Arch Comput Methods Eng (2008) 15: 447–488
DOI 10.1007/s11831-008-9025-y

ORIGINAL PAPER

Aerodynamic Shape Optimization Using First and Second Order Adjoint and Direct Approaches

Dimitrios I. Papadimitriou · Kyriakos C. Giannakoglou

Received: 24 December 2007 / Accepted: 24 December 2007 / Published online: 14 August 2008

Abstract This paper focuses on discrete and continuous adjoint approaches and direct differentiation methods that can efficiently be used in aerodynamic shape optimization problems. The advantage of the adjoint approach is the computation of the gradient of the objective function at cost which does not depend upon the number of design variables. An extra advantage of the formulation presented below, for the computation of either first or second order sensitivities, is that the resulting sensitivity expressions are free of field integrals even if the objective function is a field integral. This is demonstrated using three possible objective functions for use in internal aerodynamic problems; the first objective is for inverse design problems where a target pressure distribution along the solid walls must be reproduced; the other two quantify viscous losses in duct or cascade flows, cast as either the reduction in total pressure between the inlet and outlet or the field integral of entropy generation. From the mathematical point of view, the three functions are defined over different parts of the domain or its boundaries, and this strongly affects the adjoint formulation. In the second part of this paper, the same discrete and continuous adjoint formulations are combined with direct differentiation methods to compute the Hessian matrix of the objective function. Although the direct differentiation for the computation of the gradient is time consuming, it may support the adjoint method to calculate the exact Hessian matrix components with the minimum CPU cost. Since, however, the CPU cost is proportional to the number of design variables, a well performing optimization scheme, based on the exactly computed Hessian during the starting cycle and a quasi Newton (BFGS) scheme during the next cycles, is proposed.

D.I. Papadimitriou · K.C. Giannakoglou (✉)
National Technical University of Athens (NTUA),
P.O. Box 64069, 15710 Athens, Greece
e-mail: kgianna@central.ntua.gr

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

1 Introduction

Historically, the first study on the inverse design of an aerodynamic profile that produces a desired pressure distribution was presented by Lighthill, [1]. His method was based on the conformal mapping of the airfoil to a unit circle and was restricted to incompressible flows, although an extension to compressible flows was later presented by McFadden, [2]. Nowadays, the advent of powerful computers and the availability of flow analysis software with reasonable CPU cost, allows aerodynamic shape optimization problems for complex configurations (blades, wings or, even, an entire aircraft) at various (from incompressible to supersonic) flow conditions to run routinely.

The design of optimally shaped aerodynamic configurations based on an available analysis software (flow solver) and predefined targets (objective functions) can be solved using stochastic [3–6] or deterministic [7–11] search algorithms, without excluding any hybridization of them. Stochastic optimization methods, either population- or individual-based ones, are well known for their ability to capture the global optimal solution without being trapped into local optima. The price to pay is the higher CPU cost which can be reduced only through the smart use of surrogate analysis tools [12–15]. Deterministic optimization methods, cast in the form of gradient- and, occasionally Hessian-driven algorithms, are efficient and effective tools provided that a non-misleading starting solution is available and the fitness landscape is not multimodal close to the sought optimal solution. To employ any descent or ascent method, procedures to compute first and second order sensitivities of the objective function with respect to the design variables are needed. It should be stressed that this paper is dealing exclusively with exact gradient and Hessian computations, unless stated otherwise.

The objective function gradient required can be computed using several algorithms, most of which are quite costly, since their computational cost is proportional to the number of design variables. Finite differences, direct differentiation and the complex variable approach are among them; they are all briefly reported in subsequent sections, not only in the sake of completeness but since these constitute auxiliary supporting elements of the proposed method. Compared to the aforementioned techniques, the adjoint method (which is where this paper is mostly focusing on) computes the objective function gradient at the cost of a single flow analysis, irrespective of the number of design variables. The adjoint approach can be devised either in discrete or continuous form. In general, the discrete method in which the adjoint equations are derived from the discretized flow equations, is not that sensitive to the type of objective function used. In contrast, the continuous method is objective function sensitive and customized developments should be made whenever the objective function changes (being either a field integral, a boundary integral along the solid walls or a boundary integral at the inlet and outlet, etc.). Throughout this paper, discrete and continuous adjoint variants are developed and exposed in parallel.

Lions [16] was the first to handle an inverse design problem as a control problem, where the parameterized shape of an aerodynamic body controlled the value of a function measuring the shape efficiency. The adjoint approach to aerodynamic design was proposed by Pironneau, [17, 18], restricted to potential flows only. The theory was then extended by Jameson to transonic flows governed by inviscid, [19–21], and viscous flow equations, [22]. A detailed overview of existing adjoint methods, with possible links to the formulation presented in this paper is provided elsewhere in this paper (on a section-by-section basis).

This paper is structured in a way that allows the author to follow the adaptation of the proposed (discrete and continuous) adjoint approaches to three different problems: (a) inverse design of aerodynamic shapes where the target is a known pressure distribution, (b) optimization of a duct or cascade for minimum total pressure losses through it and (c) optimization of the same configuration targeting at minimum entropy generation within the flow field. The last two objective functions aim at minimizing viscous losses by handling different flow quantities and are shown to perform similarly. With the continuous adjoint approach, since the three functionals are defined over a different part of the domain (along the solid wall boundaries, at the inlet–outlet boundaries and over the flow domain, respectively), different formulations (adjoint equations, boundary conditions, sensitivity derivatives) arise; these are all discussed below, in detail. Irrespective of the objective function, a common feature of the continuous adjoint variants presented is that the expression for the sensitivities does not include field integrals; this is numerically convenient, as discussed below.

The second part in this paper is concerned with the computation of second order sensitivities. It is known that the relevant literature is extremely poor, although the knowledge of the (exact) Hessian matrix allows the use of the very efficient Newton method; in the majority of papers dealing with the Hessian in the area of aerodynamic shape optimization, this is approximated through various methods in combination with a quasi Newton method. Four approaches are presented, both in their discrete and continuous form, derived from all possible combinations of the direct differentiation and the adjoint approach.

Without loss in generality, the demonstrated applications are all in the field of turbomachines (design of compressor and turbine cascade airfoils).

2 Computation of the Gradient of an Objective Function

In this section, an overview of the most widely used methods to compute the gradient of an objective function F (a scalar one in single-objective optimization or a vector one with L components in multi-objective optimization) with respect to the design variables b_i, $(i = 1, \dots, N)$ is presented. In aerodynamic shape optimization problems, F or $F_1, \dots, F_L$ depend on the flow (state) variables U_k that satisfy the flow (state) equations $R_m = 0, (k, m = 1, \dots, M)$. Here, M is the product of the number of grid nodes and that of flow variables per grid node, i.e. the total number of state variables. Since the present paper is concerned with the adjoint approach (primarily) and the direct differentiation method (secondarily), both will be presented briefly at the end of this section; in subsequent sections, however, thorough analyses of their (discrete and continuous) variants can be found.

2.1 The Finite Difference Method

The simplest way to compute function derivatives is through *finite differences*. $\frac{\delta F}{\delta b_i}$ can be approximated by the central finite difference scheme

$$\frac{\delta F}{\delta b_i} \cong \frac{F(b_1, \dots, b_i + \epsilon, \dots, b_N) - F(b_1, \dots, b_i - \epsilon, \dots, b_N)}{2\epsilon}, \tag{1}$$

where the outcome must become insensitive to the selected value of ϵ. The CPU cost for the computation of sensitivities with respect to the N design variables is $2N$ function calls, i.e. $2N$ calls to the (flow) analysis software. The method is fully amenable to parallelization since function computations can be carried out concurrently. The cost can be halved by using the one-sided, first order finite difference scheme, namely

$$\frac{dF}{db_i} \cong \frac{F(b_1,\ldots,b_i+\epsilon,\ldots,b_N)-F(b_1,\ldots,b_i,\ldots,b_N)}{\epsilon}. \tag{2}$$

Accuracies in the computed function and gradient values are strongly linked. This might become a huge problem in some aerodynamic design applications where turbulent flows over extremely stretched computational grids are modeled; if adequate convergence of the flow equations is not achievable, the accuracy in the gradient, computed through (1) and (2), becomes questionable.

Note that, throughout this paper, the (total) derivative of F with respect to b_i will be denoted by $\frac{\delta F}{\delta b_i}$ (instead of $\frac{dF}{db_i}$) whereas $\frac{\partial F}{\partial b_i}$ stands for the direct or partial sensitivities of F with respect to b_i.

2.2 The Complex Variable Method

An alternative, well performing method to compute the gradient of an objective function is the so-called *complex variable method*, [23–28].

Although the objective function F is a function with exclusively real arguments, we may consider it to accept complex arguments. So, in case of a single argument x, this can be expanded in a Taylor series as follows

$$F(x+i\epsilon)=F(x)+i\epsilon\frac{\partial F}{\partial x}+\mathcal{O}(\epsilon^2). \tag{3}$$

Equation (3) can be solved for the imaginary part to get

$$\frac{\partial F}{\partial x}=\lim_{\epsilon\to 0}\frac{\text{imag}(F(x+i\epsilon))}{\epsilon}, \tag{4}$$

where imag( ) denotes the imaginary part of a complex quantity. The real part of F is the function value, computed with truncation error of $\mathcal{O}(\epsilon)$. The last expression can be generalized to cover optimization problems with N design variables, so

$$\frac{\partial F}{\partial b_i}\cong\frac{\text{imag}(F(b_i+i\epsilon))}{\epsilon}. \tag{5}$$

Note that a single call to function F (i.e. a single evaluation) per partial derivative is required. Although the cost of running a code with complex variables is higher, the use of (5) outperforms that of (1). Compared to the finite difference method, the use of complex variables leads to derivatives that are less sensitive to the value of ϵ. The complex variable method can be used in multi-objective optimization problems as well.

2.3 Automatic Differentiation

Starting from an existing source code which numerically solves the state pde's to compute the scalar or vector objective function value(s), an *automatic differentiation* software, [29, 30], generates a new code which additionally computes the $\frac{\delta F}{\delta b_i}$ values, according to a set of linguistic and mathematical rules. On condition that the analysis software is available as source code, the development cost of the optimization method through automatic differentiation becomes negligible, despite the high memory requirements, [31].

Two approaches to automatic differentiation i.e. the so-called *forward* and *reverse* modes have been developed. Research is now focusing on hybrid modes which combine the best features of both. Both employ the chain rule to accumulate contributions to derivatives in a different manner. For N independent variables $(b_1,\ldots,b_N)$ and a scalar function F $(L=1)$, the F value is considered to be the outcome of $m-N$ successive operations, each of which corresponds to a single program instruction. These instructions generate $m-N$ intermediate (dependent) variables $b_{N+1},\ldots,b_m$.

The *forward mode*, [30], propagates derivatives of intermediate variables with respect to the independent variables. Let us consider the sequence of auxiliary operations (corresponding to virtual code instructions)

$$\begin{aligned}
b_{N+1} &= f_{N+1}(b_1,\ldots,b_N),\\
b_{N+2} &= f_{N+2}(b_1,\ldots,b_{N+1}),\\
&\vdots\\
b_m &= f_m(b_1,\ldots,b_{m-1}),\\
F &= b_m
\end{aligned}$$

which generate the desired response F. The forward mode holds a storage location for all intermediate gradients (denoted by ∇b_i) which are computed by means of the chain rule, as follows

$$\begin{aligned}
b_{N+1} &= f_{N+1}(b_1,\ldots,b_N),\\
\nabla b_{N+1} &= \sum_{i=1}^{N}\frac{\delta f_{N+1}}{\delta b_i}e_i,\\
b_{N+2} &= f_{N+2}(b_1,\ldots,b_{N+1}),\\
\nabla b_{N+2} &= \sum_{i=1}^{N}\frac{\delta f_{N+2}}{\delta b_i}e_i+\frac{\delta f_{N+2}}{\delta b_{N+1}}\nabla b_{N+1},\\
&\vdots\\
b_m &= f_m(b_1,\ldots,b_{m-1}),\\
\nabla b_m &= \sum_{i=1}^{N}\frac{\delta f_m}{\delta b_i}e_i+\sum_{i=N+1}^{m-1}\frac{\delta f_m}{\delta b_i}\nabla b_i,
\end{aligned}$$

$$F \equiv b_m,$$
$$\nabla F \equiv \nabla b_m,$$

where e_i is the i-th column of the $N \times N$ unit matrix. In the derivative code, with instructions in accordance with the previous sequence of operations, the desired gradient is the outcome of the last instruction (∇F), obtained in the expense of considerable memory requirements.

In the *reverse mode*, a scalar derivative (denoted by $\bar{b}_i$) is associated with each intermediate variable ($i = N+1, \ldots, m$); the new quantities are defined by

$$\bar{b}_i = \frac{\delta b_m}{\delta b_i}.$$

By definition, $\bar{b}_m \equiv 1$. This definition is also extended to the independent variables of the problem ($i = 1, \ldots, N$). According to the chain rule, all but the m-th $\bar{b}_i$'s, associated with the dependent variables, can be computed using

$$\bar{b}_i = \sum_{j=i+1}^{m} \frac{\delta b_m}{\delta b_j}\frac{\delta f_j}{\delta b_i} = \sum_{j=i+1}^{m} \bar{b}_j \frac{\delta f_j}{\delta b_i}, \quad i = N+1, \ldots, m-1 \tag{6}$$

whereas, for the independent variables, we may also write

$$\bar{b}_i = \sum_{j=N+1}^{m} \frac{\delta b_m}{\delta b_j}\frac{\delta f_j}{\delta b_i} = \sum_{j=N+1}^{m} \bar{b}_j \frac{\delta f_j}{\delta b_i}, \quad i = 1, \ldots, N. \tag{7}$$

From the algorithmic viewpoint, the reverse mode consists of a forward and a backward sweep. During the forward sweep, all dependent variables are computed and stored and their derivatives initialized, i.e.

$$b_i = f_i(b_1, \ldots, b_{i-1}), \quad i = N+1, \ldots, m, \tag{8}$$
$$\bar{b}_i = 0, \quad i = 1, \ldots, m-1, \tag{9}$$
$$\bar{b}_m = 1. \tag{10}$$

The reverse loop accumulates contributions to $\bar{b}_i$, $i = m, \ldots, N+1$ and consists of two nested loops, which can be written in pseudo-code as follows

```
do $j = m, N+1, -1$
  do $i = 1, j-1$
    $\bar{b}_i = \bar{b}_i + \bar{b}_j \frac{\delta f_j}{\delta b_i}$
  enddo
enddo
```

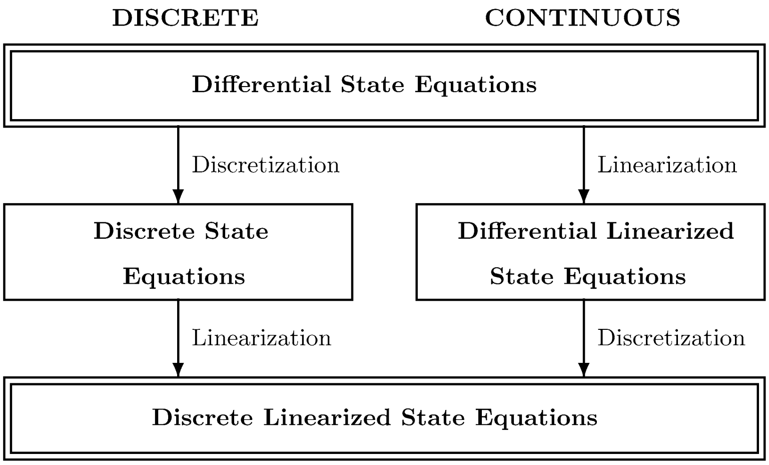
There are many automatic differentiation tools available, [32]. Among them, for either *Fortran 77/90* or *ANSI C/C++* codes, we briefly mention *ADIFOR* (Automatic Differentiation of Fortran, [33]), *TAMC* (Tangent linear and Adjoint Model Compiler) or its successor *TAF* (Transformation of Algorithms in Fortran, [34]), *DAFOR* (*Differential Algebraic Extension of Fortran*, [35]), *GRESS* (Gradient–Enhanced Software System, [36]), *Odyssée*, [37], *AD01*, [38], *ADOL-F* (Automatic Differentiation of FORTRAN Codes, [39]), *IMAS* (Integrated Modeling and Analysis System, [40]), *OPTIMA90*, [41], *ADIC* (Automatic Differentiation of C Programs, [42]), *ADOL-C* (Automatic Differentiation of Algorithms written in C/C++, [43]).

2.4 Direct Differentiation

Direct differentiation is based on the computation of the gradient of the state variables with respect to the design variables ($\frac{\delta U_k}{\delta b_i}$) and the application of the chain rule

$$\frac{\delta F_l}{\delta b_i} = \frac{\partial F_l}{\partial b_i} + \frac{\partial F_l}{\partial U_k}\frac{\delta U_k}{\delta b_i} \tag{11}$$

to obtain the required gradient components of the objective function(s). Direct differentiation can be applied in discrete or continuous mode, as shown schematically below:


DISCRETE
CONTINUOUS
Differential State Equations
Discretization
Linearization
Discrete State Equations
Differential Linearized State Equations
Linearization
Discretization
Discrete Linearized State Equations

According to the discrete direct differentiation method, the state equations are first cast in discrete form, $R_m = 0$ and, then, differentiated to yield

$$\frac{\delta R_m}{\delta b_i} = \frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{\delta U_k}{\delta b_i} = 0. \tag{12}$$

The system of $M \times N$ equations, (12), must be solved for $\frac{\delta U_k}{\delta b_i}$ and these values are introduced into (11), to compute $\frac{\delta F_l}{\delta b_i}$.

In the continuous direct differentiation method, differential equations and boundary conditions for $\frac{\delta U_k}{\delta b_i}$ are derived starting from the state equations and their boundary conditions. Let us assume that

$$\begin{aligned} L\left(\frac{\delta U}{\delta b_i}\right) &= \phi, \quad \text{over } \Omega, \\ B_2\left(\frac{\delta U}{\delta b_i}\right) &= \epsilon, \quad \text{along } S, \end{aligned} \tag{13}$$

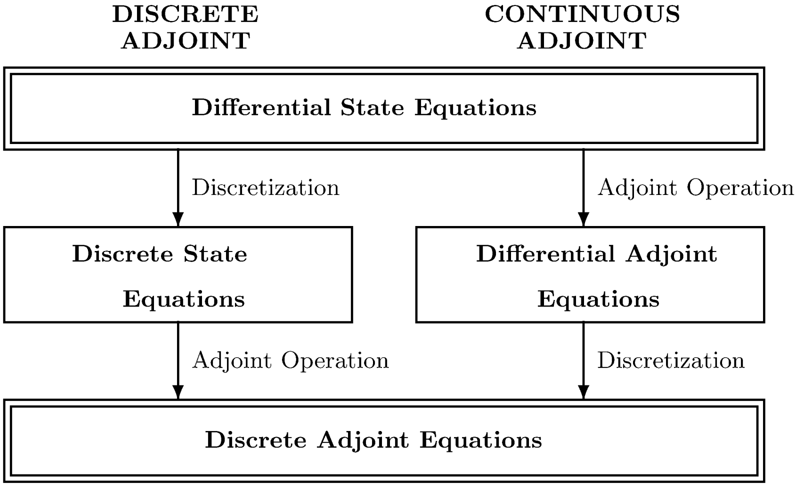
where S is the boundary of the flow domain Ω. Equations (13) must be discretized and solved (most likely with the same numerical scheme used for the state equations) for the nodal values of $\frac{\delta U_k}{\delta b_i}$. Let us also assume that the objective functions F_l involve field and boundary integrals as well as a term F_l^G that depends exclusively on the sensitivities of geometrical quantities. Then, $\frac{\delta F_l}{\delta b_i}$ can be symbolically expressed as

$$\frac{\delta F_l}{\delta b_i} = \int_\Omega \gamma \frac{\delta U}{\delta b_i} d\Omega + \int_S \zeta B_1\left(\frac{\delta U}{\delta b_i}\right) dS + \frac{\delta F_l^G}{\delta b_i}, \tag{14}$$

where γ and ζ are functions of geometrical quantities and state variables and B_1 is a differential operator. Once $\frac{\delta U_k}{\delta b_i}$ become available, $\frac{\delta F_l}{\delta b_i}$ can be deduced by numerically integrating (14). Practically, the discrete and continuous direct differentiation methods have the same computational cost.

2.5 The Adjoint Approach

The *adjoint approach* for the computation of the gradient of F_l also appears in either discrete or continuous mode; both are schematically shown below:



The discrete adjoint approach is based on properly defined augmented functionals $F_{aug,l}$. Their total derivatives with respect to the design variables are expressed by introducing the adjoint variables (to be also referred to as costate variables or Lagrange multipliers) Ψ_{ml} as follows

$$\frac{\delta F_{aug,l}}{\delta b_i} = \frac{\delta F_l}{\delta b_i} + \Psi_{ml}\frac{\delta R_m}{\delta b_i}. \tag{15}$$

Since $\frac{\delta R_m}{\delta b_i} = 0$, the derivatives of $F_{aug,l}$ and those of F_l are identical. Using (11) and (12), (15) yields

$$\begin{aligned} \frac{\delta F_{aug,l}}{\delta b_i} &= \frac{\partial F_l}{\partial b_i} + \frac{\partial F_l}{\partial U_k}\frac{\delta U_k}{\delta b_i} \\ &\quad + \Psi_{ml}\left(\frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{\delta U_k}{\delta b_i}\right) \end{aligned} \tag{16}$$

or

$$\begin{aligned} \frac{\delta F_{aug,l}}{\delta b_i} &= \frac{\partial F_l}{\partial b_i} + \Psi_{ml}\frac{\partial R_m}{\partial b_i} \\ &\quad + \left(\frac{\partial F_l}{\partial U_k} + \Psi_{ml}\frac{\partial R_m}{\partial U_k}\right)\frac{\delta U_k}{\delta b_i}. \end{aligned} \tag{17}$$

By satisfying the adjoint equations

$$\frac{\partial F_l}{\partial U_k} + \Psi_{ml}\frac{\partial R_m}{\partial U_k} = 0 \tag{18}$$

the derivatives of the augmented functionals are given by

$$\frac{\delta F_{aug,l}}{\delta b_i} = \frac{\partial F_l}{\partial b_i} + \Psi_{ml}\frac{\partial R_m}{\partial b_i}. \tag{19}$$

Compared to the discrete direct differentiation method, the discrete adjoint approach requires the solution of (18), i.e. $M \times L$ equations and, once the adjoint variables are available, $\frac{\delta F_l}{\delta b_i}$ is computed through (19). Consequently, in aerodynamic shape optimization problems, where the number of design variables N is expected to be much greater than the number of functionals, $N \gg L$, the gain from using the adjoint approach is clear.

In the continuous adjoint approach, according to (14), the gradient of the augmented functionals $F_{aug,l}$ is formulated as

$$\begin{aligned}\frac{\delta F_{aug,l}}{\delta b_i} &= \int_\Omega \gamma \frac{\delta U}{\delta b_i} d\Omega + \int_S \zeta B_1\left(\frac{\delta U}{\delta b_i}\right) dS + \frac{\delta F_l^G}{\delta b_i}\\ &\quad - \int_\Omega \Psi\left[L\left(\frac{\delta U}{\delta b_i}\right) - \phi\right] d\Omega\\ &\quad - \int_S (B_1^*\Psi)\left[B_2\left(\frac{\delta U}{\delta b_i}\right) - \epsilon\right] dS,\end{aligned} \tag{20}$$

where B_1^* is a linear operator. Through the Gauss divergence theorem and integration by parts, the second field integral is transformed as

$$\begin{aligned}\int_\Omega \Psi L\left(\frac{\delta U}{\delta b_i}\right) d\Omega &= \int_\Omega (L^*\Psi)\frac{\delta U}{\delta b_i} d\Omega\\ &\quad + \int_S (C_1^*\Psi) C_2\left(\frac{\delta U}{\delta b_i}\right) dS,\end{aligned} \tag{21}$$

where L^* is the adjoint operator to L, [44], and C_1^*, C_2 are operators derived from integration by parts. If B_1^* and B_2^* satisfy

$$\begin{aligned}\int_S (C_1^*\Psi) C_2\left(\frac{\delta U}{\delta b_i}\right) dS &= \int_S (B_2^*\Psi) B_1\left(\frac{\delta U}{\delta b_i}\right) dS\\ &\quad - \int_S (B_1^*\Psi) B_2\left(\frac{\delta U}{\delta b_i}\right) dS\end{aligned} \tag{22}$$

(20) is finally expressed as

$$\begin{aligned}\frac{\delta F_{aug,l}}{\delta b_i} &= \int_\Omega \left(\gamma - L^*\Psi\right)\frac{\delta U}{\delta b_i} d\Omega\\ &\quad + \int_S \left(\zeta - B_2^*\Psi\right) B_1\left(\frac{\delta U}{\delta b_i}\right) dS\\ &\quad + \int_\Omega \Psi\phi d\Omega + \int_S (B_1^*\Psi)\epsilon dS + \frac{\delta F_l^G}{\delta b_i}.\end{aligned} \tag{23}$$

By selecting the adjoint variable to satisfy the field and boundary adjoint equations

$$\begin{aligned}L^*\Psi &= \gamma, \quad \text{other } \Omega,\\ B_2^*\Psi &= \zeta, \quad \text{along } S\end{aligned} \tag{24}$$

the gradient of the objective functions is given by

$$\frac{\delta F_l}{\delta b_i} = \int_\Omega \Psi\phi d\Omega + \int_S (B_1^*\Psi)\epsilon dS + \frac{\delta F_l^G}{\delta b_i}. \tag{25}$$

The conclusion drawn is the same with that of the discrete adjoint approach: if $N \gg L$, the continuous adjoint approach considerably outperforms the direct differentiation method. A comparison between the discrete and continuous approach can be found in [45, 46].

3 First Order Sensitivities in Aerodynamic Shape Optimization—Direct Differentiation and Adjoint Approaches

3.1 State Equations

Depending on the application, the governing (state) equations to be satisfied in aerodynamic shape optimization problems are either the Euler or the Navier–Stokes ones. For turbulent flows, a turbulence model should be used as well. In this section, the Favre averaged Navier–Stokes equations along with the low-Reynolds Spalart–Allmaras turbulence model are used, [47]. However, variations in turbulent viscosity are omitted; although this may reduce the accuracy of the computed objective function sensitivities, the convergence of the optimization algorithm towards the optimum is not deteriorated significantly.

The Navier–Stokes equations for a compressible viscous fluid are cast in tensor form as

$$R_n = \frac{\partial U_n}{\partial t} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} = 0, \tag{26}$$

where for inviscid flows the last term is omitted. For 3D steady flows, where t is the pseudo-time, the conservative variables U_n and the inviscid f_{nk}^{inv} and viscous f_{nk}^{vis} fluxes, are given by

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{bmatrix}, \quad \begin{bmatrix} f_{1k}^{inv} \\ f_{2k}^{inv} \\ f_{3k}^{inv} \\ f_{4k}^{inv} \\ f_{5k}^{inv} \end{bmatrix} = \begin{bmatrix} \rho u_k \\ \rho u_1 u_k + p\delta_{k1} \\ \rho u_2 u_k + p\delta_{k2} \\ \rho u_3 u_k + p\delta_{k3} \\ u_k(E+p) \end{bmatrix}, \tag{27}$$

$$\begin{bmatrix} f_{1k}^{vis} \\ f_{2k}^{vis} \\ f_{3k}^{vis} \\ f_{4k}^{vis} \\ f_{5k}^{vis} \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ u_m \tau_{km} + q_k \end{bmatrix}.$$

According to the standard notation, u_k, $E = \rho e + \frac{1}{2}\rho u_k^2$, $\tau_{km} = \mu(\frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k}) + \lambda \delta_{km} \frac{\partial u_l}{\partial x_l}$ ($\lambda = -\frac{2}{3}\mu$, where μ is the sum of molecular and turbulent viscosity) and $q_k = k\frac{\partial T}{\partial x_k}$ stand for the velocity components, total energy per unit volume, viscous stresses and heat fluxes, respectively. Also, δ_{km} is the Kronecker symbol.

