INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Optimization loops for shape and error control: extended lecture notes

A. Dervieux, F. Courty, T. Roy, M. Vázquez, B. Koobus

N° 5413

December 15, 2004

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--5413--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Optimization loops for shape and error control: extended lecture notes

A. Dervieux*, F. Courty†, T. Roy‡, M. Vázquez§, B. Koobus¶

Thème NUM — Systèmes numériques
Projet Tropics

Rapport de recherche n° 5413 — December 15, 2004 — 41 pages

Abstract: The power and versatility of the so-called adjoint methods used in optimization problems is addressed by presenting a combined methodology: shape and error control. In a first part we describe novel technologies for solving shape design problems: constrained optimum formulation with adjoint, one-shot optimization algorithms and multi-level optimization preconditioning. The application to a shape design problem related to sonic boom minimization is reported. In a second part, after some remarks on mesh adaptation, we examine the problem of formulating mesh adaptation in terms of an optimization problem, presenting some applications to interpolation and to Partial Differential Equations. The implications of the proposed method become clear: by combining together both ideas cost effective methods can be developed where the optimization loops involve both shape optimization, to improve an aerodynamic design, *and* mesh adaptivity, to cope with the difficulties of the CFD problem involved.
Ce document présente la version détaillée des notes de cours au "PROMUVAL Short Course on Multidisciplinary Modelling Simulation and Validation in Aeronautics", qui s'est tenu à Barcelone (E), les 28-29 juin 2004.

Key-words: Partial Differential Equations, Finite Element Method, Mesh Adaptivity, Multilevel Optimization, Computational Fluid Dynamics, Optimal Shape Design

* INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia-Antipolis, France
† INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia-Antipolis, France
‡ INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia-Antipolis, France
§ Universitat de Girona, Av. LLuìs Santaló s/n - 17071 Girona, Spain
¶ Université de Montpellier II, Dept Mathématiques, CC.051, 34095 MONTPELLIER Cedex 5, France

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Contrôle de forme et d'erreur par Optimisation

Résumé : On utilise la puissance et la souplesse de la méthode de l'état adjoint pour construire une approche associant le contrôle du maillage et de la forme. Dans une première partie nous décrivons des méthodes nouvelles pour résoudre des problèmes de conception optimale de forme: formulation de la contrainte égalité par adjoint puis optimisation one-shot, optimisation multi-niveau. L'application à la réduction du bang sonique sert d'illustration. Dans une second partie, après une revue d'avancées récentes en adaptation de maillage, on examine cette problématique sous l'angle de l'optimisation, et on présente quelques applications à l'interpolation adaptative et à l'adaptation pour des EDP. Il est alors loisible, par l'association de ces techniques, de développer des méthodes efficaces d'optimisation avec adaptation incorporée, afin de mieux répondre aux besoins du design en Mécanique des Fluides Numérique.
This is the extended version of lecture notes for the "PROMUVAL Short Course on Multidisciplinary Modelling Simulation and Validation in Aeronautics", held in Barcelona (Spain), june 28-29, 2004

Mots-clés : Equations aux Dérivées Partielles, Eléments Finis, Adaptation de maillage, optimisation multi-niveau, Mécanique des Fluides Numérique, optimisation de forme

1 INTRODUCTION

Multidisciplinary modelling is an important ingredient for the solution of an engineering problems of utmost importance in industry: Multidisciplinary Optimization, MDO. Indeed, in many practical cases, the different disciplines involved interact with each other in a deep manner, a fact reflected in complex multi-physics design analysis. For instance, to determine the flutter speeds for an airplane, fluid-structure interaction must be simulated. Computer-aided optimization favourize a further quantification of design conditions, since constraints parameters have to be fixed and trade-off in costs have to be specified.

The rest of the process can then be automated by the integration into an optimization loop of the different analyses and of the cost and constraints evaluation. In the case where the number of parameters to optimize is very large, the sensitivity step necessary for improving the design parameters needs the computation of adjoint states as important ingredients of functional gradients. Now, if the different underlying analyses are obtained with numerical simulation, it becomes compulsory that **error margins** are imposed and respected. In order to respect error margins, two actions should be sucessful.

First, error has to be defined and measured. The optimal control formalism suggests to consider the error as a cost (or a constraint). This means that typically the set of errors should be gathered into a real-valued functional thanks to weights to be specified by the engineer. Then error estimates relying on adjoint states can be applied and give an error measure.

Second, when the error is too large, an efficient mechanism has to be applied in order to reduce it to an acceptable level. This can be done from **a posteriori** estimates by applying a stategy relying on a tolerance (see for instance [3]). In these notes we describes a strategy pushing further the idea of controlling the error, by considering the problem as an optimization one: to optimize the error functional with respect to mesh descriptors.

It becomes apparent then that the methods of Optimization and of Multidisciplinary Optimization can be very useful for the validation of Computational Mechanics, and in particular for Multidisciplinary computations, rendering them a very atractive and potentially powerful joint solution.

