Applied Optimal Shape Design

B. Mohammadi
O. Pironneau Universities of Montpellier II and Paris VI
Institut Universitaire de France
Analyse Numérique, 175 rue du Chevaleret, F-75013
Olivier.Pironneau@math.jussieu.fr

April 8, 2002

Abstract

This paper is a short survey of Optimal Shape Design (OSD) for fluids. OSD is an interesting field both mathematically and for industrial applications. Existence, Sensitivity, correct Discretization are important theoretical issues. Practical implementiation issues for airplane designs are critical too. The paper is also a summary of the material covered in our recent book[7]

keywords: Optimization, Shape design, Finite Elements, mesh adaptation, aerodynamics

1 Industrial Demand

Because control is a natural desire once the simulation is completed, the applications of OSD are uncountable. For instance the design of a harbour which minimizes the waves coming from far can be done at little cost by standard optimization methods once the numerical simulation of Helmholtz equation is mastered (see Baron[1]) as shown on figure 1 Other applications include

- Weight reduction in car engine, aircraft structures, etc
- Electromagnetically optimal shapes, such as in stealth airplanes
- Wave canceling fore bulbe in boat design
- Drag reduction for airplanes,cars and boats

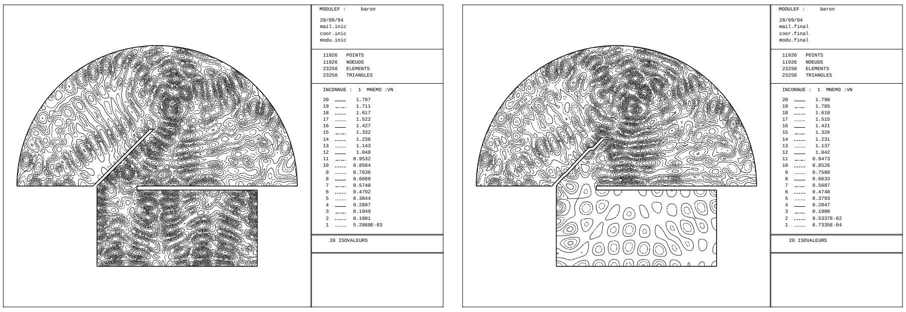

Figure 1: Optimization of a brake-water (Computed by A.Baron).

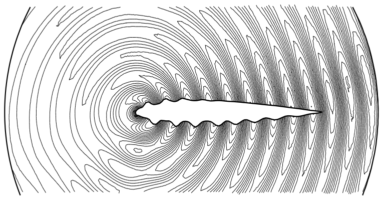

Figure 2: Optimal design of an airfoil to minimize in a sector (angle between 180 degrees to 225) the reflection of a monochromatic incident radar wave. The optimal shape is not admissible from the aerodynamic view point; a multi-disciplinary optimization is necessary (Computed by A.Baron).

However there are no automatic solutions to these problems because most of the time engineering design is made of compromises due to the multi-disciplinary aspects of the problems (see figure 2, the necessity of doing *multi-point constrained design* and because the solvers are not always made in house and appear as *black-box* solvers.

1.1 An Example in 1D

To understand the difficulties of OSD problem let us consider the problem of design a string of thickness α and length s for a musical instrument which gives a response as close to ψ_d as possible:

Find $s > b$ and $\alpha > 0$ such that

$$\min_{\mathbf{s}\in\mathcal{S}_d}\{\int_a^b |\psi-\psi_d|^2 \ : \ \psi + \frac{d}{dx}(\alpha\frac{d\psi}{dx}) = f \quad \text{in } (0,s) \quad \psi(0)=0 \;\; \psi(s)=\psi_d(s)\}$$

To discretize the problem, call $\psi_i \equiv \psi(x_i), \;\; \delta x_{i+1/2} = x_{i+1} - x_i, \;\; x_I = s$

and consider

$$\min_{\delta \mathbf{x_i} \in \mathcal{DS}_d} \{ \sum_{i\in[i_a,i_b]} |\psi_i - \psi_{d_i}|^2 \delta x_i \ : \ \psi_i + \frac{1}{\delta x_i}[\alpha_{i+1/2}\frac{\psi_{i+1}-\psi_i}{\delta x_{i+1/2}} - \alpha_{i-1/2}\frac{\psi_i - \psi_{i-1}}{\delta x_{i-1/2}}] = f_i \}$$

with $\psi_0 = 0 \ \ \psi_I = \psi_{d_I}$. i.e.

