An adjoint-based design methodology for CFD problems

Orlando Soto, Rainald Löhner and Chi Yang
SCS/Laboratory for Computational Fluid Dynamics, George Mason University, Fairfax, Virginia, USA

Received February 2003
Revised August 2003
Accepted September 2003

Keywords *Finite element analysis, Optimization techniques, Differential equations, Compressible flow*

Abstract *A complete CFD design methodology is presented. The main components of this methodology are a general edge-based compressible/incompressible flow solver; a continuous adjoint formulation for the gradient computations; a steepest descent technique for the change of design variables; evaluation of the gradient of the discretized flow equations with respect to mesh by finite differences; a CAD-free pseudo-shell surface parametrization, allowing every point on the surface to be optimized to be used as a design parameter; and a level type scheme for the movement of the interior points. Several examples are included to demonstrate the methodology developed.*

1. Introduction

The relentless advance in numerical methods and computer power has made accurate flow simulations involving realistic geometries a reality. Such simulations are increasingly reducing the amount of lengthy (and costly) experiments in the aerospace, car, train and shipbuilding industries, substituting them for high fidelity CFD runs. This way of utilizing CFD is nothing more than an exchange of real for virtual experiment. However, CFD and its underlying mathematics offers the possibility to step beyond the capabilities of any experiment. While the experiment (or stand alone CFD run) only measures the performance of the product "as is", numerical methods can also predict the effect of changes in the shape of the product. This has led, over the last decade, to a large body of literature on optimal shape design (Anderson and Venkatakrishnan, 1997; Drela, 1998; Dreyer and Matinelli, 2001; Elliott and Peraire, 1997, 1998; Gumbert *et al.*, 2001; Jameson, 1988, 1995; Korte *et al.*, 1997; Kuruvila *et al.*, 1995; Li *et al.*, 2001; Medic *et al.*, 1998; Mohammadi, 1997; Mohammadi and Pironneau, 2001; Nielsen and Anderson, 1998; Reuther *et al.*, 1999; Soto and Löhner, 2001a, b, 2002; Soto *et al.*, 2002). In order to compare the merit of different designs, a function I is defined. This cost function depends on design parameters β, and the changes in flow variables $\mathbf{u}(\beta)$ due to them. The aim is then to minimize (or maximize) this cost function:

$$I(\beta, \mathbf{u}(\beta)) \rightarrow \min, \tag{1}$$

subject to the following constraints:

This research was partially supported by AFSOR, with Dr Leonidas Sakell as the technical monitor.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 14 No. 6, 2004
pp. 734-759

0961-5539
DOI 10.1108/09615530410544292

- PDE constraints:

$$\mathbf{R}(\mathbf{u}) = 0; \tag{2}$$

- Geometric constraints:

$$\mathbf{g}(\beta) \geq 0; \tag{3}$$

- Physical constraints:

$$\mathbf{h}(\mathbf{u}) \geq 0. \tag{4}$$

Examples for the cost function I are drag or prescribed pressure, for PDE constraints $\mathbf{R}(\mathbf{u})$ the Euler/Navier-Stokes equations, for geometric constraints $\mathbf{g}(\beta)$ the volume or wing area cross section, and for physical constraints $\mathbf{h}(\mathbf{u})$ a minimal pressure to prevent cavitation.

The simplest (and most expensive) way to proceed is by copying what nature has done in the course of evolution: try variations of β, recalculate the flowfield and keep the ones that minimize (i.e. improve) the cost function $I(\beta, \mathbf{u}(\beta))$. This class of optimization techniques are known as genetic algorithms (Crispin, 1994; Gage and Kroo, 1993; Quagliarella and Cioppa, 1994). While simple to code and use, robust and suited for "rough" cost functions, the number of the CFD runs required for N design variables is at least of $O(N^2)$. The speed of convergence is also strongly dependent on the crossover, mutation and selection criteria.

The second class of optimization techniques is based on evaluating gradients of $I(\beta, \mathbf{u}(\beta))$. From a Taylor series expansion we have

$$I + \Delta I \approx I + I^{\mathrm{T}}_{,\beta} \Delta\beta. \tag{5}$$

This implies that if one chooses:

$$\Delta\beta = -\lambda I_{,\beta}, \tag{6}$$

for sufficiently small λ the new functional has to diminish:

$$I + \Delta I = I - \lambda I^{\mathrm{T}}_{,\beta} I_{,\beta} \leq I. \tag{7}$$

There exist a variety of ways of computing the required gradients $I_{,\beta}$. The easiest way is via *finite differences*. For each β_i, its value vary by a small amount, recompute the cost function I, and measure the gradient with respect to β_i. For central differences, this implies $O(2N)$ field solutions for each gradient evaluation. An alternative is to use first-order differences with complex variables, which result in a difference scheme of second-order for the desired real CFD variables (Newman *et al.*, 1999). This requires $O(N)$ field solutions for each gradient evaluation, but at the cost of a flow solver with complex variables. For "noisy" or "rough" cost functions, gradients may be computed from the so-called *response surfaces*. The parameter space in a region close to the present design is populated, and a low-order polynomial is fitted through these data points. The gradients are then obtained from the low-order polynomial. This type of technique also requires $O(N)$ field solutions for each gradient evaluation. The only alternative to obtain gradients in a more expeditious manner is via adjoint solvers.

2. Gradients via adjoint variables

The main steps required for the evaluation of gradients based on adjoint variables are well known (Elliott and Peraire, 1998; Jameson, 1988, 1995; Mohammadi and Pironneau, 2001; Soto and Löhner, 2001a, b, 2002; Soto *et al.*, 2002), and therefore only a summary is given here. A variation in the objective function I and the PDE constraint $\mathbf{R}$ result in:

$$\delta I = I_{,\beta}\delta\beta + I_{,\mathrm{u}}\delta\mathbf{u}, \tag{8}$$

$$\delta\mathbf{R} = \mathbf{R}_{,\beta}\delta\beta + \mathbf{R}_{,\mathrm{u}}\delta\mathbf{u} = 0. \tag{9}$$

We can now introduce a Lagrange multiplier Ψ to merge these two expressions:

$$\delta I = I_{,\beta}\delta\beta + I_{,\mathrm{u}}\delta\mathbf{u} - \Psi^{\mathrm{T}}[\mathbf{R}_{,\beta}\delta\beta + \mathbf{R}_{,\mathrm{u}}\delta\mathbf{u}]. \tag{10}$$

After rearrangement of terms this results in:

$$\delta I = [I_{,\beta} - \Psi^{\mathrm{T}}\mathbf{R}_{,\beta}]\delta\beta + [I_{,\mathrm{u}} - \Psi^{\mathrm{T}}\mathbf{R}_{,\mathrm{u}}]\delta\mathbf{u}. \tag{11}$$

This implies that if we can solve:

$$\Psi^{\mathrm{T}}\mathbf{R}_{,\mathrm{u}} = I_{,\mathrm{u}}, \tag{12}$$

the variation of I is given by:

$$\delta I = [I_{,\beta} - \Psi^{\mathrm{T}}\mathbf{R}_{,\beta}]\delta\beta = [G^I]^{\mathrm{T}}\delta\beta. \tag{13}$$

The consequences of this rearrangement are profound:

- the variation of I exhibits only derivatives with respect to β, i.e. no explicit derivatives with respect to $\mathbf{u}$ appear; and
- the cost for the evaluation of gradients is independent of the number of design variables (!).

The original equation for the adjoint is given by:

$$\Psi^{\mathrm{T}}\mathbf{R}_{,\mathrm{u}} = I^{\Omega}_{,\mathrm{u}} + I^{\Gamma}_{,\mathrm{u}} \tag{14}$$

where I^{Ω} denotes a cost function defined over the domain, and I^{Γ} a cost function defined over the boundary. We remark that for most shape optimization problems $I^{\Omega} = 0$. In the sequel, the Einstein summation convention for double indices is adopted unless explicitly noted.

2.1 Residuals with first derivatives

If $\mathbf{R}$ can be written as a system of conservation laws of the form:

$$\mathbf{R} = \mathbf{F}^i_{,i} = 0, \tag{15}$$

where $\mathbf{F}^i(\mathbf{u})$ denotes a flux function with the property that $\mathbf{F}^i(\mathbf{u}) = \mathbf{A}^i\mathbf{u}$ where $\mathbf{A}^i = \mathbf{F}^i_{,\mathrm{u}}$ is the flux Jacobian, the following is obtained:

$$\mathbf{R}_{,\mathrm{u}}\delta\mathbf{u} = \left(\mathbf{F}^i_{,i}\right)_{,\mathrm{u}}\delta\mathbf{u} = \delta\mathbf{F}^i_{,i} = (\delta\mathbf{A}^i\mathbf{u})_{,i} + (\mathbf{A}^i\delta\mathbf{u})_{,i} + \mathrm{O}(\delta\mathbf{u}^2). \tag{16}$$

