MULTILEVEL FUNCTIONAL PRECONDITIONING FOR SHAPE OPTIMISATION

F. Courty, A. Dervieux

INRIA, Centre de Sophia-Antipolis,
BP 93, 2004 Route des Lucioles,
06902 Sophia-Antipolis Cedex, France

Francois.Courty@inria.fr, Alain.Dervieux@Inria.fr

web page: http://www-sop.inria.fr/tropics/Francois.Courty/index.html

Abstract

We study the application of a multi-level preconditioner to a practical optimal shape design problem. The preconditioner is based on the Bramble-Pasciak-Xu series. We extend it to the unstructured parametrization of 3D shapes by using the volume-agglomeration heuristics. The choice of the smoothing parameter is analysed from functional arguments. Application to the shape design for optimising aerodynamic and sonic boom performances of a wing is demonstrated.

Key words: multilevel, optimal shape design, partial differential equations, Computational Fluid Dynamics, finite element, unstructured mesh.

1 Introduction

During the last two decades, researchers have looked for efficient solvers for CFD problems. Now that efficient CFD solvers are available, researchers are focusing on solving Optimum Shape design problems for CFD. Many of them use gradient-based methods, which have gained efficiency thanks to the use of adjoint states. Adjoint states are easier to develop thanks to the progress of Automated Differentiation tools, see for example [1]. Two families of problems must be distinguished. In the first family, optimization is applied to find a small fixed number of design parameters such as CAD (Computer Aided Design) parameters. We are in this paper interested by the second family refered as "CAD-free" ([2]), in which the optimization is applied for finding a function, for example the implicit representation of an airfoil. When gradient-based methods are used for finding a function, they generally require preconditioners. Indeed, let us consider the application of gradient iteration in a space of functions. The functional analysis tells us that the continuous iterate $u^{n+1} = u^n - \rho g^n$ can be less regular than u^n when the gradient g^n involves k-th derivatives of u^n . In other words the iteration operator can be unbounded, with a *regularity loss* equal to k. The continuous iteration then diverges. In the discrete case, this situation translates into an amplification of high frequency error components, except when very small, mesh dependent, step lengths ρ are used.

Experience shows that this problem is very frequent in CAD-free Optimal Shape Design. In this paper, we start from an analysis of this, related to the lack of smoothness of the Hadamard derivative of a function with respect to the domain shape. In practice, many authors have observed that discrete gradients produce oscillatory shapes. To cure this, Reuther and Jameson propose in [3] a method of correction smoothing that solves a Laplace-Beltrami system set on the shape manifold. This approach is also used in [2]. In [4], Arian and Ta'asan analyse the Fourier symbol of the Hadamard-type functional Hessian. According to the degree of this Hessian, these authors propose to use either a Laplace-Beltrami operator when the regularity loss is 2 or a Neumann-Dirichlet pseudo-differential operator when regularity loss is 1. This latter method can be more expensive, since 3D systems are then solved.

In this paper, we propose to push further the idea of Arian and Ta'asan by using a family of preconditioners for which the regularity gain is a parameter that can be specified according to the problem to solve. For this, we consider the Bramble-Pasciak-Xu (BPX) additive multilevel preconditioner (see [5, 6, 7]). A version applicable to an arbitrary unstructured fine level can be built by applying the agglomeration principle of [8].

The agglomeration technique was first used for building multi-grid solution methods for CFD problems (see also [9]) and was extended later to other applications. A multilevel cycle of multiplicative type was proposed in [10] for optimisation problems. An agglomeration-based additive preconditioner for solving elliptic systems was proposed in [11]. One of our contributions here is to investigate how this preconditioner can be adapted to the solution of shape optimisation problems.

In a first experiment, we examine the introduction of this kind of preconditioner in a popular class of quasi Newton optimisation algorithms, namely the BFGS technique.

We end by applying the proposed preconditioner to a pre-industrial shape optimisation loop introduced in [12] and addressing the aerodynamical and sonic boom optimization of supersonic aircraft geometries.

Contents

2 Hadamard formula and functional preconditioning

2.1 The Dirichlet case

Large scale problems coming from Partial Differential Equations generally result in a poor conditioning that further degrades as the number of degrees of freedom is increased. To explain and solve this problem, we can either analyse directly the behavior of discrete eigenvalues when the number of unknowns is increased or analyse the continuous -functional- problem and the continuous version of the algorithm. We concentrate on the second way.

We are interested in minimizing a functional $j(\gamma)$ with respect to a "shape parameter" γ. If γ were an n-uple of real numbers, with n fixed and small, the problem of conditioning would not arise as in the case we address, for which γ is assumed to be a *function* to be discretized with a potentially unlimited number of nodes. More precisely, an initial geometry Ω_0 is equipped with a vector field $\vec{n}_0$ normal to its boundary. A family of domains Ω_γ, all included in a background domain $\mathcal{O}$ of R^d is parameterized by a displacement $\gamma \in$

$\mathcal{C}^{l+\alpha}(\partial\Omega_0)$ of the boundary in direction $\vec{n}_0$:

$$\begin{aligned} \Omega_\gamma &\subset \mathcal{O} \; , \; \forall \, \gamma \\ \partial\Omega_\gamma &= \{\vec{x} \, + \, \gamma(\vec{x})\vec{n}_0 \; , \vec{x} \, \in \, \partial\Omega_0\} \; . \end{aligned} \tag{1}$$

The state equation is the Poisson problem:

$$-\Delta \; z(\gamma) \;=\; f \;\text{ on }\; \Omega_\gamma \;\; ; \;\; z(\gamma) \;=\; 0 \;\text{ on }\; \partial\Omega_\gamma \; .$$

Let D be a subdomain of Ω_γ (inside Ω_γ for any γ admissible). The functional j to minimize is defined by:

$$j(\gamma) \;=\; \frac{1}{2}||z(\gamma) - z_{target}||^2_D \; .$$

We are interested in computing the derivative of the functional with respect to the shape parameter. We have to do it by a chain rule involving the state equation. A classical difficulty comes from the variable domain in the state equation. Early solution to this variation calculus dates back to Hadamard. We shall instead invoke some techniques initated by the interior variation of Garabedian, see [13], and extended later on in [14],[15], [16], [17],[18].

We introduce first a family of diffeomorphisms $(T_\gamma)_\gamma$, smooth with respect to γ, and such that:

$$T_\gamma \;\; maps \;\; \mathcal{O} \;\; on \;\; \mathcal{O},$$

$$T_\gamma \ \ maps \ \ \Omega_0 \ \ on \ \ \Omega_\gamma,$$

The problem can be stated in terms of a new state variable:

$$j(\gamma) \ = \ \frac{1}{2}||\tilde{z}(\gamma) \circ T_\gamma^{-1} - z_{target}||_D^2$$

$$\tilde{z}(\gamma) = z(\gamma) \circ T_\gamma \ .$$

The new state $\tilde{z}(\gamma)$ is a solution of a well-posed Dirichlet problem with coefficients and source depending on γ through T_γ and T_γ^{-1} but formulated in the domain D which does not vary with the control γ. Then $\tilde{z}(\gamma)$ depends implicitly on γ and under regularity assumptions on f and $\partial\Omega_0$, we can prove by applying the implicit function theorem that the mapping:

$$\tilde{z} : \quad \gamma \mapsto \tilde{z}(\gamma)$$

is continuously differentiable from $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$ in $\mathcal{C}^{l-1+\alpha}(\Omega_0)$.