$$\min_{\vec{\delta}x \in D} \{ \Psi^T B(\vec{\delta}x)\Psi \ : \ A(\vec{\alpha}, \vec{\delta}x)\Psi = F \}$$

It is seen here that the problem is akin to control in the coefficient of PDEs, that the discrete problem has a greater unknown space than the continuous problem because the mesh comes also as a degree of freedom.

Nevertheless the problem is differentiable and so gradient methods should work. For this we will need the derivatives of the cost function with respect to all the unknowns, s, α,δx_i. This can be a momentous task and so whenever possible Automatic Differentiation is of great help.

2 Principle of Automatic Differentiation

Consider the problem of finding $J'(u)$ when $j(u)$ is given by a computer program. Because the program is made of differentiable lines, J' can be computed by differentiating every line and adding them to the computer program immediately above each line. For instance

Program for J	.	**lines to add to the program**
$x = 2u(u+1)$		$dx = 2*u*du + 2*du*(u+1)$
$y = x + \sin(u)$		$dy = dx + \cos(u)*du$
$J = x*y$		$dJ = dx*y + x*dy$

If this new program is run with `u=u0, du=1, dx=0, dy=0, dJ=0`, then dJ is the derivative of J with respect to u at $u0$.

2.1 Automatic AD

However differentiating each line can be long and tedious. It can be done by the compiler by overloading the operators of arithmetics and the functions in the standard C-library. Operator overloading is available in C++ and so we have the following procedures:

- Step 1: if the program is in FORTRAN use f2C from http://www.netlib.org/f2c/

- Step 2: change float and double into `ddouble` and add `#include ddouble.h`

2.2 The library ddouble

Each variable has now two field: its value and the value of its derivative. So we define a class of differentiable variables and stores these values in `v`. Every time an arithmetic operation is done, the corresponding operation on the derivatives must be done too. For instance below we give the overloading of the multiplication and of the addition:

```
class ddouble{public:
  float v[2];
  ddouble(double a, double b=0)
       { v[0] = a; v[1]=b;}
       // intialise la derivee a 0 sauf si b!=0
  friend ddouble operator *
              (const ddouble& a, const ddouble& b)
          { ddouble c;
            c.v[1] = a.v[1] * b.v[0] + a.v[0] * b.v[1];
       // (fg)'=f'g+fg'
            c.v[0] = a.v[0] * b.v[0];
            return c;
          }
  friend ddouble operator +
       (const ddouble& a, const ddouble& b)
          { ddouble c;
            c.v[1] = a.v[1] + b.v[1];   // (f+g)'=f'+g'
            c.v[0] = a.v[0] + b.v[0];
            return c;
          }
 // ...
 };
```

3 Well Posedness

Consider the academic problem of designing a wind tunnel with required flow properties in a region of space D. (see figure 3). With a stream function formulation this would be

$$\min_{\mathbf{S}\in\mathcal{S}_d}\{\int_D |\psi-\psi_d|^2 \ : \quad -\Delta\psi=0, \quad \psi|_{\mathbf{S}}=0 \ \ \psi|_{\partial C}=\psi_d\} \tag{1}$$

Before attempting any numerical simulation we may study the existence of solutions. These are by no means non-practical questions because many of these optimal shape design problems don't have solutions. For example the optimization of a hook, clamped to

