\frac{\delta F}{\delta b_n} = \frac{\delta}{\delta b_n} \int_S F_{S_i} n_i dS + \frac{\delta}{\delta b_n} \int_\Omega F_\Omega d\Omega \quad (10)$$

3.3.1 Differentiation of the Surface Integral in Eq. 10

The surface integral in Eq. 10 is differentiated as follows

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_S F_{S_i} n_i dS &= \int_S \frac{\delta_s F_{S_i}}{\delta b_n} n_i dS + \int_S F_{S_i} \frac{\delta n_i}{\delta b_n} dS \\ &\quad + \int_S F_{S_i} n_i \frac{\delta(dS)}{\delta b_n} \end{aligned} \quad (11)$$

To simplify the notation, in Eq. 11, symbol δ is used instead of δ_S for variations in n_i and dS , since these quantities are defined exclusively on the surface.

By considering Eq. 8 for the total derivative of the surface-constrained F_{S_i} , Eq. 11 is written as

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_S F_{S_i} n_i dS &= \int_S \frac{\partial F_{S_i}}{\partial b_n} n_i dS + \int_S n_i \frac{\partial F_{S_i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS \\ &\quad + \int_S F_{S_i} \frac{\delta n_i}{\delta b_n} dS + \int_S F_{S_i} n_i \frac{\delta(dS)}{\delta b_n} \end{aligned} \quad (12)$$

Using the chain rule, the partial derivatives of F become

$$\frac{\partial F_{S_i}}{\partial b_n} = \frac{\partial F_{S_i}}{\partial v_k} \frac{\partial v_k}{\partial b_n} + \frac{\partial F_{S_i}}{\partial p} \frac{\partial p}{\partial b_n} + \frac{\partial F_{S_i}}{\partial \tau_{kj}} \frac{\partial \tau_{kj}}{\partial b_n} \quad (13)$$

where $\tau_{kj} = (\nu + \nu_t) \left(\frac{\partial v_k}{\partial x_j} + \frac{\partial v_j}{\partial x_k} \right)$ is the stress tensor.

After substituting Eq. 13 to Eq. 12, we get

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_S F_{S_i} n_i dS &= \int_S \frac{\partial F_{S_i}}{\partial v_k} n_i \frac{\partial v_k}{\partial b_n} dS + \int_S \frac{\partial F_{S_i}}{\partial p} n_i \frac{\partial p}{\partial b_n} dS \\ &\quad + \int_S \frac{\partial F_{S_i}}{\partial \tau_{kj}} n_i \frac{\partial \tau_{kj}}{\partial b_n} dS + \int_S n_i \frac{\partial F_{S_i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS \\ &\quad + \int_S F_{S_i} \frac{\delta n_i}{\delta b_n} dS + \int_S F_{S_i} n_i \frac{\delta(dS)}{\delta b_n} \end{aligned} \quad (14)$$

Derivatives of geometrical quantities appearing in Eq. 14 can further be developed by using identities from differential geometry [64],

$$\frac{\delta n_i}{\delta b_n} = -\frac{\partial_t}{\partial x_i} \left(\frac{\delta x_k}{\delta b_n} n_k \right), \quad \frac{\delta(dS)}{\delta b_n} = -\kappa \frac{\delta x_k}{\delta b_n} n_k dS \quad (15a)$$

where $\partial_t()/ ∂x_i and κ stand for the tangential derivative and mean curvature of the surface, respectively.$

3.3.2 Differentiation of the Volume Integral in Eq. 10

Based on the Leibniz theorem for differentiating volume integrals with moving boundaries,

$$\frac{\delta}{\delta b_n} \int_{\Omega} F_{\Omega} d\Omega = \int_{\Omega} \frac{\partial F_{\Omega}}{\partial b_n} d\Omega + \int_S F_{\Omega} n_k \frac{\delta x_k}{\delta b_n} dS \quad (16)$$

Since, in general, F may contain the flow variables v_i , p as well as differential operators of these quantities, such as the stress tensor, Eq. 16 is written as

$$\begin{aligned} \int_{\Omega} \frac{\partial F_{\Omega}}{\partial b_n} d\Omega &= \int_{\Omega} \tilde{F}_{\Omega,i}^v \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} \tilde{F}_{\Omega}^p \frac{\partial p}{\partial b_n} d\Omega \\ &+ \int_S \tilde{F}_{S,i}^v \frac{\partial v_i}{\partial b_n} dS + \int_S \tilde{F}_S^p \frac{\partial p}{\partial b_n} dS \end{aligned} \quad (17)$$

where $\tilde{F}_{\Omega}^{\Phi}$ includes the partial derivative $\partial F_{\Omega}/\partial \Phi$ plus any term resulting from the use of the Gauss divergence theorem for integrals of the form $\int_{\Omega} \frac{\partial}{\partial b_n} \left(\frac{\partial \Phi}{\partial x_j} \right) d\Omega$. The surface integrals on the r.h.s. of Eq. 17 are non-zero only if F_{Ω} includes differential operators of v_i or p . By substituting Eq. 17 into Eq. 16, we get

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_{\Omega} F_{\Omega} d\Omega &= \int_{\Omega} \tilde{F}_{\Omega,i}^v \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} \tilde{F}_{\Omega}^p \frac{\partial p}{\partial b_n} d\Omega \\ &+ \int_S \tilde{F}_{S,i}^v \frac{\partial v_i}{\partial b_n} dS + \int_S \tilde{F}_S^p \frac{\partial p}{\partial b_n} dS + \int_S F_{\Omega} n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \quad (18)$$

By substituting Eqs. 14 and 18 into Eq. 10 and taking into consideration that only S_{W_p} can be deformed, a final relation for the derivatives of an arbitrary function F w.r.t. to the design variables is reached,

$$\begin{aligned} \frac{\delta F}{\delta b_n} &= \int_S \left(\frac{\partial F_{S,i}}{\partial v_i} n_k + \tilde{F}_{S,i}^v \right) \frac{\partial v_i}{\partial b_n} dS \\ &+ \int_S \left(\frac{\partial F_{S,i}}{\partial p} n_i + \tilde{F}_S^p \right) \frac{\partial p}{\partial b_n} dS + \int_S \frac{\partial F_{S,i}}{\partial \tau_{ij}} n_k \frac{\partial \tau_{ij}}{\partial b_n} dS \\ &+ \int_{S_{W_p}} n_i \frac{\partial F_{S,i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S,i} \frac{\delta n_i}{\delta b_n} dS \\ &+ \int_{S_{W_p}} F_{S,i} n_i \frac{\delta(dS)}{\delta b_n} + \int_{S_{W_p}} F_{\Omega} n_k \frac{\delta x_k}{\delta b_n} dS \\ &+ \int_{\Omega} \tilde{F}_{\Omega,i}^v \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} \tilde{F}_{\Omega}^p \frac{\partial p}{\partial b_n} d\Omega \end{aligned} \quad (19)$$

Since in all applications presented in this article, $F_{\Omega} = 0$, all terms resulting from its differentiation (such as $\tilde{F}_{S,i}^v$, \tilde{F}_S^p , $\tilde{F}_{\Omega,i}^v$, \tilde{F}_{Ω}^p) are hereafter omitted.

3.4 Formulation of the Field Adjoint Equations

Equation 6, includes the partial derivatives of the mean-flow equations w.r.t. to the design variables, i.e., $\partial R_i^v/\partial b_n$ and $\partial R^p/\partial b_n$. These can be developed by applying the $\partial()/\partial b_n$ operator to Eqs. 3a and 3b. Note that the following permutation

$$\frac{\partial}{\partial b_n} \left(\frac{\partial \Phi}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left(\frac{\partial \Phi}{\partial b_n} \right) \quad (20)$$

is allowed since Eq. 20 includes only partial derivatives.

By differentiating Eqs. 3a and 3b and taking into consideration Eq. 20, we get,

$$\frac{\partial R^p}{\partial b_n} = -\frac{\partial}{\partial x_j} \left(\frac{\partial v_j}{\partial b_n} \right) \quad (21)$$

and

$$\begin{aligned} \frac{\partial R_i^v}{\partial b_n} &= \frac{\partial v_j}{\partial b_n} \frac{\partial v_i}{\partial x_j} + v_j \frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) \\ &- \frac{\partial}{\partial x_j} \left[(v + v_t) \frac{\partial}{\partial b_n} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{\partial}{\partial x_i} \left(\frac{\partial p}{\partial b_n} \right) \end{aligned} \quad (22)$$

In Eq. 22, $\partial v_t/\partial b_n$ is omitted due to the “frozen turbulence” assumption used in this section. The appropriate treatment of $\partial v_t/\partial b_n$ is presented in Sects. 4 and 5.

By using Eq. 21 and the Gauss divergence theorem, the volume integral $\int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega$ in Eq. 6 can be written as

$$\begin{aligned} - \int_{\Omega} q \frac{\partial}{\partial x_j} \left(\frac{\partial v_j}{\partial b_n} \right) d\Omega &= - \int_S q \frac{\partial v_j}{\partial b_n} n_j dS \\ &+ \int_{\Omega} \frac{\partial q}{\partial x_j} \frac{\partial v_j}{\partial b_n} d\Omega \end{aligned} \quad (23)$$

Taking into consideration Eq. 22, the inviscid terms in $\int_{\Omega} u_i \frac{\partial R_i^v}{\partial b_n} d\Omega$, Eq. 6, become

$$\begin{aligned} \int_{\Omega} u_i \frac{\partial v_i}{\partial x_j} \frac{\partial v_j}{\partial b_n} d\Omega &+ \int_{\Omega} u_i v_j \frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) d\Omega \\ &= \int_{\Omega} u_j \frac{\partial v_j}{\partial x_i} \frac{\partial v_i}{\partial b_n} d\Omega + \int_S u_i v_j n_j \frac{\partial v_i}{\partial b_n} dS \\ &- \int_{\Omega} \frac{\partial}{\partial x_j} (u_i v_j) \frac{\partial v_i}{\partial b_n} d\Omega \end{aligned} \quad (24)$$

$$\int_{\Omega} u_i \frac{\partial}{\partial x_i} \left(\frac{\partial p}{\partial b_n} \right) d\Omega = \int_S u_i n_i \frac{\partial p}{\partial b_n} dS - \int_{\Omega} \frac{\partial u_i}{\partial x_i} \frac{\partial p}{\partial b_n} d\Omega \quad (25)$$

The viscous terms in Eq. 6 can be written as

$$\begin{aligned} & - \int_{\Omega} u_i \frac{\partial}{\partial x_j} \left[(v + v_t) \frac{\partial}{\partial b_n} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] d\Omega \\ & = - \int_S u_i (v + v_t) \frac{\partial}{\partial b_n} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) n_j dS \\ & \quad + \int_{\Omega} (v + v_t) \frac{\partial u_i}{\partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) d\Omega \\ & \quad + \int_{\Omega} (v + v_t) \frac{\partial u_i}{\partial x_j} \frac{\partial}{\partial x_i} \left(\frac{\partial v_j}{\partial b_n} \right) d\Omega \end{aligned} \quad (26)$$

The first volume integral on the r.h.s. of Eq. 26 is further developed as

$$\begin{aligned} \int_{\Omega} (v + v_t) \frac{\partial u_i}{\partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) d\Omega & = \int_S (v + v_t) \frac{\partial u_j}{\partial x_i} n_j \frac{\partial v_i}{\partial b_n} dS \\ & - \int_{\Omega} \frac{\partial}{\partial x_j} \left((v + v_t) \frac{\partial u_j}{\partial x_i} \right) \frac{\partial v_i}{\partial b_n} d\Omega \end{aligned} \quad (27)$$

A similar development is used for the second volume integral on the r.h.s. of Eq. 26. By substituting Eqs. 21–27 into Eq. 6 and utilizing Eq. 19, we get

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} & = \int_S BC_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \left(u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS \\ & + \int_S \left(-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \right) \frac{\partial \tau_{ij}}{\partial b_n} dS \\ & + \int_{S_{W_p}} n_i \frac{\partial F_{S_i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_i} \frac{\delta n_i}{\delta b_n} dS \\ & + \int_{S_{W_p}} F_{S_i} n_i \frac{\delta (dS)}{\delta b_n} + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \\ & + \int_{\Omega} \left\{ u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \right. \\ & \quad \left. + \frac{\partial q}{\partial x_i} \right\} \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} \left(-\frac{\partial u_j}{\partial x_j} \right) \frac{\partial p}{\partial b_n} d\Omega \end{aligned} \quad (28)$$

where

$$BC_i^u = u_i v_j n_j + \tau_{a,ij} n_j - q n_i + \frac{\partial F_{S_k}}{\partial v_i} n_k \quad (29)$$

and the adjoint stresses are

$$\tau_{a,ij} = (v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (30)$$

To overcome the computation of the partial derivatives of p and v_i w.r.t. the design variables, the multipliers of $\partial v_i / \partial b_n$

and $\partial p / \partial b_n$, in the volume integrals of Eq. 28, are set to zero, giving rise to the adjoint mean-flow equations

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \quad (31)$$

$$\begin{aligned} R^u & = u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \\ & \quad + \frac{\partial q}{\partial x_i} = 0, \quad i = 1, 2, (3) \end{aligned} \quad (32)$$

Equation 31 is the adjoint continuity equation whereas Eq. 32 are the adjoint momentum equations. The field adjoint equations are quite similar to the primal ones; for instance, according to the continuity equation, the adjoint velocity is divergence-free. The adjoint momentum equations contain the convection, diffusion and adjoint pressure gradient terms. In contrast to the primal system of equations, the adjoint one is linear since the adjoint velocity is convected by the primal velocity. Theoretically, this would lead to smoother and faster convergence compared to the primal equations. However, due to the nature of the boundary conditions (see next section) and the presence of the non-conservative “adjoint transpose convection” (ATC; first term on the r.h.s. of Eq. 32) term, this is not always the case.

After satisfying the field adjoint equations, the remaining terms in Eq. 28 read

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} & = \int_S BC_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \left(u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS \\ & + \int_S \left(-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \right) \frac{\partial \tau_{ij}}{\partial b_n} dS \\ & + \int_{S_{W_p}} n_i \frac{\partial F_{S_i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_i} \frac{\delta n_i}{\delta b_n} dS \\ & + \int_{S_{W_p}} F_{S_i} n_i \frac{\delta (dS)}{\delta b_n} + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \end{aligned} \quad (33)$$

Starting from Eq. 33, the adjoint boundary conditions and the final expression of the sensitivity derivatives are extracted in Sects. 3.5 and 3.6, respectively.

3.5 Adjoint Boundary Conditions

3.5.1 Inlet Boundaries, S_I

At the inlet boundaries S_I , $\delta v_i / \delta b_n = \partial v_i / \partial b_n = 0$ since $\delta x_k / \delta b_n = 0$. Thus, the first integral on the r.h.s. of Eq. 33, written along S_I , vanishes automatically.

The second and third integrals in Eq. 33 can be eliminated by setting

$$u_{\langle n \rangle} = u_j n_j = -\frac{\partial F_{S_{I,i}}}{\partial p} n_i \quad (34a)$$

$$u_{\langle t \rangle}^I = \frac{\partial F_{S_{I,k}}}{\partial \tau_{ij}} n_k t_i^I n_j + \frac{\partial F_{S_{I,k}}}{\partial \tau_{ij}} n_k t_j^I n_i \quad (34b)$$

$$u_{\langle t \rangle}^{II} = \frac{\partial F_{S_{I,k}}}{\partial \tau_{ij}} n_k t_i^{II} n_j + \frac{\partial F_{S_{I,k}}}{\partial \tau_{ij}} n_k t_j^{II} n_i \quad (34c)$$

where t_i^I, t_i^{II} are the components of the tangent to the surface unit vectors and $u_{\langle t \rangle}^I, u_{\langle t \rangle}^{II}$ are the corresponding components of the adjoint velocity.

3.5.2 Outlet Boundaries, S_O

Along the outlet boundaries S_O , $\delta p / \delta b_n = \partial p / \partial b_n = 0$. Taking this into consideration, the second integral in Eq. 33, written along S_O , vanishes automatically. By assuming an almost uniform velocity profile along S_O , the third integral on the r.h.s. of Eq. 33 can be neglected.

In order to eliminate the first integral in the same equation, its integrand must be zeroed, resulting to 2(3) boundary condition equations in 2D (3D).

3.5.3 Unparameterized/Fixed Wall Boundaries, S_W

Along S_W , the boundary conditions for the primal variables are of the same type as those used at S_I and, also, $\delta x_k / \delta b_n = 0$. So, the adjoint boundary conditions given by Eq. 34 hold for S_W as well, by using F_{S_W} instead of F_{S_I} .

3.5.4 Parameterized/Controlled Wall Boundaries, S_{W_p}

Since $v_i = 0$ is imposed on S_{W_p} , $\delta v_i / \delta b_n = 0$. However, since $\delta x_k / \delta b_n \neq 0$, Eq. 8 yields

$$\frac{\partial v_i}{\partial b_n} = -\frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m \quad (35)$$

With the help of Eq. 35, the first term on the r.h.s. of Eq. 33, written along S_{W_p} , can be written as

$$\int_{S_{W_p}} \mathcal{BC}_i^u \frac{\partial v_i}{\partial b_n} dS = - \int_{S_{W_p}} \mathcal{BC}_i^u \frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m dS \quad (36)$$

to be included into the gradient expression.

The second and third integrals in Eq. 33 can be eliminated by setting

$$u_{\langle n \rangle} = -\frac{\partial F_{S_{W_p,i}}}{\partial p} n_i \quad (37a)$$

$$u_{\langle t \rangle}^I = \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k t_i^I n_j + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k t_j^I n_i \quad (37b)$$

$$u_{\langle t \rangle}^{II} = \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k t_i^{II} n_j + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k t_j^{II} n_i \quad (37c)$$

During the proof of Eq. 37, a number of terms appears due to the fact that $\delta x_k / \delta b_n \neq 0$ along S_{W_p} . These terms contribute to the sensitivity derivatives expression.

3.6 Final Expression of the Sensitivity Derivatives

After satisfying the adjoint mean-flow equations and their boundary conditions and taking into consideration Eqs. 33, 36 and the terms arising during the derivation of Eq. 37, the final expression for the sensitivity derivatives becomes

$$\frac{\delta F_{aug}}{\delta b_n} = \mathcal{ST}_L \quad (38)$$

where

$$\begin{aligned} \mathcal{ST}_L = & - \int_{S_{W_p}} \mathcal{BC}_i^u \frac{\partial v_i}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m dS \\ & + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS \\ & + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta (dS)}{\delta b_n} + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS \\ & - \int_{S_{W_p}} \left(-u_{\langle n \rangle} + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k n_l n_m \right) \mathcal{T} \mathcal{S}_1 dS \\ & - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^I t_m^I \mathcal{T} \mathcal{S}_2 dS \\ & - \int_{S_{W_p}} \left(\frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k (t_l^{II} t_m^I + t_l^I t_m^{II}) \right) \mathcal{T} \mathcal{S}_3 dS \\ & - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^{II} t_m^{II} \mathcal{T} \mathcal{S}_4 dS \end{aligned} \quad (39)$$

and

$$\mathcal{T} \mathcal{S}_1 = \tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k n_i n_j \quad (40a)$$

$$\mathcal{T} \mathcal{S}_2 = \tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^I t_j^I \quad (40b)$$

$$\mathcal{T} \mathcal{S}_3 = \tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^I \quad (40c)$$

$$\mathcal{TS}_4 = \tau_{ij} \frac{\delta(t_i^{II} t_j^{II})}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^{II} \quad (40d)$$

More specific forms of Eq. 38 can be derived only after considering the objective function of interest and casting it in the form of Eq. 9.

To summarize, the adjoint equations together with the boundary conditions derived in Sect. 4 as well as the sensitivity derivatives expressions given in Eqs. 38 and 40 can be used in either laminar or turbulent flows; however, for the latter, the “frozen turbulence” assumption was made since the PDEs which provide the turbulent viscosity field ν_t have not been differentiated. Depending on the Reynolds number and the local flow features, this omission might become detrimental. This is why Sects. 4 and 5 focus on alleviating this weakness.

4 Continuous Adjoint to Two Low-Re Number Turbulence Models for Shape Optimization

In this section, the continuous adjoint to two low-Reynolds number (low-Re) turbulence models, namely the Spalart–Allmaras [151] and Launder–Sharma $k - \epsilon$ [93] ones, is presented. By differentiating the turbulence model PDEs, the “frozen turbulence” assumption is overcome, leading to the computation of exact gradients.

4.1 Additional State Equations

The mean-flow equations, Eqs. 3, include the turbulent viscosity ν_t , which according to the Spalart–Allmaras model [151], is given by

$$\nu_t = \tilde{\nu} f_{v_1} \quad (41)$$

to be computed after solving the PDE

$$R\tilde{\nu} = \frac{\partial(\nu_j \tilde{\nu})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\tilde{\nu}}{\sigma} \right) \frac{\partial \tilde{\nu}}{\partial x_j} \right] - \frac{c_{b2}}{\sigma} \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 - \tilde{\nu} P(\tilde{\nu}) + \tilde{\nu} D(\tilde{\nu}) = 0 \quad (42)$$

for the turbulence state variable $\tilde{\nu}$.

In Eq. 42, the production and destruction terms are given by $P(\tilde{\nu}) = c_{b1} \tilde{Y}$, $D(\tilde{\nu}) = c_{w1} f_w(\tilde{Y}) \frac{\tilde{\nu}}{\Delta^2}$ where $\tilde{Y} = S + \frac{\tilde{\nu}}{\Delta^2 \kappa^2} f_{v_2}$, $S = |e_{ijk} \frac{\partial \nu_k}{\partial x_j} \mathbf{i}_i|$ is the vorticity magnitude, e_{ijk} is the permutation symbol, Δ the distance from the wall. Model functions and constants read $f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3}$, $f_{v_2} = \frac{1}{(1 + \frac{\chi}{c_{v_2}})^3}$, $f_{v_3} = \frac{(1 + \chi f_{v_1})}{c_{v_2}} \left[3 \left(1 + \frac{\chi}{c_{v_2}} \right) + \left(\frac{\chi}{c_{v_2}} \right)^2 \right] \left(1 + \frac{\chi}{c_{v_2}} \right)^{-3}$, $\chi = \frac{\tilde{\nu}}{\nu}$,

$$f_w = g \left(\frac{1 + c_{w3}^6}{g + c_{w3}^6} \right)^{1/6}, g = r + c_{w2}(r^6 - r), r = \frac{\tilde{\nu}}{Y \kappa^2 \Delta^2}, c_{b1} = 0.1355, c_{b2} = 0.622, \kappa = 0.4187, \sigma = \frac{2}{3}, c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}, c_{w2} = 0.3, c_{w3} = 2, c_{v1} = 7.1 \text{ and } c_{v2} = 5.$$

On the other hand, the low-Re Launder–Sharma $k - \epsilon$ turbulence model [93], is based on the following system of PDEs

$$\begin{aligned} R^k &= v_j \frac{\partial k}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{Pr_k} \right) \frac{\partial k}{\partial x_j} \right] - P_k + \epsilon + D = 0 \\ R^\epsilon &= v_j \frac{\partial \epsilon}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{Pr_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] - c_1 P_k \frac{\epsilon}{k} + c_2 f_2 \frac{\epsilon^2}{k} \\ &\quad - E = 0 \end{aligned} \quad (43)$$

and

$$\nu_t = c_\mu f_\mu \frac{k^2}{\epsilon} \quad (44)$$

The production term P_k and the low-Re number terms D and E are given by

$$P_k = \nu_t P_{KE}, \quad P_{KE} = \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial v_i}{\partial x_j} \quad (45)$$

$$D = 2\nu \left(\frac{\partial(\sqrt{k})}{\partial x_j} \right)^2, \quad E = 2\nu \nu_t \left(\frac{\partial^2 v_k}{\partial x_i \partial x_j} \right)^2 \quad (46)$$

Also,

$$\begin{aligned} f_\mu &= \exp \left[\frac{-3.4}{\left(1 + \frac{Re_t}{50} \right)^2} \right], \\ f_2 &= 1 - 0.3 \exp \left[-\min(Re_t^2, 50) \right] \end{aligned} \quad (47)$$

where $Re_t = k^2/(\nu \epsilon)$. The model constants are $c_\mu = 0.09$, $c_1 = 1.44$, $c_2 = 1.92$, $Pr_k = 1.0$ and $Pr_\epsilon = 1.3$.

4.2 Objective Function Differentiation

After taking into consideration the development presented in Sect. 3.3 and the fact that the objective function may directly depend on the turbulence model variables, Eq. 19 must be completed by adding a turbulence dependent term (dFT) to its r.h.s. The extra term reads

$$dFT_{SA} = \int_S \frac{\partial F_{S_i}}{\partial \tilde{\nu}} n_i \frac{\partial \tilde{\nu}}{\partial b_n} dS \quad (48a)$$

$$dFT_{KE} = \int_S \frac{\partial F_{S_i}}{\partial k} n_i \frac{\partial k}{\partial b_n} dS + \int_S \frac{\partial F_{S_i}}{\partial \epsilon} n_i \frac{\partial \epsilon}{\partial b_n} dS \quad (48b)$$

for the Spalart–Allmaras or the Launder–Sharma $k - \epsilon$ model, respectively.

