

# General Methodologies for Incompressible Flow Design Problems

O. Soto and R. Löhner  
CSI/Laboratory for Computational Fluid Dynamics  
George Mason University, MS 4C7  
Fairfax, VA 22030-4444, USA

## Abstract

Two design methodologies based on Incomplete-Gradient adjoint approaches for flow problems governed by the incompressible Navier-Stokes (NS) equations are presented. The main features of the algorithms is that they avoid solving the adjoint equations, saving an important amount of CPU time. Furthermore, the methodologies are general in the sense that they do not depend on the geometry representation, and all the gridpoints on the surface to be optimized can be chosen as design parameters. 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 and the type of mesh. Once the sensitivities and the direction of movement have been computed, 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. Furthermore, this surface parametrization allows to impose the problem geometrical restrictions in a very easy manner. The volume mesh is updated to fix the new boundary using an innovative level approach, which allows to compute the sensitivity contribution of the interior mesh points by using finite differences in a very fast manner. The methodologies can deal with multi-objective function problems, and flow restrictions such as constant lift, etc. Some 2D and 3D numerical examples are shown.

## 1 Introduction

Genetic algorithms [18, 25, 9, 7], approximate objective function surface schemes [10, 6, 19] and gradient based methods [3, 4, 5, 12, 20, 30, 35, 19, 29, 3, 4, 5, 12, 20,

30, 35, 19, 29] are usual methodologies found in the literature to solve optimization problems. Due to their high computational cost, most of them are not suitable for CFD optimization problems. For example, most of the gradient based methods require at least the solution of a large linear system of equations to compute the objective function gradient with respect to one design parameter. Thus, for  $n$  design parameters,  $n$  large linear systems have to be solved, which makes the methods of at least order  $O(n+1)$ . This cost can be reduced significantly if the LU decomposition of the flow 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 [23, 24, 11, 28, 14, 27, 22, 2, 8, 16, 1, 26, 29, 13, 19, 21]. 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, i.e. the cost is now only  $O(2)$  CFD solutions per design cycle. The present article describes two approximate adjoint formulations for the incompressible NS equations, with a cost of  $O(1)$  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  $\beta = (\beta_1, \dots, \beta_m)$ .

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

---

© Copyright by the authors. Published by the AIAA with permission.

compressible NS equations, which can be written in conservative form as:

$$\begin{aligned} \mathcal{R}_u &= \nabla \cdot (u \otimes u) + \nabla p - 2\nu \nabla \cdot \varepsilon(u) = 0, \\ R_p &= \nabla \cdot u = 0, \end{aligned} \quad (1)$$

with appropriate boundary conditions. In the above equation  $\nu$  is the kinematic viscosity and  $\varepsilon(\cdot)$  is the symmetric gradient of a vectorial field.

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 continuous NS 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 redefined as:

$$I = I_c + \int_{\Omega} \Psi \cdot \mathcal{R} \, d\Omega. \quad (2)$$

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

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

The optimization process proceeds by solving the two first 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  $\beta$  fixed (which is the same as  $\Omega$  fixed), and with the computed velocity and pressure fields, the equation (3)b is solved to obtain the set of Lagrange multipliers  $\Psi$ . 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 (3)c is achieved.

### 3 Flow Solution

The solution of (3)a can be performed with several methods (projection, artificial compressibility, GLS, OSS, etc.). In the present work, an OSS-type implicit monolithic formulation was used (see [33] for details). The stabilized momentum equation is solved using a standard GMRES algorithm, and the pressure is integrated by solving a Poisson equation with a 4th order damping for the divergence constraint with a CG-ILU solver.

## 4 Continuous Adjoint Formulation

If the objective function  $I_c$  is only defined on the solid boundary, the adjoint variables can be obtained from the following PDE problem (see [31, 32] for details):

$$\begin{aligned} (u \cdot \nabla) \psi_u + \nabla \psi_u \cdot u + \nabla \psi_p &= -2\nu \nabla \cdot \varepsilon(\psi_u) \quad \text{in } \Omega, \\ \nabla \cdot \psi_u &= 0 \quad \text{in } \Omega, \end{aligned} \quad (4)$$

and the boundary condition must satisfy:

$$\begin{aligned} \int_{\Gamma_s} 2\nu n \cdot \varepsilon(\delta u) \cdot \psi_u \, d\Gamma &= \frac{\partial I_c}{\partial u} \delta u, \\ - \int_{\Gamma_s} \delta p \psi_u \cdot n \, d\Gamma &= \frac{\partial I_c}{\partial p} \delta p \end{aligned} \quad (5)$$

where  $\delta u$  and  $\delta p$  are admissible perturbations of the velocity and pressure field, respectively,  $\Gamma_s$  the solid boundary and  $n$  its exterior normal.

