INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2001; **51**:735–753

CFD shape optimization using an incomplete-gradient adjoint formulation

O. Soto*,† and R. Löhner‡

SCS/Laboratory for Computational Fluid Dynamics, George Mason University, MS 4C7, 4400 University Drive, Fairfax, VA 22030-4444, U.S.A.

SUMMARY

A design methodology based on the adjoint approach for flow problems governed by the incompressible Euler equations is presented. The main feature of the algorithm is that it avoids solving the adjoint equations, which saves an important amount of CPU time. Furthermore, the methodology is general in the sense it does not depend on the geometry representation. All the grid points on the surface to be optimized can be chosen as design parameters. In addition, the methodology can be applied to any type of mesh. The partial derivatives of the flow equations with respect to the design parameters are computed by finite differences. In this way, this computation is independent of the numerical scheme employed to obtain the flow solution. Once the design parameters have been updated, the new solid surface is obtained with a pseudo-shell approach in such a way that local singularities, which can degrade or inhibit the convergence to the optimal solution, are avoided. Some 2D and 3D numerical examples are shown to demonstrate the proposed methodology. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: shape optimization; incompressible flow; adjoint formulation

1. INTRODUCTION

Genetic algorithms [1–4], approximate objective function surface schemes [5–8] and gradient-based methods [8–16] are usual methodologies found in the literature to solve optimization problems. The computational cost of each scheme is very different. In general, if the problem contains n design variables, a genetic algorithm must perform n^2 objective function evaluations to improve the original design. Schemes based on approximate surfaces, as well as gradient-based methods require $O(n)$ objective function evaluations per design cycle. Given

*Correspondence to: O. Soto, SCS/Laboratory for Computational Fluid Dynamics, George Mason University, MS 4C7, 4400 University Drive, Fairfax, VA 22030-4444, U.S.A.
†E-mail: soto@science.gmu.edu
‡E-mail: rlohner@rossini.gmu.edu

Contract/grant sponsor: CIMNE, Barcelona, Spain – partially
Contract/grant sponsor: NRL LCP&FD – partially

Received 3 May 2000
Revised 3 November 2000

that each objective function evaluation is equivalent to a CFD solution of the problem, these methods are only attractive for problems where objective functions can be easily evaluated. In general, a detailed flow solution is relatively expensive for problems governed by the Euler and RANS equations. The situation is not much better for so-called direct methods [8–16]. They require the solution of a large linear system of equations to compute the flow-variable gradient with respect to each design parameter. Thus, for n design parameters, n large linear systems have to be solved. This makes the method of order $O(n+1)$ ($n+1$ large linear system of equations have to be solved, one to evaluate the cost function and n to obtain its gradient). This cost can be reduced significantly if the LU decomposition of the matrix can be stored. However, this alternative is presently only possible for 2-D problems.

For problems involving many design parameters and few cost functions, a better alternative is to employ an adjoint formulation [8, 16–30]. In this approach, the effort to compute each cost-function gradient requires one CFD solution for the usual variables and one for the adjoint variables, if an unconstrained optimization is being performed or if the constraints and objective function can be combined into one cost function. Then, the cost is now only $O(2)$ CFD solutions per design cycle. The present article describes an approximate adjoint formulation for the incompressible Euler equations, and a methodology to employ such an approach in engineering design problems. The cost of this methodology is $O(1)$ CFD solutions per design cycle (!).

2. OPTIMIZATION PROBLEM

The optimization problem considered is minimizing (or maximizing) a cost function $I_c(U,\beta)$ that depends on the flow variables $U=(u,p)$, where u and p denote the velocity and pressure field, respectively, and on the physical location of the boundary, which is described by a set of design parameters $\beta=(\beta_1,\ldots,\beta_m)$.

The governing flow equations may be written as: $\mathscr{R}=(\mathscr{R}_u,\mathscr{R}_p)$, where $\mathscr{R}_u$ is the momentum equation for two or three-dimensional problems, and $\mathscr{R}_p$ refers to the conservation of mass. $\mathscr{R}$ expresses the implicit dependence of U and β in the flow domain Ω. For the present work, it is assumed that the steady flow is governed by the incompressible Euler equations, which can be written in conservative form as

$$\mathscr{R}_u=\nabla\cdot(u\otimes u)+\nabla p=0, \quad \text{and} \quad R_p=\nabla\cdot u=0 \tag{1}$$

with appropriate boundary conditions.

The flow equations can be thought of as a set of ‘restrictions’associated with the optimization problem, which must be fulfilled by the optimal solution. Following this idea, the Euler equations are added to the cost function by introducing a set of Lagrangian multipliers, or co-state variables, $\Psi=(\psi_u,\psi_p)=(\psi_1,\psi_2,\psi_3,\psi_p)$.This set of variables enforces in a weak form the restrictions imposed by the flow equations. The cost function is then given by:

$$I=I_c+\int_\Omega \Psi\cdot\mathscr{R}\,d\Omega \tag{2}$$

The necessary conditions for an optimal point of the minimization (or maximization) problem are

$$\text{(a) } \mathscr{R}(u, p) = 0, \quad \text{(b) } \frac{\partial I}{\partial U} = 0, \quad \text{(c) } \frac{\partial I}{\partial \beta} = 0 \tag{3}$$

The optimization process proceeds by solving the first two equations in a staggered manner, and updating the design parameters until the last one is satisfied. Given a set of initial values for the design parameters, the Equation (3)a is solved to find the respective velocity and pressure fields. Then, with β fixed (which is the same as Ω fixed), and with the computed velocity and pressure fields, the Equation (3)b is solved to obtain the set of Lagrange multipliers Ψ. Finally, the desired gradient is obtained from the Equation (3)c, and the design parameters are updated with some optimization algorithm (steepest descent method, conjugate gradient method, Newton type method, etc.). The cycle is repeated until some convergence criterion for Equation (3)c is achieved.

3. FLOW SOLUTION

The solution of Equation (3a) can be performed with several methods (projection, artificial compressibility, GLS, etc.). Two different schemes were employed for the present work.

The first was an implicit standard Galerkin-Least-squares (GLS) finite element method, in which the Euler equations are solved using a backward Euler scheme for the time discretization and a Picard linearization until steady state. The resulting non-symmetric and non-definite system of equations is solved using a standard GMRES algorithm [31]. The details of the flow formulation may be found in References [32–38].

The second scheme is an equal-order projection-type finite element scheme [39, 40]. The convective term is integrated explicitly using an edge-based second order upwind scheme with MUSCL limiting [41]. The pressure is integrated implicitly by solving a Poisson equation with fourth order damping for the divergence constraint.

