CFD and OPTIMIZATION 2011
An ECCOMAS Thematic Conference
23-25 May 2011, Antalya TURKEY

DISCRETE AND CONTINUOUS ADJOINT METHODS IN AERODYNAMIC ROBUST DESIGN PROBLEMS

Evangelos M. Papoutsis-Kiachagias*, Dimitrios I. Papadimitriou* and Kyriakos C. Giannakoglou*

* National Technical University of Athens, Greece,
School of Mechanical Engineering,
Lab. of Thermal Turbomachines,
Parallel CFD & Optimization Unit.
e-mail: vaggelisp@gmail.com, dpapadim@mail.ntua.gr, kgianna@central.ntua.gr

Key words: Aerodynamic robust design, adjoint method, high-order sensitivity analysis, method of moments

Abstract. *The implementation of the adjoint formulation for the solution of robust design problems in aerodynamic shape optimization, with affordable CPU cost, is presented. In robust design, the minimization of both the mean value μ_F and standard deviation σ_F of the target objective function F, should take into account the so-called environmental or robust variables c_i, $i=1,M$, is required. According to the second-order, second-moment approach, μ_F and σ_F are expressed in terms of the first and second-order derivatives of F with respect to c_i. To perform a gradient-based optimization, μ_F and σ_F must, then, be differentiated with respect to the shape controlling (design) variables b_q, $q=1,N$. Thus, methods for computing up to third-order mixed sensitivities, such as $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$, must be devised. In this paper, this is carried out through the appropriate combination of direct-differentiation and adjoint approaches. The present method is used to perform the robust design of quasi-1D and 2D shapes. Both the discrete and continuous adjoint approaches are presented, having CPU cost that scales with M and is independent of N. The proposed method, in either discrete or continuous form, is the best choice for solving robust design problems with much fewer environmental variables than the design ones ($M << N$).*

1 INTRODUCTION

In aerodynamics, the adjoint variable (AV) method, since its first appearance, [1], has been used in several shape optimization problems in fluid mechanics. Using the AV method, the gradient of any objective function F that quantifies the performance of a given configuration, with respect to (w.r.t.) the design or control variables ($b_q,\ q = 1, N$) can be computed. In constrast to other rival techniques such as finite differences, direct differentiation (DD, which is the equivalent of the tangent linear mode in automatic differentiation, often differentiated by hand, as in the present paper) or the complex variables' method, the CPU cost of the AV method does not scale with N. Thus, in aerodynamic shape optimization problems, the use of the AV method for the computation of the gradient of F is suitable. One may use either the discrete AV method, [2, 3], where the discrete adjoint equations are derived directly from the discretized flow equations or the continuous one [4, 5, 6, 7], where the adjoint PDEs are first derived from the flow PDEs and, then, discretized. However, regarding the computation of higher-order derivatives (such as the Hessian of F w.r.t. b_q) things are different. In previous publications, [8, 9, 10, 11, 12, 13, 14, 15, 16], it has been proved that the computational cost is minimum if the so-called DD-AV method is used. In the DD-AV method, the first-order derivatives are computed using DD and, then, the AV method provides the Hessian of F. In the literature, DD-AV schemes for computing $\frac{\delta^2 F}{\delta b_q \delta b_p}$, based on either the discrete, [12, 13, 14], or the continuous adjoint approach, [14, 15, 16], can be found. Their CPU cost, for computing the exact Hessian, scales with N and so does the CPU cost per Newton cycle.

Previous comments are all related to shape optimization problems at given operating conditions and without geometry imprecisions, etc, i.e. design problems that must be solved in "fixed environment". To account for designs with acceptable performance even if their environment changes, the so-called *robust design* methods have been developed, [17, 18, 19, 20]. These rely on a new appropriate objective function (which will be denoted by $\widehat{F}$ in this paper; see section 2), defined by quantifying the way F changes if the M environmental variables ($c_i,\ i = 1, M$) vary, based on a known probability density function. Robust design methods are found to outperform multi-point shape optimization methods [21, 22, 23], where the shape is optimized for a number of distinct operating points. In robust design methods, the definition of the objective function $\widehat{F}$, according to the method of moments, [17, 18, 19, 20], requires the derivatives of F w.r.t. c_i. These express the dependency of F on the environment and are written in terms of the mean value μ_F and standard deviation σ_F of F, using first- or second-order Taylor expansions.

Papers [17, 18, 19, 20] are restricted to the computation of derivatives of F w.r.t. c_i only, without considering the computation of the gradient of moments w.r.t. b_q, to support a gradient-based solution to the robust design problem. The computation of the latter and its use in gradient-based robust design of aerodynamic shapes is the main objective of the present paper. Let us also make clear that the present paper does not make use of either surrogate models, [24, 25] or numerical integrations, [26, 27] or, even, evolutionary

algorithms, [28], which are all alternative ways to cope with the robust design problem.

The structure of this paper is as follows: In section 2, the method of moments is presented and the required sensitivities of the objective function for the application of a gradient-based SOSM approach are identified. In section 3, their computations, based on both discrete and continuous DD and AV methods, are presented. Finally, in section 4, the developed methods are validated by paying attention to the accuracy of the computed sensitivities and, then, used for the minimization of the mean value and standard deviation of F.

2 ROBUST DESIGN: THE SECOND-MOMENT APPROACH

One of the possible ways to cope with robust design problems is to minimize an objective function expressed in terms of the mean value of F (first statistical moment),

$$\mu_F(\mathbf{b},\mathbf{c}) = \int F g(\mathbf{c}) d\mathbf{c} \tag{1}$$

and its variance (second statistical moment),

$$\sigma_F{}^2(\mathbf{b},\mathbf{c}) = \int (F-\mu_F)^2 g(\mathbf{c}) d\mathbf{c} \tag{2}$$

where $g(\mathbf{c})$ is the probability density function. A method that minimizes both μ_F and σ_F is referred to as a second-moment one. Neglecting the minimization of σ_F gives rise to the so-called first-moment methods. The integrals in eqs. 1 and 2 can be computed using either the exact integration of a function approximating the integrands (by means of surrogate models or the method of moments) or the numerical integration of a limited number of exactly evaluated points.

Assuming a symmetric distribution for c_i, it can be shown that

$$\begin{aligned} \mu_F &\simeq F + \frac{1}{2}\left[\frac{\delta^2 F}{\delta c_i^2}\right]_{\overline{\mathbf{c}}} \sigma_i^2 \\ \sigma_F^2 &\simeq \left[\frac{\delta F}{\delta c_i}\right]_{\overline{\mathbf{c}}}^2 \sigma_i^2 + \frac{1}{2}\left[\frac{\delta^2 F}{\delta c_i \delta c_j}\right]_{\overline{\mathbf{c}}}^2 \sigma_i^2 \sigma_j^2 \end{aligned} \tag{3}$$

Regarding the minimization procedure itself, stochastic methods such as evolutionary algorithms (EAs) are well suited for these problems since these are gradient-free. However, EAs require an excessive number of evaluations. Even if efficient ways to cut down the number of evaluations required by an EA are now available, the overall cost remains high. Gradient-based methods (GBMs), such as steepest descent or quasi-Newton schemes based on Hessian matrix approximations can be used instead. In GBMs, a scalar function

$$\widehat{F}(\mathbf{b},\mathbf{c}) = w_1 \mu_F + w_2 \sigma_F \tag{4}$$

is minimized, where w_1 and w_2 are user-defined weights. The gradient components $\frac{d\widehat{F}}{db_q}$ can be obtained by differentiating eq. 4 w.r.t. b_q, which yields

$$\frac{\delta\widehat{F}}{\delta b_q} = w_1\left(\frac{\delta F}{\delta b_q} + \frac{1}{2}\frac{\delta^3 F}{\delta c_i^2 \delta b_q}\sigma_i^2\right) + w_2 \frac{2\frac{\delta F}{\delta c_i}\frac{\delta^2 F}{\delta c_i \delta b_q}\sigma_i^2 + \frac{\delta^2 F}{\delta c_i \delta c_j}\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}\sigma_i^2\sigma_j^2}{2\sqrt{\left[\frac{\delta F}{\delta c_i}\right]^2\sigma_i^2 + \frac{1}{2}\left[\frac{\delta^2 F}{\delta c_i \delta c_j}\right]^2\sigma_i^2\sigma_j^2}} \tag{5}$$

So, the calculation of $\frac{\delta\widehat{F}}{\delta b_q}$ requires the computation of up to third-order mixed sensitivities w.r.t. c_i and b_q. In this paper, the steepest descent algorithm

$$b_q{}^{n+1} = b_q{}^n - \eta\frac{\delta\widehat{F}}{\delta b_q} \tag{6}$$

is used to update the design variable values and, through them, the aerodynamic shape; quasi-Newton methods can be certainly used instead. This paper is dealing with inverse design problems, where F is defined as

$$F = \frac{1}{2}\int_{S_w}(p - p_{tar})^2 dS \tag{7}$$

where p is the pressure, p_{tar} is the target pressure distribution and S_w denotes the solid wall. It is a matter of just a few modifications for using the same method with other objective functions.