An example of boundary conditions for (4) is given by considering the minimization of the drag force over certain surface (subject to some geometrical or flow restrictions to avoid the flat plate solution). In this case,  $I_c$  is defined as:

$$I_c = \int_{\Gamma_s} n \cdot (-pI + 2\nu \varepsilon(u)) \cdot e \, d\Gamma \quad (6)$$

where  $e$  is the unit vector in the drag direction. Equation (5) dictates the following boundary conditions for the adjoint variables:

$$\begin{aligned} \int_{\Gamma_s} 2\nu n \cdot \varepsilon(\delta u) \cdot \psi_u \, d\Gamma &= \int_{\Gamma_s} 2\nu n \cdot \varepsilon(\delta u) \cdot e \, d\Gamma, \\ \int_{\Gamma_s} \delta p n \cdot \psi_u \, d\Gamma &= \int_{\Gamma_s} \delta p n \cdot e \, d\Gamma. \end{aligned} \quad (7)$$

Clearly, the right boundary condition for such objective function is  $\psi_u = e$  on  $\Gamma_s$ .

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 (4) (i.e. GLS, projection schemes, artificial compressibility schemes, OSS, etc.).

## 5 Discrete Adjoint Formulation

Following equation (2), a discrete adjoint method is formulated by redefining the cost function as:

$$I = I_c + \Psi(AU - b) \quad (8)$$

where  $\mathcal{R}(U) = AU - b = \mathbf{0}$  is the system of equations obtained after discretizing the NS problem (1), and applying its respective Dirichlet boundary conditions. Then, the adjoint variables are computed by solving the following first order approximation of (3)b:

$$\frac{\partial I_c}{\partial U} + A^T \Psi = \mathbf{0}. \quad (9)$$

To compute accurate derivatives, the continuous or discrete systems, (4) or (9), have to be solved. However, as will be shown below, the adjoint solution can be avoided by using an Incomplete-Gradient approach. In [17], 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  $\beta$  (see (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.

## 6 Computation of Sensitivities

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

$$\frac{\partial I}{\partial \beta} = \frac{\partial I_c}{\partial \beta} + \frac{\partial}{\partial \beta} \left( \int_{\Omega} \Psi \cdot \mathcal{R} \, d\Omega \right). \quad (10)$$

for the continuous adjoint approach. The discrete adjoint counterpart is straightforward and will be presented later. The desired optimal solution is obtained when the gradient is equal to zero.

Equations (10) and (8) may be evaluated in a variety of ways, e.g. finite differences, automatic differentiation, exact differentiation using flow and/or geometry parametrization, etc. [16, 8, 2, 1, 28, 27, 26, 29, 13, 22, 21]. 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. The finite difference approach was chosen to compute all the terms of (10), as it offers the following additional advantages: Simplicity, independence of the problem dimension (2D or 3D), and, independence of the flow solver employed. To approximate the sensitivities by finite differences, the steps presented in Algorithm 1 (see next box) have to be implemented. 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

### Algorithm 1

- (i) Obtain  $\Psi$  from (4) with the boundary condition (5).
- (ii) Evaluate  $I$  from (2) continuous approach, or (8) discrete one.
- (iii) Perturb the coordinates of a design variable  $\beta_k$  in the external normal direction by a small  $\epsilon$ .
- (iv) Move (or smooth) the boundary  $\Gamma_s$  according to the design variable perturbation. This is, the coordinates of the grid boundary points on  $\Gamma_s$  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  $\beta_k$  is perturbed,  $\Gamma_s$  has to be moved, and a new boundary  $\Gamma'_s$  and domain  $\Omega'$  are obtained.
- (v) Move the volume mesh according to the boundary movement.
- (vi) Evaluate  $I'$  from (2) or (8) on  $\Gamma'_s$ .
- (vii) The desired sensitivity is obtained as:

$$\frac{\partial I}{\partial \beta_k} = \frac{(I' - I)}{\epsilon}$$

- (viii) Repeat (ii) to (vii) for each design point.

to the gradient obtained by finite differences. However, this procedure is very expensive if the number of design variables is high.

## 7 Incomplete Continuous Adjoint Gradient

A considerable simplification for the continuous adjoint approach is obtained by taking into account that the main contribution of the adjoint terms to the gradient are at the boundary  $\Gamma_s$ . 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.

With these arguments in mind, the sensitivity can be approximated by the Algorithm 2 presented below (see [31, 32] for a detailed explanation).