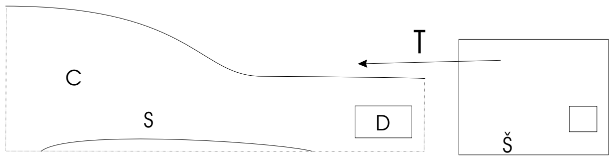

Figure 3: Inverse design for a wind tunnel with desired properties ψ_d in D

the wall on the left and pulled by so weight on the right. With respect to weight under a given max constraints so that the structure does not break: optimal structures are of composite materials; there is no simply connected solution to this problem.
Although existence can be studied directly by using continuity results with respect to domain boundaries, one may also map the unknown domain from a fixed domain and consider that the unknown is now $T : C \to \Omega$

$$\min_{\mathbf{T}\in\mathcal{T}_d}\{\int_{\hat{D}} |\psi-\psi_d|^2 \ : \ \nabla\cdot[A\nabla\psi]=0 \ \text{ in } C \quad \psi|_{\partial C}=\psi_d, \ \ A={T'^{-1}}^{T}\,T'^{-1}\text{det}T'\} \quad (2)$$

Or extend the operators by zero in S and take the characteristic function for unknown:

$$\min_{\chi\in X_d}\{\int_D |\psi-\psi_d|^2 \ : \ -\nabla\cdot[\chi\nabla\psi]=0, \quad \psi(1-\chi)=0 \ \ \psi|_{\partial C}=\psi_d\} \qquad (3)$$

This last approach, suggested by L. Tartar[14] has lead to *topological optimization*.

3.1 Results

Most results are obtained by considering minimizing sequence S^n and (in the case of our academic example) show that $\psi^n \to \psi$ for some ψ, which is the solution of the PDE.

By using directly regularity with respect to the domain, D. Chenais[3] (see also Neittanmaki[9]) showed that in the class of all S uniformly Lipschitz, problem (1) has a solution.

Similarly Murat-Simon[8] working with (2) showed that in the class of $T \in W^{1,\infty}$ uniformely, the solution exists.

However working with (3) generally leads to weaker results because if $\chi^n \to \chi$, χ may not be a characteristic function; one is lead to a *relaxed problem*.

In 2D and for the Dirichlet problem there is a very elegant result due to Sverak[13]: *if a maximum number connected components is imposed then the solution exists*.

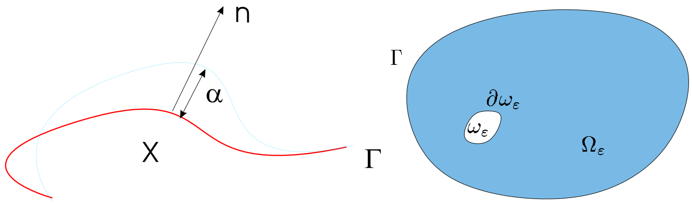

Figure 4: Normal variations on a reference shape (left). Topological variation on the shape (right)

3.2 Well Posedness by Regularization

Another way to insure well posedness is to *regularize* the problem by changing the criteria and adding a “cost” to the control. For problem (1)

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

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 a delicate one. In general for second order problem anything related to the second derivatives would be likely to work, but it is not know if weaker norms would work too.

4 Sensitivity Analysis

Even though AD can solve the problem it is wise to check differentiability analytically. This can be done by using normal variation on a reference shape (see figure 4)

$$\partial\Omega^\alpha = \{x + \alpha\mathbf{n} \ : \ x \in \partial\Omega\} \tag{4}$$

Following Cea[2], Delfour-Zolezio[4] one may consider a velocity of deformation $V(x)$ and define a time dependant shape

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

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

Lately, for Neumann problems, the concept of topological derivative was introduced by Sokolowski[12]. One digs a small circular hole of center $x - 0$ in the domain and study the limit of $\frac{1}{\epsilon}(\psi^\epsilon - \psi)$ where ψ^ϵ is the solution of the PDE with the hole and ψ the solution without the whole (see figure 4).