It then interesting to introduce a linear extension operator from Ω_0 to $\mathcal{O}$:

$$\mathcal{P} : \mathcal{C}^k(\Omega_0) \rightarrow \mathcal{C}^k(\mathcal{O})$$

$$f \mapsto \mathcal{P}f$$

$$\mathcal{P}f|_{\Omega_0} = f \ .$$

Let:

$$\bar{z}(\gamma) = (\mathcal{P}\tilde{z}(\gamma)) \circ T_\gamma^{-1},$$

then, by chain rule and with a possible reduction of regularity l, we deduce that

$$\gamma \to \bar{z}(\gamma) \tag{2}$$

is continuously differentiable from $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$ in $\mathcal{C}^{l-1+\alpha}(\Omega_0)$. Now, by construction, $\bar{z}(\gamma)$ is an extension of $z(\gamma)$:

$$\bar{z}(\gamma)|_{\Omega_\gamma} = z(\gamma) \ . \tag{3}$$

We can exhibit the derivative of $\bar{z}(\gamma)$ with respect to γ on Ω_γ, in which it is the solution of the non-homogeneous Dirichlet problem:

$$\begin{aligned} -\Delta \frac{\partial \bar{z}}{\partial \gamma}(\gamma).\delta\gamma &= 0 \text{ in } \Omega_\gamma \\ \frac{\partial \bar{z}}{\partial \gamma}(\gamma).\delta\gamma &= \frac{\partial \bar{z}(\gamma)}{\partial n_\gamma}(n_\gamma, n_0)\delta\gamma \end{aligned} \tag{4}$$

where $\vec{n}_\gamma$ is the normal to $\partial\Omega_\gamma$. One interesting way to state the above result is to compute the *total* derivative of the following state variational residual, which is possible due to the derivability of an extension of $z(\gamma)$:

$$\begin{aligned} \Psi(\gamma, z; \phi_1, \phi_2) &= -\int_{\Omega_\gamma} (\Delta\ z\ -\ f)\ \phi_1\ dv\ \ +\ \int_{\partial\Omega_\gamma} z\ \phi_2\ d\sigma\ , \\ \Psi(\gamma, \bar{z}(\gamma); \phi_1, \phi_2) &= 0\ \forall\phi_1, \forall\phi_2\ . \end{aligned}$$

Indeed the derivative with respect to $\bar{z}$ produces the left-hand side of (4), and after injection of the state equation, the derivative with respect to γ reduces to:

$$\int_{\partial\Omega_\gamma} \frac{\partial \bar{z}(\gamma)}{\partial n_\gamma}(n_\gamma, n_0)\delta\gamma \ \phi_2 \ d\sigma \ .\square$$

By chain rule, j is also differentiable and its gradient is expressed as follows:

$$j'(\gamma).\delta\gamma \quad = \quad \int_{\partial\Omega_\gamma} \frac{\partial z(\gamma)}{\partial n_\gamma} \ \frac{\partial p(\gamma)}{\partial n_\gamma} \quad < \vec{n}_\gamma, \vec{n}_0 > \delta\gamma \ \ d\partial\Omega_\gamma$$

where $p(\gamma)$ the following adjoint state:

$$-\Delta \ p(\gamma) \ = \ z_{target} - z(\gamma) \ \ ; \ \ p(\gamma) \ = \ 0 \quad on \quad \partial\Omega_\gamma \ .$$

Taking the L^2 space as pivot space for a *continuous* gradient method would produce the following iteration (ρ is a positive step length):

$$\gamma^* \quad = \quad \gamma \quad - \quad \rho \ \ g_{L^2} \ \ \text{where} \ \ g_{L^2} \quad = \quad \frac{\partial z(\gamma)}{\partial n_\gamma} \ \frac{\partial p(\gamma)}{\partial n_\gamma} \quad < \vec{n}_\gamma, \vec{n}_0 > \ \ .$$

The mapping transforming γ into to γ^* and set on the $d-1$-dimensional manifold $\partial\Omega_0$ contains pseudo-differential ingredients, namely the solutions of Partial Differential Equations set on domain Ω_0. In order to evaluate its degree, we observe that, starting from a previous iterate γ belonging to $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$, we get a corrected γ^* that is only of regularity $\mathcal{C}^{l-1+\alpha}(\partial\Omega_0)$. Since one degree of regularity is lost at each iteration, iteration cannot continue after a few steps. To cure this, we need to precondition the iteration by writing:

$$\gamma^* \quad = \quad \gamma \quad - \quad \rho \ \ B \ \ g_{L^2} \ .$$

where the self-adjoint invertible operator B has to be chosen in order to recover the degree of regularity lost by the L^2 gradient. The functional interpretation of preconditioning was early pointed in the Least Square formulation of [19]. The preconditioned continuous algorithm will converge at a rate that evidently does not depend on a mesh size. Then we can try to build some consistent discretization that will hopefully converge with a rate not so different from the continuous rate. This means that *essentially mesh-independent rates* might be obtained. Another way to understand that point is to remember that in the linear periodic case, Fourier's analysis shows that operators involving p-th order spatial derivatives will have eigenvalues of the order of $\left(\frac{1}{\Delta x}\right)^p$. The smaller this degree, the better the condition number. The strategy which we propose is to build the operator B in such a way that **the resulting order of spatial differentiation after preconditioning be equal to zero**.

Remark 1: One possible option is to consider the Sobolev space $H^1(\partial\Omega_0)$ as pivot space for the continuous gradient iteration. This means that the descent direction to use can be computed on the boundary by solving the following system set:

$$G \;=\; B^{LB} g_{L^2} \text{ where } -\Delta^{LB}\, G \,+\, G \;=\; g_{L^2} \text{ on } \partial\Omega_0$$

in which Δ^{LB} is the Laplace-Beltrami operator. The resulting preconditioned iteration is the one proposed in [3][2]. Since the Laplace-Beltrami operator is of degree 2, the gain in regularity with this preconditioner is $+2$, that might be not optimal since our analysis shows that the right gain is 1. □

2.2 The Euler case

It is probably a generic property of shape first derivatives (with respect to shape location) that the shape gradient of the state variable has a derivative loss of 1. The above analysis shows it rigorously for the Dirichlet problem. We consider now a more complex model: the flow around an obstacle of boundary $\partial\Omega_\gamma$ is considered. We denote by $W_1, W_2, W_3, W_4, W_5, p$ the conserved variables (density, three moments, energy) and the pressure of the flow of a perfect gas. The state equation is now the system of steady Euler equation that we write under a variational form as follows: for all

$\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$ belonging to the appropriate space,

$$\begin{aligned}(\Psi(\gamma, W), \phi) &= - \int_{\Omega_\gamma} (F(W).\frac{\partial \phi}{x} + G(W).\frac{\partial \phi}{y} + H(W).\frac{\partial \phi}{z})\ d\Omega_\gamma \\ &+ \int_{\partial \Omega_B} (F_B n^x + G_B n^y + H_B n^z).\phi d\sigma \\ &+ \int_{\partial \Omega_\gamma} p\ (n_x^\gamma\ \phi_2\ +\ n_y^\gamma\ \phi_3\ +\ n_z^\gamma\ \phi_4)\ d\sigma = 0, \qquad (5)\end{aligned}$$

where $F(W)$, $G(W)$ and $H(W)$ hold for the usual Euler fluxes, corresponding respectively to each of the space directions, where $\partial\Omega_\gamma$ is the variable shape with slip boundary, and where $\partial\Omega_B$ holds for a fixed part of domain boundary with conditions represented by fluxes F_B, G_B, H_B that we do not need to specify further for our argument.

