INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1929–1943
Published online 11 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1584

Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems

D. I. Papadimitriou and K. C. Giannakoglou*, †, ‡

Laboratory of Thermal Turbomachines, School of Mechanical Engineering, National Technical University of Athens, Athens, Greece

SUMMARY

In this paper, four approaches to compute the Hessian matrix of an objective function used often in aerodynamic inverse design problems are presented. The computationally less expensive among them is selected and applied to the reconstruction of cascade airfoils that reproduce a prescribed pressure distribution over their walls, under inviscid and viscous flow considerations. The selected approach is based on the direct sensitivity analysis method for the computation of first derivatives, followed by the discrete adjoint method for the computation of the Hessian matrix. The applications presented in this paper show that the Newton method, based on exact Hessian matrices, outperforms other gradient-based algorithms such as steepest descent or BFGS algorithm. Copyright © 2007 John Wiley & Sons, Ltd.

Received 21 November 2006; Accepted 2 July 2007

KEY WORDS: aerodynamic shape optimization; adjoint method; Hessian matrix

1. INTRODUCTION

The use of gradient-based methods in aerodynamic shape optimization problems calls for an efficient tool to compute the gradient of the objective function with respect to the design variables, with maximum accuracy and minimum CPU cost. The adjoint approach perfectly suits this purpose. A thorough investigation of the discrete and continuous adjoint approaches as well as their relative advantages and disadvantages can be found in detail in [1]. A literature survey shows that, so far, the adjoint technique has been successfully used in various applications, such as the inverse design of 2D and 3D shapes [2], design of airfoils or wings with desirable drag and/or lift [3], sonic boom reduction [4], optimization in unsteady flows [5], design of optimal turbomachinery cascades [6], etc.

*Correspondence to: K. C. Giannakoglou, Laboratory of Thermal Turbomachines, School of Mechanical Engineering, National Technical University of Athens, P.O. Box 64069, 15710 Athens, Greece.
†E-mail: kgianna@central.ntua.gr
‡Associate Professor.

Copyright © 2007 John Wiley & Sons, Ltd.

Historically, the adjoint method for potential flows was introduced by Pironneau [7] and extended to transonic flows by Jameson [8]. Nowadays, adjoint methods are in widespread use, both in their discrete (where discretization precedes adjoint operation [9–11]) and continuous variants (adjoint operation precedes discretization [8, 12, 13]). At the same time, efforts have been made to improve the efficiency of conventional adjoint approaches. Among them, it is appropriate to mention the one-shot algorithm, in which the flow (state), adjoint (costate) and optimization equations are solved simultaneously [14, 15], the incomplete gradient method that skips the computation of costly terms [16], the reduced gradient method that overcomes grid coordinate sensitivities ([17] for inviscid flows and [18] for viscous flows).

In the field of aerodynamics, most of the existing deterministic optimization algorithms make use of the objective function gradient without accounting for the Hessian matrix. The most well-known gradient-based algorithms are steepest descent, several variants of the conjugate gradient method and quasi-Newton approaches. Some other efficient descent algorithms, such as the truncated-Newton method which takes advantage of the second-order information without being too costly and the L-BFGS quasi-Newton one, which is suitable for problems with large memory requirements, are also in use. In the literature on aerodynamic optimization, the computation of the exact Hessian matrix using adjoint approaches is rare. In [19], the adjoint approach is used to compute the Hessian matrix in the neighbourhood of the optimal solution. In [20], the exact Hessian matrix is computed and used in structural optimization problems. The computation and use of the exact Hessian matrix may also be found in variational data assimilation problems in meteorology with the shallow-water equations as state equations [21]. In this kind of problems, the limited memory BFGS optimization method with diagonal-preconditioner update formulas [22], or the combination of L-BFGS and truncated Newton method [23] improves the optimization performance. Finally, the use of the Hessian matrix as a preconditioner in the steepest descent algorithm is shown in [24] demonstrating the efficiency of the exact-Newton method near the optimum.

In view of the above, this paper investigates possible ways to compute the exact Hessian, based on the adjoint approach (fully or partially) and assesses its use in aerodynamic optimization. The paper is concerned with both inviscid and viscous flows; hence, the state equations are either the Euler or the Navier–Stokes ones, depending on the problem under consideration. All possible direct, adjoint and mixed approaches are presented; the *direct–direct*, *direct–adjoint*, *adjoint–direct* and *adjoint–adjoint* approaches (according to whether the first and second derivatives are computed using the direct or the adjoint approach) are investigated with respect to the CPU cost for computing the Hessian matrix in a problem with N design variables. The investigation concludes that the *direct–adjoint* approach is the most efficient one. The Hessian matrix values computed by the four approaches (which are practically identical since all of them correspond to the exact Hessian) are compared with those computed using finite differences. The most efficient one (*direct–adjoint*) is used to solve airfoil inverse design problems together with the Newton method. The performance of the overall optimization algorithm is compared with that of the quasi-Newton (BFGS) method and proves to be much more efficient.

2. ON THE COMPUTATION OF FUNCTIONAL GRADIENT AND HESSIAN

In aerodynamic shape optimization problems, the objective functional F can be expressed as $F = F(\mathbf{b}, \mathbf{U}(\mathbf{b}))$, denoting the dependency of F on both flow variables $\mathbf{U}$ and design variables $\mathbf{b}$. Any variation in $\mathbf{b}$ $(b_i, i = 1, \ldots, N)$ causes variations in the aerodynamic shape and the

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

corresponding computational grid. The total derivative of F with respect to b_i may be expressed as

$$\frac{\mathrm{d}F}{\mathrm{d}b_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} \tag{1}$$