3 COMPUTATION OF SENSITIVITY DERIVATIVES

3.1 The AV Approach for the Computation of $\frac{\delta F}{\delta b_q}$

In discrete adjoint, starting point is the discrete form of the system of flow PDEs denoted by $R_{k,d} = 0$, where the first subscript is the equation number ($k \in [1, n_e]$; n_e is the number of governing equations per node) and the second one denotes the grid node ($d \in [1, n_p]$ for discretization on a grid with n_p nodes). Let $U_{k,d}$ stand for the flow variables (same notation). After defining the augmented objective function F_{aug} in terms of the nodal adjoint variables $\mathcal{N}_{n,a}$ (subscripts as in $U_{k,d}$), its sensitivities w.r.t. the design variables b_q ($q \in [1, N]$) become

$$\frac{\delta F_{aug}}{\delta b_q} = \frac{\partial F}{\partial b_q} + \frac{\partial F}{\partial U_{k,d}}\frac{\delta U_{k,d}}{\delta b_q} + \mathcal{N}_{n,a}\left(\frac{\partial R_{n,a}}{\partial b_q} + \frac{\partial R_{n,a}}{\partial U_{k,d}}\frac{\delta U_{k,d}}{\delta b_q}\right) \tag{8}$$

The summation rule applies for repeated indices. In discrete adjoint, symbols $\frac{\delta}{\delta b_q}$ and $\frac{d}{db_q}$ are used indifferently; $\frac{\delta}{\delta b_q}$ is, however, preferably used for the closest similarity with the continuous approach. Eq. 8 is made independent of $\frac{\delta U_{k,d}}{\delta b_q}$ by satisfying the system of the discrete adjoint equations

$$\frac{\partial F}{\partial U_{k,d}} + \mathcal{N}_{n,a}\frac{\partial R_{n,a}}{\partial U_{k,d}} = 0 \tag{9}$$

Once the $\mathcal{N}_{n,a}$ values have been computed, at the cost of solving an "equivalent flow system" (EFS), the gradient of F w.r.t. the design variables results from

$$\frac{\delta F}{\delta b_q} = \frac{\delta F_{aug}}{\delta b_q} = \frac{\partial F}{\partial b_q} + \mathcal{N}_{n,a}\frac{\partial R_{n,a}}{\partial b_q} \tag{10}$$

Throughout this paper, EFS is considered to be the CPU time unit. Practically, solving an optimization problem at the cost of T EFS is as if the system of flow PDEs is solved T times.

An alternative way to compute $\frac{\delta F}{\delta b_q}$, with practically the same CPU cost (1 EFS), is by using the continuous adjoint method. For instance, let us start from the steady-state 2D Euler equations for a compressible fluid flow, as in the second example presented below; in conservative form, this reads

$$\frac{\partial f_{nk}}{\partial x_k} = 0 \tag{11}$$

where $k = 1, 2$, $n = 1, 4$. The inviscid fluxes f_{nk} are

$$[f_{1k}, f_{2k}, f_{3k}, f_{4k}] = [\rho u_k, \rho u_k u_1 + p\delta_{k1}, \rho u_k u_2 + p\delta_{k2}, u_k(E + p)]$$

where ρ, p, u_k and E stand for the density, pressure, Cartesian velocity components and total energy per unit volume, respectively. In conformity to eqs. 11, the array of conservative flow variables yields

$$[U_1, U_2, U_3, U_4] = [\rho, \rho u_1, \rho u_2, E]$$

Similar to eq. 8, in continuous form however, the sensitivities of F_{aug} w.r.t. b_q become

$$\frac{\delta F_{aug}}{\delta b_q} = \frac{\delta F}{\delta b_q} + \frac{\delta}{\delta b_q}\int_\Omega \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_k} d\Omega \tag{12}$$

where Ω is the flow domain, $(\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4)$ are the adjoint functions and

$$\frac{\delta F}{\delta b_q} = \int_{S_w} (p - p_{tar})\frac{\delta p}{\delta b_q} dS + \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_q} \tag{13}$$

The differentiation under the integral sign of the last term in eq. 12, based on the Leibniz integral rule, gives

$$\frac{\delta F_{aug}}{\delta b_q} = \frac{\delta F}{\delta b_q} + \int_\Omega \mathcal{N}_n \frac{\partial}{\partial b_q}\left(\frac{\partial f_{nk}}{\partial x_k}\right) d\Omega + \int_S \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_k}\frac{\delta x_l}{\delta b_q} n_l dS \tag{14}$$

where S is the boundary of Ω and n_k are the components of the outward unit vector. Since partial derivatives permute, the first integral in eq. 14 can be integrated by parts,

$$\int_\Omega \mathcal{N}_n \frac{\partial}{\partial b_q}\left(\frac{\partial f_{nk}}{\partial x_k}\right) d\Omega = -\int_\Omega \frac{\partial \mathcal{N}_n}{\partial x_k}\frac{\partial f_{nk}}{\partial b_q} d\Omega + \int_S \mathcal{N}_n \frac{\partial f_{nk}}{\partial b_q} n_k dS \tag{15}$$

Using the no-penetration condition, the second integral on the r.h.s. of eq. 15, taken only along S_w, yields

$$\begin{aligned}\int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial b_q} n_k dS &= \int_{S_w} \mathcal{N}_{k+1} n_k \frac{\delta p}{\delta b_q} dS \\ &+ \int_{S_w} \left(\mathcal{N}_{k+1} p - \mathcal{N}_n f_{nk}\right) \frac{\delta \left(n_k dS\right)}{\delta b_q} - \int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_q} n_k dS \end{aligned} \tag{16}$$

A term-by-term development as in [14, 15], yields

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_q} &= \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_q} + \int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta b_q} dS \\ &- \int_\Omega A_{nmk} \frac{\partial \mathcal{N}_n}{\partial x_k} \frac{\partial U_m}{\partial b_q} d\Omega + \int_{S_{I,O}} \mathcal{N}_n \frac{\partial f_{nk}}{\partial b_q} n_k dS + \int_{S_w} \mathcal{N}_{k+1} n_k \frac{\delta p}{\delta b_q} dS \\ &+ \int_{S_w} \left(\mathcal{N}_{k+1} p - \mathcal{N}_n f_{nk}\right) \frac{\delta \left(n_k dS\right)}{\delta b_q} - \int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_q} n_k dS \\ &+ \int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_k} \frac{\delta x_l}{\delta b_q} n_l dS \end{aligned} \tag{17}$$

where $A_{nmk} = \frac{\partial f_{nk}}{\partial U_m}$ $(n = 1, 4, \quad m = 1, 4, \quad k = 1, 2)$ is the Jacobian matrix of the inviscid fluxes. From eq. 17, dependencies on $\frac{\partial U_m}{\partial b_q}$ are eliminated by formulating the system of 4 $(m = 1, 4)$ field adjoint PDEs

$$-A_{nmk} \frac{\partial \mathcal{N}_n}{\partial x_k} = 0 \tag{18}$$

to be discretized and solved along with appropriate boundary conditions. For instance, the inlet/outlet (along $S_{I,O}$) boundary conditions are derived by eliminating all integrals depending on $\frac{\delta U_m}{\delta b_q}$ from eq. 17. Along S_w, the following condition

$$p - p_{tar} + \mathcal{N}_{k+1} n_k = 0 \tag{19}$$

must be satisfied; eq. 19 mimics the no-penetration condition $u_k n_k = U_{k+1} n_k = 0$ for the primal velocity. The remaining terms in eq. 17 provide the gradient of F as follows

$$\begin{aligned}\frac{\delta F}{\delta b_q} &= \frac{1}{2}\int_{S_w} (p - p_{tar})^2 \frac{\delta (dS)}{\delta b_q} + \int_{S_w} \left(\mathcal{N}_{k+1} p - \mathcal{N}_n f_{nk}\right) \frac{\delta \left(n_k dS\right)}{\delta b_q} \\ &- \int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_q} n_k dS + \int_{S_w} \mathcal{N}_n \frac{\partial f_{nk}}{\partial x_k} \frac{\delta x_l}{\delta b_q} n_l dS \end{aligned} \tag{20}$$

Note that, along S_w, which is affected by variations in the design variables, the total $(\frac{\delta \Phi}{\delta b_q})$ and local $(\frac{\partial \Phi}{\partial b_q})$ sensitivities of any function Φ are related as follows

$$\frac{\delta \Phi}{\delta b_q} = \frac{\partial \Phi}{\partial b_q} + \frac{\partial \Phi}{\partial x_k} \frac{\delta x_k}{\delta b_q} \tag{21}$$

3.2 Computation of $\frac{\delta F}{\delta c_i}$ and $\frac{\delta^2 F}{\delta c_i \delta c_j}$ using DD

The computation of $\frac{\delta F}{\delta c_i}$ is based on the general expression

$$\frac{\delta \Phi}{\delta \gamma_i} = \frac{\partial \Phi}{\partial \gamma_i} + \frac{\partial \Phi}{\partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta \gamma_i} \tag{22}$$

written for $\Phi = F$ and $\gamma_i = c_i$. This requires the knowledge of $\frac{\delta U_{k,d}}{\delta c_i}$ which can be computed using the DD$_c$ of the flow equations ($\frac{\delta R_{n,a}}{\delta c_i} = 0$). The CPU cost for computing all $\frac{\delta U_{k,d}}{\delta c_i}$, $i = 1, M$, fields is equal to M EFS.