4.3 Formulation of the Field Adjoint Equations

By taking the differentiation of the turbulence model PDEs into consideration, the derivative of the augmented objective function becomes

$$\frac{\delta F_{\text{aug}}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_{\Omega} u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS + [dFaT] \quad (49)$$

where, for the two turbulence models examined, term $[dFaT]$ stands for

$$[dFaT]_{SA} = \int_{\Omega} \tilde{v}_a \frac{\partial R^{\tilde{v}}}{\partial b_n} d\Omega + \int_{S_{W_p}} \tilde{v}_a R^{\tilde{v}} \frac{\delta x_k}{\delta b_n} n_k dS \quad (50a)$$

$$[dFaT]_{KE} = \int_{\Omega} k_a \frac{\partial R^k}{\partial b_n} d\Omega + \int_{\Omega} \epsilon_a \frac{\partial R^{\epsilon_a}}{\partial b_n} d\Omega + \int_{S_{W_p}} (k_a R^k + \epsilon_a R^{\epsilon}) \frac{\delta x_k}{\delta b_n} n_k dS \quad (50b)$$

In Eqs. 50, \tilde{v}_a is the adjoint to \tilde{v} , k_a is the adjoint kinetic energy and ϵ_a is the adjoint kinetic energy dissipation rate.

To overcome the “frozen turbulence” assumption, Eq. 22 should be replaced by

$$\begin{aligned} \frac{\partial R_i^v}{\partial b_n} &= \frac{\partial v_j}{\partial b_n} \frac{\partial v_i}{\partial x_j} + v_j \frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) + \frac{\partial}{\partial x_i} \left(\frac{\partial p}{\partial b_n} \right) \\ &\quad - \frac{\partial}{\partial x_j} \left\{ (v + v_t) \left[\frac{\partial}{\partial x_j} \left(\frac{\partial v_i}{\partial b_n} \right) + \frac{\partial}{\partial x_i} \left(\frac{\partial v_j}{\partial b_n} \right) \right] \right\} \\ &\quad - \frac{\partial}{\partial x_j} \left[\frac{\partial v_t}{\partial b_n} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] \end{aligned} \quad (51)$$

Compared to the “frozen turbulence” assumption, Sect. 3.4, extra terms result from the volume integrals in Eqs. 50 and the presence of $\partial v_t / \partial b_n$ in the last term in Eq. 51. The latter term, contributes the additional integrals

$$\begin{aligned} TM_{SA} &= - \int_S u_i \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) n_j \frac{\partial v_t}{\partial \tilde{v}} \frac{\partial \tilde{v}}{\partial b_n} dS \\ &\quad + \int_{\Omega} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial v_t}{\partial \tilde{v}} \frac{\partial \tilde{v}}{\partial b_n} d\Omega \end{aligned} \quad (52)$$

$$\begin{aligned} TM_{KE} &= - \int_S \frac{\partial v_t}{\partial k} u_i \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) n_j \frac{\partial k}{\partial b_n} dS \\ &\quad - \int_S \frac{\partial v_t}{\partial \epsilon} u_i \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) n_j \frac{\partial \epsilon}{\partial b_n} dS \\ &\quad + \int_{\Omega} \frac{\partial v_t}{\partial k} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial k}{\partial b_n} d\Omega \\ &\quad + \int_{\Omega} \frac{\partial v_t}{\partial \epsilon} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial \epsilon}{\partial b_n} d\Omega \end{aligned} \quad (53)$$

to the r.h.s. of Eq. 49.

After a lengthy development of the volume integrals in Eq. 50 based on the Gauss divergence theorem, Eq. 49 becomes

$$\begin{aligned} \frac{\delta F_{\text{aug}}}{\delta b_n} &= \int_S \mathcal{BC}_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \left(u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS \\ &\quad + \int_S \left(-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \right) \frac{\partial \tau_{ij}}{\partial b_n} dS \\ &\quad + \int_{\Omega} R_i^u \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} R^q \frac{\partial p}{\partial b_n} d\Omega \\ &\quad + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p},i}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS \\ &\quad + \int_{S_{W_p}} F_{S_{W_p},i} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p},i} n_i \frac{\delta (dS)}{\delta b_n} \\ &\quad + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS + \mathcal{ED} \end{aligned} \quad (54)$$

where \mathcal{ED} includes variations in turbulent quantities, with different forms for the two models. After setting the multipliers of $\partial p / \partial b_n$ and $\partial v_i / \partial b_n$ in the fields integrals of Eq. 54 to zero, the adjoint mean-flow equations read

$$R^q = - \frac{\partial u_j}{\partial x_j} = 0 \quad (55)$$

$$\begin{aligned} R_i^u &= u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \\ &\quad + \frac{\partial q}{\partial x_i} + AMS_i = 0, \quad i = 1, 2, (3) \end{aligned} \quad (56)$$

The extra terms in the adjoint momentum equations (AMS_i), arising from the differentiation of the turbulence models, are

$$AMS_i^{SA} = \tilde{v}_a \frac{\partial \tilde{v}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\tilde{v}_a \tilde{v} \frac{\mathcal{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} \right) \quad (57a)$$

$$\begin{aligned} AMS_i^{KE} &= k_a \frac{\partial k}{\partial x_i} + \epsilon_a \frac{\partial \epsilon}{\partial x_i} - 4 \frac{\partial^2}{\partial x_l \partial x_j} \left(\epsilon_a v v_t \frac{\partial^2 v_i}{\partial x_l \partial x_j} \right) \\ &\quad + 2 \frac{\partial}{\partial x_j} \left[\left(k_a + \epsilon_a c_1 \frac{\epsilon}{k} \right) v_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] \end{aligned} \quad (57b)$$

Equation 57 depend on the adjoint turbulence model fields \tilde{v}_a or k_a and ϵ_a . These satisfy the adjoint turbulence model PDEs, which emerge by zeroing the multipliers of $\partial \tilde{v} / \partial b_n$, $\partial k / \partial b_n$ and $\partial \epsilon / \partial b_n$ in the volume integrals of Eq. 54 (contained in \mathcal{ED}), yielding

$$R^{\tilde{v}_a} = - \frac{\partial (v_j \tilde{v}_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial \tilde{v}_a}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \tilde{v}_a}{\partial x_j} \frac{\partial \tilde{v}}{\partial x_j}$$

$$\begin{aligned}
& + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\tilde{v}_a \frac{\partial \tilde{v}}{\partial x_j} \right) + \tilde{v}_a \tilde{v} \mathcal{C}_{\tilde{v}} \\
& + \frac{\partial v_t}{\partial \tilde{v}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + (-P + D) \tilde{v}_a = 0
\end{aligned} \tag{58a}$$

$$\begin{aligned}
R^{k_a} = & - \frac{\partial (v_j k_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{Pr_k} \right) \frac{\partial k_a}{\partial x_j} \right] \\
& + \left(\frac{B_1}{Pr_k} - \frac{v}{k} \right) \frac{\partial k}{\partial x_j} \frac{\partial k_a}{\partial x_j} + \frac{B_1}{Pr_\epsilon} \frac{\partial \epsilon}{\partial x_j} \frac{\partial \epsilon_a}{\partial x_j} \\
& + B_1 \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} \\
& + \left[\frac{v}{2k^2} \left(\frac{\partial k}{\partial x_j} \right)^2 - \frac{v}{k} \frac{\partial^2 k}{\partial x_j^2} - P_{KE} B_1 \right] k_a \\
& - \left[c_1 \frac{\epsilon}{k} P_{KE} B_1 + 2v \left(\frac{\partial^2 v_k}{\partial x_i \partial x_j} \right)^2 B_1 + c_2 f_2 \frac{\epsilon^2}{k^2} \right. \\
& \left. - 1.2 c_2 \frac{k^2}{v^2} e^{-Re_t^2} - c_1 P_k \frac{\epsilon}{k^2} \right] \epsilon_a = 0
\end{aligned} \tag{58b}$$

$$\begin{aligned}
R^{\epsilon_a} = & - \frac{\partial (v_j \epsilon_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{Pr_\epsilon} \right) \frac{\partial \epsilon_a}{\partial x_j} \right] \\
& + \frac{B_2}{Pr_\epsilon} \frac{\partial \epsilon}{\partial x_j} \frac{\partial \epsilon_a}{\partial x_j} + \frac{B_2}{Pr_k} \frac{\partial k}{\partial x_j} \frac{\partial k_a}{\partial x_j} \\
& + B_2 \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j} + (1 - P_{KE} B_2) k_a \\
& + \left[-2v \left(\frac{\partial^2 v_k}{\partial x_i \partial x_j} \right)^2 B_2 - c_1 \frac{\epsilon}{k} P_{KE} B_2 + 2c_2 f_2 \frac{\epsilon}{k} \right. \\
& \left. - 0.6 c_2 \frac{k^3}{v^2 \epsilon} e^{-Re_t^2} - c_1 P_k \frac{1}{k} \right] \epsilon_a = 0
\end{aligned} \tag{58c}$$

The detailed expressions of terms $\mathcal{C}_{\tilde{v}}$, B_1 and B_2 can be found in [127, 173].

After satisfying the field adjoint equations, the field integrals including variations in the flow variables are eliminated and Eq. 59 takes the following form

$$\begin{aligned}
\frac{\delta F_{aug}}{\delta b_n} = & \int_S \mathcal{BC}_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \left(u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS \\
& + \int_S \left(-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \right) \frac{\partial \tau_{ij}}{\partial b_n} dS \\
& + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\partial x_k}{\partial b_n} n_k dS \\
& + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta (dS)}{\delta b_n} \\
& + \int_{S_{W_p}} (u_i R_i^v + q R^p) \frac{\delta x_k}{\delta b_n} n_k dS + \mathcal{SD}
\end{aligned} \tag{59}$$

where, depending on the turbulence model,

$$\begin{aligned}
\mathcal{SD}_{SA'} = & \int_S \mathcal{BC}^{\tilde{v}_a} \frac{\partial \tilde{v}}{\partial b_n} dS + \int_{S_{W_p}} \tilde{v}_a R^{\tilde{v}} \frac{\delta x_k}{\delta b_n} n_k dS \\
& - \int_S \tilde{v}_a \left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial \tilde{v}}{\partial x_j} \right) n_j dS \\
& + \int_{\Omega} \tilde{v} \tilde{v}_a \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega
\end{aligned} \tag{60a}$$

$$\begin{aligned}
\mathcal{SD}_{KE'} = & \int_S BC^{k_a} \frac{\partial k}{\partial b_n} dS + \int_S BC^{\epsilon_a} \frac{\partial \epsilon}{\partial b_n} dS \\
& - \int_S k_a \left(v + \frac{v_t}{Pr_k} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial k}{\partial x_j} \right) n_j dS \\
& - \int_S \epsilon_a \left(v + \frac{v_t}{Pr_\epsilon} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial \epsilon}{\partial x_j} \right) n_j dS \\
& - \int_S 4\epsilon_a v v_t \frac{\partial^2 v_k}{\partial x_i \partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial v_k}{\partial b_n} \right) n_i dS \\
& + \int_{S_{W_p}} (k_a R^k + \epsilon_a R^\epsilon) \frac{\delta x_k}{\delta b_n} n_k dS
\end{aligned} \tag{60b}$$

The last (volume) integral in Eq. 60a, containing the distance variation w.r.t. the design variables, is analyzed in the next section.

The adjoint boundary conditions are formulated by properly treating the boundary integrals in Eqs. 59 and 60 that depend on the variations in the flow variables.

The complete development of the adjoint boundary conditions can be found in [127, 173].

4.4 Distance Variation: The Adjoint to the Eikonal Equation

For the adjoint to the Spalart–Allmaras model, after satisfying the field adjoint equations and their boundary conditions, see Sect. 4.3, the sensitivity derivatives expression reads

$$\frac{\delta F_{aug}}{\delta b_n} = \mathcal{ST}_L + \mathcal{ST}_T + \mathcal{VT} \tag{61}$$

where \mathcal{ST}_T and \mathcal{VT} are given by

$$\begin{aligned}
\mathcal{ST}_T = & - \int_{S_{W_p}} \mathcal{BC}^{\tilde{v}_a} \frac{\partial \tilde{v}}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m dS \\
& + \int_{S_{W_p}} \tilde{v}_a R^{\tilde{v}} \frac{\delta x_k}{\delta b_n} n_k dS
\end{aligned} \tag{62}$$

$$\mathcal{VT} = \int_{\Omega} \tilde{v} \tilde{v}_a \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega \tag{63}$$

and \mathcal{ST}_L is given by Eq. 39. All integrals in terms \mathcal{ST}_L and \mathcal{ST}_T can be computed at a cost that is, practically, negligible compared to the cost of solving the primal or adjoint equations, since they contain primal and adjoint flow variables and their gradients written on S_{W_p} . This is not, however,

the case for the volume integral in Eq. 63, which contains the distance variation for the entire domain w.r.t. the design variables. This variation could be computed through finite differences, i.e. by perturbing each of the design variables by an infinitesimally small quantity ϵ in the positive and negative directions and re-computing nodal distances for the entire domain. The total variation in the distances can, then, be computed through Eq. 1 and includes $2N$ computations of the distance field (conducted for instance, by an exhaustive search of all cell centers with all boundary faces). The partial variation of Δ , appearing in Eq. 63 can, then, be computed through Eq. 7. The above-mentioned procedure is affordable only when N is relatively small. In addition, the use of finite-differences may introduce errors in the computation of $\partial\Delta/\partial b_n$ and is sensitive to the ϵ value.

4.4.1 The Eikonal Equation

In view of the above, a less costly and more robust way to deal with $\partial\Delta/\partial b_n$ is required. A literature survey [156–158, 166, 167], indicates that, in the past, PDEs have been used to compute the distances field. In particular, the solution of the eikonal equation

$$\frac{\partial\Delta}{\partial x_j} \frac{\partial\Delta}{\partial x_j} = 1 \quad (64)$$

gives a very good approximation to the Euclidean distance field. Due to numerical difficulties arising when solving Eq. 64 using standard finite-volume schemes, this is transformed into

$$R^\Delta = \frac{\partial(c_j\Delta)}{\partial x_j} - \Delta \frac{\partial^2\Delta}{\partial x_j^2} - 1 = 0 \quad (65)$$

where $c_j = \partial\Delta/\partial x_j$ act as the “convection velocity” components. Equation 65 is also known as the Hamilton–Jacobi equation, comprising non-linear convection and diffusion terms. The boundary conditions of Eq. 65 comprise a zero Dirichlet condition for the solid wall boundaries and $\frac{\partial\Delta}{\partial x_i} n_i = 1$ elsewhere. An indicative application of the Hamilton–Jacobi equation for the computation of distances is presented in Fig. 2. A square box with a side length of 1.4 m containing three square solid bodies is used. The side length of the internal bodies is equal to 0.2 m and their contours are considered to be the solid wall boundaries of the domain. Distances, computed by solving Eq. 65, are illustrated in Fig. 2a. In Fig. 2b, the relative error

$$d_e = \frac{|\Delta - y|}{y} \quad (66)$$

between the distance Δ computed by solving the Hamilton–Jacobi equation and the Euclidean distance y (computed by a

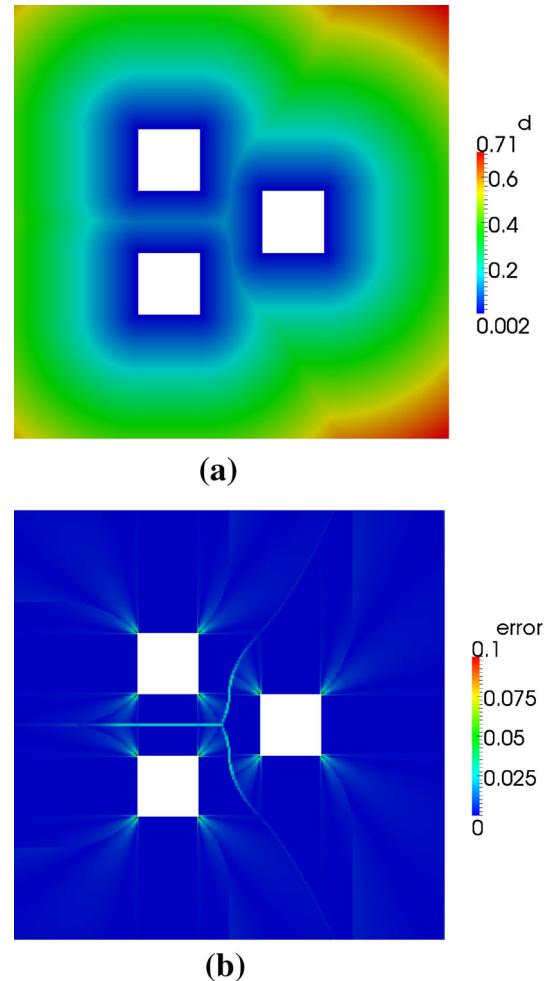


Fig. 2 A 2D application of the Hamilton–Jacobi equation. *Top* distances, computed by solving the Hamilton–Jacobi equation, from the three squares in the centre of the domain. *Bottom* relative error between the distance computed by the numerical solution of Eq. 65 and the Euclidean distance. The maximum relative error appears close to the corners of the three squares. It should be noted that, in the solution of the RANS equations using the Spalart–Allmaras turbulence model, only parts of the flow domain which are close to the solid walls are really sensitive to errors in distances. **a** Computed distance, **b** relative error

computationally expensive software based on the exhaustive search of the domain, i.e. by matching all cell centers with all boundary faces) is presented. The accuracy of the obtained distances is very satisfactory, since only a few cells have a relative error greater than 10 %.

4.4.2 Adjoint to the Hamilton–Jacobi Equation

Since a PDE for the computation of distances exists, Eq. 65, it can be added to the mathematical formulation of the optimization problem as an extra constraint. This was originally proposed by Bueno-Orovio et al. in [29], dealing with the low-Re Spalart–Allmaras model for compressible flows.

Taking this into consideration and in order to make the final sensitivity derivatives expression independent of $\partial\Delta/\partial b_n$, the augmented objective function is written as

$$F_{\text{aug}} = F + \underbrace{\int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega + \int_{\Omega} \tilde{v}_a R^{\tilde{v}} d\Omega}_{I_f} + \underbrace{\int_{\Omega} \Delta_a R^{\Delta} d\Omega}_{I_{\Delta}} \quad (67)$$

where Δ_a is the adjoint to the distance field. Equation 67 differs from the one presented in Sect. 4.3 only due to the introduction of the last volume integral containing the residual of the Hamilton–Jacobi equation. Differentiating Eq. 67 w.r.t. the design variables yields

$$\frac{\delta F_{\text{aug}}}{\delta b_n} = \frac{\delta I_f}{\delta b_n} + \frac{\delta I_{\Delta}}{\delta b_n} \quad (68)$$

The development of $\delta I_f/\delta b_n$ led to Eq. 61. For this reason, only the development of $\delta I_{\Delta}/\delta b_n$ is presented here. This yields

$$\frac{\delta I_{\Delta}}{\delta b_n} = \int_{\Omega} \Delta_a \frac{\partial R^{\Delta}}{\partial b_n} d\Omega + \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS \quad (69)$$

The differentiation of the Hamilton–Jacobi equation, Eq. 65, yields

$$\begin{aligned} \frac{\partial R^{\Delta}}{\partial b_n} &= \frac{\partial}{\partial x_j} \left[\frac{\partial \Delta}{\partial x_j} \frac{\partial \Delta}{\partial b_n} + \Delta \frac{\partial}{\partial x_j} \left(\frac{\partial \Delta}{\partial b_n} \right) \right] - \frac{\partial \Delta}{\partial b_n} \frac{\partial^2 \Delta}{\partial x_j^2} \\ &\quad - \Delta \frac{\partial}{\partial x_j} \left[\frac{\partial}{\partial x_j} \left(\frac{\partial \Delta}{\partial b_n} \right) \right] \\ &= 2 \frac{\partial \Delta}{\partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial \Delta}{\partial b_n} \right) \end{aligned} \quad (70)$$

Substituting Eq. 70 into Eq. 69, we get

$$\begin{aligned} \frac{\delta I_{\Delta}}{\delta b_n} &= \int_{\Omega} 2 \Delta_a \frac{\partial \Delta}{\partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial \Delta}{\partial b_n} \right) d\Omega + \int_S \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS \\ &= \int_S 2 \Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial b_n} dS + \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS \\ &\quad - \int_{\Omega} 2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) \frac{\partial \Delta}{\partial b_n} d\Omega \end{aligned} \quad (71)$$

After substituting Eqs. 61 and 71 into Eq. 68 and zeroing the multiplier of $\partial\Delta/\partial b_n$ in the resulting volume integrals, the field adjoint distance equation is derived, namely

$$R^{\Delta_a} = -2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) + \tilde{v} \tilde{v}_a \mathcal{C}_{\Delta} = 0 \quad (72)$$

where \mathcal{C}_{Δ} can be found in [173].

A zero Dirichlet condition should be imposed on Δ_a at S_I and S_O so as to eliminate of the first integral in Eq. 71 along these boundaries. Along S_{W_p} ,

$$\frac{\delta \Delta}{\delta b_n} = 0 \Rightarrow \frac{\partial \Delta}{\partial b_n} = -\frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} \quad (73)$$

Having satisfied the field adjoint distance equation, Eq. 72, and after taking Eq. 73 into consideration, the gradient expression becomes

$$\frac{\delta F_{\text{aug}}}{\delta b_n} = \mathcal{ST}_L + \mathcal{ST}_T + \frac{\delta I_{\Delta}}{\delta b_n} \quad (74)$$

where

$$\begin{aligned} \frac{\delta I_{\Delta}}{\delta b_n} &= \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS \\ &\quad - \int_{S_{W_p}} 2 \Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \quad (75)$$

and $\mathcal{ST}_L, \mathcal{ST}_T$ are given by Eqs. 39 and 62, respectively. In contrast to the present formulation (see also the next paragraph), in [29], Bueno-Orovio et al. propose a zero Dirichlet condition for Δ_a along S_{W_p} , for the low-Re Spalart–Allmaras model for compressible flows. This, however, would cause both integrals on the r.h.s. of Eq. 75 to be zero, leading to a zero impact of the distance differentiation on the adjoint formulation.¹

Since the distance variation may contribute a significant amount to the sensitivity derivatives value (see Sect. 4.5), herein it is proposed to impose a zero Neumann condition on Δ_a along S_{W_p} , allowing thus information from the differentiation of the distance field to be conveyed to the sensitivity derivatives. This is facilitated by the fact that all other quantities in the integrals appearing on the r.h.s. of Eq. 75 can be computed, practically, at zero cost.

It should be noted that Eq. 72 can be solved decoupled from the rest of the field adjoint equations, since the differentiation of the Hamilton–Jacobi equation does not contribute extra terms to them. So, the solution of the adjoint equations, including the adjoint to the Hamilton–Jacobi equation, involves the coupled solution of Eqs. 55–58a and then, the

¹ “[...]It is important to recall here that neither the Navier–Stokes adjoint equation in Eq. (21) nor the Spalart–Allmaras adjoint equation in Eq. (22) depends on the adjoint distance variable ψ_d (wn: the d_a variable in the current notation). This is also the case of the functional sensibility equation (28), since typical objective functionals in aerodynamics do not depend explicitly on the distance to the surface. In this situation, it is therefore not necessary to solve Eq. (23) (wn: the adjoint to the eikonal equation), and the adjoint system simply reduces to Eqs. (21) and (22).[...].” From [29].

numerical solution of Eq. 72 which depends on the adjoint turbulence model variable, \tilde{v}_a .

4.5 Validation: The Advantages of Solving the Adjoint to the Turbulence Model PDEs

Sections 3 and 4 present two possible ways for computing the gradient of objective functions in turbulent flows. The first approach (Sect. 3) neglects all variations in the turbulent viscosity, by assuming that changes in the shape to be designed affect the mean-flow quantities though the turbulence variables remain unaffected. This assumption leads to a system of adjoint equations which doesn't include the adjoint to the turbulence model PDEs. The second approach (Sect. 4) includes the differentiation of the turbulence model equation(s) (and the eikonal equation, if needed) and leads to the need of additionally solving the adjoint to the turbulence model equation(s). The gain from overcoming the “frozen turbulence” assumption is that the computed gradient is exact. This will be demonstrated in some cases below.

In what follows, the flow equations are solved using the open-source CFD toolbox OpenFOAM[©], utilizing a cell-centered, collocated finite-volume scheme. The mean-flow equations are solved in a segregated manner by employing the SIMPLE algorithm [129], and the solution of the turbulence model PDEs is decoupled.

4.5.1 Flow around a NACA0012 Airfoil, Spalart–Allmaras Turbulence Model

The adjoint to the low-Re Spalart–Allmaras turbulence model and the Hamilton–Jacobi equation presented in Sects. 4.3 and 4.4, is validated in a case involving the flow around the isolated NACA0012 airfoil. The flow Reynolds number based on the airfoil chord is $Re = 1 \times 10^6$ and the farfield flow angle is $\alpha_\infty = 2^\circ$. A hybrid mesh consisting of approximately 250,000 cells is used and based on the current flow conditions, the average y^+ value of the first cell-centres off the wall is around 0.2. The airfoil is parameterized using Bézier–Bernstein polynomials with 12 control points for each of its pressure and suction sides, Fig. 3.

Instead of using Euclidean distances for the production and dissipation terms in the turbulence model, the distance field obtained after solving the Hamilton–Jacobi equation, Eq. 65, is utilized.