**Algorithm 2**

- (i) Obtain only  $\psi_n = \psi_u \cdot n$  on the boundary by using the condition (5).
- (ii) Evaluate  $I = I_c + \int_{\Omega} \psi_n \cdot \mathcal{R} \, d\Omega$ . Here,  $\mathcal{R}$  is evaluated from equation (1) on the first layer of elements next to  $\Gamma_s$ .
- (iii) Perturb the coordinates of a design variable  $\beta_k$  in the external normal direction by a small  $\epsilon$ .
- (iv) Move (or smooth) the boundary  $\Gamma_s$  according to the design variable perturbation (the same as step (iv) in Algorithm 1).
- (v) Evaluate  $I' = I'_c + \int_{\Omega'} \psi_n \cdot \mathcal{R} \, d\Omega$ .
- (vi) The desired sensitivity is:

$$\frac{\partial I}{\partial \beta_k} = \frac{(I' - I)}{\epsilon}$$

- (vii) Repeat (ii) to (vi) for each design point.

This scheme 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.

## 8 Incomplete Discrete Adjoint Gradient

Even though the gradient approximation obtained by Algorithm 2 is enough for certain types of problems, if the objective function is defined in the interior of the domain it can not be used. In addition, for incompressible Euler flow problems, the boundary condition given by (5) only define the normal component of  $\psi_u$  on  $\Gamma_s$ . Hence, not only the boundary adjoint pressure contribution and the interior adjoint contributions to the gradient are neglected, but also the tangential contribution of the adjoint on the boundary.

An efficient methodology to taking into account the mentioned contributions was developed by computing the adjoint variables in the interior of the domain in an approximated manner, and by developing a very fast algorithm to compute the gradient due to the interior point movement.

The discrete adjoint equation (5) is solved using the same projection type algorithm developed for the flow equations. This is, the adjoint momentum equation is computed using a GMRES solver to obtain  $\psi_u$ , and such values are replaced into the preconditioned adjoint pressure equation to obtain  $\psi_p$ . This last equation is solved with a CG-ILU solver.  $\psi_p$  is replaced into the momentum equation to re-compute  $\psi_u$ . Then  $\psi_u$  into the pressure equation to correct  $\psi_p$ , and so on until convergence is achieved. An incomplete solution is obtained by taking a large value for the convergence error of such algorithm. Numerical experience has shown that a good approximation to the complete gradient is obtained by decreasing the residuals of (5) only one order of magnitude. Such procedure saves at least a 90% of the adjoint complete calculation.

Then, the only ingredient missing to compute the incomplete gradient in a fast manner without neglecting the interior gradient contributions, and the pressure and tangential boundary adjoint contributions (using Algorithm 1 instead of Algorithm 2), is a fast boundary and volume mesh movement algorithm. For the boundary movement, the pseudo-shell approach presented in [32] was used in most of the problems. Also, a standard B-Spline representation was implemented for some two dimensional cases. On the other hand, the algorithm for the volume mesh movement is totally new, and will be presented in detail below.

As a final remark, it can be noted that the incomplete methodology described in this section may be also used with the continuous adjoint equation. However, the discrete adjoint was chosen for this incomplete formulation due to the following advantages: It is totally consistent with the method used to solve the flow equations. Therefore, the sensitivities always approximate in a good manner the finite difference ones. For the continuous formulation, this is true only for very fine meshes. In addition, no boundary conditions have to be applied for  $\Psi$  as  $A^T$  already includes the flow ones, and the value of  $\psi_k$  where  $U_k$  is prescribed is immaterial (it has no effect on the gradients). This can be easily verified by writing (3)c as:

$$\frac{\partial I}{\partial \beta} = \frac{\partial I_c}{\partial \beta} + \psi_i \frac{\partial \mathcal{R}_i}{\partial \beta} \quad (11)$$

and taking into account that for the prescribed nodes,

$\mathcal{R}_k = C$  being  $C$  the constant dirichlet value. Then,  $\partial\mathcal{R}_k/\partial\beta = 0$ . Moreover, in the continuous approach a certain level of effort is required not only to arrive at the appropriate boundary conditions, but to determine whether boundary conditions can be obtained [2]. Furthermore, equation (4) has to be discretized and coded, in contrast with the discrete adjoint approach, where the same routines used for the flow discretization can be reused (only a routine to transpose the system matrix has to be written). Finally, new cost functions are more easily added to the discrete adjoint formulation because they enter the problem only through the right-hand side of a linear system of equations, and it is usually a simple matter to obtain all the necessary derivatives by differentiating the discrete objective function directly [2].

## 9 Volume Mesh Movement

Given an old and a new solid boundary ( $\Gamma_s$  and  $\Gamma'_s$ ), a mesh “velocity” can be computed for all the grid points on the solid surface as:

$$v_i = \frac{(x'_i - x_i)}{\delta t}, \quad (12)$$

where  $x'_i$  and  $x_i$  are the new and old position vectors of the nodal point  $i$  respectively, and  $\delta t$  is a constant normally taken as one. To compute  $v$  in the rest of the mesh, each grid point  $i$  is assigned a level  $l_i$  based on the following rules:

- (i) A nodal point  $i$  belongs to the level  $l_i = 0$ , if it is on  $\Gamma_s$ .
- (ii) A nodal point  $i$  belongs to a level  $l_i > 0$ , if it is connected to at least one node of level  $l_{i-1}$ .