The cost functional is denoted in a generic manner as follows:

$$j(\gamma) = J(\gamma, W(\gamma))\ . \qquad (6)$$

where $W(\gamma)$ is the solution of state system for the parameter γ. On the way of a rigorous Hadamard differentiation with respect to shape, we would need the invertibility of Jacobian for the Euler system if we want justify the application of implicit function theorem. We propose instead a formal derivation following the lines of [20]. The adjoint state Π is solution of:

$$\left(\frac{\partial F}{\partial W}\right)^* \frac{\partial \Pi}{\partial x} + \left(\frac{\partial G}{\partial W}\right)^* \frac{\partial \Pi}{\partial y} + \left(\frac{\partial H}{\partial W}\right)^* \frac{\partial \Pi}{\partial z} \quad = \quad - \ \frac{\partial J}{\partial W}\ .$$

in combination with boundary conditions, among which we mention only the boundary condition on the shape boundary controlled by the parameter γ:

$$\Pi_2 n_x^\gamma + \Pi_3 n_y^\gamma + \Pi_4 n_z^\gamma \quad = \quad 0 \quad \text{on} \quad \partial\Omega_\gamma .$$

Similarly to the previous section, the gradient of cost functional is a product of adjoint state by the -formal, this time- derivative of state residual in which state equation has been re-injected. After simplifications, we get:

$$\begin{aligned} g_{L^2}(\gamma, W, \Pi) = \quad & - \quad (F(W)\frac{\partial \Pi}{\partial x} + G(W)\frac{\partial \Pi}{\partial y} + H(W)\frac{\partial \Pi}{\partial z}) \ (\vec{n}^\gamma \cdot \vec{V}) \\ & + \quad (\nabla p \ \Pi \ + \ p \ \nabla \Pi) \ (\vec{n}^\gamma \cdot \vec{V}) \ . \end{aligned} \tag{7}$$

The gradient correction then writes:

$$\gamma = \gamma \ - \ \rho g_{L^2}(\gamma, W, \Pi) \ . \tag{8}$$

We observe that, as in the case of Dirichlet model, this correction is probably less regular than the boundary parameter γ. Indeed, inspired by elliptic smoothness theory, we can estimate that the state variables are at most as regular as the boundary, but the above correction involves first derivatives of the adjoint state. Additionally, the normal vector $\vec{n}^\gamma$ is a first derivative of the boundary parametrization γ. Therefore the formal loss of regularity is again 1.

3 Additive multilevel preconditioners

Additive multilevel preconditioners have been initially derived in a discrete context for solving elliptic Partial Differential systems which are typically of second order or of even order. A rich litterature exists on this topic. The hierarchical basis method was first analysed by Yserentant in his pioneering paper [21]. The work of Yserentant was apparently motivated by the famous unpublished technical report of Bank and Dupont [22]. A more complete theory was proposed by Bramble, Pasciak and Xu [5], [23]. See also the wavelet extensions, for example in [7]. An extended theory can be found in [24, 6]. The purpose of this section is to recall some results of [6] [25] in a format adapted to our purpose. For simplicity, we state them in the case of Dirichlet boundary conditions.

Let Ω be a regular subdomain of R^n. Let $H^1(\Omega)$ the usual Sobolev space and $V = H_0^1(\Omega)$ its subspace of functions vanishing at boundary $\partial\Omega$. It is included in $H = L^2(\Omega)$, a pivot space the scalar product and norm of which are denoted by:

$$(u, v) \;=\; \int_\Omega u \; v \;\; dx; \;\; \|u\| = \; (u, u)^{1/2}$$

Let $(V_k)_{k=0,1,\ldots}$ be a sequence of discretisation subspaces of V :

$$V_0 \subset \cdots \subset V_k \subset \cdots \subset V$$

To fix the ideas, subspaces V_k's can be considered as built from nested quasi uniform meshes with mesh size:

$$h_k \ \approxeq \ 2^{-k} \ .$$

For any k we introduce the fine-to-coarse projection operator $Q_k : V \longrightarrow V_k$ defined for all $u \in V$ by

$$Q_{-1}u = 0 \ ; \ \text{and for } k \ \geq \ 0, \ (Q_k u, v) = (u, v) \ \ \forall v \in V_k \ .$$

Let $a \in [0, 3/2]$ and B_a defined by:

$$B_a u \ = \ \sum_{k=0}^{+\infty} \left(\frac{1}{2^a}\right)^k \ (Q_k u - Q_{k-1}u) \ . \tag{9}$$

In the case $a = 0$, B_1 is the identity. The difference $Q_k u - Q_{k-1}u$ is the component u that is exactly of scale k. For $a > 0$, the higher is k, the more the corresponding component is damped. The functional smoothing effect of B_a has been evaluated by Dahmen and Kunoth in Sobolev spaces as follows:

Theorem [25] :

Let s be such that $-3/2 \ \leq \ s \ \leq \ s + a \ \leq \ 3/2$. Then the operator B is bounded from $H_0^s(\Omega)$ into $H_0^{s+a}(\Omega)$, in particular, there exists two positive numbers c and C such that for all $u \in H_0^s(\Omega)$, we have

$$c \|u\|_{H^s(\Omega)} \leq \|B_a u\|_{H^{s+a}(\Omega)} \leq C \|u\|_{H^s(\Omega)} .$$

□

This statement shows that the operator B has smoothing properties that can be quantified as a gain in regularity. This gain is exactly a. It can be prescribed by the user in order to precondition an operator having a as regularity loss. We introduce in the next sections the agglomeration-based construction of the V_k spaces.

4 Preconditioning by node agglomeration

The usual BPX preconditioner essentially needs embedded meshes, an option that is not compatible with engineering applications which rely on unstructured meshes. An extension of this preconditioner to unstructured meshes was proposed in [11] for equations. It can be in a rather straightforward manner further extended to optimization problems. The first section describes the case of a 2D or 3D mesh. The second section addresses the particular features of an unknown defined on a non-plane surface discretised by triangles in 3D.

4.1 Multidimensional Agglomeration

We start from a fine triangulation or tetrahedrization. Nodes are located at vertices. The discrete fine level is the subspace of linear combinations of $P1$ shape functions: $E_h = span\{\varphi_i,\ i = 1 \cdots n_h\}$.

A dual finite-volume cell C_i is defined around each vertex i by splitting each neighboring element with median plans and keeping subelements containing vertex i. The volume of C_i is denoted $\mathcal{M}eas\ (i)$. E_h is equipped by the following weighted scalar product:

$$\forall\ u_h\ and\ v_h \in E_h \quad (u_h, v_h)_h = \sum_i^{n_h} (u_h)_i\ (v_h)_i\ \mathcal{M}eas\ (i) \tag{10}$$

The agglomeration process relies on a partition of the set $I^f = \{1, \cdots, i, \cdots, n_h\}$ of fine indices i.

$$I^f = I_1 \cup \cdots \cup I_J \cup \cdots \cup I_{n_{2h}} \quad (n_{2h} \ll n_h) \tag{11}$$

where any I_J involves the indices of a few neighbouring nodes. An algorithm for building such a partition can be found in [8]. For any I_J, a coarser basis

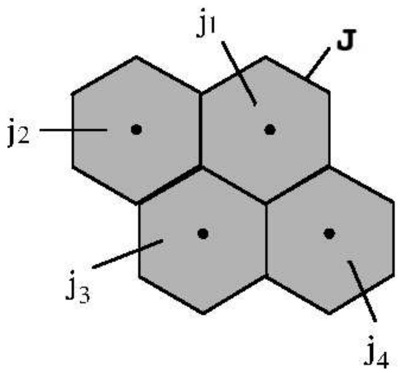

Figure 1: *Sketch of the agglomeration of four fine (2D) cells,* $C_{j1}, C_{j2}, C_{j3}, C_{j4}$ *into a coarser one* $\bar{C}_J = C_{j1} \cup C_{j2} \cup C_{j3} \cup C_{j4}$

function is defined by: $\Phi_J = \sum_{i \in I_J} \varphi_i$ and the coarser space is given by:

$$E_{2h} = Span\{\Phi_J,\ J = 1 \cdots n_{2h}\}$$

The linear prolongation operator, $\overline{P}$, from E_{2h} to E_h is defined by

$$\forall u_{2h} \in E_{2h} \quad \overline{P} u_{2h} = u_{2h} \ \in \ E_h \ . \tag{12}$$