The distance field obtained through the solution of the Hamilton–Jacobi equation is illustrated in Fig. 4a. In Fig. 4b, the relative error between the Hamilton–Jacobi distance distribution and the Euclidean one, Eq. 66, is presented. The relative error is almost zero close to the airfoil surface where the distance field must be accurate, indicating that the use of the distance field obtained through the solution of Eq. 65 does not introduce important errors.

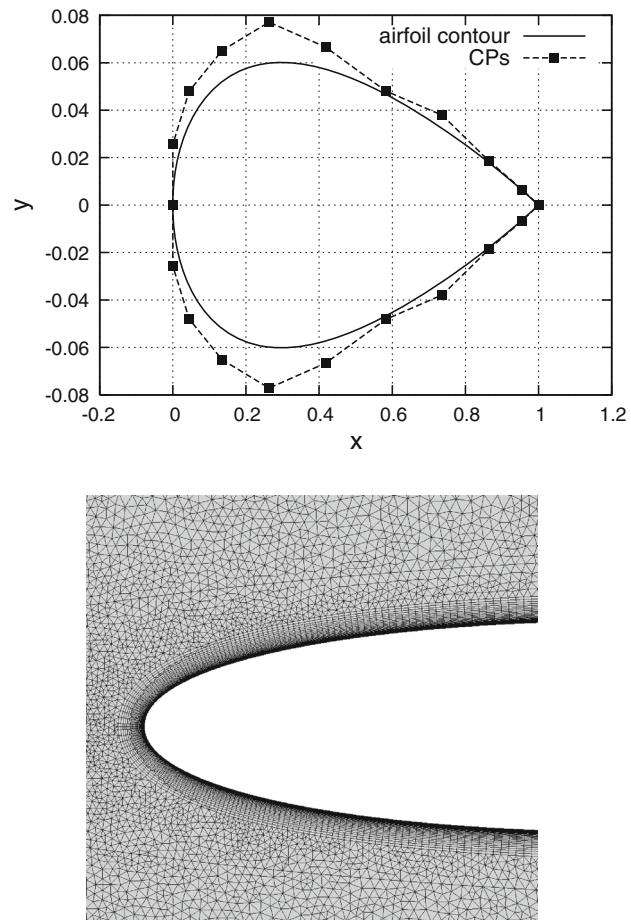


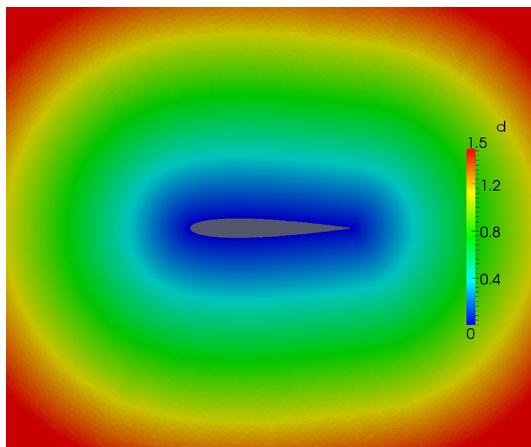
Fig. 3 Flow around the NACA0012 airfoil, adjoint to the low-Re Spalart–Allmaras model: Airfoil contour and Bézier control points, not in scale, (top) along with a blow-up view of the mesh near the leading edge (bottom)

The objective function used in the problem investigated in this section is the force exerted on the surface of the aero-dynamic body, projected on a predefined direction, \mathbf{r} . This objective function, to be minimized, can be written as

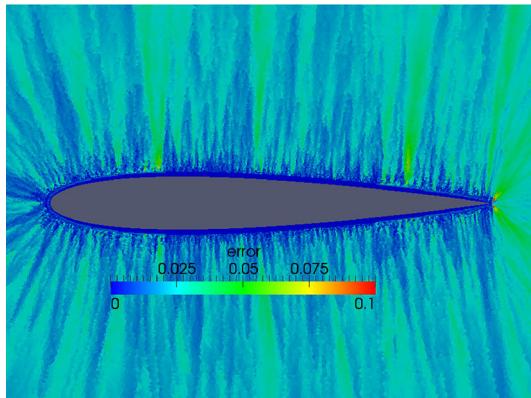
$$F_F = \int_{S_{W_p}} \left(-\tau_{ij} + p\delta_i^j \right) n_j r_i dS \quad (76)$$

The adjoint boundary conditions read

$$\begin{aligned} S_I : u_i = 0, \quad \frac{\partial q}{\partial n} = 0, \quad \tilde{v}_a = 0, \quad \Delta_a = 0 \\ S_O : \frac{\partial u_{\langle n \rangle}}{\partial n} = 0, \quad \frac{\partial \tilde{v}_a}{\partial n} = 0, \quad \Delta_a = 0 \\ q = u_{\langle n \rangle} v_{\langle n \rangle} + 2(v + v_t) \frac{\partial u_{\langle n \rangle}}{\partial n} \\ + \tilde{v}_a \tilde{v} \frac{\mathcal{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} n_l n_i = 0, \end{aligned}$$



(a)



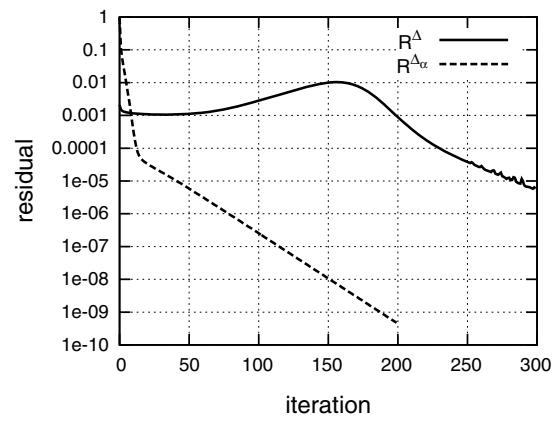
(b)

Fig. 4 Flow around the NACA0012 airfoil, adjoint to the low-Re Spalart–Allmaras model: *Top* distances computed after solving the Hamilton–Jacobi equation. *Bottom* relative error between distances computed by solving Eq. 65 and Euclidean distances. The cells of the flow domain which are close to the airfoil have a very low relative error, indicating that the differentiable Hamilton–Jacobi equation can be used to compute the distance field without introducing important errors. **a** Computed distance, **b** relative error

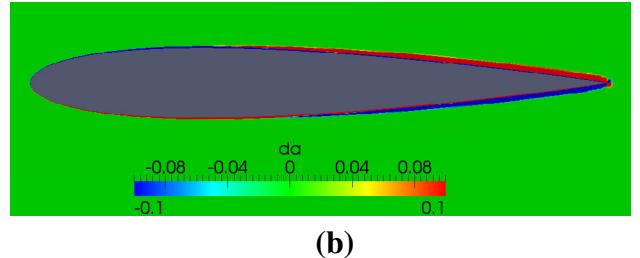
$$\begin{aligned}
 & v_n u_{\langle t \rangle}^l + (v + v_t) \left(\frac{\partial u_{\langle t \rangle}^l}{\partial n} + \frac{\partial u_{\langle n \rangle}}{\partial t^l} \right) \\
 & + \tilde{v}_a \tilde{v} \frac{C_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mz} n_z t_i^l = 0 \\
 S_{W_p} : u_i = -r_i, \quad & \frac{\partial q}{\partial n} = 0, \quad \tilde{v}_a = 0, \quad \frac{\partial \Delta_a}{\partial n} = 0
 \end{aligned} \tag{77}$$

and the sensitivity derivatives expression is given by Eq. 74.

In Eq. 74, term $\delta I_\Delta / \delta b_n$ quantifies the contribution of the distance differentiation to the sensitivity derivatives; $\delta I_\Delta / \delta b_h$ depends on the adjoint distance Δ_a , computed through the solution of Eq. 72. The convergence of the latter is presented in Fig. 5a and the adjoint distance field is illustrated in Fig. 5b.



(a)



(b)

Fig. 5 Flow around the NACA0012 airfoil, adjoint to the low-Re Spalart–Allmaras model: *Top* convergence of the Hamilton–Jacobi equation (marked as R^Δ) and its adjoint equation (marked as R^{Δ_a}). The CPU cost of solving either Eq. 65 or 72 is minor compared to the cost of solving the primal or the adjoint equations. *Bottom*: The adjoint distance field Δ_a . As expected, high values of the adjoint distance appear close to the airfoil. **a** Convergence, **b** adjoint distance field, Δ_a

After solving the primal equations, the adjoint equations to the low-Re Spalart–Allmaras turbulence model are solved, as presented in Sect. 4.3, targeting maximum lift ($r = [\sin(a_\infty), -\cos(a_\infty)]$). Only the y coordinates of the Bézier–Bernstein control points are allowed to vary, Fig. 3, giving rise to the 24 design variables of the optimization problem.

In Fig. 6, the importance of the turbulence model and distance differentiation is investigated by comparing sensitivities obtained using four different approaches: (a) the proposed approach which differentiates the turbulence model and uses the adjoint to the Hamilton–Jacobi equation to deal with the distance variation (sensitivities computed through Eq. 74), (b) the adjoint method in which the turbulence model is differentiated but the distance variation is neglected (Eq. 74 without $\delta I_\Delta / \delta b_n$), (c) the adjoint method which makes the “frozen turbulence” assumption (Sect. 3) and (d) FD, which serve as the reference values. It should be noticed that, if the differentiation of the turbulence model and distance field are neglected, the adjoint method computes sensitivities with the wrong sign along the entire length of the airfoil. This can have

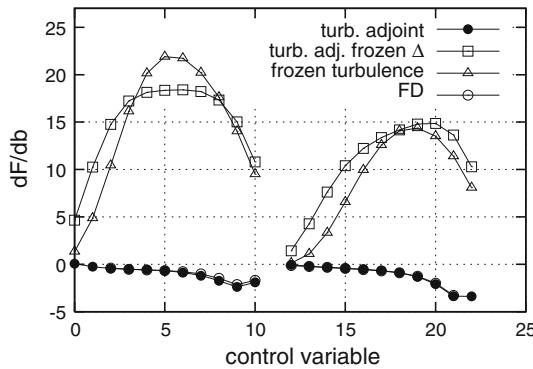


Fig. 6 Flow around the NACA0012 airfoil, adjoint to the low-Re Spalart–Allmaras model: Sensitivity derivatives of the minus lift force w.r.t. to the y coordinates of the control points of the Bézier–Bernsteinpolynomials parameterizing the airfoil. The first 12 points correspond to the suction side of the airfoil whereas the last 12 to the pressure side (numbering increasing from leading to trailing edge, along both sides). Four curves are presented: (a) the proposed method which takes into consideration the adjoint to the turbulence model equation and the adjoint to the Hamilton–Jacobi equation, labeled “turb. adjoint” (b) a variant of the adjoint method that takes into account the adjoint to the turbulence model but neglects the distance variation ($\delta I_\Delta / \delta b_n$ is neglected in Eq. 74), labeled “turb. adjoint, frozen Δ ” (c) the adjoint method that neglects the differentiation of the turbulence model entirely, labeled “frozen turbulence” and (d) finite differences, labeled “FD”. Approach (a) perfectly matches the outcome of FD whereas approaches (b) and (c) compute sensitivities with the wrong sign for the entire length of the airfoil

a devastating effect on any GBM, as the algorithm will be steered towards the wrong direction for all the control points parameterizing the airfoil.

4.5.2 Flow in an S-Shaped Duct, Spalart–Allmaras Turbulence Model

The accuracy of the proposed approach for computing sensitivity derivatives for flows governed by the Spalart–Allmaras turbulence model and, especially, the validity of the Hamilton–Jacobi equation and its adjoint for the treatment of the distance variation, is also examined in an internal aerodynamics case. In particular, the flow in an S-shaped duct is investigated. The structured mesh consists of 200×160 cells and the flow Reynolds number based on the inlet height is $Re = 2 \times 10^4$. The average y^+ value of the first cell-centres off the wall is approximately equal to 0.2, for this Reynolds number. The upper and lower duct contours are parameterized using Bézier–Bernsteinpolynomials with 12 control points per side, Fig. 7.

The optimization problem targets minimum total pressure losses between the inlet and outlet of the flow domain,

$$F_{p_t} = - \int_{S_I} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS - \int_{S_O} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS \quad (78)$$

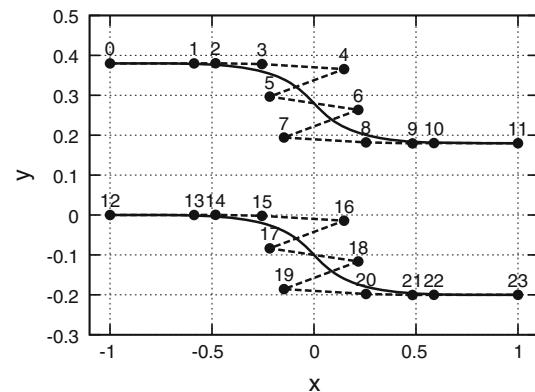


Fig. 7 Flow in an S-shaped duct, adjoint to the low-Re Spalart–Allmaras model: Shape of the duct and parameterization; axes not in scale. The lower and upper duct walls are parameterized using Bézier–Bernsteinpolynomials with 12 control points each. Only the y coordinates of the control points are allowed to vary, giving rise to an optimization problem with 24 design variables

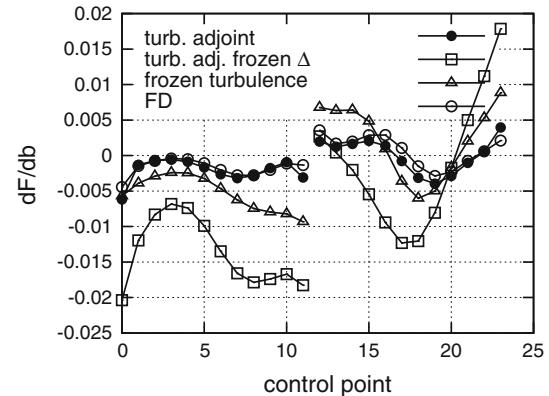


Fig. 8 Flow in an S-shaped duct, adjoint to the low-Re Spalart–Allmaras model: Sensitivity derivatives of F_{p_t} w.r.t. to the y coordinates of the control points parameterizing the duct. The first 12 points correspond to the upper wall whereas the last 12 correspond to the lower one. Notation of curves as in Fig. 6. It can be observed that the complete adjoint formulation reproduces the outcome of FD with a satisfactory precision, whereas the approaches that neglect turbulence model and distance variations (omission of $\delta I_\Delta / \delta b_n$ in Eq. 74) deviate from the reference values

The sensitivity derivatives, including terms from the distance differentiation, are given by Eq. 74. The four approaches used to compute sensitivity derivatives in the NACA0012 case are also utilized herein and the outcome is presented in Fig. 8.

In general, computing the exact gradient is not mandatory in optimization algorithms. In several applications the gradient descent algorithm is based on approximate gradients. However, the two above-mentioned examples illustrate that, depending on the case, omitting the differentiation of the turbulence model and that of the distance field, if applicable, may have severe consequences on the accuracy of the computed sensitivities. More examples in which the differentiation of the Spalart–Allmaras turbulence model is important,

along with a study on the impact of the “frozen turbulence” assumption depending on the flow Reynolds number can be found in [173].

4.5.3 Flow in an S-Shaped Duct, Launder–Sharma $k - \epsilon$ Turbulence Model

Even without computing the wrong sensitivity sign, following the “frozen turbulence” assumption can decelerate the convergence of GBMs. This is illustrated in the shape optimization of an S-shaped duct targeting minimum F_{p_t} , Fig. 9. The flow Reynolds number based on the inlet height is $Re = 1.2 \times 10^5$ and the Launder–Sharma $k - \epsilon$ is used. The upper and lower duct contours are parameterized using Bézier–Bernstein polynomials with 12 control points each. The steepest descent method was utilized to minimize F_{p_t} based on (a) the proposed adjoint method to compute $\delta F / \delta b_n$ and (b) adjoint with the “frozen turbulence” assumption.

The starting duct shape along with the optimal ones, based on the two variants of the adjoint formulation, are presented in Fig. 9-top. Practically, both have reached the same result. The shape resulting from (a) has an objective function value that is about 3 % lower than that of (b) and reaches the optimal solution after approximately 30 % less cycles than that making the “frozen turbulence” assumption.

5 Continuous Adjoint to High-Re Number Turbulence Models for Shape Optimization: Adjoint Wall Functions

In Sect. 4, the course of deriving the continuous adjoint field equations, their boundary conditions and the gradient expressions for flows modeled using low-Re turbulence models was presented.

Due to the strict mesh requirements associated with low-Re turbulence models (particularly close to the solid walls), the latter can not always be used in industrial applications. In contrast, high-Reynolds number (high-Re) turbulence models, employing the wall functions technique, are usually utilized to analyse the flow in industrial applications. The need to obtain accurate sensitivity derivatives for these applications dictates the necessity for differentiating high-Re turbulence models and the, so-called, law of the wall. This is analyzed in the current section of the article.

In low-Re turbulence models, the multipliers of flow variables variations are set to zero in the boundary integrals in order to derive the adjoint boundary conditions; similarly, when dealing with high-Re turbulence models, flow variations at the wall boundaries should be expressed as functions of the friction velocity variation, the multiplier of which should be set to zero in order to derive the adjoint high-Re boundary conditions. The first work dealing with the differ-

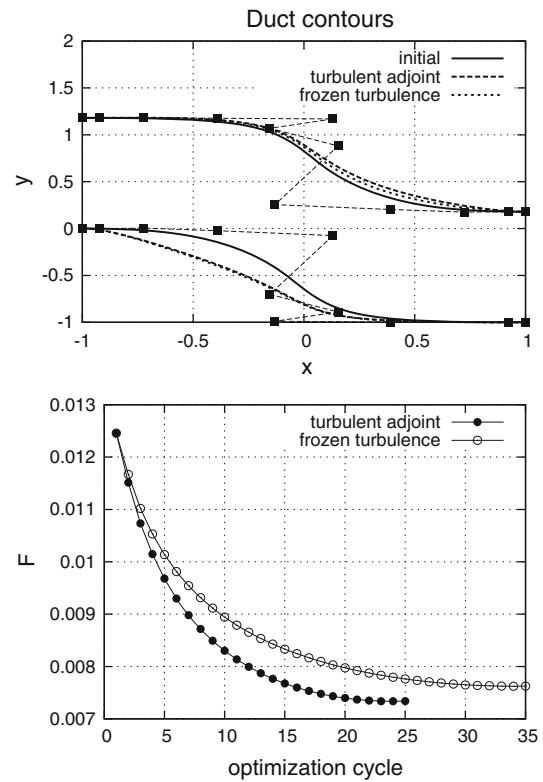


Fig. 9 Shape optimization of an S-shaped duct targeting min. F_{p_t} , using the adjoint to the Launder–Sharma $k - \epsilon$ model. **Top** starting duct shape compared to the optimal shapes resulting from the steepest descent algorithm that uses **a** the proposed method to compute $\delta F / \delta b_n$ (“turbulent adjoint”) and **b** the adjoint method based on the “frozen turbulence” assumption; axes are not in scale. **Bottom** Convergence history of the optimization algorithm driven by different adjoint methods. Case presented in detail in [127]

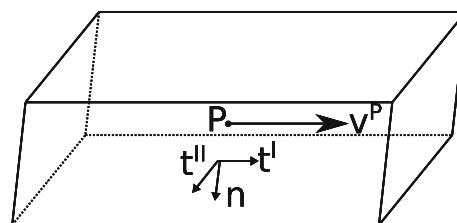


Fig. 10 Finite volume adjacent to the wall, where n is the outwards normal unit vector, t^I is parallel to the velocity vector at the first cell centre P and $t_i^{II} = e_{ijk} n_j t_k^I$

entiation of high-Re turbulence models using the continuous adjoint method was presented in [174]. There, the high-Re $k - \epsilon$ turbulence model [81], was differentiated by introducing the adjoint friction velocity and the so-called adjoint wall functions. The theory and implementation were based on an in-house, vertex-centered finite-volume code, using a pseudo-compressibility scheme [37], to solve the primal and adjoint equations for incompressible flows. More important, the implementation of the primal wall functions was based on a slip velocity condition [139, 140, 149], where the “real”

solid wall was assumed to lie at a distance Δ_{rw} underneath the grid boundary.

Here, the notion of the adjoint wall functions is expanded and applied to the differentiation of the high-Re Spalart–Allmaras model for incompressible flows. The cell-centered, pressure-based implementation of the high-Re Spalart–Allmaras model as programmed in OpenFOAM[©] is used as the basis of the adjoint formulation. According to the latter, a no-slip velocity boundary condition is imposed on the wall boundaries along with the law of the wall, expressed by a single formula governing both the linear sublayer and the logarithmic part of the boundary layer [52].

5.1 State Equations

The primal equations coincide with those presented in Sect. 4. The only difference lies on the way the wall viscous flux is computed. The wall function technique, as programmed in OpenFOAM[©], is based on a single formula modeling both the inner sublayer and the logarithmic region of the turbulent boundary layer [52, 164]

$$f_{WF} = -e^{-\kappa B} \left[e^{\kappa v^+} - 1 - \kappa v^+ - \frac{(\kappa v^+)^2}{2} - \frac{(\kappa v^+)^3}{6} \right] + y^+ - v^+ = 0 \quad (79)$$

where $B \approx 5.5$ and the non-dimensional distance and velocity at cell-centre P are

$$y_P^+ = \frac{\Delta^P v_\tau}{v}, \quad v_P^+ = \frac{|v_i|^P}{v_\tau} \quad (80)$$

In Eq. 80, v_τ is the friction velocity, computed by

$$v_\tau^2 = - \left[(\nu + v_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right]^f n_j t_i^I \quad (81)$$

where n_j and t_i^I are the components of the normal to the wall and parallel to the velocity at the first cell P (this velocity is considered to be parallel to the wall) unit vectors. In Eqs. 80 and 81, indices f and P denote quantities defined at the boundary wall face and the first cell centre, respectively, Fig. 10.

Using the wall functions technique, on the face $v_i^f = 0$ (no-slip condition) and the viscous flux for the i -th momentum equation at the wall boundary face f is given by

$$- \left[(\nu + v_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right]^f n_j \approx - \left(\nu + v_t^f \right) \frac{v_i^f - v_i^P}{|Pf|} \quad (82)$$

i.e. the normal velocity gradient at f is computed through a local FD scheme. However, it is known that any differ-

entiation normal to the boundary must be avoided on the coarse meshes used with high-Re turbulence models. The error made due the discretization of the normal velocity gradient in Eq. 82 is corrected by computing and using an “artificial” v_t^f , so that the wall shear stress satisfying Eq. 79 and that computed by differentiating the velocity field in space and multiplying by v_t^f be equal. So, solving the flow equations using the wall function technique means that $f_{WF}(v_\tau, |v_i|^P, \Delta^P) = 0$ (Eq. 79 at each face f) is solved for v_τ using a Newton–Raphson scheme and, then, Eq. 81 adjusts v_t^f as follows

$$v_t^f = - \frac{v_\tau^2}{\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)^f n_j t_i^I} - v \approx |Pf| \frac{v_\tau^2}{v_i^P t_i^I} - v \quad (83)$$

Eq. 82 is, then, used to calculate the contribution of the boundary viscous flux to the momentum equilibrium at the first cell adjacent to the wall.

5.2 Adjoint Formulation

The flow field adjoint equations coincide with those obtained for the low-Re Spalart–Allmaras model, Eqs. 55–58a. In addition, the adjoint to the Hamilton–Jacobi equation, Eq. 72, should also be taken into consideration. When dealing with the adjoint to the high-Re Spalart–Allmaras model, appropriate changes to Eq. 72 have to be made, in order to make the final sensitivity expression independent of the distance variation contained within the law of the wall, Eq. 79.

After satisfying the field adjoint equations, the remaining terms of the augmented objective function gradient read

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} = & \int_S BC_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \left(u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS \\ & + \int_S BC^{\tilde{v}_a} \frac{\partial \tilde{v}}{\partial b_n} dS + \int_S \left(-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \right) \frac{\partial \tau_{ij}}{\partial b_n} dS \\ & - \int_S \tilde{v}_a \left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial \tilde{v}}{\partial x_j} \right) n_j dS \\ & + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\partial x_k}{\partial b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\partial n_i}{\partial b_n} dS \\ & + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\partial (dS)}{\partial b_n} \\ & + \int_{S_{W_p}} (u_i R_i^v + q R^P + \tilde{v}_a R^{\tilde{v}} + \Delta_a R^{\Delta}) \frac{\partial x_k}{\partial b_n} n_k dS \\ & + \int_S 2 \Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial b_n} dS \end{aligned} \quad (84)$$

The adjoint boundary conditions are derived by appropriately treating the surface integrals in Eq. 84, containing variations in the flow variables. Apart from the adjoint boundary conditions imposed on S_W and S_{W_p} , the rest of the adjoint bound-

ary conditions coincide with those of the low-Re Spalart–Allmaras model, section 4.3.