Then, the  $k$  mesh velocity component of the interior grid points  $i$  (points that not belong to  $\Gamma_s$ ), is computed as:

$$v_i^k = \frac{\sum_{j \in l_{i-1}} w_{ij} v_j^k}{\sum_{j \in l_{i-1}} w_{ij}}, \quad (13)$$

if  $v_i^k$  is not prescribed to a given value, or  $v_i^k = \bar{v}$  if it is prescribed to  $\bar{v}$  ( $\bar{v}$  is given by geometrical or symmetric restrictions).  $j \in l_{i-1}$  is the set of points that are connected to point  $i$  and belongs to the level  $l_{i-1}$  ( $i$  belongs to the level  $l_i$ ), and  $w_{ij}$  are weights computed as:

$$w_{ij} = \frac{1}{d_{ij}^2}, \quad (14)$$

where  $d_{ij}$  is the distance between the nodal point  $i$  and  $j$ .

Once the  $v_i^k$  is obtained for all gridpoints, they are corrected for the nodal points that are far from the boundary. This is, for the nodal points belonging to a level greater than  $n/C$ , being  $n$  the number of mesh levels (maximum  $l_i$ ), and  $C$  a constant normally taken as 2. Such correction has to be done to match the zero mesh velocity condition, commonly imposed at the far field boundary. The idea of the correction is to redefine  $v_i^k$  as  $v_i^k := \alpha v_i^k$ , with a linear variation of  $\alpha$  between 1 and 0 based on the point level (1 for the level  $l_i = n/C$ , and 0 for the level  $l = n$ ).

The technique described above decreases the element distortion close to the solid boundary considerably, reducing the need for local or global remeshing, and in most cases avoiding it altogether. As a final remark, it was noted that the level methodology produces very similar results to the technique presented in [15], which is based on the solution of a Laplacian of the mesh velocities with variable diffusivity depending on the distance from the moving boundary. The main difference is in the computer time: The level technique is more than 100 times faster than the Laplacian one, which allows the use of Algorithm 1 for the gradient computation.

## 10 Optimization Algorithm

Once the sensitivities have been obtained, the solid boundary is updated using a standard steepest-descent method for the incomplete discrete approach (Algorithm 1), or an accumulated displacement procedure for the incomplete continuous approach (Algorithm 2). Such accumulated displacement method had to be used in the last case, because the incomplete continuous gradient gives the right direction of the design parameter gradients, but not the correct ratios among them. Therefore, a standard steepest-descent method may produce surface with waves, which are undesirable in most of the real design problems.

The accumulated displacement procedure mentioned above is as follows: Due to the pseudo-shell approach surface parametrization [32], the normal displacement of one point  $i$  on  $\Gamma_s$  produces a continuous displacement and rotation field in all the nodal points on  $\Gamma_s$ . Hence, if a normal displacement  $w_i$  of a design variable  $\beta_i$ ,

$$w_i = \frac{\partial I}{\partial \beta_i} n, \quad (15)$$

is imposed, the rest of the design variables will have a normal displacement  $w_j^i$  obtained by the solution of the pseudo-shell equations. Then, an accumulated normal

displacement for a design variable  $\beta_j$  can be defined as:

$$W_j = \sum_{i=1}^{n_d} w_j^i \quad (16)$$

where  $n_d$  is the number of design variables of the problem. Note that  $w_j^i$  is obtained with a very small extra cost from the step (iv) of algorithm 1 and 2, taking into account that the pseudo-shell problem is linear (in step (iv) the displacement of  $j$  due to a displacement  $\epsilon n$  of  $i$  is already computed).

The design variable  $\beta_j$  with greatest  $W_j$  is chosen as the surface movement direction. Its final position is computed by a steepest descent method as:

$$x'_j = x_j - \lambda \frac{\partial I}{\partial \beta_j} \quad (17)$$

where  $x'_j$  will be the new coordinate vector of the design parameter  $\beta_j$ ,  $x_j$  the old one, and  $\lambda$  a positive real constant. Then, the new surface  $\Gamma'_s$  is obtained by solving the pseudo-shell equations with  $x'_j$  fixed, and with the problem geometric restrictions.

Finally, the surface (or line in 2D) is smoothed following the procedure presented in [34], and all the volume mesh points are updated to fix the new boundary.

It is important to remark that, the accumulated displacement definition (15)-(16), only used for the continuous approach, is totally analog to an accumulated gradient expression. In this sense, the presented scheme chooses the biggest gradient direction to improve the design.

## 11 Numerical Examples

### 11.1 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). The solid boundary was always rebuilt after each design cycle using the pseudo-shell (beam) approach.

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  $\psi_u$  (vectors) and  $\psi_p$  (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:  $-0.7281, 0.3537$ ).