In the present method and software, a vertex-centered finite volume method is used for the discretization of (26). Inviscid fluxes crossing the boundaries of the control volume formed around grid node P are computed using the Roe's upwind scheme, [48]. So, the numerical inviscid flux associated with the edge formed by nodes P and Q is given by

$$\begin{aligned} h_{n,PQ} = & \frac{1}{2}\left(A_{nk,P} U_{k,L} + A_{nk,Q} U_{k,R}\right) \\ & - \frac{1}{2}|A_{nk,PQ}|\left(U_{k,L} - U_{k,R}\right), \end{aligned} \tag{28}$$

where

$$\begin{aligned} A_{nkl} &= \frac{\partial f_{nl}^{inv}}{\partial U_k}, \\ A_{nk} &= A_{nkl} n_l \end{aligned} \tag{29}$$

n_l being the unit vector which is normal to each segment defining the finite volume boundary. Second order accuracy is obtained using appropriate Taylor expansions for $U_{k,L}$ and $U_{k,R}$, based on $U_{k,P}$, $U_{k,Q}$ and the local gradients of U_k, Fig. 1.

Equations (26) are then discretized, at vertex P as

$$\sum_{Q=1}^{Q(P)} h_{n,PQ} = 0, \tag{30}$$

where $Q(P)$ is the number of the neighbouring nodes to P.

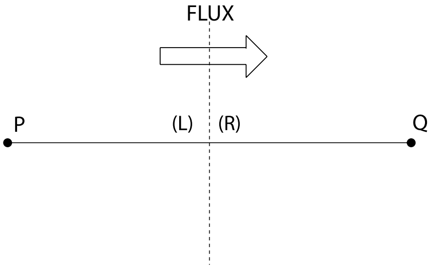


Fig. 1 Grid edge PQ and the associated finite volume boundary (i.e. the interface of the finite volumes defined around grid nodes P and Q, *dotted line*). A 1D Riemann problem is solved between the left (L) and right (R) states, which are defined using the P, Q nodal values and gradients of state variables

The computation of viscous fluxes is straightforward, by considering a linear distribution of U_j over any grid cell, an assumption that leads to constant gradient within each grid cell (unstructured grids) or central differences (structured grids).

3.2 Inverse Design Using the Euler Equations

In most inverse design problems, the objective function to be minimized is defined as the overall deviation of the pressure or pressure coefficient distribution of any candidate solution from a desired (target) pressure p or pressure coefficient c_p distribution. The shape that reproduces the target distribution, is sought. The first adjoint formulation for transonic flows developed by Jameson, [19], was, in fact, dealing with inverse design problems.

Provided that the target distribution is based on pressure, the objective function may be defined as

$$F = \frac{1}{2}\int_{S_w} (p - p_{tar})^2 dS \tag{31}$$

or

$$F = \frac{1}{2}\int_{S_w} (c_p - c_{p,tar})^2 dS. \tag{32}$$

This form of F may be used in many aerodynamic shape optimization problems such as the optimization of airfoils, wings, blades, ducts, etc. [22, 49–51]. The target pressure distribution should be known in advance but, in this paper, we will refrain from discussing possible ways for defining the target pressure distributions.

3.2.1 Discrete Direct Differentiation and Adjoint Approaches

Either the direct differentiation or the adjoint approach, both used in their discrete mode, require the computation of four quantities: $\frac{\partial R_m}{\partial b_i}$, $\frac{\partial R_m}{\partial U_k}$, $\frac{\partial F}{\partial b_i}$ and $\frac{\partial F}{\partial U_k}$, prior to calculating $\frac{\delta F}{\delta b_i}$, according to the theory presented in (2.5). The way these quantities can be computed is exposed below.

Among the various methods to compute the nodal values of $\frac{\partial R_m}{\partial b_i}$, the use of finite differences (based on (28), where the design variables are perturbed one by one), appears to be a logical choice. Other possible approaches are the hand differentiation, the automatic differentiation or the complex variable method. The nodal values of $\frac{\partial R_m}{\partial U_k}$ can be computed using (30), by analytically differentiating it with respect to the flow variables.

Concerning the direct sensitivities of F with respect to the design and flow variables, we get (based on (31))

$$\frac{\partial F}{\partial b_i} = \frac{1}{2}\sum_{S_w}(p-p_{tar})^2\frac{\delta(\Delta S)}{\delta b_i} \tag{33}$$

and

$$\frac{\partial F}{\partial U_k} = \sum_{S_w}(p-p_{tar})\frac{\delta p}{\delta U_k}\frac{\delta U_k}{\delta b_i}\Delta S, \tag{34}$$

where ΔS denotes the finite arc length (2D) or surface element (3D) over the solid walls controlled by the design variables. All the aforementioned sensitivities ($\frac{\partial R_m}{\partial b_i}$, $\frac{\partial R_m}{\partial U_k}$, $\frac{\partial F}{\partial b_i}$ and $\frac{\partial F}{\partial U_k}$) can be computed in a straightforward manner, at negligible CPU cost.

In this paper, we will refrain from discussing sensitivities of geometrical quantities with respect to the design variables, such as $\frac{\delta(\Delta S)}{\delta b_i}$, etc. These sensitivities depend on the selected parameterization; more on this topic can be found in [50–52].

3.2.2 Continuous Direct Differentiation

The first order partial sensitivity of the Euler equations, where the pseudo-time derivative has been omitted, (the extension to the Navier–Stokes equations is straightforward) is zero, so

$$\frac{\partial}{\partial b_i}\left(\frac{\partial f^{inv}_{nk}}{\partial x_k}\right) = 0. \tag{35}$$

The two partial derivatives are interchangeable (just for this reason, the partial $\frac{\partial}{\partial b_i}$, instead of the total $\frac{\delta}{\delta b_i}$, sensitivity was used), thus

$$\frac{\partial}{\partial x_k}\left(A_{nmk}\frac{\partial U_m}{\partial b_i}\right) = 0. \tag{36}$$

Equation 36 is solved separately for each of the N nodal sensitivities $\frac{\partial U_n}{\partial b_i}$. Then, the computation of $\frac{\delta U_n}{\delta b_i}$ is straightforward, based on the general expression

$$\frac{\delta \Phi}{\delta b_i} = \frac{\partial \Phi}{\partial b_i} + \frac{\partial \Phi}{\partial x_l}\frac{\delta x_l}{\delta b_i} \tag{37}$$

written for $\Phi = U_m$. In conformity with the discretization of the state equations, a Roe-like scheme is used for (36), as well. At any node P, the discretized forms of (36) become

$$\sum_{Q=1}^{Q(P)} g_{ni,PQ} = 0, \tag{38}$$

where the flux "along" a segment PQ, (Fig. 1), is given by

$$\begin{aligned} g^{inv}_{ni,PQ} = {}& \frac{1}{2}\left(A_{nk,P}\frac{\partial U_k}{\partial b_i}|_L + A_{nk,Q}\frac{\partial U_k}{\partial b_i}|_R\right) \\ & - \frac{1}{2}|A_{nk,PQ}|\left(\frac{\partial U_k}{\partial b_i}|_L - \frac{\partial U_k}{\partial b_i}|_R\right). \end{aligned} \tag{39}$$

The boundary conditions imposed to (36) are also in conformity with those imposed to the flow equations. For instance, for subsonic flows in ducts or cascades (where the stagnation pressure p_t, temperature T_t and flow angle α at the inlet, I, and the static pressure p at the outlet, O, are fixed), $\frac{\partial p_t}{\partial b_i}|_I = 0$, $\frac{\partial T_t}{\partial b_i}|_I = 0$, $\frac{\partial \alpha}{\partial b_i}|_I = 0$ and $\frac{\partial p}{\partial b_i}|_O = 0$, $i = 1, \dots, N$, should be enforced as boundary conditions.

The numerical fluxes $g_{ni} = \frac{\partial f^{inv}_{nk}}{\partial b_i}n_k$ across any solid wall element are computed by taking into account the no-penetration condition $u_k n_k = 0$ which, in a 3D case, leads to

$$\begin{aligned} \begin{bmatrix} g^{inv}_{1i} \\ g^{inv}_{2i} \\ g^{inv}_{3i} \\ g^{inv}_{4i} \\ g^{inv}_{5i} \end{bmatrix} = {}& \begin{bmatrix} 0 \\ n_1 \\ n_2 \\ n_3 \\ 0 \end{bmatrix}\frac{\partial p}{\partial b_i} - \begin{bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ \frac{E+p}{\rho} \end{bmatrix} \\ & \times \rho\left(u_k\frac{\delta n_k}{\delta b_i} + \frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k\right). \end{aligned} \tag{40}$$

3.2.3 The Continuous Adjoint Approach

According to Sect. 2.5, the adjoint variables or Lagrange multipliers Ψ_n are introduced in the expression defining the variation in F_{aug} as follows

$$\frac{\delta F_{aug}}{\delta b_i} = \frac{\delta F}{\delta b_i} + \int_\Omega \Psi_n\frac{\delta R^{inv}_n}{\delta b_i}d\Omega. \tag{41}$$

However, instead of using the (zero) total sensitivity of the flow equations, the (also zero) partial sensitivity of R^{inv}_n (i.e. $\frac{\partial R^{inv}_n}{\partial b_i}$) can be used. As it will be proved later on, this leads to sensitivities of same accuracy through less demanding computations. So, instead of (41), let

$$\frac{\delta F_{aug}}{\delta b_i} = \frac{\delta F}{\delta b_i} + \int_\Omega \Psi_n\frac{\partial R^{inv}_n}{\partial b_i}d\Omega. \tag{42}$$

Partial derivatives with respect to the design variables and grid coordinates may be interchanged, yielding

$$\begin{aligned}\int_\Omega \Psi_n \frac{\partial R_n^{inv}}{\partial b_i} d\Omega &= \int_\Omega \Psi_n \frac{\partial}{\partial b_i}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) d\Omega \\ &= \int_\Omega \Psi_n \frac{\partial}{\partial x_k}\left(\frac{\partial f_{nk}^{inv}}{\partial b_i}\right) d\Omega.\end{aligned} \tag{43}$$

Integration by parts and the application of the Gauss divergence theorem in (43) gives

$$\begin{aligned}\int_\Omega \Psi_n \frac{\partial R_n^{inv}}{\partial b_i} d\Omega &= -\int_\Omega \frac{\partial \Psi_n}{\partial x_k} \frac{\partial f_{nk}^{inv}}{\partial b_i} d\Omega \\ &+ \int_S \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS.\end{aligned} \tag{44}$$

The first term on the r.h.s. is further analyzed as

$$-\int_\Omega \frac{\partial \Psi_n}{\partial x_k} \frac{\partial f_{nk}^{inv}}{\partial b_i} d\Omega = -\int_\Omega A_{nmk} \frac{\partial \Psi_n}{\partial x_k} \frac{\partial U_m}{\partial b_i} d\Omega. \tag{45}$$

The second integral on the r.h.s of (44) splits into two integrals, one over the wall boundary and another at the inlet/outlet of the domain, as follows

$$\begin{aligned}\int_S \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS &= \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS \\ &+ \int_{S_{i,o}} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS.\end{aligned} \tag{46}$$

Along the solid wall boundaries, f_{nk}^{inv} depends on b_i both directly and indirectly (i.e. through the corresponding movement δx_l of solid wall nodes), so the total sensitivity of f_{nk}^{inv} is given by (37) for $\Phi = f_{nk}^{inv}$. Using (37), the first integral on the r.h.s of (46 is expressed as

$$\begin{aligned}\int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS &= \int_{S_w} \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_i} n_k dS \\ &- \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l} \frac{\delta x_l}{\delta b_i} n_k dS.\end{aligned} \tag{47}$$

The first integral on the r.h.s. of (47) yields

$$\begin{aligned}\int_{S_w} \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_i} n_k dS &= \int_{S_w} \Psi_n \frac{\delta \left(f_{nk}^{inv} n_k dS\right)}{\delta b_i} \\ &- \int_{S_w} \Psi_n f_{nk}^{inv} \frac{\delta \left(n_k dS\right)}{\delta b_i}.\end{aligned} \tag{48}$$

From the definition of the inviscid fluxes, (27), and considering the no-penetration condition for the velocity along the solid walls, we get

$$\begin{aligned}\begin{bmatrix} f_{1k}^{inv} \\ f_{2k}^{inv} \\ f_{3k}^{inv} \\ f_{4k}^{inv} \\ f_{5k}^{inv} \end{bmatrix} n_k|_{S_w} &= \begin{bmatrix} \rho u_k n_k \\ \rho u_k n_k u_1 + p\delta_{k1} n_k \\ \rho u_k n_k u_2 + p\delta_{k2} n_k \\ \rho u_k n_k u_3 + p\delta_{k3} n_k \\ u_k n_k (E+p) \end{bmatrix}_{S_w} \\ &= \begin{bmatrix} 0 \\ pn_1 \\ pn_2 \\ pn_3 \\ 0 \end{bmatrix}_{S_w} = p \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{bmatrix}_{S_w}.\end{aligned} \tag{49}$$

Consequently,

$$\frac{\delta}{\delta b_i}\left(f_{nk}^{inv} n_k|_{S_w}\right) = \frac{\delta p}{\delta b_i} N_n + p\frac{\delta N_n}{\delta b_i}. \tag{50}$$

Note that, in viscous flows, (50) is still valid since $u_k = 0$. Using (50), (48) becomes

$$\begin{aligned}\int_{S_w} \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_i} n_k dS &= \int_{S_w} \Psi_{k+1} n_k \frac{\delta p}{\delta b_i} dS \\ &+ \int_{S_w} \Psi_{k+1} p \frac{\delta \left(n_k dS\right)}{\delta b_i} \\ &- \int_{S_w} \Psi_n f_{nk}^{inv} \frac{\delta \left(n_k dS\right)}{\delta b_i} \\ &= \int_{S_w} \Psi_{k+1} n_k \frac{\delta p}{\delta b_i} dS + \\ &\int_{S_w} \left(\Psi_{k+1} p - \Psi_n f_{nk}^{inv}\right) \frac{\delta \left(n_k dS\right)}{\delta b_i}.\end{aligned} \tag{51}$$

By means of (43) to (51) the final expression for the integral that appears in (42) becomes

$$\begin{aligned}\int_\Omega \Psi_n \frac{\partial R_n^{inv}}{\partial b_i} d\Omega &= -\int_\Omega A_{nmk} \frac{\partial \Psi_n}{\partial x_k} \frac{\partial U_m}{\partial b_i} d\Omega \\ &+ \int_{S_{i,o}} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS \\ &+ \int_{S_w} \Psi_{k+1} n_k \frac{\delta p}{\delta b_i} dS \\ &+ \int_{S_w} \left(\Psi_{k+1} p - \Psi_n f_{nk}^{inv}\right) \frac{\delta \left(n_k dS\right)}{\delta b_i} \\ &- \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l} \frac{\delta x_l}{\delta b_i} n_k dS.\end{aligned} \tag{52}$$

The total sensitivities of the objective function of (31) are

$$\frac{\delta F}{\delta b_i} = \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_i} + \int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta b_i} dS. \tag{53}$$

Combining (52) and (53), the corresponding sensitivities of F_{aug} are expressed as

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_i} &= \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_i} \\ &\quad + \underbrace{\int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta b_i} dS}_{SWCR} \underbrace{\int_{\Omega} A_{nmk} \frac{\partial \Psi_n}{\partial x_k} \frac{\partial U_m}{\partial b_i} d\Omega}_{FAE} \\ &\quad + \underbrace{\int_{S_{i,o}} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial b_i} n_k dS}_{IOBC} + \underbrace{\int_{S_w} \Psi_{k+1} n_k \frac{\delta p}{\delta b_i} dS}_{SWCR} \\ &\quad + \int_{S_w} \left(\Psi_{k+1} p - \Psi_n f_{nk}^{inv}\right) \frac{\delta (n_k dS)}{\delta b_i} \\ &\quad - \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l} \frac{\delta x_l}{\delta b_i} n_k dS. \end{aligned} \tag{54}$$

In (54), the field integral depending on the partial sensitivities of the flow variables $\frac{\partial U_m}{\partial b_i}$ (marked with FAE) is eliminated by satisfying the field adjoint equations

$$-A_{nmk} \frac{\partial \Psi_n}{\partial x_k} = 0. \tag{55}$$

Like the flow equations, (55) are solved using a pseudo time-marching scheme, by introducing a pseudo-temporal term, as follows

$$\frac{\partial \Psi_m}{\partial t} - A_{nmk} \frac{\partial \Psi_n}{\partial x_k} = 0. \tag{56}$$

A Roe-like scheme is used for the discretization of the adjoint equations. According to this scheme, the adjoint fluxes crossing the boundaries of finite volumes, Fig. 1, are given by

$$\begin{aligned} t_{n,PQ}(P) &= \frac{1}{2} A_{kn,P}\left(\Psi_{k,L} + \Psi_{k,R}\right) \\ &\quad + \frac{1}{2} |A_{kn,PQ}| \left(\Psi_{k,L} - \Psi_{k,R}\right), \\ t_{n,PQ}(Q) &= \frac{1}{2} A_{kn,Q}\left(\Psi_{k,L} + \Psi_{k,R}\right) \\ &\quad + \frac{1}{2} |A_{kn,PQ}| \left(\Psi_{k,L} - \Psi_{k,R}\right), \end{aligned} \tag{57}$$

where $\Psi_{k,L}$ and $\Psi_{k,R}$ are expressed in terms of the nodal values and gradients of Ψ_k. The wall boundary conditions for the adjoint equations are derived from (54), by eliminating boundary integrals that depend on the sensitivities of the flow variables (marked with $SWCR$). So, the adjoint boundary condition over the wall boundary yields

$$p - p_{tar} + \Psi_{k+1} n_k = 0. \tag{58}$$

Equation 58 closely resembles the no-penetration condition along solid walls. When the optimal geometry is approached, p tends to p_{tar} and (58) postulates that the adjoint velocity $\Psi_{k+1} n_k$ normal to the wall vanishes.

The inlet/outlet conditions are derived by eliminating the integrals marked with $IOBC$ from (54). By taking into account that Q_I and Q_O flow variables are fixed at the inlet and outlet, respectively, the elimination of the aforementioned integrals leads to $5 - Q_I$ (or $4 - Q_I$ in 2D flows) fixed adjoint variables at the inlet and $5 - Q_O$ (or $4 - Q_O$) at the outlet. The remaining ones are extrapolated from the interior of the flow domain.

After eliminating the field and boundary integrals depending on $\frac{\partial U_m}{\partial b_i}$, the remaining terms in (54) provide the sensitivity derivatives of the objective function as follows

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_i} &= \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_i} \\ &\quad + \int_{S_w} \left(\Psi_{k+1} p - \Psi_n f_{nk}^{inv}\right) \frac{\delta (n_k dS)}{\delta b_i} \\ &\quad - \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l} \frac{\delta x_l}{\delta b_i} n_k dS. \end{aligned} \tag{59}$$

This expression is free of field integrals of any derivative of geometrical quantities with respect to the design variables (the so-called geometrical sensitivities). This is advantageous since, repetitive remeshing processes within each optimization cycle would be needed otherwise. The computation of sensitivities is, thus, faster and more precise. According to (59), the sensitivities of the objective function are expressed exclusively in terms of boundary integrals along the solid walls.

3.2.4 Inverse Design with Targets other than Pressure

In the continuous adjoint approach for inverse design problems, whenever the objective function does not depend on pressure, the derivation of the adjoint boundary conditions along the wall is not straightforward. Using a pressure-based functional, such as the one of (31), and the Euler equations as state equations, the use of the Gauss divergence theorem in the development of $\frac{\delta F_{aug}}{\delta b_i}$ leads to an integral over the solid walls that depends on the pressure sensitivity with respect to the design variables. In these integrals, the effect of pressure sensitivity is eliminated by satisfying the adjoint wall boundary conditions, (58). On the opposite, if F is defined in terms of velocities, i.e.

$$F=\frac{1}{2}\int_{S_w}(u-u_{tar})^2dS, \tag{60}$$

where $u_{tar}(S)$ is the target velocity distribution, a relation similar to (58) cannot be derived and integrals containing

$$A_u\frac{\delta u}{\delta b_i}+A_p\frac{\delta p}{\delta b_i} \tag{61}$$

cannot be eliminated. A way to overcome this problem has been proposed in [53], where additional sets of adjoint variables are introduced over the wall boundary. A straightforward handling of the so-called inadmissible objective functions, such as that of (60), can be found in [54], where the sensitivity of pressure is associated with the sensitivities of other flow variables. With the objective function of (60), the correlation between pressure and velocity sensitivities, [54],

$$\frac{\delta p}{\delta b_i}=-\rho u\frac{\delta u}{\delta b_i} \tag{62}$$

can be used, leading to a new adjoint boundary condition at the wall, namely

$$(u-u_{tar})-\rho u\Psi_{k+1}n_k=0. \tag{63}$$

3.2.5 An Alternative Continuous Adjoint Approach Based on Metrics

An alternative continuous adjoint approach can be developed starting from (41), instead of (42). Without loss in generality, using the standard transformation ($x_i\leftrightarrow\xi^i$) associated with structured grids, according to which $\frac{\partial\Phi}{\partial x_k}=\frac{\partial\Phi}{\partial\xi^m}\frac{\partial\xi^m}{\partial x_k}$, we get

$$\frac{\delta}{\delta b_i}\left(\frac{\partial\Phi}{\partial x_k}\right)=\frac{\partial}{\partial x_k}\left(\frac{\delta\Phi}{\delta b_i}\right)+\frac{\partial\Phi}{\partial\xi^m}\frac{\delta}{\delta b_i}\left(\frac{\partial\xi^m}{\partial x_k}\right), \tag{64}$$

where $\frac{\partial\xi^m}{\partial x_k}$ are the grid metrics. Since the transformed grid is invariant with respect to design variables, in (64), the relation $\frac{\delta}{\delta b_i}(\frac{\partial\Phi}{\partial\xi^m})=\frac{\partial}{\partial\xi^m}(\frac{\delta\Phi}{\delta b_i})$ was used. Then,

$$\int_\Omega\Psi_n\frac{\delta}{\delta b_i}\left(\frac{\partial f^{inv}_{nk}}{\partial x_k}\right)d\Omega=\int_\Omega\Psi_n\frac{\partial}{\partial x_k}\left(\frac{\delta f^{inv}_{nk}}{\delta b_i}\right)d\Omega+\int_\Omega\Psi_n\frac{\partial f^{inv}_{nk}}{\partial\xi^m}\frac{\delta}{\delta b_i}\left(\frac{\partial\xi^m}{\partial x_k}\right)d\Omega. \tag{65}$$

In this case, the sensitivities of F_{aug} are given by

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_i}&=\frac{1}{2}\int_{S_w}(p-p_{tar})^2\frac{\delta(dS)}{\delta b_i}+\underbrace{\int_{S_w}(p-p_{tar})\frac{\delta p}{\delta b_i}dS}_{SWCR}\\&\quad-\underbrace{\int_\Omega A_{nmk}\frac{\partial\Psi_n}{\partial x_k}\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE}+\underbrace{\int_{S_{i,o}}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial b_i}n_kdS}_{IOBC}\\&\quad+\underbrace{\int_{S_w}\Psi_{k+1}n_k\frac{\delta p}{\delta b_i}dS}_{SWCR}\\&\quad+\int_{S_w}\left(\Psi_{k+1}p-\Psi_nf^{inv}_{nk}\right)\frac{\delta\left(n_kdS\right)}{\delta b_i}\\&\quad+\int_\Omega\Psi_n\frac{\partial f_{nk}}{\partial\xi^m}\frac{\delta}{\delta b_i}\left(\frac{\partial\xi^m}{\partial x_k}\right)d\Omega.\end{aligned} \tag{66}$$

The elimination of terms marked with FAE, $SWCR$ and $IOBC$ gives rise to the same field adjoint equations and boundary conditions as those presented in the previous section. However, the gradient of the objective function is expressed differently, as follows

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_i}&=\frac{1}{2}\int_{S_w}(p-p_{tar})^2\frac{\delta(dS)}{\delta b_i}\\&\quad+\int_{S_w}\left(\Psi_{k+1}p-\Psi_nf^{inv}_{nk}\right)\frac{\delta\left(n_kdS\right)}{\delta b_i}\\&\quad+\int_\Omega\Psi_n\frac{\partial f_{nk}}{\partial\xi^m}\frac{\delta}{\delta b_i}\left(\frac{\partial\xi^m}{\partial x_k}\right)d\Omega.\end{aligned} \tag{67}$$

In contrast to (59), (67) contains a field integral with the sensitivities of grid metrics. In order to compute these sensitivities at any optimization cycle, a finite difference approach based on the generation of new meshes as many times as the number of the design variables should be used. This makes this approach quite costly and prone to numerical errors, due to the selection of the finite difference step value. In a subsequent section, a brief presentation of the transformation of the field integral of metrics to a boundary integral, see [55], is presented.

3.2.6 Inverse Design—Applications

The adjoint formulations presented in the previous sections are demonstrated by performing the inverse design of a 2D compressor and a 2D turbine cascade. In both cases, the target is a known pressure distribution along the airfoil contours. First order sensitivity derivatives computed using the adjoint approach are compared with finite differences. Based on the so-computed sensitivities, different gradient-based algorithms are used and their convergence behaviours are plotted and compared. Finally, the optimal airfoils are compared

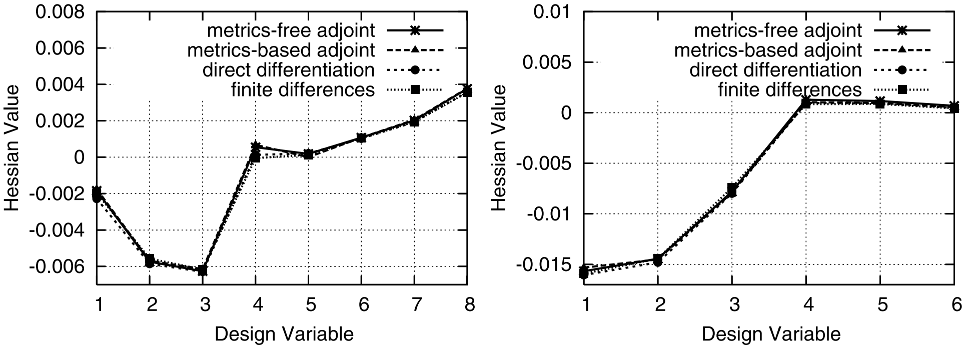


Fig. 2 Inverse design of a 2D turbine cascade, *left*, and a 2D compressor cascade, *right*. First order sensitivities of the objective function computed using the metrics-free continuous adjoint approach, (59), a metrics-based continuous adjoint approach based on metrics, (67), the direct differentiation method, (53) and a central finite difference scheme. The first half values correspond to the design variables of the suction side and the rest to those of the pressure side

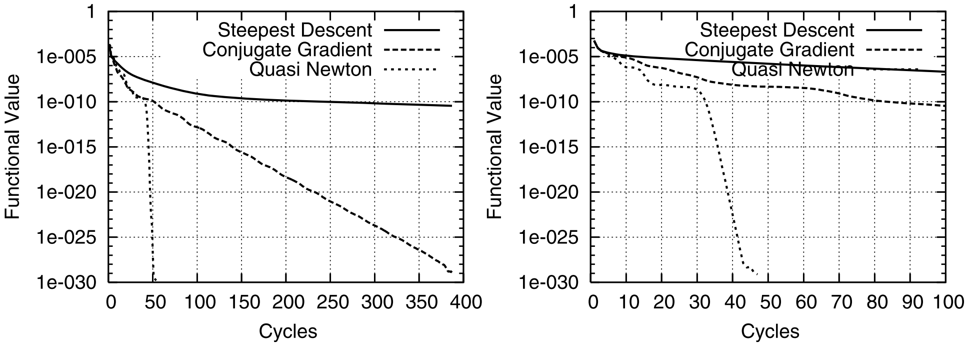


Fig. 3 Inverse design of a 2D turbine cascade, *left*, and a 2D compressor cascade, *right*. Reduction rate of the objective function value using three optimization algorithms; steepest descent, Fletcher–Reeves conjugate gradient and BFGS quasi Newton algorithm

with reference shapes, i.e. those used to derive the target pressure distributions.