5.2.1 Unparameterized/Fixed Wall Boundaries, S_W

Since S_W is fixed, the partial ($\partial(\cdot)/\partial b_n$) and total ($\delta(\cdot)/\delta b_n$) derivatives of any flow quantity are identical and the total derivatives of the normal and tangent unit vectors are equal to zero. Due to the Dirichlet condition imposed on \tilde{v} , the third integral in Eq. 84, written along S_W vanishes automatically. To make Eq. 84 independent of $\frac{\partial}{\partial b_n} \left(\frac{\partial \tilde{v}}{\partial x_j} \right) n_j$, a zero Dirichlet condition is imposed on \tilde{v}_a . To eliminate the dependency of $\delta F_{aug}/\delta b_n$ on $\partial p/\partial b_n$, the normal adjoint velocity must be equal to

$$u_{\langle n \rangle} = -\frac{\partial F_{S_{W,i}}}{\partial p} n_i \quad (85)$$

By further developing the first and fourth integrals in Eq. 84, a Dirichlet condition for $u_{\langle t \rangle}^{II}$ results, along with the following expression

$$u_{\tau}^2 = (v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j t_i^I = 0 \quad (86)$$

which can be characterized as the (squared) adjoint friction velocity. Its role is similar to that of the primal friction velocity, i.e. is used to compute the adjoint viscous flux in order to complete the adjoint momentum equilibrium at the first cell adjacent to S_W , Fig. 10. The adjoint friction velocity is an indispensable part of the adjoint system of equations to the high-Re Spalart–Allmaras model since, due to the “long distance” between f and P , Fig. 10, differentiating normal to the wall is prone to important errors. The implications of neglecting the adjoint law of the wall are discussed in section 5.3.

5.2.2 Parameterized/Controlled Wall Boundaries, S_{W_p}

The primal boundary conditions for the S_W and S_{W_p} boundaries are identical. As a result, the adjoint boundary conditions for S_{W_p} are the same to those imposed along S_W .

The proof of the adjoint boundary conditions along S_{W_p} is similar to the one presented in Sect. 5.2.1 for S_W , with the exception that, since S_{W_p} may vary in shape, the total and partial derivatives of the flow quantities are different. In addition, the total variations in the normal and tangent surface vectors are not zero, contributing thus extra terms during the formulation of the adjoint boundary conditions. These extra terms appear in the sensitivity derivatives expression which is given in the next subsection.

5.2.3 Sensitivity Derivatives

After satisfying the field adjoint equations and their boundary conditions, the remaining terms comprise the gradient expression, which reads

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} = & T_{SD}^{WF} - \int_{S_{W_p}} \mathcal{SD}_1 \frac{\partial \tau_{ij}}{\partial x_m} n_j t_i^I n_m n_k \frac{\delta x_k}{\delta b_n} dS \\ & - \int_{S_{W_p}} \mathcal{SD}_1 \tau_{ij} \frac{\delta(n_j t_i^I)}{\delta b_n} dS + \int_{S_{W_p}} \mathcal{SD}_{2,i} v_{\langle t \rangle}^I \frac{\delta t_i^I}{\delta b_n} dS \\ & - \int_{S_{W_p}} \mathcal{SD}_{2,i} \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left[\left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial \tilde{v}_a}{\partial x_j} n_j + \frac{\partial F_{S_z}}{\partial \tilde{v}} n_z \right] \frac{\partial \tilde{v}}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left(-u_{\langle n \rangle} + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k n_l n_m \right) \mathcal{T} \mathcal{S}_1 dS \\ & - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^I t_m^I \mathcal{T} \mathcal{S}_2 dS \\ & - \int_{S_{W_p}} \left(\frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k (t_l^{II} t_m^I + t_l^I t_m^{II}) \right) \mathcal{T} \mathcal{S}_3 dS \\ & - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^{II} t_m^{II} \mathcal{T} \mathcal{S}_4 dS \\ & - \int_{S_{W_p}} 2\Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\ & + \int_{S_{W_p}} (u_i R_i^v + q R^p + \tilde{v}_a R^{\tilde{v}}) \frac{\delta x_k}{\delta b_n} n_k dS \end{aligned} \quad (87)$$

where

$$\mathcal{SD}_1 = -u_{\langle t \rangle}^I + \phi_{ij} t_i^I n_j + \phi_{ij} n_i t_j^I \quad (88)$$

$$\mathcal{SD}_{2,i} = \tau_{a,ij} n_j - q n_i + \frac{\partial F_{S_{W_p,k}}}{\partial v_i} n_k \quad (89)$$

$$\phi_{ij} = \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k \quad (90)$$

Functions $\mathcal{T} \mathcal{S}_1$ to $\mathcal{T} \mathcal{S}_4$ are given by Eqs. 40; term T_{SD}^{WF} results from the differentiation of the law of the wall.

5.3 Validation

In this section, the adjoint formulation presented in Sect. 5.2 is validated on a 2D airfoil case, by comparing its outcome with the result of FD. In addition, the presented method is used to compute the drag sensitivity map on the surface of a glider and the computed sensitivities are compared to those obtained by making the “frozen turbulence” assumption.

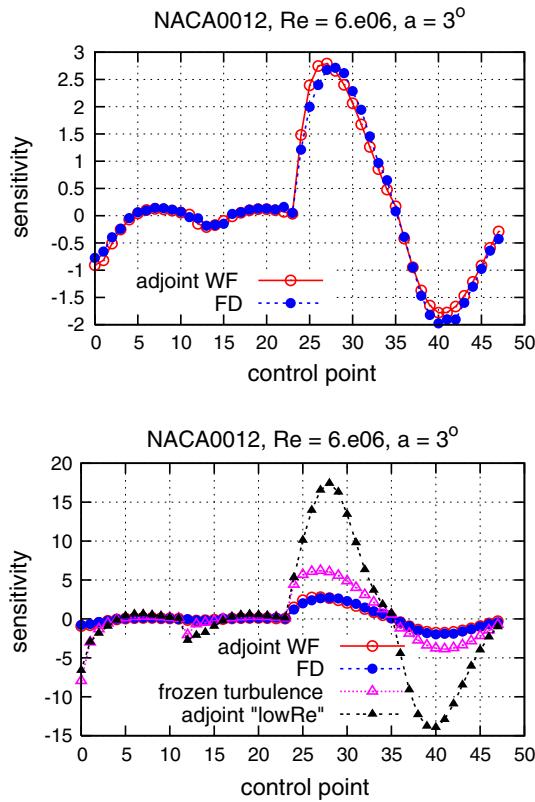


Fig. 11 Flow around the NACA0012 airfoil, adjoint to the high-Re Spalart–Allmaras model: *Top* drag sensitivities computed using the proposed method (marked as “adjoint WF”) are compared to the outcome of FD. The first 24 points correspond to the derivatives w.r.t. the x coordinates of the suction and pressure side control points while the last 24 to those w.r.t. the y coordinates. The two curves are in excellent agreement. *Bottom* The aforementioned curves are compared with those resulting from the adjoint method using the “frozen turbulence” assumption and the adjoint method with the “low-Re” approach (different scale on the *vertical axis*). The latter implies that the turbulence model is differentiated but the differentiation of the wall functions is disregarded

5.3.1 Flow Around the NACA0012 Airfoil, Adjoint to the High-Re Spalart–Allmaras Model

A hybrid grid was generated with a mean non-dimensional distance of the first cell adjacent to the wall equal to $y^+ \approx 10$. The flow Reynolds number is $Re = 6 \times 10^6$ and the infinite flow angle is $\alpha_\infty = 3^\circ$. Bézier–Bernstein polynomials with 12 control points for each of the pressure and suction sides are used to parameterize the airfoil surface.

The proposed method for computing sensitivity derivatives using the adjoint wall functions technique is used to obtain the sensitivities of the drag force (Eq. 76 with $\mathbf{r} = [\cos \alpha_\infty, \sin \alpha_\infty]^T$) exerted on the airfoil w.r.t. the (x, y) coordinates of the 24 control points, resulting to a total of 48 design variables. The outcome of this computation is compared with the result of FD in Fig. 11-top. The two curves are in a very good agreement even for the leading and trailing

edge control points. In Fig. 11-bottom, the two aforementioned curves are also compared with those computed using the “frozen turbulence” assumption and the adjoint “low-Re” approach. The latter implies that the primal solver is based on the high-Re Spalart–Allmaras model; however, the adjoint is based on the differentiation of the low-Re variant of the same model [173]. The gain in accuracy is obvious. In this case, the adjoint “low-Re” approach performs even worse than the “frozen turbulence” assumption. In other words, the incomplete differentiation of the turbulence model may produce worse results than the complete omission of its differentiation.

5.3.2 Flow Around a Glider; Adjoint to the High-Re Spalart–Allmaras Model

The goal of this application is to compute the sensitivity derivatives of the drag force exerted on the surface of a glider w.r.t. the normal displacement of its wall boundary nodes. The latter is also referred to as the drag sensitivity map. The mesh modeling half of the glider consists of approximately 4.7 million cells, the farfield flow angle is $\alpha_\infty = 8^\circ$, the flow Reynolds number is 1.5×10^6 and the average y^+ of the first cell centers off the wall is 60. The geometry was provided by Pipistrel Podjetje za alternativno letalstvo d.o.o. Ajdovscina in the context of the EC-funded project “RBF4AERO-Innovative Benchmark Technology for Aircraft Engineering Design And Efficient Design Phase Optimisation”. The sensitivity maps computed using the adjoint wall functions approach and the adjoint method which makes the “frozen turbulence” assumption are compared in Fig. 12. It can be clearly seen that the “frozen turbulence” assumption leads to wrongly signed sensitivities close to the leading edge of the wing suction side.

The colour map used to interpret sensitivity maps is given in Table 1 and holds for every sensitivity map presented in this article.

6 Industrial Applications

In Sects. 4 and 5, the adjoint to the high- and low-Re number variants of the Spalart–Allmaras model were presented and validated mainly on the basis of academic test cases. In this chapter, the developed methods and software are applied to two industrial cases originating from the automotive industry. The computational meshes and flow data of all test cases are kindly provided by Volkswagen AG. The two cases deal with the drag, lift and rear lift minimization of passenger cars. In addition, the optimization of a Francis turbine runner targeting cavitation suppression is also presented.

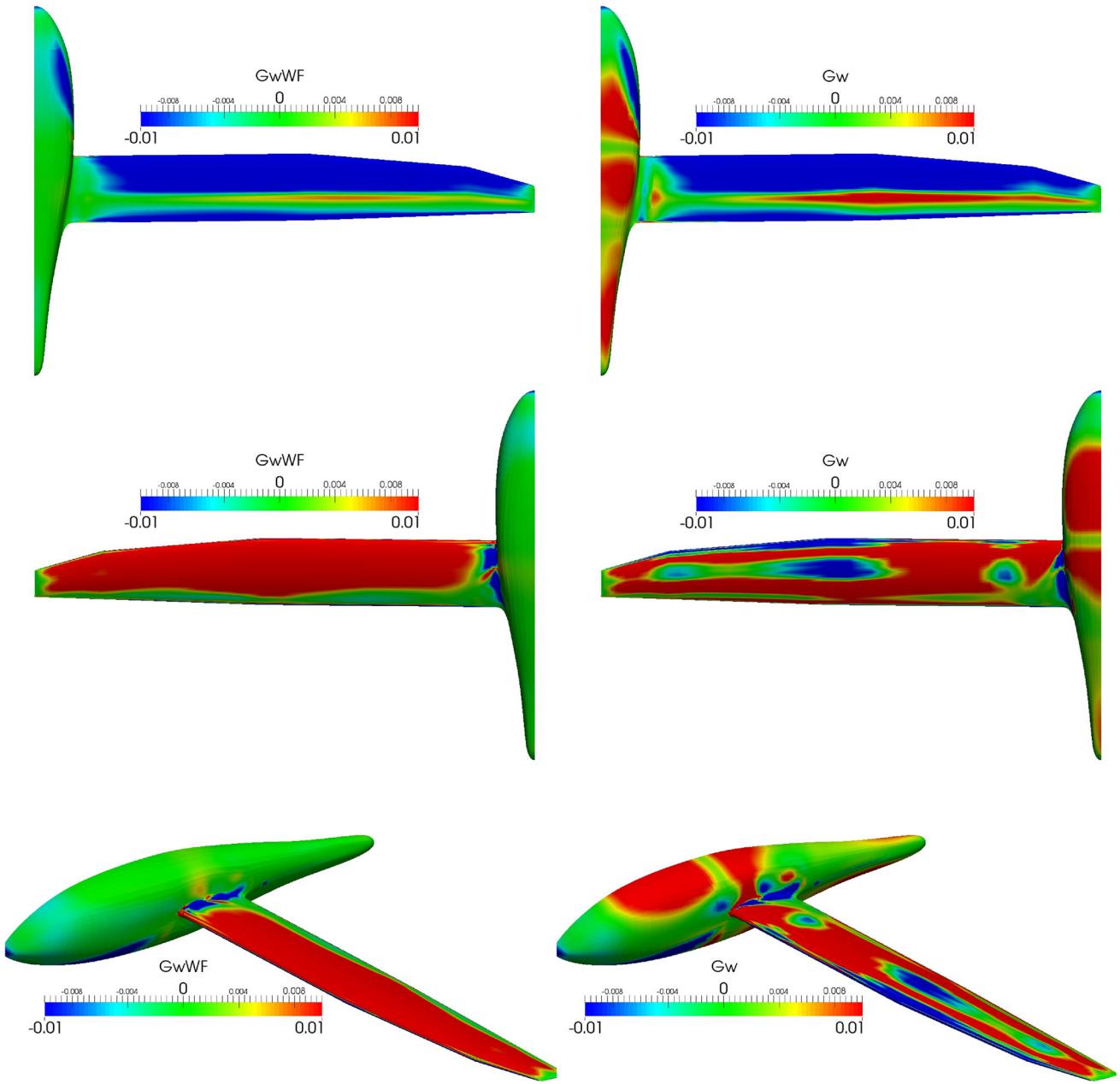


Fig. 12 Flow around a glider, adjoint to the high-Re Spalart–Allmaras model: drag sensitivity maps computed on the surface of a glider, using (a) the complete adjoint approach which utilizes adjoint wall functions (left) and (b) the adjoint approach making the “frozen turbulence”

assumption (right). Colour map as in Table 1. It can be observed that the “frozen turbulence” assumption computes the wrong sign over a portion of the wing suction side and overestimates the sensitivity magnitude on the fuselage

6.1 Optimization of a Concept Car Targeting Drag Reduction

In the first case, the drag minimization of the VW L1 concept car, Fig. 13a, is targeted. The L1 model features a very efficient aerodynamic design, resulting to a C_D value of about 0.19, when an indicative C_D value of a conventional passenger car is around 0.3 [2]. The goal of the optimization process is to further decrease the drag force exerted on the

car. A computational mesh of 18 million cells was generated to model half of the car, comprising boundary layers with prismatic elements and hexahedral elements for the rest of the computational domain. At the flow conditions under consideration, the average y^+ value of the first cell centres off the wall is close to unity. The flow is modeled using the RANS equations along with the low-Re Spalart–Allmaras model. The primal velocity magnitude is illustrated in Fig. 13b. The adjoint method that takes into consideration the differentia-

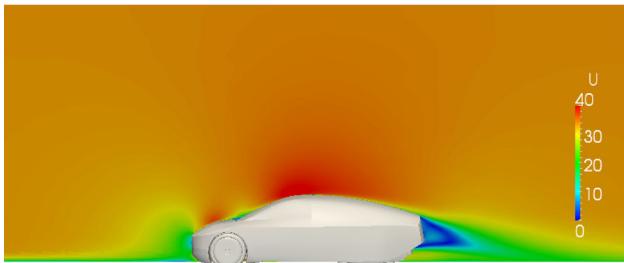
Table 1 Colour map used to interpret sensitivity maps

| Colour | Displacement direction |
|--------|------------------------|
| Red | Inwards |
| Blue | Outwards |
| Green | Neutral |

Red areas should be displaced inwards in order to decrease F while blue areas should be displaced outwards. Green-coloured areas have very small sensitivity values and have, thus, a small optimization potential



(a)



(b)



(c)

Fig. 13 VW L1 drag minimization: **a** Geometry of the VW L1 concept passenger car. With a drag coefficient below 0.2, its aerodynamic performance is approximately 30 % better than that of a regular passenger car. **b** Primal velocity field computed using the RANS equations along with the low-Re Spalart–Allmaras model. **c** Adjoint velocity field computed by solving the adjoint equations, including the adjoint to the Spalart–Allmaras model equation. Case presented in detail in [120]

tion of the turbulence model, Sect. 4, is used to compute the drag sensitivity map, Fig. 14 top and mid. Even if an optimization algorithm was not going to be used to minimize the objective function, the sensitivity map contains useful information for the designer.

According to Fig. 14 top and mid, the L1 exhibits significant potential for drag minimization in several areas, among

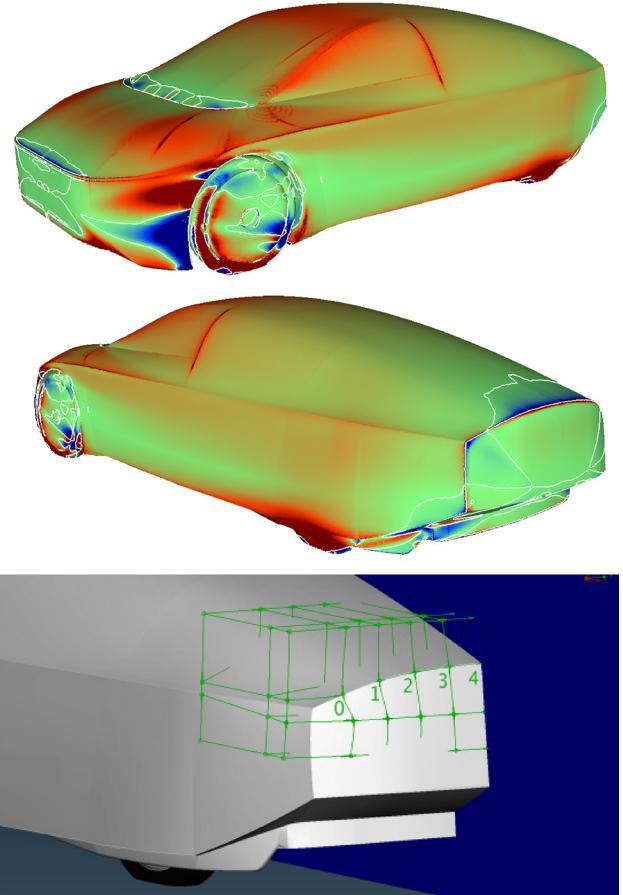


Fig. 14 VW L1 drag minimization: *Top-Mid* drag sensitivity derivatives w.r.t. the normal displacement of the wall boundary nodes, a.k.a the drag sensitivity map. Colour map as in Table 1. Areas with high absolute valued sensitivity derivatives (intense colours in the figures) have the greater potential for optimization. Zero sensitivity derivative isolines are coloured in white, highlighting the sign change in the sensitivity derivatives. An outward displacement is recommended for the rear part of the car, leading practically to the formation of a rear spoiler. *Bottom* Morphing boxes drawn around the high valued sensitivity derivatives area on the rear of the car. All nodes, both surface and volume, included in these morphing boxes get affected by the displacement of the control points (green vertices on the figure). The z coordinates of the control points, numbered from 0 to 4, which are snapped on the surface of the car, are the design variables of the drag minimization problem. Case presented in detail in [120]. (Color figure online)

them the front wing, the front end of the bonnet and the rear of the car; interestingly, the sensitivity map suggests the formation of a rear spoiler there. In order to proceed with the optimization procedure, the surface of the car must be parameterized. The parameterization of the commercial package ANSA [1], is used to define morphing boxes at the rear of the car, Fig. 14-bottom. The z coordinates of five variable control points, snapped on the surface of the car, are used as the design variables of the optimization problem, Fig. 14-bottom. A variation in these control points causes the displacement of the surface and volume nodes contained in the correspond-

ing morphing boxes, through a Free Form Deformation (FFD) [16]) algorithm. This allows for closed form expressions of the surface and volume grid nodal displacement w.r.t. the control points displacement, i.e. $\delta x_k / \delta b_n$ can be extracted through the parameterization. Then, the sensitivity derivatives can be computed through Eq. 74. The steps of the adjoint-based optimization algorithm are summarized below:

1. Parameterize the surface and volume mesh, obtaining an analytical expression for $\delta x_k / \delta b_n$. Parameterization is based on a FFD algorithm, implemented in ANSA.
2. Solve the flow equations, Eqs. 3 and 42.
3. Compute F through Eq. 76. Stop here if converged.
4. Solve the adjoint equations, Eqs. 55—58a, taking into consideration the adjoint to the turbulence model, along with the appropriate boundary conditions.
5. Compute $\delta F / \delta b_n$ using Eq. 74.
6. Update the design variables using steepest descent, $\Delta b_n = b_n^{k+1} - b_n^k = -\eta \left. \frac{\delta F}{\delta b_n} \right|_k$, where η is a user-defined step and k the optimization cycle index.
7. Compute the mesh deformation through $\Delta x_k = \frac{\delta x_k}{\delta b_n} \Delta b_n$, where $\delta x_k / \delta b_n$ is known from step 1. Update the mesh point coordinates.
8. Go to step 2.

After 19 optimization cycles, the drag force exerted on the surface of the car was reduced by more than 2 %. In addition, the lift force was also reduced by approximately 30 %. The latter was not part of the objective function, however, this is welcome since it results to better traction between the car wheels and the road surface. The optimization history and a comparison between the baseline (existing concept car) and optimized geometries are presented in Fig. 15-top-left and top-right, respectively. In the latter, the formation of a rear spoiler with a height of approximately 20 mm can be observed.

In Fig. 15-bottom, the pressure distribution on the baseline and optimized car surfaces is illustrated. It can be observed that the pressure on top and in front of the formed spoiler has increased. The increased pressure on top of the spoiler itself has a negative effect on the drag since, the corresponding force is directed backwards. However, the increased pressure in the area in front of the spoiler results to a force pointing in the opposite direction. The resultant force has the same direction with the car motion, leading eventually to lower drag. This is facilitated by the increased pressure on the back of the car.

The above-mentioned study was completed during the stay of the first author at the Volkswagen premises in Wolfsburg and was conducted with the collaboration of Dr. Carsten Othmer, Volkswagen, and Konstantinos Haliskos, BETA

CAE Systems, who provided the computational grid and the mesh deformation software [120].

6.2 VW Polo Rear Lift/Resultant Lift Reduction

When analyzing the flow around either passenger or racing cars, the lift force exerted on the vehicle is of primary importance. In contrast to airplanes, a low -or even negative- lift force is targeted, since such a force increases the wheels traction on the road. Due to the high pressure applied on the front of the car where the stagnation point is located, the lift force exerted on the front wheel axis (front lift) is usually negative (downforce), providing sufficient traction of the front part of the car on the ground. On the contrary, due to the relatively low pressure exerted on the top-rear of the car, the lift force applied on the rear wheel axis (rear lift) is usually positive. The objective of the application presented in this section is the computation of the rear lift sensitivities w.r.t. the normal displacement of the car surface.

The resultant lift force L exerted on the car can be written as

$$L = L_F + L_R \quad (91)$$

where L_F and L_R are the front and rear lift forces, respectively, Fig. 16. L , L_F and L_R are positive in the direction of the positive z axis, $\mathbf{r} = [0, 0, 1]^T$, Fig. 16.

The y component of the moment vector computed w.r.t. the center of rotation \mathbf{c}^R , located on the ground, in the middle of the wheelbase w_b , Fig. 16, which tends to lift the wheels of the car off the ground, is

$$M_y = L_F \frac{w_b}{2} - L_R \frac{w_b}{2} \quad (92)$$

and, therefore,

$$L_F = \frac{L}{2} + \frac{M_y}{w_b} \quad (93)$$

$$L_R = \frac{L}{2} - \frac{M_y}{w_b} \quad (94)$$

The lift force L can be derived from Eq. 76 with $\mathbf{r} = [0, 0, 1]^T$ and the y component of the moment from

$$M_y = \int_{S_W} r_l^M e_{lkj} (x_k - c_k^R) \left(-\tau_{ij} + p \delta_i^j \right) n_j dS \quad (95)$$

where $\mathbf{r}^M = [0, 1, 0]$.