4. ADJOINT EQUATION

Equation (3b) can be expanded as:

$$\frac{\partial I_c}{\partial U_k} + \frac{\partial}{\partial U_k}\left(\int_\Omega \psi_l \mathscr{R}_l \, d\Omega\right) = 0 \tag{4}$$

Multiplying the above equation by an arbitrary δU_k, taking into account that the domain is fixed (the derivatives are at constant β), and given that $\mathscr{R}(u, p) = 0$, the following equation is obtained:

$$\frac{\partial I_c}{\partial U_k}\delta U_k + \int_\Omega \psi_l \frac{\partial \mathscr{R}_l}{\partial U_k}\delta U_k \, d\Omega = 0 \tag{5}$$

The term $(\partial \mathscr{R}_l / \partial U_k)\delta U_k$ can be approximated to order $(\delta U_k)^2$ as

$$\frac{\partial \mathscr{R}_u}{\partial U}\delta U = \nabla \cdot (\delta u \otimes u) + \nabla \cdot (u \otimes \delta u) + \nabla \delta p, \quad \frac{\partial R_p}{\partial U}\delta U = \nabla \cdot \delta u \tag{6}$$

where δu and δp are admissible perturbations of the velocity and pressure fields, respectively. Introducing (6) in (5), integrating by parts in such a way that the terms containing derivatives of δu and δp are cancelled, and splitting the resulting equation by taking the functional basis $(\delta u, 0)$ and $(0, \delta p)$ to approximate δU, the following expressions are obtained

$$\int_\Omega \delta u \cdot (u \cdot \nabla)\psi_u \, d\Omega + \int_\Omega \delta u \cdot (\nabla \psi_u \cdot u) \, d\Omega + \int_\Omega \delta u \cdot \nabla \psi_p \, d\Omega$$
$$= \delta u \cdot \frac{\partial I_c}{\partial u} + \int_\Gamma \psi_p \delta u \cdot n \, d\Gamma + \int_\Gamma (\psi_u \cdot \delta u)(u \cdot n) \, d\Gamma + \int_\Gamma (\psi_u \cdot u)(\delta u \cdot n) \, d\Gamma, \quad \forall \delta u \tag{7}$$

$$\int_\Omega \delta p \nabla \cdot \psi_u \, d\Omega = \delta p \frac{\partial I_c}{\partial p} + \int_\Gamma \delta p \psi_u \cdot n \, d\Gamma, \quad \forall \delta p \tag{8}$$

where Γ is the boundary of the domain Ω and n its exterior normal.

If the objective function I_c is only defined on the boundary, and if one takes into account that these equations have to be fulfilled $\forall \delta u$ and $\forall \delta p$, then Equations (7) and (8) are the corresponding 'weak' or variational form of the PDE problem:

$$(u \cdot \nabla)\psi_u + \nabla \psi_u \cdot u + \nabla \psi_p = 0, \quad \text{and} \quad \nabla \cdot \psi_u = 0 \tag{9}$$

with boundary conditions defined in such a way that the right-hand side terms in Equations (7) and (8) are cancelled. In most of the interesting problems, I_c does not depend on u, and the boundary integrals in Equation (7) are equal to zero (in the far field $\delta u = 0$, and on the solid boundaries $u \cdot n = 0$ and $\delta u \cdot n = 0$). Then, the right-hand side of Equation (7) cancels automatically and no boundary conditions are necessary. Therefore, the boundary conditions of problem (9) are deduced from the right-hand side terms of Equation (8). In general, the cost function I_c is defined over the solid boundaries and depends on the pressure field and the boundary shape. Hence, Equation (8) can be rewritten as

$$\int_\Omega \delta p \nabla \cdot \psi_u \, d\Omega = \int_\Gamma \delta p \frac{\partial F(p)}{\partial p} G(\beta) \, d\Gamma + \int_\Gamma \delta p \psi_u \cdot n \, d\Gamma, \quad \forall \delta p \tag{10}$$

where F is a function of the pressure p, and G a function of the boundary shape defined by β. Finally, the boundary conditions for Equation (9) are given by cancelling the two boundary terms of Equation (10). This yields the following boundary condition:

$$\psi_u \cdot n = -\frac{\partial F(p)}{\partial p} G(\beta) \quad \text{on } \Gamma_s \tag{11}$$

where Γ_s is the part of the solid boundary where the cost function is defined. In the far field the right-hand side of Equation (10) is automatically satisfied, and no boundary conditions have to be imposed.

In summary, the adjoint problem is defined by (9), with the boundary condition given by (11). At this point it is important to remark that, if the incompressible Euler equations are written in the well-known advective form (this is easily verified by writing Equation (1) in a Cartesian system):

$$A_k(u, p)\frac{\partial U}{\partial x_k} \tag{12}$$

where A_k are matrices of 4×4 in 3D, and the adjoint problem is written in the same manner, one obtains:

$$A_k^{\mathrm{T}}(u, p)\frac{\partial \Psi}{\partial x_k} \tag{13}$$

where A_k^{T} is the transpose of A_k. This feature has been used by some authors to choose a stable algorithm for the solution of the adjoint problem, i.e. [24, 25].

The adjoint equations can be discretized using the same schemes employed for the flow solution, or whatever scheme that stabilizes the convective and the incompressible part of the problem (9) (i.e. GLS, projection schemes, artificial compressibility schemes, etc.).

However, as will be shown below, the adjoint solution can be avoided by using an Incomplete-Gradient approach. In Reference [42], this type of approach, neglecting the contribution of the adjoint variables to the total cost-function gradient, has already been employed. The required sensitivity was simply approximated as the gradient of I_c with respect to β (see Equation (2)). Nevertheless, the adjoint contribution to the sensitivity may be important and it can accelerate the convergence of the design problem. For this reason, a part of this contribution may be taken into account with little computational effort, as shown below.

5. COMPUTATION OF SENSITIVITIES

The cost-function gradient with respect to the design variables is computed according to Equation (3), which can be written as:

$$\frac{\partial I}{\partial \beta} = \frac{\partial I_c}{\partial \beta} + \frac{\partial}{\partial \beta}\left(\int_\Omega \Psi \cdot \mathscr{R}\, \mathrm{d}\Omega\right) \tag{14}$$

The desired optimal solution is obtained when the gradient is equal to zero.

Equation (14) may be evaluated in a variety of ways, e.g. finite differences, automatic differentiation, exact differentiation using flow and/or geometry parametrization, etc. References [16, 20, 22–30]. In the present work, both finite differences and exact differentiation of the numerical flow schemes were studied. The resulting derivatives were practically the same for both methods. In addition, a sensitivity analysis for the step size ε of the finite difference scheme was done, and it was found that for step sizes small enough (step sizes smaller than 10^{-4}) the value of the gradient was practically constant. The finite difference approach was chosen to compute all the terms of Equation (14) as it offers the following additional advantages: simplicity, independence of the problem dimension (2D or 3D), and independence of the flow solver employed. However, the finite-difference sensitivity analysis was only performed for grids with isotropic elements (inviscid flows or low-*Re* flows). It is possible that for highly stretched grids an exact differentiation of the numerical flow scheme will have to be used.

To approximate Equation (14) by finite differences, the steps presented in Algorithm 1 (see box below) have to be implemented.

Algorithm 1

(i) Obtain Ψ from Equation (9) with the boundary condition (11).
(ii) Evaluate $I = I_c + \int_\Omega \psi \cdot \mathscr{R}\, d\Omega$.
(iii) Perturb the co-ordinates of a design variable β_k in the external normal direction by a small ε.
(iv) Move (or smooth) the boundary Γ_s according to the design variable perturbation. This is, the coordinates of the grid boundary points on Γ_s must depend on the design variable positions by some interpolating curve or surface (i.e. splines, B-splines, Nurbs, etc,), or some smoothing operator. Then, if β_k is perturbed, Γ_s has to be moved, and a new boundary Γ_s' and domain Ω' are obtained.
(v) Move the volume mesh according to the boundary movement. Again, the volume grid points have to be moved using some mesh movement strategy that depends on the boundary movement.
(vi) Evaluate $I' = I_c' + \int_\Omega' \psi \cdot \mathscr{R}\, d\Omega$ where I_c' is I_c evaluated on Γ_s'.
(vii) The desired sensitivity is obtained as: $\partial I/\partial \beta_k = (I' - I)/\varepsilon$.
(viii) Repeat (ii) to (vii) for each design point.