The blade airfoil shapes are parameterized using Bézier-Bernstein polynomials. For the turbine cascade, each airfoil side is parameterized using eight control points. The first two and last two control points per blade airfoil side are kept constant resulting to a total number of eight design variables. The camber and stagger angles are fixed to 60° and 25°, respectively; the pitch-to-chord ratio is equal to 0.8. The inlet flow is axial and the exit isentropic Mach number is equal to 0.7.

For the compressor cascade, the airfoil suction and pressure sides are formed by two arcs, the stagger angle is 30° and the pitch-to-chord ratio is 0.6. The inlet flow angle and the exit isentropic Mach number are equal to 56° and 0.36, respectively. The two airfoil sides are parameterized using seven control points each. Here, also, the first two and last two control point points at each side are fixed, so the total number of design variables is six.

The first order sensitivity derivatives of F, computed using four different approaches are plotted in Fig. 2. The four curves are almost identical, proving that all four approaches are equally accurate. However, these approaches have different computational costs, since a different number of equivalent flow solutions is required by each of them. The most costly algorithm is the finite difference scheme. It requires $2N$ calls to the flow solver. We recall that, at a preprocessing level, a parametric study was necessary to select the proper ϵ value. Note, also, that the convergence of the flow equations to machine accuracy is indispensable when computing sensitivities through finite differences. The direct differentiation method (numerical solution of (36) for $\frac{\partial U_k}{\partial b_i}$) has only half of the previous cost (i.e. N equivalent flow solutions) and overcomes ambiguities related to the proper selection of ϵ.

As already mentioned, the adjoint approaches are the most efficient ones, since the CPU cost for the computation of the first order sensitivities is almost twice that for solving the flow equations. In Fig. 2 two adjoint approaches are presented. The metrics-based adjoint approach uses (67) and requires some extra cost for remeshing the computational domain ($2N$ mesh regenerations within bifurcated boundaries) so as to compute metrics variations $\frac{\delta}{\delta b_i}(\frac{\partial \xi^m}{\partial x_k})$. The metrics-free adjoint, i.e. the use of (59), avoids any auxiliary remeshing and the so-induced inaccuracies and produces very accurate sensitivities.

In Fig. 3, three gradient-based optimization algorithms are used and compared in terms of the objective function value reduction rate: steepest descent, the Fletcher–Reeves

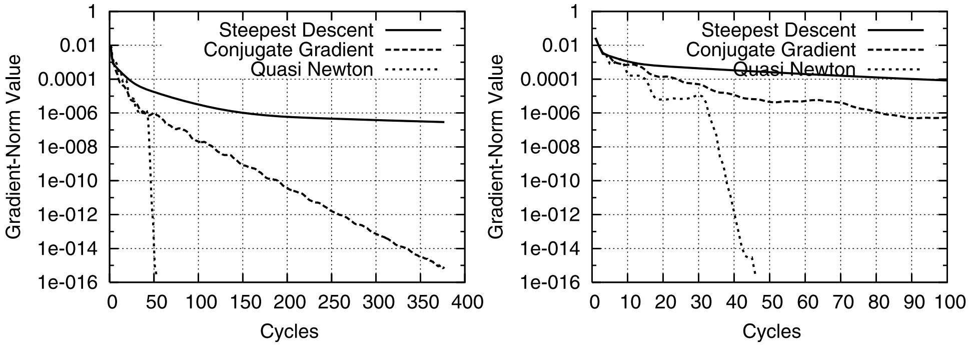


Fig. 4 Inverse design of a 2D turbine cascade, *left*, and a 2D compressor cascade, *right*. Reduction rate of the first order sensitivity values of the objective function using three gradient-based optimization algorithms; steepest descent, Fletcher–Reeves conjugate gradient and BFGS quasi Newton algorithm

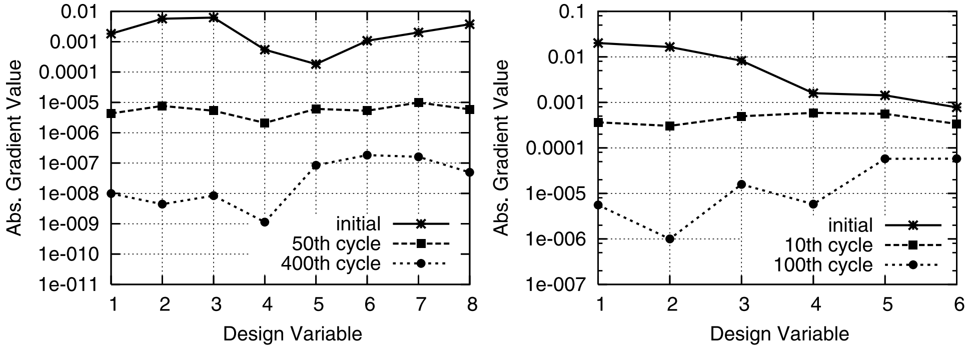


Fig. 5 Inverse design of a 2D turbine cascade, *left*, and a 2D compressor cascade, *right*. First order sensitivities of F computed during the steepest descent method (absolute values, semi-log scale) at three cycles within the optimization loop

conjugate gradient and the BFGS (Broyden, Fletcher, Goldfarb, Shanno, [8]) quasi Newton algorithm. Using the same three gradient-based algorithms, the reduction rate of the first order sensitivity norm is shown in Fig. 4. Figures 3 and 4 reveal the same convergence behaviour. Steepest descent (with an “optimal” fixed stepsize) is the slowest method. A much better performance is that of conjugate gradient but the quasi Newton method by far outperforms even the conjugate gradients.

The change in first order sensitivities during a steepest descent based optimization is shown in Fig. 5. The first order sensitivity values are reduced by almost three orders of magnitude within one hundred cycles.

In Fig. 6, the initial, optimal and reference airfoils as well as the corresponding pressure distributions are plotted. The optimal shapes and the corresponding pressure distributions shown herein have been obtained using the quasi Newton algorithm. The optimal pressure distribution coincides with the target one. The same conclusion can be drawn by comparing the optimal and reference airfoil contours, although this was not (direct) objective of the inverse design problems.

3.3 Inverse Design Using the Navier–Stokes Equations

In viscous flows, the continuous adjoint approach requires the development of the additional term $\int_\Omega \Psi_n \frac{\partial R_n^{vis}}{\partial b_i} d\Omega$ which appears in (42). Equations (43) and (44) are still valid, both for f_{nk}^{inv} and f_{nk}^{vis}. So,

$$-\int_\Omega \Psi_n \frac{\partial}{\partial b_i}\left(\frac{\partial f_{nk}^{vis}}{\partial x_k}\right) d\Omega = -\int_\Omega \Psi_n \frac{\partial}{\partial x_k}\left(\frac{\partial f_{nk}^{vis}}{\partial b_i}\right) d\Omega$$
$$= \int_\Omega \frac{\partial \Psi_n}{\partial x_k}\frac{\partial f_{nk}^{vis}}{\partial b_i} d\Omega$$
$$-\int_S \Psi_n \frac{\partial f_{nk}^{vis}}{\partial b_i} n_k dS. \tag{68}$$

The first term on the r.h.s. of (68) is developed as

$$\int_\Omega \frac{\partial \Psi_n}{\partial x_k}\frac{\partial f_{nk}^{vis}}{\partial b_i} d\Omega = \int_\Omega \left[\frac{\partial \tau_{km}}{\partial b_i}\left(\frac{\partial \Psi_{m+1}}{\partial x_k} + u_m \frac{\partial \Psi_\Lambda}{\partial x_k}\right)\right.$$
$$\left. + \frac{\partial u_m}{\partial b_i}\tau_{km}\frac{\partial \Psi_\Lambda}{\partial x_k} + \frac{\partial q_k}{\partial b_i}\frac{\partial \Psi_\Lambda}{\partial x_k}\right] d\Omega, \tag{69}$$

where, in what follows, $\Lambda = 4$ for 2D flows and $\Lambda = 5$ for 3D flows. The partial sensitivities of the viscous stresses can be written as

$$\frac{\partial \tau_{km}}{\partial b_i} = \mu\left[\frac{\partial}{\partial x_m}\left(\frac{\partial u_k}{\partial b_i}\right) + \frac{\partial}{\partial x_k}\left(\frac{\partial u_m}{\partial b_i}\right)\right]$$
$$+ \lambda \delta_{km}\frac{\partial}{\partial x_l}\left(\frac{\partial u_l}{\partial b_i}\right) \tag{70}$$

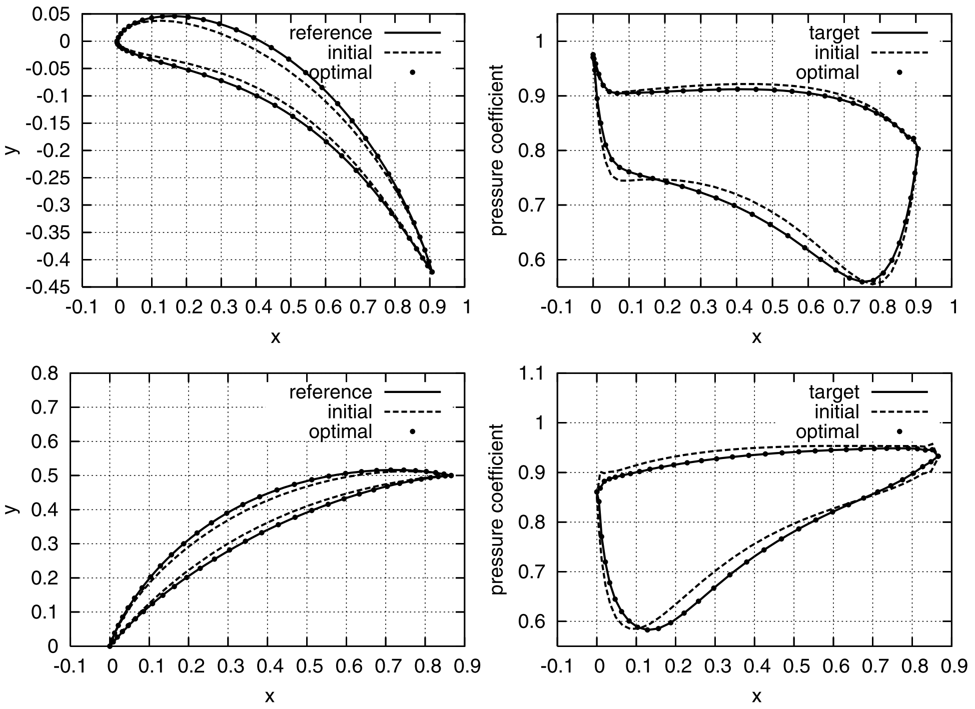


Fig. 6 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Initial, optimal and reference cascade airfoils and the corresponding pressure distributions. We recall that reference airfoils are those used to produce the target pressure distributions

and the first part on the r.h.s. of (69), can be integrated by parts. After rearranging the indices, we get

$$\begin{aligned}
&\int_\Omega \frac{\partial \tau_{km}}{\partial b_i}\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\right)d\Omega\\
&\quad=\int_\Omega\left\{\mu\left[\frac{\partial}{\partial x_m}\left(\frac{\partial u_k}{\partial b_i}\right)+\frac{\partial}{\partial x_k}\left(\frac{\partial u_m}{\partial b_i}\right)\right]\right.\\
&\qquad\left.+\lambda\delta_{km}\frac{\partial}{\partial x_l}\left(\frac{\partial u_l}{\partial b_i}\right)\right\}\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\right)d\Omega\\
&\quad=-\int_\Omega \frac{\partial u_k}{\partial b_i}\frac{\partial}{\partial x_m}\left[\mu\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}+\frac{\partial \Psi_{k+1}}{\partial x_m}\right.\right.\\
&\qquad\left.\left.+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\right)+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\right]d\Omega\\
&\qquad+\int_S \frac{\partial u_k}{\partial b_i}\left[\mu\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}+\frac{\partial \Psi_{k+1}}{\partial x_m}\right.\right.\\
&\qquad\left.\left.+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\right)+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\right]n_m dS. \qquad (71)
\end{aligned}$$

The last term on the r.h.s. of (69) is also integrated by parts to become

$$\begin{aligned}
\int_\Omega \frac{\partial q_k}{\partial b_i}\frac{\partial \Psi_\Lambda}{\partial x_k}d\Omega &= \int_\Omega k\frac{\partial}{\partial x_k}\left(\frac{\partial T}{\partial b_i}\right)\frac{\partial \Psi_\Lambda}{\partial x_k}d\Omega\\
&=-\int_\Omega \frac{\partial T}{\partial b_i}\frac{\partial}{\partial x_k}\left(k\frac{\partial \Psi_\Lambda}{\partial x_k}\right)d\Omega\\
&\quad+\int_S \frac{\partial T}{\partial b_i}\left(k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k\right)dS. \qquad (72)
\end{aligned}$$

The second integral on the r.h.s. of (68) is written as

$$\begin{aligned}
-\int_S \Psi_n\frac{\partial f^{vis}_{nk}}{\partial b_i}n_k dS = -\int_S\Bigg[&(\Psi_{k+1}+u_k\Psi_\Lambda)\frac{\partial \tau_{km}}{\partial b_i}\\
&+\Psi_\Lambda\tau_{km}\frac{\partial u_k}{\partial b_i}+\Psi_\Lambda\frac{\partial q_m}{\partial b_i}\Bigg]n_m dS.
\end{aligned} \qquad (73)$$

The total first order sensitivities of the viscous stresses are expressed using (37), for $\Phi=\tau_{km}$. Also, the use of the identity $\tau_{km}n_kn_m=0$ leads to

$$\frac{\delta(\tau_{km}n_kn_m)}{\delta b_i}=\frac{\delta\tau_{km}}{\delta b_i}n_kn_m+\tau_{km}\frac{\delta(n_kn_m)}{\delta b_i}=0. \qquad (74)$$

Using the equation of state $p=R\rho T$, the sensitivities of temperature are written in terms of density and pressure sensitivities as

$$\frac{\partial T}{\partial b_i}=-\frac{p}{R\rho^2}\frac{\partial \rho}{\partial b_i}+\frac{1}{R\rho}\frac{\partial p}{\partial b_i}. \qquad (75)$$

Also, the sensitivities of the non-conservative flow variables $[V_1,V_2,V_3,V_4,V_5]=[\rho,u_1,u_2,u_3,p]$ are expressed in terms of those of the conservative ones U_m as

$$\frac{\partial V_k}{\partial b_i}=\frac{\partial V_k}{\partial U_m}\frac{\partial U_m}{\partial b_i}. \qquad (76)$$

Using (69) to (76), the variation in the viscous term, (68), is finally given by

$$
\begin{aligned}
&-\int_{\Omega}\Psi_n\frac{\partial}{\partial b_i}\left(\frac{\partial f_{nk}^{vis}}{\partial x_k}\right)d\Omega\\
&\quad=\int_{\Omega}\frac{\partial V_k}{\partial U_m}K_k\frac{\partial U_m}{\partial b_i}d\Omega\\
&\qquad+\int_S\frac{\delta u_k}{\delta b_i}\left[\mu\left(\frac{\partial\Psi_{m+1}}{\partial x_k}+u_m\frac{\partial\Psi_\Lambda}{\partial x_k}\right.\right.\\
&\qquad+\left.\frac{\partial\Psi_{k+1}}{\partial x_m}+u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\right)\\
&\qquad+\left.\lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l}+u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\right]n_m dS\\
&\qquad-\int_S\left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\left[\mu\left(\frac{\partial\Psi_{m+1}}{\partial x_k}+u_m\frac{\partial\Psi_\Lambda}{\partial x_k}\right.\right.\\
&\qquad+\left.\frac{\partial\Psi_{k+1}}{\partial x_m}+u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\right)\\
&\qquad+\left.\lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l}+u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\right]n_m dS\\
&\qquad+\int_S\frac{\delta T}{\delta b_i}k\frac{\partial\Psi_\Lambda}{\partial x_k}n_k dS+\int_S\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_kn_m)}{\delta b_i}dS\\
&\qquad-\int_S\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}k\frac{\partial\Psi_\Lambda}{\partial x_k}n_k dS\\
&\qquad-\int_S\frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_kn_m)}{\delta b_i}dS+\int_S\Psi_{k+1}\frac{\partial\tau_{km}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_m dS\\
&\qquad+\int_S\Psi_\Lambda\tau_{km}\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_m dS\\
&\qquad+\int_S\Psi_\Lambda\frac{\partial q_m}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_m dS-\int_S\Psi_\Lambda\frac{\delta(q_mn_m)}{\delta b_i}dS\\
&\qquad+\int_S\Psi_\Lambda q_m\frac{\delta n_m}{\delta b_i}dS, \qquad (77)
\end{aligned}
$$

where

$$
\begin{aligned}
K_1&=-\frac{T}{\rho}\frac{\partial}{\partial x_j}\left(k\frac{\partial\Psi_\Lambda}{\partial x_j}\right),\\
K_{i+1}&=\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial\Psi_{j+1}}{\partial x_i}+u_j\frac{\partial\Psi_m}{\partial x_i}+\frac{\partial\Psi_{i+1}}{\partial x_j}+u_i\frac{\partial\Psi_m}{\partial x_j}\right)\right. \qquad (78)\\
&\qquad+\left.\lambda\delta_{km}\left(\frac{\partial\Psi_{k+1}}{\partial x_k}+u_k\frac{\partial\Psi_m}{\partial x_k}\right)\right]-\tau_{ij}\frac{\partial\Psi_\Lambda}{\partial x_j},\\
&\quad i=1,\ldots,\Lambda-2,\\
K_\Lambda&=\frac{1}{R\rho}\frac{\partial}{\partial x_j}\left(k\frac{\partial\Psi_\Lambda}{\partial x_j}\right).
\end{aligned}
$$

The sensitivities of the inviscid and viscous parts together with that of the objective function lead to the total sensitivity of F_{aug}. Using (52), (53) and (77), the expressions for the viscous fluxes and their variations and taking also into account that the inlet/outlet nodal sensitivities $\frac{\delta x_l}{\delta b_i}$ are zero as well as $u_k=0$ and $\frac{\delta u_k}{\delta b_i}=0$ over the walls, one gets

$$
\begin{aligned}
\frac{\delta F_{aug}}{\delta b_i}&=\frac{1}{2}\int_{S_w}(p-p_{tar})^2\frac{\delta(dS)}{\delta b_i}+\underbrace{\int_{S_w}(p-p_{tar})\frac{\delta p}{\delta b_i}dS}_{SWCR1}\\
&\quad-\underbrace{\int_{\Omega}A_{nmk}\frac{\partial\Psi_n}{\partial x_k}\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE}+\underbrace{\int_{S_{i,o}}\Psi_n\frac{\partial f_{nk}^{inv}}{\partial b_i}n_k dS}_{IOBC}\\
&\quad+\underbrace{\int_{S_w}\Psi_{k+1}n_k\frac{\delta p}{\delta b_i}dS}_{SWCR1}\\
&\quad+\int_{S_w}\left(\Psi_{k+1}p-\Psi_n f_{nk}^{inv}\right)\frac{\delta(n_kdS)}{\delta b_i}\\
&\quad-\int_{S_w}\Psi_n\frac{\partial f_{nk}^{inv}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS\\
&\quad-\underbrace{\int_{\Omega}\frac{\partial V_k}{\partial U_m}K_k\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE}\\
&\quad+\underbrace{\int_{S_{i,o}}\frac{\delta u_k}{\delta b_i}\left[\mu\left(\frac{\partial\Psi_{m+1}}{\partial x_k}+u_m\frac{\partial\Psi_\Lambda}{\partial x_k}\right.\right.}_{IOBC...}\\
&\quad\underbrace{+\left.\frac{\partial\Psi_{k+1}}{\partial x_m}+u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\right)}_{...IOBC...}\\
&\quad\underbrace{+\left.\lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l}+u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\right]n_m dS}_{...IOBC}\\
&\quad-\int_{S_w}\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\left[\mu\left(\frac{\partial\Psi_{m+1}}{\partial x_k}+u_m\frac{\partial\Psi_\Lambda}{\partial x_k}\right.\right.\\
&\quad+\left.\frac{\partial\Psi_{k+1}}{\partial x_m}+u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\right)\\
&\quad+\left.\lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l}+u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\right]n_m dS\\
&\quad+\underbrace{\int_{S_w}\frac{\delta T}{\delta b_i}k\frac{\partial\Psi_\Lambda}{\partial x_k}n_k dS}_{SWCR2}\\
&\quad+\underbrace{\int_{S_w}\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_kn_m)}{\delta b_i}dS}_{SWCR1}\\
&\quad-\int_{S_w}\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}k\frac{\partial\Psi_\Lambda}{\partial x_k}n_k dS
\end{aligned}
$$

$$
\begin{aligned}
&-\int_{S_w} \frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS \\
&+\int_{S_w} \Psi_n \frac{\partial f_{nk}^{vis}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS \\
&+\underbrace{\int_{S_w} \Psi_\Lambda \frac{\delta(q_m n_m)}{\delta b_i}dS}_{SWCR2} + \int_{S_w} \Psi_\Lambda q_m \frac{\delta n_m}{\delta b_i}dS \\
&+\underbrace{\int_{S_{i,o}} \frac{\delta T}{\delta b_i} k \frac{\partial \Psi_\Lambda}{\partial x_k} n_k dS}_{IOBC} \\
&+\underbrace{\int_{S_{i,o}} \frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_k n_m)}{\delta b_i}dS}_{IOBC} \\
&-\underbrace{\int_{S_{i,o}} \Psi_\Lambda \frac{\delta(q_m n_m)}{\delta b_i}dS}_{IOBC}.
\end{aligned}
\qquad (79)
$$

The field integrals that depend on the variation in the flow variables (FAE) produce the field adjoint equations

$$
\frac{\partial \Psi_m}{\partial t} - A_{nmk}\frac{\partial \Psi_n}{\partial x_k} - \frac{\partial V_k}{\partial U_m}K_k = 0. \qquad (80)
$$

The wall boundary integrals that depend on flow variable sensitivities produce the adjoint boundary conditions. The terms marked with $SWCR1$ lead to the adjoint velocity boundary conditions. So, the elimination of the first two $SWCR1$ integrals is possible by satisfying (58). On the other hand, the third $SWCR1$ integral vanishes if the conditions

$$
\Psi_{k+1} = cn_k \qquad (81)
$$

($k = 1, \dots, \Lambda - 2$) where c is a constant, are satisfied. The satisfaction of both (58) and (81) leads to the boundary conditions for the adjoint velocities (i.e. Ψ_2, Ψ_3 and Ψ_4 in 3D)

$$
\Psi_{k+1} = -(p - p_{tar})n_k. \qquad (82)
$$

The boundary condition for the last adjoint variable (Ψ_Λ) is derived by the elimination of those $SWCR2$ boundary integrals that depend on $\frac{\delta U_k}{\delta b_i}$. If, along the solid wall, the temperature is fixed, the first term marked with $SWCR2$ vanishes, since $\frac{\delta T}{\delta b_i} = 0$. For the second $SWCR2$ integral to vanish, the condition $\Psi_\Lambda = 0$ over the wall boundary should be imposed. For adiabatic walls, $\frac{\delta(q_m n_m)}{\delta b_i} = 0$ and the second integral is zero, so the elimination of the first integral leads to Neumann conditions for Ψ_Λ, i.e. $\frac{\partial \Psi_\Lambda}{\partial x_i}n_i = 0$. We may summarize these conditions as follows

$$
\begin{aligned}
&T = const. \quad \Rightarrow \quad \Psi_\Lambda = 0, \\
&\frac{\partial T}{\partial n} = 0 \quad \Rightarrow \quad \frac{\partial \Psi_\Lambda}{\partial n} = 0.
\end{aligned}
\qquad (83)
$$

The adjoint boundary conditions at the inlet and outlet of the domain result from the elimination of the five integrals marked with $IOBC$ in (79). These conditions are practically determined by the first $IOBC$ integral, as in the inviscid inverse design problems. The other integrals can be eliminated since the spatial derivatives of the flow and adjoint variables over the inlet and outlet vanish (flow uniformity).

The expression for the functional sensitivities is, finally, given by

$$
\begin{aligned}
\frac{\delta F_{aug}}{\delta b_i} = &\frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta(dS)}{\delta b_i} \\
&+ \int_{S_w} \left(\Psi_{k+1}p - \Psi_n f_{nk}^{inv}\right)\frac{\delta(n_k dS)}{\delta b_i} \\
&- \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS \\
&- \int_{S_w} \left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\Bigg[\mu\left(\frac{\partial \Psi_{m+1}}{\partial x_k} + u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\right. \\
&\left. + \frac{\partial \Psi_{k+1}}{\partial x_m} + u_k \frac{\partial \Psi_\Lambda}{\partial x_m}\right) \\
&+ \lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l} + u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\Bigg]n_m dS \\
&- \int_{S_w} \left(\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS \\
&- \int_{S_w} \frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS \\
&+ \int_{S_w} \Psi_n \frac{\partial f_{nk}^{vis}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS + \int_{S_w} \Psi_\Lambda q_m \frac{\delta n_m}{\delta b_i}dS.
\end{aligned}
\qquad (84)
$$

Similarly to the equations governing inverse designs based on the Euler equations, the expression of the sensitivity derivatives is still free of field integrals. The use of (84) for the computation of the gradient does not require auxiliary grid generations to numerically compute field integrals. The lack of field integrals of geometrical sensitivities and the exclusive presence of boundary integrals considerably lowers the CPU cost.

3.4 Shape Optimization for Minimum Viscous Losses

3.4.1 Minimization of Entropy Generation

A different functional which is appropriate for use in optimization problems in internal aerodynamics (design of ducts or turbomachinery cascades) is the one expressing the entropy generation, throughout the flow domain due to viscous stresses. This is expressed as

$$
F = \int_\Omega \rho u_i \frac{\partial s}{\partial x_i} d\Omega, \qquad (85)
$$

where s is the (specific) entropy. Through the Gauss divergence theorem and by taking into account the continuity equation for compressible flows, F can be transformed to

$$F = \int_{S_i} \rho u_n s dS - \int_{S_o} \rho u_n s dS, \tag{86}$$

where u_n is the normal to the inlet/outlet boundary velocity. Note that other integrals along the solid walls or periodic boundaries, if any, vanish. As expected, (86) expresses the difference in entropy between the inlet and outlet of the flow domain.