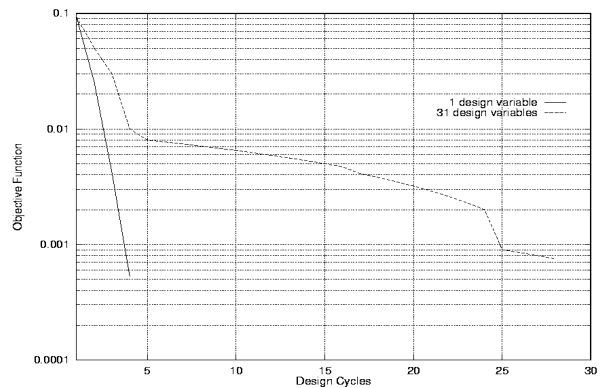


Figure 2: Cost function evolution (Design cycles vs. Cost function).

The objective function was defined as:

$$I_c = \int_{\Gamma_s} (p_d - p)^2 d\Gamma \quad (18)$$

where  $p_d$  denotes the target pressure and  $\Gamma_s$  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 pre-

sented. 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 [8]. In addition, to perform the optimization process using all the nodal points on  $\Gamma_s$  as design parameters, the solid surface was smoothed after each optimization cycle, following the procedure presented in [34]. This smoothing may deteriorate the convergence (the design parameters were moved not only following the gradient computation, but also the smoothing procedure).

In Table 1, a comparison between the Incomplete Gradient (i.e. adjoint problem is not solved, Algorithm 2) and the Complete one (i.e. adjoint problem (4) is solved and the interior mesh sensitivities are taken into account, Algorithm 1), 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 must converge to the same result using any of the two algorithms. Note also that incomplete gradients approximate complete ones as convergence achieved.

Cycle	Incomplete Gradient	Complete Gradient
1	+0.07197557	+0.05905559
2	+0.20084590	+0.11986370
3	-0.26925220	-0.21840380
4	-0.03114764	-0.03867715

Table 1: Comparison between the Incomplete and Complete Gradient algorithms for the one design variable case.

## 11.2 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 five degrees. The cost function for this example was defined as follows:

$$I_c = w_1 \left( \int_{\Gamma_s} p n_y \, d\Gamma - C_L^* \right) + w_2 \int_{\Gamma_s} \left| \frac{\partial p}{\partial t} \right| \, d\Gamma \quad , \quad (19)$$

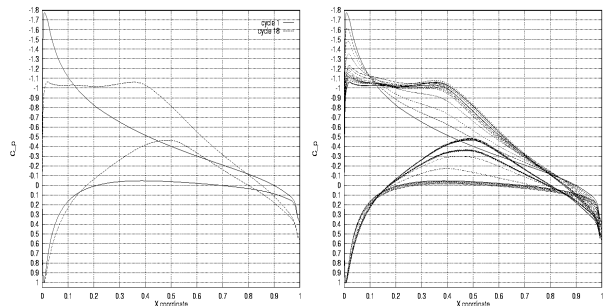


Figure 3: From left to right: Pressure coefficient distribution for the initial and final hydrofoils. Evolution of the pressure coefficient along the optimization process

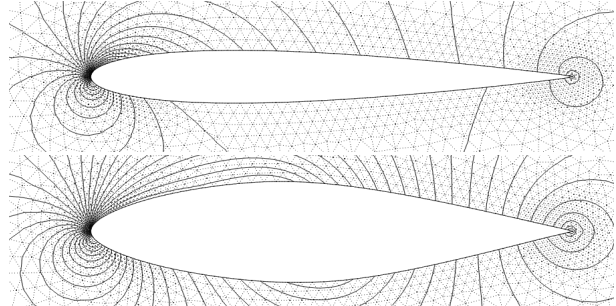


Figure 4: From top to bottom: Mesh and pressure distribution for the initial hydrofoil (peaks: -0.8884, 0.5008). Mesh and pressure distribution for the optimum (final) hydrofoil (peaks: -0.5321, 0.5020)

where  $\Gamma_s$  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  $\Gamma_s$ ,  $C_L^*$  the fixed lift (computed for the initial configuration), and  $t$  the tangential vector along the hydrofoil boundary. The first term of (19) 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  $\Gamma_s$  increases as the design progresses.

The optimization was done using the continuous incomplete gradient approach (Algorithm 2) 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% of improvement. In addition, the value of the lift restriction (first term of equation (19)) at the end of the design cycles was  $2.7 \times 10^{-6}$ . In Figure 4 the evolution of the hydrofoil profiles along the design cycles and the pressure distribution for the optimum can be observed.