4.1 Sensitivity: Example

For the Laplace equation with Dirichlet conditions

$$-\Delta\psi^{\epsilon\alpha} = f \quad \text{in } \Omega^{\epsilon\alpha} \quad \psi^{\epsilon\alpha} = 0 \text{ on } \Gamma^{\epsilon\alpha} = \{x + \epsilon\alpha\mathbf{n} \ : \ x \in \Gamma\}$$

where $\Omega^{\epsilon\alpha}$ is obtained by (4) the derivative with respect to α is calculated by assuming

$$\psi^{\epsilon\alpha} = \psi + \epsilon\psi'_\alpha + \frac{\epsilon^2}{2}\psi''_\alpha$$

By linearity ψ' and ψ'' satisfy the PDE with zero rhs. By Taylor expansion:

$$0 = \psi^{\epsilon\alpha}(x + \epsilon\alpha n) = \psi^{\epsilon\alpha}(x) + \epsilon\alpha\frac{\partial\psi^{\epsilon\alpha}}{\partial n}(x) + \frac{\epsilon^2\alpha^2}{2}\frac{\partial^2\psi}{\partial n^2}(x) + ...$$

Therefore

$$-\Delta\psi'_\alpha = -\Delta\psi''_\alpha = 0 \qquad \psi'_\alpha|_\Gamma = -\alpha\frac{\partial\psi}{\partial n} \qquad \psi''_\alpha|_\Gamma = -\alpha\frac{\partial\psi'_\alpha}{\partial n} - \frac{\alpha^2}{2}\frac{\partial^2\psi}{\partial n^2}$$

4.2 Navier-Stokes Equations

4.2.1 The minimum drag problem.

$$E(\Omega) \equiv \min_{\Omega\in C}\int_\Omega \frac{1}{2}||\nabla u||^2 dx \ : \qquad u|_{\partial\Omega} = g$$
$$u\nabla u + \nabla p - \nu\Delta u = 0, \quad \nabla\cdot u = 0,$$

Sensitivity Analysis by local variations gives

$$\partial\Omega' = \{x + \alpha n(x) \in \partial\Omega\} \Rightarrow \ \delta E = \int_{\partial\Omega}\chi\alpha ds + o(|\alpha|)$$

$$\delta E = \int_\Omega \nabla u\cdot\nabla\delta u + \frac{1}{2}\int_{\partial\Omega}\alpha|\nabla u|^2 = \frac{1}{2}\int_{\partial\Omega}\alpha\partial_n u\cdot(\partial_n u + 2\underline{\partial_n w}) + o(|\alpha|)$$
$$-\mathbf{u}\nabla\mathbf{w} + \mathbf{w}\nabla\mathbf{u}^\mathbf{T} + \nabla\mathbf{q} - \nu\mathbf{\Delta w} = \mathbf{u}\nabla\mathbf{u}, \quad \nabla\cdot\mathbf{w} = \mathbf{0}, \quad \mathbf{w}|_{\partial\mathbf{\Omega}} = \mathbf{0}$$

More can be found in [10][11]

4.3 Gradient Methods

So one starts with a smooth shape, moves each point in its normal direction by

$$\alpha = \partial_n u\cdot(\partial_n u + 2\partial_n w).$$

But will the new shape have the same regularity? In general the answer is no, and this loss of regularity is numerically dangerous and prone to generation of oscillations. The

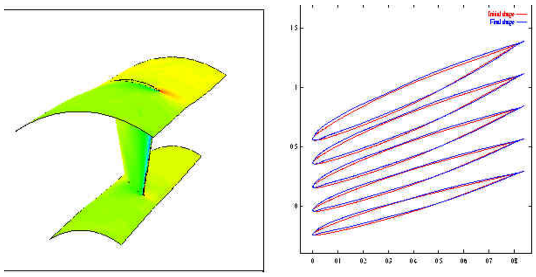

Figure 5: Optimization of a cooling fan for a car engine. This 3D optimization improved the design by 10%. The picture displays the final shape and the change at some cross section from the original hand optimized original design.

cure is to use a *smother* which mathematically amounts to change the norm of the gradient method. For instance if the shape is moved by β solution of

$$-\frac{d^2\beta}{ds^2} = -\alpha \quad \Rightarrow$$

then

$$j(S(\beta)) - j(S) = \frac{1}{2}\int_{\partial\Omega} \beta\partial_n u \cdot (\partial_n u + 2\partial_n w)$$
$$= -\frac{1}{2}\int_{\partial\Omega} \beta\alpha = -\int_{\partial\Omega} |\frac{d\alpha}{ds}|^2$$