According to [56–58], F can be transformed to a field integral in terms of temperature T, stresses τ_{km} and velocity gradients, as follows

$$F = \int_\Omega \frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m} d\Omega. \tag{87}$$

The continuous adjoint formulation for this functional has been presented by the authors in [59]. According to (87), the sensitivity of F is written as

$$\frac{\delta F}{\delta b_i} = \int_\Omega \frac{\delta}{\delta b_i}\left(\frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\right) d\Omega + \int_\Omega \frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta(d\Omega)}{\delta b_i}. \tag{88}$$

The last integral on the r.h.s. of (88) can be treated in two different ways. A first way is to keep it as a field integral. A more appealing way of handling it is by transforming it in such a way that the final expression of $\frac{\delta F}{\delta b_i}$ does not depend on sensitivities of internal grid coordinates, although F itself is a field integral. This can be done by means of the following equation

$$\frac{\delta(d\Omega)}{\delta b_i} = \frac{\partial}{\partial x_l}\left(\frac{\delta x_l}{\delta b_i}\right) d\Omega. \tag{89}$$

The proof of (89), for a 2D grid, follows. Although the proof makes use of metrics, (89) is metrics-free and can be generalized to any type of grid, either structured or unstructured. $d\Omega$ is written in terms of the volume $d\Omega_\xi$ defined in the computational or transformed domain (ξ^1, ξ^2) or (ξ^1, ξ^2, ξ^3) as

$$d\Omega = J d\Omega_\xi, \tag{90}$$

where J is the Jacobian of the grid transformation, given by

$$J = \frac{1}{6}\frac{\partial x_m}{\partial \xi^r}\frac{\partial x_p}{\partial \xi^j}\frac{\partial x_q}{\partial \xi^k}\epsilon_{rjk}\epsilon_{mpq}, \tag{91}$$

where ϵ_{rjk} is the sign of the permutation (rjk) of (123). The computational or transformed domain is absolutely invariant to any variation in the design variables. From (91),

$$\begin{aligned}\frac{\delta J}{\delta b_i} &= \frac{1}{6}\frac{\partial}{\partial \xi^r}\left(\frac{\delta x_m}{\delta b_i}\right)\frac{\partial x_p}{\partial \xi^j}\frac{\partial x_q}{\partial \xi^k}\epsilon_{rjk}\epsilon_{mpq} \\ &+ \frac{1}{6}\frac{\partial x_m}{\partial \xi^r}\frac{\partial}{\partial \xi^j}\left(\frac{\delta x_p}{\delta b_i}\right)\frac{\partial x_q}{\partial \xi^k}\epsilon_{rjk}\epsilon_{mpq} \\ &+ \frac{1}{6}\frac{\partial x_m}{\partial \xi^r}\frac{\partial x_p}{\partial \xi^j}\frac{\partial}{\partial \xi^k}\left(\frac{\delta x_q}{\delta b_i}\right)\epsilon_{rjk}\epsilon_{mpq}\end{aligned} \tag{92}$$

which, using the identity

$$\frac{\partial \xi^r}{\partial x_m} = \frac{1}{2J}\frac{\partial x_p}{\partial \xi^j}\frac{\partial x_q}{\partial \xi^k}\epsilon_{rjk}\epsilon_{mpq} \tag{93}$$

becomes

$$\frac{\delta J}{\delta b_i} = \frac{J}{3}\left[\frac{\partial}{\partial \xi^r}\left(\frac{\delta x_m}{\delta b_i}\right)\frac{\partial \xi^r}{\partial x_m} + \frac{\partial}{\partial \xi^j}\left(\frac{\delta x_p}{\delta b_i}\right)\frac{\partial \xi^j}{\partial x_p} + \frac{\partial}{\partial \xi^k}\left(\frac{\delta x_q}{\delta b_i}\right)\frac{\partial \xi^k}{\partial x_q}\right] \tag{94}$$

or, finally,

$$\frac{\delta J}{\delta b_i} = J\frac{\partial}{\partial x_l}\left(\frac{\delta x_l}{\delta b_i}\right). \tag{95}$$

Then, according to (90), the sensitivity of $d\Omega$ is written as

$$\frac{\delta(d\Omega)}{\delta b_i} = \frac{\delta J}{\delta b_i} d\Omega_\xi = \frac{\partial}{\partial x_l}\left(\frac{\delta x_l}{\delta b_i}\right) J d\Omega_\xi$$

as was to be proved.

The two integrals on the r.h.s. of (88) can be developed as follows

$$\begin{aligned}&\int_\Omega \frac{\delta}{\delta b_i}\left(\frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\right) d\Omega \\ &= \int_\Omega \left[-\frac{1}{T^2}\frac{\partial T}{\partial b_i}\tau_{km}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\frac{\partial \tau_{km}}{\partial b_i}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\tau_{km}\frac{\partial}{\partial b_i}\left(\frac{\partial u_k}{\partial x_m}\right)\right] d\Omega \\ &+ \int_\Omega \left(-\frac{1}{T^2}\frac{\partial T}{\partial x_l}\tau_{km}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\frac{\partial \tau_{km}}{\partial x_l}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\tau_{km}\frac{\partial^2 u_k}{\partial x_m \partial x_l}\right)\frac{\delta x_l}{\delta b_i} d\Omega\end{aligned} \tag{96}$$

and, based on (89),

$$\begin{aligned}&\int_\Omega \frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta(d\Omega)}{\delta b_i} \\ &= -\int_\Omega \left(-\frac{1}{T^2}\frac{\partial T}{\partial x_l}\tau_{km}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\frac{\partial \tau_{km}}{\partial x_l}\frac{\partial u_k}{\partial x_m} + \frac{1}{T}\tau_{km}\frac{\partial^2 u_k}{\partial x_m \partial x_l}\right)\frac{\delta x_l}{\delta b_i} d\Omega \\ &+ \int_S \frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta x_l}{\delta b_i} n_l dS.\end{aligned} \tag{97}$$

Through integration by parts and the Gauss divergence theorem applied to the second and third terms of the first integral on the r.h.s. of (96) and by taking into account the no-slip condition, (88) can be written as

$$\begin{aligned}\frac{\delta F}{\delta b_i} =& -\int_{\Omega}\frac{1}{T^2}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\partial T}{\partial b_i}d\Omega\\ &-2\int_{\Omega}\frac{\partial}{\partial x_m}\left(\frac{1}{T}\tau_{km}\right)\frac{\partial u_k}{\partial b_i}d\Omega\\ &-2\int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_m}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS\\ &+\int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta x_l}{\delta b_i}n_l dS\end{aligned}\tag{98}$$

Equations (98), (52) and (77) lead to the final expression for $\frac{\delta F}{\delta b_i}$, namely

$$\begin{aligned}\frac{\delta F}{\delta b_i} =& \underbrace{-\int_{\Omega}\frac{1}{T^2}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\partial T}{\partial b_i}d\Omega}_{FAE}\\ &\underbrace{-2\int_{\Omega}\frac{\partial}{\partial x_m}\left(\frac{1}{T}\tau_{km}\right)\frac{\partial u_k}{\partial b_i}d\Omega}_{FAE}\\ &-2\int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_m}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS\\ &+\int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta x_l}{\delta b_i}n_l dS\\ &\underbrace{-\int_{\Omega}A_{nmk}\frac{\partial \Psi_n}{\partial x_k}\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE}+\underbrace{\int_{S_{i,o}}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial b_i}n_k dS}_{IOBC}\\ &+\underbrace{\int_{S_w}\Psi_{k+1}n_k\frac{\delta p}{\delta b_i}dS}_{SWCR1}\\ &+\int_{S_w}\left(\Psi_{k+1}p-\Psi_n f^{inv}_{nk}\right)\frac{\delta(n_k dS)}{\delta b_i}\\ &-\int_{S_w}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS-\underbrace{\int_{\Omega}\frac{\partial V_k}{\partial U_m}K_k\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE}\\ &+\underbrace{\int_{S_{i,o}}\frac{\delta u_k}{\delta b_i}\left[\mu\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\right.\right.}_{IOBC...}\\ &\underbrace{\left.+\frac{\partial \Psi_{k+1}}{\partial x_m}+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\right)}_{...IOBC...}\\ &\underbrace{\left.+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\right]n_m dS}_{...IOBC}\\ &-\int_{S_w}\left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\left[\mu\left(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\right.\right.\\ &\left.+\frac{\partial \Psi_{k+1}}{\partial x_m}+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\right)\\ &\left.+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\right]n_m dS\\ &+\underbrace{\int_{S_w}\frac{\delta T}{\delta b_i}k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS}_{SWCR2}+\underbrace{\int_{S_w}\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_k n_m)}{\delta b_i}dS}_{SWCR1}\\ &-\int_{S_w}\left(\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS\\ &-\int_{S_w}\frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS\\ &+\int_{S_w}\Psi_n\frac{\partial f^{vis}_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS\\ &+\underbrace{\int_{S_w}\Psi_\Lambda\frac{\delta(q_m n_m)}{\delta b_i}dS}_{SWCR2}+\int_{S_w}\Psi_\Lambda q_m\frac{\delta n_m}{\delta b_i}dS\\ &+\underbrace{\int_{S_w}\frac{\delta T}{\delta b_i}k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS}_{IOBC}\\ &+\underbrace{\int_{S_{i,o}}\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_k n_m)}{\delta b_i}dS}_{IOBC}\\ &\underbrace{-\int_{S_{i,o}}\Psi_\Lambda\frac{\delta(q_m n_m)}{\delta b_i}dS.}_{IOBC}\end{aligned}\tag{99}$$

Integrals marked with FAE can be eliminated by satisfying the field adjoint equations, namely

$$\frac{\partial \Psi_m}{\partial t}-A_{nmk}\frac{\partial \Psi_n}{\partial x_k}-\frac{\partial V_k}{\partial U_m}K_k-\frac{\partial V_k}{\partial U_m}L_k=0,\tag{100}$$

where

$$\begin{aligned}L_1&=-\frac{1}{\rho T}\tau_{ij}\frac{\partial u_i}{\partial x_j},\\ L_{i+1}&=2\frac{\partial}{\partial x_j}\left(\frac{\mu}{T}\tau_{ij}\right),\quad i=1,\dots,\Lambda-2,\\ L_\Lambda&=\frac{1}{pT}\tau_{ij}\frac{\partial u_i}{\partial x_j}.\end{aligned}$$

The elimination of terms marked with $SWCR1$ leads to homogeneous Dirichlet conditions for the adjoint velocity variables over the solid walls

$$\Psi_{i+1} = 0, \quad i = 1, \ldots, \Lambda - 2 \tag{101}$$

which are equivalent to the no-slip condition for the velocities ($u_i = 0$). The boundary conditions for Ψ_Λ at the wall and all Ψ_i's at the inlet and outlet boundaries result from the elimination of terms marked with $SWCR2$ and $IOBC$, respectively, and are the same to those used for inverse design problems at viscous flows.

The remaining terms express the sensitivity derivatives of F or F_{aug} with respect to the design variables,

$$\begin{aligned}
\frac{\delta F_{aug}}{\delta b_i} = & -2\int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_m}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS \\
& + \int_{S_w}\frac{1}{T}\tau_{km}\frac{\partial u_k}{\partial x_m}\frac{\delta x_l}{\delta b_i}n_l dS \\
& + \int_{S_w}\left(\Psi_{k+1}p - \Psi_n f^{inv}_{nk}\right)\frac{\delta\left(n_k dS\right)}{\delta b_i} \\
& - \int_{S_w}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS \\
& - \int_{S_w}\left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\Bigg[\mu\Bigg(\frac{\partial\Psi_{m+1}}{\partial x_k} + u_m\frac{\partial\Psi_\Lambda}{\partial x_k} \\
& + \frac{\partial\Psi_{k+1}}{\partial x_m} + u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\Bigg) \\
& + \lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l} + u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\Bigg]n_m dS \\
& - \int_{S_w}\left(\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)k\frac{\partial\Psi_\Lambda}{\partial x_k}n_k dS \\
& - \int_{S_w}\frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS \\
& + \int_{S_w}\Psi_n\frac{\partial f^{vis}_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS + \int_{S_w}\Psi_\Lambda q_m\frac{\delta n_m}{\delta b_i}dS.
\end{aligned} \tag{102}$$

Once more, the gradient expression is free of field integrals, leading to a more accurate and less computationally demanding algorithm, compared to adjoint formulations that maintain the field integrals.

3.4.2 *Minimization of Total Pressure Losses in Ducts/Cascades*

Aiming at the minimization of viscous losses, another suitable functional for shape optimization problems in internal aerodynamics is the difference in total pressure between the inlet and outlet of the flow domain, [60]. In [61, 62], comparisons of the two functionals that quantify viscous losses (entropy generation and total pressure losses), using the discrete and continuous approaches, have been presented by the present authors.

The total pressure losses functional is defined as

$$F = \int_{S_i} p_t dS - \int_{S_o} p_t dS. \tag{103}$$

This form of F is not "directly" associated with any change in the body shape, since the direct sensitivity of the inlet and outlet boundary shapes (sensitivity of their nodal coordinates) with respect to the design variables is zero. However, any change in the design variables and, thus, the aerodynamic shape results to changes in the flow variables (total pressure) at the outlet which, certainly, affect F.

The sensitivity of F, (103), depends only on the sensitivity of the outlet total pressure, since the inlet p_t value is fixed. Consequently, (103) becomes

$$\frac{\delta F}{\delta b_i} = -\int_{S_o}\frac{\delta p_t}{\delta b_i}dS. \tag{104}$$

Using (104), (52) and (77), the sensitivity of the new augmented function can be expressed as

$$\begin{aligned}
\frac{\delta F}{\delta b_i} = & -\underbrace{\int_{S_o}\frac{\delta p_t}{\delta b_i}dS}_{IOBC} - \underbrace{\int_\Omega A_{nmk}\frac{\partial\Psi_n}{\partial x_k}\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE} \\
& + \underbrace{\int_{S_{i,o}}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial b_i}n_k dS}_{IOBC} + \underbrace{\int_{S_w}\Psi_{k+1}n_k\frac{\delta p}{\delta b_i}dS}_{SWCR1} \\
& + \int_{S_w}\left(\Psi_{k+1}p - \Psi_n f^{inv}_{nk}\right)\frac{\delta\left(n_k dS\right)}{\delta b_i} \\
& - \int_{S_w}\Psi_n\frac{\partial f^{inv}_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS \\
& - \underbrace{\int_\Omega\frac{\partial V_k}{\partial U_m}K_k\frac{\partial U_m}{\partial b_i}d\Omega}_{FAE} \\
& + \underbrace{\int_{S_{i,o}}\frac{\delta u_k}{\delta b_i}\Bigg[\mu\Bigg(\frac{\partial\Psi_{m+1}}{\partial x_k} + u_m\frac{\partial\Psi_\Lambda}{\partial x_k}}_{IOBC\ldots} \\
& + \underbrace{\frac{\partial\Psi_{k+1}}{\partial x_m} + u_k\frac{\partial\Psi_\Lambda}{\partial x_m}\Bigg)}_{\ldots IOBC\ldots} \\
& + \underbrace{\lambda\delta_{km}\left(\frac{\partial\Psi_{l+1}}{\partial x_l} + u_l\frac{\partial\Psi_\Lambda}{\partial x_l}\right)\Bigg]n_m dS}_{\ldots IOBC} \\
& - \int_{S_w}\left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\Bigg[\mu\Bigg(\frac{\partial\Psi_{m+1}}{\partial x_k} + u_m\frac{\partial\Psi_\Lambda}{\partial x_k}
\end{aligned}$$

$$
\begin{aligned}
&+\frac{\partial \Psi_{k+1}}{\partial x_m}+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\Bigg)\\
&+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\Bigg]n_m dS\\
&+\underbrace{\int_{S_w}\frac{\delta T}{\delta b_i}k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS}_{SWCR2}+\underbrace{\int_{S_w}\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_k n_m)}{\delta b_i}dS}_{SWCR1}\\
&-\int_{S_w}\left(\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS\\
&-\int_{S_w}\frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS\\
&+\int_{S_w}\Psi_n\frac{\partial f_{nk}^{vis}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS\\
&+\underbrace{\int_{S_w}\Psi_\Lambda\frac{\delta(q_m n_m)}{\delta b_i}dS}_{SWCR2}+\int_{S_w}\Psi_\Lambda q_m\frac{\delta n_m}{\delta b_i}dS\\
&+\underbrace{\int_{S_{i,o}}\frac{\delta T}{\delta b_i}k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS}_{IOBC}\\
&+\underbrace{\int_{S_{i,o}}\frac{\Psi_{k+1}}{n_k}\frac{\delta(\tau_{km}n_k n_m)}{\delta b_i}dS}_{IOBC}\\
&-\underbrace{\int_{S_{i,o}}\Psi_\Lambda\frac{\delta(q_m n_m)}{\delta b_i}dS.}_{IOBC}
\end{aligned}
\tag{105}
$$

It can be shown that the field adjoint equations and all boundary conditions, apart from those imposed at the inlet and outlet, are identical to those derived in the case of optimization targeting at minimum entropy generation. At the inlet and outlet, the second $IOBC$ integral is transformed to

$$
\begin{aligned}
\int_{S_{i,o}}\Psi_n\frac{\partial f_{nk}^{inv}}{\partial b_i}n_k dS&=\int_{S_{i,o}}\Psi_n A_{nmk}\frac{\partial U_m}{\partial b_i}n_k dS\\
&=\int_{S_{i,o}}\Psi_n P_{nq}\Lambda_{qr}\bar{P}_{rm}\frac{\partial U_m}{\partial b_i}dS\\
&=\int_{S_{i,o}}\Psi_n P_{nq}\Lambda_{qr}\frac{\partial W_r}{\partial b_i}dS,
\end{aligned}
\tag{106}
$$

where P_{nq} and $\bar{P}_{rm}$ are formed by the right and left eigenvectors of $A_{nmk}n_k$, respectively ($P_{nq}\bar{P}_{qm}=I_{nm}$) and Λ_{qr} is the diagonal matrix with the eigenvalues of $A_{nmk}n_k$ (with $A_{nmk}n_k=P_{nq}\Lambda_{qr}\bar{P}_{rm}$). W_r denotes the characteristic flow variables with $\delta W_r=\bar{P}_{rm}\delta U_m$.

The first $IOBC$ integral along the outlet can be written as

$$
\int_{S_o}\frac{\delta p_t}{\delta b_i}dS=\int_{S_o}\frac{\partial p_t}{\partial V_n}L_{nr}\frac{\partial W_r}{\partial b_i}dS,
\tag{107}
$$

where L_{nr} is the matrix formed by the right eigenvectors of the Jacobian matrix of $\frac{\partial f_{nk}^{inv}}{\partial V_n}n_k$. In 3D flows,

$$
\begin{aligned}
&\frac{\partial p_t}{\partial V_1}=g\frac{u_k^2}{2},\qquad \frac{\partial p_t}{\partial V_i}=g\rho u_k,\quad i=1,\ldots,\Lambda-2,\\
&\frac{\partial p_t}{\partial V_\Lambda}=\left(1-\frac{R\rho u_k^2}{2(\gamma-1)c_p p}\right)g,
\end{aligned}
\tag{108}
$$

where $g=\frac{\gamma R}{(\gamma-1)c_p}(1+\frac{u_k^2}{2c_p}\frac{Rp}{\rho})^{\frac{1}{\gamma-1}}$.

Neglecting any other $IOBC$ integral in (105), the integral that defines the boundary condition at the outlet is

$$
\int_{S_o}\left(\Psi_n P_{nq}\Lambda_{qr}-\frac{\partial p_t}{\partial V_n}L_{nr}\right)\frac{\delta W_r}{\delta b_i}dS=0.
\tag{109}
$$

If Q characteristic flow variables are defined at the inlet to the flow domain, their sensitivities are zero and Q adjoint variables are extrapolated from the interior of the flow domain. The remaining $\Lambda-Q$ adjoint variables, i.e. as many as the characteristic flow variables that are extrapolated from the interior of Ω, are computed by solving

$$
\Psi_n P_{nq}\Lambda_{qr}-\frac{\partial p_t}{\partial V_n}L_{nr}=0.
\tag{110}
$$

Along the inlet, where $\frac{\partial p_t}{\partial b_i}=0$, the corresponding boundary condition for Ψ_n is derived from

$$
\Psi_n P_{nq}\Lambda_{qr}\frac{\delta W_r}{\delta b_i}=0.
\tag{111}
$$

The final expression for the objective function gradient is given by

$$
\begin{aligned}
\frac{\delta F_{aug}}{\delta b_i}=&-\int_{S_w}\left(\frac{\partial u_k}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)\Bigg[\mu\Bigg(\frac{\partial \Psi_{m+1}}{\partial x_k}+u_m\frac{\partial \Psi_\Lambda}{\partial x_k}\\
&+\frac{\partial \Psi_{k+1}}{\partial x_m}+u_k\frac{\partial \Psi_\Lambda}{\partial x_m}\Bigg)\\
&+\lambda\delta_{km}\left(\frac{\partial \Psi_{l+1}}{\partial x_l}+u_l\frac{\partial \Psi_\Lambda}{\partial x_l}\right)\Bigg]n_m dS\\
&-\int_{S_w}\left(\frac{\partial T}{\partial x_l}\frac{\delta x_l}{\delta b_i}\right)k\frac{\partial \Psi_\Lambda}{\partial x_k}n_k dS\\
&-\int_{S_w}\frac{\Psi_{k+1}}{n_k}\tau_{km}\frac{\delta(n_k n_m)}{\delta b_i}dS\\
&+\int_{S_w}\Psi_n\frac{\partial f_{nk}^{vis}}{\partial x_l}\frac{\delta x_l}{\delta b_i}n_k dS+\int_{S_w}\Psi_\Lambda q_m\frac{\delta n_m}{\delta b_i}dS.
\end{aligned}
\tag{112}
$$

3.4.3 Viscous Losses Minimization—Applications

The application of the continuous adjoint approach to the redesign of a turbine cascade (the same that was previously used to demonstrate the inverse design application) so as to give minimum total pressure losses is presented herein. Constraints related to the minimum allowed airfoil thickness are also imposed. The thickness of the airfoil at several points along the chord was not allowed to reduce more than 10%. To do so, the augmented Lagrange multipliers method, [8], was used with the constraint defined as the airfoil thickness at several points along the cascade. The Reynolds number is equal to 5×10^5 and the Spalart–Allmaras model, [47], is employed to account for turbulence. An H-type structured grid is used with $201 \times 101 = 20301$ nodes. The computational grid and the total pressure iso-surfaces for the initial and optimal airfoil are shown in Figs. 7 and 8.

The reduction in total pressure losses achieved during the optimization is shown in Fig. 9. The loss coefficient is reduced by almost 6.7% within the first 180 cycles; although the corresponding optimization cost appears to be high, it should be stressed that a "simple" (i.e. non-optimized with respect to the stepsize value used) steepest descent algorithm was used. A great part of this reduction is, in fact, achieved within the first 40 cycles but during these cycles, the constraint on the minimum allowed airfoil thickness was often violated compared to the reference airfoil. This constraint violation during the first cycles is shown, also, in Fig. 9. After the 40th cycle, the constraint was satisfied, which means that the thickness in all chosen positions is more than 90% of the initial thickness at the same positions.

The change in the values of the objective function gradient is shown in Fig. 10. It can be seen that the sensitivities with respect to the suction side control points (first half values) are greater than the sensitivities with respect to the pressure side ones. So, at first, changes in the suction side control points is greater than that in pressure side ones. Also, the positive values of the suction side sensitivities drives the steepest descent algorithm towards thinner blades. After the very first cycles, the constraint starts being violated and the gradient of the constraint function is added to that of the objective function causing the movement of both the suction and pressure side control points towards a thicker airfoil.

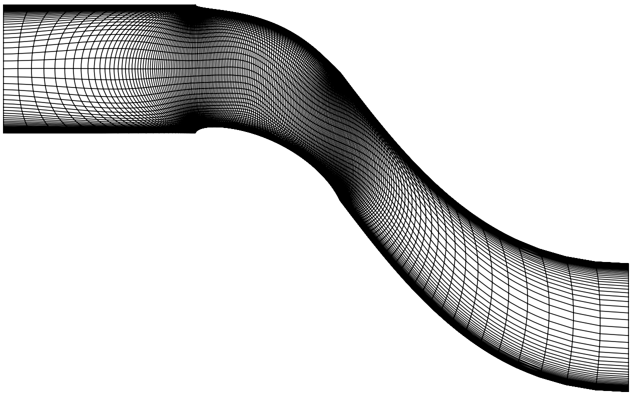


Fig. 7 Minimization of total pressure losses in a 2D turbine cascade. H-type computational grid

The whole procedure leads to the progressive movement of the control points along both sides towards the (fixed) chord of the airfoil, reducing thus the camber angle, as shown in Fig. 11. It should become clear that, in Fig. 10, the gradient of F, before being penalized by the (gradient of the) constraint function, is plotted. This explains why the gradient is not zeroed at the optimum.

The pressure coefficient for the initial and optimal configurations are shown in Fig. 12.

The adjoint formulations for minimum total pressure losses and for minimum entropy generation are compared on the following case which is concerned with the optimization of a compressor cascade airfoil. The Reynolds number is equal to 10^6 and an H-type structured grid with $161 \times 91 = 14651$ nodes is used. The steepest descent algorithm is employed and the airfoil thickness is not allowed to reduce more than the 90% of its initial thickness, measured at several chordwise positions. The computational grid as well as the total pressure isolines for the initial and optimal airfoil (using the total pressure losses functional) are shown in Figs. 13 and 14.

The progressive reduction in total pressure losses, (104), and entropy generation, (87), are shown in Fig. 15. Results from two studies are shown. In the first study, the minimization of total pressure losses was used as target but, at the end of each cycle, the value of the function expressing entropy generation, (87), was also computed and plotted. In the second, the optimization was based on the reduction of entropy generation and the difference in total pressure between the inlet and outlet was simultaneously recorded, by just computing and plotting the value of F, (103).

Similar behaviours can be observed in all four curves, demonstrating, thus, the equivalence of the two functionals and the corresponding adjoint formulations. Note that the two functionals are defined over different parts of the flow domain and none of them is defined along the solid walls which are controlled by the design variables.

Some oscillations in the convergence of both functionals are due to the violation of constraints on the minimum allowed airfoil thickness. This violation is shown in Fig. 16. Using either functional, the constraint is violated during the early cycles of the optimization loop but, later on, the violation decreases and, finally, disappears.

In Fig. 17, the evolution (during the optimization) of the sensitivities of the total pressure losses functional, left, and the entropy generation one, right, is shown. It can be seen that the gradient values are higher at the suction side control points. As a consequence, along with the imposition of the minimum allowed thickness constraint, the optimization al-

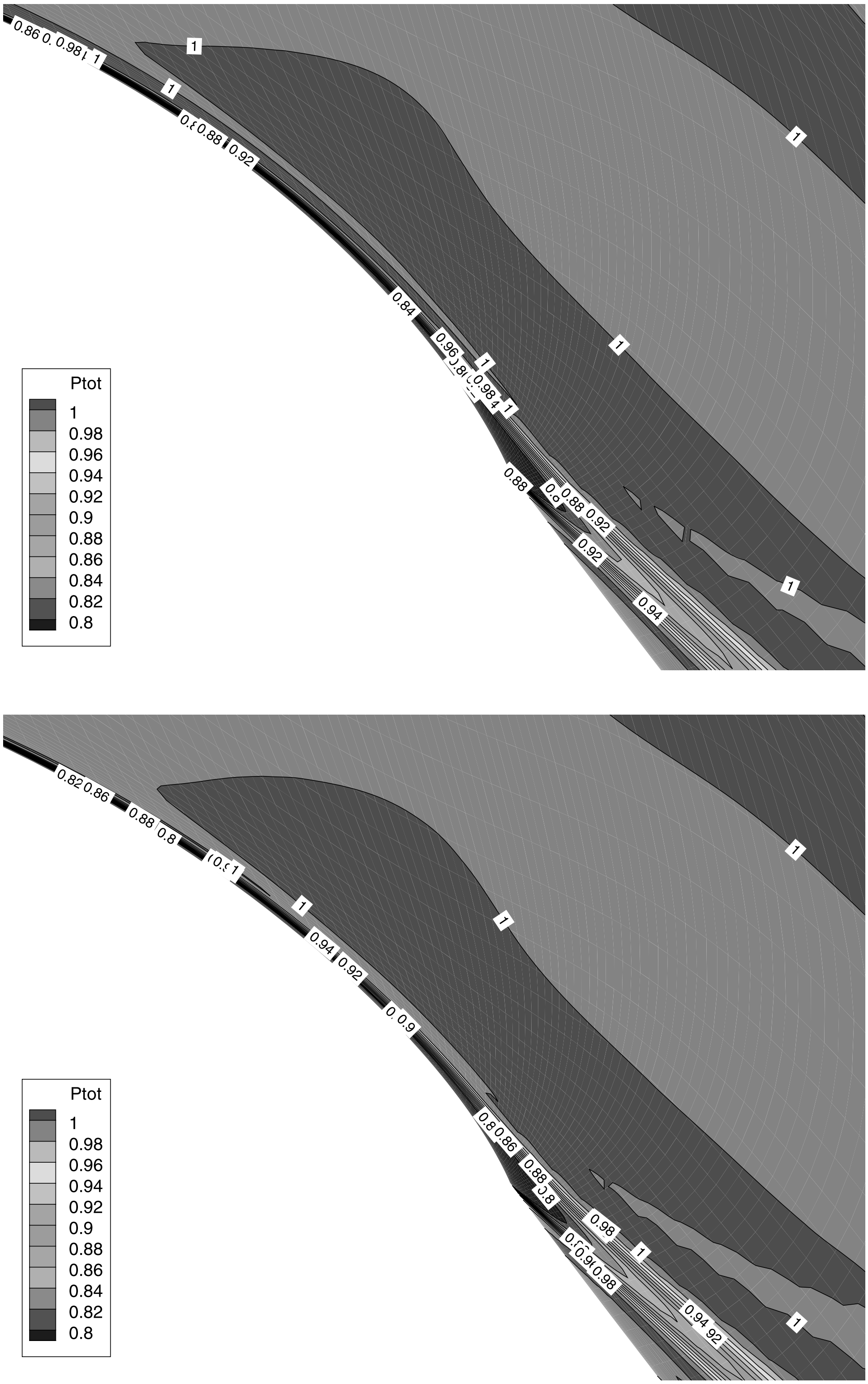


Fig. 8 Minimization of total pressure losses in a 2D turbine cascade. Non-dimensional total pressure distribution at the initial and optimal turbine cascades. Blow-up view of the airfoil suction side

gorithm moves both blade airfoil sides towards the chord of the airfoil.