### 11.3 3D Hydrofoil Optimization

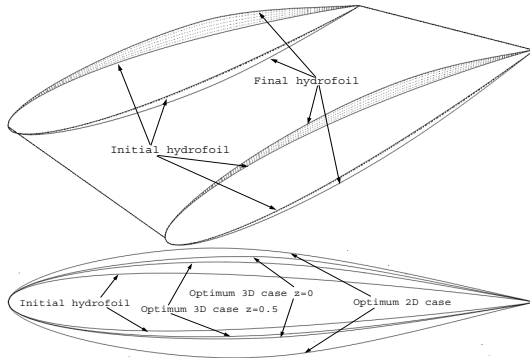


Figure 5: From top to bottom: Shape evolution through the design cycles. Superposition of the initial cross section, the optimum cross sections for the 3D case, and the optimum cross section for the 2D one.

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 five degrees, 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 (19)) as:

$$I_c = w_1 \left( \int_{\Gamma_s} p n_y d\Gamma - C_L^* \right) + w_2 \sum_{j \in N} |p_i - p_j| \quad (20)$$

$\forall i$  on  $\Gamma_s$ , where  $\Gamma_s$  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  $\Gamma_s$ ,  $C_L^*$  the fixed lift (computed for the initial configuration),  $p_i$  and  $p_j$  refer to the pressure at the nodal points

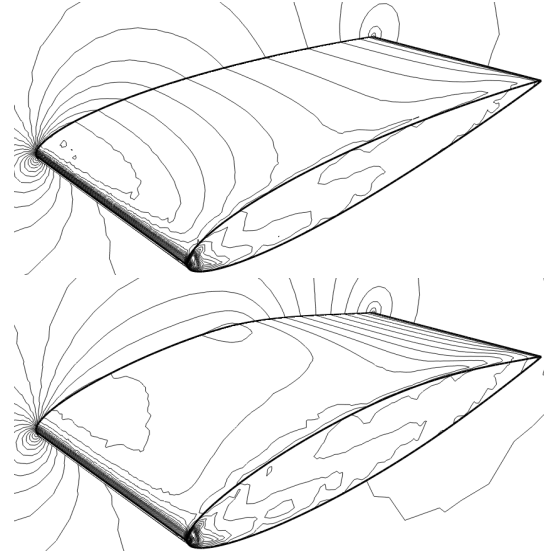


Figure 6: From top to bottom: Pressure distribution over the initial hydrofoil (30 contours from -0.7 to 0.5). Pressure distribution over the optimum hydrofoil (30 contours from -0.7 to 0.5).

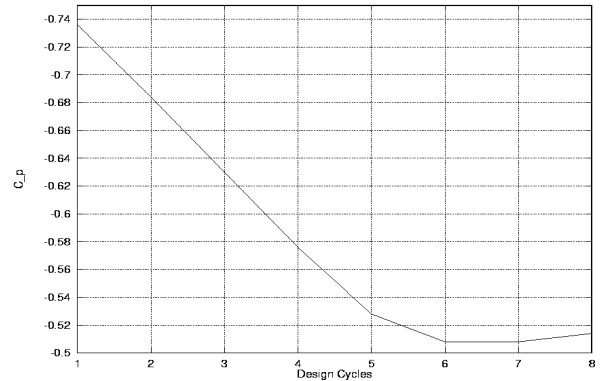


Figure 7: Cost function evolution (Design cycles vs. Minimum pressure coefficient over the hydrofoil).

$i$  and  $j$  on  $\Gamma_s$  respectively, and  $N$  refers to the nodal points connected to  $i$ . The first term of (19) 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  $\Gamma_s$  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 11.2). 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  $z = 0.5$ .

To arrive at the optimal solution six design cycles were needed to maximize the minimum pressure (second term of (19)). 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  $-0.514$ ; a 30.2% improvement. The value of the lift restriction (first term of equation (19)) at the end of the design cycles was  $5.0 \times 10^{-5}$ .

In Figure 5 the evolution of the hydrofoil surface, and the superposition of the optimum 3D profiles at  $z = 0$ , at  $z = 0.5$ , 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 6 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 7 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 (20), allowing a decrease of the maximum minimum pressure.

## 11.4 2D Drag minimization

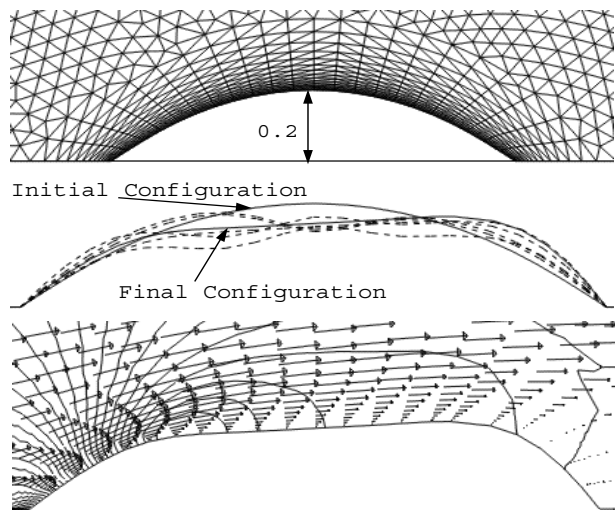


Figure 8: From top to bottom: Bump geometry and detail of the mesh. Surface evolution through the design cycles. Pressure and velocity field at the final configuration, pressure peaks:  $-0.2791, 0.5161$

