
Theory and practice of optimal shape design

Bijan Mohammadi* — **Olivier Pironneau****

** Mathematics & Modelling Institute, University of Montpellier
bijan.mohammadi@univ-montp2.fr*

*** Laboratory J.-L. Lions, University Paris VI
pironneau@ann.jussieu.fr*

ABSTRACT. This paper is a short survey of some recent developments in Optimal Shape Design (OSD) for fluids. Existence, sensitivity, compatibility of discretizations as well as efficient algorithmic implementations with low complexity are critical. In this paper we will discuss some of these issues with application to shape optimization for aerodynamic noise reduction.

RÉSUMÉ. L'article présente quelques développements récents en conception optimale de formes pour les écoulements de fluides. Existence de solutions, calculs de gradients, compatibilité entre discrétisation et formulation continue, mais aussi implémentations efficaces sont importants. Nous abordons quelques-uns de ces concepts et une application d'optimisation de forme pour réduction de bruit aérodynamique.

KEYWORDS: global optimization, shape design, parameterization, reduced order model, incomplete sensitivity.

MOTS-CLÉS : optimisation globale, conception de formes, paramétrisation, modèle réduit, gradient incomplet.

DOI:10.3166/REM.N.17.13-30 © 2008 Lavoisier, Paris

1. Introduction

Control is a natural desire once the simulation is completed. The applications of optimal shape design (OSD) in industry are uncountable and increasingly involve multidisciplinary physics. Different physics to account for makes that optimum design is not a once and for all solution tool and involves compromises with the necessity of doing *multi-point constrained design*. Not talking about noise in data which is a major issue enforcing needs for robust optimization. This multi-criteria-multi-point situation requires cheap robust global optimization tools with, as much as possible, sub-solution approaches to break the complexity of the problem.

In multi-criteria optimization sensitivity analysis is important to discriminate between Pareto points and this even if a gradient free approach is used. Indeed, the knowledge of sensitivity permits to qualify various points of a Pareto front from the point of view of robustness. Indeed, two points on a Pareto front can be compared if one considers the sensitivity of the functional with respect to the independent variables which are not control parameter. The robust optimum is the one with lowest sensitivity.

Also, sensitivity evaluation is important because often in simulations information on the uncertainties on the results is more important than the results themselves. For instance, it is essential to be able to identify dominant independent variables in a system. As these will need more accurate monitoring and for which precise measurements should be provided.

Sensitivity analysis is also useful to qualify the impact on the results of a given modelling, or evolution of the modelling, and also the way a model is discretized.

These concepts are central in Validation and Verification (V&V) issues which refer to all of the activities that are aimed at making sure that a software will function as required. Indeed, it is important to include robustness issues into the specifications using sensitivity analysis and see that a simulation should be therefore seen as multi-criteria minimization.

Indeed, consider the following simulation loop linking a set of independent to dependent variables and eventually leading to the calculation of a functional $J(x, \varphi, h)$ function of independent variables x , for instance, a geometric parameterization, φ the physical variables defining the flow and h the solution procedure parameters (discretization, accuracy, convergence rates...)

$$(x, \varphi, h) \rightarrow q(x, h) \rightarrow U(\varphi, q(x, h)) \rightarrow J(x, \varphi, h, q(x, h), U(\varphi, q(x, h))) \quad [1]$$

Flow calculations enter this class where q denotes geometric quantities and U field flow variables solution of the state equation $F(U(q(x, h), \varphi, h)) = 0$. For a geometrical set of parameters (x^*) and flow conditions (φ^*), the solution of the state equation can be seen as minimization of

$$J_1(x^*, \varphi^*, h) = \|F(U(q(x^*, h), \varphi^*, h))\|$$

Hence, we look for the best solution procedure which minimizes the residual: $h = \text{Argmin}_{h \in H} J_1(x^*, \varphi^*, h)$. The admissible space H includes the constraint on the so-

lution procedure (e.g. maximum discretization one can afford, accuracy in the solution of nonlinear and linear systems...).

Robustness issues can now be introduced through the control of the sensitivity with respect to the other independent variables around the functioning point (x^*, φ^*) :

$$J_2(x^*, \varphi^*, h) = \|\nabla_{x,\varphi} J_1(x^*, \varphi^*, h)\|$$

The simplest way to proceed is by penalizing these sensitivities and looking for the solution of $h = \text{Argmin}_{h \in H} J_1(x^*, \varphi^*, h) + J_2(x^*, \varphi^*, h)$. This is because no calculation can be reliable if it is too sensitive to small perturbations in data. This also shows that the solution procedure needs probably changes once including robustness issues, except if the optimum of the constrained and unconstrained problems are the same, which is quite unlikely. Despite the cost of this approach, one should however pay attention and at least *a posteriori* evaluate the sensitivity of the solution to perturbation of independent variables (Su *et al.*, 1997).

2. Global optimization and free surfaces

Discrete optimal time control problem and OSD have similar structure. Thus many of the tools of control theory and of the Calculus of Variations have been extended to PDEs and we shall use them to solve OSD problems numerically.

Consider the minimization of a functional $J(S)$, $S \in \mathcal{O}_{ad}$, S is the shape parameterization and belongs to a admissible space \mathcal{O}_{ad} (we show some examples of shape parameter in the next section).

Most minimization algorithms can be seen as discretizations of (Ivorra *et al.*, 2006; Mohammadi *et al.*, 2003):

$$\begin{cases} M(S(\zeta))S_\zeta = -d(S(\zeta)) \\ S(\zeta = 0) = S_0 \end{cases} \quad [2]$$

M is aimed to be positive definite and $M^{-1}d$ is built to be an admissible direction.