This tendency is shown in Fig. 18 where the initial and optimal sets of control points and the corresponding airfoil contours, using either of the two functional, are shown. Note that the optimal geometries are much alike.

Finally, the pressure and friction coefficients for the initial and optimal configurations are shown in Fig. 19. In both

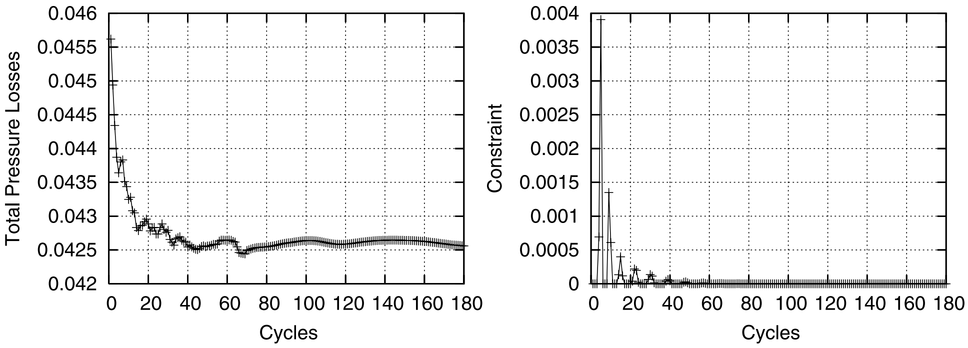


Fig. 9 Minimization of total pressure losses in a 2D turbine cascade. *Left*: reduction in the objective function value. *Right*: sum of violated geometrical constraints on minimum allowed airfoil thickness at several chordwise positions. A zero constraint value corresponds to fully satisfied (geometrical) constraints

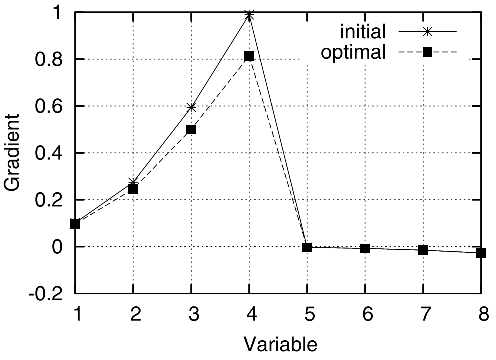


Fig. 10 Minimization of total pressure losses in a 2D turbine cascade. The objective function gradient components for the initial and optimal airfoil shapes are plotted. The gradient of the constraint function is not included

figures, it is obvious that the flow separation region has been reduced with either objective function.

3.5 Other Objective Functions

The adjoint formulation for first order sensitivities presented in the previous section can be used with other objective functions as well. For instance, in external aerodynamics, the optimization is often carried out based on the lift and/or drag coefficients which, for a 2D aerodynamic body, are defined by

$$c_d = \int_{S_w} p\left(n_1 \cos a_\infty - n_2 \sin a_\infty\right) dS,$$
$$c_l = \int_{S_w} p\left(n_1 \sin a_\infty + n_2 \cos a_\infty\right) dS, \tag{113}$$

where (n_1, n_2) is the normal to the wall unit vector and α_∞ is the infinite flow angle.

Several studies in which an adjoint formulation is developed for the computation of the sensitivities of drag and/or lift coefficients with respect to the design variables parameterizing an aerodynamic shape can be found in the literature, [63, 64]. Quite often, the functional to be minimized is a combination of the two coefficients. Such a generalized functional can be defined as

$$F = \frac{w_1}{2}\left(c_d - c_{d,tar}\right)^2 + \frac{w_2}{2}\left(c_l - c_{l,tar}\right)^2, \tag{114}$$

where w_1 and w_2 are user defined weight functions and $c_{d,tar}$ and $c_{l,tar}$ are the target drag and lift coefficients, respectively. Equation (114) also governs the minimization of drag coefficient constrained by a desired c_l value, by just setting $c_{d,tar} = 0$.

A continuous adjoint formulation can be developed for (114), based on what exposed thus far. In the interest of space, the detailed formulation is omitted. It is easy to show,

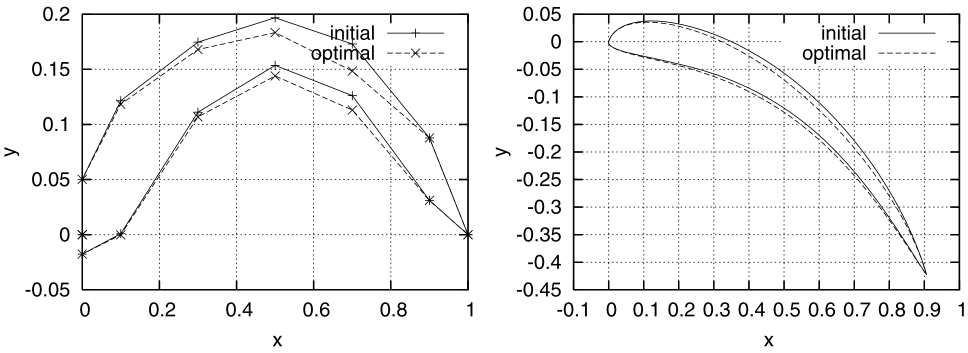


Fig. 11 Minimization of total pressure losses in a 2D turbine cascade. Initial and optimal set of control points (at zero stagger) and initial and optimal turbine airfoil contour (at the desired stagger angle). Not in scale

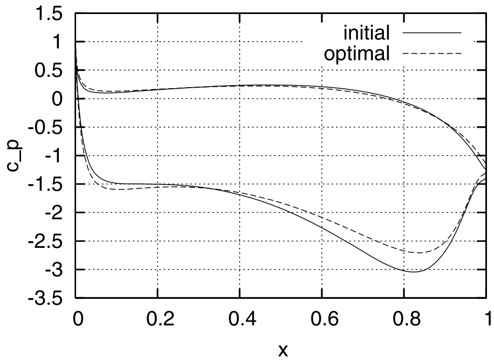


Fig. 12 Minimization of total pressure losses in a 2D turbine cascade. Pressure coefficient for the initial and optimal turbine cascade airfoil

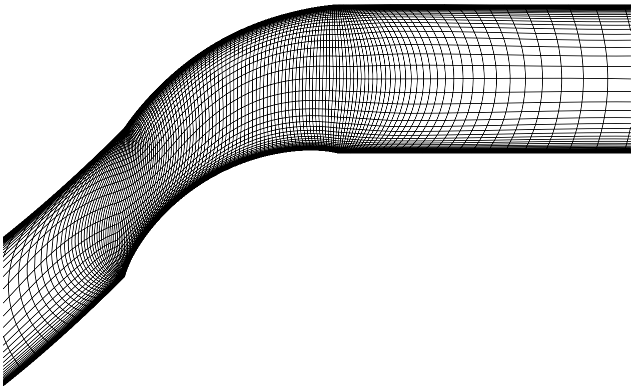


Fig. 13 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. H-type computational grid

however, that the corresponding compatibility relation takes the form

$$n_1 \cos a_\infty - n_2 \sin a_\infty + \Psi_{k+1} n_k = 0 \tag{115}$$

by maintaining the same inlet/outlet conditions for the adjoint variables as in inverse design problems. Considering inviscid flows, the sensitivity derivatives of F, (114), are expressed as

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_i} &= \frac{1}{2} \int_{S_w} p \left[\frac{\delta (n_1 dS)}{\delta b_i} \cos a_\infty - \frac{\delta (n_2 dS)}{\delta b_i} \sin a_\infty \right] \\ &\quad + \int_{S_w} \left(\Psi_{k+1} p - \Psi_n f^{inv}_{nk} \right) \frac{\delta (n_k dS)}{\delta b_i} \\ &\quad - \int_{S_w} \Psi_n \frac{\partial f^{inv}_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_i} n_k dS \end{aligned} \tag{116}$$

which is also free of field integrals. In a similar manner, the planform design of a wing is also possible using the objective function defined as

$$F = w_1 c_d + w_2 c_w, \tag{117}$$

where c_w is the normalized wing structure weight, [65–68].

3.6 Adjoint Approaches for Transonic and Supersonic Flows

In shocked flows, a discontinuity in flow variables appears at the shock. However, as shown in [69, 70], the adjoint variables are continuous at the shock position. Thus, an additional condition for the adjoint variables at the shock is not required.

The reduction and elimination of the shock produced by transonic wings through an adjoint formulation has been presented in [71], where the functional is a weighted sum of the lift and drag coefficients. A shock exists in the initial configuration and the reduction of the drag coefficient leads to the reduction of the shock, giving rise to shock-free flow over the optimal wing.

On the other hand, in supersonic flows, relevant optimization efforts are mostly concerned with sonic boom reduction. In such a case, the objective function is defined over a different part of the flow domain boundary than the one controlled by the design variables. From this viewpoint, this case presents certain similarities with the already presented problem of minimizing the total pressure losses. So, a similar technique can be devised so as to profit of the advantages of the proposed method.

For the sonic boom reduction, the target is to minimize a functional which measures the sonic boom impact such as overpressures, impulses, etc. [72–76]. The functional can be defined either in a near field boundary, [73, 74], or at the ground level. The flow equations are solved over the domain defined between the aircraft and the near field boundary while the flow analysis at the domain between the near field boundary and the ground plane is based on principles of geometrical acoustics and nonlinear wave propagation. The problem of finding the optimal shape that produces desired pressure signature at the ground is addressed in [75].

4 Other Algorithms Based on the Adjoint Approach

4.1 One-Shot Algorithms

Standard adjoint-based optimization algorithms rely on the sequential solution of the flow and adjoint equations followed by an equation for updating the design variables. This procedure may become costly enough if the flow and adjoint equations converge to machine accuracy (although this is not necessary especially during the first optimization cycles).

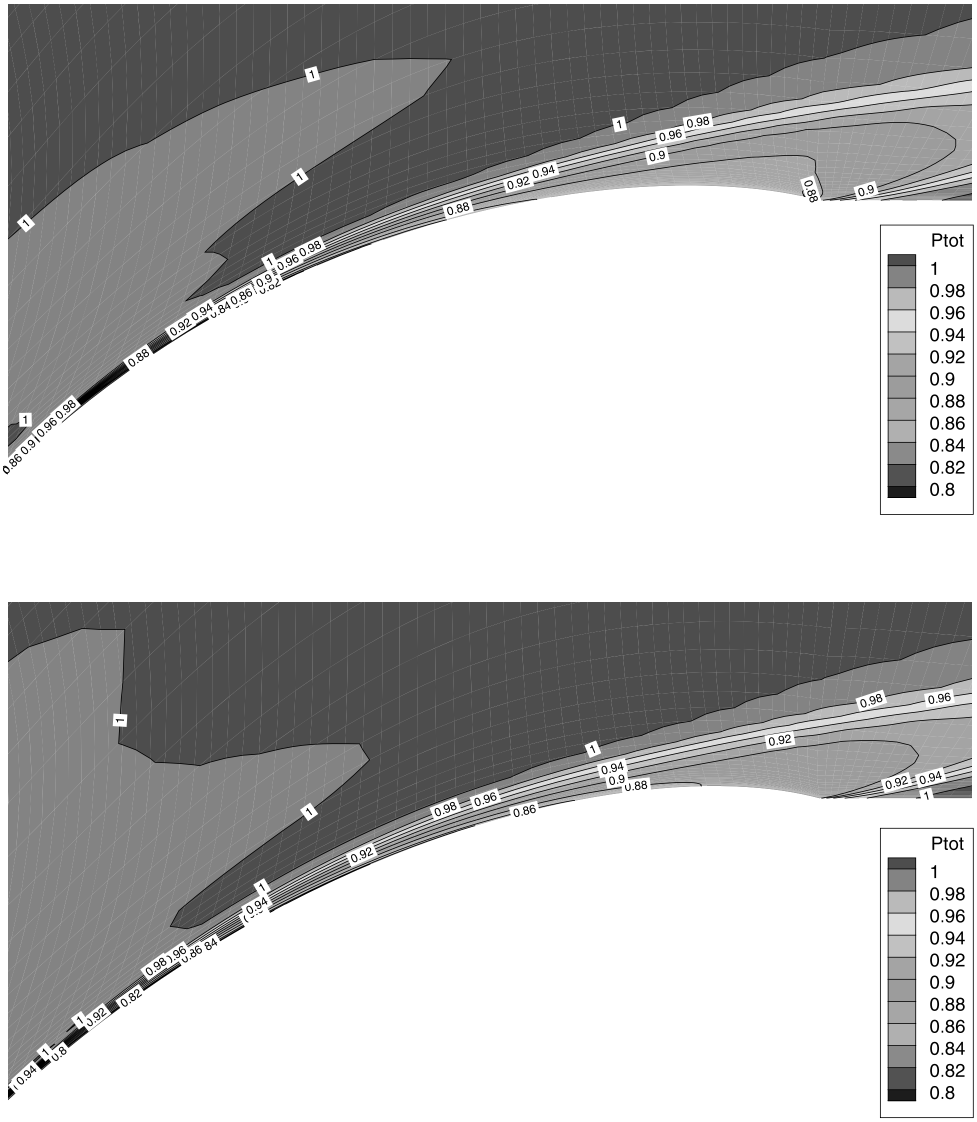


Fig. 14 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. Non-dimensional total pressure distribution at the initial and optimal compressor cascade. Blow-up view of the airfoil suction side

As a faster alternative to the sequential solution, the three sets of equations can be solved at once, giving rise to the so-called one-shot algorithms. The one-shot algorithm was first presented in [77, 78], where a multigrid strategy was employed for the solution of the flow and adjoint equations, while the design process was embedded within the multigrid cycles. In any grid, only the design variables, which produce high-frequency perturbations in the flow field, are changed.

A different one-shot approach can be found in [79–81]. It is based on the simultaneous pseudo-timestepping approach. If

$$\begin{aligned} R_U &= 0, \\ R_\Psi &= 0, \\ R_b &= 0 \end{aligned} \tag{118}$$

is the system of the state, adjoint and sensitivity equations, the pseudo-time embedded evolution equations yield

$$\begin{aligned} \frac{dU}{dt} + R_U &= 0, \\ \frac{d\Psi}{dt} + R_\Psi &= 0, \\ \frac{db}{dt} + R_b &= 0. \end{aligned} \tag{119}$$

A reduced SQP algorithm is used in [79–81], so, instead of solving the stiff system of (119), the preconditioned system

$$\begin{bmatrix} \frac{dU}{dt} \\ \frac{db}{dt} \\ \frac{d\Psi}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & A^* \\ 0 & B & C^* \\ A & C & 0 \end{bmatrix}^{-1} \begin{bmatrix} -R_\Psi \\ -R_b \\ -R_U \end{bmatrix} \tag{120}$$

is solved, where A is the Jacobian matrix of the flow equations, A^* is the adjoint Jacobian matrix, B is an estimation

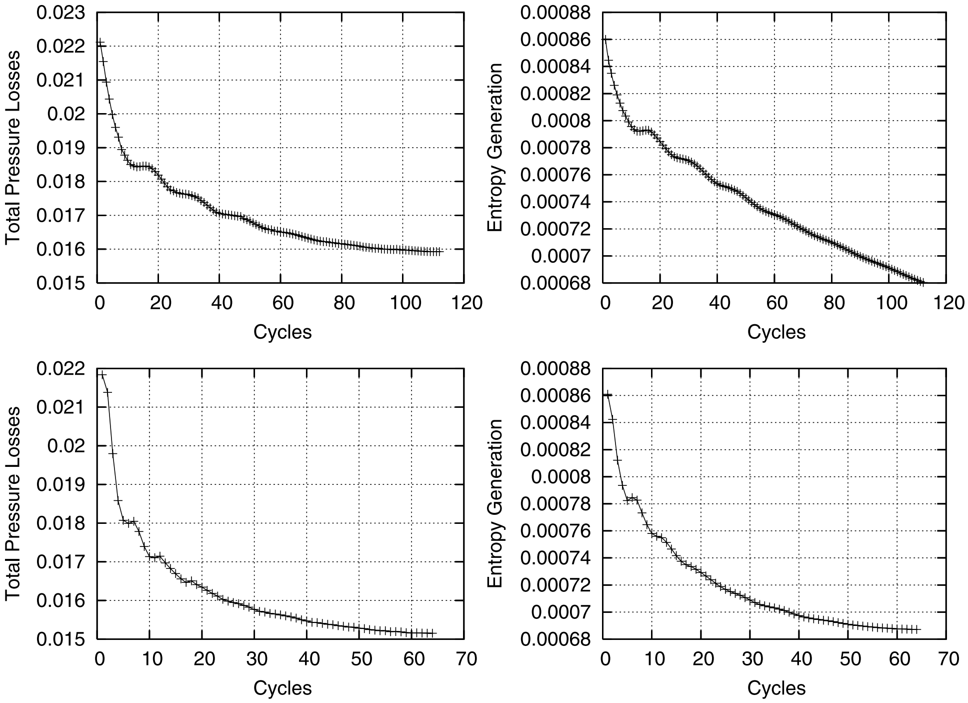


Fig. 15 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. *Top*: first computation based on the minimization of total pressure losses. *Bottom*: second computation based on the minimization of entropy generation. The reduction of the objective function values are, thus, shown in the top-left and bottom-right figures

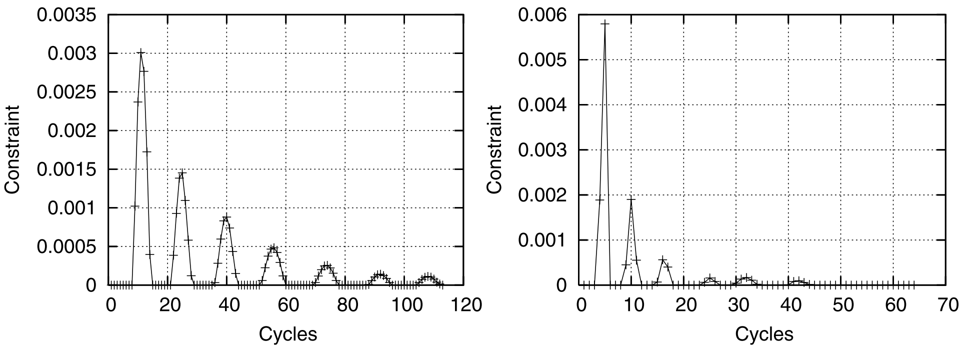


Fig. 16 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. Sum of violated geometrical constraints on minimum allowed airfoil thickness at several chordwise positions. *Left*: first computation aiming at total pressure losses minimization. *Right*: second computation aiming at entropy generation minimization

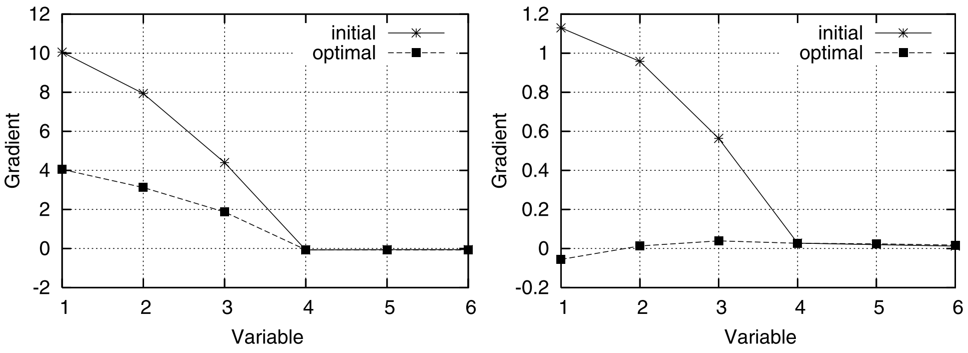


Fig. 17 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. Gradient of the objective function with respect to the design variables, without taking into account the gradient of the constraint function for the initial and optimal airfoils. *Left*: first computation aiming at total pressure losses minimization. *Right*: second computation aiming at entropy generation minimization

of the Hessian matrix and C and C^* are the derivatives of the flow equations with respect to the design variables.

Other variations of the standard one-shot approach can be found in the literature; for instance, in [82] divided dif-

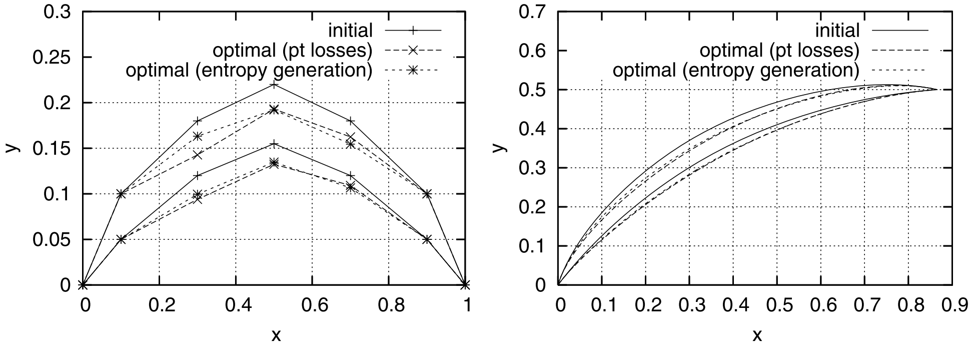


Fig. 18 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. Initial and optimal set of control points (at zero stagger) and initial and optimal turbine airfoil contour (at the desired stagger angle), using the total pressure losses and the entropy generation functionals. Not in scale

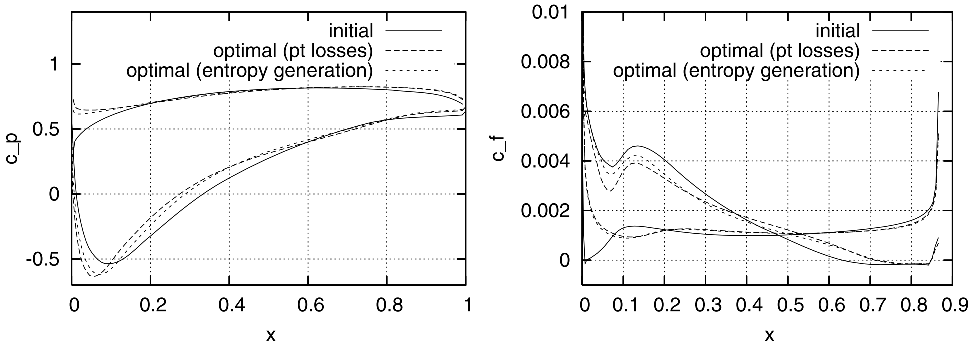


Fig. 19 Minimization of total pressure losses or entropy generation in a 2D compressor cascade. Pressure coefficient, *left*, and friction coefficient, *right*, of the initial and optimal turbine cascade airfoils, using the total pressure losses and the entropy generation functionals

ferences are used instead of the adjoint approach and in [83, 84] a progressive algorithm is used, where the design variables are updated after a partially converged flow solution followed by a partially converged adjoint solution.

The gain from the one-shot approach is the reduction of the total computational cost for the convergence of the optimization algorithm to almost three to five times the cost of solving the flow equations.

It is important to stress that the one-shot approach may be combined with the adjoint formulation presented in this paper. The gain would then be cumulative, since the presented adjoint formulation increases the efficiency of computing the gradient while the one-shot approach accelerates the optimization procedure itself.

4.2 Other Expressions for the Exact/Inexact Gradient

The continuous adjoint approach presented in this paper is not the only one leading to objective function gradients free of field integrals. To cover other possible formulations, the so-called reduced method, [55] and the surface sensitivity method, [85], are presented and discussed in brief.

In the reduced continuous adjoint formulation on structured grids, [55], all terms depending on the sensitivities of grid metrics are transformed accordingly, leading to an expression which is free of field integrals. This method is presented in [55] and its extension to unstructured grids in [86]. The method was used only for inviscid design problems, while its extension to viscous flows is discussed in [87]. Although it is equivalent to the one proposed by the authors, it is the authors' opinion that our derivation of the final expression is more general and straightforward.

In the metrics-based adjoint approach that was firstly presented by Jameson, [22], the sensitivities are computed by the expression

$$\delta F_{aug} = -\int_{\Omega_\xi} \frac{\partial \Psi_n}{\partial \xi^k} \delta S_{km} f_{nm}^{inv} - \int_{S_\xi} \delta S_{2k} \Psi_{k+1} p d\xi_1 d\xi_3, \tag{121}$$

where let $\xi_2 = const.$ be the parameterized part of the domain and $S_{km} = J\frac{\partial \xi^k}{\partial x_m}$ the Jacobian of the transformation. By means of the equation

$$\frac{\partial(\delta S_{km} f_{nm}^{inv})}{\partial \xi^k} = -\frac{\partial}{\partial \xi^k}\left(C_k \frac{\partial U_n}{\partial x_l}\delta x_l\right) \tag{122}$$

(121) is transformed to the final expression

$$\delta F_{aug} = -\int_{S_\xi} \Psi_n \left(\delta S_{2m} f_{nm}^{inv} + C_2 \frac{\partial U_n}{\partial x_l}\delta x_l\right) d\xi_1 d\xi_3 - \int_{S_\xi} \delta S_{2k} \Psi_{k+1} p d\xi_1 \xi_3, \tag{123}$$

where $C_k = S_{km} A_m$. It is a matter of transformations to prove that (123) is equivalent to the expression proposed by the authors.

The approach proposed in [85] results to the gradient of the objective function with respect to the normal displacements of the grid nodes. The method is extended to internal flows, [88], i.e. for maximizing the efficiency of the flow through 3D complex ducts. It is a matter of postprocessing to find the equivalence of this approach with the one presented herein. The difference in the adjoint formulation for the inverse design using the approach presented in [85, 88] is based on the fact that it computes the sensitivities of the objective function with respect to normal displacements of the grid nodes n instead of the components b_i of the control points that parameterize the shape. So, the first integral of the r.h.s. of (47), with n instead of b_i, becomes

$$\int_{S_w} \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_i} n_k dS = \int_{S_w} \Psi_n \frac{\delta (f_{nk}^{inv} n_k)}{\delta n} dS = \int_{S_w} \Psi_k n_k \frac{\delta p}{\delta n} dS. \tag{124}$$

The second integral on the r.h.s. at the same equation is, simply, written as

$$\int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_l} \frac{\delta x_l}{\delta n} n_k dS = \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_k} dS. \tag{125}$$

So the adjoint equations and boundary conditions are the same as those exposed above, whereas the sensitivity expression is given by

$$\frac{\delta F_{aug}}{\delta n} = \frac{1}{2} \int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta n} + \int_{S_w} \Psi_n \frac{\partial f_{nk}^{inv}}{\partial x_k} dS. \tag{126}$$

There is another similar category of methods, which are named as incomplete gradient methods, [89–95], in which the gradient values are not exact but less expensive to compute. The simplest way to compute an approximate gradient is to omit the second term on the r.h.s. of (11). Thus, the gradient may be computed by the expression

$$\frac{dF_l}{db_i} \cong \frac{\partial F_l}{\partial b_i} \tag{127}$$

and is, certainly, expected to be of poor accuracy. However, the solution of the adjoint equations is not required at all, so the gradient is approximated by only solving the flow equations and substituting the flow variable values to (127). It is proved, [89, 90], using some simple cases, that this approach leads to sensitivity derivatives of at least the right sign. Thus, a steepest descent algorithm driven by this approximate gradient would possibly lead to the optimal solution.