Equation (1) can be used to compute $\mathrm{d}F/\mathrm{d}b_i$, which is sufficient to drive a steepest descent algorithm ($b_j^{\lambda+1} = b_j^{\lambda} - \eta \mathrm{d}F^{\lambda}/\mathrm{d}b_j$) towards the optimal solution, provided that $\mathrm{d}U_k/\mathrm{d}b_i$ is known. The latter can be computed using the nonlinear system of discrete residual equations (arising from some discretization of the flow partial differential equations) $R_m = 0$ ($m = 1, \ldots, M$, where M stands for the product of the number of grid nodes and that of the equations per node). For the sake of convenience, in all subsequent formulas, $i, j \in [1, N]$ and $k, m, n \in [1, M]$. It is worth noting that in aerodynamic optimization problems $N \ll M$. Since $R_m = 0$, its derivatives with respect to b_i are zero too, namely

$$\frac{\mathrm{d}R_m}{\mathrm{d}b_i} = \frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} = 0 \tag{2}$$

According to the Newton method, provided that the gradient and the Hessian matrix of F with respect to b_i have been computed (or approximated), the design variables are updated as follows:

$$b_j^{\lambda+1} = b_j^{\lambda} + \mathrm{d}b_j^{\lambda} \tag{3}$$

where λ is the optimization cycle counter and $\mathrm{d}b_j^{\lambda}$ is the solution of

$$\frac{\mathrm{d}^2 F^{\lambda}}{\mathrm{d}b_i\,\mathrm{d}b_j}\,\mathrm{d}b_j^{\lambda} = -\frac{\mathrm{d}F^{\lambda}}{\mathrm{d}b_i} \tag{4}$$

As already explained, the main scope of this paper is to evaluate the efficiency of four possible approaches to compute $\mathrm{d}F/\mathrm{d}b_i$ and $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$, to be used in the Newton method for the minimization of F. Practically, we are exclusively concerned with the computational cost associated with each optimization cycle, since all four approaches provide exact values for the first and second derivatives. This cost is expressed in terms of the number of system solutions required to compute $\mathrm{d}F/\mathrm{d}b_i$ and $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ (which should be increased by one for the solution of the discrete flow equations).

The gradient $\mathrm{d}F/\mathrm{d}b_i$ can be computed in a straightforward manner by solving Equation (2) for $\mathrm{d}U_k/\mathrm{d}b_i$ and substituting their values into Equation (1). This corresponds to the so-called *direct* approach [11], and its cost is equivalent to that of N system solutions. Similarly, the Hessian matrix can be computed using

$$\frac{\mathrm{d}^2F}{\mathrm{d}b_i\,\mathrm{d}b_j} = \frac{\partial^2 F}{\partial b_i\,\partial b_j} + \frac{\partial^2 F}{\partial b_i\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \frac{\partial^2 F}{\partial U_k\,\partial b_j}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} + \frac{\partial^2 F}{\partial U_k\,\partial U_m}\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\frac{\mathrm{d}U_m}{\mathrm{d}b_j} + \frac{\partial F}{\partial U_k}\frac{\mathrm{d}^2U_k}{\mathrm{d}b_i\,\mathrm{d}b_j} \tag{5}$$

and

$$\begin{aligned}\frac{\mathrm{d}^2R_n}{\mathrm{d}b_i\,\mathrm{d}b_j} &= \frac{\partial^2 R_n}{\partial b_i\,\partial b_j} + \frac{\partial^2 R_n}{\partial b_i\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \frac{\partial^2 R_n}{\partial U_k\,\partial b_j}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} + \frac{\partial^2 R_n}{\partial U_k\,\partial U_m}\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\frac{\mathrm{d}U_m}{\mathrm{d}b_j} + \frac{\partial R_n}{\partial U_k}\frac{\mathrm{d}^2U_k}{\mathrm{d}b_i\,\mathrm{d}b_j}\\ &= 0\end{aligned} \tag{6}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

It is a matter of derivation to obtain Equations (5) and (6) starting from Equations (1) and (2). Note that $d^2F/db_i\,db_j = d^2F/db_j\,db_i$ and $d^2R_n/db_i\,db_j = d^2R_n/db_j\,db_i$; hence, a prerequisite for the computation of $d^2F/db_i\,db_j$ using Equation (5) is the solution of $N(N+1)/2$ systems of equations, Equations (6), for $d^2U_k/db_i\,db_j$. By also adding the cost for computing dU_k/db_j, the use of Equations (3) and (4) requires $N + (N(N+1)/2)$ system solutions, in total. This cost is extremely high, making the so-called *direct–direct* approach inappropriate for the optimization of high-dimensional problems.

3. HESSIAN MATRIX COMPUTATION BASED ON ADJOINT APPROACHES

As mentioned in the Introduction, the use of the adjoint approach for the computation of the objective function gradient is in widespread use [9–11]. It suffices to formulate the gradient of the augmented functional $\overline{F}$, by multiplying the (zero) gradient of the residual of the flow equations with the Lagrange multipliers or adjoint variables $\mathbf{\Psi}$ and adding their product to the functional gradient. For each gradient component, we obtain

$$\frac{d\overline{F}}{db_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{db_i} + \Psi_m\left(\frac{\partial R_m}{\partial b_i} + \frac{\partial R_m}{\partial U_k}\frac{dU_k}{db_i}\right) \tag{7}$$

or

$$\frac{d\overline{F}}{db_i} = \frac{\partial F}{\partial b_i} + \Psi_m\frac{\partial R_m}{\partial b_i} + \left(\frac{\partial F}{\partial U_k} + \Psi_m\frac{\partial R_m}{\partial U_k}\right)\frac{dU_k}{db_i} \tag{8}$$

The terms in parenthesis are eliminated by satisfying the discrete adjoint equations

$$R_k^{\Psi} = \frac{\partial F}{\partial U_k} + \Psi_m\frac{\partial R_m}{\partial U_k} = 0 \tag{9}$$

and the functional gradient is, finally, given by

$$\frac{d\overline{F}}{db_i} = \frac{\partial F}{\partial b_i} + \Psi_m\frac{\partial R_m}{\partial b_i} \tag{10}$$

The use of the so-called adjoint approach reduces the CPU cost for the computation of dF/db_i, $i=1, N$, to that of solving the system of adjoint equations, i.e. one-system solution is required, irrespective of the number N of the design variables.