It was found that the procedure outlined in the mentioned algorithm yields a very good approximation to the true gradient. Numerical experiments have shown that it is practically equal to the gradient obtained by finite differences if the mesh is fine enough (mesh size h tending to zero). However, this procedure is very expensive if the number of design variables is high.

A considerable simplification is obtained by taking into account that the main contribution of the adjoint terms to the gradient are at the boundary Γ_s. In the interior of the domain, the adjoint variables quickly decrease to zero and the movement of the grid points due to a boundary movement decreases rapidly with the distance to the boundary. This will not be the case if the flow variables are discontinuous or if highly stretched elements are present in the mesh [24]. For these cases a complete gradient has to be computed (i.e. by using Algorithm 1). However, for the class of problems considered here (incompressible flows, isotropic grids), these assumptions are valid.

With these arguments in mind, the sensitivity can be approximated by:

$$\frac{\partial I}{\partial \beta} = \frac{\partial I_c}{\partial \beta} + \frac{\partial}{\partial x_s}\left(\int_\Omega \psi \cdot \mathscr{R}\, d\Omega\right)\frac{\partial x_s}{\partial \beta} + \frac{\partial}{\partial x_i}\left(\int_\Omega \psi \cdot \mathscr{R}\, d\Omega\right)\frac{\partial x_i}{\partial \beta} \tag{15}$$

where x_s are the positions of the grid points on Γ_s and x_i the positions for the rest of the points. Then, the third term of Equation (15) can be neglected by taking the value of $\psi = 0$ at x_i (which is close to the real behaviour of Ψ), and by assuming a rapid decrease in the movement of the interior points x_i due to a boundary movement $\partial x_i/\partial \beta \approx 0$. Then, the adjoint gradient can be approximated only from its contributions near Γ_s, i.e. for the first row of

element contributions, as:

$$\frac{\partial I_\psi}{\partial \beta} \approx \frac{\partial}{\partial \beta} \int_\Omega \psi_u \cdot ((\mathscr{R}_u \cdot n)n + (\mathscr{R}_u \cdot t)t)\, \mathrm{d}\Omega \tag{16}$$

where n is the exterior normal on Γ_s, and t the tangential direction co-planar with $\mathscr{R}_u$ and n. Taking into account that the boundary perturbation is in the normal direction of Γ_s, the important change in $\mathscr{R}_u$ should be in that direction, and, therefore the tangential part can be neglected, i.e.

$$\frac{\partial}{\partial \beta} \int_\Omega \psi_u \cdot ((\mathscr{R}_u \cdot t)t)\, \mathrm{d}\Omega \approx 0 \tag{17}$$

Numerical experience has verified the correctness of the simplifications outlined above. In summary, an *Incomplete Gradient* can be obtained from the Algorithm 2 [43] (see box below).

Algorithm 2

(i) Obtain only $\psi_n = \psi_u \cdot n$ on the boundary by using the condition (11).
(ii) Evaluate $I = I_c + \int_\Omega \psi_n \cdot \mathscr{R}\, \mathrm{d}\Omega$.
(iii) Perturb the co-ordinates of a design variable β_k in the external normal direction by a small ε.
(iv) Move (or smooth) the boundary Γ_s according to the design variable perturbation (the same as step (iv) in Algorithm 1).
(v) Evaluate $I' = I'_c + \int_{\Omega'} \psi_n \cdot \mathscr{R}\, \mathrm{d}\Omega$.
(vi) Then, the desired sensitivity is obtained from the step (vii) of Algorithm 1.
(vii) Repeat (ii)–(vi) for each design point.

The scheme presented in the mentioned algorithm avoids the necessity of solving the adjoint problem, and of moving the entire volume mesh for each design parameter perturbation. In addition, a very fast method may be implemented to move (or smooth) the boundary due to a design variable perturbation (step (iv) of Algorithm 2).

As a final remark, the first-order forward finite differences shown in both algorithms (steps (vii) and (vi) of Algorithms 1 and 2, respectively), may be replaced by a second-order central differences in a straightforward manner. This has been the scheme adopted in this work.

6. MESH MOVEMENT

Once the sensitivities have been computed, a standard steepest descendent method is used to update the co-ordinates of each design parameter:

$$x'_j = x_j - \lambda \frac{\partial I}{\partial \beta_j} \tag{18}$$

where x_j' will be the new co-ordinate vector of the design parameter β_j, x_j the old one, λ a positive real constant and $\partial I/\partial \beta_j$ the required sensitivity computed as explained above.

The surface defined by the new positions x_j' is obtained by solving the following problem on Γ_s:

$$\theta - \nabla w = 0, \quad \nabla \cdot \theta = 0 \tag{19}$$

where w is the normal displacement of each point on Γ_s, and θ its rotation vector. The boundary conditions for these equations are given by the new design parameter positions, and by geometric restrictions (i.e. symmetry planes, fixed points, given rotations, etc.). This scheme can be seen as a simplification of a solid-shell formulation on Γ_s (or beam in 2D). The Equation (19) and their boundary conditions, are the strong form of the variational problem of a infinitely thin shell [44]. The scheme has the desired property of producing a continuous normal displacement field and also a continuous rotation field. This last property avoids local singularities on the new surface. Finally, with the normal displacements w obtained from Equation (19), the on-plane displacements are smoothed by solving the problem:

$$\Delta d = 0 \tag{20}$$

where d is the total displacement field on Γ_s. The boundary condition for this problem are given by the normal displacements w already obtained, and the geometrical restrictions of the problem. This last step avoids the distortion of the surface mesh over Γ_s in the case that large normal displacements appear. The solution of Equations (19) and (20) are performed by using a direct LU decomposition. The LU matrices are computed and saved only once at the beginning. The computational cost of the shell problem over Γ_s is similar to the cost of a 2D problem. The total surface reconstruction due to a normal perturbation on Γ_s is obtained by a very fast backward and forward substitutions. Therefore, the CPU time to obtain the gradient (14) by finite differences is very low.

Finally, the surface (or line in 2D) is smoothed following the procedure presented in Reference [45], and all the volume mesh points are updated to fix the new boundary. The technique implemented to do this was taken from Reference [46]. It is based on the solution of a Laplacian of the mesh displacements with variable diffusivity depending on the distance from the moving boundary. This procedure decreases element distortion considerably, reducing the need for local or global remeshing, and in most cases avoiding it altogether.