Assume the infimum J_m of J known and the problem admissible (*i.e.* the infimum is reached inside the admissible domain: $\exists S_m \in \Omega_{ad}$, *s.t.* $J(S_m) = J_m$ & $J'(S_m) = 0$).

Global solution of [2] means, for instance, finding $S_m = S(1)$ verifying

$$\begin{cases} M(S(\zeta))S_\zeta = -d(S(\zeta)) \\ S(0) = S_0 \\ J(S(1)) = J_m \end{cases} \quad [3]$$

This is an over-determined boundary value problem. This over-determination tells us why one should not solve global optimization problems with methods which are dis-

crete form of Cauchy problems for first order differential systems. We propose to consider global shape optimization solving the following boundary value problem:

$$\begin{cases} \eta S_{\zeta\zeta} + M(S(\zeta))S_{\zeta} = -d(S(\zeta)), \\ J(S(1)) = J_m, \quad J'(S(1)) = 0 \end{cases} \quad [4]$$

We have dropped the initial condition on S which is misleading in the context of global optimization. This can be solved using solution techniques for BVPs with free surface. Indeed, an analogy can be given with the problem of finding the interface between water and ice which is only implicitly known through the iso-value of zero temperature.

In the sequel d will be based on approximation of J' and M will be a smoothing operator for the CAD-free or level set parameterization (see section 3.2). There is a global controllability result for this system if J is $C^2(\Omega_{ad}, \mathbb{R})$ and coercive, $M = I$ and $d = J'$. System [4] is solved using the recursive semi-deterministic shooting algorithm (Ivorra *et al.*, 2006).

3. A model problem

Consider the academic problem of designing one boundary S of a domain Ω with required flow properties ψ_d in a region of space D . ψ_d is defined outside D to be used as boundary condition too. We assume that the flow is potential and two dimensional. With a stream function formulation this would be

$$\min_{S \in \mathcal{S}_d} \{j(S) := \int_D |\psi - \psi_d|^2 : -\Delta\psi = 0, \text{ in } \Omega \quad \psi|_S = 0 \quad \psi|_C = \psi_d\} \quad [5]$$

where $C = \Gamma \setminus S$ and $\Gamma = \partial\Omega$.

It can be discretized by

$$\min_{\mathcal{T}_h} \{j_h := \int_D |\psi - \psi_d|^2 : \int_{\Omega} \nabla\psi_h \nabla w_h + \frac{1}{\epsilon} \int_C (\psi_h - \psi_d) w_h = 0 \quad \forall w_h \in V_h\} \quad [6]$$

where V_h is the finite element space of piecewise linear continuous functions on the triangulation \mathcal{T}_h of Ω ; h denotes the average edge length in the triangulation. If w^i denotes the function of V_h which is one at vertex q^i and zero at all other vertices, then, with $\Psi_j := \psi_h(q^j)$,

$$\psi_h(x) = \psi_d(x) + \sum_{i \notin \Gamma} \Psi_i w^i(x)$$

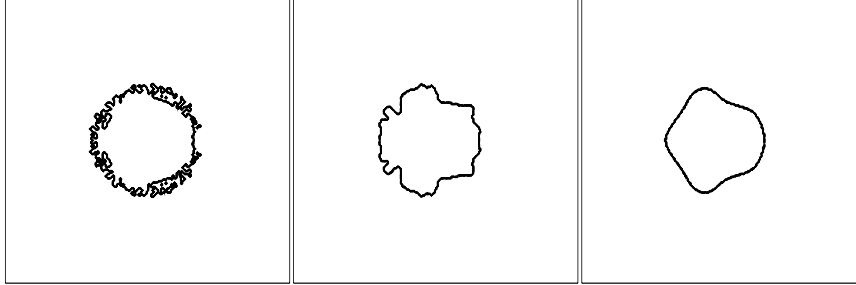


Figure 1. Effect of the regularization operator on the level set function after few iterations of a descent method for a drag reduction problem. Iso-lines show $\text{iso-}\chi(\psi)$. Dislocations appear if regularity is not monitored. This is however useful if one looks for topology changes

and, problem [6] is of the form

$$\min_{\vec{q}} \Psi^T B(\vec{q}) \Psi : A(\vec{q}) \Psi = F(\vec{q}) \quad [7]$$

where $A_{ij} = (\nabla w^i, \nabla w^j)$, $B_{ij} = (w^i, w^j)$, and F is the discrete Laplacian of ψ_d . It is clear that these depend on the position of all the vertices (stored here in the vector \vec{q}) and not just of the vertices of S . In the sequel, we will see how to break this complexity with sub-solution methods.

3.1. Existence of solution

Many of optimal shape design problems do not have solutions. For example, if $\psi_d \in L^2(\Omega)$ but $\psi_d \notin H^1(\Omega)$, [5] does not have a solution because $\psi \rightarrow \psi_d$ is possible yet $\psi = \psi_d$ is impossible.

Existence can be studied in several ways and it is interesting to notice that each way gives rise to a different numerical method. Below we consider three ways to parameterize the shape (direct shape parameter S , using mapping T and using characteristic functions χ).

Most existence results are obtained by considering minimizing sequences S^n , (resp. T^n or χ^n) and, in the case of our academic example, showing that $\psi^n \rightarrow \psi$ for some ψ when $S^n \rightarrow S$ (resp. $T^n \rightarrow T$ or $\chi^n \rightarrow \chi$), and that the PDE is satisfied at the limit.