These notes address two issues of numerical modelization:

- the control of approximation error by mesh adaptation, and
- the optimization (or steady control) of a system with respect to a parameter.

There are two basic issues interacting with each other: (a) a functional can be optimised only if numerical accuracy of its evaluation is maintained small enough and (b), we shall see that numerical error control can become a genuine control problem. The key point of this work is that, in both cases, there is a common factor: the *adjoint state*. The adjoint state is both useful for optimal control for design and becoming very popular for the control of the approximation error in the evaluation of a functional computed from simulation data. In both cases we are lead to solve optimal control problems in which the number of parameters can be very large and it is mandatory to find better optimization algorithms in order to efficiently solve these problems. For that reason, we consider a joint and integrated work programme as a very appealing approach.

In the first part of these notes, we shall consider efficent methods for functional optimization under state equations constraints. The central application used for illustrate them will be the shape optimization for a supersonic wing or aircraft. The main points introduced here, appart from the general algorithm itself, are the so called one-shot convergence acceleration and the multilevel preconditioner for the optimization correction. The second part will focus on mesh adaptivity issues. We shall present mesh adaptivity methods as a key for effective second-order accuracy in aerodynamics, and as an issue well addressed by optimal control.

The reader can find some background informations in the following monographs: optimization methods, [35]; shape optimization in aerodynamics, [32]; approximation errors, [1].

The paper is organized as follows. The first part, devoted to Optimal Shape Design, describes a particular application in aerospace engineering and two of the basic concepts used: one-shot optimization and multi-level preconditioning. The second part addressess the issue of Numerical Error Reduction. It contains a brief but deep introduction of the main ideas involved, next it shows some numerical examples and finally it develops the issue of mesh adaptivity following the aforementioned lines. The last section is devoted to draw the master lines of an integrated scheme of optimization and mesh adaptivity.

Prologue

As a prologue, let us introduce the problem we have in hands. We consider the following constrained minimization problem:

$$\min J(u,Y), \quad \text{subject to} \quad \Psi(u,Y)=0 \tag{1}$$

where the minimum is taken with respect to the composite variable x:

$$x=(u,Y). \tag{2}$$

The variables u and Y describe the problem within certain spaces of work. Let us say that u are the control variables and Y are the state variables. They can take different roles according to the problem we are involved in: flow field variables or approximation error for Y and shape parameterization or a metric defining a particular local mesh description for u. The idea is to minimize the *objective function* $J(u,Y)$ looking for $\overline{x}$ which satisfies the *equality constraint* $\Psi(x)=0$. Let us assume that the Jacobian

$$A=\frac{\partial \Psi}{\partial Y} \tag{3}$$

is always invertible.

Starting from the $x_k=(u_k,Y_k)$, we define an iterative optimization procedure by taking

$$\begin{aligned} \Psi_k &= \Psi(x_k) \\ A_k &= \frac{\partial \Psi}{\partial Y}(x_k). \end{aligned}$$

In order to efficiently compute the required system output derivatives with respect to the inputs, we use the adjoint method, which provides generalized Lagrange multipliers in the form of a spatial field of the same dimension of the original problem unkowns. For each iteration k, the *co-state* or *adjoint state* Π_k is a solution of

$$\Pi_k = \left(\frac{\partial \Psi}{\partial Y}(x_k)\right)^{-T} \frac{\partial J}{\partial Y}(x_k) \ . \tag{4}$$

We denote the *functional gradient* by $g_k = g(x_k,\Pi_k)$. It writes:

$$g(x_k,\Pi_k) = \frac{\partial J}{\partial u}(x_k) - < \Pi_k \ , \ \frac{\partial \Psi}{\partial u}(x_k) > \tag{5}$$

In the case on no other constraint, the stationarity of the functional writes:

$$g(x_k,\Pi_k) = 0 \ , \tag{6}$$

and the set of three equations 4, 6 and the constraint definition in 1 is the Karush-Kuhn-Tucker (KKT) set of optimality conditions. As stated in the introduction, we will see how much both pervasive and useful is the concept of the adjoint problem.

2 OPTIMAL CONTROL LOOPS

In this section, we describe this formalism applied to Optimal Shape Design for a particular aerodynamics problem.

2.1 The application example

In [41], the authors propose to measure the "sonic boom downwards emission" (SBDE) by evaluating the volume integral of the squared pressure gradient in an "observation box" Ω^B (as shown in Fig. 1) below the object. The cost functional is therefore the following:

$$j(u)=\alpha_1(C_D-C_D^t)^2+\alpha_2(C_L-C_L^t)^2+\alpha_3\int_{\Omega^B}|\nabla p|^2 dV \tag{7}$$

where α_1, α_2 and α_3 are constants that prescribe the weights of three sub-criteria, related to lift, drag and sonic boom emission. They are evaluated by solving the 3D Euler equations around the geometry specified by the shape parameter. 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. Of course, in presence of shocks, the value of the SBDE term is infinite in the continuous case, but finite (and mesh dependent) in the discrete case. The control parameter u will be related to the shape γ of the aircraft.