7. NUMERICAL EXAMPLES

7.1. 2-D bump

The first numerical example consist of matching a defined pressure distribution over a two-dimensional bump, which was built using a B-Spline curve. The target pressure was determined by computing the Euler equations around the bump. The boundary conditions were: At inflow $u = (1, 0)$, at outflow $p = 0$ and symmetry conditions along the line $y = 0$. In this example, the incomplete-gradient procedure was used (i.e. the adjoint equation was not solved, see Algorithm 2). For comparison, the example was also performed using the complete-gradient formulation (i.e. the adjoint equation was solved, see Algorithm 1). In this, and in all the

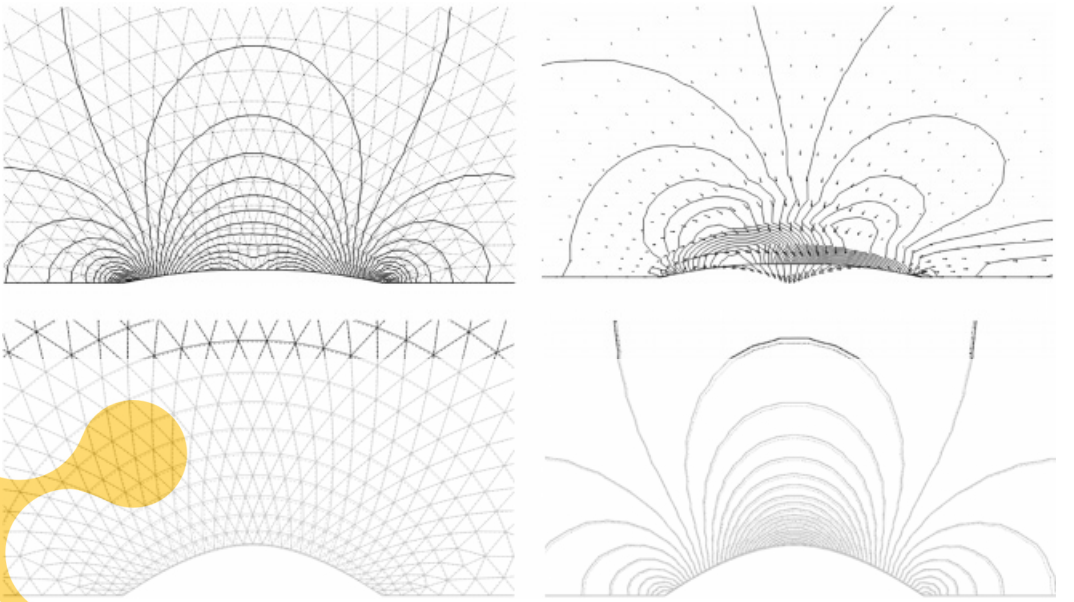

Figure 1. From left to right and top to bottom: Initial mesh and pressure contours (peaks: −0.1393, 0.1432). Adjoint solution on the initial geometry ψ_u (vectors) and ψ_p (contours, peaks: −1.682, 1.721). Target mesh (continuous line) and final mesh (dashed line). Target pressure (continuous line, peaks: −0.7312, 0.3599) and final pressure (dashed line, peaks: −0.7281, 0.3537).

following 2D examples, the solid boundary was rebuilt after each design cycle using the pseudo-shell (beam) approach described by Equations (19).

The adjoint solution was obtained as

$$I_c = \int_{\Gamma_s} (p_d - p)^2 \, d\Gamma \tag{21}$$

where p_d denotes the target pressure and Γ_s the bump surface. In Figure 1 the initial geometry and pressure field, the adjoint solution on the initial geometry, the target and final meshes, and the target and final pressure fields are shown. The results were obtained using one design variable in the middle of the solid surface. The design process was performed two times using a different number of design variables, obtaining similar results for all the cases. The first optimization cycles were done using one design variable in the middle of the solid boundary. The last design process was performed using all the nodal points on the solid surface as design variables.

In Figure 2 the evolution of the cost function is presented. It is important to note that the value of I_c always decreases. However, the convergence to the optimum is faster using a small number of design variables. This behaviour was already noted in Reference [25]. In addition, to perform the optimization process using all the nodal points on Γ_s as design parameters, the solid surface was smoothed after each optimization cycle, following the procedure presented in Reference [44]. This smoothing may deteriorate the convergence (the design parameters were moved not only following the gradient computation, but also the smoothing procedure).

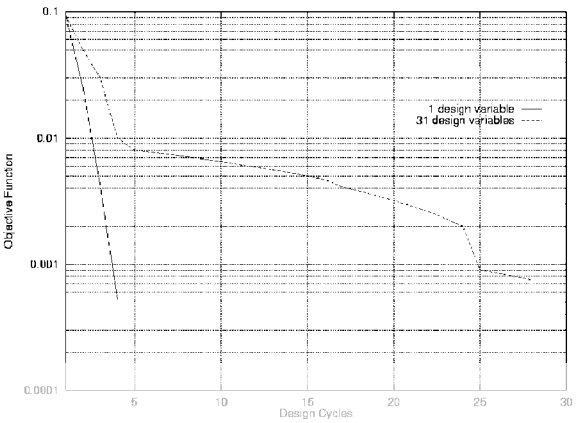

Figure 2. Cost function evolution (Design cycles vs. Cost function).

Table I. Comparison between the incomplete gradient, complete gradient and finite difference (FD) gradient for the one design variable case.

Design cycle	Incomplete gradient	Complete gradient	FD gradient
1	+0.07197557	+0.05905559	+0.563175
2	+0.20084590	+0.11986370	+0.264075
3	−0.26925220	−0.21840380	−0.179846
4	−0.03114764	−0.03867715	−0.039663

In Table I a comparison between the incomplete gradient (i.e. adjoint problem not solved, Algorithm 2), the complete gradient (i.e. adjoint problem (9) solved and interior mesh sensitivities taken into account, Algorithm 1), and the finite difference (FD) gradient is shown. The comparison was done for the case of one design variable in the middle of the solid surface. It is important to remark that, although the gradients are not equal, their signs are. Therefore, a steepest descendent optimization method will converge to the same result using any of the three schemes. This result corroborates the hypothesis given in Section 5 of this paper, stating that the main contribution to the gradient comes from the solid boundary. Note also that incomplete gradients approximate complete gradients and finite difference gradients once convergence is achieved.

7.2. 2D hydrofoil optimization

The objective of the example is to maximize the minimum pressure over a hydrofoil at a fixed lift. This type of optimization objective is often encountered in hydrodynamics, where cavitation is always a concern. The initial hydrofoil profile is a NACA0012, and 16 equally spaced design variables were used for the design process. The design points at the leading and trailing edges remain fixed along the optimization cycles to avoid rigid body movements of the hydrofoil. The angle of attack of the flow was fixed at a value of 5°. The cost

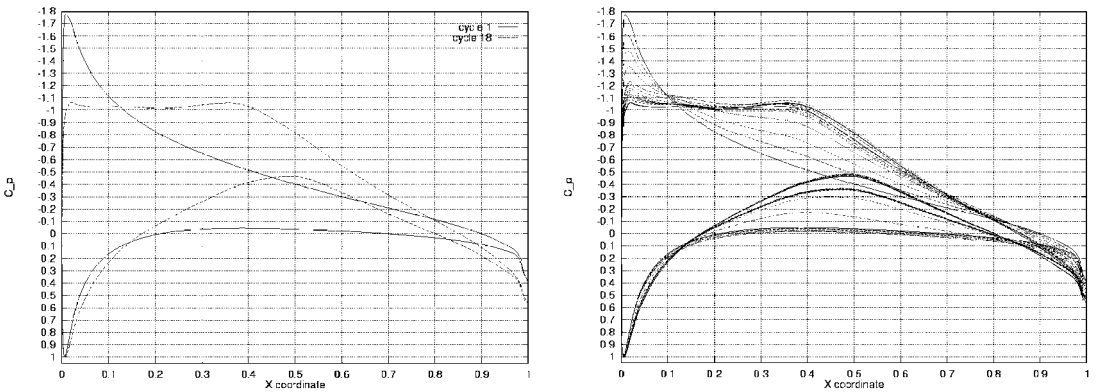

Figure 3. From left to right: Pressure coefficient distribution for the initial and final hydrofoils. Evolution of the pressure coefficient along the optimization process.