Its adjoint $\overline{P}^*$ is the restriction operator from E_h to E_{2h}, and it is defined via $\overline{P}$ as its adjoint with respect to scalar product 10). by :

$$(\overline{P}^* u_h)_J = \frac{\displaystyle\sum_{j_m \subset J}^{n_h} (u_h)_{j_m} \mathcal{M}eas(j_m)}{\mathcal{M}eas(J)} \tag{13}$$

where J is a coarse cell, j_m are fine cells included in J. $\mathcal{M}eas(J)$ represents the measure of coarse cell J . These operators where introduced in [10] for a different multi-level approach, of multiplicative type. In the BPX original work, the Galerkin nested finite element sequence of discretised spaces enjoy the necessary regularity for the theory to apply. In contract to this, the proposed volume agglomeration method produces step functions on finer meshes. More precisely, the sequence of spaces generated with the transfers $\bar{P}$ and $\bar{P}^*$ defined in (12) and (13) does not enjoy enough regularity according the theory of Section 3. This is related to the fact that the orders of accuracy of transfers have a sum equal to $1+1=2$, not strictly larger than 2, see [26]. We have to define **smoother transfers**. In case where the solution of a well identified linear system is considered, its relaxation can be used for smoothing the transfer operators, as in [27]. We choose here the more simple option of [10]. It consists of an averaging between a node and its neighbors:

$$(Lu)_i = (1-\theta)u_i + \theta \, \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} \mathcal{M}eas(j)\ u_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} \mathcal{M}eas(j)} \tag{14}$$

where $\mathcal{N}(i)$ represents the set of neighbors of cell i and θ is set equal to $\frac{1}{2}$ according to the analysis of [10].

The adjoint L^* of L for L^2 scalar product is:

$$(L^*u)_i = (1-\theta)u_i + \theta \sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{u_j\ \mathcal{M}eas(j)}{\displaystyle\sum_{k\in\mathcal{N}(j)\cup\{j\}} \mathcal{M}eas(k)}\ . \tag{15}$$

The coarse level V_k is built from V_{k+1} as follows:

$$V_k \quad = \quad L_k \ \bar{P}_k \ (V_{k+1}) \tag{16}$$

Starting from the initial fine space that we denote V_N, we define the analog of the BPX projection operator (3) to the directly coarser level as follows:

$$Q_N \quad = \quad L_N \ \bar{P}_N \ \bar{P}_N^* \ L_N^* \tag{17}$$

and for any coarser level:

$$Q_k \quad = \quad \Pi_{i=k+1}^{i=N} L_i \ \bar{P}_i \ \Pi_{j=N}^{j=k+1} \ \bar{P}_j^* \ L_j^* \ k = 0, N-1 \ , \ Q_1 = 0. \tag{18}$$

From this we get:

$$B_a \quad = \quad \sum_{k=0}^{N} \left(\frac{1}{2^a} \right)^k (Q_k - Q_{k-1}) \ . \tag{19}$$

In practice, for any u, $B_a u$ is computed within a unique cycle from fine to coarse and back.

4.2 Agglomeration for a surface in 3D

Let Σ_0 be the initial 3D discrete surface, made of triangles. The generic discrete surface Σ_γ, is defined by the translation of length γ, of the vertices of Σ_0 along an approximate unit normal vector $\vec{n}$ to Σ_0 defined at vertices.

$$\vec{x}_i^\gamma \text{ is a vertex of } \ \Sigma_\gamma \quad \Leftrightarrow \quad \vec{x}_i^\gamma = \vec{x}_i^o + \gamma(i) \ \vec{n} \tag{20}$$

where i is the index of the vertex, $\vec{x}_i^o$ is the physical position of the vertex of Σ_0 with same index i. In order to precondition a correction on γ, we can construct a sequence of spaces and operators following the same process as in (11),(12), (13), but restricted to the surface and with the area of surfacic cells $\mathcal{A}rea(j)$ instead of cell measures. In order to adapt our operators to irregular surfaces, the smoothing operator L, is now weighted by a scalar product of normals to the surface:

$$(L\ \vec{x})_i = (1-\theta)\vec{x}_i + \theta \, \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}\vec{x}_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}} \qquad (21)$$

where w_{ij} are the weights defined by :

$$w_{ij} = \max\ \left(\mathcal{A}rea(i)\vec{n}_i \cdot \mathcal{A}rea(j)\vec{n}_j\ ,\ 0\right) \quad \|\vec{n}_i\| = 1\ \ \forall\ i\ . \qquad (22)$$

With this weighting, the smoothing is avoided on shape's dihedra (e.g. on wing trailing edges) and then allows changes in the angle. The rest of preconditioner definition is the same as in previous section.

5 Combination with the BFGS acceleration

Modern optimisation algorithms are equipped with quasi-Newton processes which allow a rather good efficiency for many applications. One of the most efficient quasi-Newton acceleration relies on the BFGS method, see for example [28]. The purpose of this section is to demonstrate on a simplified example that even for a very favourable context, a good quasi-Newton optimizer needs the complement of a preconditioner if the number of discretization unknowns is large.

We consider the following minimisation problem in 1D:

$$u_{opt} \quad = \quad \text{ArgMin}\ \frac{1}{2}\int_0^1 (\ |\frac{\partial u}{\partial x}|^2 - fu)\ dv\ ;\ u(0) = u(1) = 0\ . \qquad (23)$$

We can precondition this problem with the 1D version of our agglomeration preconditioner. The Hessian of this functional is a second-order differential operator, i.e. derivative loss is 2. This problem can be solved by a conjugate gradient and in that case we just need, for preconditioning into that algorithm, to multiply the gradient by B_2 defined as in as in (19) (with $a = 2$). The use of BGFS is a little more complex. That method builds progressively an approximate Hessian $\mathcal{M}$ for the functional to minimise, together with its inverse $\mathcal{W}$. This is done by *regula falsi* heuristics relying on the knowledge of successive values u_{k-1}, u_k of the control variable and the corresponding values of the functional gradient g_{k-1}, g_k. In order to introduce our preconditioner, we define:

$$\hat{u} \;=\; B_1 u \;\text{ and }\; \hat{g} \;=\; B_{-1} g \tag{24}$$

These variables allows the BFGS construction of $\hat{\mathcal{W}}$ and $\hat{\mathcal{M}}$. We can then derive and inverse the matrix to apply to the gradient:

$$\mathcal{W} \;=\; B_{-1}\hat{\mathcal{W}}B_{-1} \quad \mathcal{M} \;=\; B_1\hat{\mathcal{M}}B_1 \tag{25}$$

We stress that, in contrast to the conjugate gradient, in order to precondition BFGS, we need to handle B_1 and B_{-1} for the assembly of the BFGS matrix and of its inverse. This is easily done with the proposed multilevel preconditioner: we just choose respectively $a = 1,\; a = -1$.

In Figure 2, four experiments are depicted. The number m of unknowns in the discretisation of (23) is first set to 32. As rather classically, the non-preconditioned BFGS algorithm solves exactly the optimum in about $5 + m/2 = 21$ iterations. The preconditioned version has a convergence which is more progressive, but not faster. For a larger number of unknowns, $m = 256$, the unpreconditioned BGFS shows a very slow convergence during at least the 100 first iterations. In large scale shape design, this convergence is not acceptable. In contrast, the preconditioned BFGS convergence is about the same as for the coarse-mesh one (21 iterations).

6 Application to an optimal shape design problem

We now study the adaptation of the proposed multi-level preconditioner to an optimal shape design loop.

6.1 The aerodynamical shape design problem

We now make more precise the statement of the aerodynamical application introduces in Sec.2.2. In [12], the authors propose a very simplified model for measuring the "sonic boom downwards emission" (SBDE). It consists in evaluating the volume integral of the squared pressure gradient in an "observation box" Ω^B (as shown in Figure 3) below the object. The cost functional is therefore the following:

$$j(\gamma) = J(\gamma, W(\gamma)) \ . \tag{26}$$

where $W(\gamma)$ is the solution of state system (5) for the parameter γ, and

$$J(\gamma, W) = \alpha_1 (C_D(W) - C_D^t)^2 + \alpha_2 (C_L(W) - C_L^t)^2 + \alpha_3 \int_{\Omega^B} |\nabla \tilde{p}(W)|^2 dV. \tag{27}$$

Here α_1, α_2 and α_3 are constants that prescribe the weights of three subcriteria, related to lift, drag and sonic boom emission. C_L^t and C_D^t hold for target lift and drag. Lift coefficient $C_L(W)$, drag coefficient $C_D(W)$, and pressure field $p(W)$ are computed from the design variable γ by solving the state equation (5) and obtaining the state variable W. The notation $\tilde{p}(W)$ indicates that the pressure $p(W)$ has been smoothed in order to integrate its gradient, even when shocks arise.