The extension of the adjoint approach to account for the computation of the Hessian matrix is possible. However, as it will be demonstrated below, there are at least three different ways to compute the Hessian matrix, fully or partially based on the adjoint approach. These are presented and discussed below.

The first scheme will be referred to as the *adjoint–direct* approach. It is based on the use of the adjoint formulation for the computation of first derivatives, Equations (9) and (10), followed by the direct approach for the computation of second derivatives. By differentiating Equation (10), we obtain

$$\frac{d^2\overline{F}}{db_i\,db_j} = \frac{\partial^2 F}{\partial b_i\,\partial b_j} + \frac{\partial^2 F}{\partial b_i\,\partial U_k}\frac{dU_k}{db_j} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial b_j} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial U_k}\frac{dU_k}{db_j} + \frac{d\Psi_m}{db_j}\frac{\partial R_m}{\partial b_i} \tag{11}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

The required derivatives $\mathrm{d}\Psi_m/\mathrm{d}b_j$ can be computed by zeroing the gradient of the residual of the adjoint equations, Equations (9), namely

$$\frac{\mathrm{d}R_n^{\Psi}}{\mathrm{d}b_j} = \frac{\partial^2 F}{\partial U_n\,\partial b_j} + \frac{\partial^2 F}{\partial U_n\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial b_j} + \Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \frac{\mathrm{d}\Psi_m}{\mathrm{d}b_j}\frac{\partial R_m}{\partial U_n} = 0 \tag{12}$$

The cost for computing $\mathrm{d}F/\mathrm{d}b_i$ and $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ using the *adjoint–direct* approach is equivalent to that of $1+2N$ system solutions; among them, N are needed for the computation of $\mathrm{d}U_k/\mathrm{d}b_j$, Equation (2), and N for $\mathrm{d}\Psi_m/\mathrm{d}b_j$, Equation (12).

The second scheme is the *adjoint–adjoint* approach that introduces extra adjoint variables, denoted by $\overline{\Psi}_{im}$ and $\overline{\overline{\Psi}}_{im}$. A twice-augmented objective function is first defined as

$$\frac{\mathrm{d}^2\overline{\overline{F}}}{\mathrm{d}b_i\,\mathrm{d}b_j} = \frac{\mathrm{d}^2\overline{F}}{\mathrm{d}b_i\,\mathrm{d}b_j} + \overline{\Psi}_{im}\frac{\mathrm{d}R_m}{\mathrm{d}b_j} + \overline{\overline{\Psi}}_{in}\frac{\mathrm{d}R_n^{\Psi}}{\mathrm{d}b_j} \tag{13}$$

which, after substituting the functional Hessian and the gradient of the flow and adjoint equations, yields

$$\begin{aligned}\frac{\mathrm{d}^2\overline{\overline{F}}}{\mathrm{d}b_i\,\mathrm{d}b_j} &= \frac{\partial^2 F}{\partial b_i\,\partial b_j} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial b_j} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial b_j} + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F}{\partial U_n\,\partial b_j} + \overline{\overline{\Psi}}_{in}\Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial b_j}\\ &\quad + \left(\frac{\partial^2 F}{\partial b_i\,\partial U_k} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial U_k} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial U_k} + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F}{\partial U_n\,\partial U_k} + \overline{\overline{\Psi}}_{in}\Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial U_k}\right)\frac{\mathrm{d}U_k}{\mathrm{d}b_j}\\ &\quad + \frac{\mathrm{d}\Psi_m}{\mathrm{d}b_j}\left(\frac{\partial R_m}{\partial b_i} + \overline{\overline{\Psi}}_{in}\frac{\partial R_m}{\partial U_n}\right)\end{aligned} \tag{14}$$

In Equation (14), the two terms in parenthesis are eliminated by satisfying the adjoint system of equations

$$\frac{\partial^2 F}{\partial b_i\,\partial U_k} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial U_k} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial U_k} + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F}{\partial U_n\,\partial U_k} + \overline{\overline{\Psi}}_{in}\Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial U_k} = 0 \tag{15}$$

and

$$\frac{\partial R_m}{\partial b_i} + \overline{\overline{\Psi}}_{in}\frac{\partial R_m}{\partial U_n} = 0 \tag{16}$$

One should, firstly, solve Equation (16) for $\overline{\overline{\Psi}}_{in}$ and, then, Equation (15) for $\overline{\Psi}_{im}$. The Hessian matrix is, then, given by

$$\frac{\mathrm{d}^2\overline{\overline{F}}}{\mathrm{d}b_i\,\mathrm{d}b_j} = \frac{\partial^2 F}{\partial b_i\,\partial b_j} + \Psi_m\frac{\partial^2 R_m}{\partial b_i\,\partial b_j} + \overline{\Psi}_{im}\frac{\partial R_m}{\partial b_j} + \overline{\overline{\Psi}}_{in}\frac{\partial^2 F}{\partial U_n\,\partial b_j} + \overline{\overline{\Psi}}_{in}\Psi_m\frac{\partial^2 R_m}{\partial U_n\,\partial b_j} \tag{17}$$

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. Both are less costly than the *direct–direct* approach, but as it will be shown in the next section, the *direct–adjoint* approach is much more efficient.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