Multiplying equation (16) by the adjoint variables, integrating by parts, and neglecting the terms containing $\delta\mathbf{A}^i$ to obtain a first-order approximation yields:

$$\int_\Omega \Psi^{\mathrm{T}}\mathbf{R}_{,\mathrm{u}}\delta\mathbf{u}\,\mathrm{d}\Omega = -\int_\Omega \Psi^{\mathrm{T}}_{,i}\mathbf{A}^i\delta\mathbf{u}\,\mathrm{d}\Omega + \int_\Gamma \Psi^{\mathrm{T}}\mathbf{A}^i\delta\mathbf{u}\;\; n_i\,\mathrm{d}\Gamma, \tag{17}$$

where n_i is the ith component of the unit external normal vector on Γ. By combining equations (14) and (17), the following PDE problem is obtained:

$$-[\mathbf{A}^i]^{\mathrm{T}}\Psi_{,i} = I^\Omega_{\mathrm{u}}, \tag{18}$$

$$\int_\Gamma [\mathbf{A}^i n_i]^{\mathrm{T}}\Psi\,\mathrm{d}\Gamma = I^\Gamma_{,\mathrm{u}}. \tag{19}$$

2.2 Residuals with second derivatives

Suppose $\mathbf{R}$ can be written as a "Laplacian" of the form:

$$\mathbf{R} = -\nabla\mu\nabla\mathbf{w} = 0, \tag{20}$$

where μ denotes a diffusion or viscosity, and $\mathbf{w}(\mathbf{u})$ a set of different (e.g. non-conserved) variables. We can again obtain,

$$\Psi^{\mathrm{T}}\mathbf{R}_{,\mathrm{u}} = -\Psi^{\mathrm{T}}(\nabla\mu\nabla\mathbf{w})_{,\mathrm{u}} = -\Psi^{\mathrm{T}}\nabla\mu\nabla\mathbf{w}_{,\mathrm{u}} = -\Psi^{\mathrm{T}}\nabla\mu\nabla\mathbf{B} = I_{,\mathrm{u}}, \tag{21}$$

where $\mathbf{B} = \mathbf{w}_{,\mathrm{u}}$ denotes the Jacobian of $\mathbf{w}$. Repeated integration by parts allows us to obtain the PDE for the adjoint:

$$-\mathbf{B}^{\mathrm{T}}\nabla\mu\nabla\Psi = I^\Omega_{,\mathrm{u}}, -\int_\Gamma \mathbf{B}^{\mathrm{T}}_{,n}\mu\Psi\,\mathrm{d}\Gamma + \int_\Gamma \mathbf{B}^{\mathrm{T}}\mu\Psi_{,n}\,\mathrm{d}\Gamma = I^\Gamma_{,\mathrm{u}}. \tag{22}$$

Note that:

(1) A *linear* system of *advection* or *diffusion equations* is obtained.

(2) The eigenvalues of this system are the same as those of the original PDE.

(3) Owing to the negative sign in front of the Jacobians $\mathbf{A}^i$, the "advection direction" for the adjoint is opposite to the advection direction of the original PDE ($\mathbf{R}$).

(4) The sign of the operator has not changed, i.e. diffusion of the adjoint variables occurs in the same way as with the original PDE variables; this was to be expected as the diffusion (Laplacian) operator is self-adjoint.

(5) The derivatives of the diffusion/viscosity with respect to the unknowns have been neglected; this assumption may not be applicable to all cases (particularly complex turbulence models and/or hypersonic flows), but has been found to yield acceptable solutions over a wide range of flow regimes.

(6) In most applications, the residual (or PDE describing the field solution) will contain a combination of first and second spatial derivatives. As these are additive, the same sum applies to the adjoints.

3. Steps required per design cycle

The complete design procedure is shown in Figure 1. In principle, after a new surface has been obtained, a new mesh can be generated, and the solution restarted. Given that typically the solution does not change appreciably between design cycles, it is far more expedient to simply move the mesh based on the surface changes. This is the approach taken here.

Thus, for every design cycle we require the following steps:

(1) a flow solver;

(2) an adjoint solver;

(3) the evaluation of gradients; and

(4) a mesh movement module;

In the following sub-sections, we describe each one of these steps in more detail.

3.1 Flow solver

The flow code used is FEFLO, an unstructured, edge-based, tetrahedral finite element solver (Löhner, 2002). For the compressible Euler equations (see Appendix 1) consistent numerical fluxes are obtained using the approximate Riemann solver of vanLeer, Roe, AUSM+ or HLLE (Löhner, 2001). Fast convergence to steady-state is achieved via an iterative algorithm based on the generalized minimal residuals (GMRES) with lower-upper symmetric Gauss-Seidel (LU-SGS) preconditioning (Luo *et al.*, 1998, 2001; Sharov *et al.*, 2000).

For the incompressible Euler/Navier-Stokes equations (see Appendix 2) consistent numerical fluxes for the momentum equations are obtained using the approximate

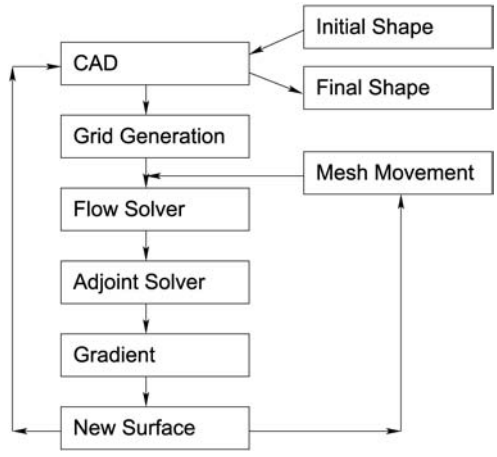

Figure 1.
Design cycle

Riemann solver of Roe. For the divergence equation a fourth-order pressure damping operator is employed. The equations are advanced in time using a projection scheme (Löhner, 2001; Löhner *et al.*, 1999).

3.2 Adjoint solver

The adjoint equations to be solved are of the form:

$$\Psi_{,t} - [\mathbf{A}^i]^{\mathrm{T}}\Psi_{,i} = \mathbf{B}^{\mathrm{T}}\nabla\mu\nabla\Psi + I^{\Omega}, \tag{23}$$

where

$$\mathbf{A}^i = \mathbf{F}^i_{,\mathrm{u}}, \quad \mathbf{B} = \mathbf{w}_{,\mathrm{u}}. \tag{24}$$

The same kind of edge-based, tetrahedral finite element solver as employed for the flow solver is also used for the adjoint equations. At the beginning of the run, the steady flow results from the flow solver are read in and the Jacobian matrices are computed and stored. Consistent numerical fluxes are obtained using a blended second- and fourth-order edge-based dissipation. The solution is advanced in time using explicit Runge-Kutta timestepping schemes with residual smoothing.

3.2.1 Adjoint solver. Boundary conditions for the adjoint

Inviscid forces. The inviscid forces are given by

$$\mathbf{f} = \int_{\Gamma} p\mathbf{n}\,\mathrm{d}\Gamma. \tag{25}$$

For a cost function of the form:

$$I = \mathbf{f}\cdot\mathbf{c}_{\mathrm{w}}, \tag{26}$$

where $\mathbf{c}_{\mathrm{w}} = c^x_{\mathrm{w}}, c^y_{\mathrm{w}}, c^z_{\mathrm{w}}$ represent weights, we obtain, from equation (19):

$$\mathbf{n}\cdot\Psi_{\mathrm{v}} = \mathbf{c}_{\mathrm{w}}\cdot\mathbf{n}, \tag{27}$$

i.e. the normal adjoint velocity is prescribed while the tangential adjoint velocity is free to change. This condition is similar to the no-penetration boundary condition of inviscid flows. No condition is required for the pressure.

Prescribed pressure. This condition is given by:

$$I_{p_0} = \int_{\Gamma}(p - p_0)^2\,\mathrm{d}\Gamma, \tag{28}$$

where p_0 denotes the prescribed pressure. From equation (19) this implies:

$$\mathbf{n}\cdot\Psi_{\mathrm{v}} = 2(p - p_0). \tag{29}$$

As earlier, the normal adjoint velocity is prescribed while the tangential adjoint velocity is free to change. No condition is required for the pressure.

3.3 Gradient evaluation

The gradient evaluation with respect to the design variables β is separated into three parts using the chain rule:

$$\frac{\partial I}{\partial \beta_m} = \frac{\partial y_l^j}{\partial \beta_m} \frac{\partial x_k^i}{\partial y_l^j} \frac{\partial I}{\partial x_k^i}. \tag{30}$$

The different expressions on the right hand side represent the gradient of surface points y with respect to changes in the design parameters β, the gradient of mesh points x with respect to the movement of surface points y, and the gradient of the cost function I with respect to the movement of the mesh points x. More importantly, they represent different parts of a comprehensive design methodology. The first expression is intimately linked to the choice of design variables and the second to the mesh movement technique selected.

3.3.1 Choice of design variables. One of the main objectives of the present design effort was to keep the algorithms as general as possible. This led to the decision to allow every point on the surface to be a design parameter. In order to obtain smooth surfaces with C^1 and C^2 continuity, a pseudo-shell approach (Soto and Löhner, 2002; Soto *et al.*, 2002) was adopted. In this way, the effect of a normal displacement for an arbitrary surface point i on all the surface points can be obtained. Once these influence coefficients have been obtained, the gradient $\mathbf{y}_{,\beta}$ can be quickly evaluated. The LU decomposition of the pseudo-shell matrix is obtained once and stored, speeding up the evaluation of gradients. The pseudo-shell approach has the intrinsic benefit of always providing a smooth surface, i.e. it has an inherent smoothing effect.