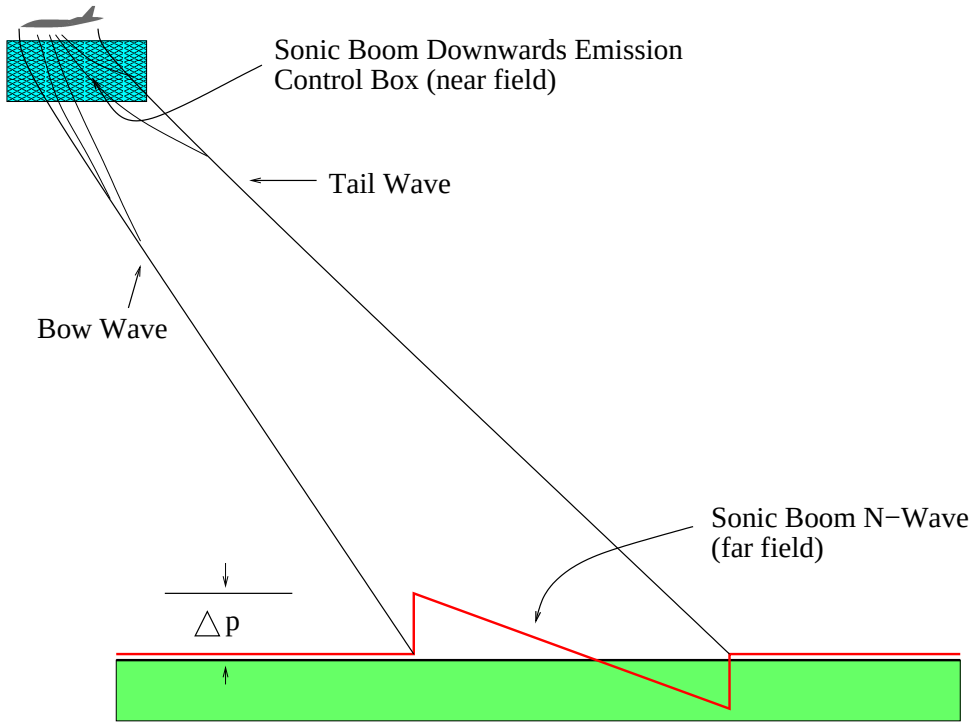

Figure 1: The sonic boom. Sketch of near and far field shock wave patterns of a supersonic aircraft. SBDE's control box Ω^B.

Discretized problem

The discrete CFD model uses an upwind Euler solver applying to an unstructured tesselation composed of tetrahedra. 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`, [25].

2.2 One-shot optimization

Modern finite-dimensional optimization methods relying on adjoints are issued from the Sequential Quadratic Programming methodology. A popular prototype is the Byrd-Omojokun algorithm, see [35]. This algorithm in its basic form for assumes that the resolution of the different linearisations of state systems (Newton iteration of state and solution of adjoint) are not expansive. However, this assumption is not valid in Optimal Shape Design. This fact has lead some authors to attack the problem using **one-shot** -or progressive, or simultaneous- algorithms (see for example [40], [14]), which are based in the following two points:

- First, advance the three equations of the KKT system at the same time in each optimization iteration.

- Next, instead of using linear generic solvers, use discipline-specific iterative, maybe nonlinear solvers (for example, pseudo unsteady solvers for Fluid Mechanics).

2.2.1 One-Shot-Trust-Region algorithm

The state equation solver to be integrated in the proposed optimizer is assumed to be an iterative fixed point one. It relies on a *pseudo-Newton* step on variable Y:

$$\delta Y \quad = \quad -S\ \Psi(x_k)$$

where the operator S is assumed to be reasonably inexpensive to evaluate and to (very) roughly approximate the inverse of the Jacobian $\frac{\partial \Psi}{\partial Y}$. Typically, it can be the result of a few Gauss-Seidel sweeps or diagonally preconditioned Conjugate Gradient iterations. This simplified solution is to be applied in place of both the SQP restoration phase and the computation of h_k in the SQP minimization phase.
We want to also do this in a similar way for the adjoint. In the proposed algorithm, the adjoint equation is considered as an additional equality constraint. Instead of solving

$$\Pi = (\frac{\partial \Psi}{\partial Y})^{-T} \frac{\partial J}{\partial Y},$$

its restoration will then consist in solving the simpler problem:

$$p = S^T \left((\frac{\partial \Psi}{\partial Y})^T \Pi - \frac{\partial J}{\partial Y} \right).$$

It is then natural to ask the new algorithm to restore approximatively both state constraint and adjoint constraint in two restoration phases before the minimization phase. By comparison with a classical SQP algorithm, applying only the incomplete solutions will decrease considerably the cost of one global iteration.
The main assumptions concerning the above pseudo-Newton iteration should involve:

- convergence properties of the state iteration,
- convergence properties of the adjoint iteration,
- their complexity,
- restoration of the descent property.