4. THE DIRECT–ADJOINT APPROACH

A new augmented functional $\hat{F}$, with new adjoint variables $\hat{\Psi}_n$, is defined as follows:

$$\frac{\mathrm{d}^2\hat{F}}{\mathrm{d}b_i\,\mathrm{d}b_j} = \frac{\mathrm{d}^2F}{\mathrm{d}b_i\,\mathrm{d}b_j} + \hat{\Psi}_n\frac{\mathrm{d}^2R_n}{\mathrm{d}b_i\,\mathrm{d}b_j} \tag{18}$$

where $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ and $\mathrm{d}^2R_n/\mathrm{d}b_i\,\mathrm{d}b_j$ are given by Equations (5) and (6), respectively. After some rearrangements, we obtain

$$\begin{aligned}\frac{d^2\hat{F}}{\mathrm{d}b_i\,\mathrm{d}b_j} = {} & \frac{\partial^2F}{\partial b_i\,\partial b_j} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial b_i\,\partial b_j} + \frac{\partial^2F}{\partial U_k\,\partial U_m}\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\frac{\mathrm{d}U_m}{\mathrm{d}b_j} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial U_k\,\partial U_m}\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\frac{\mathrm{d}U_m}{\mathrm{d}b_j} \\ & + \frac{\partial^2F}{\partial b_i\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial b_i\,\partial U_k}\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \frac{\partial^2F}{\partial U_k\,\partial b_j}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial U_k\,\partial b_j}\frac{\mathrm{d}U_k}{\mathrm{d}b_i} \\ & + \left(\frac{\partial F}{\partial U_k} + \hat{\Psi}_n\frac{\partial R_n}{\partial U_k}\right)\frac{\mathrm{d}^2U_k}{\mathrm{d}b_i\,\mathrm{d}b_j}\end{aligned} \tag{19}$$

The last term in parenthesis is eliminated by satisfying the following adjoint equation:

$$\frac{\partial F}{\partial U_k} + \hat{\Psi}_n\frac{\partial R_n}{\partial U_k} = 0 \tag{20}$$

Thus, the Hessian matrix elements are given by

$$\begin{aligned}\frac{\mathrm{d}^2\hat{F}}{\mathrm{d}b_i\,\mathrm{d}b_j} = {} & \frac{\partial^2F}{\partial b_i\,\partial b_j} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial b_i\,\partial b_j} + \left(\frac{\partial^2F}{\partial U_k\,\partial U_m} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial U_k\,\partial U_m}\right)\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\frac{\mathrm{d}U_m}{\mathrm{d}b_j} \\ & + \left(\frac{\partial^2F}{\partial b_i\,\partial U_k} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial b_i\,\partial U_k}\right)\frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \left(\frac{\partial^2F}{\partial U_k\,\partial b_j} + \hat{\Psi}_n\frac{\partial^2R_n}{\partial U_k\,\partial b_j}\right)\frac{\mathrm{d}U_k}{\mathrm{d}b_i}\end{aligned} \tag{21}$$

The total computational cost is equal to that of N system solutions to compute $\mathrm{d}U_n/\mathrm{d}b_j$, Equation (2), plus one more to solve the adjoint equation, Equation (20) ($N+1$, in total). To summarize, in a single-objective optimization problem in aerodynamics, the CPU costs for computing the gradient and (exact) Hessian of the objective function, based on direct and/or adjoint approaches, are tabulated below:

Approach	Required system solutions (equivalent flow solver calls)
Direct–direct	$N + N(N+1)/2$
Direct–adjoint	$N + 1$
Adjoint–direct	$1 + 2N$
Adjoint–adjoint	$1 + 2N$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

The tabulated costs should be increased by one to account for the solution of the flow equations, summing up to the overall cost of a Newton cycle. After selecting the more efficient approach among them (which is, evidently, the *direct–adjoint* one), the remainder of this paper focuses on whether the use of the exact Hessian matrices in optimization methods, Equations (3) and (4), may reduce the overall optimization cost compared with other known descent methods.

5. AIRFOIL RECONSTRUCTION STUDIES—COMPARISONS AND DISCUSSION

The two applications that follow are concerned with airfoil shape reconstructions. These airfoils form 2D cascades with fixed pitch and stagger angle. The flow equations are numerically solved by means of a finite-volume, time-marching Euler/Navier–Stokes solver. Structured grids and a second-order upwind scheme for the convection terms are used. The target pressure distributions are computed using known airfoil shapes and the objective function is defined as the line integral of the deviation between the actual (p) and target (p_{tar}) pressure distribution over the airfoil contour (S_w):

$$F = \int_{S_w} (p - p_{\text{tar}})^2 \, \mathrm{d}S \tag{22}$$

Thus, upon convergence of the optimization method, comparisons between pressure distributions (primarily) and airfoil shapes (secondarily) are possible. The second case is examined twice: firstly, at inviscid and then at laminar flow conditions. We refrain from examining problems with turbulent flows since the frequently made assumption of freezing the turbulent viscosity coefficient in the adjoint formulation and the corresponding error may mislead any comparison, which is the basic scope of this paper.

As mentioned at the end of the previous section and according to the outcome of our theoretical investigation, the *direct–adjoint* approach (the other three approaches are also included for the purpose of comparison) is used to study the subsequent examples. We recall that the computation of $\mathrm{d}U_n/\mathrm{d}b_j$ requires the solution of N systems, Equation (2); these are followed by the solution of the adjoint system of equations, Equation (20), and the computation of the Hessian matrix, Equation (21), before updating the set of design variables using Equations (3) and (4). The solution of Equation (2) for $\mathrm{d}U_n/\mathrm{d}b_j$ and Equation (20) for $\hat{\Psi}_n$ is an exact process. However, the computation of the Hessian matrix, Equation (21), requires the computation of

$$\frac{\partial^2 R_n}{\partial b_i \, \partial U_k} \frac{\mathrm{d}U_k}{\mathrm{d}b_j} \quad \text{and} \quad \frac{\partial^2 R_n}{\partial U_k \, \partial U_m} \frac{\mathrm{d}U_k}{\mathrm{d}b_i} \frac{\mathrm{d}U_m}{\mathrm{d}b_j}$$

These computations are cumbersome and they increase memory requirements. For this reason, instead of Equation (21), the expression

$$\frac{\mathrm{d}^2 \hat{F}}{\mathrm{d}b_i \, \mathrm{d}b_j} = \frac{\partial^2 F}{\partial b_i \, \partial b_j} + \hat{\Psi}_n \frac{\partial^2 R_n}{\partial b_i \, \partial b_j} + \frac{\partial^2 F}{\partial b_i \, \partial U_k} \frac{\mathrm{d}U_k}{\mathrm{d}b_j} + \frac{\partial^2 F}{\partial U_k \, \partial b_j} \frac{\mathrm{d}U_k}{\mathrm{d}b_i} + \frac{\partial^2 F}{\partial U_k \, \partial U_m} \frac{\mathrm{d}U_k}{\mathrm{d}b_i} \frac{\mathrm{d}U_m}{\mathrm{d}b_j} \tag{23}$$

is used. This is harmless, in fact, as proved by the comparison of the values of second derivatives, computed using the presented approaches and finite differences. In addition, it should be noted that $\partial^2 F/\partial b_i \, \partial b_j$ is herein equal to zero, since parameterization is based on the linear Bézier–Bernstein