The primal problem equations used to model the flow around the VW Polo geometry are the Navier–Stokes equations coupled with the Detached Eddy Simulation (DES) model proposed by Spalart et al. [152],

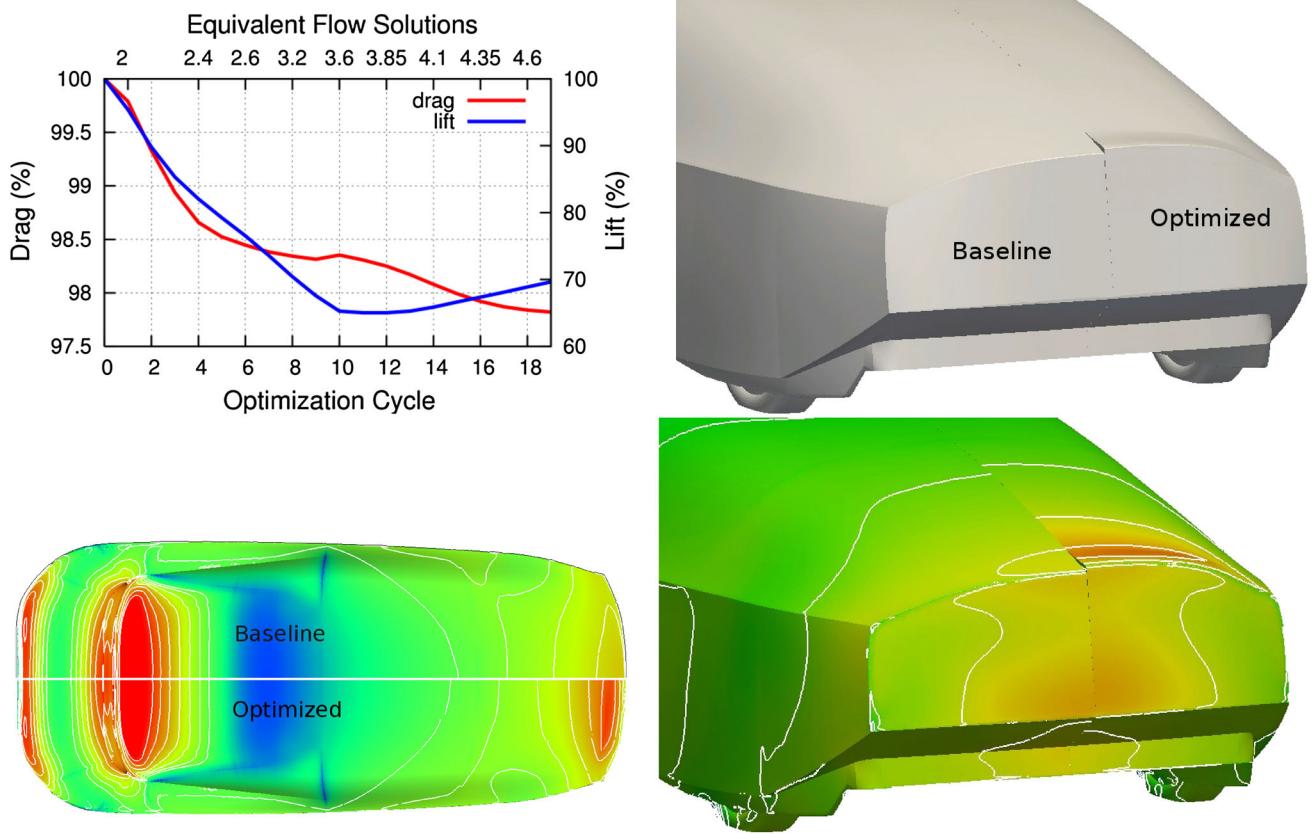


Fig. 15 VW L1 drag minimization: *Top-left* convergence of the drag and lift values. After 19 optimization cycles, a drag decrease by more than 2 % has been achieved, along with a 30 % lift decrease (not included in the objective function). In the *top horizontal axis*, the CPU cost measured in equivalent flow solutions (EFS) is presented. The primal and adjoint equations are solved until convergence only for the first optimization cycle, in order to have an accurate starting point for the optimization procedure. Then, every optimization cycle is initialized with the primal and adjoint fields of the previous cycle and the primal and adjoint equations are solved for just a few iterations. This results to an overall CPU cost of the optimization algorithm that is less than 5 EFS. *Top-right* A close-up view of the rear part of the car. Baseline and optimized geometries are illustrated. As suggested by the drag sensitivity map, Fig. 14-mid, a spoiler has been formed by the optimization

algorithm. A total deformation of less than 20 mm has been induced, indicating that even small changes in the geometry may have a significant impact on the aerodynamic performance of the car. In addition, the fact that the optimization algorithm has, practically, followed the direction suggested by the drag sensitivity map, indicates that the designer can draw valuable information from the adjoint method, even without running an optimization loop. *Bottom* Pressure distribution on the baseline and optimized car geometries (two points of view). While the pressure in the front of the car remains practically unchanged, a significant pressure increase on top and in front of the formed spoiler can be observed. The pressure increase near the spoiler area and on the back of the car leads to a decrease in the drag and lift forces exerted on its surface. Case presented in detail in [120]

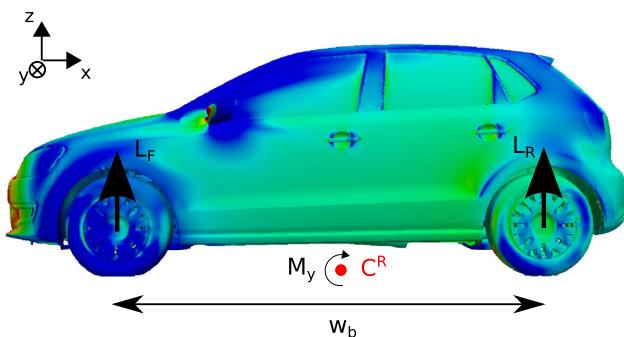


Fig. 16 VW Polo rear lift reduction: Pressure contours along with a schematic of the variables and constants included in the computation of the rear lift force, L_R

$$R\tilde{v} = \frac{\partial\tilde{v}}{\partial t} + \frac{\partial(v_j\tilde{v})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial\tilde{v}}{\partial x_j} \right] - \frac{c_{b2}}{\sigma} \left(\frac{\partial\tilde{v}}{\partial x_j} \right)^2 - \tilde{v} P(\tilde{v}, \tilde{d}) + \tilde{v} D(\tilde{v}, \tilde{d}) = 0 \quad (96)$$

where the model constants and functions are those presented in Sect. 4. The major difference between the RANS and the DES variants of the Spalart–Allmaras model is the turbulence length scale used in the production and dissipation terms. In the RANS variant, the distance, Δ , from the wall boundaries is used as the turbulence length scale whereas, in DES, this scale is given by

$$\tilde{d} = \min(\Delta, C_{DES}d) \quad (97)$$

where $C_{DES} = 0.65$ and d is related to the grid cell in question. In fact d is either the largest dimension of the cell or the cubic root of its volume. Equation 97 practically forces Eq. 96 to act as a RANS model in the regions close to the wall and as an LES model far from the walls [150]. The law of the wall presented in Sect. 5.1 is also utilized for the DES model used herein.

Another critical point to be stressed out, is that the DES model always relies on time-varying flow computations, Eq. 96. From the adjoint point of view, this poses significant challenges. To demonstrate these challenges, the adjoint to the first term in Eq. 96 is presented in brief below

$$\begin{aligned} T_t &= \frac{\delta}{\delta b_n} \int_t \int_{\Omega} \tilde{v}_a \frac{\partial \tilde{v}}{\partial t} d\Omega dt \\ &= \int_t \int_S \tilde{v}_a \frac{\partial \tilde{v}}{\partial t} n_k \frac{\delta x_k}{\delta b_n} dS dt + \int_t \int_{\Omega} \tilde{v}_a \frac{\partial}{\partial t} \left(\frac{\partial \tilde{v}}{\partial b_n} \right) d\Omega dt \\ &= \int_t \int_S \tilde{v}_a \frac{\partial \tilde{v}}{\partial t} n_k \frac{\delta x_k}{\delta b_n} dS dt + \left[\int_{\Omega} \tilde{v}_a \frac{\partial \tilde{v}}{\partial b_n} d\Omega \right]_{t=T} \\ &\quad - \left[\int_{\Omega} \tilde{v}_a \frac{\partial \tilde{v}}{\partial b_n} d\Omega \right]_{t=0} - \int_t \int_{\Omega} \frac{\partial \tilde{v}_a}{\partial t} \frac{\partial \tilde{v}}{\partial b_n} d\Omega dt \end{aligned} \quad (98)$$

where T is the last time step of the primal problem. The third integral on the r.h.s. of Eq. 98 vanishes automatically due to the Dirichlet type initial condition imposed on \tilde{v} at $t = 0$. In order to make Eq. 98 independent of $\partial \tilde{v} / \partial b_n$ at $t = T$, the adjoint turbulence variable initial condition must be

$$\tilde{v}_a(t = T) = 0 \quad (99)$$

The fact that the adjoint initial condition resides in the last time step of the primal problem has significant implications on the solution of the adjoint equations, since these cannot be solved simultaneously with the primal ones. The adjoint to unsteady flow problems is currently receiving a lot of attention. A first approach would be to store all the instants of the primal solution in the memory or the hard drive of the computer and, then, use them for the solution of the adjoint PDEs. Storing the entire time-series of the primal solution to memory is infeasible for industrial problems concerned with flows modeled using the DES model (mesh cell number of the order of 10^7 – 10^8 , stored for 2×10^4 up to 3×10^4 time steps). Since reading from or writing to the hard drive are operations slower than accessing the memory by some orders of magnitude, storing to the hard drive is not an acceptable solution either. An alternative would be to re-compute every time-step of the primal problem, starting from $t = 0$ and, then, based on it, solve the current adjoint time-step. Due to the mesh size and the small time-steps involved, this would be an extremely expensive solution (CPU-wise) and is, thus, discarded. A compromise between excessive memory and CPU-time requirements can be reached via the binomial

check-pointing method [63, 163]. According to the latter, as time integration proceeds, the check-pointing technique maintains a specified number of checkpoints based on an algorithm ensuring that the resulting checkpoints, at any snapshot during the time integration, have the optimal repetition number. In addition, in case of periodic unsteady flows, the adjoint to harmonic balance solvers has been developed [31, 44, 68, 71], greatly decreasing the CPU and memory cost for this family of applications. Lately, the adjoint to reduced order models used to compress the series of primal flow fields computed during an unsteady run has also been explored [20, 161].

Even though the binomial check-pointing method is the best practice available for the solution of the adjoint to unsteady flows, its cost was deemed unaffordable for the industrial applications involving the solution of the DES-based equations around full car geometries at the time this simulation took place. Instead, in order to obtain an approximation to the rear lift sensitivity map, the instantaneous primal fields were time-averaged and, then, used to solve the steady state adjoint equations presented in Sect. 5. The adjoint to the steady state RANS Spalart–Allmaras model was also included, since the RANS and DES variants of the Spalart–Allmaras model are very similar in nature. Using the time-averaged primal fields for the solution of adjoint problems to unsteady flows has also been proposed earlier in the literature [108], as a less expensive, though also less accurate, alternative.

After solving the primal equations and averaging the primal fields, the steady-state adjoint equations are solved, targeting minimum rear lift force. The adjoint equations include the differentiation of the Spalart–Allmaras model PDE and the adjoint wall functions presented in Sect. 5. The sensitivity derivatives are computed through Eq. 87. In Fig. 17-left, the rear lift sensitivities w.r.t. the normal displacements of the VW Polo boundary points are presented. Comments on the physical meaning of the sensitivities can be found in the caption.

For the sake of comparison, the sensitivity derivatives of the resultant lift force, computed through Eq. 76 with $\mathbf{r} = [0, 0, 1]$, are presented in Fig. 17-right. The comparison of the sensitivity derivatives for the lift and rear lift objective functions on the bonnet of the car is rich in physical meaning and analyzed in the caption of Fig. 17. For both objective functions, the quantitative information obtained from the computed sensitivity derivatives is intuitive, i.e. the sensitivity maps point the designer towards the correct direction. This indicates that even though the averaged primal fields were used to solve the steady-state adjoint equations instead of solving the unsteady adjoint equations to the DES model, the present approach can be of practical value for the industrial designer, as it correctly highlights areas with significant potential for optimization.

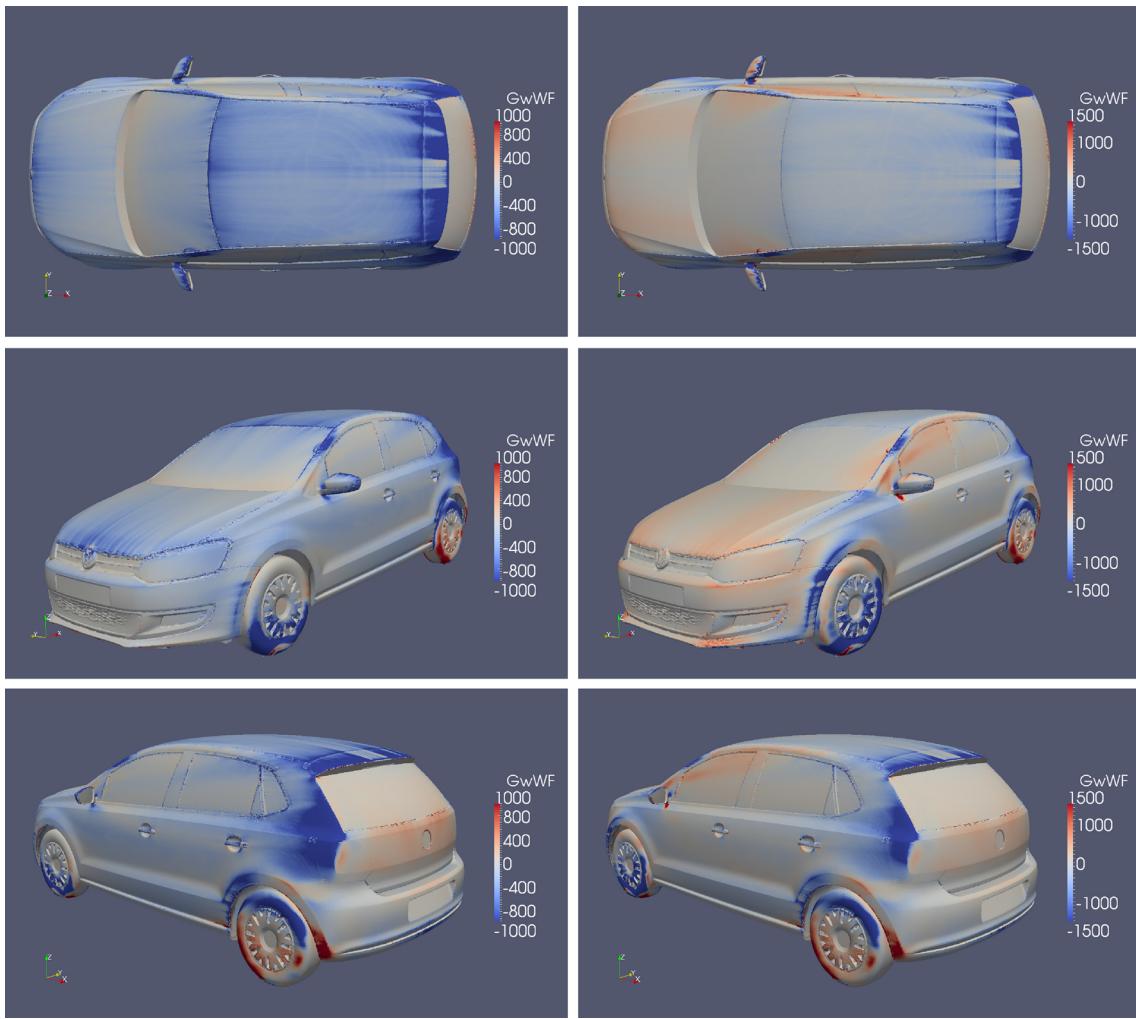


Fig. 17 VW Polo rear lift/resultant lift reduction: Rear lift (left) and resultant lift (right) sensitivity maps plotted over the car surface. Colour map as in Table 1. For both objectives, the spoiler area has an *intense blue colour*, indicating an outward displacement in order to create high pressure areas on the top surface of the car. This will not only reduce lift but will also increase the y component of the moment, causing further decrease in the rear lift objective function. The rear lift sensitivity map also suggests an outward displacement for the bonnet. The flow over

the bonnet will thus be accelerated, leading to lower local pressures. Even though such a displacement is expected to increase the resultant lift force (opposite sensitivity sign for the two objectives over the bonnet area), the local increase in lift will also lead to higher M_y . Since the bonnet is relatively far from the centre of rotation, the moment increase exceeds the lift increase, causing finally a drop in rear lift. (Color figure online)

The cases presented in this section were part of a project funded by Volkswagen AG.

6.3 Cavitation Suppression

In hydraulic turbomachines, cavitation manifests when the local pressure becomes lower than the vapour pressure of the fluid, resulting to the formation of bubbles. The collapse of these bubbles may cause structural problems, through removal of material from the runner and performance degradation, due to the modified shape of the runner [26]. In order to avoid cavitation, the lowest pressure on the runner surface should become greater than the vapour pres-

sure of the fluid (p_c). Thus, the objective is to minimize the part of the runner surface area having $p < p_c$. This can be seen as a max./min. problem since the minimum pressure should be increased in case $p < p_c$; thus, the problem is non-differentiable. Making cavitation suppression treatable with gradient-based optimization methods requires a differentiable objective function which approximates the max./min. nature of the problem. In [126], the minimization of the integral of an appropriate sigmoid function was proposed. This function is given by

$$F_c = \int_{S_{W_p}} F_{sig} dS = \int_{S_W} \left(1 - \frac{1}{1 + e^{k_2(p-p_c)+k_1}} \right) dS \quad (100)$$

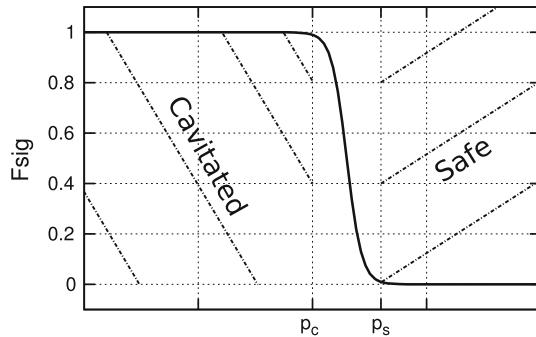


Fig. 18 Optimization of a Francis runner blade targeting at cavitation suppression: the sigmoid function used to convert the otherwise step-wise objective function to a differentiable one. Areas of the runner blade where $p > p_s$ are considered to be safe regarding cavitation and the integrand appearing on the r.h.s. of Eq. 100 has an infinitesimally small value. Areas where $p < p_c$ are considered to be cavitated and are penalized by locally applying $F_{sig} \approx 1$ in Eq. 100. Case presented in detail in [126]

where k_1 and k_2 depend on p_s , p_c and ϵ_{inf} . Here, $p_s (> p_c)$ is a safety pressure threshold to be defined by the designer and ϵ_{inf} stands for a user-defined infinitesimal positive number. If locally $p > p_s$, F_{sig} becomes zero, whereas if $p < p_c$, F_{sig} is close to unity. The integrand F_{sig} in Eq. 100 is plotted in Fig. 18.

The adjoint boundary conditions and the sensitivity derivatives expression can be derived based on the methodology presented in Sect. 5. A steepest descent algorithm is used to drive Eq. 100 towards its minimum value. The pressure distributions on the starting and optimized blade geometries are presented in Fig. 19-top; it can be observed that, in the optimized geometry, the cavitated area practically vanished. In Fig. 19-bottom-left, the sensitivity map of the cavitation related objective function is plotted on the suction side of the starting runner blade. The highest sensitivity values appear upstream and downstream of the initially cavitated area, but not precisely on it. Finally, the cumulative deformation of the blading, resulting after 20 optimization cycles, is plotted in Fig. 19-bottom-right. The largest deformation occurs on the suction side of the blade, just upstream of the initially cavitated area.

7 Topology Optimization

Sections 3–6 were primarily concerned with the development and application of the continuous adjoint method to shape optimization problems. Shape optimization algorithms usually require the parameterization of the aerodynamic shape under consideration using, for instance, Bézier–Bernstein polynomials, NURBS curves or surfaces, splines etc. Parameterization schemes have the advantage of controlling the surface with a relatively low number of design variables and the designed shape is guaranteed to have a certain

level of smoothness (depending on the specific parameterization used). Nevertheless, the choice of the parameterization can limit the solutions/geometries that can be achieved, since only a certain portion of the design space is explored. For these reasons, shape optimization is usually employed to optimize/refine aero/hydrodynamic shapes based on a pre-specified shape pattern without expecting/allowing new “exotic” solutions to come up. On the other hand, when dealing with the initial design of aerodynamic shapes, especially in internal aerodynamics where only the inlet and outlet of the computational domain are defined a priori, the choice of an appropriate parameterization scheme is non-trivial and primarily influences/restricts the outcome of the optimization procedure. An alternative approach to designing aerodynamic shapes is to search for the optimal topology of the flow domain without relying on predetermined parameterization schemes; this approach is referred to as topology optimization.

The notion of topology optimization was initially introduced in structural mechanics by Bendsoe and Kikuchi [17]. There, instead of searching for the optimal shape of a structure by modifying its boundary, the optimization procedure was targeting the optimal material density distribution in order to identify areas in which material should be added so as to increase the structural stiffness under certain load. The design problem was expressed in terms of a real valued continuous function per point, with values ranging from zero (indicating the presence of void/absence of material) to unity (indicating solid). The method has, then, been developed and applied to numerous problems in structural mechanics [7, 18, 24, 35, 36, 141, 146, 148, 162], a great variety of which can be found in [19]. Among other, topology optimization has been used in structural optimization problems dominated by heat conduction, targeting structures with minimum residual heat under the presence of constant heat sources [28, 53, 147, 172].

In fluid mechanics, the same idea was adapted to Stokes flows by Borrvall and Peterson [25], by introducing a real-valued porosity (α) dependent term into the flow equations. Based on the local porosity values, domain areas corresponding to the fluid flow are identified as those with nodal values $\alpha = 0$, or practically, $\alpha \leq \varepsilon$ where ε is a user-defined infinitesimally small positive number. All the remaining areas where $\alpha \neq 0$ or, practically, $\alpha > \varepsilon$ define the part of the domain to be solidified. The sought optimal solid walls correspond to the interfaces between the two aforementioned areas. In Fig. 20, the concept of topology optimization in fluid mechanics is illustrated. 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 $\delta F/\delta\alpha$, since its cost is independent of the

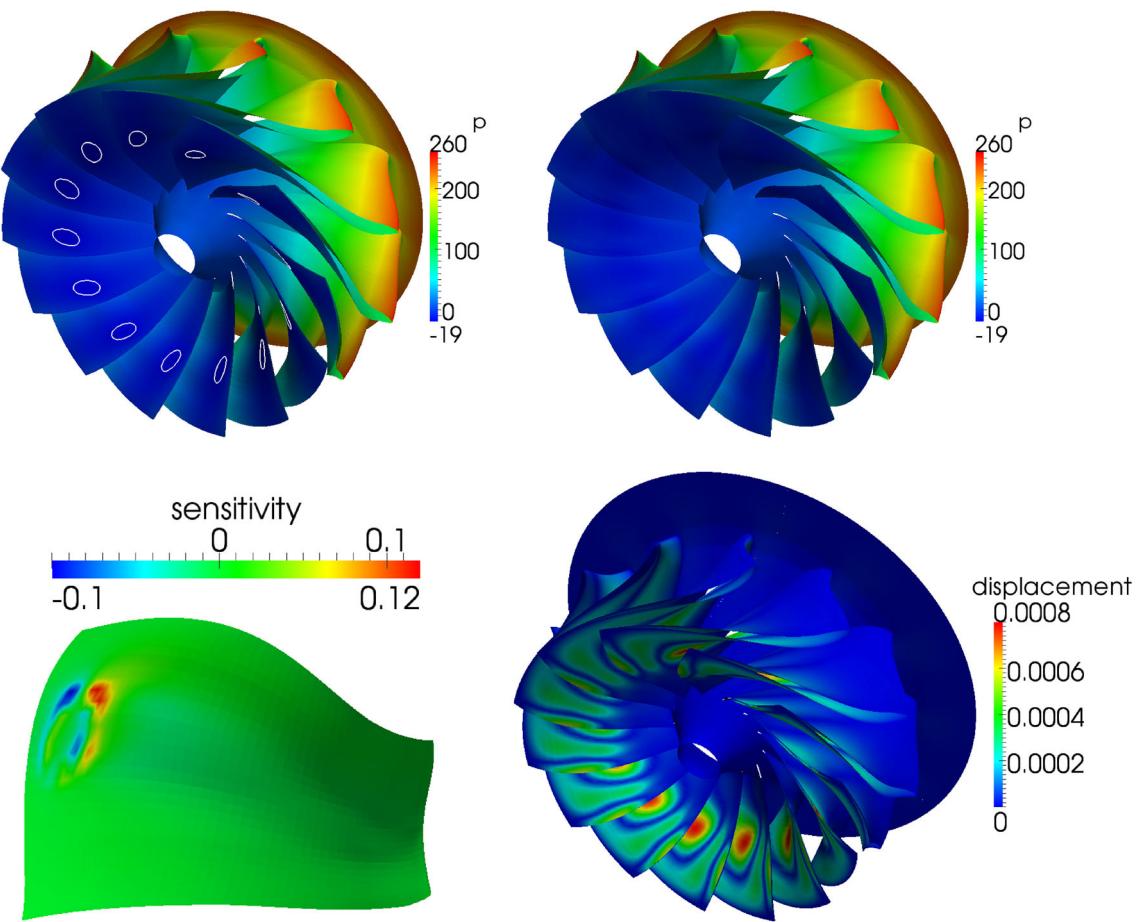


Fig. 19 Optimization of a Francis runner blade targeting at cavitation suppression. *Top* Pressure distribution on the starting (left) and optimized (right) bladings; the white isolines enclose the cavitating area, which has, practically, disappeared in the optimized geometry. *Bottom-left* Sensitivity map for the cavitation suppression, at the first optimiza-

tion cycle, on the suction side of the blade. The sensitivity values on the pressure side have practically zero values and are, thus, not presented herein. *Bottom-right* deformation (normal displacement) of the suction side of the runner blade after 20 optimization cycles. Case presented in detail in [126]

number of design variables. Since [25], a number of relevant publications has appeared for Stokes flows [3, 50, 65, 66, 165]. The notion of topology optimization was extended to laminar flows by Gersborg-Hansen et al. [54] and Olesen et al. [116]. Though the majority of publications concerned with topology optimization in fluid mechanics deals with internal aerodynamics, a recent publication [88], has expanded the method to external aerodynamics, with applications to minimizing the drag and maximizing the lift of airfoils under laminar, incompressible flow conditions.