This leads to introduce the following assumptions:

- **Assumption 1 :** For $u \in I\!R^{N_u}$ and for $p \in I\!R^{N_Y}$,

  $$\|\Psi(Y - S\Psi(Y,u),u)\| \ \leq \ 0.99\ \|\Psi(Y,u)\|$$

  and

  $$\|\frac{\partial \Psi}{\partial Y}^*\ (\Pi + p) - \frac{\partial J}{\partial Y}\| \ \leq \ 0.99\ \|\frac{\partial \Psi}{\partial Y}^*\ \Pi - \frac{\partial J}{\partial Y}\|$$

- **Assumption 2 :** Both quasi-Newton steps are as complex as a few equations assemblies.

- **Assumption 3 (Wolfe condition):** $\forall\ u_0$, after k_1 partial resolution steps, $g(Y_k, u_0, \Pi_k)$ is close enough to the gradient to satisfy a "good" descent direction:

  $$\exists\ k_1, \exists\ \rho_{max} \quad / \quad \forall\ \rho, \ \ 0 < \ \rho \ \leq \ \rho_{max} \quad \text{and} \quad \forall\ k \geq k_1 \ \ ,$$

  $$J(Y_{k+k'}, u_0 \ - \ \rho W_k^{-1} g(Y_k, u_0, \Pi_k)) \ - \ J(Y_k, u_0) \ \leq \ -0.95\rho\ \|j'(u_0)\|^2$$

  where $\Pi = (\frac{\partial \Psi}{\partial Y})^{-T} \frac{\partial J}{\partial Y}$ and $j'(u) \ = \ \frac{\partial J}{\partial u}(x) \ - \ < \Pi\ ,\ \frac{\partial \Psi}{\partial u}(x) >$.

Note that **Assumption 1** is a convergence assumption concerning the pseudo-Newton iteration. **Assumption 2** is a natural consequence of the smoothness of state equation and cost function. The values of the positive constants .99 and .95 is not important as far as they are smaller than unity. For **Assumption 3** we observe that for $k \to +\infty$, Y_k converges towards the solution of the state equation, Π_k converges towards the solution of the adjoint state equation and then g is the gradient of the functional j, being $j : u \to J(Y(u), u)$. **Assumption 3** is a smoothness assumption and is a consequence of assumption 1 if the functional J and the operator Y are smooth enough.

Main algorithm branchings

In transforming a SQP algorithm into the one-shot extension, the way the constraint is managed has to be deeply reconsidered.

- In SQP, a step can be refused (or reduced) by the Trust Region heuristics, essentially because the Wolfe condition is not sufficiently satisfied. In that case, the state variable is reset to its previous value, and CPU effort spent in restoring it is lost.
- In one-shot, CPU effort for advancing the state variables should not be wasted. When state iteration is not sufficient for satisfying the Wolfe condition, we propose to suppress the minimization update, but not the pre-restoration, which, instead, is re-iterated repeatedly until a minimisation step satisfying the Wolfe condition is obtained (and accepted).

We also have to control the level of resolution of the state system in minimization. In SQP, the correction does not increase the state equation residual, as long as the global minimisation step is small, thanks to the third system to solve. This property is controlled by a penalty mechanism. Since, in our new algorithm, the third linear system is replaced by progressive restorations (in line search,...), we propose to replace the penalty heuristics by a direct control of the residual norm by imposing that in the minimization step, including line search, the residual norm is always lower than the one obtained in the pre-restoration phase.

Global view of the algorithm

A. Initialisation

Initialize x_0, Δ_0 and μ_0. $(nstep = 1)$

For k=0,...

Let $\|G_k\|_\infty = Max(\|\frac{\partial J}{\partial Y} - \left(\frac{\partial \Psi}{\partial Y}\right)^T \Pi\|\ ,\ \|\frac{\partial J}{\partial u} - \left(\frac{\partial \Psi}{\partial u}\right)^T \Pi\|\ ,\ \|\Psi(x)\|)$.

If $\|G_k\|_\infty < \varepsilon$ then stop.

Compute the cost function J.

State restoration phase:

- Compute a Pseudo-Newton step $v_Y = -S(Y,u)\Psi(Y,u)$.
- Update the state Y: $Y^+ = Y + v_Y$.

Adjoint restoration phase:

- Compute a Pseudo-Newton step:
  $v_\Pi = -S(Y^+,u)^T \left(\left(\frac{\partial \Psi}{\partial Y}(Y^+,u)\right)^T \Pi - \frac{\partial J}{\partial Y}(Y^+,u)\right)$.
- Update the adjoint Π: $\Pi^+ = \Pi + v_\Pi$.
- Let $u^+ = u$.