The second-order derivatives w.r.t. $\mathbf{c}$ are computed using equation

$$\begin{aligned} \frac{\delta^2 \Phi}{\delta \gamma_i \delta \gamma_j} &= \frac{\partial^2 \Phi}{\partial \gamma_i \partial \gamma_j} + \frac{\partial^2 \Phi}{\partial \gamma_i \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta \gamma_j} + \frac{\partial^2 \Phi}{\partial \gamma_j \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta \gamma_i} \\ &+ \frac{\partial^2 \Phi}{\partial U_{k,d} \partial U_{m,e}} \cdot \frac{\delta U_{k,d}}{\delta \gamma_i} \cdot \frac{\delta U_{m,e}}{\delta \gamma_j} + \frac{\partial \Phi}{\partial U_{k,d}} \cdot \frac{\delta^2 U_{k,d}}{\delta \gamma_i \delta \gamma_j} \end{aligned} \tag{23}$$

for $\Phi = F$, $\gamma_i = c_i$, $\gamma_j = c_j$. Apart from the already computed $\frac{\delta U_{k,d}}{\delta c_i}$ fields, eq. 23 also requires $\frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j}$ which, based on eq. 23 for $\Phi = R_{n,a}$, $\gamma_i = c_i$, $\gamma_j = c_j$ yields $\frac{M(M+1)}{2}$ systems of equations $\frac{\delta^2 R_{n,a}}{\delta c_i \delta c_j} = 0$ to be solved at the cost of $\frac{M(M+1)}{2}$ EFS. This is referred to as the DD$_c$-DD$_c$ approach.

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

$$\frac{\partial}{\partial x_k} \left(A_{nmk} \frac{\delta U_m}{\delta c_i} \right) = 0 \tag{24}$$

along with appropriate boundary conditions. For instance, along S_w, the no–penetration condition yields

$$\frac{\delta u_k}{\delta c_i} n_k = \frac{\delta U_{k+1}}{\delta c_i} n_k = 0 \tag{25}$$

which is an equivalent no-penetration condition for $\frac{\delta u_k}{\delta c_i}$.

For the $\frac{M(M+1)}{2}$ systems of equations, to be solved for $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$, $i = 1, M$, $j = 1, M$, eq. 24 is differentiated once more to give

$$\frac{\partial}{\partial x_k} \left(A_{nmk} \frac{\delta^2 U_m}{\delta c_i \delta c_j} + \frac{\delta A_{nmk}}{\delta c_j} \frac{\delta U_m}{\delta c_i} \right) = 0 \tag{26}$$

Finally, the first- and second-order sensitivities of F w.r.t. the environmental variables are given by

$$\frac{\delta F}{\delta c_i} = \int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta c_i} dS \tag{27}$$

and

$$\frac{\delta^2 F}{\delta c_i \delta c_j} = \int_{S_w} \left[\frac{\delta p}{\delta c_i} \frac{\delta p}{\delta c_j} + (p - p_{tar}) \frac{\delta^2 p}{\delta c_i \delta c_j} \right] dS \tag{28}$$

and can be computed from the known $\frac{\delta U_m}{\delta c_i}$ and $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$ fields. Similarities between the discrete and continuous approaches are evident.

3.3 Computation of $\frac{\delta^2 F}{\delta c_i \delta b_q}$ and $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$

Previously computed derivatives of F w.r.t. the environmental variables should be differentiated w.r.t. the design variables. To this end, in the discrete approach, a different augmented function F'_{aug} is defined and its derivatives are given by

$$\frac{\delta^2 F'_{aug}}{\delta c_i \delta b_q} = \frac{\delta^2 F}{\delta c_i \delta b_q} + \mathcal{L}^i_{n,a} \frac{\delta R_{n,a}}{\delta b_q} + \mathcal{N}_{n,a} \frac{\delta^2 R_{n,a}}{\delta c_i \delta b_q} \tag{29}$$

by introducing appropriate adjoint variables $\mathcal{L}^i_{n,a}$ and $\mathcal{N}_{n,a}$. The same symbol $\mathcal{N}_{n,a}$ is used as before, on purpose, since, as it can easily be shown, the equations for $\mathcal{N}_{n,a}$ are the ones derived above (eqs. 9, 18) Based on the "standard" way for deriving adjoint equations, $\mathcal{L}^i_{n,a}$ $(i = 1, M;\ n = 1, n_e;\ a = 1, n_p)$ result from the solution of

$$\frac{\partial^2 F}{\partial c_i \partial U_{k,d}} + \frac{\partial^2 F}{\partial U_{k,d} \partial U_{m,e}} \cdot \frac{\delta U_{m,e}}{\delta c_i} + \mathcal{L}^i_{n,a} \frac{\partial R_{n,a}}{\partial U_{k,d}} + \mathcal{N}_{n,a} \left(\frac{\partial^2 R_{n,a}}{\partial c_i \partial U_{k,d}} + \frac{\partial^2 R_{n,a}}{\partial U_{k,d} \partial U_{m,e}} \cdot \frac{\delta U_{m,e}}{\delta c_i} \right) = 0 \tag{30}$$

The computation of $\mathcal{L}^i_{n,a}$, based on eqs. 30, costs M EFS and makes use of the already computed $\mathcal{N}_{n,a}$ fields.

Finally, $\frac{\delta^2 F}{\delta c_i \delta b_q}$ are computed from the following equation

$$\begin{aligned} \frac{\delta^2 F}{\delta c_i \delta b_q} &= \frac{\partial^2 F}{\partial c_i \partial b_l} + \mathcal{N}_{n,a} \frac{\partial^2 R_{n,a}}{\partial c_i \partial b_q} + \frac{\partial^2 F}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_i} + \mathcal{N}_{n,a} \frac{\partial^2 R_{n,a}}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \\ &+ \mathcal{L}^i_{n,a} \frac{\partial R_{n,a}}{\partial b_q} \end{aligned} \tag{31}$$

To compute $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$, one may start from the following equation (defined by differentiating eq. 23 w.r.t. γ_l)

$$\frac{\delta^3 \Phi}{\delta \gamma_i \delta \gamma_j \delta \gamma_l} = \frac{\partial^3 \Phi}{\partial \gamma_i \partial \gamma_j \partial \gamma_l}$$

$$
\begin{aligned}
&+ \frac{\partial^3\Phi}{\partial\gamma_i\partial\gamma_j\partial U_{k,d}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_l} + \frac{\partial^3\Phi}{\partial\gamma_i\partial\gamma_l\partial U_{k,d}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_j} + \frac{\partial^3\Phi}{\partial\gamma_j\partial\gamma_l\partial U_{k,d}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_i} \\
&+ \frac{\partial^3\Phi}{\partial\gamma_i\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_j}\frac{\delta U_{m,e}}{\delta\gamma_l} + \frac{\partial^3\Phi}{\partial\gamma_j\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_i}\frac{\delta U_{m,e}}{\delta\gamma_l} \\
&+ \frac{\partial^3\Phi}{\partial\gamma_l\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_i}\frac{\delta U_{m,e}}{\delta\gamma_j} + \frac{\partial^3\Phi}{\partial U_{k,d}\partial U_{m,e}\partial U_{r,g}}\cdot\frac{\delta U_{k,d}}{\delta\gamma_i}\frac{\delta U_{m,e}}{\delta\gamma_j}\frac{\delta U_{r,g}}{\delta\gamma_l} \\
&+ \frac{\partial^2\Phi}{\partial\gamma_i\partial U_{k,d}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_j\delta\gamma_l} + \frac{\partial^2\Phi}{\partial\gamma_j\partial U_{k,d}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_i\delta\gamma_l} + \frac{\partial^2\Phi}{\partial\gamma_l\partial U_{k,d}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_i\delta\gamma_j} \\
&+ \frac{\partial^2\Phi}{\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_i\delta\gamma_j}\cdot\frac{\delta U_{m,e}}{\delta\gamma_l} + \frac{\partial^2\Phi}{\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_j\delta\gamma_l}\cdot\frac{\delta U_{m,e}}{\delta\gamma_i} \\
&+ \frac{\partial^2\Phi}{\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta^2 U_{k,d}}{\delta\gamma_i\delta\gamma_l}\cdot\frac{\delta U_{m,e}}{\delta\gamma_j} + \frac{\partial\Phi}{\partial U_{k,d}}\cdot\frac{\delta^3 U_{k,d}}{\delta\gamma_i\delta\gamma_j\delta\gamma_l}
\end{aligned}
\tag{32}
$$