An alternative to the porosity-based approach for topology optimization problems in fluids, based on the level set method, has been attracting attention during the last few years [4, 34, 41–43, 91, 168, 171]. There, instead of seeking the optimal porosity distribution that minimizes the objective function in hand, the zero values of a level set equation are sought, which clearly mark the boundaries between the flow and solidified areas of the computational domain. The great advantage of the level set approach is the clear dis-

tinction between the fluid and solidified domains. Note that, in the porosity-based approach for the Navier–Stokes equations, “grey” areas that do not belong to either the flow or solidified domain, usually emerge; a user-defined criterion must be applied to track the boundary between the two). In addition, by using the level set method, the fluid equations can be solved only on the fluid part of the computational domain, thus minimizing the CPU time required to numerically solve the problem.

Thermal–fluid interactions have been studied for laminar flows, by interpolating heat transfer coefficients based on local porosity values [40, 100, 169]. Topology optimization has also been applied in other areas of multidisciplinary optimization with applications in fluid–structure interaction [8, 13], aeroelastic [101, 153] and acoustic–structure [170] problems.

In topology optimization problems for turbulent flows, the continuous adjoint method was presented in [118] by making, however, the “frozen turbulence” assumption, i.e. by

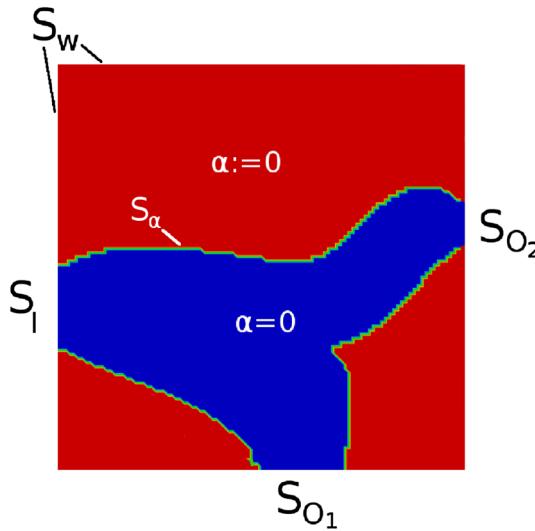


Fig. 20 Schematic representation of a porous media (Ω) domain and its boundaries. S_I is the predefined inlet (or inlets, if more than one), $S_{O_l}, l = 1, L$ are the outlets and predefined S_W are the solid wall boundaries along the porous media domain contour. Topology optimization algorithms seek the optimal field of the real-valued variable porosity over the flow domain, in order to minimize the objective function under consideration. Upon completion of the optimization loop, areas with non-zero porosity values ($\alpha \neq 0$; painted in red) correspond to parts which must be solidified. This is in full agreement with the computed velocity field which, for these areas, is zero. Blue areas indicate the formed flow passage ($\alpha = 0$). The resulting “solid wall” boundaries, S_α , i.e. the desirable outcome of the optimization process, are the interfaces between the two aforementioned areas. (Color figure online)

neglecting the variation in the turbulent viscosity w.r.t. the design variables. Recently [90, 125], the continuous adjoint method was extended to cover 2D turbulent flows, governed by the low-Re Spalart–Allmaras model. This work is extended herein, by adapting the adjoint wall functions technique presented in Sect. 5 to 3D turbulent flows governed by the high-Re variant of the Spalart–Allmaras model. Application of the developed code are initially demonstrated in 2D turbulent flows. Then, two industrial applications of the developed method are presented. The optimization procedure targets either minimum total pressure losses or minimum fluid power dissipation within the computational domain. The Augmented Lagrange Multipliers (ALM) method [115], is used to apply constraints to the optimization problem, without having to predetermine their weight factors; this facilitates the application of topology optimization to large scale industrial applications.

7.1 Primal Equations

In the most general case among those presented in this section, the flow is governed by the steady-state Navier–Stokes equations for incompressible turbulent flows, in which appropriate terms depending on the real-valued porosity field α

have been introduced. The mean-flow primal equations are given by

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \quad (101a)$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{\partial p}{\partial x_i} + \underbrace{\alpha v_i}_{T_{a,v}} = 0 \quad (101b)$$

The extra term in Eq. 101b marked as $T_{a,v}$ is in accordance with the porosity approach for topology optimization problems [118]. In areas where $\alpha \gg$, the first three terms on the r.h.s. of Eq. 101b become negligible and the only term remaining (αv_i) produces a locally zero velocity field in order to satisfy Eq. 101b. These areas correspond to the solidified part of the porous media domain. On the other hand, in areas where $\alpha \approx 0$ (or, practically, $\alpha < \epsilon$), the last term in Eq. 101b vanishes and Eq. 101b returns to its standard form, describing the momentum equilibrium for incompressible flows. These areas correspond to the fluid part of the domain. The goal of the optimization algorithm is to compute the optimal α distribution in order to minimize the objective function in hand.

Following the same line of reasoning, a porosity dependent term is also added to the Spalart–Allmaras model equation, yielding

$$R^{\tilde{v}} = v_j \frac{\partial \tilde{v}}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial \tilde{v}}{\partial x_j} \right] - \frac{c_{b2}}{\sigma} \left(\frac{\partial \tilde{v}}{\partial x_j} \right)^2 - \tilde{v} P(\tilde{v}) + \tilde{v} D(\tilde{v}) + \underbrace{\alpha \tilde{v}}_{T_{a,\tilde{v}}} = 0 \quad (102)$$

This allows the computation of zero \tilde{v} values in the solidified parts, whereas the original Spalart–Allmaras equation is solved in the fluid parts.

The law of the wall is applied to the fluid–solid interfaces, S_a , after following the steps described below:

- Given a porosity distribution, track the fluid–solid interfaces. This is done by employing a simple criterion: let face f belong to cells C_1 and C_2 , Fig. 21. If $\alpha(C_1) \cdot \alpha(C_2) = 0$ and $\alpha(C_1) + \alpha(C_2) > \epsilon$, then, add face f to the list with the internal mesh faces comprising the fluid–solid interface.
- Identify which of the C_1, C_2 cells belongs to the fluid domain by checking whether $\alpha(C_1) > 0$ or $\alpha(C_2) > 0$. Based on the velocity magnitude of this cell, compute the friction velocity, v_t , through Eq. 79.
- Compute the “artificial” $v_t|_f$ value through Eq. 83. It should be noted that, here, $v_i|_f = 0$ is not imposed as a “hard” condition but, practically, results from the fact that f is located at the fluid–solid boundary.

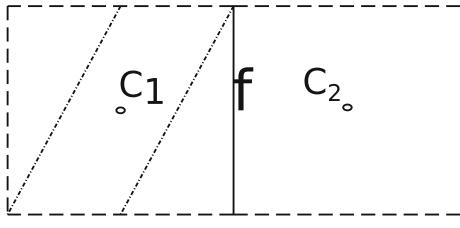


Fig. 21 Internal mesh face f belongs to cells C_1 and C_2 . If C_1 belongs to the solidified domain whereas C_2 to the fluid domain, face f is added to the fluid–solid interface

– Compute the viscous flux at f through Eq. 82.

Regarding the rest of the domain boundaries, typical boundary conditions imposed for internal aerodynamics simulations are used.

Finally, the Hamilton–Jacobi equation, presented in Sect. 4.4.1, is used to compute the distances from the evolving fluid–solid interface, S_a .

7.2 Augmented Objective Function

Let us consider an optimization problem aiming at the minimization of the objective function F subject to E equality constraints. This can be expressed as

$$\min(F), \text{ subject to } c_k = 0, \quad k = 1, \dots, E$$

where c_k is the k -th equality constraint to be satisfied. In what follows, the Augmented Lagrange Multiplier (ALM) method [115], is used to cope with equality constraints. In the ALM method, the Lagrangian function L is defined as

$$L = F - \lambda_k c_k + w c_k^2 \quad (103)$$

where λ_k is the k -th Lagrange multiplier and w a scalar weight factor.

Since $R^P = R_i^v = R^{\tilde{v}} = R^\Delta = 0$, an augmented objective function L_{aug} can be defined and minimized instead. This is defined as

$$L_{aug} = L + \int_{\Omega} q R^P d\Omega + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} \tilde{v}_a R^{\tilde{v}} d\Omega + \int_{\Omega} \Delta_a R^\Delta d\Omega \quad (104)$$

Since, in topology optimization, there are no changes in the computational domain and mesh, Eqs. 7 and 8, degenerate to

$$\frac{\delta\phi}{\delta\alpha} = \frac{\partial\phi}{\partial\alpha}, \quad \frac{\delta_S\phi}{\delta\alpha} = \frac{\partial\phi}{\partial\alpha} \quad (105)$$

i.e., the total and partial derivatives of any flow quantity ϕ coincide.

Taking Eq. 105 into account, the variation in the augmented function w.r.t. the porosity variables is expressed as

$$\begin{aligned} \frac{\delta L_{aug}}{\delta\alpha} = & \frac{\delta F}{\delta\alpha} + (-\lambda_k + 2w c_k) \frac{\delta c_k}{\delta\alpha} + \int_{\Omega} q \frac{\partial R^P}{\partial\alpha} d\Omega \\ & + \int_{\Omega} u_i \frac{\partial R_i^v}{\partial\alpha} d\Omega + \int_{\Omega} \tilde{v}_a \frac{\partial R^{\tilde{v}}}{\partial\alpha} d\Omega + \int_{\Omega} \Delta_a \frac{\partial R^\Delta}{\partial\alpha} d\Omega \end{aligned} \quad (106)$$

The expression for the variation of F w.r.t. α is given by Eq. 19, after setting $b_n = a_n$ and dropping terms that depend on geometry variations. A similar expression can be written for $\delta c_k / \delta\alpha$ as well.

The partial derivatives of the state equations appearing on the r.h.s. of Eq. 106 have been analyzed in detail in Sects. 3 and 4, there w.r.t. the shape design variables, b_n . If the porosity variable α is used instead of b_n , the development presented in those sections remains intact. The only terms that need to be analyzed further are the porosity dependent ones, namely $T_{a,v}$ and $T_{a,\tilde{v}}$ in Eqs. 101b and 102, respectively. Their analysis reads

$$T_{a,v} : \int_{\Omega} u_i \frac{\partial(\alpha v_i)}{\partial\alpha} d\Omega = \int_{\Omega} u_i v_i d\Omega + \int_{\Omega} \alpha u_i \frac{\partial v_i}{\partial\alpha} d\Omega \quad (107a)$$

$$T_{a,\tilde{v}} : \int_{\Omega} \tilde{v}_a \frac{\partial(\alpha \tilde{v})}{\partial\alpha} d\Omega = \int_{\Omega} \tilde{v} \tilde{v}_a d\Omega + \int_{\Omega} \alpha \tilde{v}_a \frac{\partial \tilde{v}}{\partial\alpha} d\Omega \quad (107b)$$

After taking into account the expressions for $\delta F / \delta\alpha$, $\delta c_k / \delta\alpha$, Eq. 107 along with the analysis presented in Sects. 3 and 4, the field adjoint equations, their boundary conditions and the gradient expression can be derived. These are presented in the next sections. The objective and constraint functions used in the applications of Sect. 7.8 are analyzed in Sect. 7.6. Finally, the ALM-based optimization algorithm is presented in Sect. 7.7.

7.3 Field Adjoint Equations

After zeroing the multipliers of $\partial v_i / \partial\alpha$, $\partial p / \partial\alpha$, $\partial \tilde{v} / \partial\alpha$ and $\partial \Delta / \partial\alpha$ in the field integrals arising after the development of Eq. 106, the field adjoint equations read

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \quad (108)$$

$$\begin{aligned} R_i^u = & u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial(v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \\ & + \frac{\partial q}{\partial x_i} + \tilde{v}_a \frac{\partial \tilde{v}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\tilde{v}_a \tilde{v} \frac{C_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} \right) \\ & + \underbrace{\alpha u_i}_{T_{a,u}} = 0 \end{aligned} \quad (109)$$

$$\begin{aligned}
R^{\tilde{v}_a} = & -\frac{\partial(v_j \tilde{v}_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(v + \frac{\tilde{v}}{\sigma} \right) \frac{\partial \tilde{v}_a}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \tilde{v}_a}{\partial x_j} \frac{\partial \tilde{v}}{\partial x_j} \\
& + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\tilde{v}_a \frac{\partial \tilde{v}}{\partial x_j} \right) + \tilde{v}_a \tilde{v} \mathcal{C}_{\tilde{v}} + \frac{\partial v_t}{\partial \tilde{v}} \frac{\partial u_i}{\partial x_j} \\
& \times \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \\
& + (-P + D) \tilde{v}_a + \underbrace{\alpha \tilde{v}_a}_{T_{a,\tilde{v}_a}} = 0
\end{aligned} \tag{110}$$

$$R^{\Delta_a} = -2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) + \tilde{v} \tilde{v}_a \mathcal{C}_{\Delta} = 0 \tag{111}$$

The terms marked as $T_{a,u}$ and T_{a,\tilde{v}_a} in Eqs. 109 and 110, respectively, result from the differentiation of the porosity dependent terms in Eqs. 101b and 102, respectively. Their presence indicates that u_i and \tilde{v}_a will have, practically, zero values in the solidified parts of the porous media domain.

7.4 Adjoint Boundary Conditions

The primal equations of the topology optimization problem, Sect. 7.1, differ from those used in shape optimization, Sect. 4.1, only due to the presence of terms marked as $T_{a,v}$ and $T_{a,\tilde{v}}$ in Eqs. 101b and 102, respectively. Since none of these terms contains a differential operator, they do not contribute any additional term to the adjoint boundary conditions. Another fact that should be taken into consideration is that, in topology optimization problems, the total and partial derivatives of any flow quantity are identical, Eq. 105. Taking these observations into account, one can easily conclude that the boundary conditions of the topology optimization problem based on the high-Re Spalart–Allmaras model are, practically, identical to those presented in Sects. 3.5 and 5.2.1, there for shape optimization problems governed by the same turbulence model.

7.5 Sensitivity Derivatives

After satisfying the field adjoint equations (Sect. 7.3) and their boundary conditions (Sect. 7.4), the remaining terms giving the gradient expression reads

$$\begin{aligned}
\frac{\delta L_{aug}}{\delta \alpha} = & \underbrace{\int_{\Omega} v_i u_i d\Omega}_{\mathcal{SD}_1} + \underbrace{\int_{\Omega} \tilde{v} \tilde{v}_a d\Omega}_{\mathcal{SD}_2} \\
& + \underbrace{\int_{\Omega} \left[\tilde{F}_{\Omega}^{\alpha} + (-\lambda_k + 2w c_k) \dot{c}_{\Omega}^{\alpha} \right] d\Omega}_{\mathcal{SD}_3}
\end{aligned} \tag{112}$$

Terms $\tilde{F}_{\Omega}^{\alpha}$ and $\dot{c}_{\Omega}^{\alpha}$ appearing in \mathcal{SD}_3 denote the direct dependency of the objective and constraint functions on the porosity variable, i.e., the fact that by definition, F and c_k may

depend on the porosity variable, α . Numerical experiments have shown that the second term on the r.h.s. of Eq. 112 is negligible. Nevertheless, term \mathcal{SD}_2 is maintained in all problems examined in Sect. 7.8. Term \mathcal{SD}_1 consists of the inner product of the primal and adjoint velocities. In areas where v_i and u_i form an obtuse angle, the local sensitivity value is negative, leading to an increase in the local porosity values, thus locally blocking the flow. On the contrary, areas where v_i and u_i form an acute angle, the local porosity values tend to decrease. This holds irrespective of the objective function in hand. Term \mathcal{SD}_3 depends on the objective and constraint functions used and its impact is examined in the applications section (Sect. 7.8).

7.6 Objective and Constraint Functions

The adjoint formulation presented in the previous sections was generic and can be adapted to any objective function and equality constraint. The two objective functions considered herein are

$$F_{p_t} = - \int_{S_I} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS - \int_{S_O} \left(p + \frac{1}{2} v_k^2 \right) v_i n_i dS \tag{79}$$

$$F_{PL} = \int_{\Omega} \left[\frac{(v + v_t)}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)^2 + \alpha v_i^2 \right] d\Omega \tag{113}$$

and correspond to the volume-averaged total pressure losses and the fluid power dissipation within the porous media domain. Both functions should be minimized. An interesting comparison between F_{p_t} and F_{PL} is presented in “Appendix”, where it is proved that the two functions are, practically, identical when the inlet and outlet boundaries are clear from intense velocity gradients.

Finally, the constraint used in this section deals with the volume of the overall porous media domain covered by the fluid. If V_{tar} is the desired percentage of the porous media volume to be occupied by the fluid, the corresponding constraint can be written as

$$c = \left[\frac{\int_{\Omega} \left(1 - \frac{\alpha}{\alpha_{max}} \right) d\Omega}{\int_{\Omega} d\Omega} - V_{tar} \right]^2 \tag{114}$$

where α_{max} is the maximum value α can take on. Comments about the value of α_{max} can be found in Sect. 7.7. Since c does not include flow quantities, its differentiation contributes directly to the sensitivity derivatives expression but not to the adjoint mean-flow equations or their boundary conditions.

7.7 Optimization Algorithm

In the most general case (turbulent flow, constrained optimization), the ALM-based algorithm [115], comprises the following steps:

1. Initialize the porosity field α and the Lagrangian multipliers λ_k (the latter with zero values). The penalty factor w takes on a user-defined value. A constant γ with which the value of w is multiplied in each optimization cycle along with the maximum allowed w^{max} value are also defined. The initial value of w along with the γ and w^{max} values are quantities linked with the imposition of constraints that the user has to specify.
2. Solve the state equations in which porosity-dependent terms have been added to the turbulence model and momentum equations (Sect. 7.1).
3. Compute the values of the objective and constraint functions (Sect. 7.6). Terminate the algorithm if L (Eq. 103) is less than a threshold value.
4. Solve the adjoint equations, including the adjoint to the turbulence model equations.
5. Compute the sensitivities $\delta L/\delta\alpha$ (Eq. 112).
6. Update the porosity field using the steepest descent formula

$$a^{n+1} = a^n - \eta \left. \frac{\delta L}{\delta \alpha} \right|_n \quad (115)$$

where n is the optimization cycle index and η a user-defined step. The porosity value is allowed to vary within the range $[0, \alpha_{max}]$. The maximum porosity value, α_{max} , needs to have a sufficiently large value in order to zero the velocity values in the solidified parts of the porous media domain. In [116], the Darcy dimensionless number is introduced as

$$Da = \frac{\nu}{\alpha_{max} l^2} \quad (116)$$

where l is a characteristic length. The Darcy number expresses the ratio of the viscous to the porous friction forces (αv_i in Eq. 101b). Solving Eq. 116 for α_{max}

$$\alpha_{max} = \frac{\nu}{Da l^2} \quad (117)$$

The solidified part of the domain becomes almost impermeable for $Da \leq 10^{-5}$ [116]. Taking this into consideration, the α_{max} value, common for every cell of the porous media domain, is computed through Eq. 117.

7. Update λ_k and w as follows

$$\lambda_k^{n+1} = \lambda_k^n - 2w c_k^n \quad (118)$$

$$w^{n+1} = \min(\gamma w^n, w^{max}) \quad (119)$$

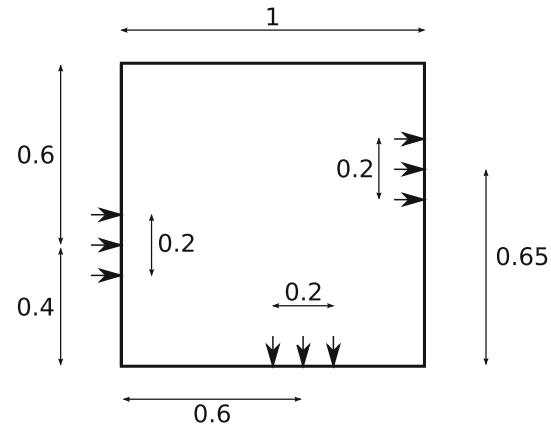


Fig. 22 Topology Problem I: Dimensions, in meters, of the porous media domain and its boundaries. A square porous media domain containing a single inlet and two asymmetrically positioned outlets is used. The inlet height is used as the characteristic length of the problem

8. Move to step 2.

7.8 Applications

The ALM-based optimization algorithm presented in Sect. 7.7 is applied to three topology optimization problems:

- *Problem I*: Design of a 2D duct system at turbulent flow conditions. Both objective functions presented in Sect. 7.6 are used, along with the constraint of Eq. 114.
- *Problem II*: Unconstrained optimization of an air-conditioning duct of a passenger car at turbulent flow conditions. The optimization is targeting minimum F_{p_t} .
- *Problem III*: Constrained topology optimization of a plenum chamber mounted on a school formula car at turbulent flow conditions. The optimization is targeting minimum F_{PL} and a certain percentage of the porous media domain to be covered by fluid (constraint c , Eq. 114).

7.9 Problem I: Design of a 2D Duct System

7.9.1 Minimum F_{p_t}

In this problem, topology optimization based on the high-Re Spalart–Allmaras turbulence model is used to design a duct between a single inlet and two outlets of a square porous media domain, Fig. 22. Here, the optimization aims at minimum total pressure losses ($F = F_{p_t}$, Eq. 78) whereas 30 % of the porous media domain must also be occupied by fluid ($c = 0$ with $V_{tar} = 0.3$, Eq. 114). The initial structured mesh consists of 200×200 cells and the flow Reynolds number based on the inlet height is $Re = 7,000$. The porosity variable field α and the Lagrange multiplier λ are initialized to zero.

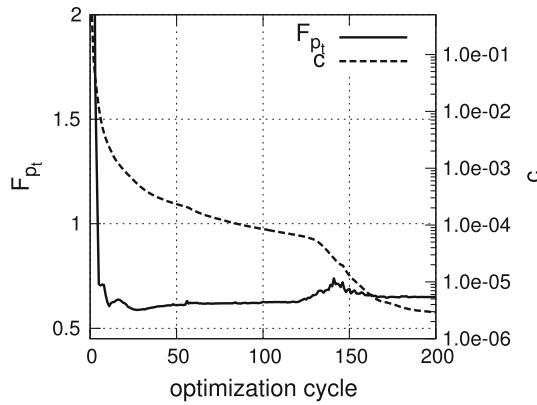


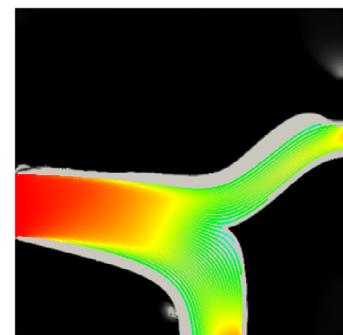
Fig. 23 Topology Problem I: Convergence of the constrained optimization problem. F_{p_t} values correspond to the *left vertical axis* whereas c values are plotted over the *right vertical axis*, using a logarithmic scale

After executing 200 steps of the optimization algorithm presented in Sect. 7.7, an optimal porosity field is obtained, Fig. 24a. It should be noted that the primal and adjoint equations are not solved until full convergence, but only 10 iterations are performed for each system of equations in each optimization cycle. This “one-shot” like approach has proved to be very efficient in topology optimization problems, yielding the optimal solution to the present problem in less than 13 min, using 8 processors of an Intel Xeon E5620 CPU (2.40 GHz). The convergence of the objective and constraint functions is presented in Fig. 23.

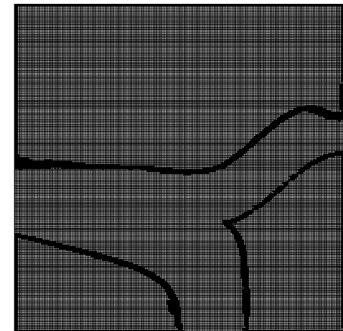
Mesh adaptation is used during the optimization to refine the evolving fluid–solid interface as much as possible. The mesh of the optimized geometry is illustrated in Fig. 24b and consists of 53,384 cells. The mesh adaptation criterion is based on the distance from the fluid–solid interface. In particular, all faces belonging to cells at a distance less than 0.005 m from the evolving interface are bisected during each optimization cycle. To avoid an excessively fine mesh, cells with a distance longer than 0.005 m from the interface are unrefined during each optimization step. The criterion with which the fluid–solid interface is tracked has been analyzed in Fig. 21. The primal and adjoint velocity fields of the optimized geometry are presented in Fig. 24c and d, respectively.