3.3.2 Mesh movement. After each design cycle, a new surface is obtained. In order not to regenerate the mesh, the interior grid points have to be moved in as smooth a way as possible (Nielsen and Anderson, 2001). In order not to have to remesh in the RANS region, a movement technique based on layer indexing is employed. Defining levels $l_i = 0, \ldots, n$ such that a point on the surface has $l_i = 0$ and any given point with $l_i > 0$ has at least one connection to a point of l_{i-1}, the displacements $\mathbf{v}_i$ for an arbitrary point i are given by:

$$\mathbf{v}_i = \frac{\sum_{j \in l_{i-1}} w_{ij} \mathbf{v}_j}{\sum_{j \in l_{i-1}} w_{ij}}, \tag{31}$$

where the edge-weights w_{ij} are computed from the distance d_{ij} as $w_{ij} = d_{ij}^{-2}$. For more details, see Soto and Löhner (2002) and Soto *et al.* (2002).

3.3.3 Field gradient evaluation. The third term of equation (30) can be computed independently of the design variables and mesh movement chosen. This gradient (of length ndimn*npoin) is computed and stored. Let us consider this third expression in more detail. Given the solution of the flow equations and the solution of the adjoint equations, the task is to obtain the gradients with respect to the movement of a point:

$$\frac{\partial I_L}{\partial x_k^i} = \left[\frac{\partial I}{\partial x_k^i} - \mathbf{\Psi}^{\mathrm{T}} \frac{\partial \mathbf{R}}{\partial x_k^i} \right]. \tag{32}$$

This expression is evaluated using *finite differences*. Each point is moved in the x, y, z direction, the residual $\mathbf{R}$ is evaluated and the gradient is computed. In theory, a new residual evaluation would be required for each point, raising the cost to $2{\cdot}3{\cdot}N_p$ evaluations (2 for central differences, 3 for the dimensions, N_p for the number of points).

However, one can use the fact that even with higher order schemes the points are connected by no more than nearest neighbours (+1) to reduce the required number of residual evaluations. The points are grouped (or coloured) in such a way that the point in each group are at least two neighbours away. This results in typically O(25-30) point groups, i.e. a total of O(150-180) (explicit) residual evaluations. An outer loop over these groups or colours is performed. The next inner loop is over the dimensions (the movement directions). The next inner loop is over the positive and negative increments. The geometrical parameters are recalculated, and a time step is performed using an explicit solver. Rewriting the conservation law given by equation (15) as a time-dependent problem:

$$\mathbf{u}_{,t} + \mathbf{F}^{i}_{,i} = \mathbf{u}_{,t} + \mathbf{R} = 0, \tag{33}$$

the increments in the unknowns are related to the residual by:

$$\mathbf{M}_{\mathrm{l}} \frac{\Delta \mathbf{u}}{\Delta t} = \mathbf{R}, \tag{34}$$

where $\mathbf{M}_{\mathrm{l}}$ denotes the lumped mass-matrix. From this expression, the residuals are evaluated. Note that using this procedure, the gradient can be computed in "black box fashion", allowing the use of different flow solvers. For the evaluation of the finite differences, the movement of each point typically corresponds to 1 per cent of the typical element length.

3.4 Optimization algorithm

Once the gradients $I_{,\beta}$ have been obtained, the solid boundary is updated using a steepest-descendent method as follows:

$$x^{i} := x^{i} - \lambda I_{,\beta_m} \delta^{mi}, \tag{35}$$

where δ^{mi} is the deformation mode associated with the design variable m (i.e. the displacement of the nodal point i induced for an imposed unitary displacement of design variable m), and λ a positive real constant. λ is taken in such a way that the total displacement of a gridpoint is only a fraction of the average length (l_i) of the edges surrounding the point (between 0.1 and $1.0l_i$), provided the gradient is greater than a small threshold value.

4. Geometric constraints

The optimization process typically involves a compromise of many factors. The classic example cited so often for fluid dynamics is the reduction of drag. It is well known that for a non-lifting body, the flat plate represents the optimum optimorum. Yet, flat plates, devoid of volume, cannot carry a payload, a prime requirement for a device. Therefore, a compromise between volume and drag is required. Another example from aerodynamics is the wing. In the transonic regime, thickness again increases (wave) drag. A thin wing, on the other hand, requires a stronger structure, i.e. more weight. Fuel capacity may at some stage also become a problem. As earlier, a compromise between volume and performance must be reached. The volume constraint is one of the many possible *geometric constraints*. Other constraints of this kind include:

- prescribed (min/max) volume;
- prescribed (min/max) cross-sectional area;
- prescribed (min/max) sectional thickness;
- prescribed (min/max) deviation from original shape; and
- prescribed (min/max) surface curvature;

These constraints may be treated in a variety of ways by:

- inclusion in the cost function;
- projection of the gradient; and
- post-processing the new shape; etc.

4.1 Volume constraint

This most common constraint may be added to the cost function in the form:

$$I = I^0 + I^V = I^0 + c_{\mathrm{V}} \left[\left(\frac{V - V_{\mathrm{O}}}{V_{\mathrm{O}}} \right) \right]^q, \tag{36}$$

where c_{V}, V_{O} denote a weight factor and a reference volume, respectively, and I_0 is the unconstrained cost function. The derivative is given by:

$$I^V_{,x} = c_{\mathrm{V}} 2q \left[\left(\frac{V - V_{\mathrm{O}}}{V_{\mathrm{O}}} \right)^2 \right]^{q-\frac{1}{2}} \frac{\mathrm{sign}(V - V_{\mathrm{O}})}{V_{\mathrm{O}}} V_{,x}. \tag{37}$$

Observe that for the common choice $q = 1/2$, the first term becomes unity, i.e. there is no effect from the deviation from the desired volume V_0, but the gradient simply flips from positive to negative with a constant weight factor. For this reason, we favour either $q = 1$ or 0.75. Numerically, the volume can be evaluated by:

- summing up the volumes of all elements (and subtracting it from a reference volume); and
- utilizing the divergence theorem.

We discuss the second option in more detail. The divergence theorem states that:

$$\int_\Omega \nabla \cdot \mathbf{v} \, \mathrm{d}\Omega = \int_\Gamma \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}\Gamma. \tag{38}$$

Given that the volume is

$$\int_\Omega \mathrm{d}\Omega,$$

we desire $\nabla \cdot \mathbf{v} = 1$, which can be obtained in a variety of ways, e.g. $\mathbf{v} = (x, y, z)/3$. We therefore, have:

$$V = \frac{1}{3}\int_{\Gamma}(x, y, z)\cdot \mathbf{n}\, d\Gamma. \tag{39}$$

This expression can be separated into x, y, z components, yielding

$$V = \frac{1}{3}\sum_{id=1,3} V_{id} = \frac{1}{3}\sum_{id=1,3}\int_{\Gamma} x_{id} n_{id}\, d\Gamma. \tag{40}$$

The separation into components is useful cases with planes of symmetry or unclosed objects, where one or more of the V_{id} components are incomplete.

5. Multipoint optimization

Shapes that are optimal for one given flow (angle of attack, speed, height), seldomly work well across a wide range of flight conditions (Drela, 1998; Li *et al.*, 2001). The classical example of this phenomenon is the Korn airfoil, which exhibits a shock-free behaviour at a certain Mach number and angle of attack. However, just changing slightly the Mach number or the angle of attack leads to the appearance of very strong shocks, making this type of airfoil unsuitable for airplanes. It has been observed (Jameson, 1988; Li *et al*, 2001; Reuther *et al.*, 1999) that the best way to steer a design away from such singular behaviour is to conduct a so-called multipoint optimization. In this case, the design has to produce a favourable cost-function for several flight conditions. This can be cast in the form:

$$I = \sum_{i} \gamma_i I(Ma_i, \alpha_i), \tag{41}$$

where I denotes the original cost function and Ma_i, α_i the flight conditions for which the multipoint design is carried out. Several schemes have been proposed to choose the weights γ_i (Li *et al.*, 2001), although in most cases they have been kept constant during optimization. Note that this is an additive cost function of different flight conditions (states), implying that gradient evaluations are also additive. The flow and adjoint solutions have to computed for each flight condition. Given that the CPU requirement of each one of these solutions is similar, this lends itself naturally to parallel processing. A current topic of research is concerned with the optimal (smallest) number of design points required to ensure a so-called *robust design* (Li *et al.*, 2001).

6. Examples

6.1 Nozzle

This example considers the thrust maximization of a nozzle for two different external Mach numbers: Ma = 3.0 and 5.0, with equal weighting for each Mach number. Hence, two flow and adjoint solutions were required per design cycle. The rest of the variables were set at the nozzle exit as follows: $\rho = 1.0$, $v = 1.0$, pr = 0.7, $\alpha = 0.0°$. The mesh consisted of 33 Ktet elements and 7 Kpts (2D example run with a 3D code). Figures 2-4 show the evolution of the shape, and the initial and final pressure and Mach number distributions for the first case. The thrust increased by approximately 78 per cent for both cases. Although 98 design cycles were run for the 307 design variables (every point on the nozzle surface), after 50 design cycles the additional benefit is small (Figure 5). The "buzzing" observed was due to the size of the descent step λ chosen.