which, for $\Phi = F,\ \gamma_i = c_i,\ \gamma_j = c_j,\ \gamma_l = b_q$, gives

$$
\begin{aligned}
\frac{\delta^3 F}{\delta c_i\delta c_j\delta b_q} &= \frac{\partial^3 F}{\partial c_i\partial c_j\partial b_q} + \frac{\partial^3 F}{\partial c_i\partial b_q\partial U_{k,d}}\cdot\frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 F}{\partial c_j\partial b_q\partial U_{k,d}}\cdot\frac{\delta U_{k,d}}{\delta c_i} \\
&+ \frac{\partial^3 F}{\partial b_q\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{k,d}}{\delta c_i}\frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 F}{\partial b_q\partial U_{k,d}}\cdot\frac{\delta^2 U_{k,d}}{\delta c_i\delta c_j} \\
&+ \left(\frac{\partial^3 F}{\partial c_i\partial c_j\partial U_{r,g}} + \frac{\partial^3 F}{\partial c_i\partial U_{k,d}\partial U_{r,g}}\cdot\frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 F}{\partial c_j\partial U_{k,d}\partial U_{r,g}}\cdot\frac{\delta U_{k,d}}{\delta c_i}\right. \\
&+ \left.\frac{\partial^3 F}{\partial U_{k,d}\partial U_{m,e}\partial U_{r,g}}\cdot\frac{\delta U_{k,d}}{\delta c_i}\cdot\frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 F}{\partial U_{k,d}\partial U_{r,g}}\cdot\frac{\delta^2 U_{k,d}}{\delta c_i\delta c_j}\right)\cdot\frac{\delta U_{r,g}}{\delta b_q} \\
&+ \left(\frac{\partial^2 F}{\partial c_j\partial U_{k,d}} + \frac{\partial^2 F}{\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{m,e}}{\delta c_j}\right)\cdot\frac{\delta^2 U_{k,d}}{\delta c_i\delta b_q} \\
&+ \left(\frac{\partial^2 F}{\partial c_i\partial U_{k,d}} + \frac{\partial^2 F}{\partial U_{k,d}\partial U_{m,e}}\cdot\frac{\delta U_{m,e}}{\delta c_i}\right)\cdot\frac{\delta^2 U_{k,d}}{\delta c_j\delta b_q} \\
&+ \frac{\partial F}{\partial U_{k,d}}\cdot\frac{\delta^3 U_{k,d}}{\delta c_i\delta c_j\delta b_q}
\end{aligned}
\tag{33}
$$

Based on eq. 33,

$$
\frac{\delta U_{k,d}}{\delta c_i},\ \frac{\delta^2 U_{k,d}}{\delta c_i\delta c_j},\ \frac{\delta U_{k,d}}{\delta b_q},\ \frac{\delta^2 U_{k,d}}{\delta c_i\delta b_q},\ \frac{\delta^2 U_{k,d}}{\delta c_j\delta b_q},\ \frac{\delta^3 U_{k,d}}{\delta c_i\delta c_j\delta b_q} \tag{34}
$$

need to be known. The computation of the first two sets of derivatives by means of DD_c and DD_c-DD_c has already been presented. For the derivatives w.r.t. b_q, the AV_b method should preferably be used. A new augmented function F''_{aug} is defined with its third-order derivatives given by

$$
\frac{\delta^3 F''_{aug}}{\delta c_i\delta c_j\delta b_q} = \frac{\delta^3 F}{\delta c_i\delta c_j\delta b_q} + \mathcal{K}^{i,j}_{n,a}\frac{\delta R_{n,a}}{\delta b_q} + \mathcal{L}^{j}_{n,a}\frac{\delta^2 R_{n,a}}{\delta c_i\delta b_q} + \mathcal{M}^{i}_{n,a}\frac{\delta^2 R_{n,a}}{\delta c_j\delta b_q} + \mathcal{N}_{n,a}\frac{\delta^3 R_{n,a}}{\delta c_i\delta c_j\delta b_q}
$$

(35)

where $\mathcal{K}^{i,j}_{n,a}$, $\mathcal{L}^{j}_{n,a}$, $\mathcal{M}^{i}_{n,a}$ and $\mathcal{N}_{n,a}$, ($i = 1, M$; $j = 1, M$; $n = 1, n_e$; $a = 1, n_p$) are adjoint variables. Using eqs. 22, 23 and 32 (for $\Phi = R_{n,a}$ and appropriate substitutions for γ_i, γ_j and γ_l) for the first-, second- and third-order derivatives of the state equations, the computation of $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$ becomes independent (a) of $\frac{\delta U_{k,d}}{\delta b_q}$, thanks to the already satisfied eq. 9, (b) of $\frac{\delta^2 U}{\delta c_i \delta b_q}$ and $\frac{\delta^2 U}{\delta c_j \delta b_q}$, thanks to the already satisfied eq. 30 and, finally, (c) of $\frac{\delta^3 U}{\delta c_j \delta c_j \delta b_q}$, by satisfying

$$
\begin{aligned}
& \frac{\partial^3 F}{\partial c_i \partial c_j \partial U_{r,g}} + \frac{\partial^3 F}{\partial c_i \partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 F}{\partial c_j \partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \\
+ \; & \frac{\partial^3 F}{\partial U_{k,d} \partial U_{m,e} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \cdot \frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 F}{\partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j} + \mathcal{K}^{i,j}_{n,a} \frac{\partial R_{n,a}}{\partial U_{r,g}} \\
+ \; & \mathcal{L}^{j}_{n,a} \left(\frac{\partial^2 R_{n,a}}{\partial c_i \partial U_{r,g}} + \frac{\partial^2 R_{n,a}}{\partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \right) + \mathcal{M}^{i}_{n,a} \left(\frac{\partial^2 R_{n,a}}{\partial c_j \partial U_{r,g}} + \frac{\partial^2 R_{n,a}}{\partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_j} \right) \\
+ \; & \mathcal{N}_{n,a} \left(\frac{\partial^3 R_{n,a}}{\partial c_i \partial c_j \partial U_{r,g}} + \frac{\partial^3 R_{n,a}}{\partial c_i \partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 R_{n,a}}{\partial c_j \partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \right. \\
+ \; & \left. \frac{\partial^3 R_{n,a}}{\partial U_{k,d} \partial U_{m,e} \partial U_{r,g}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \cdot \frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 R_{n,a}}{\partial U_{k,d} \partial U_{r,g}} \cdot \frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j} \right) = 0
\end{aligned}
\tag{36}
$$

Eq. 36 can be solved for $\mathcal{K}^{i,j}_{n,a}$ at CPU cost equal to $\frac{M(M+1)}{2}$ EFS. It can be shown that $\mathcal{L}^{j}_{n,a} \equiv \mathcal{M}^{j}_{n,a}$.