A close-up view near the V-shaped crotch of the duct is presented in Fig. 25, along with the local primal and adjoint velocity vectors. Since these vectors form acute angles, term \mathcal{SD}_1 in Eq. 112 has positive values, retaining the local porosity values to zero ($\alpha \in [0, \alpha_{max}]$), as discussed in Sect. 7.7). Since F_{p_t} is defined along S_I and S_O , $\dot{F}_Q^\alpha = 0$ and term \mathcal{SD}_3 in Eq. 112 degenerates to

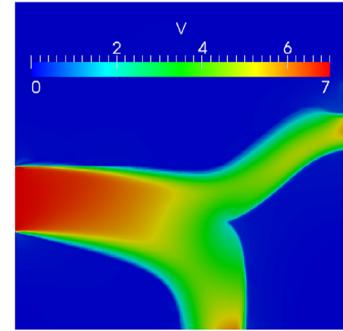
$$\mathcal{SD}_3 = \int_{\Omega} (-\lambda + 2wc) \dot{c}_Q^\alpha d\Omega$$



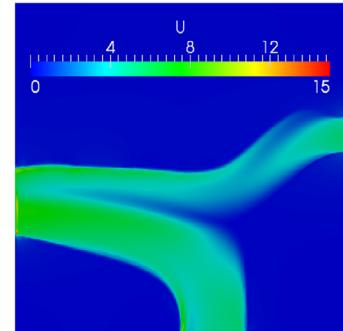
(a) α , optimized duct



(b) adapted mesh, optimized duct



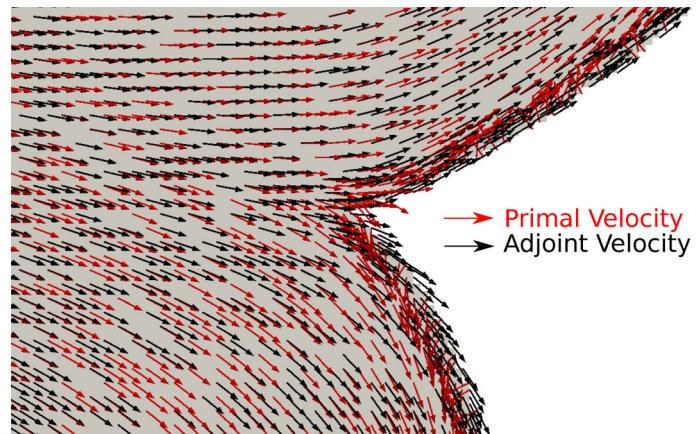
(c) $|v|$, optimized duct



(d) $|u|$, optimized duct

Fig. 24 Topology Problem I: design of a duct for minimum F_{p_t} subject to $c = 0$ with $V_{tar} = 0.3$. **a** Porosity field, α , defining the optimal geometry along with the primal velocity streamlines. Dark areas correspond to the solidified parts of the porous media domain whereas bright areas correspond to the fluid domain. **b** Computational mesh at the last optimization step, adapted to the fluid–solid interface. **c** Primal velocity field. **d** Adjoint velocity field

Fig. 25 Topology Problem I: design of a duct for minimum F_{p_i} subject to $c = 0$ with $V_{tar} = 0.3$. Primal and adjoint velocity vectors focused on the V-shaped crotch designed by the optimization algorithm. Vectors \mathbf{v} and \mathbf{u} form acute angles which, based on the comments made in Sect. 7.5, prevents any further solidification of the fluid domain. The length of all vectors has been made equal for visualization purposes. Red arrows correspond to the primal velocity vectors whereas black arrows to the adjoint ones. (Color figure online)



Constraint c is always positive since it is defined as the squared difference of the current percentage of Ω occupied by fluid from the desirable percentage. Taking into consideration Eq. 118, it is evident that λ takes on negative values (since it is initialized to zero and in each update, a positive quantity is subtracted from it). This means that $(-\lambda + 2wc)$ is positive at each optimization step. Finally, it can be concluded that $\hat{c}_\Omega^\alpha = -\frac{2}{\alpha_{max} \int_\Omega d\Omega} \left[\frac{\int_\Omega \left(1 - \frac{\alpha}{\alpha_{max}}\right) d\Omega}{\int_\Omega d\Omega} - V_{tar} \right]$ is negative as long as the percentage of the porous media domain covered by the fluid is larger than the target one. In the present case, this holds for the majority of the optimization steps since α was initialized to zero. In view of the above, it can be concluded that, given a porosity distribution, \mathcal{SD}_3 has a constant negative value at all the cells of the porous domain, i.e. it is attempting to uniformly solidify all the cells in the domain. The cells that eventually remain open to form the fluid domain are those where $u_i v_i$ has a positive value that exceeds $(-\lambda + 2wc)\hat{c}_\Omega^\alpha$.

7.9.2 Minimum F_{PL}

The same mesh and flow conditions are used to solve an optimization problem targeting minimum F_{PL} (Eq. 113) subject to $c = 0$ with $V_{tar} = 0.3$. It should be noticed that since F_{PL} is defined as a volume integral containing v_i and v_t , extra F_{PL} -based terms should be added to the adjoint momentum and turbulence model equations for this objective. For this case as well, the porosity field and the Lagrange multiplier λ were initialized to zero. The convergence of the objective and constraint values is illustrated in Fig. 26. After 200 optimization steps using an “one shot”-like technique, the optimized geometry presented in Fig. 27a is achieved. The mesh adapted to the fluid–solid interface is presented in Fig. 27b. In Fig. 27c, d, the primal and adjoint velocity magnitudes are shown.

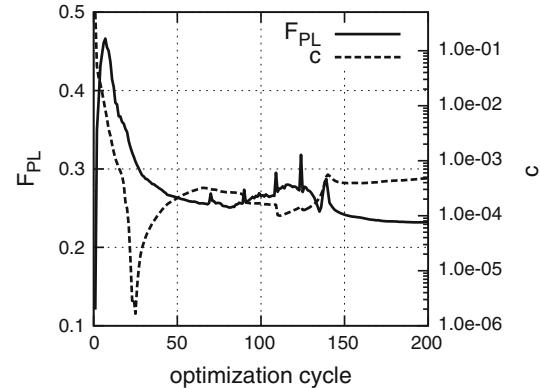


Fig. 26 Topology Problem I: Convergence of the constrained optimization problem targeting minimum F_{PL} subject to the constraint $c = 0$ with $V_{tar} = 0.3$

7.10 Problem II: Optimization of an Air-Conditioning Duct

In this section, the optimization of an air-conditioning duct, transferring air from the main console to the back seats of a passenger car is investigated, targeting minimum F_{p_i} . The case is part of the E.C. project “Flowhead” and the mesh was generated by Stefan Brück and Carsten Ehlers, EKAC, Technical Development, Volkswagen AG. The duct geometry is presented in Fig. 28-top. The mesh consists of the inlet and outlet ducts that remain unchanged through the optimization process and the main body of the duct which coincides with the design space. The hole in the middle of the domain is the space reserved for the gear-shift level which intersects with the duct. The mesh consists of about 5.5 million cells, the Reynolds flow number based on the inlet hydraulic diameter is $Re \approx 3,000$ and the flow is modeled using the high-Re Spalart–Allmaras turbulence model. Streamlines of the primal velocity, obtained after solving the flow equations in the starting geometry, are presented in Fig. 29. Intense flow recirculation occurs close to the inlet duct, where the cross-section area increases abruptly. In addition, there is no clear

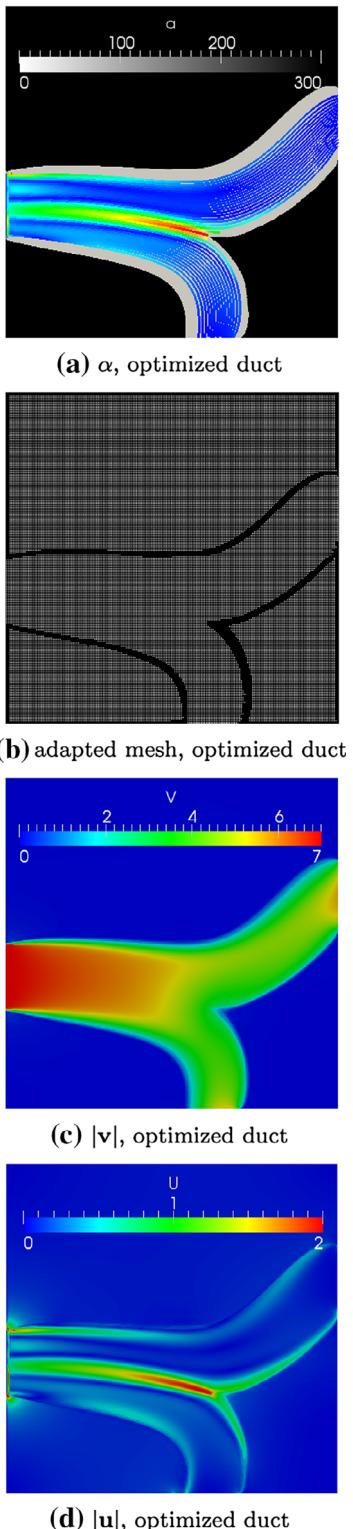


Fig. 27 Topology Problem I: design of a duct for minimum F_{PL} subject to $c = 0$ with $V_{tar} = 0.3$. **a** Porosity field, α , defining the optimal geometry along with the adjoint velocity streamlines. Colour map as in Fig. 24a. **b** Computational mesh at the last optimization step, adapted to the fluid–solid interface. **c** Primal velocity field. **d** Adjoint velocity field

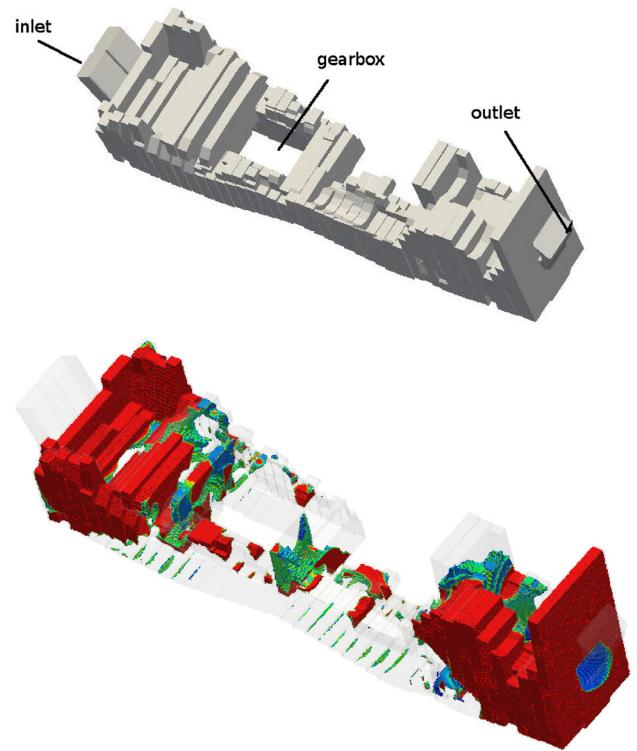


Fig. 28 Topology Problem II: Topology optimization of an air-conditioning duct targeting $\min. F = F_{PL}$. Test case from the E.C. project “Flowhead”. **Top** the geometry of the duct. The porosity variable at the inlet and outlet ducts remains zero during the optimization whereas this was allowed to vary in the rest of the porous media domain. **Bottom** porosity field at the last optimization cycle. *Red areas* correspond to the solidified part of Ω . The topology optimization has solidified areas where intense flow recirculation appeared in the starting geometry. (Color figure online)

path to lead the flow to the outlet duct. As a result, the flow recirculates close to the outlet area, leading to high losses.

After running the optimization algorithm presented in Sect. 7.7, the porosity field illustrated in Fig. 28-bottom is obtained. For this application, the variable porosity field was initialized to zero. It can be observed that the topology optimization algorithm made the cross-section of the duct near the inlet smaller, in order to minimize recirculation. In addition, a clear path connecting the body of the design space with the outlet ducts has been formed, in order to smoothly steer the flow towards the back seats of the car. The primal velocity streamlines computed in the optimized geometry are presented in Fig. 30. The optimization led to a 45 % decrease in total pressure losses.

7.11 Problem III: Optimization of a Plenum Chamber

In this section, the optimization of a plenum chamber, positioned between the air intake duct and the cylinders of a school formula car is investigated, targeting minimum F_{PL} .

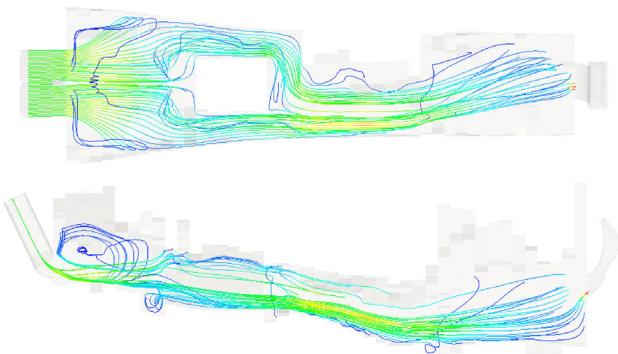


Fig. 29 Topology Problem II: Topology optimization of an air-conditioning duct targeting $\min. F = F_{p_t}$. Primal velocity streamlines computed in the starting geometry



Fig. 30 Topology Problem II: Topology optimization of an air-conditioning duct targeting $\min. F = F_{p_t}$. Primal velocity streamlines computed in the optimized geometry

The role of the plenum chamber is to decelerate the flow coming from the air intake duct in order to feed the cylinders with high static pressure air. In addition, it provides a large volume buffer against pressure changes over time. Due to the high velocity air entering the plenum and its geometry (which is, practically, a diffuser with a large ratio of the wide to the narrow cross sections), intense flow recirculation occurs, which leads to high energy losses. Here, the objective is to minimize the power dissipation within the plenum, under the constraint of a certain percentage of the volume of Ω to be occupied by fluid; this constraint is imposed by the rules of the school formula car contest.

The flow Reynolds number based on the inlet diameter is $Re = 5 \times 10^4$ and the hybrid mesh used consists of about 3 million cells. The flow is modeled using the RANS equations coupled with the high-Re Spalart–Allmaras model. The plenum geometry is presented in Fig. 31.

Primal velocity streamlines are illustrated in Fig. 33, where intense flow separation is observed. The convergence of the topology optimization algorithm targeting $\min. F_{PL}$ and $c = 0$ with $V_{tar} = 0.5$ is presented in Fig. 32. A 29 % reduction in F_{PL} has been achieved in less than 40 optimization cycles. For this case as well, the variable porosity field

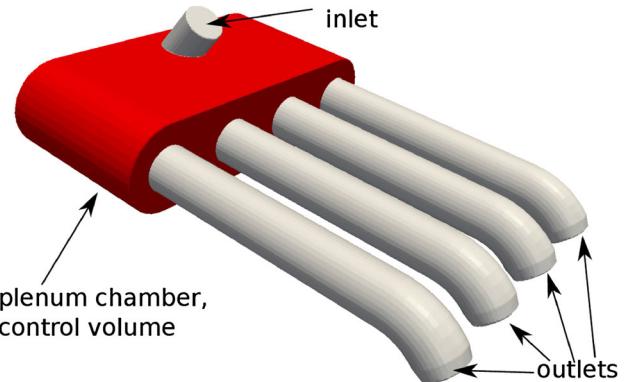


Fig. 31 Topology Problem III: Topology optimization of a plenum chamber targeting $\min. F_{PL}$ and $c = 0$ with $V_{tar} = 0.5$. Plenum geometry. The body of the plenum, coloured in red, is the design space where α is allowed to vary during the optimization. On the contrary, α remains zero in the inlet and outlet ducts. (Color figure online)

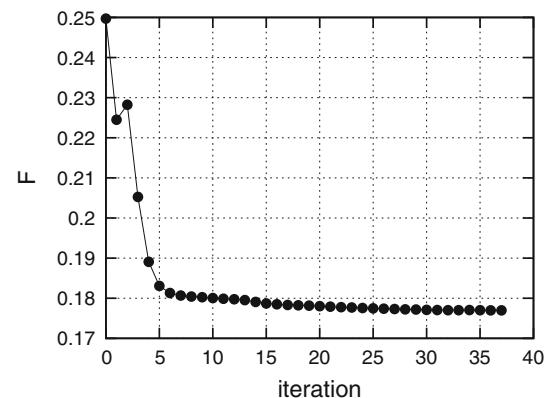


Fig. 32 Topology Problem III: Topology optimization of a plenum chamber targeting $\min. F_{PL}$ subject to $c = 0$ with $V_{tar} = 0.5$. Convergence of the objective function. A 29 % reduction in F_{PL} was achieved after a 12 h computation on 40 cores of 5 Intel Xeon E5620 CPUs (2.40 GHz)

was initialized to zero. Primal velocity streamlines computed in the optimized geometry are illustrated in Fig. 34.

8 Conclusions

The development and application of the continuous adjoint method to turbulent incompressible flows for shape and topology optimization problems have been presented. In particular, the differentiation of the low- and high-Reynolds number variants of the Spalart–Allmaras model as well as the differentiation of the Launder–Sharma $k - \epsilon$ model have been analyzed. The key findings from the cases studied are that (a) depending on the case, the omission of the turbulence model differentiation can significantly damage the accuracy of the computed sensitivities, even by computing the wrong sensitivity sign, (b) the convergence of the optimization algorithm minimizing the objective value can be accelerated when using



Fig. 33 Topology Problem III: Topology optimization of a plenum chamber targeting $\min. F_{PL}$, subject to $c = 0$ with $V_{tar} = 0.5$. Primal velocity streamlines computed in the starting geometry (two views). Streamlines coloured based on the magnitude of the primal velocity. (Color figure online)

accurate sensitivity derivatives resulting by differentiating the turbulence model PDEs, (c) it is highly recommended to also consider the differentiation of the distance field when dealing with the adjoint to the Spalart–Allmaras turbulence model, (d) the incomplete differentiation of a high-Reynolds number turbulence model, with wall functions, may lead to worse results than the complete omission of its differentiation, indicating the importance of developing and using the so called “adjoint wall functions”. The methods presented herein can be extended to the differentiation of other turbulence models, with or without the application of the wall functions technique.

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). In particular, the authors would like to acknowledge 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. Research related to topology optimization was partially supported by a Basic Research Project funded by the National Technical

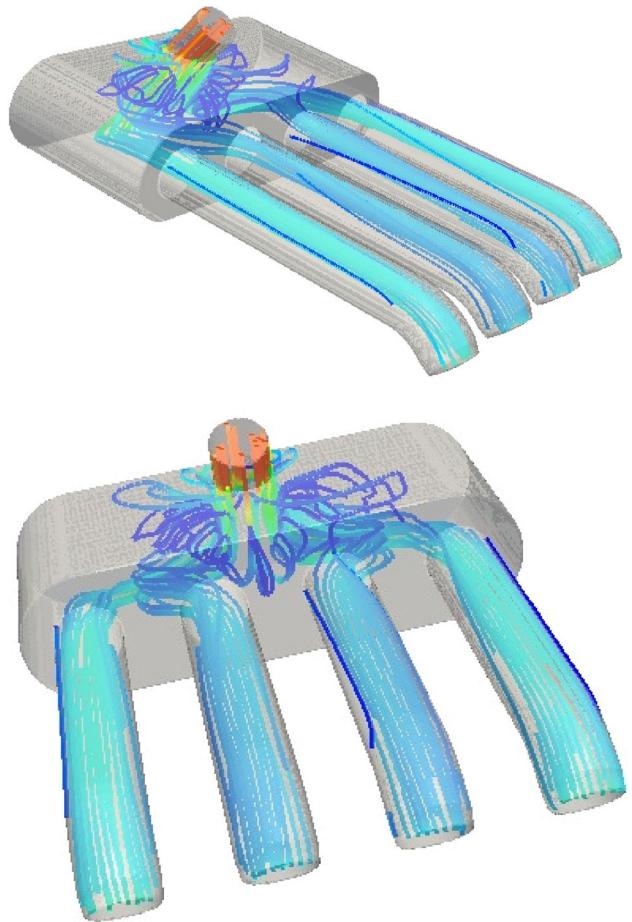


Fig. 34 Topology Problem III: Topology optimization of a plenum chamber targeting $\min. F_{PL}$, subject to $c = 0$ with $V_{tar} = 0.5$. Primal velocity streamlines computed in the optimized geometry (two views). Streamlines coloured based on the magnitude of the primal velocity. (Color figure online)

University of Athens. The authors would like to thank the partners of the “RBF4AERO” project for their permission to publish results on the glider geometry.

9 Discussion on the Total Pressure Losses and Fluid Power Dissipation as Objective Functions

In Sect. 7, the volume-averaged total pressure losses and the fluid power dissipation are used as the objective functions for problems related to internal aerodynamics.

The relation between Eqs. 78 and 113 for topology optimization problems is scrutinized in this “Appendix”. If term αv_i^2 is neglected from Eq. 113 (along with the porous friction term, $T_{a,v}$, in Eq. 101b) the proof that follows holds for shape optimization problems as well.

The total kinetic energy of an incompressible fluid is

$$E_{kin} = \frac{1}{2} \int_{\Omega} v_i^2 d\Omega \quad (120)$$

while its time derivative reads

$$\frac{\partial E_{kin}}{\partial t} = \int_{\Omega} v_i \frac{\partial v_i}{\partial t} d\Omega \quad (121)$$

Using the momentum equations in which the porosity dependent term αv_i has been added, the integrand on the r.h.s. of Eq. 121 is written as

$$v_i \frac{\partial v_i}{\partial t} = \underbrace{v_i v_j \frac{\partial v_i}{\partial x_j}}_{term1} + \underbrace{v_i \frac{\partial p}{\partial x_i}}_{term2} - \underbrace{v_i \frac{\partial}{\partial x_j} [(v + v_t) s_{ij}]}_{term3} + \alpha v_i^2 \quad (122)$$

where the strain tensor, s_{ij} , is given by

$$s_{ij} = \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (123)$$

By taking into account the continuity equation, the development of the terms appearing on the r.h.s. of Eq. 122 yields

$$term1 : v_i v_j \frac{\partial v_i}{\partial x_j} = v_j \frac{1}{2} \frac{\partial (v_i^2)}{\partial x_j} = \frac{1}{2} \frac{\partial (v_j v_i^2)}{\partial x_j} \quad (124)$$

$$term2 : v_j \frac{\partial p}{\partial x_j} = \frac{\partial (v_j p)}{\partial x_j} \quad (125)$$

$$term3 : v_i \frac{\partial}{\partial x_j} [(v + v_t) s_{ij}] = \frac{\partial}{\partial x_j} [(v + v_t) v_i s_{ij}] - (v + v_t) s_{ij} \frac{\partial v_i}{\partial x_j} \quad (126)$$

After substituting Eqs. 124–126 into Eq. 122, we get

$$v_i \frac{\partial v_i}{\partial t} = \frac{\partial}{\partial x_j} \left[v_j \left(\frac{1}{2} v^2 + p \right) \right] - \frac{\partial}{\partial x_j} [(v + v_t) v_i s_{ij}] + (v + v_t) s_{ij} \frac{\partial v_i}{\partial x_j} + \alpha v_i^2 \quad (127)$$

The Frobenius inner product of the strain tensor with the velocity gradient can be written as

$$\begin{aligned} \frac{\partial v_i}{\partial x_j} s_{ij} &= \frac{\partial v_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \\ &= \frac{1}{2} \frac{\partial v_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} + \frac{1}{2} \frac{\partial v_j}{\partial x_i} \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \frac{\partial v_j}{\partial x_i} \\ &= \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)^2 = \frac{1}{2} s_{ij}^2 \end{aligned} \quad (128)$$

Substituting Eq. 128 into Eq. 127, we get

$$\begin{aligned} v_i \frac{\partial v_i}{\partial t} &= \frac{\partial}{\partial x_j} \left[v_j \left(p + \frac{1}{2} v_i^2 \right) \right] - \frac{\partial}{\partial x_j} [(v + v_t) v_i s_{ij}] \\ &\quad + \frac{(v + v_t)}{2} s_{ij}^2 + \alpha v_i^2 \end{aligned} \quad (129)$$

For a steady state problem, the time derivative on the l.h.s. of Eq. 129 is zero. Taking this into consideration and using the Green–Gauss theorem for the conservative terms, the integration of Eq. 129 over Ω yields

$$\begin{aligned} - \int_S \left[\left(p + \frac{1}{2} v_i^2 \right) \right] v_j n_j dS &= \int_{\Omega} \left[\frac{(v + v_t)}{2} s_{ij}^2 + \alpha v_i^2 \right] d\Omega \\ &\quad - \int_S (v + v_t) v_i s_{ij} n_j dS \end{aligned} \quad (130)$$

The boundary integrals in Eq. 130 are zero along S_W due to the no-slip velocity boundary condition. Thus, Eq. 130 becomes

$$\begin{aligned} - \int_{S_{I,O}} \left[\left(p + \frac{1}{2} v_i^2 \right) \right] v_j n_j dS \\ = \int_{\Omega} \left[\frac{(v + v_t)}{2} s_{ij}^2 + \alpha v_i^2 \right] d\Omega - \int_{S_{I,O}} (v + v_t) v_i s_{ij} n_j dS \end{aligned} \quad (131)$$

If the flow at S_I and S_O is sufficiently free from intense flow gradients, it can be assumed that last integral on the r.h.s. of Eq. 130 is negligible.² Under this assumption and after taking into consideration Eq. 123, we get

$$- \int_{S_{IO}} \left[\left(p + \frac{1}{2} v^2 \right) v_i n_i \right] dS \approx \int_{\Omega} \left[\frac{(v + v_t)}{2} s_{ij}^2 + \alpha v_i^2 \right] d\Omega \quad (132)$$

or

$$F_{p_t} \approx F_{PL} \quad (133)$$

References

1. www.beta-cae.gr/ansa.htm
2. www.volkswagen.co.uk/about-us/news/282/volkswagen-unveils-the-x11-super-efficient-vehicle-in-qatar
3. Aage N, Poulsen T, Gersborg-Hansen A, Sigmund O (2008) Topology optimization of large scale Stokes flow problems. Struct Multidiscip Optim 35:175–180
4. Abdelwahed M, Hassine M, Masmoudi M (2009) Optimal shape design for fluid flow using topological perturbation technique. J Math Anal Appl 356:548–563

² Though this is very close to reality in shape optimization problems where S_I and S_O are usually far from the controlled shape, in topology optimization this is not always the case. Nevertheless, Eq. 130 gives an interesting relation between the total pressure losses and fluid power dissipation.