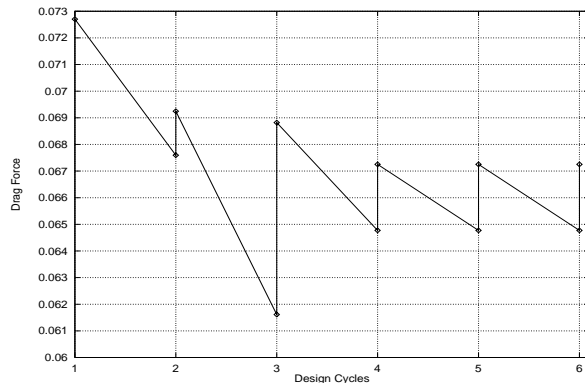


Figure 9: Drag evolution (Design cycles vs. Drag force)

The final example consists of minimizing the total drag force (viscous plus pressure) over a two dimensional bump, maintaining the area enclosed by the solid surface and the length constant (see Figure 8). The bump was parametrized using a B-spline curve and 7 design points. Symmetric flow conditions were imposed at  $y = 0$ , the velocity was prescribed to zero (NS problem) on the bump surface and to  $(1, 0)$  at inflow, the pressure was set to zero at outflow and the design points at the leading and trailing edges remained fixed along the optimization process. The drag objective function was already presented in (6), and the constant volume restriction is enforced by minimizing:

$$I_v = \sum_{b \in \Gamma_s} (V^* - y^b l^b |n_y^b|)^2 \quad (21)$$

where  $b$  refers to the boundary elements on  $\Gamma_s$ ,  $y^b$  the vertical coordinate of the middle point of  $b$ ,  $l^b$  the boundary length, and  $n_y^b$  the vertical component of the exterior normal vector at  $b$ .  $V^*$  is the initial volume, which wants to be conserved.

This example was performed using the incomplete discrete gradient approach described above (Algorithm 1 solving the discrete adjoint equations in an incomplete manner). The optimization procedure was as follows: First the sensitivities were computed at the initial geometry, and the drag was minimized using a steepest-descent method. Second, the volume was adjusted using the constant volume function (21). This function does not depend on the flow variables, therefore its minimization is very fast. These two steps were repeated until no more improvement was found in the drag force.

The example was carried out for Reynolds numbers of 1, 10 and 100, obtaining similar results for all the cases. For brevity, here only the results for Reynolds number of 100 are presented. In Figure 8 the surface evolution

through the design cycles and a detail of the pressure and the velocity field around the final bump can be observed. The drag evolution is presented in Figure 9. Note that the drag increases when the volume restriction is enforced. However, a final drag improvement of 8.2% was obtained at the end of the design process.

## 12 Conclusions

Two methodologies to solve design problems using the incompressible NS equations and incomplete gradient approaches were presented. The first was based on the boundary conditions dictated for a continuous adjoint formulation, which avoids the adjoint solution altogether. The second one was based on a partial discrete adjoint solution, which allows to take into account in an incomplete manner all the adjoint contributions to the sensitivities. In that sense, and based on the numerical experience, the last approach seems to approximate better the objective function gradients than the previous one. However, more numerical experience needs to be acquired with the incomplete discrete formulation.

In addition, the solid boundary may be parametrized using a pseudo-shell approach, which does not depend on the CAD representation, and which is very cheap from a computational point of view. Also, an innovative and very fast volume mesh movement algorithm was developed, which allows to include the interior point contributions to the gradient.

Several examples indicate that both schemes yield proper results without having to incur the cost of a complete adjoint solution. In general, the cost of the gradient computation was a 10% or less than the cost of solving the CFD problem.

## 13 Acknowledgements

This work was partially funded by NRL LCP&FD. Dr. William Sandberg was the technical monitor.