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

polynomials. Despite this simplification, we will refer to this approach as 'exact Hessian' or 'exact-Newton method' in order to distinguish it from the quasi-Newton (BFGS [25, 26]) scheme which is also used for the sake of comparison. Comparisons are made in terms of the number of cycles required for convergence as well as the number of calls to the flow and adjoint solver. The latter reflects on the total CPU cost required. However, in practice, the N direct calculations of the

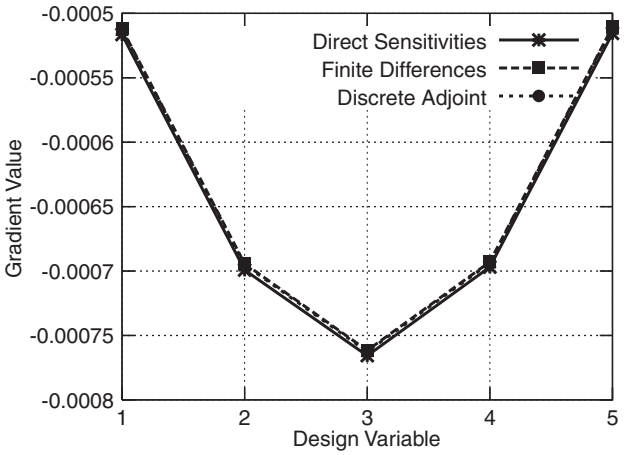

Figure 1. Symmetric cascade airfoil reconstruction; inviscid flow. Objective function gradient values computed for the initial airfoil, using the adjoint and direct approach as well as finite differences. The horizontal axis corresponds to the ordinates of the five control points parameterizing the lower of the (symmetric) airfoil sides.

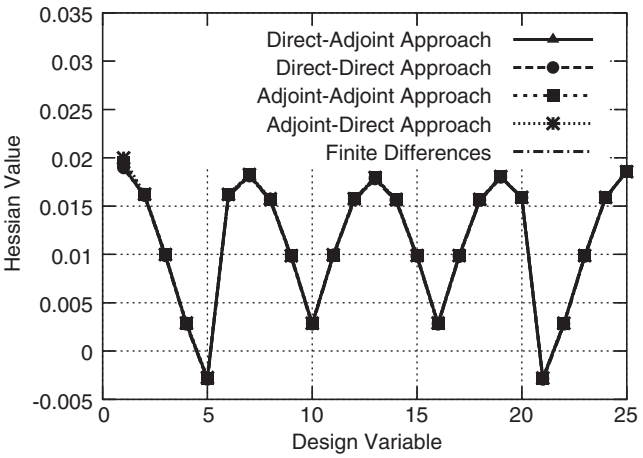

Figure 2. Symmetric cascade airfoil reconstruction; inviscid flow. Hessian matrix values computed using the *direct–adjoint*, *direct–direct*, *adjoint–adjoint*, *adjoint–direct* approaches and finite differences, for the five design variables of the lower side, as shown in Figure 1. The first five values correspond to the first row of $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ and so forth (5 columns $\times$ 5 rows $=$ 25 values; they are all shown here, although Hessian matrix is symmetric).

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

objective function gradient cost less than N flow solutions, since their left-hand side is exactly the same. This means that the cost of the Newton method is even less than that shown in figures.

5.1. Reconstruction of a 2D symmetric cascade

The first case is concerned with the reconstruction of a symmetric airfoil cascade. The flow is inviscid, with axial inlet ($\alpha_1 = 0$) and exit Mach number equal to $M_2 = 0.5$. Nine design variables

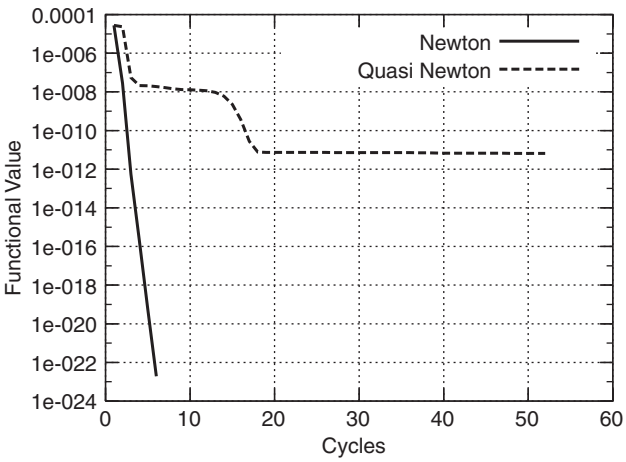

Figure 3. Symmetric cascade airfoil reconstruction; inviscid flow. Convergence rate of the inverse design functional using the exact and quasi-Newton algorithm. The horizontal axis corresponds to optimization cycles, although each curve bears different CPU cost per cycle.