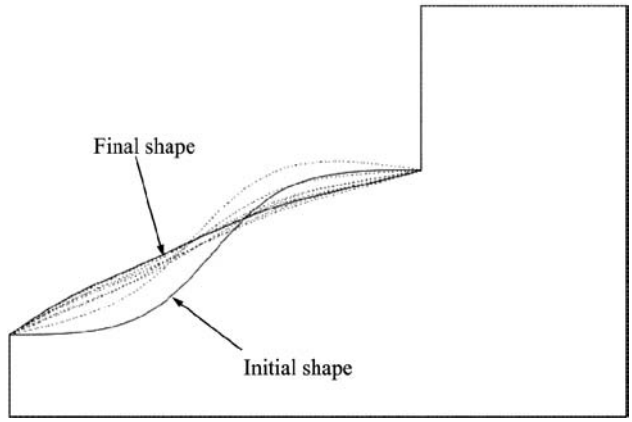

Figure 2.
Nozzle – evolution of shape

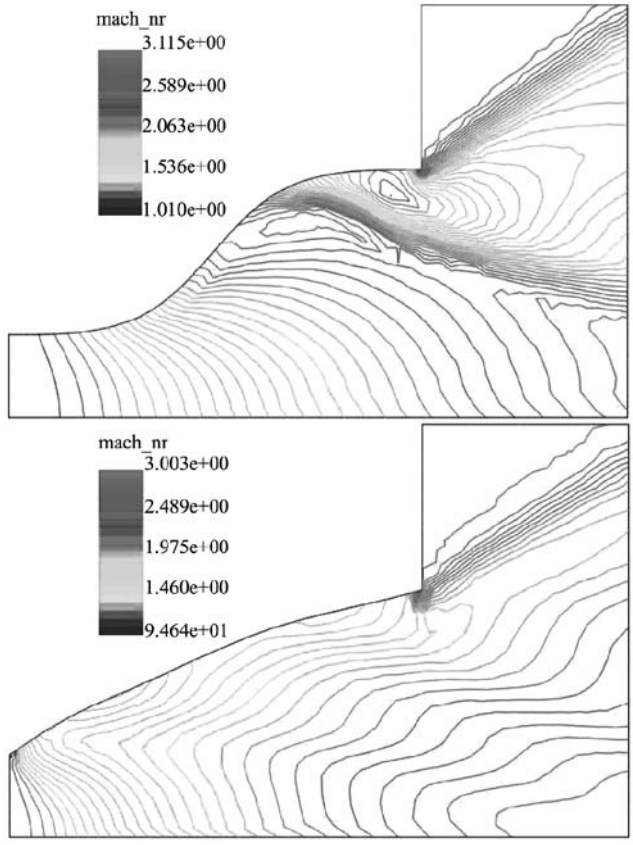

Figure 3.
Nozzle – Mach numbers for first and last shape

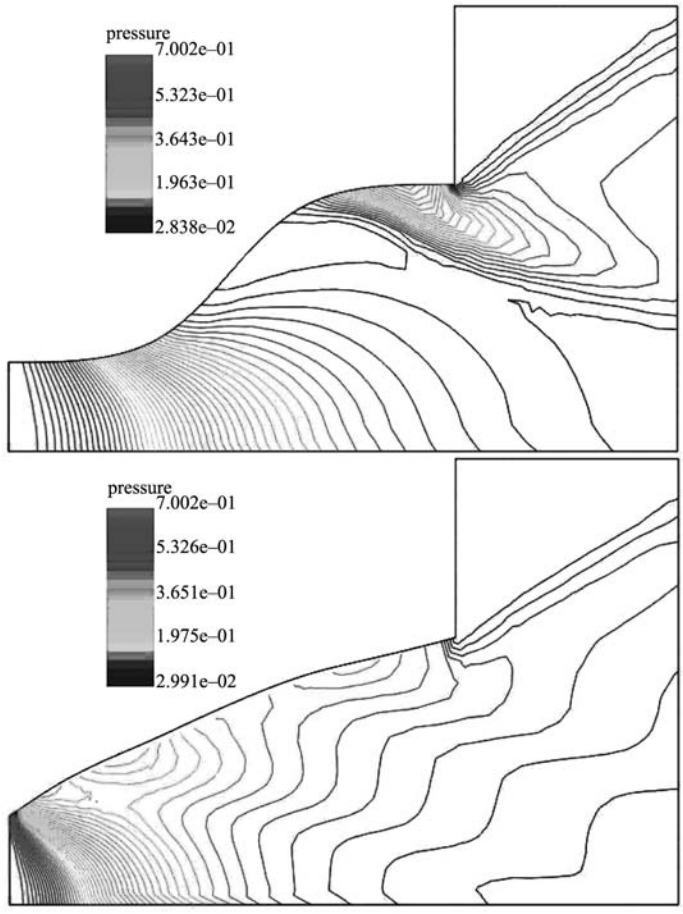

Figure 4.
Nozzle – pressure for first and last shape

At the beginning, this choice led to a smooth decrease of the objective function. However, as the design approaches the optimum, smaller steps should have been taken.

6.2 External nozzle

This example considers an external nozzle, typical of those envisioned for the X-34 airplane. There are no constraints on the shape. The objective is to maximize the thrust of the nozzle. The flow conditions were set as follows:

- Inflow (external flowfield): Ma = 3.00;
- $\rho = 0.5$, $v = 1.994$, pr = 0.15, $\alpha = 0.0°$;
- Nozzle exit: Ma = 1.01;
- $\rho = 1.0$, $v = 1.000$, pr = 0.18, $\alpha = -45.0°$

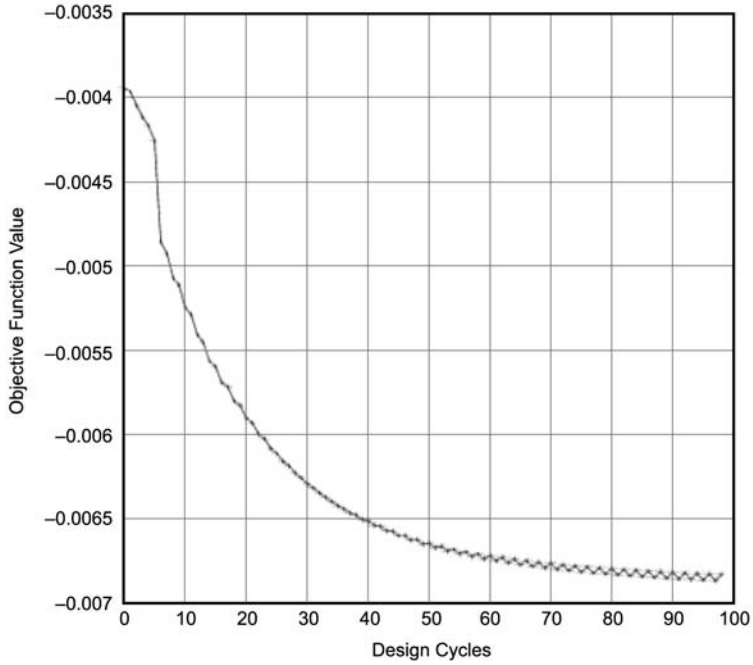

Figure 5.
Nozzle – objective function evolution

Although this is in principle a 2D problem, the case was run in 3D. The mesh had approximately 51 Kpts and 267 Ktet. Figures 6-13 show the initial and final Mach numbers and pressures, as well as the evolution of the shape and the thrust.

6.3 KRISO container ship (KCS)

This example considers a modern container ship with bulb bow and stern. The objective is to modify the shape of the bulb bow in order to reduce the wave drag. As an initial test, there are no constraints on the shape. The Froude-number was set to $Fr = 0.25$. The mesh had approximately 100 Kpts and 500 Ktet. The free surface had