3.2. Direct implicit or explicit shape parameterization

In shape optimization, different choices can be made for the parameter space following the variety of the shapes one would like to reach. If the topology of the target shape is already known and if the available CAD parameter space is thought to be enough rich, it should be considered as control parameter space during optimization. On the other hand, one might use a different parameter space, larger or smaller, during optimization having in mind that the final shape should be expressed in a CAD format for being usable.

Using continuity results with respect to domain boundaries (Pironneau, 1984; Delfour *et al.*, 2001), the unknown is an implicit or explicit parametrization of the boundary. One gives below an example of each parameterization. Although the set of admissible boundaries is not easily endowed with a vector space structure, it is easy to define boundary variations which have a Hilbertian structure. For instance, normal variations by $\alpha(x)$, $x \in S$ around a reference boundary S of normal $\vec{n}(x)$, would be

$$S(\alpha) = \{x + \alpha(x)\vec{n}(x) : x \in S\}. \quad [8]$$

By using regularity results with respect to the domain (Chenais, 1987) (see also (Neittaanmaki, 1991) and (Delfour *et al.*, 2001)) showed that in the class of all S uniformly Lipschitz, problem [5] has a solution. However the solution may depend upon the Lipschitz constant.

3.2.1. CAD-free shape parameterization

This is the simplest explicit shape parameterization: all the nodes of the surface mesh over the shape are control parameters. One particularity of this parameterization comes from the fact that, unlike in a CAD-based parameter space, regularity requirements have to be specified and handled by the user. Indeed, if the shape is described using a CAD tool and if we use the same parameterization to specify the deformations, the two entities belong to the same space in term of regularity.

Suppose Γ is a surface in a domain $\Omega \in R^3$ and suppose we want shape variations $\delta x \in C^1(\Gamma)$. In the context of shape optimization, applying to a C^1 shape Γ a gradient method does not necessarily produce a $C^1(\Gamma)$ variation δx . Actually, the variation is rather in $L^2(\Gamma)$ and therefore we need to project the variations into $H^{5/2}(\Gamma)$ for instance. The smoothing can also be seen as a modification of the metric on which minimization is performed (Mohammadi *et al.*, 2004; Mohammadi *et al.*, 2001). Figure 3 shows the effect of regularization operators on the CAD-free parameterization for a full aircraft shape optimization problem. One sees that beyond topology considerations (the topology is unchanged in both parameterizations), one major issue in CAD-based parameter spaces is that it fixes the regularity of the optimal shape at the beginning of the optimization, while CAD-free parameterizations leave open the question of final optimal regularity.

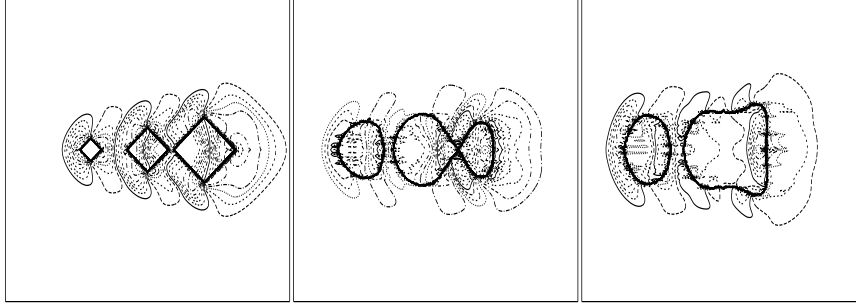


Figure 2. *Topology change with level sets with regularity control*

3.2.2. Level sets

The level set method is an established technique to represent moving interfaces ((Osher *et al.*, 1988; Dervieux *et al.*, 1979; Dervieux *et al.*, 1981)). Level set is an implicit parameterization of the shape based on the signed distance function ψ to the shape. The shape is known as the iso-0 contour of ψ . Equation [4] generalizes the Hamilton-Jacobi type equation used for the motion of the level set parameter:

$$\psi_\zeta + V\nabla\psi = 0$$

with $V = \nabla J \cdot n$ where the local normal to the iso-contours of ψ is defined by $n = \nabla\psi/|\nabla\psi|$. The variation of ψ are then given by

$$\psi_\zeta = -\nabla J|\nabla\psi|$$

We see that the particular choice of $\eta = 0$, $M(S(\zeta)) = I$ and $d = \nabla J|\nabla\psi|$ gives the classical equation used for the motion of the level set function. ψ known, we account for the boundary conditions in the state equation using a relaxed normalized distance function $\chi(\psi)$, ($0 \leq \chi(\psi) \leq 1$). One important issue here is that one needs to control regularity as for the CAD-free parameterization (Mohammadi, 2007). Figures 4, 1 and 2 show examples of optimizations with level set parameterization and effects of regularization operator.