function for this example was defined as follows:

$$I_c = w_1 \left(\int_{\Gamma_s} p n_y \, d\Gamma - C_L^* \right) + w_2 \int_{\Gamma_s} \left| \frac{\partial p}{\partial t} \right| d\Gamma \tag{22}$$

where Γ_s is the hydrofoil boundary, w_1 and w_2 are the cost function weights, p is the pressure, n_y the vertical component of the normal vector along the Γ_s, C_L^* the fixed lift (computed for the initial configuration), and t the tangential vector along the hydrofoil boundary. The first term of Equation (22) enforces a fixed lift, while the second term assures an optimally smooth pressure gradient along the hydrofoil. Given that the stagnation pressure is fixed by the external flow, this last condition indirectly assures that the minimum pressure on Γ_s increases as the design progresses.

The optimization was done as follows: First w_1 was set to zero and w_2 was set to one, and several design cycles were performed until the objective function stalled. Five design cycles were needed to maximize the minimum pressure. Clearly, during these first cycles the lift restriction was violated (the value of the lift restriction increased from 0 to 0.0031). Then, w_1 was set to one and w_2 to zero, and another set of design cycles was performed until the lift restriction held (3 design cycles). This procedure was repeated two times.

The initial minimum pressure coefficient computed along the initial hydrofoil profile was $c_p = -1.7768$. At the end of the design cycles, c_p had increased to $c_p = -1.0642$ (see Figure 3). This represents a 40.1 per cent of improvement. In addition, the value of the lift restriction (first term of Equation (22)) at the end of the design cycles was 2.7×10^{-6}. In Figure 4 the evolution of the hydrofoil profiles along the design cycles and the pressure distribution for the optimum can be observed.

7.3. 2D forebody optimization

The objective of the example is to maximize the minimum pressure over the forward part of a 2D hull, in order to avoid possible cavitation. The forebody length was set to 5.0, and its width to 1.0 (see Figure 5). Five equally spaced design points were placed on the hull.

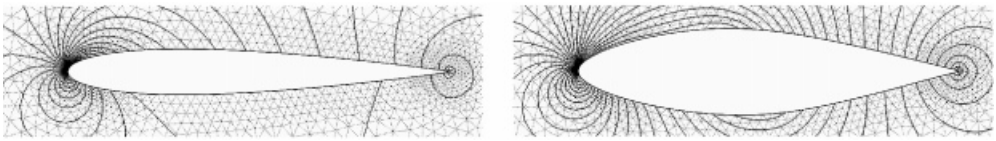

Figure 4. From left to right: Mesh and pressure distribution for the initial hydrofoil (peaks: $-0.8884, 0.5008$). Mesh and pressure distribution for the optimum (final) hydrofoil (peaks: $-0.5321, 0.5020$).

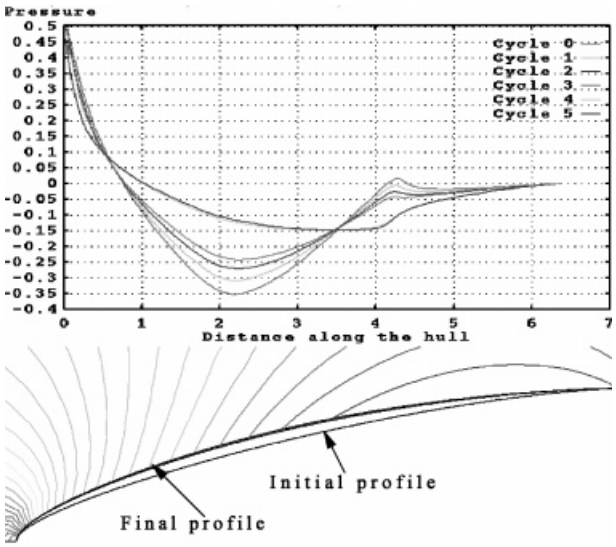

Figure 5. Top: Evolution of the pressure coefficient on the hull profile along the optimization process. Bottom: Evolution of the hull profiles, and pressure distribution for the optimum hull (contours from -0.15 to 0.5).

Vanishing slope condition were enforced at both hull ends. Symmetry conditions on the flow variables were imposed at the line $y = 0$. The objective function was defined as follows:

$$I_c = \int_{\Gamma_s} \left| \frac{\partial p}{\partial t} \right| \, d\Gamma \tag{23}$$

The initial minimum pressure computed over the hull was $p = -0.35$, and it was increased to $p = -0.15$ at the end of six design cycles (57 per cent of improvement). Figure 5 summarizes these results.

7.4. 3D hydrofoil matching pressure

The first 3D numerical example consists of matching a computed pressure distribution over a three-dimensional hydrofoil. The target hydrofoil cross-section is a NACA0012 with a span length of 0.5, and the target pressure was computed at an angle of attack of 5°.

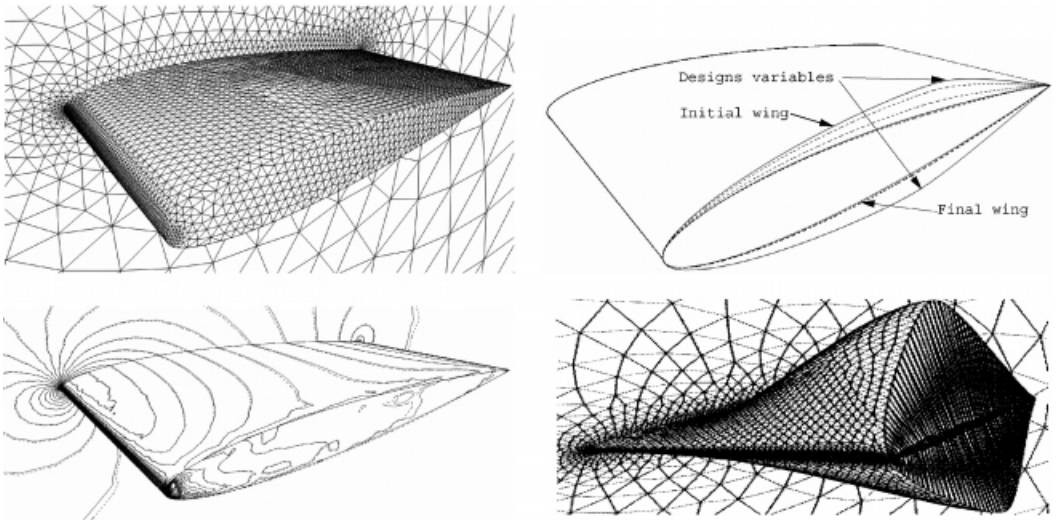

Figure 6. From left to right and top to bottom: Target geometry and surface mesh. Shape evolution through the design cycles. Target pressure distribution (continuous contours) and pressure distribution for the last design cycle (dashed contours); peaks of pressure $(-0.67, 0.50)$. Example of surface deformation using the pseudo-shell approach.