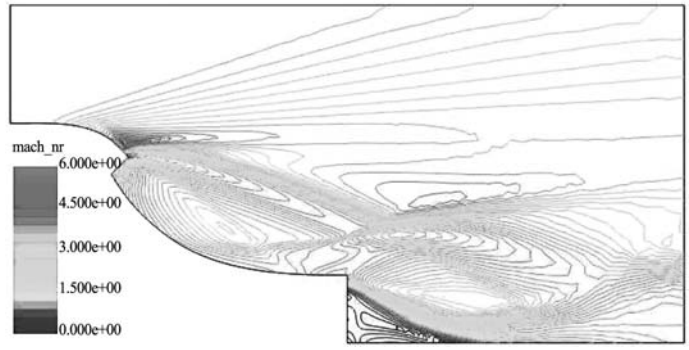

Figure 6.
External nozzle – Mach number for the first shape

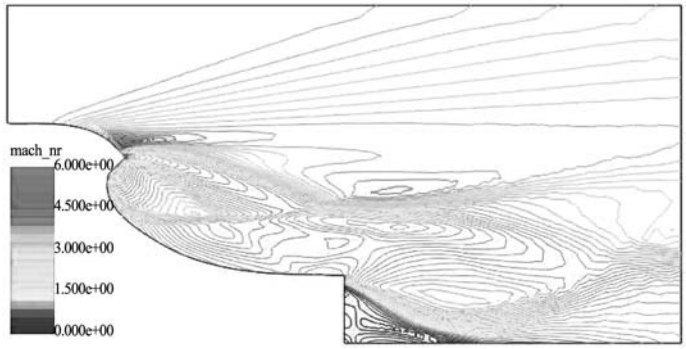

Figure 7.
External nozzle – Mach numbers for the last shape

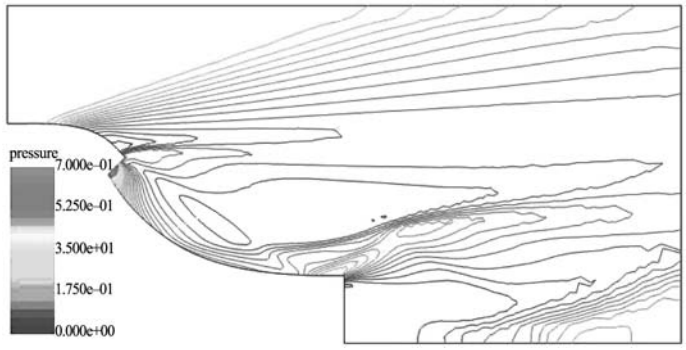

Figure 8.
External nozzle – pressure for the first shape

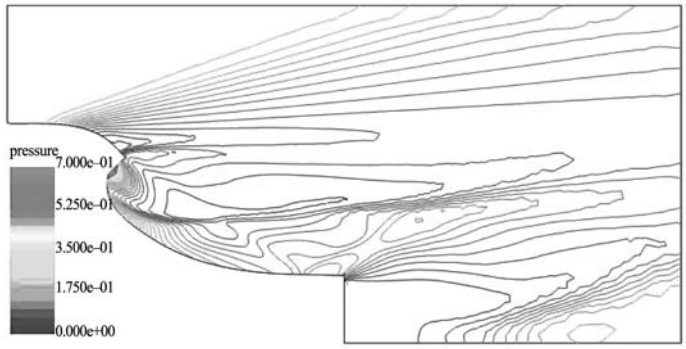

Figure 9.
External nozzle – pressure for the last shape

approximately 10 Kpts and 20 Ktria. Figure 14 shows both original and final mesh in the bulb bow region that correspond to the shape of the original and final bulb bow. Figure 15 shows the comparison of wave patterns generated by original hull form and final hull form. Figure 16 shows the comparison of wave drag coefficient obtained for both hull forms during the convergence of the flow solver to a steady-state. The wave

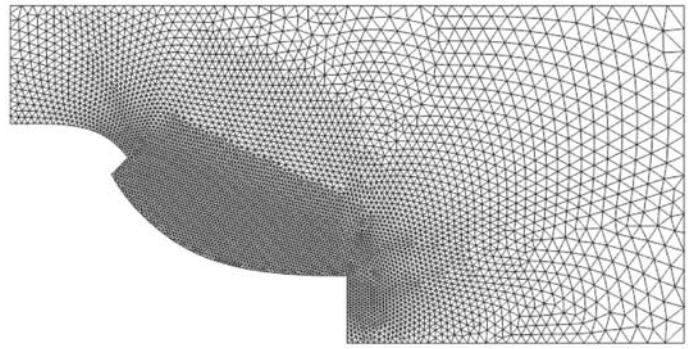

Figure 10.
External nozzle – surface mesh for the first shape

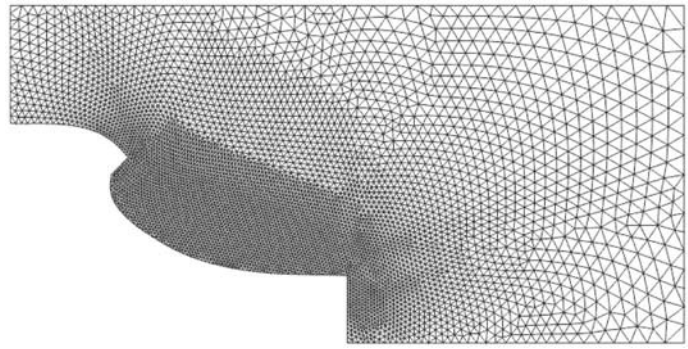

Figure 11.
External nozzle – surface mesh for the last shape

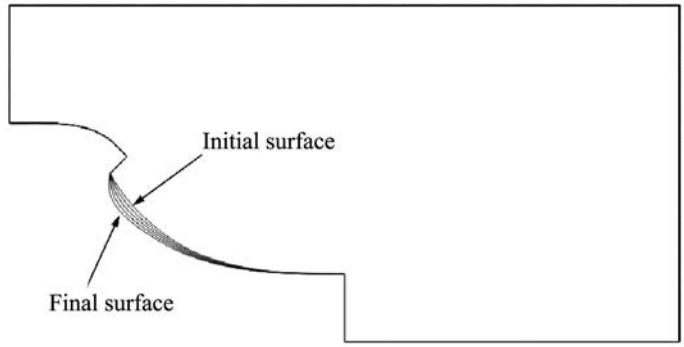

Figure 12.
External nozzle – evolution of shape

drag reduction was of the order of 4 per cent, which is considered significant given that the original shape was already deemed to be a very good design.

6.4 Air conditioning duct exit

The idea of this example is to obtain a quasi-uniform flow at the outlet of an air-conditioning duct. The initial mesh and geometry of the duct can be observed in

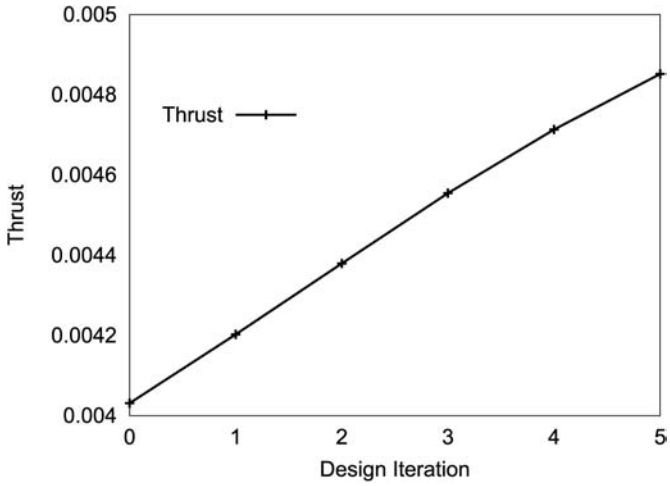

Figure 13.
External nozzle – evolution of thrust

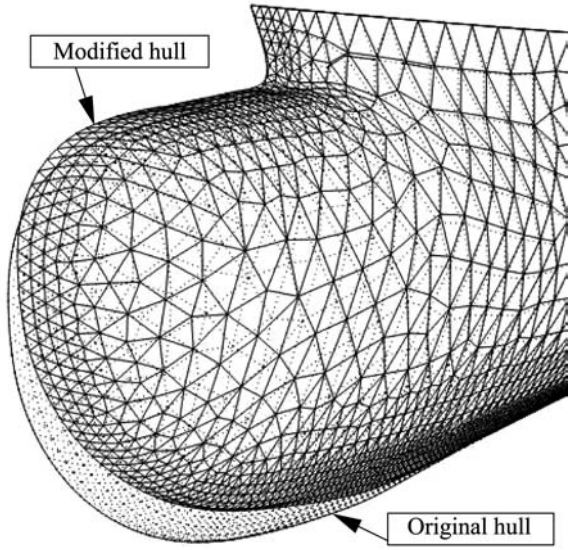

Figure 14.
KCS – surface mesh

Figure 17. This is a 2D incompressible flow application with a Reynold number, calculated with the width of the duct, of Re = 3.333 approximately. A Smagorinsky type turbulent model was utilized to obtain fully converged flow solutions. At the walls the normal velocity was set to zero, and a logaritmic law-of-the-wall tangential stress was applied. The pressure was set to zero at the outflow boundary, and a uniform velocity profile was prescribed at the inflow. For the adjoint problem, equation (19) dictates that at the outflow boundary the adjoint pressure must be set to

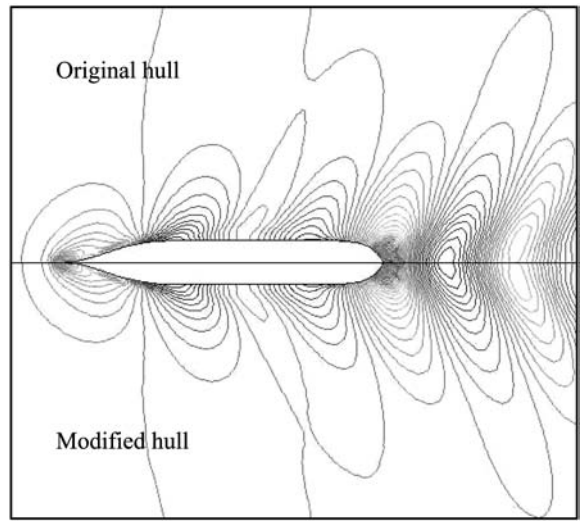

Figure 15.
KCS – wave pattern

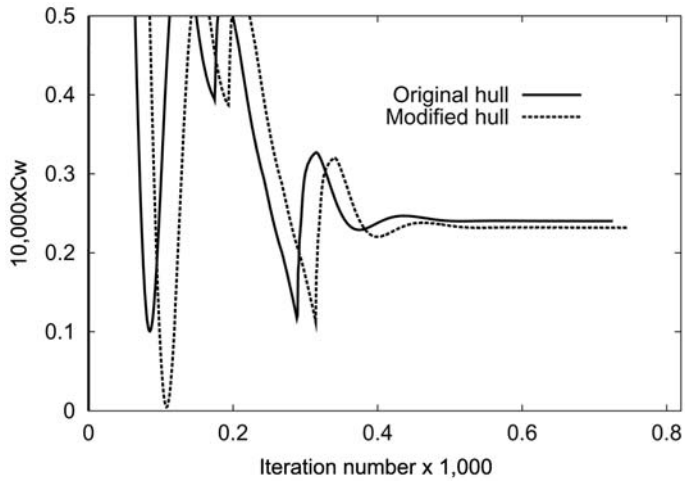

Figure 16.
KCS – wave drag coefficient

$\Psi_p = 2(u_0 - u_n)$ where u_0 is the target outflow velocity, and u_n the normal velocity obtained from the flow solver. At the same boundary, the adjoint velocities were prescribed zero. At the inflow, the adjoint pressure was set to zero and the adjoint velocities were left free to change. The normal adjoint velocity was set to zero at the walls, and the tangential one were left free. It can be demonstrated that such a set of boundary conditions are totally compatible with equation (19), making the problem well-posed. In Figure 18 the evolution of the shape can be observed. Note that towards the end, the cost function increases. This was again due to a stepsize λ that was too