3.2.3. Mapping and transformation

One may also map the unknown domain Ω from a fixed domain O and consider that the unknown is the mapping $T : O \rightarrow \Omega$. Denote by T' its Jacobian matrix, let $\hat{\psi}_d$ be $\psi_d \circ T$ with ψ_d extending the given boundary conditions and the requirement in

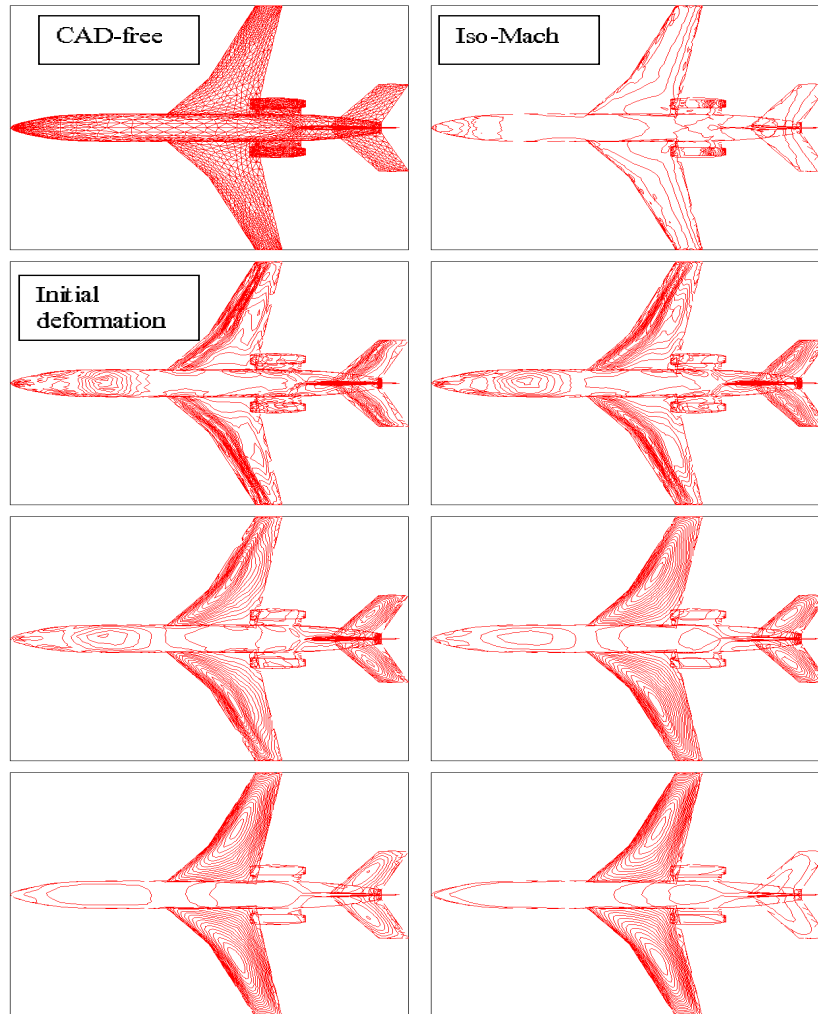


Figure 3. Effect of the regularization operator on the CAD-free parameterization for a full aircraft shape optimization problem. Upper line: surface mesh used as parameterization and iso-Mach contours for an inviscid calculation. Successive levels of regularization permits to look for the most suitable search direction and final regularity for the shape

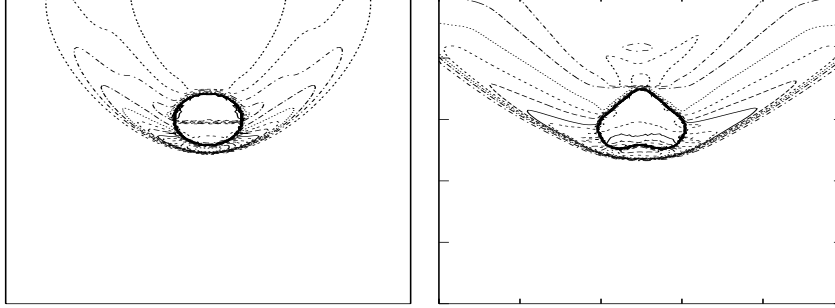


Figure 4. Finding the right shape to enter the atmosphere using a level set parameterization. The final shape does not have the same regularity at leading and trailing edges than the initial guess. Pictures show $\text{iso-}\chi(\psi)$ and iso-density contours

D (recall that $\psi_d = 0$ on S and is constant on the upper wall of the nozzle). Then we may solve

$$\min_{T \in \mathcal{T}_d} \int_{\hat{D}} |\hat{\psi} - \hat{\psi}_d|^2 : \nabla \cdot [A \nabla \hat{\psi}] = 0 \text{ in } O \quad [9]$$

together with $\hat{\psi}|_{\partial \hat{O}} = \hat{\psi}_d$ and $A = T'^{-1T} T'^{-1} \det(T')$.

As for [5] it is also possible to work with a local (tangent) variation $tV(x)$ and set

$$\Omega(tV) = \{x + tV(x) : x \in \Omega\} \quad t \text{ small and constant} \quad [10]$$

(Murat *et al.*, 1976) working with [9] showed that in the class of $T \in W^{1,\infty}$ uniformly, the solution exists.

3.2.4. Topological optimization

This last approach, suggested by (Tartar, 1974) has led to what is referred now as *topological optimization*. It extends the operators by zero below S and take the characteristic function of Ω , χ , for unknown

$$\min_{\chi \in X_d} \left\{ \int_D |\psi - \psi_d|^2 : -\nabla \cdot [\chi \nabla \psi] = 0, \quad \psi(1 - \chi) = 0, \quad \psi|_{\partial \Omega} = \psi_d \right\} \quad [11]$$

It may be difficult to work with the function χ , then, following (Allaire *et al.*, 2002), the function χ can be defined through a smooth function η by

$\chi(x) = \text{bool}(\eta(x) > 0)$ and in the algorithm we can work with a smooth η as in the level-set methods.