Symmetry conditions were prescribed in the plane $z = 0$. In Figure 6(a) the hydrofoil target geometry and surface mesh can be observed. The volume mesh consists of 28,901 nodes and 151,457 P1/P1 elements. The objective function was defined by Equation (21).

To start the design, the cross-section was perturbed by 100 per cent of its maximum thickness. The leading and trailing hydrofoil edges were fixed during the entire design process. The optimization procedure was carried out using two design variables, one in the top surface and the other in the bottom one (see Figure 6(b)). The pseudo-shell parametrization described in Equations (19) and (20) was used to rebuild the surface from the new design variable positions (see Equation (18)). Five design cycles were necessary to reduce the objective function three orders of magnitude. In Figure 6(b) the evolution of the hydrofoil shape through the design cycles is shown. The final hydrofoil shape matches very well the target one. Figure 6(c) presents the target and final pressure distributions. To show an example of the shape reconstruction procedure, in Figure 6(d) a solution of the pseudo-shell Equations (19) and (20) over the hydrofoil surface is given. A normal deformation of ten times the thickness of the wing was imposed at the design variable locations, and symmetry conditions (zero rotation in z-direction) at vertical planes. Note the smoothness of the normal deformation, and the continuity of its gradient (rotations). The cost of building the surface is less than one CPU second on a Silicon R10 000 single processor.

Finally, in Figure 7 the objective function evolution through the design cycles is presented. The CPU cost of solving the CFD problem in each design cycle was approximately 15 times higher than the cost of computing the gradients and rebuilding the new wing surface.

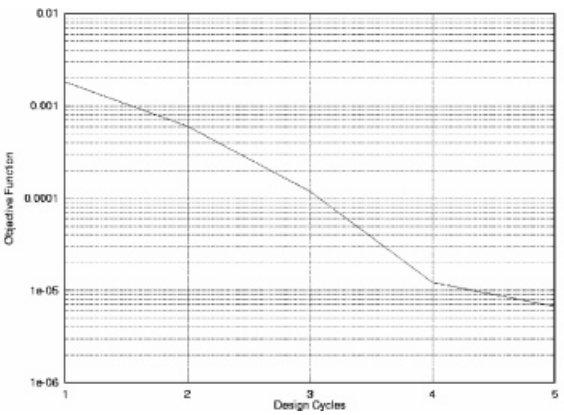

Figure 7. 3D Hydrofoil: Objective function evolution (Design cycles vs. Objective function).

7.5. 3D hydrofoil optimization

As in the 2D case, the objective of this example is to maximize the minimum pressure over the bottom and top surfaces of a 3D hydrofoil. The angle of attack of the flow was fixed again to a value of 5°, and the initial geometry was the same than the target one presented in the above example (see Section 7.4). Sixteen design variables were placed on the hydrofoil surface: Eight equal-spaced at the intersection line between the hydrofoil bottom and top surface with the vertical symmetric plane ($z = 0$), and another eight at the intersection with the vertical plane $z = 0.5$. The objective function was defined following the 2D case (see Equation (22)) as:

$$I_c = w_1 \left(\int_{\Gamma_s} p n_y \, d\Gamma - C_L^* \right) + w_2 \sum_{j \in N} |p_i - p_j|, \quad \forall i \text{ on } \Gamma_s \tag{24}$$

where Γ_s is the hydrofoil bottom and top surfaces, w_1 and w_2 are the cost function weights, p is the pressure, n_y the vertical component of the normal vector on Γ_s, C_L^* the fixed lift (computed for the initial configuration), p_i and p_j refer to the pressure at the nodal points i and j on Γ_s respectively, and N refers to the nodal points connected to i. The first term of Equation (22) enforces a fixed lift, while the second term assures an optimally smooth-pressure gradient along the hydrofoil. Given that the stagnation pressure is fixed by the external flow, this last condition indirectly assures that the minimum pressure on Γ_s increases as the design progresses.

The procedure to arrive at the optimal solution were the same as the one used in the 2D example (see Section 7.2). The geometrical restrictions used for the pseudo-shell approach were: Fixed leading and trailing edge to avoid rigid-body movements, and symmetric conditions at the intersection lines between the hydrofoil and the vertical planes $z = 0$ and 0.5.

To arrive at the optimal solution six design cycles were needed to maximize the minimum pressure (second term of Equation (22)). Clearly, during these first cycles the lift restriction was violated. Then, two design cycles were performed to fulfill the lift restriction. After these eight cycles the initial minimum pressure coefficient decreased from -0.736 to -0.514;

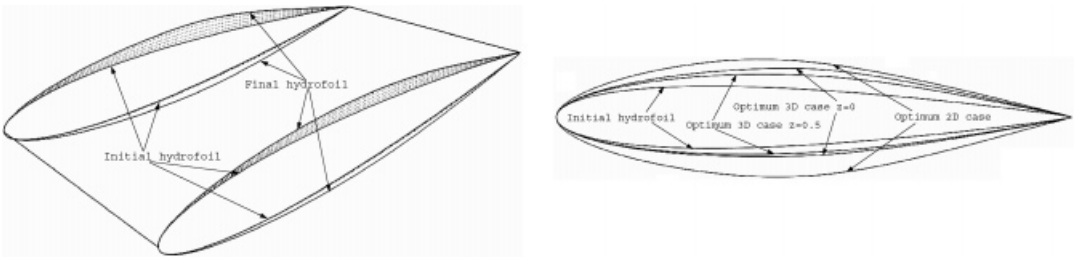

Figure 8. From left to right: Shape evolution through the design cycles. Superposition of the initial cross-section, the optimum cross-sections for the 3D case, and the optimum cross-section for the 2D one.

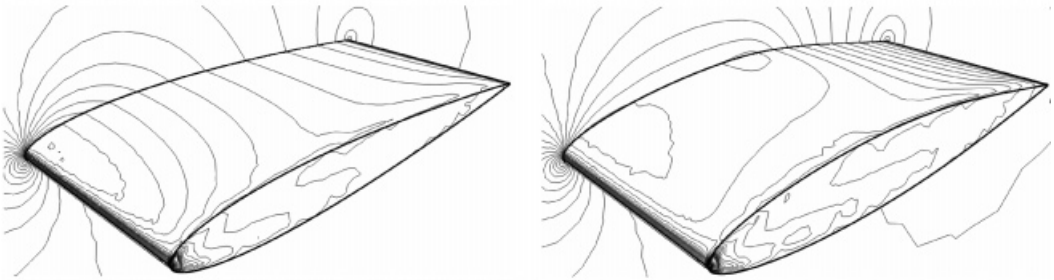

Figure 9. From left to right: Pressure distribution over the initial hydrofoil (30 contours from -0.7 to 0.5). Pressure distribution over the optimum hydrofoil (30 contours from -0.7 to 0.5).

a 30.2 per cent improvement. The value of the lift restriction (first term of Equation (22)) at the end of the design cycles was 5.0×10^{-5}.