This might be the case of inverse design or other problems in which the functional is collocated at the part of the domain boundary which is controlled by the design variables. However, for functionals defined either over different parts of the domain (such as the expression of F for total pressure losses) or through the entire domain (such as the field integral used for the minimization of entropy generation), (127) will produce zero sensitivities. In these cases, such an incomplete gradient approach cannot be applied.

There is another incomplete gradient approach, which is proved to compute more accurate sensitivities, increasing, however, the CPU cost. This approach is based on the continuous adjoint formulation but it avoids the solution of the adjoint equations. In [91], the augmented function is first written in terms of the normal and tangential projection of the flow equations' residual, taking into account only the residual of the momentum equations, assuming that the main contribution of the adjoint term to the gradient of F at the boundary is controlled by the design variables as follows

$$F_{aug} = F + \int_{S_w} \Psi_{k+1} \left(R_{m+1} n_m n_k + R_{m+1} t_m t_k \right) dS, \tag{128}$$

where $k = 1, \ldots, \Lambda - 2$ and t_k are the components of the tangent to the boundary unit vector. The tangential contribution is omitted, whereas the adjoint variables along the solid walls are computed using the compatibility relation for the inverse design problem, (58), derived from the continuous adjoint formulation. Thus, the functional is computed by the expression

$$F_{aug} = F - \int_{S_w} (p - p_{tar}) R_{m+1} n_m dS. \tag{129}$$

The design variables' perturbation is used to move the boundary, F_{aug} is reevaluated and finite differences are used to compute the desired sensitivities. The extension of this approach to viscous flows is found in [94], where the problem tackled is that of drag minimization with desired lift and smooth pressure gradient. In this paper, the partial discrete adjoint solution is also introduced, which allowed to take into account, in an incomplete manner, all the adjoint contribution to the sensitivities. The extension of this approach to turbulent flow is found in [95].

Similarly, several discrete adjoint algorithms have been presented for handling mesh sensitivities. In this case, the gradient formula is not reduced to a boundary expression and the mesh sensitivities are not eliminated using closed form expressions. In [96], the mesh sensitivities are computed limited to a region near the parameterized body. In [97] the mesh sensitivities are eliminated by solving an additional set of adjoint equations. A more generalized approach for the discrete adjoint approach in unstructured grids, which efficiently handles mesh sensitivities is shown in [98, 99], for 2D and 3D flows.

Concerning mesh sensitivities, the discrete adjoint formulation has no major difficulties for unstructured grids compared to the use of structured grids. Several relevant studies can be found in the literature for 2D or 3D, inviscid or viscous flows, [100–104]

4.3 Adjoint Approaches for Multiobjective, Multipoint or Multidisciplinary Optimization

In multi-objective optimization problems, the adjoint approach can be used to compute the gradient of the objective functions. A conventional way to cope with more than one objectives is by introducing appropriate weights w_l, [105]. Through them, a scalar F is defined as the following weighted sum

$$F = w_l F_l \tag{130}$$

with sensitivities

$$\frac{\delta F}{\delta b_i} = w_l \frac{\delta F_l}{\delta b_i}. \tag{131}$$

To compute elements of the Pareto front of non-dominated solutions in a multi-objective optimization problem, a series of different w_l values can be chosen and a descent algorithm can be used driven by the gradient of (131). The quality of the so-obtained Pareto front is not ensured, as the values have been selected at random. Of course, such a computation is very costly.

In a multi-objective problem with a single discipline (for instance, aerodynamics) there is often a single set of field (flow) equations and boundary conditions. So, with the functional of (130), a single set of adjoint equations has to be solved. However, this is not the case in multipoint optimization where, in most cases, due to the different boundary (flow) conditions at each operating point, the discretized flow equations are different. Thus, an adjoint problem is solved for each functional, the derivatives $\frac{\delta F_l}{\delta b_i}$ are computed and the total derivative $\frac{\delta F}{\delta b_i}$ results from (131). Applications of the adjoint based multipoint optimization can be found in [68, 106].

In multidisciplinary optimization, different functionals and types of design and state variables exist for each discipline. Therefore, as many systems of adjoint equations as the number of disciplines must be solved. For instance, in aerostructural optimization, let $R_a = 0$ and $R_s = 0$ be the fluid and structure state equations, U_a the flow state variables, U_s the structural displacements and b_i the design variables. The gradient of F with respect to b_i is computed by, [67, 107–109],

$$\frac{\delta F}{\delta b_i} = \frac{\partial F}{\partial b_i} + \Psi_a \frac{\partial R_a}{\partial b_i} + \Psi_s \frac{\partial R_s}{\partial b_i}, \tag{132}$$

where, to compute the adjoint variables Ψ_a and Ψ_s, the system of flow and structural adjoint equations

$$\begin{bmatrix} \frac{\partial R_a}{\partial U_a} & \frac{\partial R_a}{\partial U_s} \\ \frac{\partial R_s}{\partial U_a} & \frac{\partial R_s}{\partial U_s} \end{bmatrix} \begin{bmatrix} \Psi_a \\ \Psi_s \end{bmatrix} = \begin{bmatrix} \frac{\partial F}{\partial U_a} \\ \frac{\partial F}{\partial U_s} \end{bmatrix} \tag{133}$$

must be solved. In [107], the lagged-coupled adjoint method is used, in which, for the calculation of the adjoint variables of any discipline, the adjoint variables of the other discipline from the previous iteration are used, as follows

$$\begin{aligned} \frac{\partial R_a}{\partial U_a} \Psi_a^{new} &= \frac{\partial F}{\partial U_a} - \frac{\partial R_s}{\partial U_a} \Psi_s^{old}, \\ \frac{\partial R_s}{\partial U_s} \Psi_s^{new} &= \frac{\partial F}{\partial U_s} - \frac{\partial R_a}{\partial U_s} \Psi_a^{old}. \end{aligned} \tag{134}$$

4.4 Optimization in Unsteady Flows

There are several approaches to deal with optimization problems in unsteady inviscid or viscous flows, [110–112]. The simplest approach is based on the solution of the unsteady flow equations, post-processing to get a time-averaged flow field and a single adjoint computation based on averaged state variables. A more accurate approach is to solve the unsteady flow equations and the steady adjoint equations for each time step within the period. The most accurate approach is the fully unsteady calculation, in which both flow and adjoint unsteady equations are solved. The comparison of the computational cost associated with each method and their efficiencies are discussed in detail in [110], where the three possible approaches are also compared with multipoint optimization. Also, in [111], the optimal design in unsteady flows using the non-linear frequency domain approach, is presented. The discrete harmonic adjoint approach is presented in [113–115]. It is based on a single frequency of unsteadiness and the functional used for the design of turbomachinery blades is the generalized force acting on the blades due to arbitrary incoming time-periodic gusts. The adjoint formulations presented in this paper can be combined with all these unsteady flow optimization methods in a straightforward manner.

4.5 Dealing with Turbulence

With either the discrete or continuous adjoint approach, the computation of accurate sensitivities in turbulent flow is of great importance. The majority of published works is based on the so-called frozen turbulence model. Although the turbulence model is included in the state equations, sensitivities of the eddy viscosity coefficient are neglected during the development of the adjoint equations. No extra Lagrange multipliers are introduced for the turbulence model equations and the sensitivity of turbulent viscosity with respect to design variables is zero, [22, 49, 116]. This, of course, reduces the cost for the computation of sensitivities but it also reduces their accuracy, since the adjoint equations are no more the exact adjoints to the flow equations.

There are just a few studies in the literature that utilize the exact discrete adjoint formulation for turbulent flows. The adjoint equations to the Spalart–Allmaras turbulence model equations are solved together with the mean flow equations in [117–119], while in [69], an equivalent formulation of the turbulent adjoint equations is presented using the k–ϵ turbulence model.

5 Direct, Adjoint and Mixed Approaches in Aerodynamic Shape Optimization for the Computation of Second Order Sensitivities

Since Newton or Newton-like iterative optimization algorithms, [8],

$$b_j^{new} = b_j^{old} + \delta b_j, \tag{135}$$

where

$$\frac{\delta^2 F}{\delta b_i \delta b_j}\delta b_j = -\frac{\delta F}{\delta b_i} \tag{136}$$

are more efficient than any algorithm based on first order sensitivities only, the development of an efficient approach for the computation of (exact) second order sensitivities of the objective function with respect to the design variables is important.

Starting from the direct differentiation or the adjoint variable approach which have been presented for the computation of first order sensitivities, the second order ones can be obtained by re-employing any of the aforementioned approaches. Thus, four different approaches can be devised; these will be referred to as the *direct–direct*, *direct–adjoint*, *adjoint–direct* and *adjoint–adjoint* ones. To make this terminology as clear as possible, the direct–direct algorithm employs the direct differentiation method to compute first order sensitivities and, then, once more the same method to compute second order ones. In contrast, in the direct–adjoint method, second order sensitivities are obtained using the adjoint approach and so forth.

A literature survey shows that there are only a few studies on the computation and use of second order sensitivities in optimization problems. In [120], the computation of the second order sensitivities in aerodynamic optimization using automatic differentiation is presented. The discrete approach for inviscid and viscous turbomachinery flow is presented in [121], while its continuous counterpart for inviscid flows in [122]. Even in disciplines other than aerodynamics, the relevant literature is quite poor. The Hessian matrix computation using all possible discrete approaches for structural optimization has been presented in [123], while continuous approaches for the Hessian computation for heat conduction problems can be found in [124]. A different view of the same problem can be found in [125–127], for variational data assimilation problems in meteorology with the shallow-water equations as state equations. The importance of exactly computing the Hessian is discussed in [128], where the condition number influence on the convergence of the optimization algorithm is addressed.

In the next subsections, the four approaches will be presented in both discrete, [120, 121], and continuous, [122], forms; note that, although the discrete approach is general and applies to either the inviscid or viscous flow equations, the continuous one will be restricted to the Euler equations and, in particular, to inverse design problems.

5.1 The Direct–Direct Approach

Starting from (11) and (12), the Hessian matrix of F can be computed using

$$\frac{\delta^2 F}{\delta b_i \delta b_j} = \frac{\partial^2 F}{\partial b_i \partial b_j} + \frac{\partial^2 F}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j} + \frac{\partial^2 F}{\partial U_k \partial b_j}\frac{\delta U_k}{\delta b_i} + \frac{\partial^2 F}{\partial U_k \partial U_m}\frac{\delta U_k}{\delta b_i}\frac{\delta U_m}{\delta b_j} + \frac{\partial F}{\partial U_k}\frac{\delta^2 U_k}{\delta b_i \delta b_j}. \tag{137}$$

The Hessian of the residual of the discretized flow equations is zero, so

$$\frac{\delta^2 R_n}{\delta b_i \delta b_j} = \frac{\partial^2 R_n}{\partial b_i \partial b_j} + \frac{\partial^2 R_n}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j} + \frac{\partial^2 R_n}{\partial U_k \partial b_j}\frac{\delta U_k}{\delta b_i} + \frac{\partial^2 R_n}{\partial U_k \partial U_m}\frac{\delta U_k}{\delta b_i}\frac{\delta U_m}{\delta b_j} + \frac{\partial R_n}{\partial U_k}\frac{\delta^2 U_k}{\delta b_i \delta b_j} = 0. \tag{138}$$

To compute $\frac{\delta^2 F}{\delta b_i \delta b_j}$ using (137), the numerical solution of $\frac{N(N+1)}{2}$ systems of equations, (138), for $\frac{\delta^2 U_k}{\delta b_i \delta b_j}$, is required. In addition, since $\frac{\delta U_k}{\delta b_j}$ must be computed as well, N equivalent flow solutions should be carried out. Consequently, $1 + N + \frac{N(N+1)}{2}$ system solutions per Newton's step are needed. Unambiguously, this cost is high and renders the so-called (discrete) direct–direct approach inappropriate for use in optimization problems with many design variables.

The continuous direct–direct approach is based on the continuous direct approach for the computation of first and second order sensitivities. In inverse design problems, where the objective function is given by (31), the second order sensitivities of F are given by

$$\frac{\delta^2 F}{\delta b_i \delta b_j} = \int_{S_w} \frac{\delta p}{\delta b_i}\frac{\delta p}{\delta b_j} dS + \int_{S_w} (p - p_{tar})\frac{\delta^2 p}{\delta b_i \delta b_j} dS + \int_{S_w} (p - p_{tar})\frac{\delta p}{\delta b_i}\frac{\delta (dS)}{\delta b_j} + \int_{S_w} (p - p_{tar})\frac{\delta p}{\delta b_j}\frac{\delta (dS)}{\delta b_i} + \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta^2 (dS)}{\delta b_i \delta b_j}. \tag{139}$$

To compute the second order sensitivities of the state variables, one may start from (recall that the flow is considered to be inviscid)

$$\frac{\partial^2}{\partial b_i \partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right)=0 \tag{140}$$

which, by neglecting the sensitivities of the Jacobian matrix ($\frac{\partial^2 A_{nmk}}{\partial b_i \partial b_j}=0$), takes the form

$$\frac{\partial}{\partial x_k}\left(A_{nmk}\frac{\partial^2 U_m}{\partial b_i \partial b_j}\right)=0. \tag{141}$$

The $\frac{1}{2}N(N+1)$ resulting pde's can be discretized using the Roe-like scheme for the corresponding fluxes crossing the boundaries of the finite volumes (as already mentioned in Sect. 3) and, then, numerically solved.

The boundary conditions imposed to (141) are based on the following equations

$$\left.\frac{\partial^2 p_t}{\partial b_i \partial b_j}\right|_I=0,\qquad \left.\frac{\partial^2 T_t}{\partial b_i \partial b_j}\right|_I=0,\qquad \left.\frac{\partial^2 \alpha}{\partial b_i \partial b_j}\right|_I=0 \tag{142}$$

at the inlet and

$$\left.\frac{\partial^2 p}{\partial b_i \partial b_j}\right|_O=0 \tag{143}$$

at the outlet and the corresponding no-penetration condition at solid walls. Equations (142) and (143) express that (in 2D problems) three flow variables (p_t, T_t and the flow angle α) are fixed at the inlet and the static pressure p is fixed at the outlet. In 3D flows, the second derivative of an extra flow angle at the inlet must also be zeroed. Once $\frac{\partial^2 U_m}{\partial b_i \partial b_j}$ have been computed by means of (141), the corresponding total variations $\frac{\delta^2 U_m}{\delta b_i \delta b_j}$ can be derived using

$$\frac{\delta^2 \Phi}{\delta b_i \delta b_j}=\frac{\partial^2 \Phi}{\partial b_i \partial b_j}+\frac{\partial}{\partial x_l}\left(\frac{\partial \Phi}{\partial b_i}\right)\frac{\delta x_l}{\delta b_j}+\frac{\partial}{\partial x_l}\left(\frac{\partial \Phi}{\partial b_j}\right)\frac{\delta x_l}{\delta b_i} + \frac{\partial^2 \Phi}{\partial x_l \partial x_m}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j}+\frac{\partial \Phi}{\partial x_l}\frac{\delta^2 x_l}{\delta b_i \delta b_j} \tag{144}$$

for $\Phi=U_m$. The total CPU cost is as high as that of the discrete direct–direct approach. A practical difference between the discrete and the continuous direct–direct approach is that the former is based on the computation of $\frac{\delta U_k}{\delta b_j}$ and $\frac{\delta^2 U_k}{\delta b_i \delta b_j}$ ($N+\frac{N(N+1)}{2}$ system solutions) whereas the latter computes $\frac{\partial U_k}{\partial b_j}$ (by solving (36)) and $\frac{\partial^2 U_k}{\partial b_i \partial b_j}$; both have the same CPU cost. Partial and total derivatives are transformed to each other using (37) and (144).

5.2 The Direct–Adjoint Approach

In the direct–adjoint approach, a new augmented functional $\hat{F}$ (a new symbol, $\hat{F}$, is used to distinguish it from F_{aug} which was used to compute first order sensitivities), with new adjoint variables $\hat{\Psi}_n$, is defined as follows

$$\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j}=\frac{\delta^2 F}{\delta b_i \delta b_j}+\hat{\Psi}_n\frac{\delta^2 R_n}{\delta b_i \delta b_j}, \tag{145}$$

where $\frac{\delta^2 F}{\delta b_i \delta b_j}$ and $\frac{\delta^2 R_n}{\delta b_i \delta b_j}$ are given by (137) and (138), respectively. Rearranging, we get

$$\begin{aligned}\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j}&=\frac{\partial^2 F}{\partial b_i \partial b_j}+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial b_i \partial b_j}+\frac{\partial^2 F}{\partial U_k \partial U_m}\frac{\delta U_k}{\delta b_i}\frac{\delta U_m}{\delta b_j}\\&\quad+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial U_k \partial U_m}\frac{\delta U_k}{\delta b_i}\frac{\delta U_m}{\delta b_j}+\frac{\partial^2 F}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j}\\&\quad+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j}+\frac{\partial^2 F}{\partial U_k \partial b_j}\frac{\delta U_k}{\delta b_i}\\&\quad+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial U_k \partial b_j}\frac{\delta U_k}{\delta b_i}+\left(\frac{\partial F}{\partial U_k}+\hat{\Psi}_n\frac{\partial R_n}{\partial U_k}\right)\frac{\delta^2 U_k}{\delta b_i \delta b_j}.\end{aligned} \tag{146}$$

The last term in parenthesis, which denotes the dependency of $\frac{\delta^2 F}{\delta b_i \delta b_j}$ on $\frac{\delta^2 U_k}{\delta b_i \delta b_j}$ is eliminated by satisfying the adjoint equations

$$\frac{\partial F}{\partial U_k}+\hat{\Psi}_n\frac{\partial R_n}{\partial U_k}=0. \tag{147}$$

Thus, the Hessian matrix elements are given by

$$\begin{aligned}\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j}&=\frac{\partial^2 F}{\partial b_i \partial b_j}+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial b_i \partial b_j}\\&\quad+\left(\frac{\partial^2 F}{\partial U_k \partial U_m}+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial U_k \partial U_m}\right)\frac{\delta U_k}{\delta b_i}\frac{\delta U_m}{\delta b_j}\\&\quad+\left(\frac{\partial^2 F}{\partial b_i \partial U_k}+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial b_i \partial U_k}\right)\frac{\delta U_k}{\delta b_j}\\&\quad+\left(\frac{\partial^2 F}{\partial U_k \partial b_j}+\hat{\Psi}_n\frac{\partial^2 R_n}{\partial U_k \partial b_j}\right)\frac{\delta U_k}{\delta b_i}.\end{aligned} \tag{148}$$

The CPU cost to obtain $\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j}$ derivatives is the sum of the cost of N system solutions to compute $\frac{\delta U_n}{\delta b_j}$ via (12) plus one more to solve the single adjoint equation, (147) ($N+1$ system solutions in total).

The corresponding continuous direct–adjoint approach is based on the second order sensitivity of an augmented objective function, defined as

$$\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j} = \frac{\delta^2 F}{\delta b_i \delta b_j} + \int_\Omega \hat{\Psi}_n \frac{\partial^2}{\partial b_i \partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) d\Omega, \tag{149}$$

where the adjoint variables have been multiplied with the second order direct sensitivities of the state (Euler) equations. Here, $\frac{\partial}{\partial x_k}$ and $\frac{\partial^2}{\partial b_i \partial b_j}$ can be interchanged to give

$$\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j} = \frac{\delta^2 F}{\delta b_i \delta b_j} + \int_\Omega \hat{\Psi}_n \frac{\partial}{\partial x_k}\left(\frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial b_j}\right) d\Omega. \tag{150}$$

The integral on the r.h.s. of (150) is integrated by parts as follows

$$\int_\Omega \hat{\Psi}_n \frac{\partial}{\partial x_k}\left(\frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial b_j}\right) d\Omega = \int_\Omega \left(A_{nmk}\frac{\partial \hat{\Psi}_n}{\partial x_k}\right)\frac{\partial^2 U_m}{\partial b_i \partial b_j} d\Omega + \int_S \hat{\Psi}_n \frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial b_j} n_k dS. \tag{151}$$

Multiplying (144), for $\Phi = f_{nk}^{inv}$, with the normal unit vector at the solid walls, we obtain

$$\begin{aligned}\frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial b_j} n_k &= \frac{\delta^2 f_{nk}^{inv}}{\delta b_i \delta b_j} n_k - \frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial x_l}\frac{\delta x_l}{\delta b_j} n_k - \frac{\partial^2 f_{nk}^{inv}}{\partial b_j \partial x_l}\frac{\delta x_l}{\delta b_i} n_k \\ &\quad - \frac{\partial^2 f_{nk}^{inv}}{\partial x_l \partial x_m}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j} n_k - \frac{\partial f_{nk}^{inv}}{\partial x_l}\frac{\delta^2 x_l}{\delta b_i \delta b_j} n_k.\end{aligned} \tag{152}$$

Also,

$$\begin{aligned}\frac{\delta^2 f_{nk}^{inv}}{\delta b_i \delta b_j} n_k &= \frac{\delta^2 \left(f_{nk}^{inv} n_k\right)}{\delta b_i \delta b_j} - \frac{\delta f_{nk}^{inv}}{\delta b_i}\frac{\delta n_k}{\delta b_j} - \frac{\delta f_{nk}^{inv}}{\delta b_j}\frac{\delta n_k}{\delta b_i} \\ &\quad - f_{nk}^{inv}\frac{\delta^2 n_k}{\delta b_i \delta b_j}\end{aligned} \tag{153}$$

and, along the solid walls, the no-penetration condition for the velocity, gives

$$\begin{aligned}\frac{\delta^2 \left(f_{nk}^{inv} n_k\right)}{\delta b_i \delta b_j} &= N_n \frac{\delta^2 p}{\delta b_i \delta b_j} + \frac{\delta^2 N_n}{\delta b_i \delta b_j} p + \frac{\delta N_n}{\delta b_i}\frac{\delta p}{\delta b_j} \\ &\quad + \frac{\delta N_n}{\delta b_j}\frac{\delta p}{\delta b_i},\end{aligned} \tag{154}$$

where N_n is defined by (49). By appropriately rearranging its terms, (151) becomes

$$\begin{aligned}&\int_\Omega \hat{\Psi}_n \frac{\partial}{\partial x_k}\left(\frac{\partial^2 f_{nk}^{inv}}{\partial b_i \partial b_j}\right) d\Omega \\ &= -\underbrace{\int_\Omega \left(A_{nmk}\frac{\partial \hat{\Psi}_n}{\partial x_k}\right)\frac{\partial^2 U_m}{\partial b_i \partial b_j} d\Omega}_{FAE} \\ &\quad + \underbrace{\int_{S_{I,O}} \hat{\Psi}_n \frac{\delta^2 (A_{nm} U_m)}{\delta b_i \delta b_j} dS}_{IOBC} + \underbrace{\int_{S_w} \hat{\Psi}_{n+1} n_n \frac{\delta^2 p}{\delta b_i \delta b_j} dS}_{SWCR1} \\ &\quad + \int_{S_w} (\hat{\Psi}_{k+1} p - \hat{\Psi}_n f_{nk}^{inv})\frac{\delta^2 n_k}{\delta b_i \delta b_j} dS \\ &\quad + \int_{S_w} \left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_i} - \hat{\Psi}_n \frac{\delta f_{nk}^{inv}}{\delta b_i}\right)\frac{\delta n_k}{\delta b_j} dS \\ &\quad + \int_{S_w} \left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_j} - \hat{\Psi}_n \frac{\delta f_{nk}^{inv}}{\delta b_j}\right)\frac{\delta n_k}{\delta b_i} dS \\ &\quad - \int_{S_w} \hat{\Psi}_n A_{nqk} n_k \left(\frac{\partial^2 U_q}{\partial b_i \partial x_l}\frac{\delta x_l}{\delta b_j} + \frac{\partial^2 U_q}{\partial b_j \partial x_l}\frac{\delta x_l}{\delta b_i}\right. \\ &\quad \left. + \frac{\partial^2 U_q}{\partial x_l \partial x_m}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j} + \frac{\partial U_q}{\partial x_l}\frac{\delta^2 x_l}{\delta b_i \delta b_j}\right) dS.\end{aligned} \tag{155}$$

Combining (139) and (155), we get

$$\begin{aligned}&\frac{\delta^2 \hat{F}_{aug}}{\delta b_i \delta b_j} \\ &= \int_{S_w} \frac{\delta p}{\delta b_i}\frac{\delta p}{\delta b_j} dS + \underbrace{\int_{S_w} (p - p_{tar})\frac{\delta^2 p}{\delta b_i \delta b_j} dS}_{SWCR} \\ &\quad + \int_{S_w} (p - p_{tar})\frac{\delta p}{\delta b_i}\frac{\delta (dS)}{\delta b_j} \\ &\quad + \int_{S_w} (p - p_{tar})\frac{\delta p}{\delta b_j}\frac{\delta (dS)}{\delta b_i} \\ &\quad + \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta^2 (dS)}{\delta b_i \delta b_j} \\ &\quad - \underbrace{\int_\Omega \left(A_{nmk}\frac{\partial \hat{\Psi}_n}{\partial x_k}\right)\frac{\partial^2 U_m}{\partial b_i \partial b_j} d\Omega}_{FAE} \\ &\quad + \underbrace{\int_{S_{I,O}} \hat{\Psi}_n \frac{\delta^2 (A_{nm} U_m)}{\delta b_i \delta b_j} dS}_{IOBC}\end{aligned}$$

$$
\begin{aligned}
&+\underbrace{\int_{S_w}\hat{\Psi}_{n+1}n_n\frac{\delta^2 p}{\delta b_i\delta b_j}dS}_{SWCR}\\
&+\int_{S_w}(\hat{\Psi}_{k+1}p-\hat{\Psi}_n f^{inv}_{nk})\frac{\delta^2 n_k}{\delta b_i\delta b_j}dS\\
&+\int_{S_w}\left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_i}-\hat{\Psi}_n\frac{\delta f^{inv}_{nk}}{\delta b_i}\right)\frac{\delta n_k}{\delta b_j}dS\\
&+\int_{S_w}\left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_j}-\hat{\Psi}_n\frac{\delta f^{inv}_{nk}}{\delta b_j}\right)\frac{\delta n_k}{\delta b_i}dS\\
&-\int_{S_w}\hat{\Psi}_n A_{nqk}n_k\left(\frac{\partial^2 U_q}{\partial b_i\partial x_l}\frac{\delta x_l}{\delta b_j}+\frac{\partial^2 U_q}{\partial b_j\partial x_l}\frac{\delta x_l}{\delta b_i}\right)dS\\
&-\int_{S_w}\hat{\Psi}_n A_{nqk}n_k\left(\frac{\partial^2 U_q}{\partial x_l\partial x_m}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j}\right.\\
&\left.+\frac{\partial U_q}{\partial x_l}\frac{\delta^2 x_l}{\delta b_i\delta b_j}\right)dS. \qquad (156)
\end{aligned}
$$

Integrals marked with FAE, $SWCR$ and $IOBC$ are eliminated by satisfying appropriate field adjoint equations and the boundary conditions over the wall and inlet/outlet boundaries. It can be seen that these are given by the same equations as those solved for the computation of first order sensitivities. Finally, the remaining terms in (156) yield an expression for $\frac{\delta^2 \hat{F}_{aug}}{\delta b_i\delta b_j}$, which is as follows

$$
\begin{aligned}
&\frac{\delta^2 \hat{F}_{aug}}{\delta b_i\delta b_j}\\
&=\int_{S_w}\frac{\delta p}{\delta b_i}\frac{\delta p}{\delta b_j}dS+\int_{S_w}(p-p_{tar})\frac{\delta p}{\delta b_i}\frac{\delta(dS)}{\delta b_j}\\
&+\int_{S_w}(p-p_{tar})\frac{\delta p}{\delta b_j}\frac{\delta(dS)}{\delta b_i}\\
&+\frac{1}{2}\int_{S_w}(p-p_{tar})^2\frac{\delta^2(dS)}{\delta b_i\delta b_j}\\
&+\int_{S_w}(\hat{\Psi}_{k+1}p-\hat{\Psi}_n f^{inv}_{nk})\frac{\delta^2 n_k}{\delta b_i\delta b_j}dS\\
&+\int_{S_w}\left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_i}-\hat{\Psi}_n\frac{\delta f^{inv}_{nk}}{\delta b_i}\right)\frac{\delta n_k}{\delta b_j}dS\\
&+\int_{S_w}\left(\hat{\Psi}_{k+1}\frac{\delta p}{\delta b_j}-\hat{\Psi}_n\frac{\delta f^{inv}_{nk}}{\delta b_j}\right)\frac{\delta n_k}{\delta b_i}dS\\
&-\int_{S_w}\hat{\Psi}_n A_{nqk}n_k\left(\frac{\partial^2 U_q}{\partial b_i\partial x_l}\frac{\delta x_l}{\delta b_j}+\frac{\partial^2 U_q}{\partial b_j\partial x_l}\frac{\delta x_l}{\delta b_i}\right)dS\\
&-\int_{S_w}\hat{\Psi}_n A_{nqk}n_k\left(\frac{\partial^2 U_q}{\partial x_l\partial x_m}\frac{\delta x_l}{\delta b_i}\frac{\delta x_m}{\delta b_j}+\frac{\partial U_q}{\partial x_l}\frac{\delta^2 x_l}{\delta b_i\delta b_j}\right)dS\\
&\qquad (157)
\end{aligned}
$$

For (157) to be computable, first order flow sensitivities must be known; these are computed using (36); so, the total CPU cost corresponds to $N+1$ system solutions. Since, among other, eq, (157) includes boundary integrals written in terms of the second order sensitivities of geometrical quantities such as x_k, it is important to stress that, in some frequently used parameterizations (such as the Bézier–Bernstein polynomials), these terms automatically vanish, since the shape coordinates are linear functions of the design variables.