[11] generally leads to weaker results because if $\chi^n \rightarrow \chi$, χ may not be a characteristic function ; this leads to a *relaxed problem*, namely [11] with

$$X_d = \{\chi : 0 \leq \chi(x) \leq 1\} \text{ instead of } \tilde{X}_d = \{\chi : \chi(x) = 0 \text{ or } 1\} \quad [12]$$

These relaxed problems usually have a solution and it is often possible to show that if the solution is not in \tilde{X}_d then it is the limit of *composite domains* made of mixtures of smaller and smaller subdomains and holes (Tartar, 1997).

In 2D and for Dirichlet problems like [5] there is a very elegant result due to Sverak (Sverak, 1993) which shows that either there is no solution because the minimizing sequences converge to a composite domain or there is a regular solution ; more precisely: *if a maximum number of connected components for the complement of Ω is imposed as an inequality constraint for the set of admissible domains then the solution exists.*

This approach is powerful to answer *topological* questions which are more embarrassing for the formulations [5] and [9]. One difficulty with this approach is regularity control. In practice, one likes to couple topology parameterization and level sets to have both implicit parameterization and variable topology even though, as we saw in Figures 1 and 2, the level set approach has the capacity of handling topology changes if one reduces the required regularity to create dislocation and then recover the regularity once a new topology is found for the shape.

3.2.5. Regularized formulations

One may insure well-posedness *regularizing* the problem by changing the criterion and adding a cost to the control. This can be done with any of the parameterization above (S , T or χ). Problem [5] becomes

$$J(\Omega) = \int_D (\psi - \psi_d)^2 + \epsilon \int_S dx$$

insures existence. More generally, one may consider working with

$$J(\Omega) = \int_D (\psi - \psi_d)^2 + \epsilon \|S\|^2$$

but the choice of norm is delicate. In general for second order problems anything related to the second derivatives (*i.e.* radius of curvature) would be likely to work, but it is not known if weaker norms would work also.

4. Sensitivity analysis

We gave in introduction several reasons why sensitivity evaluation is important in minimization. Differentiability can be introduced for each of the formulations above.

For [5] it is done by using normal variations around a reference shape (see [8]. If Gateaux differentiability in $L^2(S)$ can be established, there exists $\chi \in L^2(S)$ with

$$j(S(t\alpha)) - j(S) = t \int_S \chi \alpha + o(t)$$

Fréchet differentiability will hold if in addition $o(t)$ is $o(t|\alpha|_0)$, where $|\cdot|_0$ denotes the norm of $L^2(S)$. Then χ is the L^2 -gradient, denoted by $\text{grad}_\alpha j(S)$ and we have:

$$j(S(\alpha)) - j(S) = \int_S \text{grad}_\alpha j(S) \alpha ds + o(|\alpha|_0)$$

and so $\text{grad}_\alpha j(S)$ must be zero at the solution of [5] and $\bar{\alpha} = -\text{grad}_\alpha j(S)$ is a direction of descent in the sense that if S is not the solution, $j(S(\rho\bar{\alpha})) < j(S)$ for a small enough positive constant ρ .

Following (Cea, 1986) and (Delfour *et al.*, 2001) one may consider a velocity of deformation $V(x)$ and define a time dependent shape

$$\Omega(t) = \{x + V(x)t : x \in \Omega\}$$

and compute $\frac{dJ}{dt}$, known as the *material derivative* of J .

The concept of topological derivative (Sokolowski *et al.*, 1991; Garreau *et al.*, 2001) permits to introduce naturally changes in topology. One digs a small circular hole of center x and radius ϵ in the domain and study the limit of $\frac{1}{\epsilon^m}(\psi^\epsilon - \psi)$, for the right power m , where ψ^ϵ is the solution of the PDE with the hole and ψ the solution without the whole.

There are serious difficulties of analysis with the Calculus of Variations when the solution of the partial differential equation has a discontinuity (Godlewski *et al.*, 1998). One shows that the adjoint state v has no shock because its time boundary condition is continuous and the characteristics integrated backward never cross the shock. This fact was observed by Giles (Giles *et al.*, 2001) in the more general context of the Euler equations for perfect gas. Hence, one shows that as long as the functional does not involve the position of the shock or its amplitude explicitly a blind calculus of variation is valid (Bardos *et al.*, 2002) and that there is no need to include in a calculus of sensitivity the variations of the shock position explicitly.

Algorithmic differentiation

Often the equivalent continuous equation behind a computer code is unknown or the discretization is inconsistent. One should therefore consider the discretized form (or more simply the computer program) of the equation. Automatic Differentiation (AD) is a natural way to access the derivative of discretized functions. We recall shortly the basis of the theory of AD

Let, f be a composed function given by:

$$x \in R^p \rightarrow y = h(x) \in R^n \rightarrow z = g(y) \in R^n \rightarrow u = f(z) \in R^q \quad [13]$$

Two differentiation procedures are available in automatic differentiation called the forward and the reverse modes. The forward mode consists of computing the function and its derivative at the same time using chain rule. On our example this gives:

$$u' = f'(z)g'(y)h'(x), \quad [14]$$

where $f' \in R^{q \times n}$, $g' \in R^{n \times n}$, $h' \in R^{n \times p}$. We observe that in [14] we need to introduce an intermediate matrix $M = g'(y)h'(x) \in R^{p \times n}$ to store the intermediate result before making $u' = f'(z)M$.

This can be seen as the propagation of the information on the function and its derivatives through the program. This is therefore a natural way to analyze error propagation in a solver.

The backward or reverse mode is analogous to the classical adjoint method used in optimization problems. The most important advantage of this approach over the previous one is a complexity independent of the number of control parameters.