Compute the gradient $g^+ \equiv g(x^+, \Pi^+) = g(Y^+, u^+, \Pi^+)$

Apply a BFGS or Conjugate Gradient acceleration

Minimization phase:

1. Compute the step δu using a line search and solving the state equation at same level as pre-restoration.
2. If the Wolfe condition is satisfied,
   - update the control $u' = u + \delta u$,
   - update the state Y: $Y' = Y^+ + \delta Y$,
   - otherwise reduce the linear search step.
3. If the Wolfe condition is not satisfied, the minimisation step is invalidated. Then go to the beginning of first restoration.

Adjoint post restoration phase (without Trust Region):

$$- \delta\Pi = -S(Y', u')^T \left(\left(\frac{\partial \Psi}{\partial Y}(Y', u') \right)^T \Pi - \frac{\partial J}{\partial Y}(Y', u') \right). \qquad (nstep)$$

Update the adjoint Π: $\Pi^+ = \Pi + v_\Pi$.

B. Some properties of the algorithm

We have not a proof of the convergence of the algorithm. However some simple properties can be stated. Let us assume that on a particular example, this algorithm has a fixed point, that is

$$u_n \to u \quad , \quad Y_n \to Y \quad , \quad \Pi_n \to \Pi.$$

Then this fixed point has the following properties:

- it is a point where state and adjoint are completely solved since the control is tending to a fixed value and thus the state and adjoint are necessarily progressively solved in the proposed iteration,
- it is not a point where the gradient is not a descent direction. Otherwise, the minimization step would not be validated and convergence of state and adjoint would be pushed further, then by assumption (2) the gradient would become a descent direction.
- it is not a point where the step length of the linear search tends to zero, since the pseudo-gradient satisfies the Wolfe condition.

Then the only possible scenario is that Y_n and Π_n converge to state and adjoint solutions and that u_n converges towards a solution of $j' = 0$.

Another evident property of the algorithm is that (u_n, Y_n, Π_n) is a sequence that makes the functional $J(u_n, Y_n, \Pi_n)$ decreases.

The cost of the new algorithm is now much lower since at each iteration we have just to assemble equations and to apply the pseudo-Newton steps, a procedure that is of similar complexity, accounting for Assumption 2. We estimate that the new algorithm is able to reach an optimal complexity of $N_u + 2N_Y$, or in other words, that the solution is reached with an effort of k times a single state system resolution, k being independent from the number of unknowns n.

2.3 Application to the shape design problem

We have applied both algorithms to a demonstrator for optimization of optimal aerodynamical shapes described in [41, 17]. This demonstrator involves the computation of a compressible supersonic flow around a parametrized geometry. A functional is also introduced for evaluating aerodynamical performances and sonic boom strength. The parameters are the positions of all the mesh vertices lying on the shape. Since these parameters are numerous, the modern approach of adjoint is applied. The adjoint software has

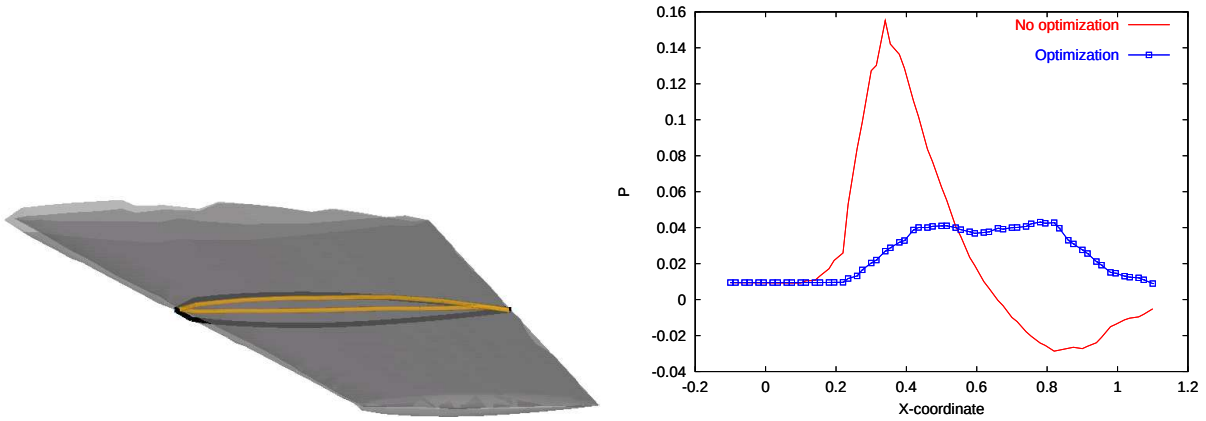

Figure 2: ONERA M6 optimization. Left, initial (black) and optimized (grey) shapes. Right, pressure along a line below the wing.