The CPU cost for the computation of all adjoint variable fields $\mathcal{K}^{i,j}_{n,a}$, $\mathcal{L}^{j}_{n,a} \equiv \mathcal{M}^{j}_{n,a}$, $\mathcal{N}_{n,a}$, is summarized in table 1. Having computed them, $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$ is, finally, given by

$$
\begin{aligned}
\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q} \; = \; & \frac{\partial^3 F}{\partial c_i \partial c_j \partial b_q} + \frac{\partial^3 F}{\partial c_i \partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 F}{\partial c_j \partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \\
+ \; & \frac{\partial^3 F}{\partial b_q \partial U_{k,d} \partial U_{m,e}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \cdot \frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 F}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j} \\
+ \; & \mathcal{K}^{i,j}_{n,a} \frac{\partial R_{n,a}}{\partial b_q} + \mathcal{L}^{j}_{n,a} \left(\frac{\partial^2 R_{n,a}}{\partial c_i \partial b_q} + \frac{\partial^2 R_{n,a}}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \right) \\
+ \; & \mathcal{M}^{i}_{n,a} \left(\frac{\partial^2 R_{n,a}}{\partial c_j \partial b_q} + \frac{\partial^2 R_{n,a}}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_j} \right) \\
+ \; & \mathcal{N}_{n,a} \left(\frac{\partial^3 R_{n,a}}{\partial c_i \partial c_j \partial b_q} + \frac{\partial^3 R_{n,a}}{\partial c_i \partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_j} + \frac{\partial^3 R_{n,a}}{\partial c_j \partial b_q \partial U_{k,d}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \right. \\
+ \; & \left. \frac{\partial^3 R_{n,a}}{\partial b_q \partial U_{k,d} \partial U_{m,e}} \cdot \frac{\delta U_{k,d}}{\delta c_i} \cdot \frac{\delta U_{m,e}}{\delta c_j} + \frac{\partial^2 R_{n,a}}{\partial b_q \partial U_{k,d}} \cdot \frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j} \right)
\end{aligned}
\tag{37}
$$

Computation of:	**Solution of**	**CPU cost (EFS)**
Flow Variables	Flow equations	1
$\frac{\delta U_{k,d}}{\delta c_i}$	eq. 22 ($\Phi = R_{n,a}, \gamma_i = c_i$)	M
$\frac{\delta^2 U_{k,d}}{\delta c_i \delta c_j}$	eq. 23 ($\Phi = R_{n,a}, \gamma_i = c_i, \gamma_j = c_j$)	$\frac{M(M+1)}{2}$
$\mathcal{N}_{n,a}$	eq. 9	1
$\mathcal{L}^i_{n,a} = \mathcal{M}^i_{n,a}$	eq. 30	M
$\mathcal{K}^{i,j}_{n,a}$	eq. 36	$\frac{M(M+1)}{2}$
Total Cost		$2 + 3M + M^2$

Table 1: CPU cost (in EFS) of computing $\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q}$, using the proposed DD$_c$-DD$_c$-AV$_b$ approach. The total CPU cost is independent of N. The symbols of the adjoint variables of the discrete approach are used (left column).

In the continuous approach, the differentiation of eq. 20 w.r.t. c_i gives

$$\begin{aligned}
\frac{\delta^2 F}{\delta c_i \delta b_q} &= \int_{S_w} (p - p_{tar}) \frac{\delta p}{\delta c_i} \frac{\delta (dS)}{\delta b_q} \\
&+ \int_{S_w} \left(\mathcal{L}^i_{k+1} p - \mathcal{L}^i_n f_{nk} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} + \int_{S_w} \left(\mathcal{N}_{k+1} \frac{\delta p}{\delta c_i} - \mathcal{N}_n \frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta \left(n_k dS \right)}{\delta b_q} \\
&- \int_{S_w} \mathcal{L}^i_n \frac{\partial f_{nk}}{\partial x_l} \frac{\delta x_l}{\delta b_q} n_k dS - \int_{S_w} \mathcal{N}_n \frac{\partial}{\partial x_l} \left(\frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta x_l}{\delta b_q} n_k dS \\
&+ \int_{S_w} \mathcal{L}^i_n \frac{\partial f_{nk}}{\partial x_k} \frac{\delta x_l}{\delta b_q} n_l dS + \int_{S_w} \mathcal{N}_n \frac{\partial}{\partial x_k} \left(\frac{\delta f_{nk}}{\delta c_i} \right) \frac{\delta x_l}{\delta b_q} n_l dS
\end{aligned} \tag{38}$$

where

$$\mathcal{L}^i_n = \frac{\delta \mathcal{N}_n}{\delta c_i} \tag{39}$$

In the sake of convenience, in the continuous approach, the adjoint variables are given by the same symbols as their discrete counterparts. The $\mathcal{L}^i_n$ fields ($i = 1, M$) are computed by solving the following system of PDEs

$$-A_{nmk} \frac{\partial \mathcal{L}^i_n}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_i} \frac{\partial \mathcal{N}_n}{\partial x_k} = 0 \tag{40}$$

derived from the DD$_c$ of the adjoint equations, eq. 18. The inlet/outlet boundary conditions imposed to eqs. 40 are in conformity to those imposed to the adjoint equations.

The solid wall boundary conditions for $\mathcal{L}_k^i$ are derived from the differentiation of eq. 19 w.r.t. c_i, which yields

$$\frac{\delta p}{\delta c_i} + \mathcal{L}_{k+1}^i n_k = 0 \tag{41}$$

Regarding the computation of $\frac{\delta f_{nk}}{\delta c_i}$, recall that $\frac{\delta U_m}{\delta c_i}$ are known from the DD$_c$ of the governing equations and $\frac{\delta f_{nk}}{\delta c_i}$ are functions of $\frac{\delta U_m}{\delta c_i}$.

The third-order mixed sensitivity derivatives of F, required for eq. 5, are obtained from the differentiation of eq. 38 w.r.t. c_j, as follows

$$\begin{aligned}
\frac{\delta^3 F}{\delta c_i \delta c_j \delta b_q} &= \int_{S_w} \frac{\delta p}{\delta c_i}\frac{\delta p}{\delta c_j}\frac{\delta (dS)}{\delta b_q} + \int_{S_w} (p - p_{tar})\frac{\delta^2 p}{\delta c_i \delta c_j}\frac{\delta (dS)}{\delta b_q} \\
&+ \int_{S_w} \left(\mathcal{K}_{k+1}^{i,j} p - \mathcal{K}_n^{i,j} f_{nk}\right)\frac{\delta\left(n_k dS\right)}{\delta b_q} + \int_{S_w}\left(\mathcal{L}_{k+1}^i \frac{\delta p}{\delta c_j} - \mathcal{L}_n^i \frac{\delta f_{nk}}{\delta c_j}\right)\frac{\delta\left(n_k dS\right)}{\delta b_q} \\
&+ \int_{S_w}\left(\mathcal{L}_{k+1}^j \frac{\delta p}{\delta c_i} - \mathcal{L}_n^j \frac{\delta f_{nk}}{\delta c_i}\right)\frac{\delta\left(n_k dS\right)}{\delta b_q} + \int_{S_w}\left(\mathcal{N}_{k+1}\frac{\delta^2 p}{\delta c_i \delta c_j} - \mathcal{N}_n \frac{\delta^2 f_{nk}}{\delta c_i \delta c_j}\right)\frac{\delta\left(n_k dS\right)}{\delta b_q} \\
&- \int_{S_w} \mathcal{K}_n^{i,j}\frac{\partial f_{nk}}{\partial x_l}\frac{\delta x_l}{\delta b_q} n_k dS - \int_{S_w} \mathcal{L}_n^i \frac{\partial}{\partial x_l}\left(\frac{\delta f_{nk}}{\delta c_j}\right)\frac{\delta x_l}{\delta b_q} n_k dS \\
&- \int_{S_w} \mathcal{L}_n^j \frac{\partial}{\partial x_l}\left(\frac{\delta f_{nk}}{\delta c_i}\right)\frac{\delta x_l}{\delta b_q} n_k dS - \int_{S_w} \mathcal{N}_n \frac{\partial}{\partial x_l}\left(\frac{\delta^2 f_{nk}}{\delta c_i \delta c_j}\right)\frac{\delta x_l}{\delta b_q} n_k dS \\
&+ \int_{S_w} \mathcal{K}_n^{i,j}\frac{\partial f_{nk}}{\partial x_k}\frac{\delta x_l}{\delta b_q} n_l dS + \int_{S_w} \mathcal{L}_n^i \frac{\partial}{\partial x_k}\left(\frac{\delta f_{nk}}{\delta c_j}\right)\frac{\delta x_l}{\delta b_q} n_l dS \\
&+ \int_{S_w} \mathcal{L}_n^j \frac{\partial}{\partial x_k}\left(\frac{\delta f_{nk}}{\delta c_i}\right)\frac{\delta x_l}{\delta b_q} n_l dS + \int_{S_w} \mathcal{N}_n \frac{\partial}{\partial x_k}\left(\frac{\delta^2 f_{nk}}{\delta c_i \delta c_j}\right)\frac{\delta x_l}{\delta b_q} n_l dS
\end{aligned} \tag{42}$$

where

$$\mathcal{K}_k^{i,j} = \mathcal{K}_k^{j,i} = \frac{\delta \mathcal{L}_k^i}{\delta c_j} \tag{43}$$

$\mathcal{K}_k^{i,j}$ $(i = 1, M;\ j = 1, M)$ are computed by solving the field equation resulting from the differentiation of eq. 40 w.r.t. c_j, which is

$$-A_{nmk}\frac{\partial \mathcal{K}_n^{i,j}}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_j}\frac{\partial \mathcal{L}_n^i}{\partial x_k} - \frac{\delta A_{nmk}}{\delta c_i}\frac{\partial \mathcal{L}_n^j}{\partial x_k} - \frac{\delta^2 A_{nmk}}{\delta c_i \delta c_j}\frac{\partial \mathcal{N}_n}{\partial x_k} = 0 \tag{44}$$

The boundary conditions imposed to eqs. 44 are, also, in conformity with those imposed to the adjoint equations. The solid wall boundary conditions for $\mathcal{K}_k^{i,j}$ are derived from the differentiation of eq. 41 w.r.t. c_i, yielding

$$\frac{\delta^2 p}{\delta c_i \delta c_j} + \mathcal{K}_{k+1}^{i,j} n_k = 0 \tag{45}$$

The solution of systems 44 and 45 for $\mathcal{K}_k^{i,j}$ and substitution to eq. 38 yields the required third-order mixed sensitivities at the cost of $M(M+1)/2$ EFS, which does not depend on N. The terms $\frac{\delta^2 f_{nk}}{\delta c_i \delta c_j}$ can be computed in a straightforward manner from the computed values of $\frac{\delta U_m}{\delta c_i}$ and $\frac{\delta^2 U_m}{\delta c_i \delta c_j}$.