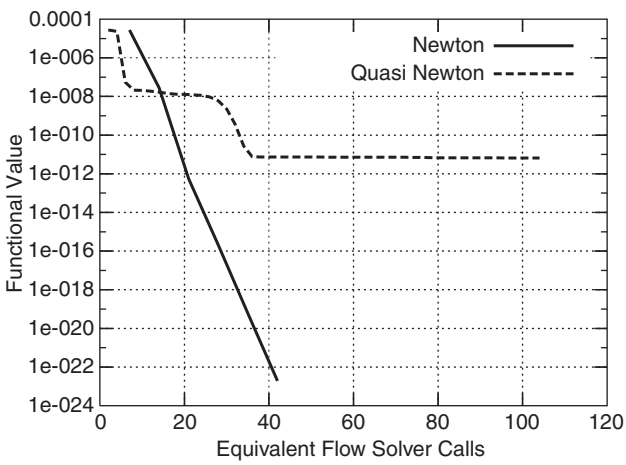

Figure 4. Symmetric cascade airfoil reconstruction; inviscid flow. Convergence rate of the inverse design functional using the exact and quasi-Newton algorithm. The horizontal axis corresponds to system solutions, i.e. equivalent flow solver calls and is proportional to the CPU cost.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

are used to parameterize each airfoil side, using Bézier–Bernstein polynomials. Five of them are allowed to vary, while the remaining ones are kept fixed at their known optimal values. The functional gradient values $\mathrm{d}F/\mathrm{d}b_i$ computed using the discrete adjoint approach, the direct approach and a central finite difference scheme are shown in Figure 1. The comparison of the Hessian matrix values $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ using the *direct–adjoint* approach, the other three more costly approaches and finite differences is shown in Figure 2. All value distributions are in excellent agreement.

The reduction rates of the functional value using the quasi-Newton (BFGS) and exact-Newton methods are shown in Figures 3 and 4. As shown in Figure 3, the exact-Newton method is run

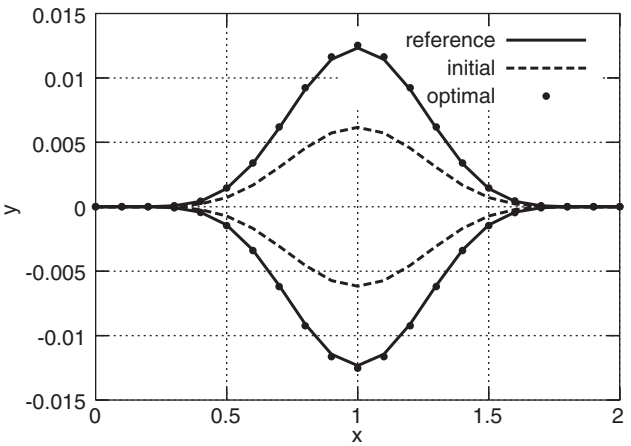

Figure 5. Symmetric cascade airfoil reconstruction; inviscid flow. Comparison of the initial and optimal airfoil contour with that used as reference shape (x, y axes not in scale).

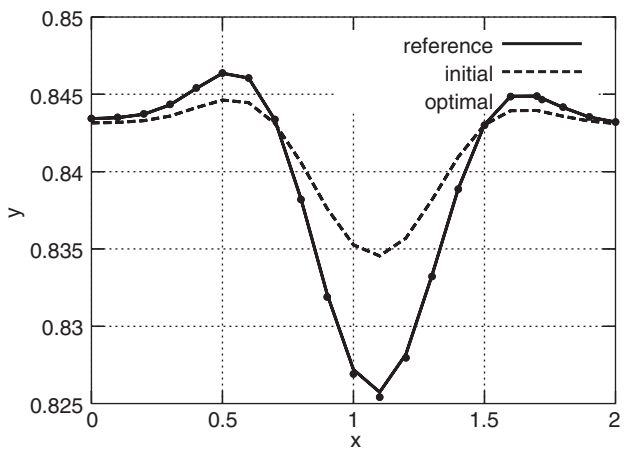

Figure 6. Symmetric cascade airfoil reconstruction; inviscid flow. Comparison of the initial, optimal and target pressure distributions over the lower airfoil side.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

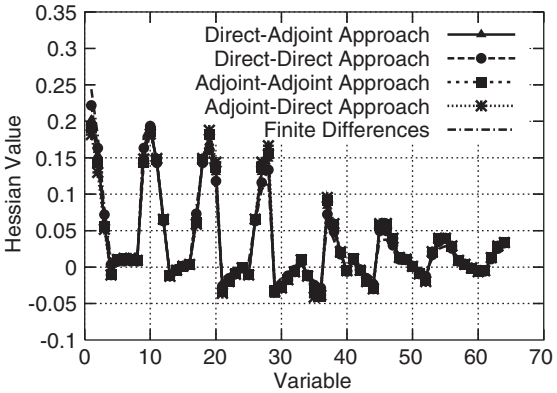

Figure 7. Compressor cascade airfoil reconstruction; inviscid flow. Hessian matrix values using the *direct–adjoint*, *direct–direct*, *adjoint–adjoint*, *adjoint–direct* approaches and finite differences. The horizontal axis corresponds to the ordinates of the eight control points (four per airfoil side). The first eight values correspond to the first row of $\mathrm{d}^2F/\mathrm{d}b_i\,\mathrm{d}b_j$ and so forth (8 columns $\times$ 8 rows $=$ 64 values).

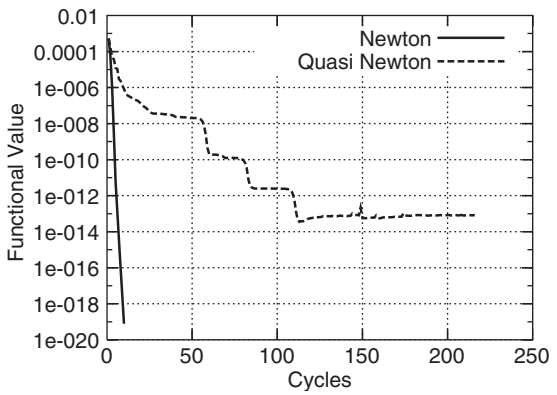

Figure 8. Compressor cascade airfoil reconstruction; inviscid flow. Convergence rate of the inverse design functional using the exact and quasi-Newton algorithms. In each method, the CPU cost per cycle is different, see Figure 9.