5. Alba E, Dorronsoro B (2008) Cellular genetic algorithms. Springer, Berlin
6. Alba E, Troya J (2001) Analyzing synchronous and asynchronous parallel distributed genetic algorithms. *Future Gener Comput Syst* 17(4):451–465
7. Amstutz S, Andra H (2006) A new algorithm for topology optimization using a level-set method. *J Comput Phys* 216:573–588
8. Andersen C, Sigmund O (2011) Saturated poroelastic actuators generated by topology optimization. *Struct Multidisc Optim* 43:693–706
9. Anderson W, Bonhaus D (1997) Aerodynamic design on unstructured grids for turbulent flows. NASA Technical Memorandum 112867
10. Anderson W, Bonhaus D (1999) Airfoil design on unstructured grids for turbulent flows. *AIAA J* 37(2):185–191
11. Anderson W, Venkatakrishnan V (1997) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA Paper* 06(43)
12. Anderson W, Venkatakrishnan V (1999) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Comput Fluids* 28:443–480
13. Andreasen C, Sigmund O (2013) Topology optimization of fluid-structure-interaction problems in poroelasticity. *J Comput Methods Appl Mech Eng* 258:55–62
14. Asouti V, Giannakoglou K (2009) Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Eng Optim* 41(3):241–257
15. Bäck T (1996) Evolutionary algorithms in theory and practice. Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, Oxford
16. Barr AH (1984) Global and local deformations of solid primitives. *SIGGRAPH Comput Graph* 18(3):21–30
17. Bendsoe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *J Comput Methods Appl Mech Eng* 71:197–224
18. Bendsoe M, Sigmund O (1999) Material interpolation schemes in topology optimization. *Arch Appl Mech* 69:635–654
19. Bendsoe M, Sigmund O (2004) Topology optimization: theory, methods and applications. Springer, Berlin
20. Beran P, Stanford B, Kurdi M (2010) Sensitivity analysis for optimization of dynamic systems with reduced order modeling. In: 48th AIAA sciences meeting including the new horizons forum and aerospace exposition, Orlando, FL
21. Bertsekas D (1996) Constrained optimization and lagrange multiplier methods, 1st edn. Athena Scientific, Belmont
22. Bertsekas D (1999) Nonlinear programming, 2nd edn. Athena Scientific, Belmont
23. Bischof C, Carle A, Khademi P, Mauer A (1994) The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs. Report CRPC-TR94491, Center for Research and Parallel Computation, Rice University
24. Borrvall T, Petersson J (2001) Large-scale topology optimization in 3D using parallel computing. *Comput Methods Appl Mech Eng* 190:6201–6229
25. Borrvall T, Peterson J (2003) Topology optimization if fluids in stokes flow. *Int J Numer Methods Fluids* 41:77–107
26. Brennen C (1995) Cavitation and bubble dynamics. Oxford Engineering Science Series. Oxford University Press, Oxford
27. Brezillon J, Dwight R (2005) Discrete adjoint of the Navier–Stokes equations for aerodynamic shape optimization. In: Evolutionary and deterministic methods for design, EUROGEN. Munich
28. Bruns T (2007) Topology optimization of convection-dominated, steady-state heat transfer problems. *Heat Mass Transf* 50:2859–2873
29. Bueno-Orovio A, Castro C, Palacios F, ZuaZua E (2012) Continuous adjoint approach for the Spalart–Allmaras model in aerodynamic optimization. *AIAA J* 50(3):631–646
30. Byrd RH, Khalfan HF, Schnabel RB (1996) Analysis of a symmetric rank-one trust region method. *SIAM J Optim* 6(4):1025–1039
31. Campobasso M, Duta M, Giles M (2003) Adjoint calculation of sensitivities of turbomachinery objective functions. *J Propul Power* 19(4):693–703
32. Campobasso M, Giles M (2004) Stabilization of a linear flow solver for turbomachinery aeroelasticity by means of the recursive projection method. *AIAA J* 42(9):1765–1774
33. Campobasso M, Giles M (2006) Stabilizing linear harmonic flow solvers for turbomachinery aeroelasticity with complex iterative algorithms. *AIAA J* 44(5):1048–1059
34. Challis V, Guest J (2009) Level set topology optimization of fluids in Stokes flow. *Int J Numer Methods Eng* 79:1284–1308
35. Chen B, Kikuchi N (2001) Topology optimization with design-dependent loads. *Comput Methods Appl Mech Eng* 37:57–70
36. Chen S, Chen W, Lee S (2010) Level set based robust shape and topology optimization under random field uncertainties. *Struct Multidisc Optim* 41:507–524
37. Chorin A (1967) A numerical method for solving incompressible viscous flow problems. *J Comput Phys* 2:12–26
38. Courty F, Dervieux A, Koobus B, Hascoët L (2003) Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation. *Optim Methods Softw* 18(5):615–627
39. Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York
40. Dede E (2010) Multiphysics optimization, synthesis, and application of jet impingement target surfaces. In: Thermal and thermomechanical phenomena in electronic systems (ITherm), 2010 12th IEEE intersociety conference, pp 1–7
41. Deng Y, Liu Y, Wu J, Wu Y (2013) Topology optimization of steady Navier–Stokes flow with body force. *J Comput Methods Appl Mech Eng* 255:306–321
42. Deng Y, Zhang P, Liu Y, Wu Y, Liu Z (2013) Optimization of unsteady incompressible Navier–Stokes flows using variational level set method. *Int J Numer Methods Fluids* 71(12):1475–1493
43. Duan X, Ma Y, Zhang R (2008) Shape-topology optimization for Navier–Stokes problem using variational level set method. *J Comput Appl Math* 222:487–499
44. Duta M, Giles M, Campobasso M (2002) The harmonic adjoint approach to unsteady turbomachinery design. *Int J Numer Methods Fluids* 40:323–332
45. Dwight R, Brezillon J (2006) Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA J* 44(12):3022–3031
46. Dwight R, Brezillon J, Vollmer D (2006) Efficient algorithms for solution of the adjoint compressible Navier–Stokes equations with applications. In: Proceedings of the ONERA-DLR aerospace symposium (ODAS), Toulouse
47. El-Beltagy M, Nair P, Keane A (1999) Metamodelling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. In: Genetic and evolutionary computation conference—GECCO 1999. Morgan Kaufmann, San Francisco
48. Elliot J, Peraire J (1996) Aerodynamic design using unstructured meshes. In: AIAA paper 96-1941, 27th Dynamics conference, New Orleans, USA
49. Elliot J, Peraire J (1997) Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA J* 35(9):1479–1485
50. Evgrafov A (2005) The limits of porous materials in the topology optimization of Stokes flows. *Appl Math Optim* 52:263–277

51. Fletcher R, Reeves CM (1964) Function minimization by conjugate gradients. *Comput J* 7:149–154
52. Frink N (1996) Assessment of an unstructured-grid method for predicting 3-d turbulent viscous flows. In: AIAA paper 96-0292, 34th Aerospace meeting and exhibit. Reno, USA
53. Gersborg-Hansen A, Bendse M, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. *Struct Multidisc Optim* 31:251–259
54. Gersborg-Hansen A, Sigmund O, Haber R (2005) Topology optimization of channel flow problems. *Struct Multidisc Optim* 30:181–192
55. Giannakoglou K (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Prog Aerosp Sci* 38(1):43–76
56. Giannakoglou K (2013) Reformulation of the continuous adjoint method on the basis of discrete adjoint. A novel idea with applications to pseudo-1D aerodynamic optimization problems. NTUA Internal Report, PCOPT-13-2013
57. Giles M, Duta M, Muller J (2001) Adjoint code developments using the exact discrete approach. In: AIAA paper 2001-2596, 15th Computational fluid dynamics conference. California, USA
58. Giles M, Duta M, Muller J, Pierce N (2003) Algorithm developments for discrete adjoint methods. *AIAA J* 41(2):198–205
59. Giles M, Pierce N (2000) An introduction to the adjoint approach to design. *Flow Turbul Combust* 65:393–415
60. Gill P, Murray W, Wright M (1981) Practical optimization. Academic Press, London
61. Griewank A (1989) On automatic differentiation. Mathematical programming: recent developments and applications. Kluwer, Dordrecht
62. Griewank A, Juedes D, Mitev H, Utke J, Vogel O, Walther A (1996) ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. *ACM Trans Math Softw* 22(2):131–167
63. 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 (TOMS)* 26(1):19–45
64. Grinfield P (2010) Hadamard's formula inside and out. *J Optim Theory Appl* 146:654–690
65. Guest J (2009) Imposing maximum length scale in topology optimization. *Struct Multidisc Optim* 37:463–473
66. Guest J, Prévost J (2006) Topology optimization of creeping fluid flows using a Darcy–Stokes finite element. *Int J Numer Methods Eng* 66:461–484
67. Hascoët L, Utke J, Naumann U (2008) Cheaper adjoints by reversing address computations. *Sci Program* 16(1):81–92
68. He L, Wang D (2010) Concurrent blade aerodynamic-aero-elastic design optimization using adjoint method. *J Turbomach* 133(1):10
69. Herrera F, Lozano M, Moraga C (1999) Hierarchical distributed genetic algorithms. *Int J Intell Syst* 14(9):1099–1121
70. Hou G, Maroju V, Taylor A, Korivi V (1995) Transonic turbulent airfoil design optimization with automatic differentiation in incremental iterative form. *AIAA Pap* 95–1692
71. Huang H, Ekici K (2014) A discrete adjoint harmonic balance method for turbomachinery shape optimization. *Aerosp Sci Technol*. doi:[10.1016/j.ast.2014.05.015](https://doi.org/10.1016/j.ast.2014.05.015)
72. Jakobsson S, Amoignon O (2007) Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Comput Fluids* 36:1119–1136
73. Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3:233–260
74. Jameson A (1995) Optimum aerodynamic design using CFD and control theory. In: AIAA paper 95-36580, 12th Computational fluid dynamics conference. San Diego, USA
75. Jameson A, Alonso J, Reuther J, Martinelli L, Vassberg J (1998) Aerodynamic shape optimization techniques based on control theory. In: AIAA paper 98-2538, 29th Fluid dynamics conference. Albuquerque, USA
76. Jameson A, Kim S (2003) Reduction of the adjoint gradient formula in the continuous limit. In: AIAA paper 2003-0040, 41th Aerospace sciences meeting and exhibit. Reno, NV
77. Jameson A, Pierce N, Martinelli L (1997) Optimum aerodynamic design using the Navier–Stokes equations. In: AIAA paper 97-0101, 35th Aerospace sciences meeting and exhibit. Reno, NV
78. Jameson A, Reuther J (1994) Control theory based airfoil design using the Euler equations. In: AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization. Panama City Beach
79. Jameson A, Shankaran S, Martinelli L (2003) Continuous adjoint method for unstructured grids. In: AIAA paper 2003-3955, 16th CFD conference. Orlando, FL
80. Jameson A, Shankaran S, Martinelli L (2008) Continuous adjoint method for unstructured grids. *AIAA J* 46(5):1226–1239
81. Jones W, Launder B (1972) The prediction of laminarization with a two-equation model of turbulence. *Int J Heat Mass Transf* 15(2):301–314
82. Kavvadias I, Papoutsis-Kiachagias E, Dimitrakopoulos G, Giannakoglou K (2014) The continuous adjoint approach to the $k - \omega$ SST turbulence model with applications in shape optimization. *Eng Optim*. doi:[10.1080/0305215X.2014.979816](https://doi.org/10.1080/0305215X.2014.979816)
83. Keller H (2002) A remedy for instability. In: Estep D, Tavener S (eds) SIAM Proceedings in applied mathematics, vol 109, pp 185–196
84. Kim C, Kim C, Rho O (2001) Sensitivity analysis for the Navier–Stokes equations with two equations turbulence models. *AIAA J* 39(5):838–845
85. Kim C, Kim C, Rho O (2002) Effects of constant eddy viscosity assumption on gradient-based design optimization. *AIAA Paper* 2002-0262
86. Kim C, Kim C, Rho O (2003) Feasibility study of the constant eddy-viscosity assumption in gradient-based design optimization. *J Aircr* 40:1168–1176
87. Kim S, Alonso J, Jameson A (1999) A gradient accuracy study for the adjoint-based Navier–Stokes design method. In: AIAA paper 99-0299, 37th Aerospace sciences meeting and exhibit. Reno, NV
88. Kondoh T, Matsumori T, Kawamoto A (2012) Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration. *Struct Multidisc Optim* 45:693–701
89. Kontoleontos E, Asouti V, Giannakoglou K (2012) An asynchronous metamodel-assisted memetic algorithm for CFD-based shape optimization. *Eng Optim* 44(2):157–173
90. Kontoleontos E, Papoutsis-Kiachagias E, Zymaris A, Paradimitriou D, Giannakoglou K (2013) Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Eng Optim* 45(8):941–961
91. Kreissl S, Maute K (2012) Level set based fluid topology optimization using the extended finite element method. *Int J Numer Methods Eng* 46:311–326
92. Landau L, Lifshitz E (1987) Fluid mechanics, Volume 6 of course of theoretical physics. Pergamon Press, New York
93. Launder B, Sharma B (1974) Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Lett Heat Mass Transf* 1(2):131–137
94. Li Z, Navo I, Hussaini M, Dimet FL (2003) Optimal control of cylinder wakes via suction and blowing. *Comput Fluids* 32:149–171
95. Lim D, Ong Y, Jin Y, Sendhoff B, Lee B (2007) Efficient hierarchical parallel genetic algorithms using grid computing. *Future Gener Comput Syst* 23(4):658–670

96. Lions J (1971) Optimal control of systems governed by partial differential equations. Springer, New York
97. Lozano C (2012) Discrete surprises in the computation of sensitivities from boundary integrals in the continuous adjoint approach to inviscid aerodynamic shape optimization. *Comput Fluids* 56:118–127
98. Luenberger D (2003) Linear and nonlinear programming, 2nd edn. Kluwer, Dordrecht
99. Marta A, Shankaran S (2013) On the handling of turbulence equations in RANS adjoint solvers. *Comput Fluids* 74:102–113
100. Matsumori T, Kawamoto A, Kondoh T (2010) Topology optimization for fluid–thermal interaction problems. In: 6th China–Japan–Korea joint symposium on optimization of structural and mechanical systems, June 22–25, 2010, Kyoto, Japan, pp 1225–1233
101. Maute K, Allen M (2004) Conceptual design of aeroelastic structures by topology optimization. *Struct Multidiscip Optim* 27:27–42
102. Mavriplis D. (2007) Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA J* 45(4):741–750
103. Mavriplis D (2006) Multigrid solution of the discrete adjoint for optimization problems on unstructured meshes. *AIAA J* 44(1):42–50
104. Michalewicz Z (1994) Genetic Algorithms + Data Structures = Evolution Programs, 2nd edn. Springer, Berlin
105. Mohammadi B, Pironneau O (2001) Applied shape optimization for fluids. Oxford University Press, Oxford
106. Nadarajah S, Jameson A (2000) A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In: AIAA paper 2000-0667, 38th Aerospace sciences meeting and exhibit. Reno, USA
107. Nadarajah S, Jameson A (2001) Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *AIAA Pap* 25(30)
108. Nadarajah S, Jameson A (2002) Optimal control of unsteady flows using a time accurate method. In: AIAA paper 2002-5436, 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization. Atlanta, USA
109. Neme N, Zingg D (2001) Towards efficient aerodynamic shape optimization based on the Navier–Stokes equations. *AIAA Pap* 2001–2532
110. Newman J, Anderson W, Whitfield D (1998) Multidisciplinary sensitivity derivatives using complex variables. MSSU-COE-ERC-98-08
111. Nielsen E, Anderson W (1999) Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations. *AIAA J* 37(11):185–191
112. Nielsen E, Kleb W (2005) Efficient construction of discrete adjoint operators on unstructured grids by using complex variables. *AIAA Pap* 03(24)
113. Nielsen E, Lu J, Park M, Darmofal D (2003) An exact dual adjoint solution method for turbulent flows on unstructured grids. In: AIAA paper 2003-0272, 41st Aerospace sciences meeting and exhibit. Reno, USA
114. Nielsen E, Park M (2005) Using an adjoint approach to eliminate mesh sensitivities in aerodynamic design. *AIAA J* 44(5):948–953
115. Nocedal J, Wright S (1999) Numerical optimization. Springer, Berlin
116. Olesen L, Okkels F, Bruus H (2006) A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow. *Int J Numer Methods Eng* 65:975–1001
117. Ong Y, Lim M, Zhu N, Wong K (2006) Classification of adaptive memetic algorithms: a comparative study. *IEEE Trans Syst Man Cybern Part B Cybern* 36(1):141–152
118. Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *Int J Numer Methods Fluids* 58(8):861–877
119. Othmer C (2014) Adjoint methods for car aerodynamics. *J Math Indus* 4(6):1–23
120. Othmer C, Papoutsis-Kiachagias E, Haliskos K (2011) CFD optimization via sensitivity-based shape morphing. In: 4th ANSA & μ ETA international conference. Thessaloniki, Greece
121. Papadimitriou D, Giannakoglou K (2007) A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. *J Comput Fluids* 36(2):325–341
122. Papadimitriou D, Giannakoglou K (2007) Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems. *Int J Numer Methods Fluids* 56(10):1929–1943
123. Papadimitriou D, Giannakoglou K (2013) Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint. *Int J Numer Methods Fluids* 71(5):652–670
124. Papadimitriou D, Giannakoglou K (2009) The continuous Direct–Adjoint approach for second order sensitivities in viscous aerodynamic inverse design problems. *Comput Fluids* 38:1528–1538
125. Papoutsis-Kiachagias E, Kontoleontos E, Zymaris A, Papadimitriou D, Giannakoglou K (2011) Constrained topology optimization for laminar and turbulent flows, including heat transfer. In: Evolutionary and deterministic methods for design, optimization and control. Capua, Italy
126. Papoutsis-Kiachagias E, Kyriacou S, Giannakoglou K (2014) The continuous adjoint method for the design of hydraulic turbomachines. *Comput Methods Appl Mech Eng* 278:621–639
127. Papoutsis-Kiachagias E, Zymaris A, Kavvadias I, Papadimitriou D, Giannakoglou K (2014) The continuous adjoint approach to the $k - \epsilon$ turbulence model for shape optimization and optimal active control of turbulent flows. *Eng Optim.* doi:[10.1080/0305215X.2014.892595](https://doi.org/10.1080/0305215X.2014.892595)
128. Pascual V, Hascoët L (2008) TAPENADE for C. In: Selected papers from AD2008 Bonn, Lecture Notes in Computational Science and Engineering. Springer, Berlin
129. Patankar S, Spalding D (1972) Calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int J Heat Mass Transf* 15:1787–1806
130. Periaux J, Winter G (1995) Genetic algorithms in engineering and computer science. Wiley, New York
131. Peter J, Dwight R (2010) Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches. *Comput Fluids* 39(3):373–391
132. Peter J, Mayeur J (2006) Improving accuracy and robustness of a discrete direct differentiation method and discrete adjoint method for aerodynamic shape optimization. In: Proceedings of ECCOMAS. Egmond ann Zee
133. Piegl L, Tiller W (1997) The NURBS book. Springer, Berlin
134. Pironneau O (1984) Optimal shape design for elliptic systems. Springer, New York
135. Reuther J, Alonso J, Rimlinger M, Jameson A (1996) Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. In: AIAA, NASA and ISSMO, symposium on multidisciplinary analysis and optimization. 6th, Bellevue, WA
136. Reuther J, Jameson A (1994) Control theory based airfoil design for potential flow and a finite volume discretization. In: AIAA paper 91-499, 32th Aerospace sciences meeting and exhibit. Reno, NV
137. Reuther J, Jameson A (1995) Aerodynamic shape optimization of wing and wing-body configurations using control theory. In: AIAA paper 95-0213, 33rd Aerospace sciences meeting and exhibit. Reno, NV

138. Reuther J, Jameson A, Farmer J, Martinelli L, Saunders D (1996) Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. AIAA Paper 96-0094
139. Rodi W (1991) Experience with two-layer models combining the $k - \epsilon$ model with a one-equation model near the wall. In: AIAA paper 91-0216, 29th Aerospace sciences meeting. Reno, USA
140. Rodi W, Scheuerer G (1986) Scrutinizing the $k - \epsilon$ model under adverse pressure gradient conditions. *J Fluid Eng* 108:174–179
141. Rozvany G (2001) Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Struct Multidisc Optim* 21:90–108
142. Saad Y, Schultz M (1988) Gmres: a generalized minimum residual algorithm for solving non-symmetric linear systems. *SIAM J Sci Stat Comput* 7(3):856–859
143. Sagebaum M, Gauger N, Naumann U, Lotz J, Leppkes K (2013) Algorithmic differentiation of a complex C++ code with underlying libraries. *Procedia Comput Sci* 18:208–217
144. Schmidt S, Schulz V (2008) Pareto-curve continuation in multi-objective optimization. *Pac J Optim* 4(2):243–257
145. Schropp G, Keller H (1993) Stabilization of unstable procedures: the recursive projection method. *SIAM J Numer Anal* 30(4):1099–1120
146. Sethian J, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. *J Comput Phys* 163:489–528
147. Seung-Hyun H, Seonho C (2005) Topological shape optimization of heat conduction problems using level set approach. *Numer Heat Transf Part B Fundam* 48:67–88
148. Sigmund O, Clausen P (2007) Topology optimization using a mixed formulation: an alternative way to solve pressure load problems. *Comput Methods Appl Mech Eng* 196:1874–1889
149. Sondak D, Pletcher R (1995) Application of wall function to generalized nonorthogonal curvilinear coordinate systems. *AIAA J* 33(1):33–41
150. Spalart P (2000) Strategies for turbulence modelling and simulations. *Int J Heat Fluid Flow* 21(3):252–263
151. Spalart P, Allmaras S (1992) A one-equation turbulence model for aerodynamic flows. In: AIAA paper 92-0439, 30th Aerospace sciences meeting and exhibit. Reno, USA
152. Spalart P, Jou W, Strelets M, Allmaras S (1997) Comments on the feasibility of LES for wings and on the hybrid RANS/LES approach. In: Proceedings of the first AFOSR international conference on DNS/LES
153. Stanford B, Ifju P (2009) Aeroelastic topology optimization of membrane structures for micro air vehicles. *Struct Multidisc Optim* 38:301–316
154. Stück A, Rung T (2013) Adjoint complement to viscous finite-volume pressure-correction methods. *J Comput Phys* 248:402–419
155. Thévenin D, Janiga G (2008) Optimization and computational fluid dynamics. Springer, Berlin
156. Tucker P (2003) Differential equation-based wall distance computation for DES and RANS. *J Comput Phys* 190:229–248
157. Tucker P, Rumsey C, Bartels R, Biedron R (2003) Transport equation based wall distance computations aimed at flows with time-dependent geometry. NASA Report 212680
158. Tucker P, Rumsey C, Spalart P, Bartels R, Biedron R (2005) Computations of wall distances based on differential equations. *AIAA J* 43(3):539–549
159. Turgeon E, Pelletier D, Borggaard J, Etienne S (2007) Application of a sensitivity equation method to the $k - \epsilon$ model of turbulence. *Optim Eng* 8:341–372
160. Utke J, Naumann U. OpenAD/F: User Manual. Technical Report, Argonne National Laboratory. <http://www.mcs.anl.gov/OpenAD/openad>
161. Vezyris C, Kavvadias I, Papoutsis-Kiachagias E, Giannakoglou K (2014) Unsteady continuous adjoint method using POD for jet-based flow control. In: 11th World congress on computational mechanics, ECCOMAS. Barcelona, Spain
162. Wang M, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192:227–246
163. Wang Q, Moin P, Iaccarino G (2009) Minimal repetition dynamic check-pointing algorithm for unsteady adjoint calculation. *SIAM J Sci Comput* 31(4):2549–2567
164. White F (1974) Viscous fluid flow. McGraw-Hill Inc, New York
165. Wiker N, Klarbring A, Borrrell T (2007) Topology optimization of regions of Darcy and Stokes flow. *Int J Numer Methods Eng* 69:1371–1404
166. Xia H, Tucker P (2010) Finite volume distance field and its application to medial axis transforms. *Int J Numer Methods Eng* 82:114–130
167. Xu J, Yan C, Fan J (2011) Computation of wall distances by solving a transport equation. *Appl Math Mech* 32(2):141–150
168. Yaji K, Yamada T, Yoshino M, Matsumoto T, Izui K, Nishiwaki S (2014) Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions. *J Comput Phys* 274:158–181
169. Yoon G (2010) Topological design of heat dissipating structure with forced convective heat transfer. *J Mech Sci Technol* 24(6):1225–1233
170. Yoon G, Jensen J, Sigmund O (2007) Topology optimization of acoustic-structure interaction problems using a mixed finite element formulation. *Int J Numer Methods Eng* 70:1049–1075
171. Zhou S, Li Q (2008) A variational level set method for the topology optimization of steady-state Navier–Stokes flow. *J Comput Phys* 227:10178–10195
172. Zhuang C, Xiong Z, Ding H (2007) A level set method for topology optimization of heat conduction problem under multiple load cases. *Comput Methods Appl Mech Eng* 196:1074–1084
173. Zymaris A, Papadimitriou D, Giannakoglou K, Othmer C (2009) Continuous adjoint approach to the Spalart–Allmaras turbulence model for incompressible flows. *Comput Fluids* 38(8):1528–1538
174. Zymaris A, Papadimitriou D, Giannakoglou K, Othmer C (2010) Adjoint wall functions: a new concept for use in aerodynamic shape optimization. *J Comput Phys* 229(13):5228–5245