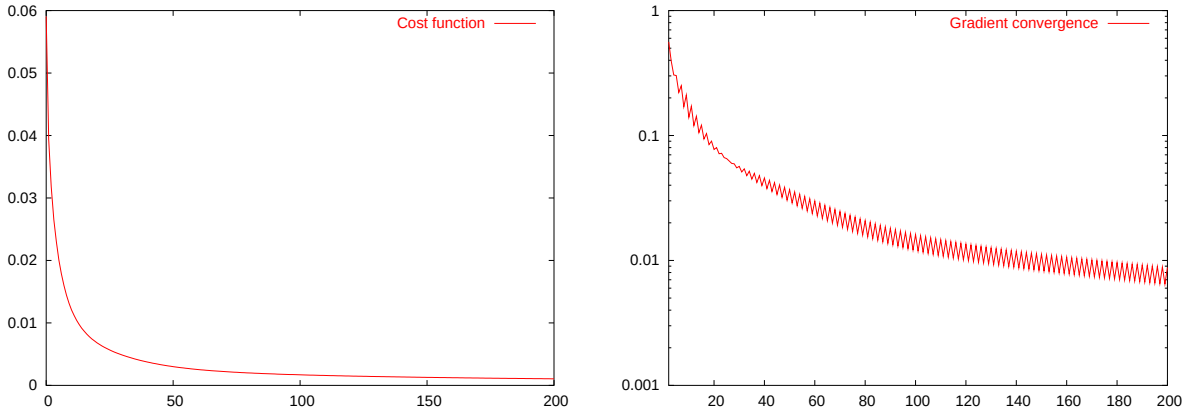

Figure 3: Application to an optimal shape design problem (780 parameters, 11,000 state variables). Figure displays the convergence of a traditional SQP method, with perfect resolution of all linear systems

been built from state routines with the help of reverse Automated Differentiation of assembly loop, we refer to [11, 32]. The Euler state equation is solved with an implicit pseudo-time advancing with at each time step the evaluation of first-order Jacobian matrix and the partial resolution of a linear system with a few tens of Jacobi sweeps. In the experiment presented here, we restrict to the "first-order accuracy" option for which state and adjoint resolutions are much less CPU consuming (by a factor 4-5). In the initial software, a conjugate gradient was applied for optimization. We concentrate on the case of an optimization starting from a ONERA-M6 wing, and with a coarse geometry of 780 shape parameters (11,000 flow variables). In that case the cost functional was decreased from 0.06 to less than 0.01. The optimised shape is quite different from the initial one, see Fig.2, and the reduction of the resulting bow shock is demonstrated by a cut of the pressure field below the wing, see Fig.2.

The protocole for our experiments is to keep exactly the specification of test case, while replacing the initial conjugate gradient optimizer by (1) a version of Byrd-Omojokun SQP algorithm involving a complete solution of state and adjoint systems, and (2) the one-shot method described here.

Coarse mesh results

Firstly, we present a test with the coarse grid of 780 parameters. More realistic experiments would lead to much larger number of parameters and would render the use of an usual BFGS not possible for memory storage reasons. So we have equipped the SQP algorithms with a nonlinear conjugate gradient of Polak-Ribière type, as used in the original non-SQP gradient loop of [41].

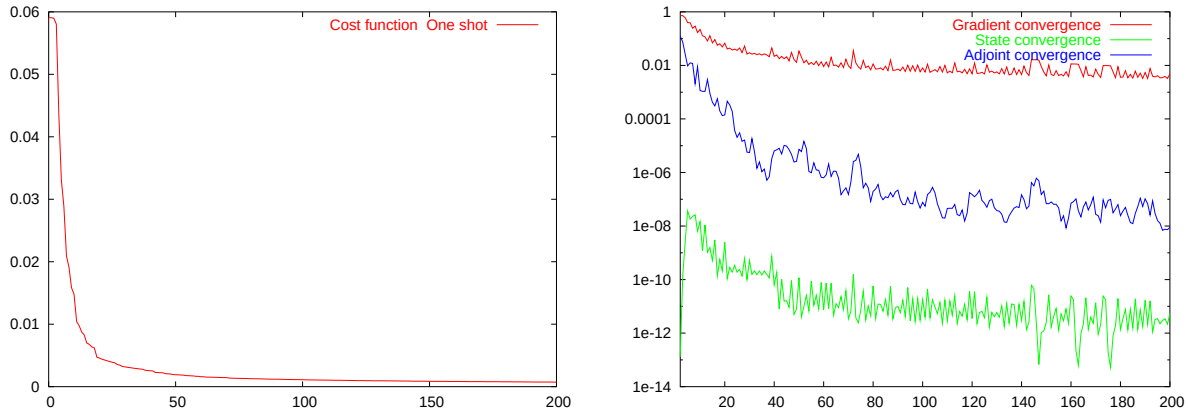

Figure 4: Application to an optimal shape design problem (780 parameters, 11,000 state variables). One-shot optimiser. Left figure displays the convergence of cost functional. Right figure shows the convergence of: functional gradient (upper curve), state residual (lower curve), and in between, adjoint state residual.

However, with one of our examples, we could try the BFGS option. Unfortunately, this option proved to be less efficient than the old conjugate gradient. We have identified the possible explanation: the so-called "curvature condition", is implemented as a necessary condition before allowing successive gradients to contribute to the approximate BFGS Hessian computation; unfortunately, for this test case, this condition was very rarely satisfyied in the first hundred iterations.