until at least 18 orders of magnitude drop in the value of F. This was achieved within only six cycles, while the quasi-Newton method was run until 7 orders of magnitude drop, within 50 cycles. The superiority of the Newton method is obvious, even in Figure 4, in which the x-axis scale is multiplied by the number of direct or adjoint calls (system solutions) per cycle (2 for the quasi-Newton method and 7 for the Newton method). The convergence rates of the steepest descent and conjugate gradient methods (not shown here) are even worse than that of the BFGS method. The initial, optimal and reference airfoil contours are shown in Figure 5, whereas in Figure 6 the initial and optimal pressure distributions are compared with the target one.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

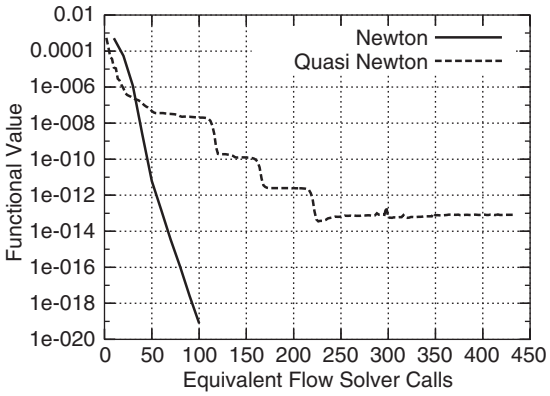

Figure 9. Compressor cascade airfoil reconstruction; inviscid flow. Convergence rate of the inverse design functional using the exact and quasi-Newton algorithms. For the horizontal axis, see caption of Figure 4.

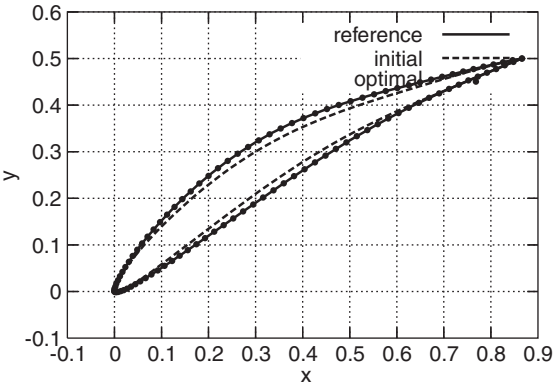

Figure 10. Compressor cascade airfoil reconstruction; inviscid flow. Comparison of the initial and optimal contour with that used as the reference contour for the reconstruction.

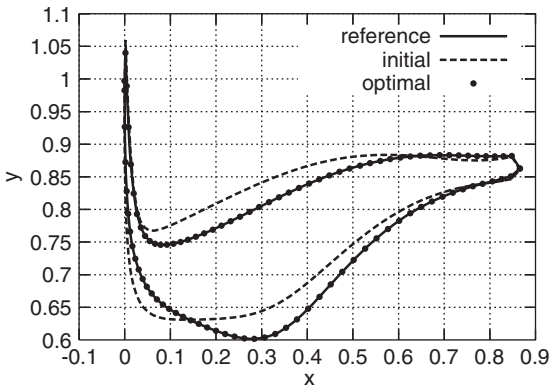

Figure 11. Compressor cascade airfoil reconstruction; inviscid flow. Comparison of the initial, optimal and target pressure distributions.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

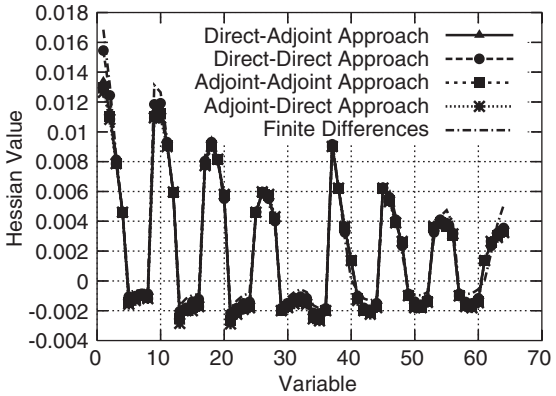

Figure 12. Compressor cascade airfoil reconstruction; laminar flow. Hessian matrix values using the *direct–adjoint*, *direct–direct*, *adjoint–adjoint*, *adjoint–direct* approaches and finite differences. For the horizontal axis, see caption of Figure 7.

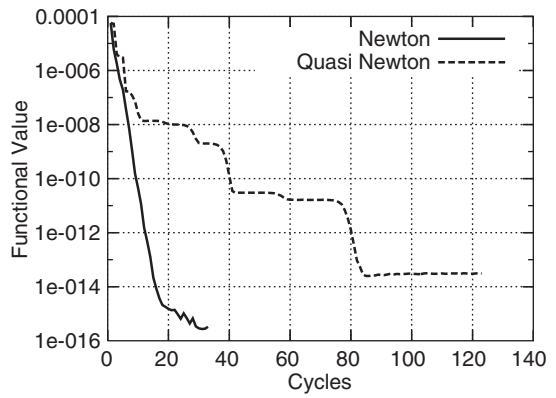

Figure 13. Compressor cascade airfoil reconstruction; laminar flow. Convergence rate of the inverse design functional using the exact and the quasi-Newton algorithms. In each method, the CPU cost per cycle is different, see Figure 9.

5.2. Reconstruction of a 2D Compressor Cascade

In this subsection, the reconstruction of a 2D compressor cascade airfoil, first at inviscid ($\alpha_1 = 47°$ and $M_{2,is} = 0.45$) and then at laminar ($Re_c = 2000$) flow, is analysed. Each airfoil side is parameterized using Bézier–Bernstein polynomials. Eight control points are allowed to vary only along the normal-to-the-chord direction.