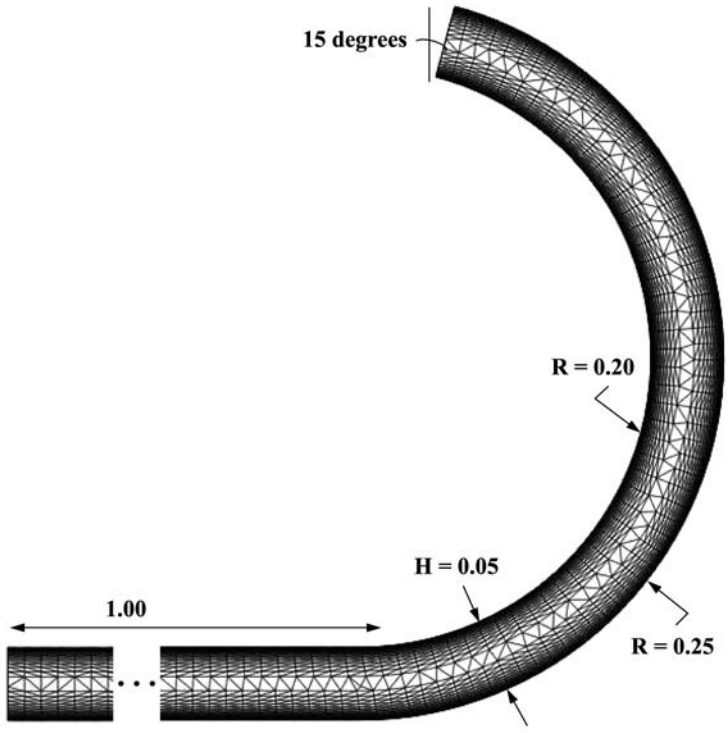

Figure 17.
Duct exit – mesh of 7,718 elements and 4,023 points

large. Only the curved part of the duct was allowed to change. The initial and final pressure distributions can be seen in Figure 19. It is noticeable how the pressure contours tend to be parallel to the outflow boundary to attempt to satisfy the required uniformity of the flow. The objective function was reduced from $I = 0.010$ to 0.005 in six design cycles, using 268 design variables (every point on the curved portion of the duct). This means a improvement of 50 per cent (Figure 20).

6.5 Generic sports utility vehicle

This numerical example considers the minimization of drag over the top part of a generic sports utility vehicle (SUV), subject to the constraint of minimum shape change. The initial mesh and geometry of the car can be observed in Figure 21. Taking advantage of the problem symmetry, just one half of the car was treated. This 3D incompressible application has a Reynolds number of 2×10^6, computed with the height of the car. Again, a Smagorinsky turbulence model was utilized to obtain fully converged flow solutions. At the car walls the normal velocity was set to zero, and a logaritmic law-of-the-wall tangential stress was applied. The pressure was set to zero at the outflow boundary, and a uniform velocity profile was prescribed at the inflow. In the rest of boundaries, the normal velocity was set to zero, and the tangential one remained free to change (symmetric planes). For the adjoint problem, equation (19) dictates that the right boundary condition for

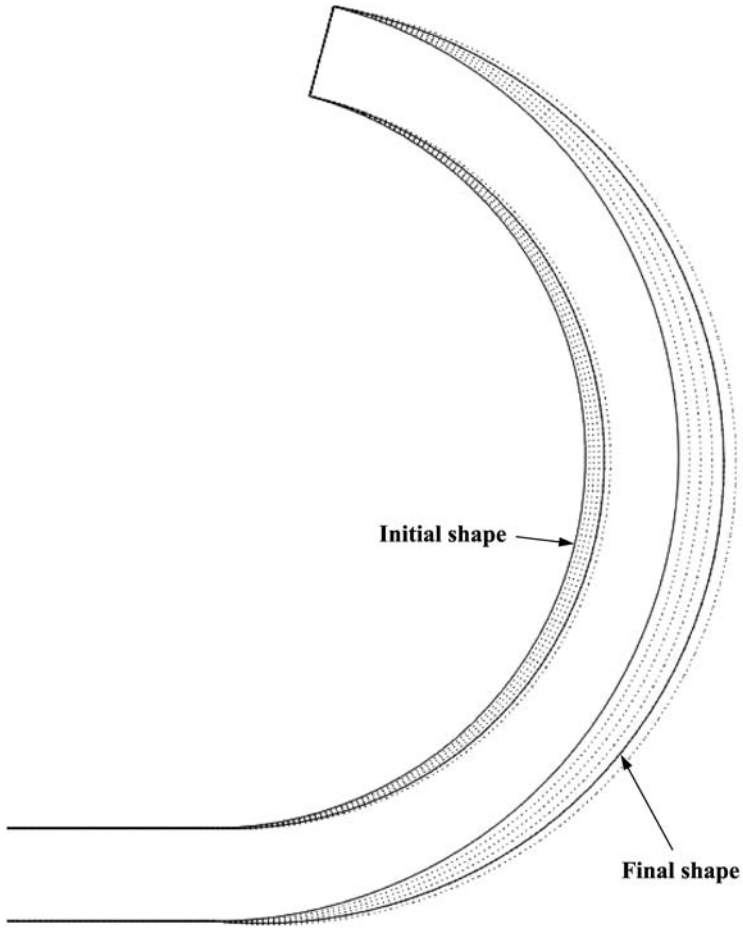

Figure 18.
Duct exit – shape evolution

the adjoint velocity on the surface to be optimized (the car surface), must be $\Psi_u = -e_d$, where e_d is the drag direction: (1,0,0) for this case. The adjoint pressure was set to zero at the inflow boundary, and the same value was given to the adjoint velocities at the outflow boundary. Only the adjoint normal velocity was prescribed zero in the rest of the boundaries.

In Figure 22, the evolution of the shape can be observed. The total normal displacement of the points on the car surface was limited to a preset displacement to avoid unrealistic solutions. In Figure 23, the drag evolution is shown. An improvement of approximately 14 per cent was achieved in nine design cycles using 944 design variables (every point on the car surface). Finally, in Figures 24 and 25 the pressure and velocity fields on the final geometry are shown. The effect of the law-of-the-wall stress over the velocity field can be noted.