## References

- [1] W. Anderson and D. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *NASA TM 112867*, 1997.
- [2] W. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *ICASE report No. 97-9*, 1997.
- [3] O. Baysal and M. Eleshaky. Aerodynamic sensitivities analysis methods for the compressible Euler equations. *J. of Fluids Engineering*, 113:681–688, 1991.
- [4] J. Borggaard, J. Burns, E. Cliff, and M. Gunzburger. Sensitivity calculation for a 2-d inviscid supersonic forebody problem. *Identification and Control Systems Governed by Partial Differential Equations*, pages 14–24, 1993.
- [5] G. Burgreen and O. Baysal. Aerodynamic shape optimization using preconditioned conjugate gradient methods. *AIAA-93-3322*, 1993.
- [6] G. Chiandussi. *Development of a shape optimization technique based on a response surface methodology*. Publication CIMNE No. 138, 1998.
- [7] Y. Crispin. Aircraft conceptual optimization using simulated evolution. *AIAA paper 94-0092*, 1994.
- [8] J. Elliot and J. Peraire. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. *AIAA journal*, 35:1479–1485, 1997.
- [9] P. Gage and I. Kroo. A role for genetic algorithms in a preliminary design environment. *AIAA paper 93-3933*, 1993.
- [10] A. Giunta, J. Dudley, R. Narducci, B. Grossman, R. Haftka, W. Mason, and L. Watson. Noisy aerodynamic response and smooth approximation in HSCT design. *AIAA paper 94-4376*, 1994.
- [11] R. Glowinski and O. Pironneau. On the numerical computation of the minimum-drag profile in laminar flow. *J. Fluid Mech.*, 72:385–389, 1975.
- [12] G. Hou, V. Maroju, A. Taylor, and V. Korivi. Transonic turbulent airfoil design optimization with automatic differentiation in incremental iterative forms. *AIAA-95-1692*, 1995.
- [13] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA-95-1729-CP*, 1995.
- [14] G. Kuruvila, S. Ta’asan, and M. Salas. Airfoil design and optimization by the one-shot method. *AIAA-95-0478*, 1995.
- [15] R. Löhner and C. Yang. Improved ALE mesh velocities for moving bodies. *Comm. in Numer. Meth. Engrg*, 12:599–608, 1996.

- [16] T. Matsuzawa and H. Hafez. Treatment of shock waves in design optimization via adjoint equation approach. *AIAA-98-2537*, 1998.
- [17] B. Mohammadi. Dynamical approaches and incomplete gradients for shape optimization and flow control. *AIAA-99-3374*, 1999.
- [18] G. Mosetti and C. Poloni. Aerodynamic shape optimization by means of a genetic algorithm. In *Proc. of the 5th Int. Symp. on Computational Fluid Dynamics- Sendai*, volume II, pages 279–284, 1993.
- [19] R. Narducci, B. Grossman, M. Valorani, A. Dadone, and R. Haftka. Optimization methods for non-smooth or noisy objective functions in fluid design problems. *AIAA-95-1648-CP*, 1995.
- [20] J. Newman and A. Taylor. Three-dimensional aerodynamic shape sensitivity analysis and design optimization using Euler equations on unstructured grids. *AIAA-96-2464*, 1996.
- [21] E. Nielsen. *Aerodynamic design sensitivities on an unstructured mesh using the Navier-Stokes equations and a discrete adjoint formulation*. PhD thesis, Virginia Polytechnic Institute and State University, 1998.
- [22] E. Nielsen and W. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA-98-4809*, 1998.
- [23] O. Pironneau. On optimum profiles in Stokes flow. *J. Fluid Mech.*, 59:117–128, 1973.
- [24] O. Pironneau. On optimum profiles in Fluid Mechanics. *J. Fluid Mech.*, 64:97–110, 1974.
- [25] D. Quagliarella and A. Cioppa. Genetic algorithms applied to the aerodynamic design of transonic airfoils. *AIAA paper 94-1896-CP*, 1994.
- [26] J. Reuter, J. Alonso, J. Martins, and S. Smith. A couple aero-structural optimization method for complete aircraft configurations. *AIAA-99-0187*, 1999.
- [27] J. Reuter, A. Jameson, J. Alonso, M. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *AIAA-97-0103*, 1997.
- [28] J. Reuter, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA-96-0094*, 1996.
- [29] M. Rizk. CFD optimization by sensitivity derivatives evaluated from finite-difference jacobians. *AIAA-95-1691-CP*, 1995.
- [30] L. Sherman, C. Taylor, L. Green, P. Newman, G. Hou, and V. Korivi. First and second order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. *AIAA-94-4262*, 1994.
- [31] O. Soto and R. Löhner. CFD optimization using an incomplete-gradient adjoint approach. *AIAA-00-0666*, 2000.
- [32] O. Soto and R. Löhner. CFD shape optimization using an incomplete-gradient adjoint formulation. *Int. J. Numer. Meth. Engrg.*, To appear, 2000.
- [33] O. Soto and R. Löhner. An implicit monolithic time accurate finite element scheme for incompressible flow problems. *submitted to 15th AIAA CFD Conference, Anaheim, California*, 2001.
- [34] G. Taubin. A signal processing approach to fair surface design. *IBM Research Report*, 1995.
- [35] D. Young, W. Huffman, R. Melvin, M. Bieterman, C. Hilmes, and F. Johnson. Inexactness and global convergence in design optimization. *AIAA-94-4386*, 1994.