The results we present are obtained with the nonlinear conjugate gradient model inside the SQP algorithm. The application of the Byrd-Omojokun SQP, with complete solution (down to round-off error) of state and adjoint is a mess from the CPU standpoint. Total CPU for 200 iterations is 14 hours of a 1GHz workstation. One state solution consists of about 40 time steps, each including one matrix evaluation and 40 Jacobi sweeps. This results in about 100 seconds CPU time. The adjoint state is only about 100-200 Jacobi sweeps and $5-10$ seconds because we have fixed the accuracy order of the whole numerical CFD model to first-order in order to reduce the cost of the test. With 200 iterations, see Fig.3, the gradient of functional is decreased by two orders of magnitude. On the cost functional level, it appears that the 100 last iterations are useful only to decrease it of one extra %. For practical applications, a few iterations, typically less than 50 would be enough.

For the one shot option, state system partial resolution involves only one time step with one matrix evaluation and 20 Jacobi sweeps. The adjoint system involves only 20 Jacobi sweeps. We present the result of 200 iterations, for a total CPU of 40 minutes. The gradient decreases at about the same rate as for the case of SQP (Fig.4). In practice, 50-100 steps ($10-20$ minutes) will be enough for decreasing significantly the cost.

Since a complete state system resolution is 1.5 minutes, the optimal solution was obtained for a cost of about 15 state plus costate solutions, a figure 40 times smaller than for the SQP option.

Fine mesh results

The same method is applied to a finer geometry with 3222 shape parameters and 77,315 state variables. For this case, a single resolution of flow takes 23 minutes. One hundred optimization iterations with the SQP method took 44 hours CPU.

With the one-shot algorithm, convergence rates (Fig. 6) are remarkably close to both those of SQP computation and those of the coarse mesh. In practice, 100 steps (6 hours) will be enough for decreasing significantly the cost. That is again for a cost of about 15 state plus costate solutions.

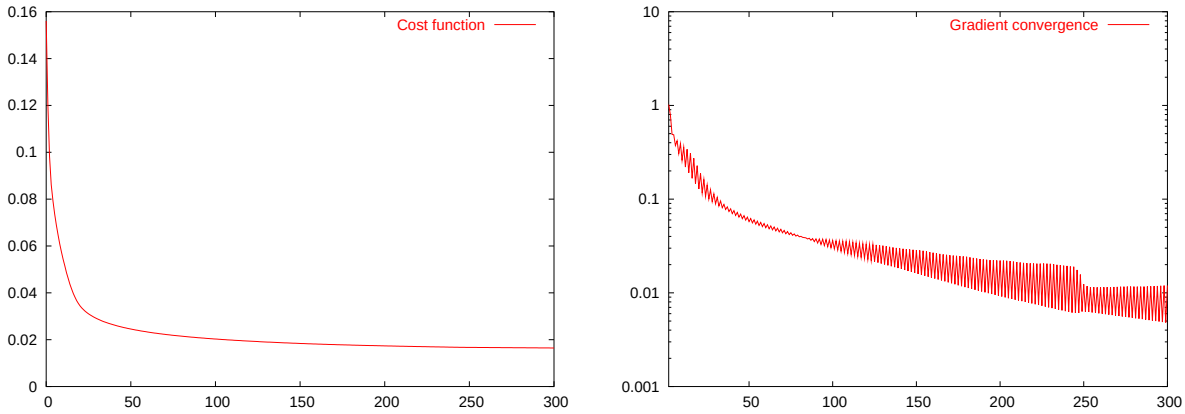

Figure 5: Application to an optimal shape design problem (3222 parameters, 77,315 state variables). SQP optimiser. Cost functional (left) and functional gradient (right) convergence.

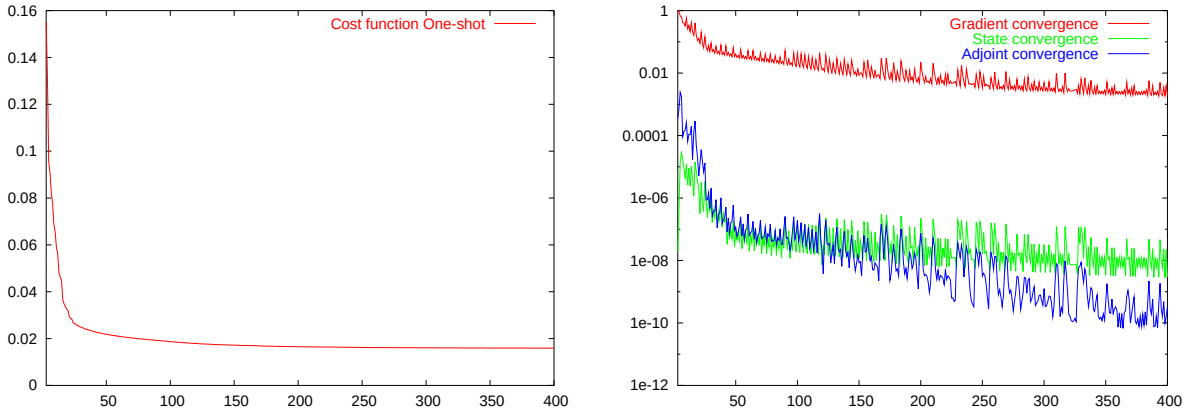

Figure 6: Application to an optimal shape design problem (3222 parameters, 77,315 state variables). One-shot optimiser. Upper figure displays the convergence of cost functional convergence. Lower figure shows the convergence of: functional gradient (upper curve) state residual convergence (lower curve, for iterations 1 to 100), and in between, adjoint state residual convergence.