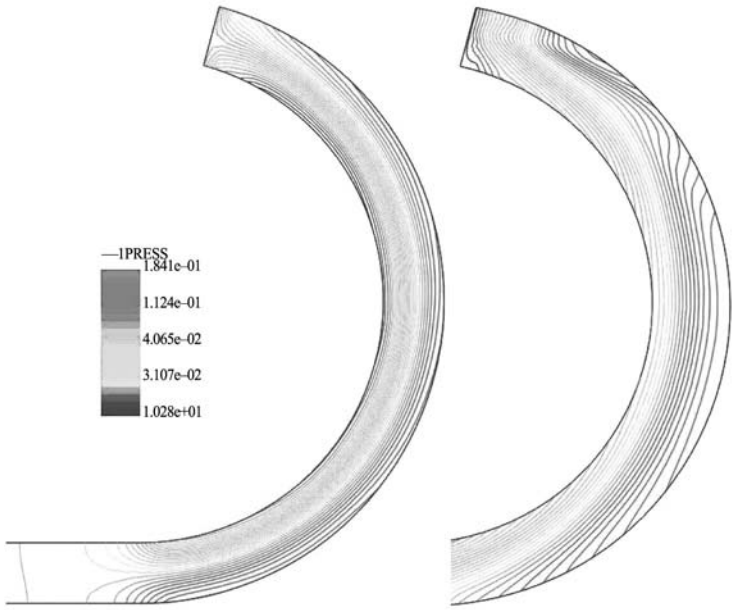

Figure 19.
Duct exit – initial and final pressures

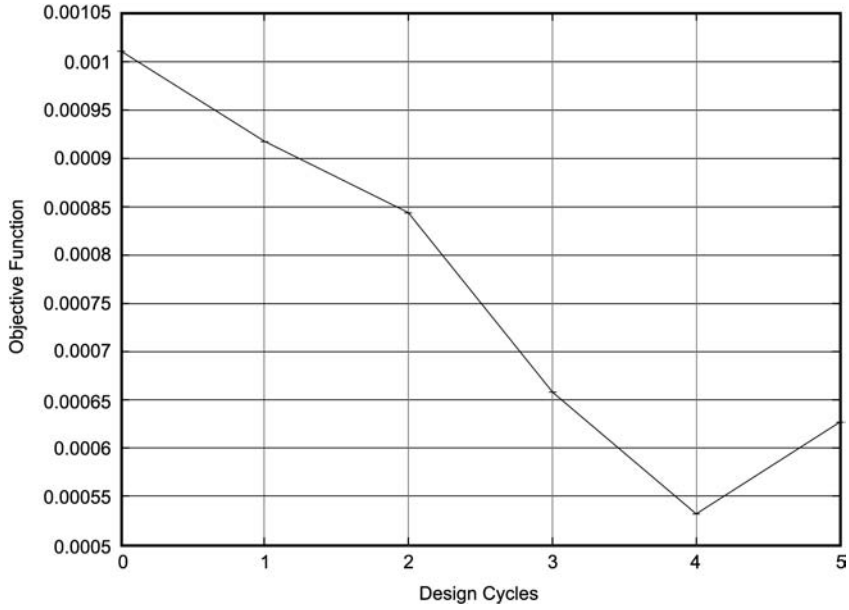

Figure 20.
Duct exit – objective function evolution

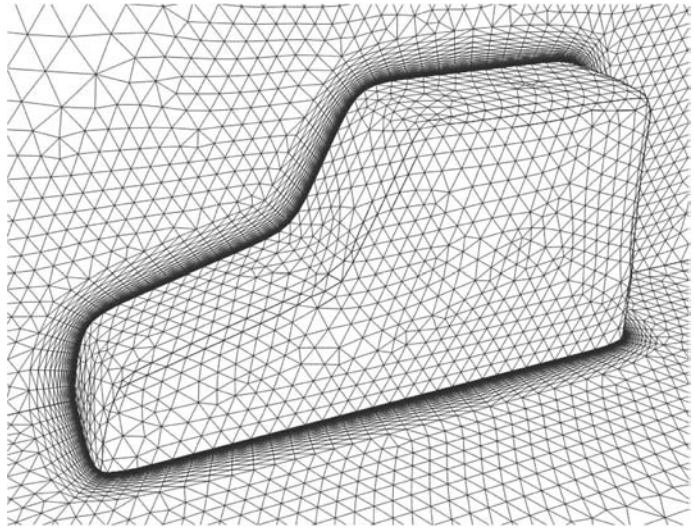

Figure 21.
SUV – mesh and geometry

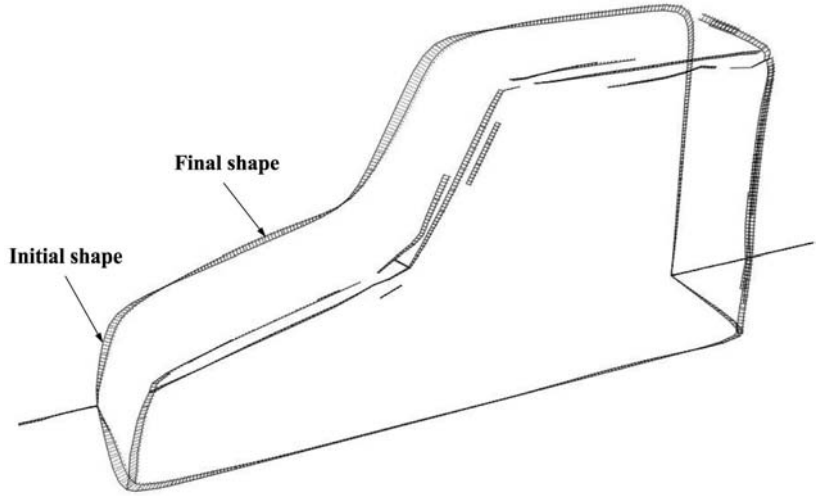

Figure 22.
SUV – shape evolution

7. Conclusions and outlook

An optimal shape design procedure based on the solution of the continuous adjoint has been implemented. Several results, spanning compressible and incompressible, viscous and inviscid flow, demonstrate the usefulness of the capabilities developed.

Future developments will center on:

- inclusion of propulsion effects;
- euler/boundary layer coupling, in particular for preliminary design;

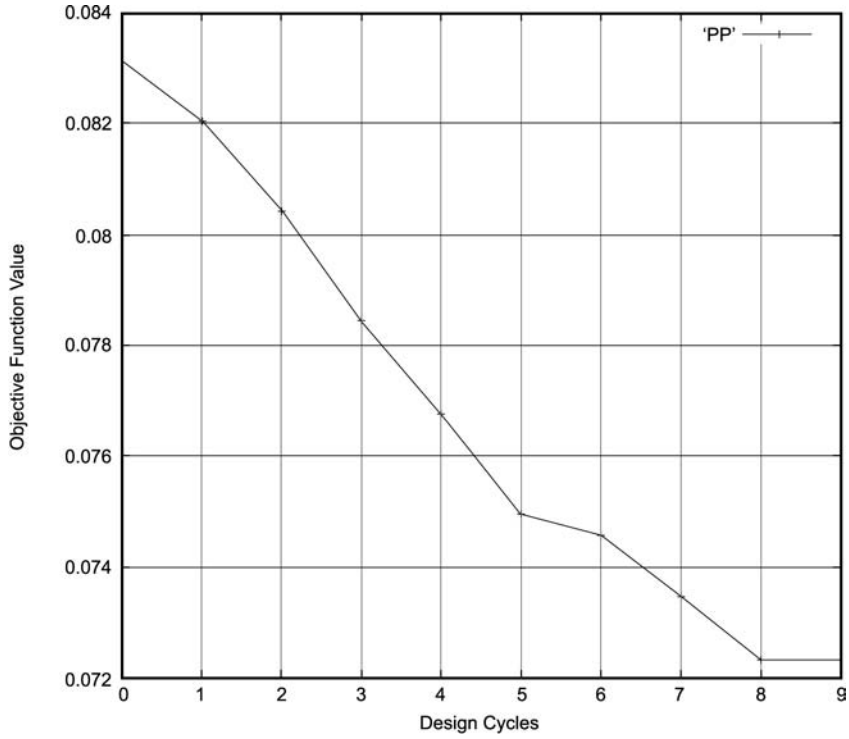

Figure 23.
SUV – objective function evolution

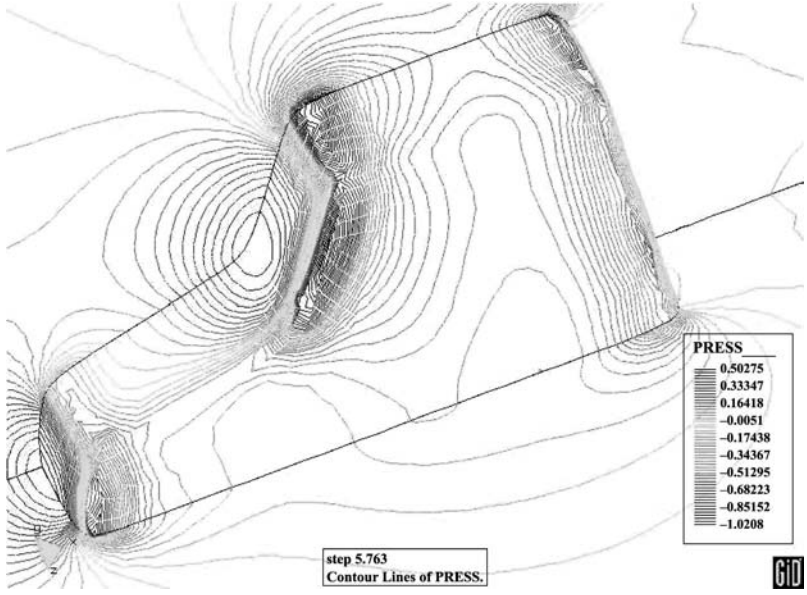

Figure 24.
SUV – pressure distribution on the final shape

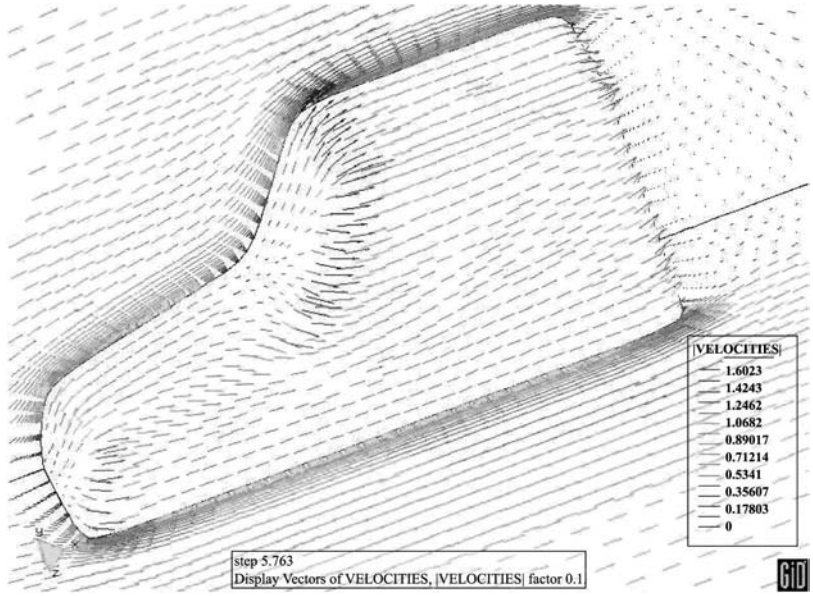

Figure 25.
SUV – velocity field on the final shape

- unsteady problems;
- better treatment of constraints;
- hierarchical design, i.e. the progressive matching of information content and physical realism; this implies combination of solvers (lifting line, potential, Euler, RANS,...), optimization techniques (genetic, gradient-based,...), and different representations of the design variables; and
- selection of proper cost function.