In Figure 8 the evolution of the hydrofoil surface, and the superposition of the optimum 3D profiles at $z = 0$, and 0.5, the optimum 2D profile, and the initial profile (same for 2D and 3D cases) are shown. Note that the optimal solution has an important 3D component: All four profiles are different. In Figure 9 the initial pressure distribution and the final one are shown. Again, in this Figure the 3D effect of the pressure distribution can be observed. Finally, in Figure 10 the evolution of the minimum pressure coefficient is shown. Note that it decreases a little bit in the last two cycles, due to the fact that they were realized only to enforce the lift restriction, i.e. w_2 was set to zero in Equation (24), allowing a decrease of the maximum minimum pressure.

7.6. 3D ship drag minimization

The last numerical example consists of minimizing the drag of a ship hull by optimizing the shape of its bulb. The geometry, as well as the initial surface mesh and the locations of the five design variable are shown in Figure 11(a). The volumetric mesh contains 591,018 tetrahedral elements and 116,551 nodal points. The Froude number was set to $Fr = 0.22$, and the cost function was defined as

$$I_c = \int_{\Gamma_s} pn_x \, d\Gamma \tag{25}$$

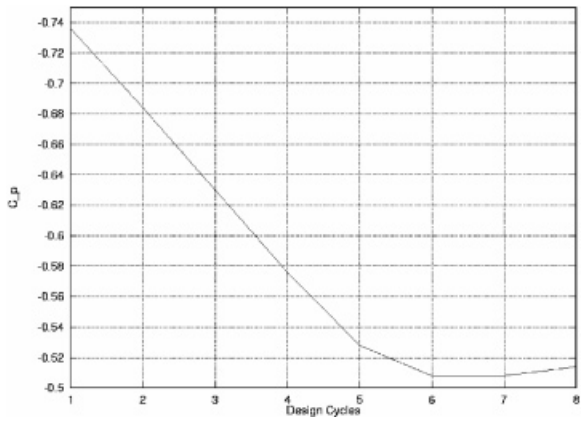

Figure 10. Cost-function evolution (design cycles vs. minimum pressure coefficient over the hydrofoil).

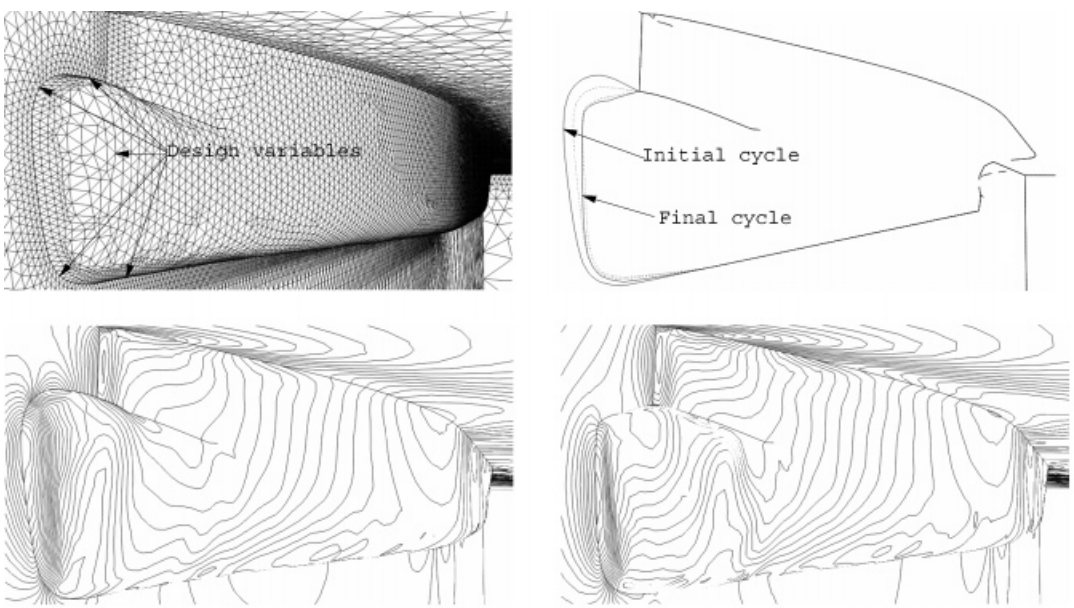

Figure 11. From left to right and top to bottom: Initial geometry and surface mesh. Shape evolution through the design cycles. Initial pressure distribution; peaks $(-0.2627, 0.5292)$ Final pressure distribution; peaks $(-0.2627, 0.4687)$

where Γ_s is the solid hull surface, n_x the component x of the normal vector on Γ_s, and p is the total pressure on Γ_s. The steady-state wave elevation pattern was computed by solving the following problem on the water surface Γ_w:

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial z} = w \tag{26}$$

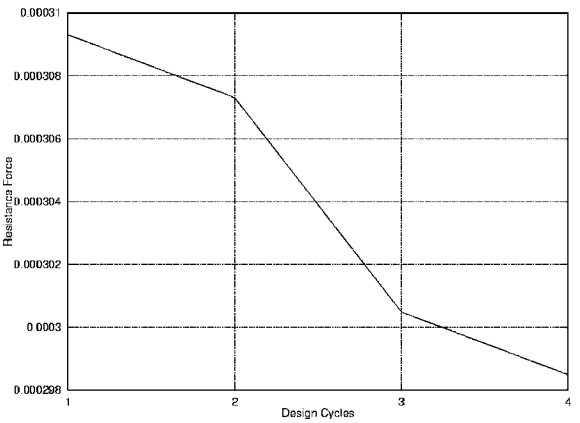

Figure 12. 3D ship drag minimization: objective function evolution (design cycles vs. resistance force).

where h is the wave elevation of each point on Γ_w, u and v the horizontal velocity components on Γ_w, and w the vertical one. Details of this approach can be consulted in References [40, 47].

Four design cycles were carried out, yielding a decrease in the drag force on the entire hull by 4 per cent. In Figure 11(b) the evolution of the bulb shape through the optimization process is shown. Observe that the bulb does not disappear, as may be thought a priori, due to the wave effects. In Figure 11(c) and (d) the pressure distribution over the initial and final shapes is presented. Finally, the evolution of the resistance force is shown in Figure 12. The CPU to compute the sensitivities and change the shape was approximately 25 times less than the one needed to solve the flow equations.

At this moment, the possibility of including the wave equation (26) as an additional restriction to the optimization process is being studied. Numerical experience has shown that this may be necessary to obtain a greater reduction of the resistance force.

8. CONCLUSIONS AND OUTLOOK

A methodology to solve design problems using the incompressible Euler equations and an incomplete-gradient Adjoint approach was presented. A continuous adjoint formulation for incompressible Euler design problems, and a scheme to compute the sensitivities which does not depend on the CAD representation, were derived. To do this, an innovative pseudo-shell surface parametrization scheme for the two and three-dimensional problems was introduced. The procedure is not only very cheap from the computational point of view, but also produces smooth, singularity-free surfaces, a highly desirable characteristic in any optimization methodology. The scheme has also the important advantage that the perturbation of a single point produces a smooth perturbation on the entire surface, allowing the finite-difference gradients to be approximated in an accurate manner.