After transposition of [14] we have:

$$u'^T = h'^T(x)g'^T(y)f'^T(z) \quad [15]$$

We can see that the storage now is $M = g'^T(y)f'^T(z) \in R^{n \times q}$. Hence, following the dimensions of the different spaces (*i.e.* p and q), we should use formula [14] or [15] to optimize the required memory. For instance, for optimization applications where p is the number of control variables and $q = 1$ with f being a cost function, the differentiation after transposition is more suitable.

In a computer program, in the direct mode, the Jacobian is produced by differentiating the program (considered as composed function) line by line, by a computer program, producing a new code for the gradient. Except if an object oriented language is used in which case this is done at no extra programming effort overloading all the operators in the native libraries (Mohammadi *et al.*, 2001; Danaila *et al.*, 2004). The reverse mode corresponds to writing the direct code in the reverse order (inverting loops as well) and differentiating each line $y = y + f(x)$ using the rule: $p_x = p_x + f'(x)p_y$ initializing all parameters p_x to zero and p_y to one (this corresponds to the last dependent variable in a program) (Griewank, 2001; Gilbert *et al.*, 1991; Mohammadi *et al.*, 2001).

One should also remark that in [13] the order of calculation of the variables is x, y, z, u while in the reverse mode one needs variables available in the reverse order u, z, y, x (see [15]). This is called intermediate stages storage and is one major limitation of reverse or adjoint calculations (even in the continuous level). This is in particular limiting with PDE based solvers using nested loops (e.g. a time iteration loop and inside loops over nodes and elements) even though check-pointing technic brings some relief (Griewank, 1995) giving maximal bounds for the combination of storage versus redundant calculations.

5. Numerical examples

In (Mohammadi *et al.*, 2001) we showed several examples of flow control using incomplete sensitivities with the control varying in time and space and expressed for instance through injection/suction or piezo-electric devices. Here, we consider an active control problem where the control does not depend on time despite the state and the cost functional are time dependent. This is the case in shape optimization for unsteady flows. Building sub-optimal solution is more difficult in this situation.

5.1. A model problem for noise control

Consider the following minimization problem:

$$\min_{x \in \mathcal{O}_{ad}} J(x(y), y, t, u(x(y), y, t)), \quad [16]$$

The state equation is a modified Burgers equation

$$\begin{aligned} u_t + 0.5(u^2)_y - \nu u_{yy} &= (\mu y + g(y, t) + x(y))u, \quad \text{on } \mathcal{D} =]-1, 1[\quad [17] \\ u(t, -1) &= u_l, \quad u(t, 1) = u_r, \quad u(t = 0, y) = u_0(y) \end{aligned}$$

Here the control is a function $x(y)$ in space. Our aim is to reduce state fluctuations in time. Of course, the unsteadiness can be completely removed only if $x = g(y, t)$, but then the control would depend on time. We consider the following configuration: $\mu = 0.3, \nu = 10^{-4}, g(y, t) = 0.5 \sin(10\pi t) \cos(6\pi t)y$. The equation is discretized on a uniform mesh with 101 nodes with a classical explicit RK3 scheme in time and a consistent numerical viscosity is used for stabilization. The admissible control space is $\mathcal{O}_{ad} = [-1, 1]^{101}$.

We consider the minimization of the following functional

$$J(x(y)) = \int_{t>0} j(t)dt, \quad j(t) = \int_{\mathcal{D}} (u_t^2 + u_{tt}^2)dy. \quad [18]$$

Figure 5 shows that a control obtained by minimizing (18) is efficient reducing the fluctuations over all modes. Here the gradient is computed using the reverse mode of AD. Approximation of gradient is based on partial reverse integration (typically over 1/20 of the total time).

5.2. Shape optimization for noise reduction

We consider now the problem of shape optimization for unsteady flows. The formulation of the problem is as in (16) but the state is now solution of the Navier-Stokes equations. The control S is the CAD-Free parameterization of the shape with regularity control.

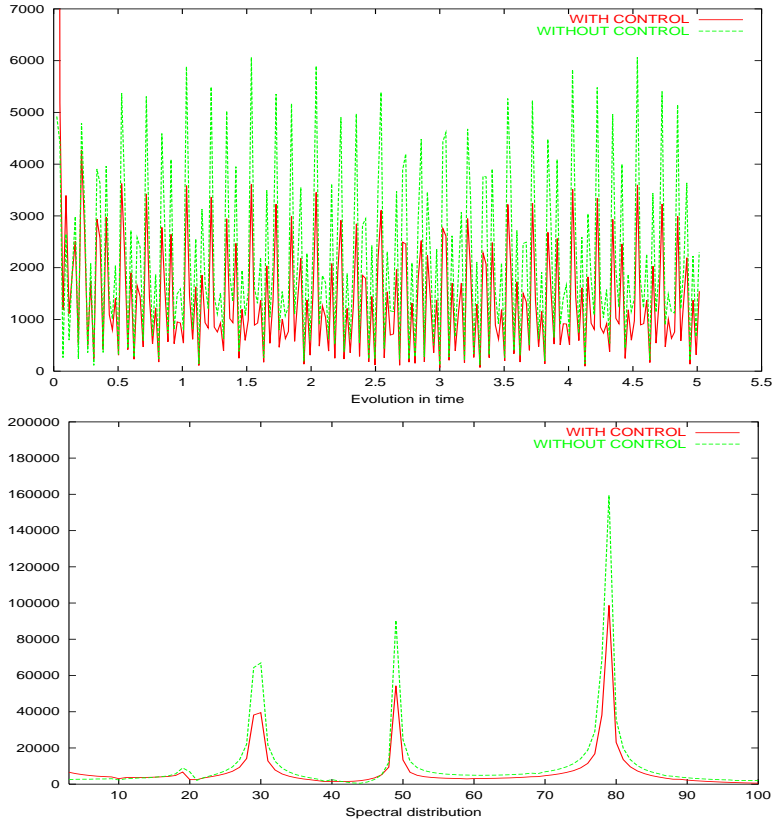


Figure 5. Model problem for noise reduction. Evolution of $j(t)$ (upper) and its spectral representation on a given time window for the uncontrolled and controlled simulations of [17] minimizing [18]