The overall computational cost of the continuous gradient-based SOSM method is, practically, equal to that of its discrete counterpart incorporating exactly the same number of EFS.

4 APPLICATIONS-VALIDATION OF THE COMPUTED DERIVATIVES

4.1 Case Study 1: Robust design using the discrete approach (1D)

In the first case, the discrete approach is tested. A steady state flow model for the quasi-1D viscous flows is considered, where the flow equations are

$$\frac{\partial \vec{f}}{\partial x} = \vec{q_s} + \vec{q_\nu} \tag{46}$$

Vectors $\vec{U}$, $\vec{f} = \vec{f}(\vec{U})$, $\vec{q_s}$ (source terms due to blockage effects) and $\vec{q_\nu}$ (source terms due to viscous effects) are given by

$$\vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix},\ \vec{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u\left(\rho E + p\right) \end{bmatrix},\ \vec{q_s} = -\frac{1}{S}\frac{dS}{dx}\begin{bmatrix} \rho u \\ \rho u^2 \\ u\left(\rho E + p\right) \end{bmatrix},\ \vec{q_\nu} = -\lambda \frac{L}{2D}\begin{bmatrix} 0 \\ \rho u^2 \\ \rho u^3 \end{bmatrix}$$

$S = S(x)$ corresponds to the cross-sectional area of the duct, which is controlled by the design variables $b_q,\ q = 1, N$; the latter stand for the ordinates of the Bézier control points

Method	DD or DD-DD	FD
$\frac{\delta F}{\delta M_{2,is}}$	−14858410.83	4227340.85
$\frac{\delta F}{\delta \lambda}$	−14857665.84	4227735.54
$\frac{\delta^2 F}{\delta M_{2,is}^2}$	6501400443.98	6498032545.33
$\frac{\delta^2 F}{\delta M_{2,is} \delta \lambda}$	−1851952901.38	−1850969770.31
$\frac{\delta^2 F}{\delta \lambda^2}$	711627191.25	710883026.73

Table 2: Case Study 1: (a) First- and second-order derivatives w.r.t. the environmental variables $M_{2,is}$ and λ, computed for the initial duct, using DD_c or DD_c-DD_c and FD (for step-size $\epsilon = 5 \times 10^{-4}$; ϵ-independent results).

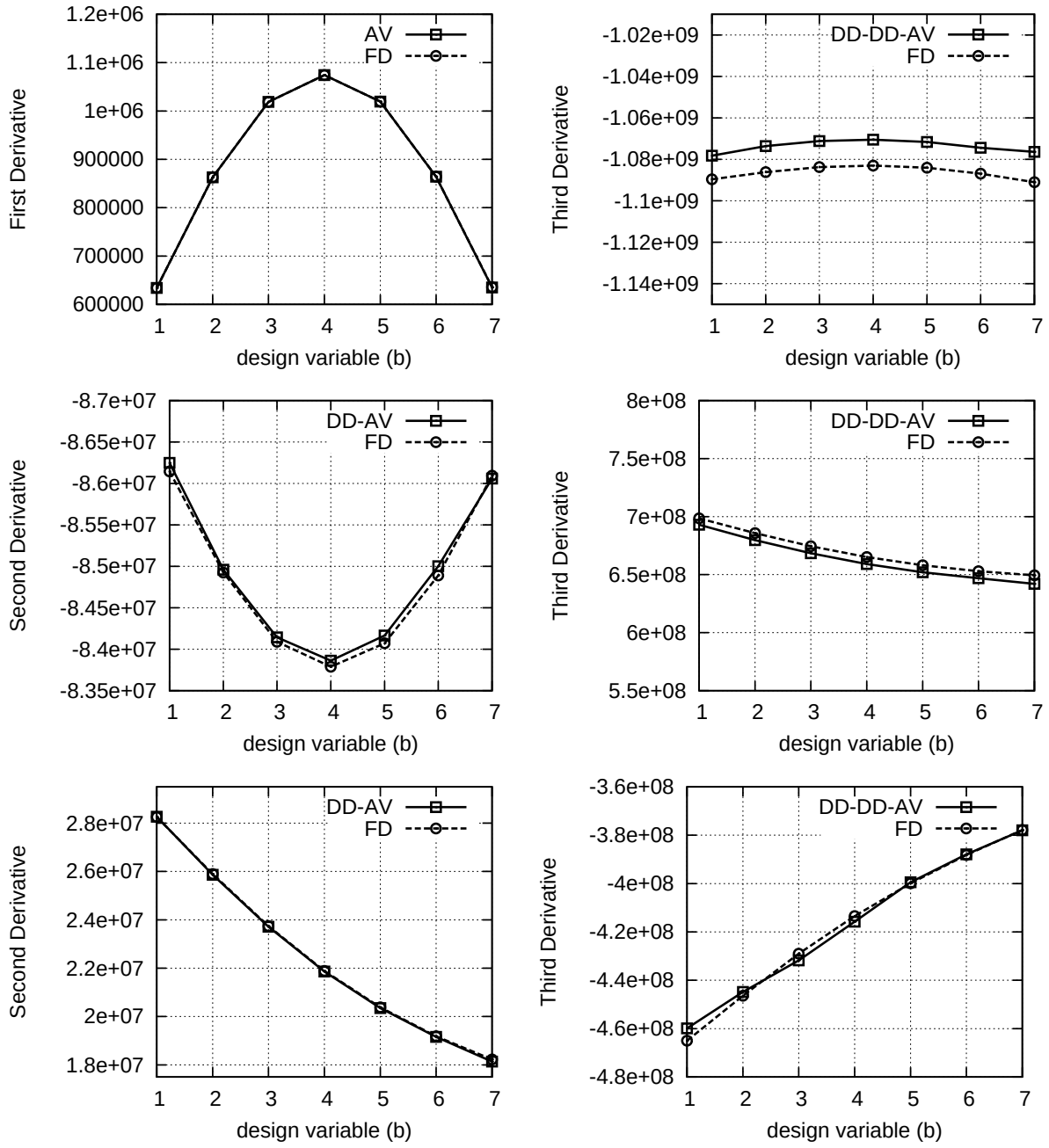

Figure 1: Case Study 1: Derivatives computed for the initial duct geometry. Top-left: $\frac{\delta F}{\delta b_q}$, using AV and FD. Middle-left: $\frac{\delta^2 F}{\delta M_{2,is} \delta b_q}, q = 1, 7$. Bottom-left: $\frac{\delta^2 F}{\delta \lambda \delta b_q}, q = 1, 7$. All second derivatives were computed for the duct, using the DD_c-AV_b approach and FD. Top-right: $\frac{\delta^3 F}{\delta^2 M_{2,is} \delta b_q}, q = 1, 7$. Middle-right: $\frac{\delta^3 F}{\delta M_{2,is} \delta \lambda \delta b_q}, q = 1, 7$. Bottom-right: $\frac{\delta^3 F}{\delta \lambda^2 \delta b_q}, q = 1, 7$. Third derivatives were computed for the same geometry, using the DD_c-DD_c-AV_b approach and FD.

with fixed abscissas, used to parameterize $S(x)$. Also, D is the local hydraulic diameter of the duct and λ is the Darcy-Weisbach friction coefficient.