Practically, the observation box Ω^B is a part of the computational domain placed below the airplane. Its upper boundary is a plane close below the aircraft.

6.2 Discretized problem

The discrete CFD model uses an upwind Euler solver applying to unstructured tetrahedrizations. The shape is changed by moving the nodes on the boundary of the mesh along normals to that boundary. Their displacement is taken into account by a transpiration condition in order to avoid costly remeshings. A gradient of the discrete functional is computed with the help of an adjoint system, built using the Automatic Differentiation tool `TAPENADE`, [1]. Since this gradient is defined as a scalar field defined on the nodes of the boundary, we can apply to it the surface multilevel preconditioner introduced in Sec. 5.2.

6.3 Numerical results

We consider the shape optimisation starting from an ONERA M6 wing. A coarse discretisation of this geometry, together with a pressure distribution on it is presented in Fig.4. After the necessary number of gradient iterations, the shape optimisation loop produces a new shape, that is quite different from the initial one, as illustrated in Fig.5. The resulting sonic boom reduction can be appreciated from the comparisons of the pressure variations under the wing for both geometries, which are depicted in Fig. 6. This can be obtained by various optimisation algorithms. The question is to obtain it in the most efficient way, and in particular to analyse the impact of our preconditioner.

First we want to show that the optimal parameter $a = 1$ predicted by the theory is numerically verified. We first study this with a coarse geometry, involving 2203 nodes for the 3D mesh, but yet 780 shape parameters. Two optimisation algorithms are applied, a gradient (Fig.7) and a conjugate gradient (Fig.8). Divergence of the Euler algorithm for distorted shapes did not allow a pertinent study of the BFGS option.

Convergence of the gradient iteration can be evaluated with the evolution of the gradient norm. The case without preconditioning is the case where $a = 0$. Then convergence is the slowest of the different options tested. Values like $a = 0.5$, $a = 1.5$, $a = 1.$, $a = 2.$ provide good speedups, in particular for the 10 first iterations. This point is important for the shape design loops which, in practice, are too computer consuming to allow for more optimisation iterations. But the theoretical value $a = 1.$ appears as numerically optimal, for both cases of a pure gradient optimiser as well as a conjugate gradient one.

Let us verify that the efficiency of the preconditioner is good for finer discretisations. We consider a second mesh with 15463 nodes. The number of shape parameters is 3222. See Fig.9 and Fig.10.

The value $a = 1.$ appears again as numerically optimal. In the case of the conjugate gradient, a good shape is obtained in about 10 iterations.

The case of $a = 2.$ deserves some comments. It is equivalent to a Laplace-Beltrami smoothing and, according to Remark 1, to the choice of a H^1 scalar product for the gradient iteration. All the above results show that that option is not bad, but is clearly not as good as the optimal one $a = 1.$.

We complete the above convergence curves by Tables. 1 and 2 which com-

Table 1: *Sonic boom optimisation: result of 10 iterations of CG*

Case	Opt. Iter.	funct.	|gradient|
GC, N=780, M=11,000	10	0.02	0.3
GC preconditioned, N=780, M=11,000	10	0.04	0.04
GC, N=3222, M=77,000	10	0.06	0.3
GC preconditioned, N=3222, M=77,000	10	0.03	0.1

pare the efficiency of the preconditioned algorithm to the standard one in the practical case where convergence of optimisation is not continued more than 10 iterations. If equivalent levels of convergence are asked from both algorithms, then an acceleration of a factor 5 is put in evidence. In [29] the proposed preconditioner is applied to more complex geometries and in [30] the method is extended to multi-discipline optimization.

7 Conclusion

This paper presents a new preconditioning strategy for optimal shape design. An additive multilevel preconditioner is built from (a) the classical Bramble-Pasciak-Xu principle, and (b) the agglomeration principle. Functional analysis considerations show the central role of the loss of regularity

Table 2: *Sonic boom optimisation: effort for dividing the cost by 8*

Case	Opt. Iter.	|gradient|	Speed up
GC, N=780, M=11,000	15	0.1	
GC preconditioned, N=780, M=11,000	3	0.1	5
GC, N=3222, M=77,000	100	0.03	
GC preconditioned, N=3222, M=77,000	20	0.04	5

in the gradient iteration. This loss can be compensated by a regularity gain carried by the preconditioner. The user can prescribe this regularity gain at any real value. This helps in particular to combine the preconditioner with a BFGS acceleration. The interest in combining both is demonstrated.

In the case of shape design, we exhibit a simplified example for which the loss of derivative can be rigorously evaluated and is equal to 1.

The final shape design application is a pre-industrial one already addressed by different methods [31]. The numerical results confirm the *a priori* analysis of the regularity loss. They demonstrate that the proposed method improves notably the convergence of the shape design iteration. The proposed preconditioner is expressed in a rather general formulation and appeared to be quite efficient for preconditioning an operator of pseudo-differential type. It could be interesting to try it on other operators of this type (influence matrix, Shur operator for example).

8 Acknowledgements

This work was partly supported by the Comite d'Orientation pour l'Avion Supersonique.

References

[1] Tropics team. On-line documentation of the Tapenade AD tool. Technical report, INRIA, Sophia-Antipolis, France. http://www.inria.fr/tropics.

[2] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids*. Clarendon Press - Oxford, 2001.

[3] J. Reuther and A. Jameson. Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory. AIAA Paper 95-0123, 1995. 33rd Aerospace Sciences Meeting and Exhibit.

[4] E. Arian and S. Ta'asan. Analysis of the Hessian for aerodynamic optimization: Inviscid flow. *Comput. Fluids*, 28(7):853–877, 1999.

[5] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.

[6] J. Xu. An introduction to multilevel methods. In M. (ed.) et al. Ainsworth, editor, *Wavelets, multilevel methods and elliptic PDEs. 7th EPSRC numerical analysis summer school, University of Leicester, Le-*

icester, GB, July 8–19, 1996., pages 213–302. Oxford: Clarendon Press. Numerical Mathematics and Scientific Computation., 1997.

[7] A. Cohen. Wavelet methods in numerical analysis. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume VII, pages 417–713. Elsevier Science, 2000.

[8] M.-H. Lallemand, H. Steve, and A. Dervieux. Unstructured multiggridding by volume agglomeration : current status. *Computer Fluids*, 21(3):397–433, 1992.

[9] D. Mavriplis. Parallel performance investigations of an unstructured mesh Navier-Stokes solver. *ICASE Report 2000-13*, March (2000).

[10] N. Marco and A. Dervieux. Multilevel parametrization for aerodynamical optimization of 3d shapes. *Finite Elements in Analysis and Design*, 26:259–277, 1997.

[11] B. Koobus N. Marco and A. Dervieux. An additive multilevel preconditioning method. *Journal of Scientific Computing*, 12(3):233–251, 1997.

[12] M. Vazquez, A. Dervieux, and B. Koobus. Aerodynamical and sonic boom optimization of a supersonic aircraft. INRIA Report 4520, 2002.

[13] P.R. Garabedian. Partial differential equations. *Wiley*, New-York, 1964.

[14] F. Murat and J. Simon. Quelques résultats sur le contrôle par un domaine géométrique (in french). *Publications du Laboratoire d'Analyse Numérique, university of Paris VI*, 1974.

[15] O. Pironneau. *Optimal shape design for elliptic systems.* Springer-Verlag, 1984.

[16] J. Céa. Conception optimale ou identification de forme. Calcul rapide de la dérivée directionnelle de la fonction coùt. *RAIRO Modél. Math. Anal. Numér.*, 20:371–402, 1986.