The comparison of the Hessian matrix values computed using the proposed *direct–adjoint* method, the other three possible approaches and finite differences is illustrated in Figure 7. All curves are almost identical. In Figures 8 and 9, the convergence rates of the functional using the Newton and quasi-Newton methods prove that the use of the exact Hessian accelerates the convergence to the optimal solution, especially near the optimum. The initial and optimal airfoil

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

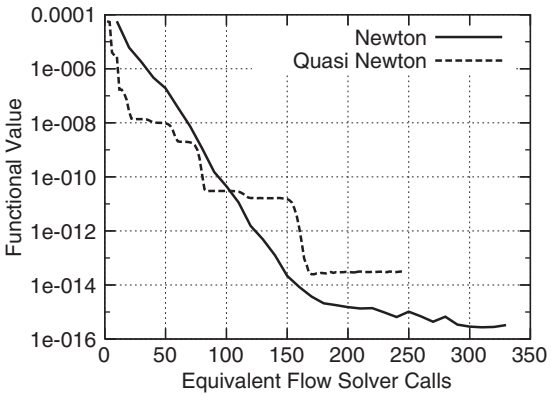

Figure 14. Compressor cascade airfoil reconstruction; laminar flow. Convergence rate of the inverse design functional using the exact and quasi-Newton algorithms. For the horizontal axis, see caption of Figure 4.

shapes are compared with the reference one in Figure 10. The corresponding pressure distributions are also compared in Figure 11. In either figure, the comparison is excellent.

The same conclusions can be drawn for the same case, if laminar flow is considered. The comparison of the Hessian matrix values computed using the four approaches presented in this paper and finite differences are shown in Figure 12, which shows that the curves are very much alike. The convergence of the quasi- and exact-Newton methods is also shown in Figures 13 and 14, where the superiority of the latter is obvious.

6. CONCLUSIONS

Four different methods for the computation of the Hessian matrix of functionals used in aerodynamic shape optimization problems were presented. Although this paper is concerned with the inverse design functional only, the proposed method can be generalized in a straightforward manner. The methods presented herein employ different approaches (direct or adjoint) to compute (a) the gradient and (b) the Hessian of the functional. The most efficient method proved to be a mixed one (*direct–adjoint*) that computes the gradient using the direct sensitivity approach and the Hessian matrix using the adjoint approach. The total computational cost per optimization cycle is proportional to the number of design variables. The convergence rate of the (exact) Newton method proved to considerably outperform that of other gradient-based algorithms, thus leading to lower overall optimization costs. This was demonstrated by analyzing the reconstruction of two cascade airfoils at inviscid and laminar flows, for which the Hessian matrix values were identical to those computed using finite differences.

REFERENCES

1. Pierce NA, Giles MB. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion* 2000; **65**(3–4):393–415.
2. Jameson A, Alonso JJ. Automatic aerodynamic optimization on distributed memory architectures. *AIAA 34th Aerospace Sciences Meeting and Exhibit*, Reno NV, January 1996; AIAA Paper 96-0409.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

3. Reuther J, Alonso JJ, Rimlinger MJ, Jameson A. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers and Fluids* 1999; **28**:675–700.
4. Nadarajah S, Jameson A, Alonso JJ. An adjoint method for the calculation of remote sensitivities in supersonic flow. *AIAA 40th Aerospace Sciences Meeting and Exhibit*, Reno NV, January 2002; AIAA-2002-0261.
5. Nadarajah S, McMullen M, Jameson A. Non-linear frequency domain based optimum shape design for unsteady three-dimensional flow. *AIAA Paper 2002-2838*, 2002.
6. Duta MC, Giles MB, Campobasso MS. The harmonic adjoint approach to unsteady turbomachinery design. *International Journal for Numerical Methods in Fluids* 2002; **40**(3–4):323–332.
7. Pironneau O. On optimum design in fluid mechanics. *Journal of Fluid Mechanics* 1974; **64**:97–110.
8. Jameson A. Aerodynamic design via control theory. *Journal of Scientific Computing* 1988; **3**:233–260.
9. Elliot J, Peraire J. Aerodynamic design using unstructured meshes. *Twenty Seventh Fluid Dynamics Conference*, New Orleans, LA, 17–20 June 1996; AIAA Paper 96-1941.
10. Anderson WK, Bonhaus DL. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal* 1999; **37**(2):185–191.
11. Giles MB, Pierce NA. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA Paper 97-1850*, 1997.
12. Jameson A. Optimum aerodynamic design using CFD and control theory. *AIAA 12th Computational Fluid Dynamics Conference*, San Diego, June 1995; AIAA Paper 95-1729.
13. Anderson WK, Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *AIAA Paper 97-0643*, 1997.
14. Kuruvila G, Taasan S, Salas MD. Airfoil design and optimization by the one-shot method. *AIAA Paper 95-0478*, 1995.
15. Hazra S, Schulz V, Brezillon J, Gauger N. Aerodynamic shape optimization using simultaneous pseudo-timestepping. *Journal of Computational Physics* 2005; **204**(1):46–64.
16. Mohammadi B, Pironneau O. *Applied Shape Optimization for Fluids*. Clarendon Press: Oxford, 2001.
17. Jameson A, Kim S. Reduction of the adjoint gradient formula in the continuous limit. *AIAA 41th Aerospace Sciences Meeting and Exhibit*, Reno NV, January 2003; AIAA-2003-0040.
18. Papadimitriou DI, Giannakoglou KC. A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. *Computers and Fluids* 2007; **36**:325–341.
19. Arian E, Taasan S. Analysis of the Hessian for aerodynamic optimization: inviscid flow. *Computers and Fluids* 1999; **28**(7):853–877.
20. Tortorelli D, Michaleris P. Design sensitivity analysis: overview and review. *Inverse Problems in Engineering* 1994; **1**:71–105.
21. Le Dimet FX, Navon IM, Daescu DN. Second-order information in data assimilation. *Monthly Weather Review* 2002; **130**(3):629–648.
22. Veerse F, Auroux D, Fisher M. Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. *Optimization and Engineering* 2000; **1**:323–339.
23. Daescu DN, Navon IM. An analysis of a hybrid optimization method for variational data assimilation. *International Journal of Computational Fluid Dynamics* 2003; **17**(4):299–306.
24. Jameson A, Pierce N, Martinelli L. Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics* 1998; **10**:213–237.
25. Fletcher R. *Practical Methods of Optimization* (2nd edn). Wiley: New York, 1988.
26. Bertsekas DP. *Nonlinear Programming* (2nd edn). Athena Scientific, 1999.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/fld