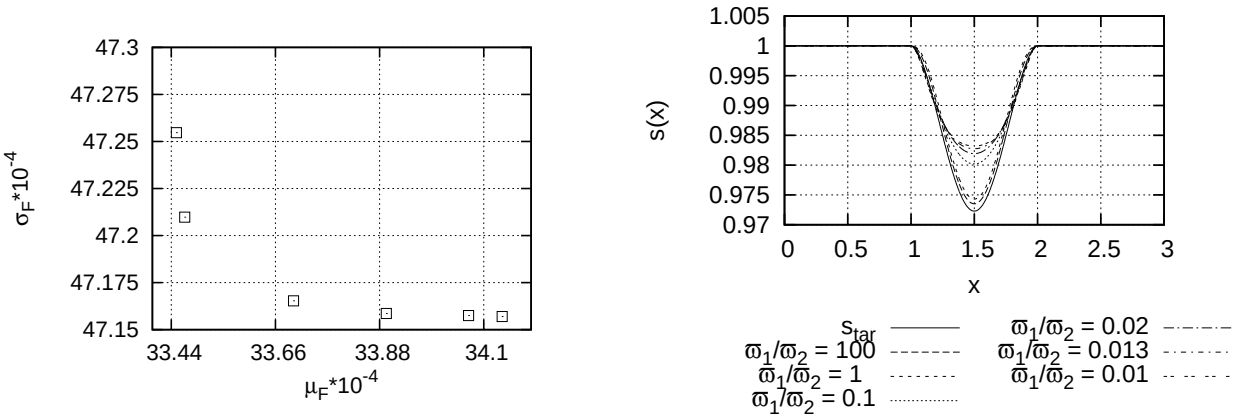

Figure 2: Case Study 1: Inverse design of an 1D symmetric duct. Left: Pareto front of non-dominated solutions on the (μ_F, σ_F) plane, calculated with the discrete AV-based SOSM algorithm. Right: The geometries calculated through the SOSM algorithm for different weight factors, in comparison with geometry $S_{tar}(x)$ which results from a "fixed-environment" optimization aiming at minimum F.

The aim of this example is not only to perform the gradient-based optimization itself but, also, to validate the computed derivatives of F w.r.t. $M_{2,is}$ and λ, before even proceeding to the computation of $\frac{\delta \widehat{F}}{\delta b_q}$. The validation is performed against finite-differences (FD), after ensuring that the selected step-size ϵ leads to ϵ-insensitive derivatives.

The isentropic exit Mach number $M_{2,is}$ and λ are considered as the two environmental variables ($M = 2$) with user-defined mean values $\overline{M}_{2,is} = 0.4, \overline{\lambda} = 0.025$ and standard deviations $\sigma_{M_{2,is}} = 0.01, \sigma_\lambda = 0.001$. In this example, eleven Bézier control points were used to parameterize $S(x)$. Seven of the control points ordinates were free to vary, while the remaining four (the first and last two control points) were fixed; thus, $N = 7$. Derivatives w.r.t. the environmental variables are tabulated, see table 2, and the ones w.r.t. the design variables are plotted in fig. 1. In all cases, the agreement is absolutely satisfactory.

These sensitivities were used to compute $\frac{\delta \widehat{F}}{\delta b_q}$ (eq. 5) for different combinations of the weight values w_1 and w_2 (eq. 4). The optimal solutions obtained for the different (w_1, w_2) pairs of values form a Pareto front of optimal solutions. In figure 2, left, the Pareto front on the (μ_F, σ_F) plane, computed by the proposed method, is presented. In figure 2, right, the optimal geometries corresponding to the aforementioned Pareto front points, are presented in comparison with the outcome of a "fixed-environment" optimization, merely targeting at minimum F.

4.2 Case Study 2: Robust design using the continuous approach (2D)

The second inverse design problem examined is governed by the 2D Euler equations and optimized using the continuous approach. The case is concerned with the robust design of a 2D cascade airfoil which is formed by two Bézier-Bernstein polynomials. The design variables are the normal to the chord coordinates of all but the first and last Bézier control points, resulting to $N=12$ design variables. The exit isentropic Mach number $M_{2,is}$ and the inlet flow angle α_1 are the two environmental variables (c_1 and c_2, respectively; so, $M=2$).

In fig. 3, first-order sensitivities $\frac{\delta\mu_F}{\delta b_q}$ and $\frac{\delta\sigma_F}{\delta b_q}$ were compared to FD. The comparison is satisfactory. The sensitivity derivatives $\frac{\delta\mu_F}{\delta b_q}$ and $\frac{\delta\sigma_F}{\delta b_q}$ were used in the steepest descent algorithm in order to provide geometries with robust aerodynamic performance. For this purpose, a series of five independent optimization runs were performed based on steepest descent, eq. 6, where the derivatives were computed by eq. 5, with five different values of the weights w_1 and w_2. In fig. 4, left, the convergence rate of the mean value of F is shown for the different weight value sets. As expected, the μ_F decreases always, whereas the higher the weight w_1 the higher the decrease in μ_F. The same statement (for w_2, instead of w_1) can be made for the convergence of σ_F, fig. 4, right.

As in case 1, repetitive runs of the programmed software with different value sets (w_1, w_2) resulted to the formation of a Pareto front of optimal solutions on the (μ_F, σ_F) plane. Fig. 5 shows this front for the computations already presented in fig. 4. It can be seen that the Pareto front is convex and continuous. Finally, the five optimal geometries, corresponding to the non-dominated members of the Pareto front, are shown in fig. 5, right, compared with the reference geometry that reproduces the target pressure distribution. Pronounced changes in the shape of the airfoil close to its rear part can be seen.

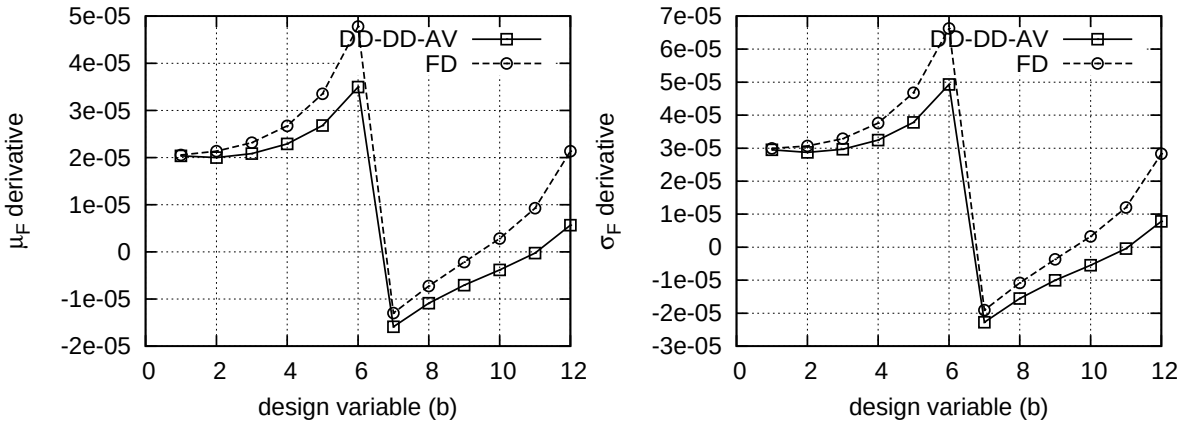

Figure 3: Case Study 2: Inverse design of a 2D symmetric cascade. Comparison of sensitivities $\frac{\delta\mu_F}{\delta b_q}$ and $\frac{\delta\sigma_F}{\delta b_q}$, computed using the proposed method and FD.

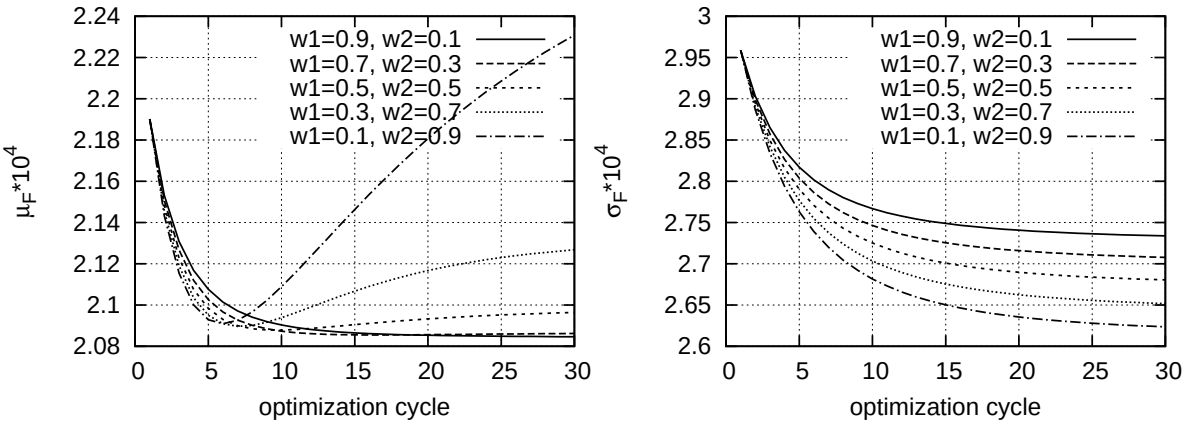

Figure 4: Case Study 2: Inverse design of a 2D symmetric cascade. Convergence of the μ_F and σ_F values, for different pairs of weight values.