For compact sources, the radiated noise is linked to lift and drag time fluctuations (Marsden *et al.*, 2001). The aim is therefore to remove these fluctuations. We consider

$$J(x) = (\max(C_l(t)) - \min(C_l(t))) + (\max(C_d(t)) - \min(C_d(t))) \quad [19]$$

for $t \in [0, T]$ where T is a given observation time window.

We briefly recall the incomplete sensitivity approach used to define the search direction. Consider a general simulation loop linking the shape parameter S to a functional J :

$$J(S) : S \rightarrow q(S) \rightarrow U(q(S)) \rightarrow J(S, q(S), U(q(S))), \quad [20]$$

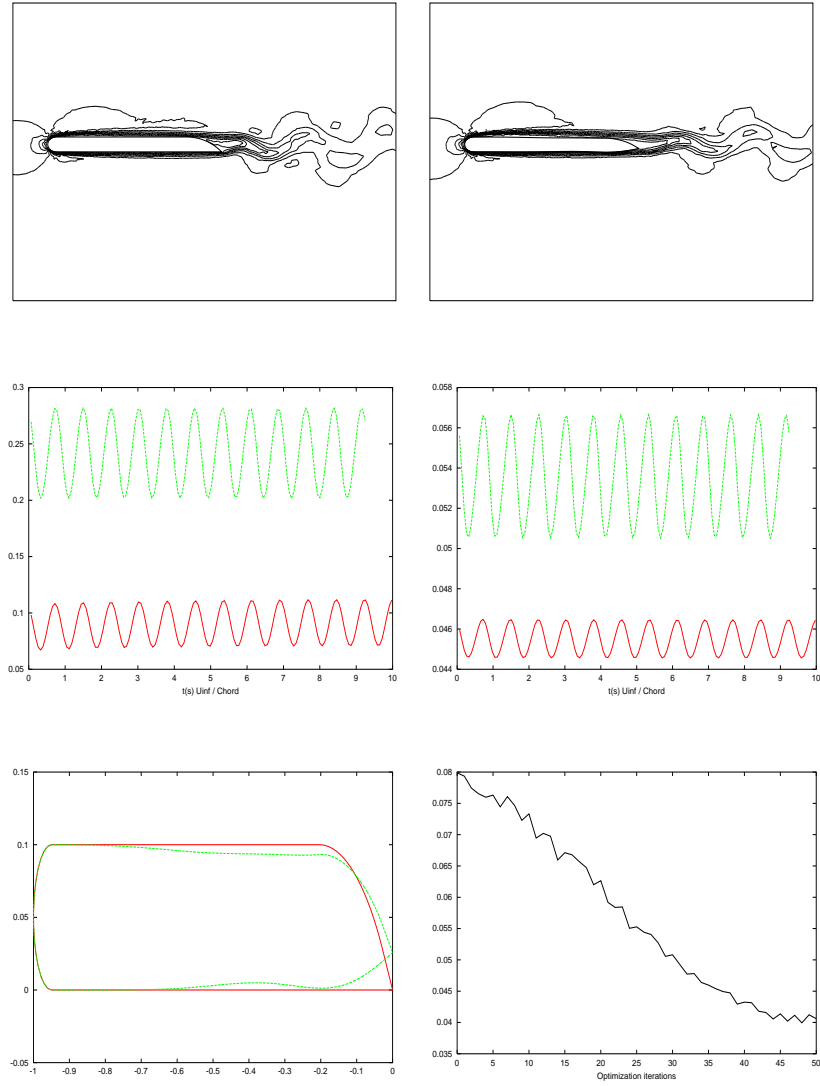


Figure 6. Shape optimization for noise reduction. Upper: Iso- $|u|$ for the initial and optimized shapes. The fluctuations have been reduced. Middle: lift and drag evolution in time for the initial and optimized shapes. Lower: Initial and final shapes and evolution of [19]

where q represents all geometrical entities and U all state related variables. The gradient of J with respect to S is:

$$J' = J_{,S} + J_{,q}q_{,S} + J_{,U}U_{,S} \quad [21]$$

The major part of the cost of this evaluation is due to $\partial U/\partial S$ in the last term.

Consider the following context for shape optimization:

- 1. both the cost function and control space are defined on the shape (or on some part of it),
- 2. J is of the form

$$J(S) = \int_{\text{Shape}} f(S)g(U)ds,$$

- 3. the local curvature of the shape is not too large (this needs to be quantified, for a wing typically we consider regions away from leading and trailing edges).

If these requirements hold, we can use an incomplete evaluation of this gradient, neglecting the sensitivity with respect to the state in (21). In the context of a time dependent phenomenon, incomplete sensitivity means also an instantaneous incomplete gradient.

The functional (19) is therefore not suitable for incomplete sensitivity evaluation as it involves time dependency. The functional we consider for incomplete sensitivity evaluation is given by

$$J(x) = |C_l(p_t)| + |C_d(p_t)| + |C_d(p_{tt})| + |C_l(p_{tt})| \quad [22]$$

where C_l and C_d are the pressure terms in aerodynamic lift and drag coefficients evaluated for the first and second time derivatives of the pressure.