Several examples indicate that the present scheme yields proper results without having to incur the cost of a complete adjoint solution. In general, for the three-dimensional problems, the cost of solving the CFD problem was between 15 and 30 times higher than the cost of gradient calculation and surface reconstruction.

ACKNOWLEDGEMENTS

It is a pleasure to acknowledge the partial support of the Centro Internacional de Métodos Numéricos en Ingeniería, Barcelona, Spain, for the first author. This work was also partially funded by NRL LCP&FD. Dr. William Sandberg was the technical monitor.

REFERENCES

1. Mosetti G, Poloni C. Aerodynamic shape optimization by means of a genetic algorithm. *Proceedings of the 5th International Symposium on Computational Fluid Dynamics-Sendai*, vol. II, 1993; 279–284.
2. Quagliarella D, Cioppa A. Genetic algorithms applied to the aerodynamic design of transonic airfoils. *AIAA Paper* 94-1896-*CP*, 1994.
3. Gage P, Kroo I. A role for genetic algorithms in a preliminary design environment. *AIAA Paper* 93-3933, 1993.
4. Crispin Y. Aircraft conceptual optimization using simulated evolution. *AIAA Paper* 94-0092, 1994.
5. Toropov V. Simulation approach to structural optimization. *Structural Optimization* 1989; **1**:37–46.
6. Giunta A, Dudley J, Narducci R, Grossman B, Haftka R, Mason W, Watson L. Noisy aerodynamic response and smooth approximation in HSCT design. *AIAA Paper* 94-4376, 1994.
7. Chiandussi G. Development of a shape optimization technique based on a response surface methodology. *Publication CIMNE No*. 138, 1998.
8. Narducci R, Grossman B, Valorani M, Dadone A, Haftka R. Optimization methods for non-smooth or noisy objective functions in fluid design problems. *AIAA*-95-1648-*CP*, 1995.
9. Baysal O, Eleshaky M. Aerodynamic sensitivities analysis methods for the compressible Euler equations. *Journal of Fluids Engineering* 1991; **113**:681–688.
10. Borggaard J, Burns J, Cliff E, Gunzburger M. Sensitivity calculation for a 2-D inviscid supersonic forebody problem. *Identification and Control Systems Governed by Partial Differential Equations* 1993; 14–24.
11. Burgreen G, Baysal, O. Aerodynamic shape optimization using preconditioned conjugate gradient methods. *AIAA*-93-3322, 1993.
12. Hou G, Maroju V, Taylor A, Korivi V. Transonic turbulent airfoil design optimization with automatic differentiation in incremental iterative forms. *AIAA*-95-1692, 1995.
13. Newman J, Taylor A. Three-dimensional aerodynamic shape sensitivity analysis and design optimization using Euler equations on unstructured grids. *AIAA*-96-2464, 1996.
14. Sherman L, Taylor C, Green L, Newman P, Hou G, Korivi V. First and second order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. *AIAA*-94-4262, 1994.
15. Young D, Huffman W, Melvin R, Bieterman M, Hilmes C, Johnson F. Inexactness and global convergence in design optimization. *AIAA*-94-4386, 1994.
16. Rizk M. CFD optimization by sensitivity derivatives evaluated from finite-difference jacobians. *AIAA*-95-1691-*CP*, 1995.
17. Pironneau O. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics* 1973; **59**:117–128.
18. Pironneau O. On optimum profiles in Fluid Mechanics. *Journal of Fluid Mechanics* 1974; **64**:97–110.
19. Glowinski R, Pironneau O. On the numerical computation of the minimum-drag profile in laminar flow. *Journal of Fluid Mechanics* 1975; **72**:385–389.
20. Reuter J, Jameson A, Farmer J, Martinelli L, Saunders D. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA*-96-0094, 1996.
21. Kuruvila G, Ta'asan S, Salas M. Airfoil design and optimization by the one-shot method. *AIAA*-95-0478, 1995.
22. Reuter J, Jameson A, Alonso J, Rimlinger M, Saunders D. Constrained multipoint aerodynamic shape optimization using and adjoint formulation and parallel computers. *AIAA*-97-0103, 1997.
23. Nielsen E, Anderson W. Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations. *AIAA*-98-4809, 1998.
24. Anderson W, Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *ICASE Report No*. 97-9, 1997.
25. Elliot J, Peraire J. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. *AIAA Journal* 1997; **35**:1479–1485.
26. Matsuzawa T, Hafez H. Treatment of shock waves in design optimization via adjoint equation approach. *AIAA*-98-2537, 1998.
27. Anderson W, Bonhaus D. Airfoil design on unstructured grids for turbulent flows. *NASA TM* 112867, 1997.
28. Reuter J, Alonso J, Martins J, Smith S. A couple aero-structural optimization method for complete aircraft configurations. *AIAA*-99-0187, 1999.
29. Jameson A. Optimum aerodynamic design using CFD and control theory. *AIAA*-95-1729-*CP*, 1995.
30. Nielsen E. Aerodynamic design sensitivities on an unstructured mesh using the Navier–Stokes equations and a discrete adjoint formulation. *Ph.D. Thesis*, Virginia Polytechnic Institute and State University, 1998.

31. Behr M, Tezduyar T. Finite element solution strategies for large-scale flow simulations. *Computer Methods in Applied Mechanics and Engineering* 1994; **112**:3–24.
32. Hughes T, Franca L, Balestra M. A new finite element formulation for computational fluid dynamics: V. circunvecting the babušca brezzi condition: a stable Petrov–Galerkin formulation for the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**:85–99.
33. Hughes T, Franca L. A new finite element formulation for computational fluid dynamics: VII. the Stokes problem with various well-posed boundary conditions: symmetric formulation that converge for all velocity/pressure spaces. *Computer Methods in Applied Mechanics and Engineering* 1987; **65**:85–96.
34. Hughes T, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
35. Franca L, Hughes T. Two classes of mixed finite element methods. *Computer Methods in Applied Mechanics and Engineering* 1988; **69**:89–129.
36. Franca L, Stenberg R. Error analysis of some Galerkin least-squares methods for the elasticity equations. *SIAM Journal of Numerical Analysis* 1991; **28**:1680–1697.
37. Franca L, Frey S. Stabilized finite element methods: II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–133.
38. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1999, accepted for publication.
39. Ramamurti R, Löhner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computer and Fluids*, 1996; **5**:119–132.
40. Löhner R, Yang C, Oñate E, Idelsohn S. An unstructured grid-based, parallel free surface solver. *AIAA*-97-1830, 1997.
41. van Leer B. Towards the ultimate conservative scheme. II. Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics* 1974; **14**:361–370.
42. Mohammadi B. Dynamical approaches and incomplete gradients for shape optimization and flow control. *AIAA*-99-3374, 1999.
43. Soto O, Löhner O. CFD optimization using an incomplete-gradient adjoint approach. *AIAA*-00-0666, 2000.
44. Braess D. *Finite Elements. Theory*, *Fast Solvers*, *and Applications in Solid Mechanics.* Cambridge University Press: Cambridge, 1997.
45. Taubin G. A signal processing approach to fair surface design. *IBM Research Report*, 1995.
46. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods and Engineering* 1996; **12**:599–608.
47. Löhner R, Yang C, Oñate E, Idelsohn S. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics* 1999; **31**:271–293.