4.4 Discretization

Consider again the academic problem (1). It can be discretized by

$$\min_{q_h}\{j_h(q_h) = \int_D |\psi - \psi_d|^2 \ : \ \int_\Omega \nabla\psi_h \nabla w_h + \frac{1}{\epsilon}\int_\Gamma (\psi_h - \psi_d) w_h = 0 \ \ \forall w_h \in V_h\}$$

where V_h is the finite element space of piecewise linear continuous functions.
Calculus of variation is possible but the degree of freedoms are now the node motion $q_h \in V_h$, a piecewise linear continuous function built from its values at the vertices, namely the motions of the same (see figure 4.4.. Let

$$\int_\Omega \nabla p_h \nabla w_h + \frac{1}{\epsilon}\int_\Gamma p_h w_h = 2\int_D (\psi_h - \psi_d) w_h \ \ \forall w_h \in V_h$$

Finally

$$\delta J_h = \int_\Omega \nabla\psi_h (\nabla\delta q_h + \nabla\delta q_h - \nabla\cdot\delta q_h)^T \nabla p_h \approx \int_\Gamma \nabla\psi_h \nabla p_h \delta q_h \cdot n$$

Recognize here the linearization of $\nabla Q_h \nabla Q_h^T \mathrm{det}\nabla Q_h$ at $x + q_h(x)$, therefore

Corollary 1 *The change of cost function due to inner nodes is* $O(h^2)$

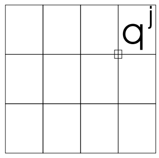

Figure 6: After discretization, not only the boundary vertices but also the inner vertices are the degrees of freedom of the optimization problem

4.5 Guidelines

To our experience success in solving an OSD problem depends on the following.

- Whenever possible second order optimization methods (BFGS for instance) should be used because the problems are stiff.
- Compute derivatives with respect to boundary nodes only and apply the theory of approximate gradients to combine mesh refinement with optimization (see [5] for example).
- Use a smoother, i.e. don't work with the $L^2(\Gamma)$ norm of the node displacements. Use it also to move the inner vertices.
- Experience also shows that

  $$\delta_S j(\Omega) = \delta_S \int_S F(\psi) \cdot n \quad \approx \int_S F(\psi) \cdot \delta n + ...$$

  Namely the most important variation is due to the change of the normal in the case of surface integrals.

5 Implementation Issues and Results

5.1 Link with CAD

In industries shapes are stored in Computed Aided Design data bases as a set of Bezier patches or others with infinite details such as screws and bolts which are not relevant to a finite element calculation in aerodynamics for instance. Furthermore the CAD system is proprietary.

Therefore it is convenient to abstract the optimization from the CAD system and ask the engineer for any triangulation of the surface and use it as our initial design. The strategy is then what we call a *CAD-free optimization plateform*:

1. Generate any surface mesh from the CAD data
2. Apply a C^1 + edges recognition software for surface mesh refinement
3. Apply a 3d volumic automatic mesh generator from the surface mesh