Figure 6 shows iso-velocity contours around the initial and optimized shapes, the evolution of the lift and drag coefficients in time and the initial and final shapes obtained.

6. Conclusions

Optimal shape design is still a difficult and computer intensive task. Even if the problem is well posed and the sensitivity are computed correctly, success is not guaranteed. One should pay attention to the computing complexity and use sub-optimal approaches whenever possible to avoid computing an adjoint state. As local minima are present for these multi-criteria problems, global optimization approaches using a mix of deterministic and nondeterministic together with surface response model reduction is necessary to break complexity. Care should also be taken when noise is present in data and always consider robustness issues, at least *a posteriori*.

Acknowledgements

The authors would like to thank Dr. L. Hascoet and the INRIA Tapenade team for having made available to us Tapenade AD tool. Part of this work has been performed through ANR-COPTER (05-BLAN-0005) project.

7. References

- Allaire G., Jouve F., Toader A., « A level-set method for shape optimization », *CRAS*, vol. 334, n° 1, p. 1-8, 2002.
- Bardos C., Pironneau O., « A formalism for the differentiation of conservation laws », *CRAS*, vol. 453, n° 1, p. 1-8, 2002.
- Cea J., *Conception optimale ou identification de forme : calcul rapide de la dérivée directionnelle de la fonction coût*, Modélisation Math. Anal.-AFCET-Dunod, Paris, 1986.
- Chenais D., « Shape optimization in shell theory », *Eng. Optimization*, vol. 1, n° 11, p. 289-303, 1987.
- Danaïla I., Hecht F., Pironneau O., *Simulation numérique en C++*, Dunod, Paris, 2004.
- Delfour M., Zolezio J.-P., *Surface derivatives*, SIAM Series, New York, 2001.
- Dervieux A., Thomasset F., « A Finite element method for the simulation of Rayleigh-Taylor instability », *Lecture Notes in Mathematics*, vol. 1, n° 771, p. 145-159, 1979.
- Dervieux A., Thomasset F., « Multifluid incompressible flows by a finite element method », *Lecture Notes in Physics*, vol. 1, n° 11, p. 158-163, 1981.
- Garreau S., Guillaume P., Masmoudi M., « The Topological Asymptotic for PDE Systems : The Elasticity », *SIAM J. Control Optimization*, vol. 39, n° 6, p. 1756-1778, 2001.
- Gilbert J., Vey G. L., Masse J., « La différentiation automatique de fonctions représentées par des programmes », *INRIA*, vol. 1, n° 11, p. 1-45, 1991.
- Giles M., Pierce N., « Analytic adjoint solutions for the quasi-one-dimensional euler equations », *Journal of Fluid Mechanics*, vol. 3, n° 426, p. 327-345, 2001.
- Godlewski E., Olazabal M., Raviart P.-A., « On the linearization of hyperbolic systems of conservation laws. Application to stability », *Equations aux dérivées partielles et applications*, vol. 1, n° 10, p. 549-570, 1998.
- Griewank A., « Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation », *Optimization Methods and Software*, vol. 1, n° 12, p. 35-54, 1995.
- Griewank A., *Computational derivatives*, Springer, New York, 2001.
- Ivorra B., Hertzog D., B. Mohammadi J. S., « Global Optimization for the Design of Fast Microfluidic Protein Folding Devices », *Int. J. Num. Meth. Eng.*, vol. 26, n° 6, p. 2220-2226, 2006.
- Marsden A., Wang M., Mohammadi B., *Shape optimization for noise control*, CTR Briefs, Stanford, 2001.
- Mohammadi B., « Global optimization, level set dynamics, incomplete sensitivity and regularity control », *IJCFD*, vol. 21, n° 2, p. 10-45, 2007.

- Mohammadi B., Pironneau O., *Applied Shape Optimization for Fluids*, Oxford U. Press, London, 2001.
- Mohammadi B., Pironneau O., « Shape Optimization in Fluid Mechanics », *Annual Revue of Fluid Mechanics*, vol. 36, n° 1, p. 255-279, 2004.
- Mohammadi B., Saiaç J., *Pratique de la simulation numérique*, Dunod, Paris, 2003.
- Murat F., Simon J., *Etude de problèmes d'optimum design*, Lecture Notes in Computer sciences, Paris, 1976.
- Neittaanmaki P., « Computer aided optimal structural design », *Surv. Math. Ind.*, vol. 1, n° 10, p. 173-215, 1991.
- Osher S., Sethian J., « Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations », *Journal of Computational Physics*, vol. 79, n° 1, p. 12-49, 1988.
- Pironneau O., *Optimal shape design for elliptic systems*, Springer, New York, 1984.
- Sokolowski J., Zolezio J.-P., *Introduction to shape optimization*, Springer Series in Computational Mathematics, Paris, 1991.
- Su J., Renaud J., « Automatic Differentiation in robust optimization », *AIAA Journal*, vol. 35, n° 6, p. 1072-1079, 1997.
- Sverak V., « On optimal design », *J. Maths. Pures et Appl.*, vol. 1, n° 72, p. 537-551, 1993.
- Tartar L., *Control problems in the coefficients of PDE*, Springer - Lecture notes in Economics and Math Systems, New York, 1974.
- Tartar L., *On the control of coefficients in partial differential equations*, Birkhauser, Boston, 1997.