5.3 The Adjoint–Direct Approach

A scheme based on the adjoint formulation for the computation of the gradient and the direct approach for the computation of the Hessian matrix will be referred to as the adjoint–direct approach. By differentiating (19) (for a single objective, $L=1$), we get

$$
\begin{aligned}
\frac{\delta^2 F_{aug}}{\delta b_i\delta b_j}&=\frac{\partial^2 F_{aug}}{\partial b_i\partial b_j}+\frac{\partial^2 F_{aug}}{\partial b_i\partial U_k}\frac{\delta U_k}{\delta b_j}+\Psi_m\frac{\partial^2 R_m}{\partial b_i\partial b_j}\\
&\quad+\Psi_m\frac{\partial^2 R_m}{\partial b_i\partial U_k}\frac{\delta U_k}{\delta b_j}+\frac{\delta \Psi_m}{\delta b_j}\frac{\partial R_m}{\partial b_i}. \qquad (158)
\end{aligned}
$$

The required derivatives $\frac{\delta\Psi_m}{\delta b_j}$ are computed by zeroing the gradient of the residual of the adjoint equations, (18), namely

$$
\begin{aligned}
\frac{\delta R^{\Psi}_n}{\delta b_j}&=\frac{\partial^2 F_{aug}}{\partial U_n\partial b_j}+\frac{\partial^2 F_{aug}}{\partial U_n\partial U_k}\frac{\delta U_k}{\delta b_j}+\Psi_m\frac{\partial^2 R_m}{\partial U_n\partial b_j}\\
&\quad+\Psi_m\frac{\partial^2 R_m}{\partial U_n\partial U_k}\frac{\delta U_k}{\delta b_j}+\frac{\delta\Psi_m}{\delta b_j}\frac{\partial R_m}{\partial U_n}=0. \qquad (159)
\end{aligned}
$$

The CPU cost for computing $\frac{\delta F_{aug}}{\delta b_i}$ and $\frac{\delta^2 F_{aug}}{\delta b_i\delta b_j}$ through the adjoint–direct approach is $1+2N$ system solutions; among them, N must be spent to compute $\frac{\delta U_k}{\delta b_j}$, (12), and N more to compute $\frac{\delta\Psi_m}{\delta b_j}$, (159).

The continuous adjoint–direct approach employs the continuous adjoint method to compute the gradient of the objective function (a method which has already been presented in detail) and the direct differentiation approach to compute the Hessian matrix. In an inverse design problem, the first order sensitivities of F are given by (59) whereas the adjoint field equations and solid wall boundary conditions by (56) and (58).

The sensitivity of (59) with respect to b_j yields

$$
\begin{aligned}
\frac{\delta^2 F_{aug}}{\delta b_i\delta b_j}&=\frac{1}{2}\int_{S_w}(p-p_{tar})^2\frac{\delta^2(dS)}{\delta b_i\delta b_j}\\
&\quad+\int_{S_w}(p-p_{tar})\frac{\delta p}{\delta b_j}\frac{\delta(dS)}{\delta b_i}
\end{aligned}
$$

$$
\begin{aligned}
&-\int_{S_w} A_{nmk}\Psi_n \frac{\partial U_m}{\partial x_k}\frac{\delta^2 x_k}{\delta b_i \delta b_j} dS \\
&-\int_{S_w} A_{nmk}\frac{\delta \Psi_n}{\delta b_j}\frac{\partial U_m}{\partial x_k}\frac{\delta x_k}{\delta b_i} dS \\
&-\int_{S_w} A_{nmk}\Psi_n \frac{\partial U_m}{\partial x_k}\frac{\delta x_k}{\delta b_i}\frac{\delta (dS)}{\delta b_j} \\
&-\int_{S_w} A_{nmk}\Psi_n \frac{\delta}{\delta b_j}\left(\frac{\partial U_m}{\partial x_k}\right)\frac{\delta x_k}{\delta b_i} dS \\
&+\int_{S_w}\left(\Psi_{k+1} p - \Psi_n f_{nk}^{inv}\right)\frac{\delta^2 (n_k dS)}{\delta b_i \delta b_j} \\
&+\int_{S_w}\left(\frac{\delta \Psi_{k+1}}{\delta b_j} p - \frac{\delta \Psi_n}{\delta b_j} f_{nk}^{inv}\right)\frac{\delta (n_k dS)}{\delta b_i} \\
&+\int_{S_w}\left(\Psi_{k+1}\frac{\delta p}{\delta b_j} - \Psi_n \frac{\delta f_{nk}^{inv}}{\delta b_j}\right)\frac{\delta (n_k dS)}{\delta b_i}. \qquad (160)
\end{aligned}
$$

Equations (160) includes quantities such as $\frac{\delta}{\delta b_j}(\frac{\partial U_m}{\partial x_k})$ (or $\frac{\partial}{\partial x_k}(\frac{\delta U_m}{\delta b_j})$) and $\frac{\delta \Psi_n}{\delta b_j}$ which are to be computed beforehand. Note that, although the adjoint method is used to compute $\frac{\delta F}{\delta b_i}$, we still need the gradient of U_m and Ψ_n with respect to the design variables. Hence, (36) along with its counterpart for Ψ_n, must be solved. It can easily be shown that, as in the discrete formulation, each Newton cycle costs as many as $2+2N$ equivalent flow solutions, being almost twice more expensive than the direct–adjoint approach.

5.4 The Adjoint–Adjoint Approach

The fourth possible scheme to compute $\frac{\delta^2 F}{\delta b_i \delta b_j}$ is the adjoint–adjoint approach which introduces extra adjoint variables, denoted by $\overline{\Psi}_{im}$ and $\overline{\overline{\Psi}}_{in}$. A twice–augmented objective function is first defined as

$$
\frac{\delta^2 \overline{F}_{aug}}{\delta b_i \delta b_j} = \frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} + \overline{\Psi}_{im}\frac{\delta R_m}{\delta b_j} + \overline{\overline{\Psi}}_{in}\frac{\delta R_n^{\Psi}}{\delta b_j} \qquad (161)
$$

which, after substituting (158) for the Hessian of F_{aug} and appropriate expressions for the gradient of the flow and adjoint equations, yields

$$
\begin{aligned}
\frac{\delta^2 \overline{F}_{aug}}{\delta b_i \delta b_j} &= \frac{\partial^2 F_{aug}}{\partial b_i \partial b_j} + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial b_j} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial b_j} \\
&\quad + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F_{aug}}{\partial U_n \partial b_j} + \overline{\overline{\Psi}}_{in}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial b_j} \\
&\quad + \left(\frac{\partial^2 F_{aug}}{\partial b_i \partial U_k} + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial U_k} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial U_k}\right. \\
&\quad \left. + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F_{aug}}{\partial U_n \partial U_k} + \overline{\overline{\Psi}}_{in}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial U_k}\right)\frac{\delta U_k}{\delta b_j} \\
&\quad + \left(\frac{\partial R_m}{\partial b_i} + \overline{\overline{\Psi}}_{in}\frac{\partial R_m}{\partial U_n}\right)\frac{\delta \Psi_m}{\delta b_j}. \qquad (162)
\end{aligned}
$$

In (162), the two terms in parenthesis are eliminated by satisfying the adjoint system of equations, namely

$$
\begin{aligned}
&\frac{\partial^2 F_{aug}}{\partial b_i \partial U_k} + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial U_k} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial U_k} + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F_{aug}}{\partial U_n \partial U_k} \\
&\quad + \overline{\overline{\Psi}}_{in}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial U_k} = 0 \qquad (163)
\end{aligned}
$$

to be solved for $\overline{\Psi}_{im}$ and

$$
\frac{\partial R_m}{\partial b_i} + \overline{\overline{\Psi}}_{in}\frac{\partial R_m}{\partial U_n} = 0 \qquad (164)
$$

to be solved for $\overline{\overline{\Psi}}_{in}$; (164) should be solved first since (163) requires the knowledge of $\overline{\overline{\Psi}}_{in}$. The Hessian matrix is, then, given by

$$
\begin{aligned}
\frac{\delta^2 \overline{F}_{aug}}{\delta b_i \delta b_j} &= \frac{\partial^2 F_{aug}}{\partial b_i \partial b_j} + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial b_j} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial b_j} \\
&\quad + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F_{aug}}{\partial U_n \partial b_j} + \overline{\overline{\Psi}}_{in}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial b_j}. \qquad (165)
\end{aligned}
$$

The adjoint–adjoint approach requires $1+2N$ system solutions for the computation of the first and second derivatives, i.e. as many as those required by the adjoint–direct one. This cost is much higher than that of the direct–adjoint approach.

In the continuous adjoint–adjoint approach, the twice-augmented objective function $\overline{F}_{aug}$ is again defined by introducing two sets of Lagrange multipliers, one ($\overline{\overline{\Psi}}_{in}, i = 1, \ldots, N$) for the flow and the other ($\overline{\Psi}_{im}, i = 1, \ldots, N$) for the adjoint equations. The second order sensitivity of $\overline{F}_{aug}$ is written as

$$
\begin{aligned}
\frac{\delta^2 \overline{F}_{aug}}{\delta b_i \delta b_j} &= \frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} + \int_{\Omega} \overline{\overline{\Psi}}_{in}\frac{\partial}{\partial b_j}\left(\frac{\partial f_{nk}^{inv}}{\partial x_k}\right) d\Omega \\
&\quad + \int_{\Omega} \overline{\Psi}_{im}\frac{\partial}{\partial b_j}\left(A_{nmk}\frac{\partial \Psi_n}{\partial x_k}\right) d\Omega. \qquad (166)
\end{aligned}
$$

Both field integrals in (166) are integrated by parts to yield

$$
\begin{aligned}
\frac{\delta^2 \overline{F}_{aug}}{\delta b_i \delta b_j} &= \frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} - \int_{\Omega} A_{nmk}\frac{\partial \overline{\overline{\Psi}}_{in}}{\partial x_k}\frac{\partial U_m}{\partial b_j} d\Omega \\
&\quad + \int_S A_{nmk}\overline{\overline{\Psi}}_{in}\frac{\partial U_m}{\partial b_j} n_k dS \\
&\quad - \int_{\Omega} \frac{\partial (A_{nmk}\overline{\Psi}_{im})}{\partial x_k}\frac{\partial \Psi_n}{\partial b_j} d\Omega \\
&\quad + \int_S A_{nmk}\overline{\Psi}_{im}\frac{\partial \Psi_n}{\partial b_j} n_k dS, \qquad (167)
\end{aligned}
$$

where $\frac{\delta^2 F_{aug}}{\delta b_i \delta b_j}$ is given by (160). In (167), terms depending on $\frac{\partial U_m}{\partial b_j}$ are eliminated, giving rise to the first set of field adjoint equations in terms of $\overline{\overline{\Psi}}_{in}$. A second set of adjoint equations in terms of $\overline{\Psi}_{im}$ is then derived which, when satisfied, eliminate integrals expressed in terms of $\frac{\partial \Psi_n}{\partial b_j}$. As in the discrete adjoint–adjoint method, the CPU cost of the adjoint–adjoint method is equal to $2+2N$ equivalent flow solutions per Newton cycle.

5.5 Further Discussion on the Computation of the Hessian

In [120], the authors have shown that an alternative formulation can directly lead to the direct–adjoint resulting expression for the Hessian. Based on the augmented function F_{aug}, (158), let us define the new, twice-augmented function $\widehat{F}_{aug}$ as follows

$$\frac{\delta^2 \widehat{F}_{aug}}{\delta b_i \delta b_j} = \frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} + \frac{\delta U_n}{\delta b_i}\frac{\delta R_n^{\Psi}}{\delta b_j} \tag{168}$$

by using, instead of adjoint variables, the total sensitivities of the state variables $\frac{\delta U_n}{\delta b_i}$.

If (158) and (159) are substituted into (168), the following expression is obtained

$$\begin{aligned}
\frac{\delta^2 \widehat{F}_{aug}}{\delta b_i \delta b_j} = {} & \frac{\partial^2 F_{aug}}{\partial b_i \partial b_j} + \frac{\partial^2 F_{aug}}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j} + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial b_j} \\
& + \Psi_m \frac{\partial^2 R_m}{\partial b_i \partial U_k}\frac{\delta U_k}{\delta b_j} + \underbrace{\frac{\delta \Psi_m}{\delta b_j}\frac{\partial R_m}{\partial b_i}}_{DDE} + \frac{\delta U_n}{\delta b_i}\frac{\partial^2 F_{aug}}{\partial U_n \partial b_j} \\
& + \frac{\partial^2 F_{aug}}{\partial U_n \partial U_k}\frac{\delta U_n}{\delta b_i}\frac{\delta U_k}{\delta b_j} + \frac{\delta U_n}{\delta b_i}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial b_j} \\
& + \frac{\delta U_n}{\delta b_i}\Psi_m \frac{\partial^2 R_m}{\partial U_n \partial U_k}\frac{\delta U_k}{\delta b_j} + \underbrace{\frac{\delta U_n}{\delta b_i}\frac{\delta \Psi_m}{\delta b_j}\frac{\partial R_m}{\partial U_n}}_{DDE},
\end{aligned} \tag{169}$$

where the two terms marked with DDE vanish, since

$$\frac{\delta \Psi_m}{\delta b_j}\left(\frac{\partial R_m}{\partial b_i} + \frac{\delta U_n}{\delta b_i}\frac{\partial R_m}{\partial U_n}\right) = \frac{\delta \Psi_m}{\delta b_j}\frac{\delta R_m}{\delta b_i} \equiv 0.$$

Equation (169) is an alternative expression of the computation of second order sensitivities. Surprisingly, this expression is free of the sensitivities of the adjoint variables; all we need to compute the Hessian of F is the adjoint variable Ψ and the first order sensitivities of the state variables $\frac{\delta U_k}{\delta b_j}$. Therefore, this alternative method is equivalent to the direct–adjoint approach (compare (169) and (148)).

Likewise, we may also devise the corresponding continuous adjoint formulation. Let us define

$$\frac{\delta^2 \widehat{F}}{\delta b_i \delta b_j} = \frac{\delta^2 F_{aug}}{\delta b_i \delta b_j} + \int_{\Omega} \frac{\delta U_m}{\delta b_i}\frac{\partial}{\partial b_j}\left(A_{nmk}\frac{\partial \Psi_n}{\partial x_k}\right) d\Omega, \tag{170}$$

where, with the assumption that $\frac{\partial A_{nmk}}{\partial b_j} = 0$,

$$\begin{aligned}
& \int_{\Omega} \frac{\delta U_m}{\delta b_i}\frac{\partial}{\partial b_j}\left(A_{nmk}\frac{\partial \Psi_n}{\partial x_k}\right) d\Omega \\
& \quad = \int_{\Omega} \frac{\delta U_m}{\delta b_i} A_{nmk}\frac{\partial}{\partial x_k}\left(\frac{\partial \Psi_n}{\partial b_j}\right) d\Omega \\
& \quad = -\int_{\Omega} \frac{\partial}{\partial x_k}\left(A_{nmk}\frac{\delta U_m}{\delta b_i}\right)\frac{\partial \Psi_n}{\partial b_j} d\Omega \\
& \quad = +\int_{S_w} A_{nmk}\frac{\delta U_m}{\delta b_i}\frac{\partial \Psi_n}{\partial b_j} n_k dS
\end{aligned} \tag{171}$$

which finally leads (inverse design problems for inviscid flows) to (157).

5.6 Hessian Computation and Parallel Computing

Summarizing, the most efficient approach for the computation of the objective function second order sensitivities is the direct–adjoint approach. The gain from the combination of the direct and adjoint approach is great, since the $1+N+\frac{N(N+1)}{2}$ equivalent flow solutions per Newton iteration are reduced to only $2+N$. At each optimization cycle, one system of flow equations, one system of adjoint equations and as many direct differentiation systems as the number of design variables must be solved. Although the CPU cost of the direct–adjoint approach appears to be the minimum possible, it is still proportional to the number of design variables.

An approach to lower the CPU time for the computation of the Hessian matrix is the use of parallel computing. The flow and the adjoint solvers can be parallelized in the ordinary way, dividing the domain into a number of subdomains, usually equal to the number of available processors. The CPU time is almost reduced by the number of the processors, depending on the grid partitioning and other programming issues.

On the other hand, the solution of the direct differentiation systems can be implemented in parallel by assigning the solution of the linearized flow equations with respect to each design variable to a single processor. Such a way of parallelization is straightforward, since the direct differentiation system solutions are independent from each other. If the number of available processors is greater than that of the design variables, each of these systems may be solved through additional parallel subdomaining, according to a parallel-in-parallel coding structure.

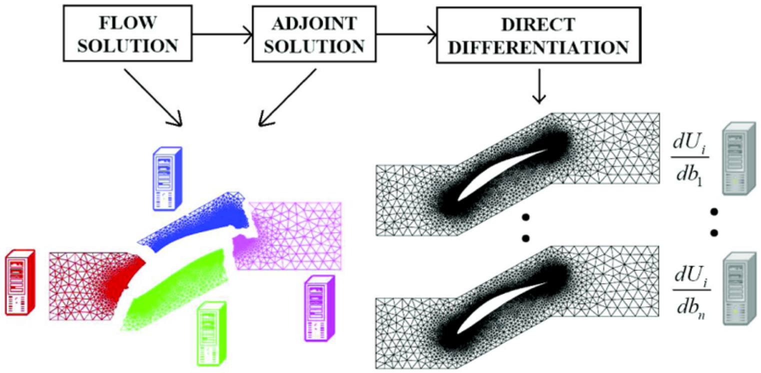


Fig. 20 Parallelization of the flow and adjoint equations through subdomaining and the direct differentiations through concurrent computations using a different processor for the flow sensitivities per design variable

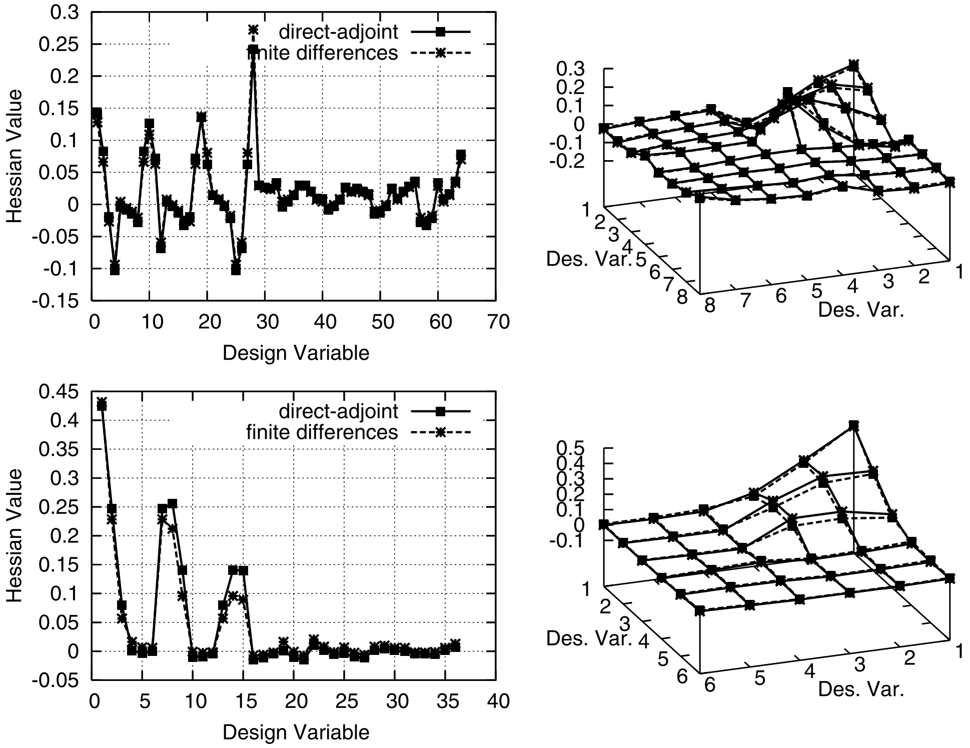


Fig. 21 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Hessian matrix values computed using the direct–adjoint approach and a central finite difference scheme. *Top-left*: $8 \times 8 = 64$ Hessian matrix values in a row. The 8 first correspond to the first column of the Hessian and so forth. *Top-right*: 3D plot of the same values. *Bottom*: same figures for the compressor case

The overall parallel optimization algorithm is illustrated in Fig. 20. The flow and adjoint equations are solved by dividing the grid to as many subdomains as the number of the available processors and the direct differentiation equations are solved separately using a different processor for each design variable. Of course, as explained, the extra parallelization of each direct differentiation system is still possible, if more processors are available.

5.7 Hessian-Based Optimization—Applications

The inverse design of the compressor and turbine cascades, namely those presented in a previous section using first order sensitivities, is repeated here with the aid of second order sensitivities. The comparison of the second order sensitivities computed using the direct–adjoint approach (i.e. the most efficient among the four examined algorithms) and central finite differences is shown in Fig. 21. For the turbine case, the cost of central finite differences is equal to 128 equivalent flow solutions: $2 \times 8 = 16$ solutions for the main Hessian matrix diagonal and $\frac{7\times 8}{2} \times 4 = 112$ solutions for the non-diagonal terms, without considering the analysis of the current configuration (computation of F). The cost of the direct–direct approach would be equal to $\frac{7\times 8}{2} = 28$ equivalent flow solutions which is still high enough compared to the $8 + 1 = 9$ solutions required by the direct–adjoint approach. The same analogy exists in the compressor case. Despite these huge differences in CPU cost, Fig. 21 shows that the direct–adjoint approach and the finite differences compute the Hessian matrix with the same accuracy.

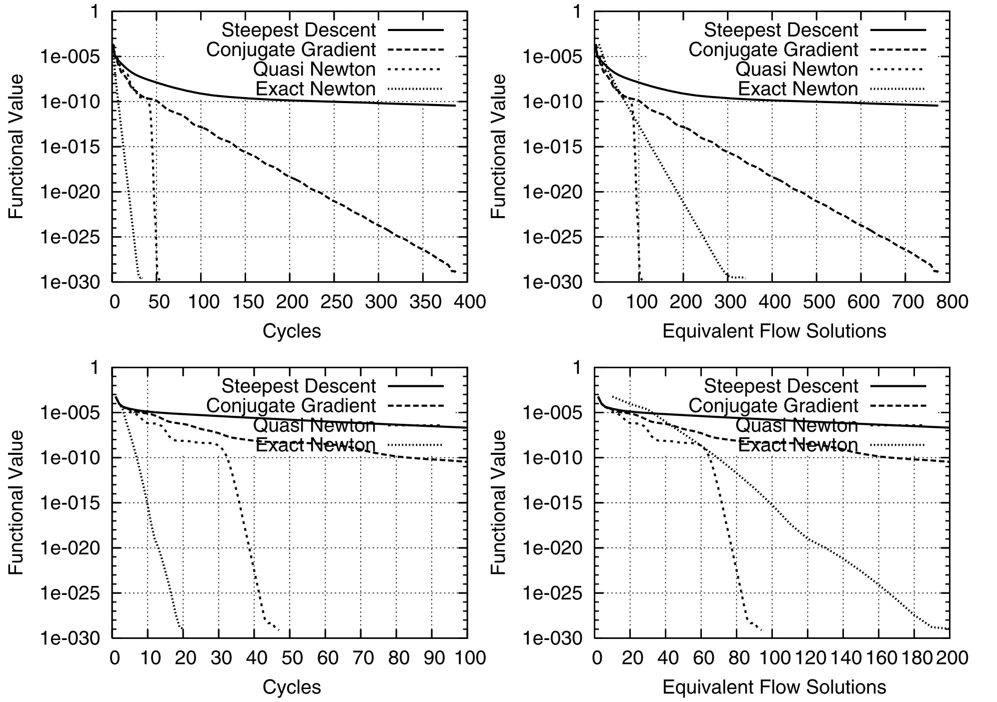


Fig. 22 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Reduction rate of the inverse design objective function value using four different optimization algorithms; steepest descent, the Fletcher–Reeves conjugate gradient, the BFGS quasi Newton and the exact Newton algorithms

The three gradient-based optimization algorithms are compared to the exact Newton algorithm which makes use of both the first and the second order sensitivity derivatives, Fig. 22. In this case, the Hessian matrix components are computed exactly using the direct–adjoint approach, in contrast to the BFGS algorithm in which these are approximated using the exact first order sensitivities of the previous iteration.

The comparison is firstly made with respect to the number of optimization cycles and, then, with respect to the CPU cost. For the latter, the x-axis is multiplied by two in the first three approaches while, in the exact Newton approach, it is multiplied by ten in the turbine design and eight in the compressor design. Methods based on the exact or approximated Hessian matrix are more efficient that gradient-based ones. In addition, the exact Newton approach needs less cycles to converge than the BFGS approach, which however seems to be faster, since it requires less equivalent flow solutions.

The need, however, of the exact second order sensitivities is demonstrated in Fig. 23. In this figure, the first two curves are exactly the same as in Fig. 22 corresponding to the standard quasi and exact Newton approaches. The third curve, however, is based on the computation of the exact second order sensitivities only once, i.e. at the first optimization cycle. The computed Hessian values are kept constant during the next cycles, where only the first order sensitivities are updated. In this case, due to the fact that the Hessian matrix value is kept frozen (i.e. it is not exact after the first cycle) more cycles than the exact Newton method but less than BFGS are needed. The great CPU cost for the repetitive computation of the Hessian matrix is avoided, leading to a lower number of equivalent flow solutions than the exact Newton approach, being, however, comparable to those required by the quasi Newton approach. A final remark on this algorithm is that, in more complex cases, in which the change in the Hessian matrix values cannot be neglected, the convergence rate is expected to deteriorate.

This is not the case, however, for the last curve in Fig. 23, which is a mixed exact-quasi Newton approach. The exact second order sensitivities are computed only at the first cycle using the direct–adjoint approach. At the next cycles, the so-computed Hessian values are updated using the BFGS update formula. It seems that the required cycles are even more reduced (only 15 cycles are needed for 26 orders of magnitude reduction in the objective function value). However, the most important remark is that the CPU cost is less than halved compared to the quasi Newton or the constant Hessian approach. Likewise the constant Hessian approach, this cost is almost independent of the number of design variables; ten equivalent flow solutions are required at the first cycle and only two during each of the subsequent cycles. From the second and fourth curve in Fig. 23, similar convergence behaviours (same slopes) are observed; the two rates of reduction differ only at the first part, due to the inexact (and probably poor guess of the) Hessian, if the quasi Newton approach is used from the very beginning.

The corresponding reduction rate of the first order sensitivity vector norm is shown in Figs. 24 and 25. Same comments are valid for these figures, as well.