References

Anderson, W. and Venkatakrishnan, V. (1997), "Aerodynamic design optimization on unstructured meshes with a continuous adjoint formulation", AIAA-97-0643.

Crispin, Y. (1994), "Aircraft conceptual optimization using simulated evolution", AIAA-94-0092.

Drela, M. (1998), "Pros and cons of airfoil optimization", in Caughey, D.A. and Hafez, M.M. (Eds), Frontiers of CFD'98, World Scientific, Singapore.

Dreyer, J.J. and Matinelli, L. (2001), "Hydrodynamic shape optimization of propulsor configurations using a continuous adjoint approach", AIAA-01-2580.

Elliott, J. and Peraire, J. (1997), "Aerodynamic optimization on unstructured meshes with viscous effects", AIAA-97-1849.

Elliott, J. and Peraire, J. (1998), "Constrained, multipoint shape optimization for complex 3D configurations", *Aeronautical J.*, Vol. 102, pp. 365-76.

Gage, P. and Kroo, I. (1993), "A role for genetic algorithms in a preliminary design environment", AIAA-93-3933.

Gumbert, C.R., Hou, G.J.W. and Newman, P.A. (2001), "Simultaneous aerodynamic and structural design optimization (SASDO) of a 3D wing", AIAA-01-2527.

Jameson, A. (1988), "Aerodynamic design via control theory", *J. Scientific Computing*, Vol. 3, pp. 233-60.

Jameson, A. (1995), "Optimum aerodynamic design using control theory", CFD Review 1995, Wiley, New York, NY.

Korte, J.J., Salas, A.O., Dunn, H.J., Alexandrov, N.N., Follett, W.W., Orient, G.E. and Hadid, A.H. (1997), "Multidisciplinary approach to aerospike nozzle design", NASA TM-11036, NASA-LaRc.

Kuruvila, G., Ta'asan, S. and Salas, M.D. (1995), "Airfoil design and optimization by the one-shot method", AIAA-95-0478.

Li, W., Huyse, L. and Padula, S. (2001), "Robust airfoil optimization to achieve consistent drag reduction over a mach range", NASA/CR-2001-211042, ICASE Report No. 2001-22.

Löhner, R. (2001), *Applied CFD Techniques*, Wiley, New York, NY.

Löhner, R., Chi Yang, Cebral, J., Soto, O., Camelli, F., Baum, J.D., Luo, H., Mestreau, E. and Sharow, D. (2002), "Advances in FEFLO", AIAA-02-1024.

Löhner, R., Chi Yang, Oñate, E. and Idelssohn, S. (1999), "An unstructured grid-based, parallel free surface solver", *Appl. Num. Math.*, Vol. 31, pp. 271-93.

Luo, H., Baum, J.D. and Löhner, R. (1998), "A fast, matrix-free implicit method for compressible flows on unstructured grids", *J. Comp. Phys.*, Vol. 146, pp. 664-90.

Luo, H., Baum, J.D. and Löhner, R. (2001), "An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids", *Comp. and Fluids*, Vol. 30, pp. 137-59.

Medic, G., Mohammadi, B., Moreau, S. and Stanciu, M. (1998), "Optimal airfoil and blade design in compressible and incompressible flows", AIAA-98-2898.

Mohammadi, B. (1997), "A new optimal shape design procedure for inviscid and viscous turbulent flows", *Int. J. Num. Meth. Fluids*, Vol. 25, pp. 183-203.

Mohammadi, B. and Pironneau, O. (2001), *Applied Shape Optimization for Fluids*, Oxford University Press, Oxford.

Newman, J.C., Taylor, A.C., Barnwell, R.W., Newman, P.A. and Hou, G.J. (1999), "Overview of sensitivity analysis and shape optimization for complex aerodynamics configurations", *J. Aircraft*, Vol. 36 No. 1, pp. 87-96.

Nielsen, E. and Anderson, W. (1998), "Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations", AIAA-98-4809.

Nielsen, E. and Anderson, W. (2001), "Recent improvements in aerodynamic design optimization on unstructured meshes", AIAA-01-0596.

Quagliarella, D. and Cioppa, A.D. (1994), "Genetic algorithms applied to the aerodynamic design of transonic airfoils", AIAA-94-1896-CP.

Reuther, J., Jameson, A., Alonso, J.J., Rimlinger, M.J. and Saunders, D. (1999), "Constrained multipoints aerodynamic shape optimization using an adjoint formulation and parallel computers", *J. Aircraft*, Vol. 36 No. 1, pp. 51-74.

Sharov, D., Luo, H., Baum, J.D. and Löhner, R. (2000), "Implementation of unstructured grid GMRES+LU-SGS method on shared-memory, Cache-based parallel computers", AIAA-00-0927.

Soto, O. and Löhner, R. (2001a), "CFD shape optimization using an incomplete-gradient adjoint formulation", *Int. J. Num. Meth. Eng.*, Vol. 51, pp. 735-53.

Soto, O. and Löhner, R. (2001b), "General methodologies for incompressible flow design problems", AIAA-01-1061.

Soto, O. and Löhner, R. (2002), "A mixed adjoint formulation for incompressible RANS problems", AIAA-02-0451.

Soto, O., Löhner, R. and Yang, C. (2002), "A stabilized pseudo-shell approach for surface parameterization in CFD design problems", *Comm. Num. Meth. Eng.*, Vol. 18 No. 4, pp. 251-8.

Further reading

Mohammadi, B. (1999), "Flow control and shape optimization in aeroelastic configurations", AIAA-99-0182.

Appendix 1. Compressible Euler equations

The compressible Euler equations are given by:

$$\mathbf{u}_{,t} + \nabla \cdot \mathbf{F} = 0,$$

where

$$\mathbf{u} = \left\{ \begin{array}{c} \rho \\ \rho v_i \\ \rho e \end{array} \right\}, \quad \mathbf{F}^j = \left\{ \begin{array}{c} \rho v_j \\ \rho v_i v_j + p\delta_{ij} \\ v_j(\rho e + p) \end{array} \right\}.$$

Here ρ, p, e, v_i denote the density, pressure, specific total energy and fluid velocity in the direction x_i, respectively. This set of equations is closed by providing an equation of state for the pressure, e.g. for a polytropic gas:

$$p = (\gamma - 1)\rho\left[e - \frac{1}{2}v_j v_j\right],$$

where γ is the ratio of specific heat. Denoting $u = v_1,\ v = v_2,\ w = v_3,\ q = u^2 + v^2 + w^2,$ $c_1 = \gamma - 1,\ c_2 = c_1/2,\ c_3 = 3 - \gamma,\ c_4 = \gamma e - c_1 q,\ c_5 = \gamma e - c_2 q,$ the Jacobian matrices are given by:

$$\mathbf{A}^x = \left\{ \begin{array}{ccccc} 0 & 1 & 0 & 0 & 0 \\ c_2 q - u^2 & c_3 u & -c_1 v & -c_1 w & c_1 \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ -c_4 u & c_5 - c_1 u^2 & -c_1 uv & -c_1 uw & \gamma u \end{array} \right\},$$

$$\mathbf{A}^y = \left\{ \begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ c_2 q - v^2 & -c_1 u & c_3 v & -c_1 w & c_1 \\ -uw & 0 & w & v & 0 \\ -c_4 v & -c_1 uv & c_5 - c_1 v^2 & -c_1 vw & \gamma v \end{array} \right\},$$

$$\mathbf{A}^z = \begin{Bmatrix} 0 & 0 & 0 & 1 & 0 \\ -uv & w & 0 & u & 0 \\ -uw & 0 & w & v & 0 \\ c_2 q - w^2 & -c_1 u & -c_1 v & c_3 w & c_1 \\ -c_4 w & -c_1 uw & -c_1 uw & c_5 - c_1 w^2 & \gamma w \end{Bmatrix}.$$

Appendix 2. Incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations are given by:

$$\mathbf{u}_{,t} + \nabla \cdot \mathbf{F} = \nabla \mu \nabla \mathbf{w},$$

where

$$\mathbf{u} = \begin{Bmatrix} p/c^2 \\ v_i \end{Bmatrix}, \quad \mathbf{F}^j = \begin{Bmatrix} v_j \\ v_i v_j + p\delta_{ij} \end{Bmatrix}, \quad \mathbf{v} = \begin{Bmatrix} 0 \\ v_i \end{Bmatrix}.$$

Here p, c, v_i denote the pressure, (infinite) speed of sound and fluid velocity in direction x_i, respectively. The Jacobian matrices and the **B**-matrix are given by:

$$\mathbf{A}^x = \begin{Bmatrix} 0 & c^2 & 0 & 0 \\ 1 & 2u & 0 & 0 \\ 0 & v & u & 0 \\ 0 & w & 0 & u \end{Bmatrix}, \quad \mathbf{A}^y = \begin{Bmatrix} 0 & 0 & c^2 & 0 \\ 0 & v & u & 0 \\ 1 & 0 & 2v & 0 \\ 0 & 0 & w & v \end{Bmatrix}, \quad \mathbf{A}^z = \begin{Bmatrix} 0 & 0 & 0 & c^2 \\ 0 & w & 0 & u \\ 0 & 0 & w & v \\ 1 & 0 & 0 & 2w \end{Bmatrix},$$

$$\mathbf{B} = \begin{Bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix}.$$