[17] A. Dervieux. Perturbation des équations d'équilibre d'un plasma confiné. INRIA Report 0018, 1980.

[18] A. Dervieux and B. Palmerio. Une formule de Hadamard dans des problèmes d'identification de domaines; exemples. *C. R. Acad. Sc. Paris*, 280, Serie A:1761–1764, 1975. (in French).

[19] M.-O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, and O. Pironneau. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods (I). least squares formulation

and conjugate gradient solutions of the continuous problems. *Comput. Meths. Appl. Mech. Engrg.*, (17/18):619–657, 1979.

[20] F. Beux and A. Dervieux. Exact–gradient shape optimization for a 2-D Euler flow. *Finite Elements in Analysis and Design*, 12:281–302, 1992.

[21] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[22] R.E. Bank and T. Dupont. Analysis of a two-level scheme for solving finite element equations. Technical report, CNA-159, Center for Numerical Analysis, University of Texas at Austin, 1980.

[23] J. Xu. *Theory of multilevel methods.* PhD thesis, Cornell university, 1989.

[24] A. Kunoth. *Multilevel Preconditioning.* PhD thesis, University of Berlin, 1994.

[25] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.

[26] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin-Heidelberg-New-York, 1985.

[27] P. Vanek, A. Janka, and H. Guillard. Convergence of Algebraic Multigrid based on smoothed aggregation ii: Extension to a Petrov-Galerkin method. INRIA report RR-3683, 1999.

[28] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, 1999.

[29] M. Vazquez, B. Koobus, and A. Dervieux. Multilevel optimization of a supersonic aircraft. *Finite Element in Analysis and Design*, 40:2101–2124, 2004.

[30] M. Vazquez, A. Dervieux, and B. Koobus. A methodology for the shape optimization of flexible wings. *Engineering Computations*, to appear, 2005.

[31] F. Courty and A. Dervieux. A SQP-like one-shot algorithm for optimal shape design. *Notes on Numerical Fluid Dynamics, V. Selmin Ed., Springer, to appear.*

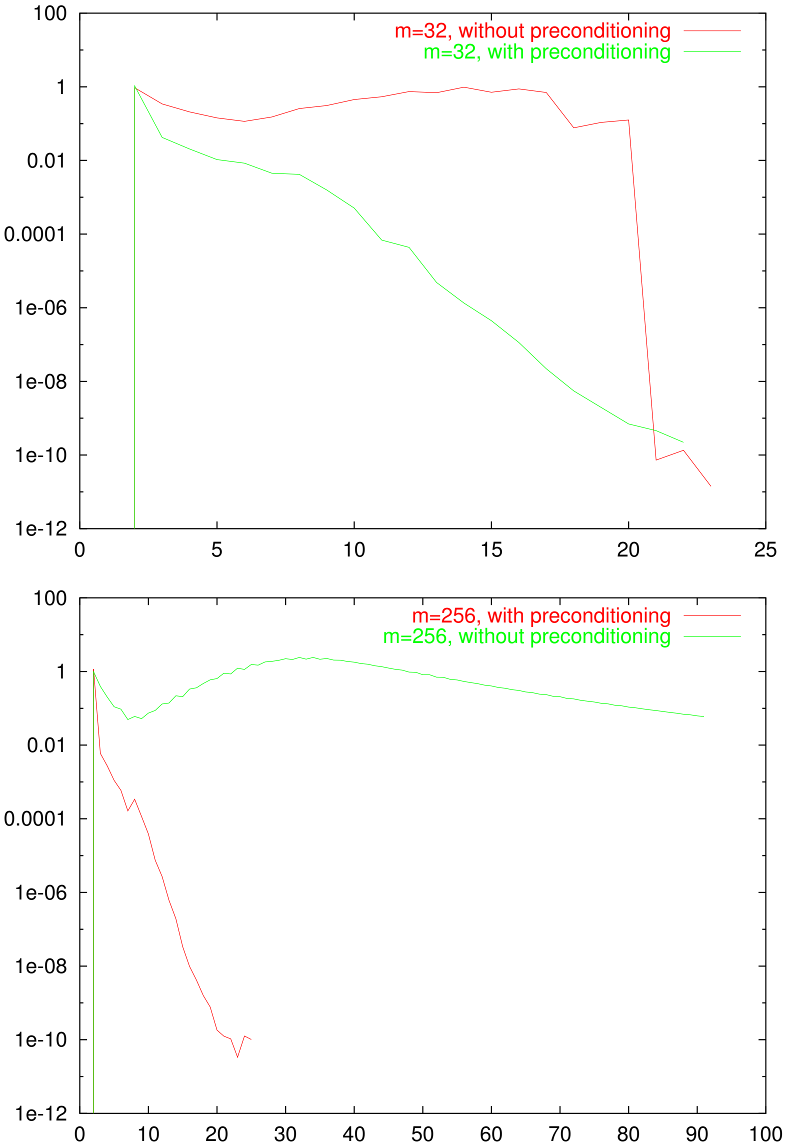

Figure 2: Resolution of a second order optimization problem with a preconditioned BFGS, behavior of the different residuals as functions of iteration number, for problem sizes m = 32 and m = 256.

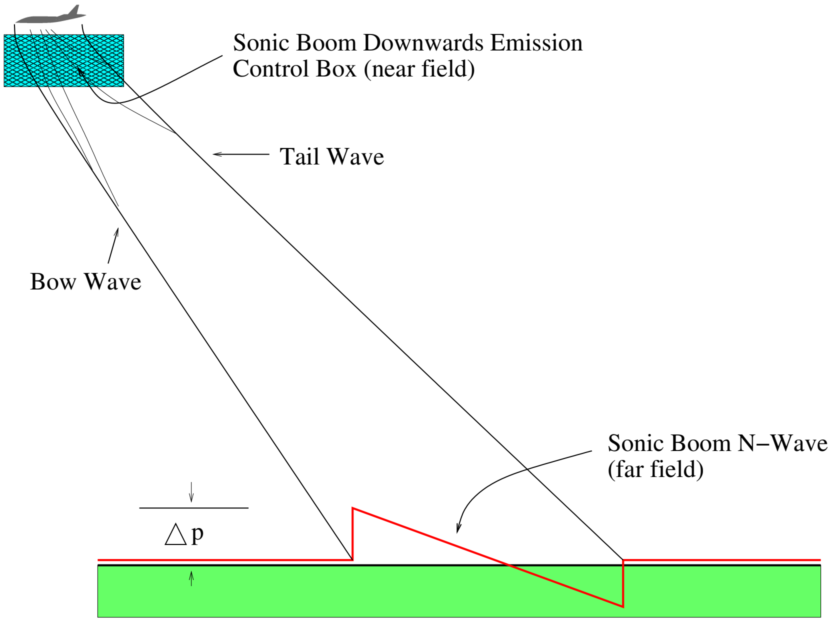

Figure 3: The sonic boom. Sketch of near and far field shock wave patterns of a supersonic aircraft. SBDE's control box B.