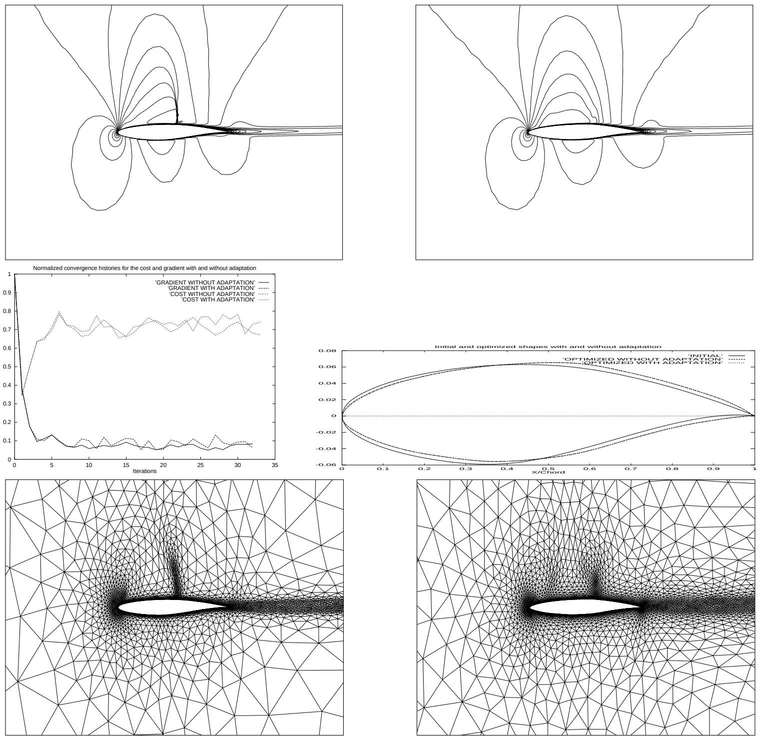

Figure 7: Optimization of a wing profile

4. Do the optimization with mesh refinement using the same module as in step 2 but coupled with the PDE solver
5. Feed back the result into the CAD system

5.2 Optimization of a wing profile

Drag is mostly pressure drag due to the shock (pressure drag). The lift & area are imposed by a penalty method with parameters ϵ, β.

$$J(u,p,\theta) = F \cdot u_\infty + \frac{1}{\epsilon}|F \times u_\infty - C_l|^2 + \frac{1}{\beta}(\int_S dx - a)^2$$

with $F = \int_S (p\mathbf{n} + (\mu\nabla u + \nabla u^T))$ and a Navier-Stokes + $k - \epsilon$ + wall laws flow solver.

6 Prospective

Recently the optimization of a complete aircraft (a business jet) was done with this method (even on a workstation when incomplete gradients are used) and a ten percent improvement

obtained after a few iterations.
However OSD is still a difficult and computer intensive task. There is a good prospect for global non differentiable optimization because there are often many local minima and because the flow solver is often available in binary format only (such would be the case if a commercial software was used). However Genetic Algorithms are still slow and difficult to couple with gradient methods.
Incomplete gradients is also a good field of research and awaits mathematical proofs.

References

[1] A. Baron. (1998). Optimisation multi-physique, Thesis, University of Paris 6.

[2] J. Cea. (1986). Conception optimale ou identification de forme: calcul rapide de la dérivée directionelle de la fonction cout, Modélisation Math Anal, AFCET, Dunod.

[3] D. Chenais. (1987). Shape optimization in shell theory, Eng. Opt. **11**, 289-303.

[4] M. Delfour, J.P. Zolezio: *Surface derivatives* SIAM series, 2001.

[5] G. Lemarchand, O. Pironneau, E. Polak: Incomplete gradients. Proc Domain Decomposition Method Conf, Lyon 2000. M Garbey ed. (to appear)

[6] J.L. Lions *Contrôle Optimal de systèmes gouvernés par des équations aux dérivées partielles.* Dunod-Gauthier Villars, 1968.

[7] B. Mohammadi, O. Pironneau: *Applied Optimal Shape Design* Oxford University Press, 2001.

[8] F. Murat and J. Simon. (1976). Etude de problèmes d'optimum design. Proc. 7th IFIP Conf. Lecture Notes in Computer sciences, **41**, 54-62.

[9] P. Neittaanmäki. (1991). Computer aided optimal structural design. Surv. Math. Ind. **1**. 173-215.

[10] O. Pironneau. (1973). On optimal shapes for Stokes flow, J. Fluid Mech. **70(2)**, 331-340.

[11] O. Pironneau. (1984). Optimal shape design for elliptic systems, Springer-Verlag.

[12] J. Sokolowski and J.P. Zolezio. (1991), Introduction to shape optimization Springer Series in Computational Mathematics.

[13] A. Sverak. (1992). On existence of solution for a class of optimal shape design problems, Note CRAS.

[14] L. Tartar. (1974). Control problems in the coefficients of PDE, Lecture notes in Economics and Math Systems. A. Bensoussan, Springer Verlag.