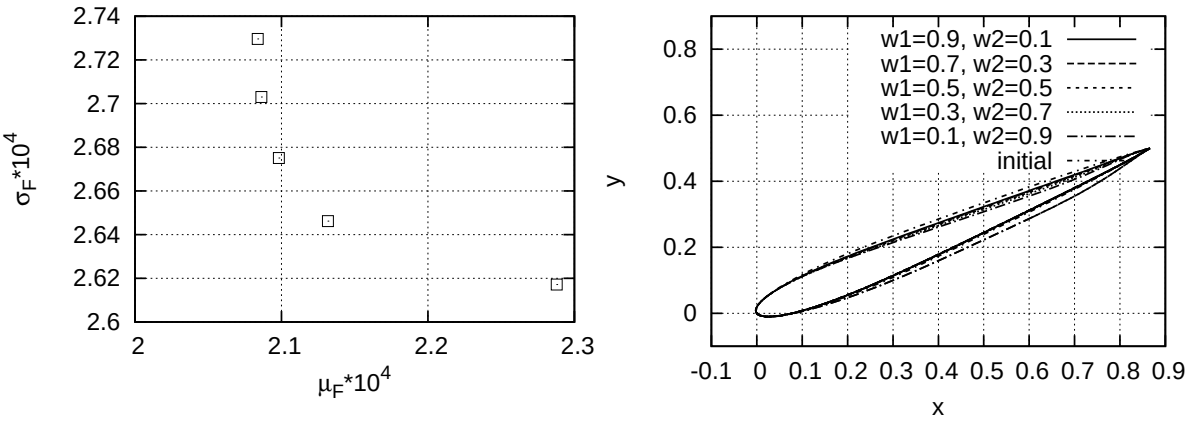

Figure 5: Case Study 2: Inverse design of a 2D symmetric cascade. Left: Pareto front with five optimal solutions, plotted on the (μ_F, σ_F) plane. Right: Reference and optimal airfoils for five different pairs of weight values (w_1, w_2).

5 CONCLUSIONS

The scope of this paper was to present efficient ways of computing the high–order derivatives required for the solution of robust design problems in aerodynamics, based on the second–order, second–moment model and gradient-based algorithms. The mean value and the standard deviation of a target function to be minimized are expressed in terms of its derivatives w.r.t. the environmental values. An additional differentiation w.r.t. the design variables is needed to provide the gradient of the objective function (formed by weighted sums of the mean value and standard deviation) and, thus, drive the steepest

descent algorithm. High–order derivatives were computed through combinations of the adjoint method and the direct differentiation of the flow (and the adjoint) equations. The cost of the proposed method scales with the square of number of environmental variables M but not with that of the design variables (N). Since $M << N$, the total CPU cost for the robust design is affordable. The development was presented according to both the discrete and continuous approaches, demonstrating (among other) the similarities between them. The proposed algorithm was successfully applied to the robust optimization of 1D and 2D geometries, and, through repetitive calls to the same optimization software, optimal Pareto fronts on the mean value and standard deviation of F plane have been computed.

6 ACKNOWLEDGEMENTS

This work was carried out during the first phase of a Basic Research Program "PEVE 2010", funded by the National Technical University of Athens. In the past, the development of some methods and tools, used as background information herein, was supported by a State Scholarships Foundation (Greece) grant to the second author.

REFERENCES

[1] O. Pironneau, On optimum design in fluid mechanics, *Journal of Fluid Mechanics*, **64**, 97–110 (1974).

[2] G. R. Shubin and P. D. Frank, A comparison of the implicit gradient approach and the variational approach to aerodynamic design optimization, *Boeing Computer Services Report AMS-TR-163*, (1991).

[3] M. C. Duta, M. B. Giles and M. S. Campobasso, The harmonic adjoint approach to unsteady turbomachinery design, *International Journal for Numerical Methods in Fluids*, **40(3-4)**, 323–332 (2002).

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

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

[6] D. I. Papadimitriou and K. C. Giannakoglou, A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows, *Computers & Fluids*, **363**, 25–341 (2007).

[7] D. I. Papadimitriou and K. C. Giannakoglou, Total pressure losses minimization in turbomachinery cascades, using a new continuous adjoint formulation, *Journal of Power and Energy*, **221(6)**, 865–872 (2007).

[8] A. C. Taylor III, L. L. Green, P. A. Newman and M. M. Putko, Some advanced concepts in discrete aerodynamic sensitivity analysis. *AIAA Paper 2001-2529* (2001).

[9] M. Martinelli, A. Dervieux and L. Hascoët, Strategies for computing second-order derivatives in CFD design problems, *Proceedings of West-East High Speed Flow Field Conference*, Moscow, Russia (2007).

[10] S. Schmidt and V. Schulz, Impulse response approximations of discrete shape Hessians with application in CFD, *SIAM Journal on Control and Optimization*, **48**(4), 2562–2580 (2009).

[11] E. Arian and S. Taasan, Analysis of the Hessian for aerodynamic optimization: Inviscid flow, *Computers & Fluids*, **28**(7), 853-877 (1999).

[12] D. I. Papadimitriou and K. C. Giannakoglou, Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems, *International Journal for Numerical Methods in Fluids*, **56**, 1929–1943 (2008).

[13] T. Zervogiannis, D. I. Papadimitriou and K. C. Giannakoglou, Total pressure losses minimization in turbomachinery cascades using the exact Hessian, *Journal of Computer Methods in Applied Mechanics and Engineering*, **199**(41-44), 2697–2708 (2010).

[14] D. I. Papadimitriou and K. C. Giannakoglou, Aerodynamic shape optimization using first and second order adjoint and direct approaches, *Archives of Computational Methods in Engineering, (State of the Art Reviews)*, **15**(4), 447–488 (2008).

[15] D.I. Papadimitriou and K. C. Giannakoglou, Computation of the Hessian matrix in aerodynamic inverse design using continuous adjoint formulations, *Computers & Fluids*, **37**, 1029–1039 (2008).

[16] D.I. Papadimitriou and K. C. Giannakoglou, The continuous direct-adjoint approach for second order sensitivities in viscous aerodynamic inverse design problems, *Computers & Fluids*, **38**, 1539–1548 (2009).

[17] R. W. Waters and L. Huyse, Uncertainty analysis for fluid mechanics with applications. *NASA/CR 2002-211449 or ICASE Report No. 2002-1* (2002).

[18] M. M. Putko, P. A. Newman, A.C. Taylor III and L. L. Green, Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives, *AIAA Tech. Rep. 2528* (2001).

[19] L. Green, H.Z. Lin and M.R. Khalessi, Probabilistic methods for uncertainty propagation applied to aircraft design, *20th AIAA Applied Aerodynamics Conference*, St. Louis, Missouri (2002).

[20] M. Martinelli and R. Duvigneau, On the use of second-order derivatives and metamodel-based Monte-Carlo for uncertainty estimation in aerodynamics, *Computers & Fluids*, **39**(6),953–964 (2010).

[21] M. Elliot and J. Peraire, Constrained, multipoint shape optimization for complex 3D configurations, *Aeronautical Journal*, **102**(1017), 365–376 (1998).

[22] J. Reuther, J. Alonso, M. J. Rimlinger and D. Saunders, Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, Part 1, *Journal of Aircraft*, **36**, 51–60 (1999), Part 2, *Journal of Aircraft*, **36**, 61–74 (1999),

[23] D. W. Zingg and S. Elias, Aerodynamic optimization under a range of operating conditions, *AIAA Journal*, **44**(11) (2006).

[24] R. Duvigneau, Aerodynamic shape optimization with uncertain operating conditions using metamodels, *Tech. Rep. 6143, INRIA* (2007).

[25] R. Duvigneau, Robust design of a transonic with uncertain Mach number, *EUROGEN 2007, Evolutionary Methods for Design, Optimization and Control*, Jyvaskyla, Finland (2007).

[26] L. Huyse and R. Lewis, Aerodynamic shape optimization of two-dimensional airfoils under uncertain operating conditions, *NASA/CR-2001-210648 or ICASE Report No. 2001-1* (2001).

[27] W. Li, L. Huyse, R. Lewis and S. Padula, Robust airfoil optimization to achieve consistent drag reduction over a Mach range, *NASA/CR-2001-211042 or ICASE Report No. 2001-22* (2001).

[28] K. Shimoyama, A. Oyama and K. Fujii, A new efficient and useful robust optimization approach design for multi-objective six sigma, *The 2005 IEEE Congress on Evolutionary Computation* (2005).