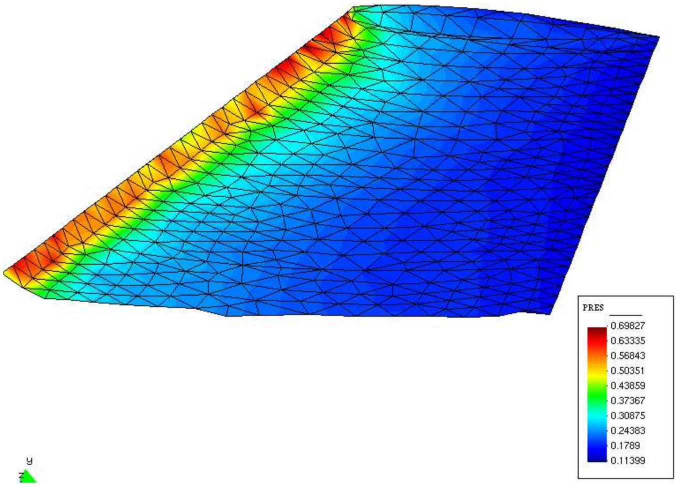

Figure 4: ONERA M6 shape optimisation: pressure contours on initial shape

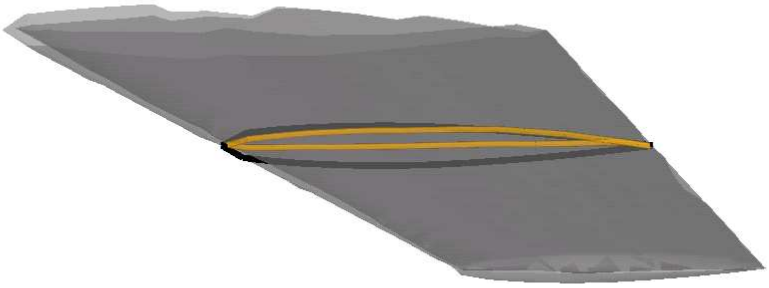

Figure 5: ONERA M6 shape optimisation: sketch of geometry, of initial and final shape.

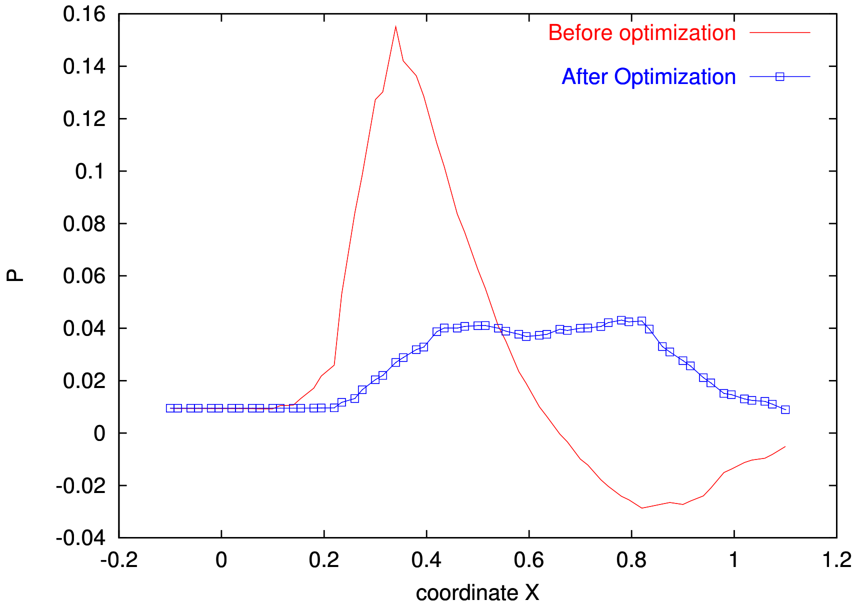

Figure 6: ONERA M6 shape optimisation: pressure signal under the wing, before, after optimisation

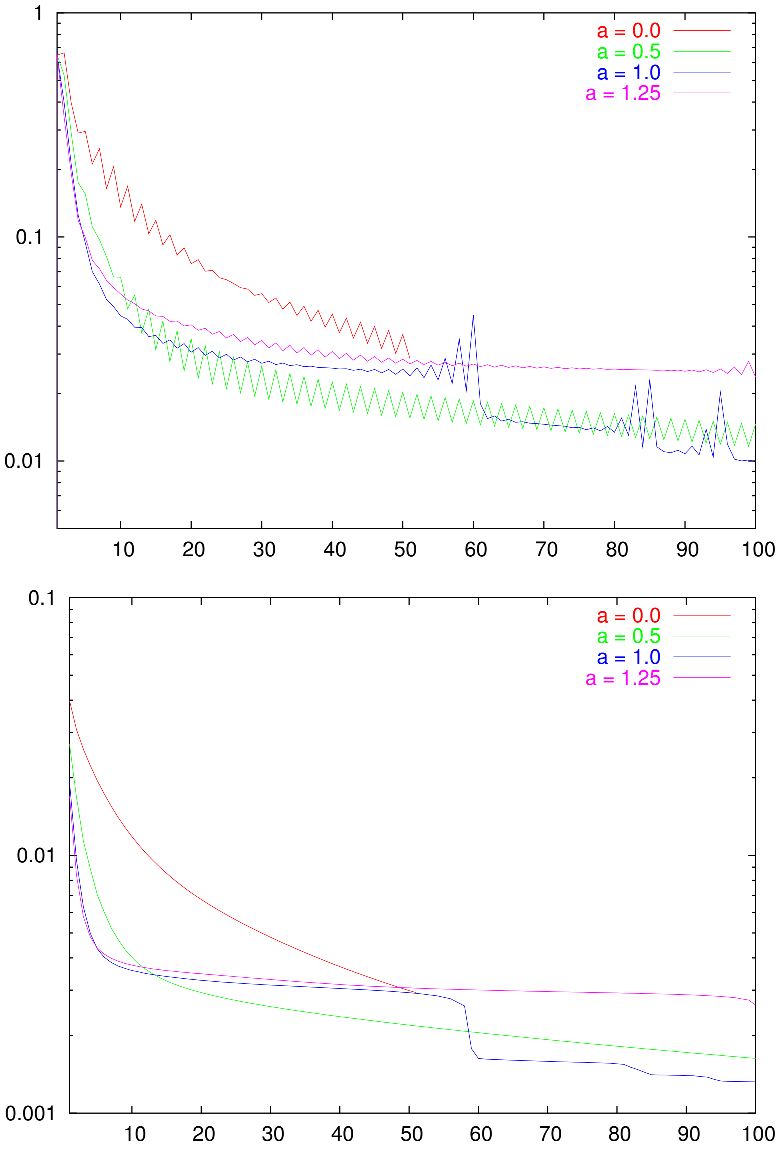

Figure 7: ONERA M6 shape optimization with a gradient method with different values of the preconditioner parameter a. Top: gradient convergence. Bottom: cost function. Coarse mesh with 2203 vertices

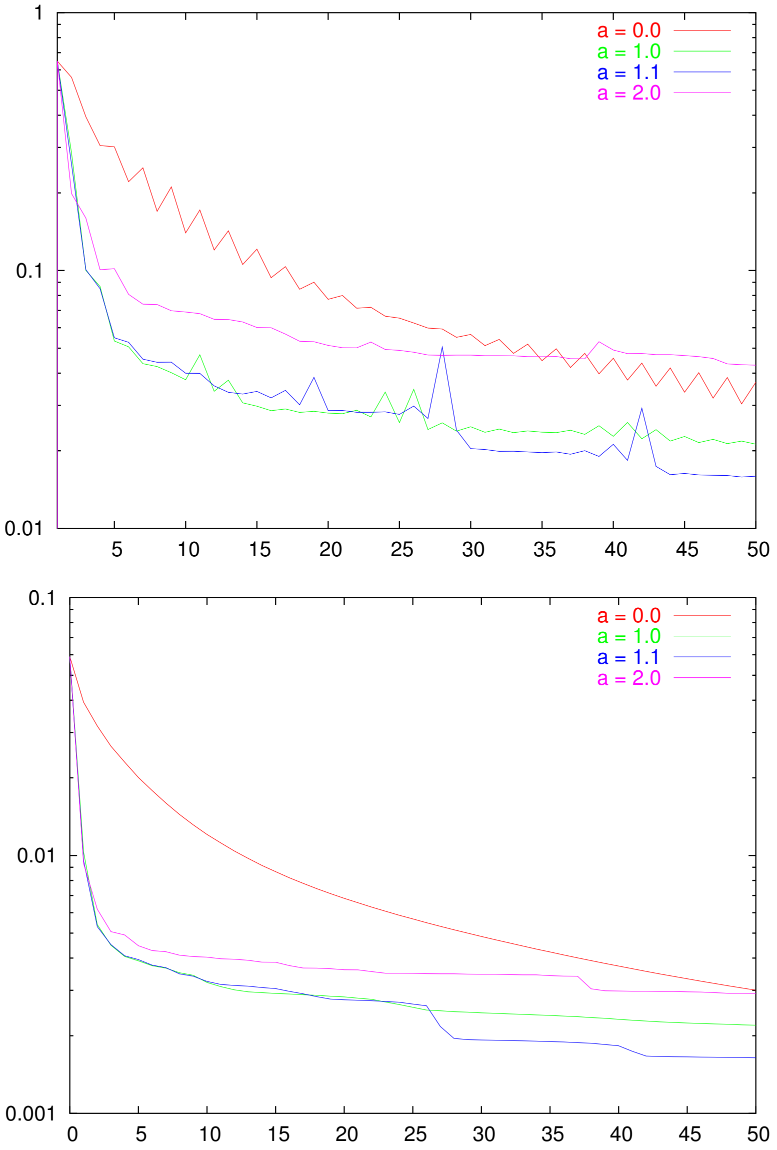

Figure 8: ONERA M6 shape optimization with a conjugate gradient method with different values of the preconditioner parameter a. Top: gradient convergence. Bottom: cost function. Coarse mesh with 2203 vertices