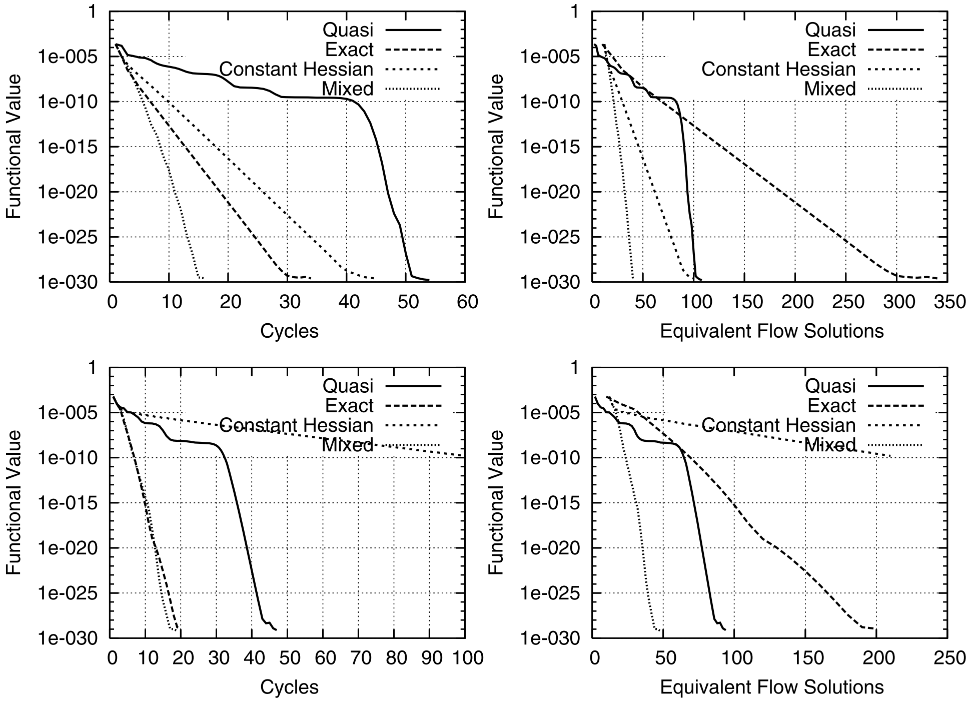


Fig. 23 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Reduction rate of the inverse design objective function value using four different optimization algorithms that make use of the approximate and/or exact second order sensitivities. The first two curves correspond to the standard quasi and exact Newton algorithms. The third curve corresponds to the computation of the exact Hessian matrix only at the first cycle which is then kept constant during the subsequent cycles, where the Newton method is employed. The last curve corresponds to the computation of the exact Hessian matrix at the first iteration and the use of the BFGS Hessian matrix update formula at the following cycles

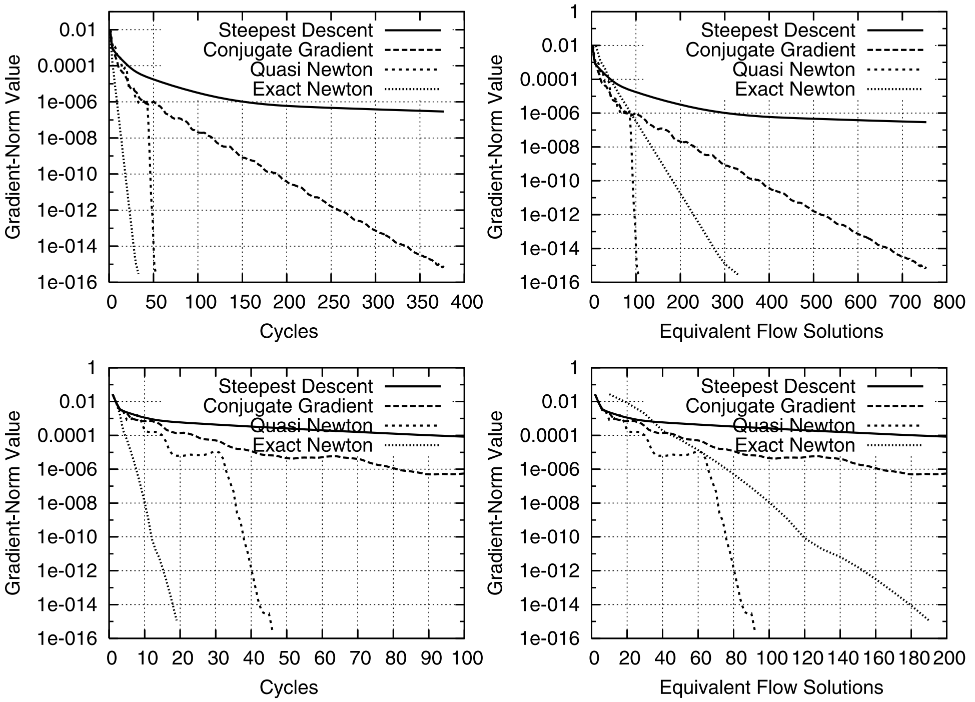


Fig. 24 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Reduction rate of the first order sensitivity values of the inverse design objective function using four different optimization algorithms; the steepest descent, the Fletcher–Reeves conjugate gradient, the BFGS quasi Newton and the exact Newton algorithms. The first three of them are based on first order sensitivities while the third one uses an approximation of the Hessian matrix. The exact Hessian matrix is employed only in the last algorithm

Changes in the first and second order sensitivities using the exact Newton approach are shown in Fig. 26. The first order sensitivity values drop by almost fourteen orders of magnitude within thirty cycles (fifteen cycles using the mixed approach) while the Hessian matrix changes only a little within these cycles. This explains why, in this case, the constant Hessian Newton approach performs quite well.

6 Conclusions

Discrete and continuous adjoint approaches to compute first and second order sensitivities and support aerodynamic shape optimization methods have been presented. For the purpose of method demonstration, the inverse design or the optimization of cascade airfoils for minimum viscous losses

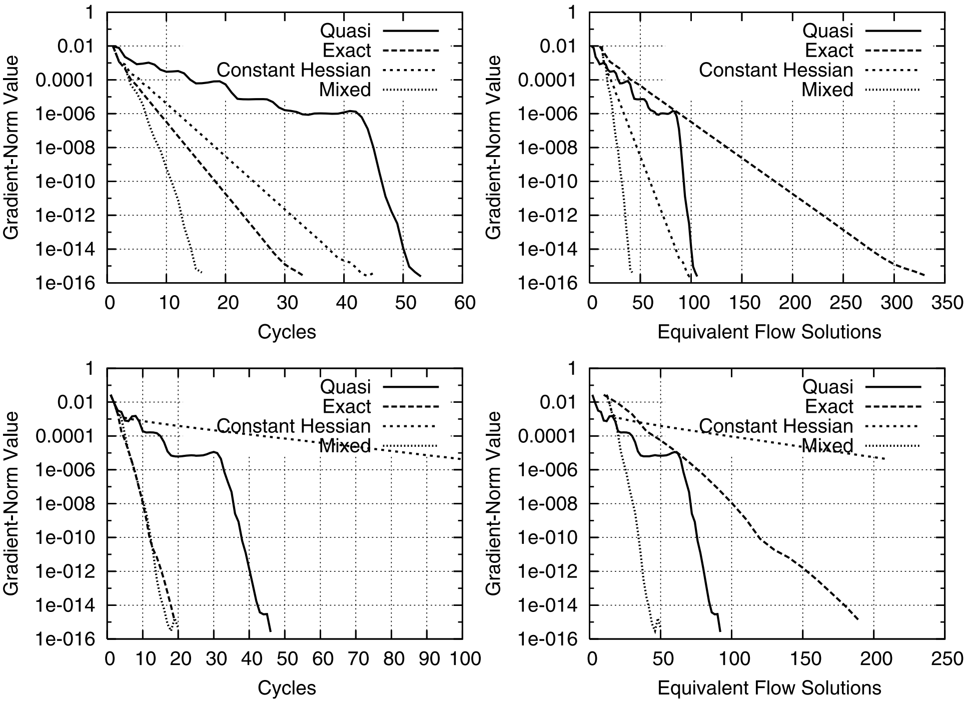


Fig. 25 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. Reduction rate of the first order sensitivity values of the inverse design objective function, using the four Newton-like algorithms also used in Fig. 23

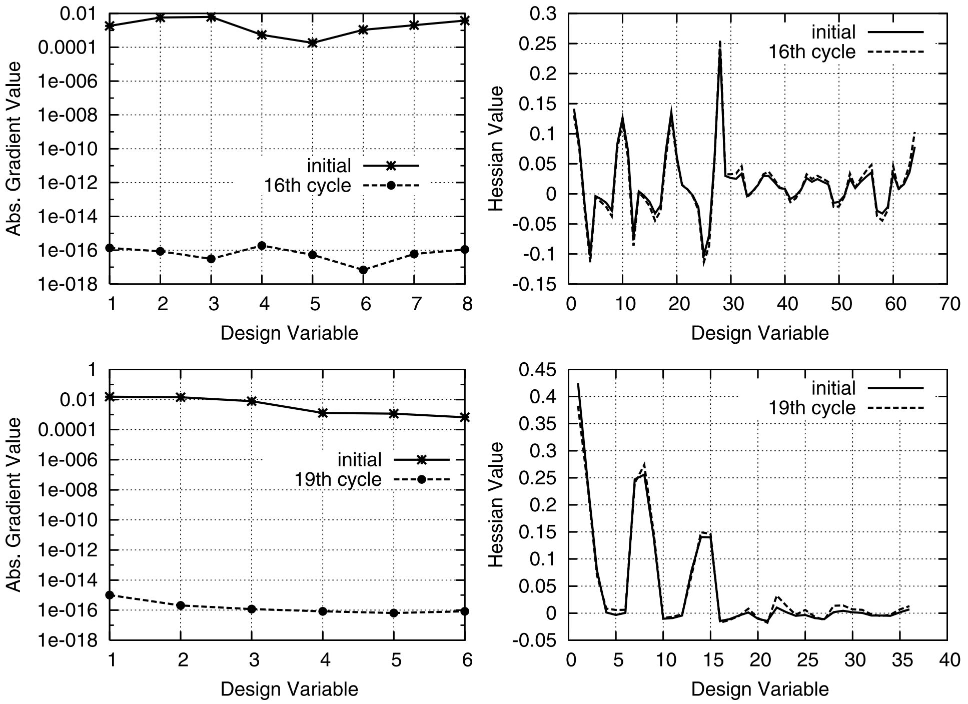


Fig. 26 Inverse design of a 2D turbine cascade, *top*, and a 2D compressor cascade, *bottom*. First (absolute values, semi-log scale) and second order sensitivities at the first and 30th optimization cycle, using the mixed Newton approach

(measured as either total pressure losses or entropy generation) have been used. The key points of the proposed methods, which include (a) the possibility of setting up continuous adjoint approaches without field integrals in the gradient or Hessian expressions, (b) the ability to compute exact Hessian matrices with either the discrete or the continuous adjoint approach, (c) a very efficient optimization method which starts by computing the exact Hessian of the objective function and continues by updating it as in the BFGS method, can be transferred to other applications based on the adjoint approach, as discussed in detail in this paper.

References

1. Lighthill MJ (1945) A new method of two-dimensional aerodynamic design. Aeronautical Research Council
2. McFadden GB (1979) An artificial viscosity method for the design of supercritical airfoils. New York University Report C00-3077-158
3. Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York
4. Michalewicz Z (1994) Genetic algorithms + data structures = evolution programs, 2nd edn. Springer, Berlin
5. Bäck T (1996) Evolutionary algorithms in theory and practice. Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, Oxford
6. Bäck T (1996) Evolutionary algorithms in theory and practice. Oxford University Press, Oxford
7. Bertsekas DP (1996) Constrained optimization and Lagrange multiplier methods, 1st edn. Athena Scientific, Nashua
8. Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific, Nashua
9. Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic, New York
10. Luenberger DG (2003) Linear and nonlinear programming, 2nd edn. Kluwer Academic, Dordrecht
11. Fletcher R (1988) Practical methods of optimization, 2nd edn. Wiley, New York
12. Jin Y (2003) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9:3–12
13. Wang GG, Shan S (2006) Review of metamodelling techniques in support of engineering design optimization. Trans ASME, J Mech Des 129(4):370–380
14. El-Beltagy MA, Nair PB, Keane AJ (1999) Metamodeling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations. In: GECCO99, genetic and evolutionary computation conference, Orlando, July 1999
15. Giannakoglou K (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Int Rev J Prog Aerosp Sci 38:43–76
16. Lions JL (1971) Optimal control of systems governed by partial differential equations. Springer, New York
17. Pironneau O (1974) On optimum design in fluid mechanics. J Fluid Mech 64:97–110
18. Pironneau O (1984) Optimal shape design for elliptic systems. Springer, New York
19. Jameson A (1988) Aerodynamic design via control theory. J Sci Comput 3:233–260
20. Jameson A, Reuther J (1994) Control theory based airfoil design using the Euler equations. AIAA Paper 94-4272
21. Jameson A (1995) Optimum aerodynamic design using CFD and control theory. AIAA Paper 95-1729
22. Jameson A, Pierce N, Martinelli L (1998) Optimum aerodynamic design using the Navier-Stokes equations. Theor Comput Fluid Dyn 10:213–237
23. Anderson WK, Nielsen E (2001) Sensitivity analysis for Navier-Stokes equations on unstructured grids using complex variables. AIAA J 39(31):56–63
24. Lyness JN, Moler CB (1967) Numerical differentiation of analytic functions. In: ACM 22nd national conference
25. Martins R, Kroo IM, Alonso J (2000) An automated method for sensitivity analysis using complex variables. AIAA Paper 2000-0689
26. Squire W, Trapp G (1998) Using complex variables to estimate derivatives of real functions. SIAM Rev 10(1):110–112
27. Newman JC, Anderson WK, Whitfield DL (1998) Multidisciplinary sensitivity derivatives using complex variables. Tech Rep MSSU-COE-ERC-98-08
28. Nielsen EJ, Kleb WL (2005) Efficient construction of discrete adjoint operators on unstructured grids by using complex variables. AIAA Paper 2005-0324
29. 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
30. Griewank A (1989) On automatic differentiation. In: Mathematical programming: recent developments and applications. Kluwer Academic, Dordrecht
31. Hovland P, Mohammadi B, Bischof C (1997) Automatic differentiation of Navier–Stokes computations. Technical Report MCS-P687-0997, Argonne National Laboratory
32. Juedes D (1991) A taxonomy of automatic differentiation tools. In: Automatic differentiation of algorithms: theory, implementation, and application. SIAM, Philadelphia, pp 315–329
33. Bischof C, Carle A, Corliss G, Griewank A, Hovland P (1991) ADIFOR Generating derivative codes from Fortran programs. Report CRPC-TR91185-S, Center for Research and Parallel Computation, Rice University
34. Giering R, Kaminski T (1998) Recipes for adjoint code construction. ACM Trans Math Softw 24:437–474
35. Berz M (1990) The DA precompiler DAFOR. Technical Report, Lawrence Berkeley National Laboratory, Berkeley, CA
36. Horwedel J (1991) GRESS a preprocessor for sensitivity studies of Fortran programs. AIAA Paper 91-005
37. Faure C (2005) An automatic differentiation platform: Odyssée. Future Gener Comput Syst 21(8):1391–1400
38. Stephens B (1991) Automatic differentiation as a general-purpose numerical tool. PhD thesis, School of Mathematics, University of Bristol, UK
39. Shiriaev D, Griewank A, Utke J (1996) A user guide to ADOL–F: automatic differentiation of Fortran codes. IOKOMO-04-1995
40. Rhodin A (1997) IMAS–integrated modeling and analysis system for the solution of optimal control problems. Comput Phys Commun 107:21–38
41. Christianson B (1992) Automatic Hessians by reverse accumulation. J Numer Anal 12:135–150
42. Bischof C, Roh L, Mauer-Oats A (1997) ADIC An extensible automatic differentiation tool for ANSI-C. Preprint ANL/MCS-P626-1196, Argonne National Laboratory
43. 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
44. Pierce NA, Giles MB (2000) An introduction to the adjoint approach to design. Flow Turbul Combust 65(3–4):393–415
45. Nadarajah S, Jameson A (2000) A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. AIAA Paper 2000-0667

46. Nadarajah S, Jameson A (2001) Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. AIAA Paper 2001-2530
47. Spalart PR, Allmaras SR (1994) A one-equation turbulence model for aerodynamic flows. Rech Aerosp (1):5–21
48. Roe P (1981) Approximate Riemann solvers, parameter vectors, and difference schemes. J Comput Phys 43:357–371
49. Papadimitriou DI, Giannakoglou KC (2007) A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. J Comput Fluids 36:325–341
50. Anderson WK, Venkatakrishnan V (1997) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA Paper 97-0643
51. Anderson WK, Venkatakrishnan V (1997) Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. Comput Fluids 28:443–480
52. Asouti VG, Zymaris AS, Papadimitriou DI, Giannakoglou KC (2008) Continuous and discrete adjoint approaches for aerodynamic shape optimization with low Mach number preconditioning. Int J Numer Methods Fluids 57:1485–1504
53. Arian E, Salas MD (1997) Admitting the inadmissible: adjoint formulation for incomplete cost functionals in aerodynamic optimization. NASA/CR-97-206269, ICASE Report No 97-69
54. Baysal O, Ghayour K (2001) Continuous adjoint sensitivities for optimization with general cost functionals on unstructured meshes. AIAA J 39(1)
55. Jameson A, Kim S (2003) Reduction of the adjoint gradient formula in the continous limit. AIAA Paper 2003-0040
56. Denton JD (1993) Loss mechanisms in turbomachines. ASME Paper 93-GT-435
57. Davies MRD, O'Donnell FK, Niven AJ (2000) Turbine blade entropy generation rate, part I: the boundary layer defined. ASME Paper 2000-GT-265
58. O'Donnell FK, Davies MRD (2000) Turbine blade entropy generation rate, part II: the measured loss. ASME Paper 2000-GT-266
59. Papadimitriou DI, Giannakoglou KC (2006) Compressor blade optimization using a continuous adjoint formulation. ASME TURBO EXPO, GT2006/90466, Barcelona
60. Papadimitriou DI, Giannakoglou KC (2007) Total pressure losses minimization in turbomachinery cascades, using a new continuous adjoint formulation. Proc Inst Mech Eng, Part A: J Power Energy 222(6):865–872
61. Papadimitriou DI, Zymaris AS, Giannakoglou KC (2005) Discrete and continuous adjoint formulations for turbomachinery applications. In: UROGEN 2005, international conference proceedings, Munich, September 2005
62. Papadimitriou DI, Giannakoglou KC (2006) A continuous adjoint method for the minimization of losses in cascade viscous flows. AIAA Paper 2006-0049
63. Kim SK, Alonso JJ, Jameson A (2000) Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method. AIAA Paper 2000-4741
64. Kim SK, Alonso JJ, Jameson A (2002) Design optimization of high-lift configurations using a viscous continuous adjoint method. AIAA Paper 2002-0844
65. Leoviriyakit K, Jameson A (2003) Aerodynamic shape optimization of wings including planform variations. AIAA Paper 2003-0210
66. Leoviriyakit K, Kim S, Jameson A (2003) Viscous aerodynamic shape optimization of wings including planform variations. AIAA Paper 2003-3498
67. Leoviriyakit K, Kim S, Jameson A (2004) Aero-structural wing planform optimization using the Navier-Stokes equations. AIAA Paper 2004-4479
68. Leoviriyakit K, Jameson A (2005) Multi-point wing planform optimization via control theory. AIAA Paper 2005-0450
69. Giles MB, Pierce NA (1997) Adjoint equations in CFD: duality, boundary conditions and solution behaviour. AIAA Paper 97-1850
70. Giles MB, Pierce NA (1998) On the properties of solutions of the adjoint Euler equations. In: 6th ICFD conference on numerical methods for fluid dynamics, Oxford, UK, 1998
71. Harbeck M, Jameson A (2005) Exploring the limits of transonic shock-free airfoil design. AIAA Paper 2005-1041
72. Reuther J, Alonso JJ, Rimlinger MJ, Jameson A (1999) Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. Comput Fluids 28:675–700
73. Nadarajah S, Kim SK, Jameson A, Alonso JJ (2002) Sonic boom reduction using an adjoint method for supersonic transport aircraft configuration. In: Symposium transsonicum IV, international union of theoretical and applied mechanics, September 2–6, 2002, DLR Gottingen, Germany
74. Nadarajah S, Jameson A, Alonso JJ (2002) Sonic boom reduction using an adjoint method for wing-body configurations in supersonic flow. AIAA Paper 2002-5547
75. Alonso JJ, Kroo IM, Jameson A (2002) Advanced algorithms for design and optimization of quiet supersonic platform. AIAA Paper 2002-0144
76. Nadarajah S, Jameson A, Alonso JJ (2002) An adjoint method for the calculation of remote sensitivities in supersonic flow. AIAA Paper 2002-0261
77. Taasan S, Kuruvila G, Salas MD (1992) Aerodynamic design and optimization in one-shot. AIAA Paper 91-005
78. Kuruvila G, Taasan S, Salas MD (1995) Airfoil design and optimization by the one-shot method. AIAA Paper 95-0478
79. Hazra SB (2004) An efficient method for aerodynamic shape optimization. AIAA Paper 2004-4628
80. Hazra S, Schulz V, Brezillon J, Gauger N (2005) Aerodynamic shape optimization using simultaneous pseudo-timestepping. J Comput Phys 204(1):46–64
81. Hazra SB, Schulz V (2006) Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. SIAM J Sci Comput 28(3):1078–1099
82. Held C, Dervieux A (2002) One-shot airfoil optimisation without adjoint. Comput Fluids 31:1015–1049
83. Dadone A, Grossman B (2000) Progressive optimization of inverse fluid dynamic design problems. Comput Fluids 29:1–32
84. Dadone A, Grossman B (2003) Fast convergence of inviscid fluid dynamic design problems. Comput Fluids 32:607–627
85. Soto O, Lohner R (2004) On the computation of flow sensitivities from boundary integrals. AIAA Paper 04-0112
86. Jameson A, Shankaran S, Martinelli L (2003) A continuous adjoint method for unstructured grids. AIAA Paper 2003-3955
87. Kim S, Leoviriyakit K, Jameson A (2003) Aerodynamic shape and planform optimization of wings using a viscous reduced adjoint gradient formula. In: 2nd MIT conference on computational fluid and solid mechanics, Cambridge, MA, June 17–20, 2003
88. Othmer C, de Villiers E, Weller HG (2007) Implementation of a continuous adjoint for topology optimization of ducted flows. AIAA Paper 2007-3947
89. Mohammadi B, Pironneau O (2001) Applied shape optimization for fluids. Clarendon, Oxford
90. Mohammadi B, Pironneau O (2004) Shape optimization in fluid mechanics. Annu Rev Fluid Mech 36:255–279
91. Soto O, Lohner R (2001) CFD shape optimization using an incomplete-gradient adjoint formulation. Int J Numer Methods Fluids 51:735–753
92. Soto O, Lohner R (2000) CFD optimization using an incomplete-gradient adjoint approach. AIAA Paper 00-0666

93. Soto O, Lohner R (2000) CFD shape optimization using an incomplete-gradient adjoint approach. In: ECCOMAS, Barcelona, September 2000
94. Soto O, Lohner R (2001) General methodologies for incompressible flow design problems. AIAA Paper 01-1061
95. Soto O, Lohner R (2002) A mixed adjoint formulation for incompressible rans problems. AIAA Paper 02-0451
96. Kim HJ, Sasaki D, Obayashi S, Nakahashi K (2001) Aerodynamic optimization of supersonic transport wing using unstructured adjoint method. AIAA J 39(6)
97. Nielsen EJ, Park MA (2005) Using an adjoint approach to eliminate mesh sensitivities in computational design. AIAA Paper 2005-0491
98. Mavriplis DJ (2005) Formulation and multigrid solution of the discrete adjoint for optimization problems on unstructured meshes. AIAA Paper
99. Mavriplis DJ (2006) A discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. AIAA Paper
100. Nielsen EJ, Anderson WK (2002) Recent improvements in aerodynamic design optimization on unstructured meshes. AIAA J 40(6):1155–1163
101. Nielsen EJ, Anderson WK (2001) Recent improvements in aerodynamic design optimization on unstructured meshes. AIAA Paper 2001-0596
102. Elliot J, Peraire J (1996) Aerodynamic design using unstructured meshes. AIAA Paper 96-1941
103. Elliot J, Peraire J (1997) Aerodynamic optimization using unstructured meshes with viscous effects. AIAA Paper 97-1849
104. Elliot J, Peraire J (1997) Practical 3d aerodynamic design and optimization using unstructured meshes. AIAA J 35(9):1479–1485
105. Pulliam TH, Nemec M, Holst TL, Zingg DW (2003) Comparison of genetic and adjoint methods for multi-objective viscous airfoil optimizations. AIAA Paper 2003-0298
106. Elliot J, Peraire J (1998) Constrained, multipoint shape optimisation for complex 3d configurations. Aeronaut J 102(1017):365–376
107. Martins JRRA, Alonso JJ, Reuther JJ (2005) A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. Optim Eng 6(1):33–62
108. Leoviriyakit K, Jameson A (2004) Case studies in aero-structural wing planform and section optimization. AIAA Paper 2004-5372
109. Leoviriyakit K, Jameson A (2004) Aero-structural wing planform optimization. AIAA Paper 2004-0029
110. Nadarajah S, Jameson A (2002) Optimal control of unsteady flows using a time accurate method. AIAA Paper 2002-5436
111. Nadarajah S, McMullen M, Jameson A (2002) Non-linear frequency domain based optimum shape design for unsteady three-dimensional flow. AIAA Paper 2002-2838
112. Nadarajah S, Jameson A (2002) Optimum shape design for unsteady three-dimensional viscous flows using a non-linear frequency domain method. AIAA Paper 2002-2838
113. Campobasso MS, Duta MC, Giles MB (2001) Adjoint methods for turbomachinery design. In: ISOABE conference, 2001
114. Duta MC, Giles MB, Campobasso MS (2002) The harmonic adjoint approach to unsteady turbomachinery design. Int J Numer Meth Fluids 40(3–4):323–332
115. Campobasso MS, Duta MC, Giles MB (2003) Adjoint calculation of sensitivities of turbomachinery objective functions. AIAA J Propuls Power 19(4)
116. Giannakoglou KC, Papadimitriou DI (2006) Formulation and application of the continuous adjoint method in aerodynamics and turbomachinery. Von-Karman institute lecture series
117. Anderson WK, Bonhaus DL (1997) Aerodynamic design on unstructured grids for turbulent flows. NASA Technical Memorandum
118. Anderson WK, Bonhaus DL (1999) Airfoil design on unstructured grids for turbulent flows. AIAA J 37(2):185–191
119. Nielsen EJ, Lu J, Park MA, Darmofal DL (2004) An implicit exact dual adjoint solution method for turbulent flows on unstructured grids. Comput Fluids 33:1131–1155
120. Sherman LL, Taylor III AC, Green LL, Newman PA, Hou GW, Korivi VM (1996) First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. J Comput Phys 129:307–331
121. Papadimitriou DI, Giannakoglou KC (2007) Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems. Int J Numer Methods Fluids 56:1929–1943
122. Papadimitriou DI, Giannakoglou KC (2007) Computation of the Hessian matrix in aerodynamic inverse design using continuous adjoint formulations. Comput Fluids 37:1029–1039
123. Tortorelli D, Michaleris P (1994) Design sensitivity analysis: overview and review. Inverse Probl Eng 1(1):71–105
124. Hou GW, Sheen J (1993) Numerical methods for second-order shape sensitivity analysis with applications to heat conduction problems. Int J Numer Methods Eng 36:417–435
125. Le Dimet FX, Navon IM, Daescu DN (2002) Second-order information in data assimilation. Mon Weather Rev 130(3):629–648
126. Veerse F, Auroux D, Fisher M (2000) Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. Optim Eng 1:323–339
127. Daescu DN, Navon IM (2003) An analysis of a hybrid optimization method for variational data assimilation. Int J Comput Fluid Dyn 17(4):299–306
128. Arian E, Taasan S (1999) Analysis of the Hessian for aerodynamic optimization: inviscid flow. Comput Fluids 28(7):853–877