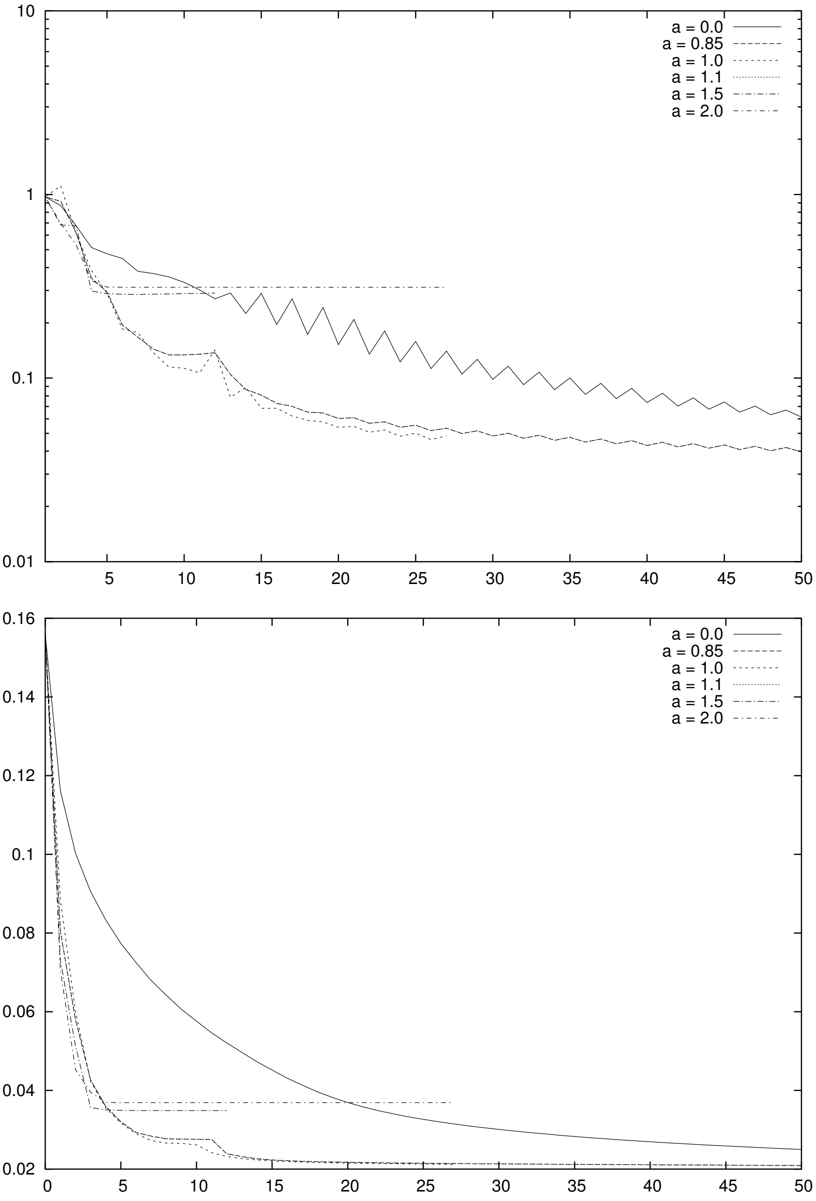

Figure 9: ONERA M6 shape optimization with a gradient method with different values of the preconditioner parameter a. Top: gradient convergence. Bottom: cost function. Medium mesh with 77315 vertices

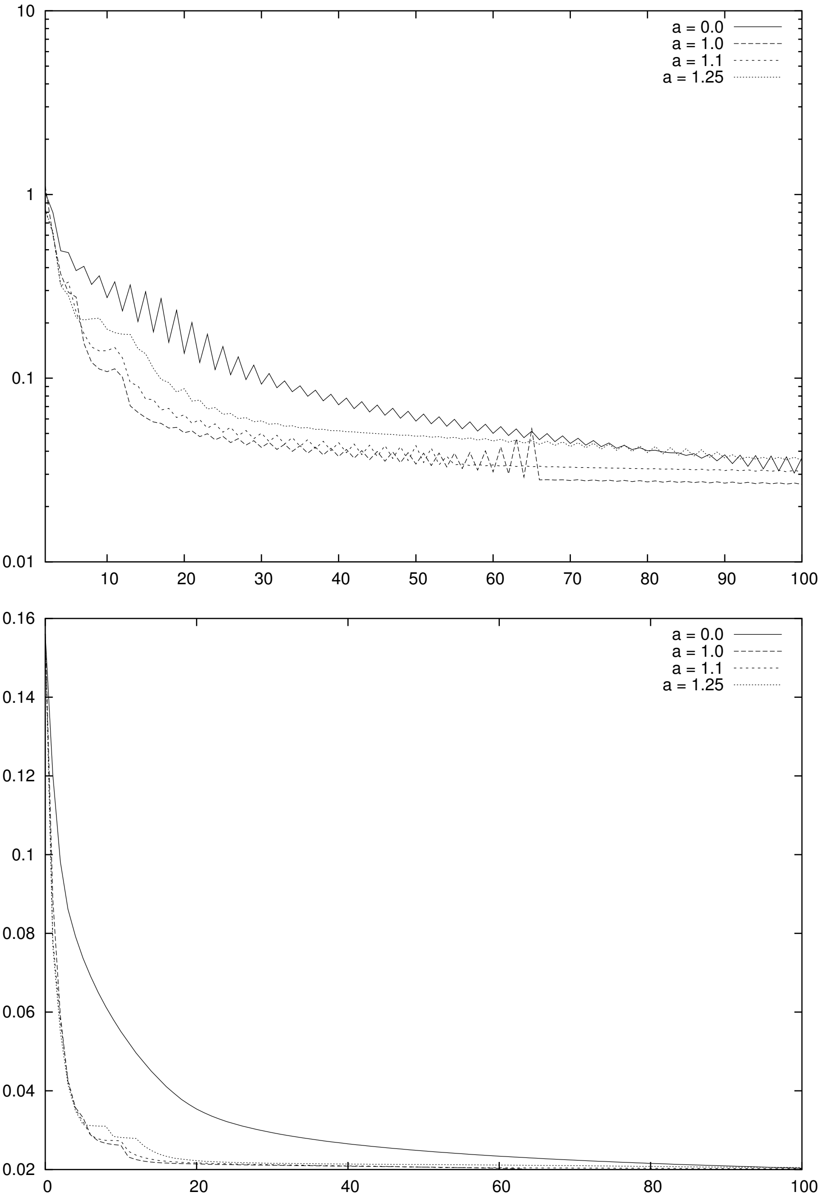

Figure 10: ONERA M6 shape optimization with a conjugate gradient method with different values of the preconditioner parameter a. Top: gradient convergence. Bottom: cost function. Medium mesh with 77315 vertices