2.4 Multi-level

Large scale problems coming from Partial Differential Equations generally result in a poor conditioning that further degrades when 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, as exposed next.

2.4.1 Hadamard formula and functional preconditioning

We seek to minimize a functional $j(\gamma)$ with respect to a "shape parameter" γ. 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 of R^d is parameterized by a displacement $\gamma \in \mathcal{C}^{l+\alpha}(\partial\Omega_0)$ of the boundary in direction $\vec{n}_0$.

Suppose that the state equation is the Poisson problem:

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

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

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

A classical difficulty for computing the gradient of j comes from the variable domain in the state equation. We choose a family of diffeomorphisms $(T_\gamma)_\gamma$ such that:

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

$$T_\gamma \;\; is\, identity \;\; on \;\; D \;.$$

Then it is possible to adapt some well known results related to Garabedian's "interior variation" method (see [21], [33],[37], [16],[20]) and to show, under regularity assumptions on f and $\partial\Omega_0$, that the mapping:

$$\gamma \rightarrow z(\gamma)|_D$$

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

In fact, the gradient of this mapping is the solution of a Dirichlet problem with a non-homogeneous Dirichlet condition expressed as a function of the normal derivative of $z(\gamma)$ and of the boundary perturbation. We recognize in the mapping:

$$\gamma \rightarrow \frac{\partial z}{\partial n}(\gamma)|_{\partial\Omega_\gamma}$$

a pseudo-differential operator close to the Dirichlet-Neumann classical one. By the chain rule j is also differentiable and its gradient is expressed as follows:

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

where $\vec{n}_\gamma$ is the normal to $\partial\Omega_\gamma$, and $p(\gamma)$ the following adjoint state:

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

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

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

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 $\mathcal{C}^{l-1+\alpha}(\partial\Omega_0)$. One degree of regularity is lost at each iteration, resulting in a total breakdown of the iterative process after a few steps.

A possible solution to this drawback would be to precondition the iteration:

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

The purpose of the self-adjoint invertible operator B is to recover the degree of regularity lost by the L^2 gradient. It is a functional preconditioner similar to the one introduced in the Least Square formulation of [6].

The main motivation here is that the continuous algorithm has a convergence 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. From a different view point, recall 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 $(\frac{1}{\Delta x})^p$: the smaller the 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 gradient iteration. This means that the descent direction to be used can be computed by solving the following system set on $\partial\Omega_0$:

$$G \;=\; B^{LB} g_{L^2} \;\; \textit{where} \;\; -\Delta^{LB}\, G \;+\; G \;=\; g_{L^2}$$

in which Δ^{LB} is the Laplace-Beltrami operator. The resulting preconditioned iteration is the one proposed in [38][32]. Since the Laplace-Beltrami operator is of degree 2, the gain in regularity with this preconditioner is $+2$, that is not optimal since our analysis shows that the right gain is 1. □

The example of this section shows that in complex shape optimization problems, the direct application of a gradient method may result in non-regular continuous iterations, and then may induce in the discrete version low convergences, in a similar way to unpreconditioned Jacobi-type elliptic solvers.

2.4.2 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. An extremely rich literature exists on this topic. The hierarchical basis method was first analysed by Yserentant in his pioneering paper [46]. The work of Yserentant was apparently motivated by the famous unpublished technical report of Bank and Dupont [2]. A more complete theory was proposed by Bramble, Pasciak and Xu [5], [44]. See also the wavelet extensions, for example in [8]. An extended theory can be found in [26, 45]. The purpose of this section is to recall some results of [45] [15] 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 $V = H^1_0(\Omega)$ 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:

$$B_a \;=\; \sum_{k=0}^{+\infty} \left(\frac{1}{2^a}\right)^k \; (Q_k - Q_{k-1}) \; . \tag{8}$$

Let u be an element of $H_0^s(\Omega)$ and s such that $-3/2 \;\leq\; s \;\leq\; 3/2$. Consider then a theorem introduced in [15]:

Theorem 1
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

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

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.

2.4.3 Preconditioning by node agglomeration

The usual BPX preconditioner essentially needs embedded meshes that are not compatible with today's engineering applications, since these applications rely often on unstructured fine meshes. The adaptation of BPX to arbitrary fine meshes was firstly proposed in [34], which turns to be a rather straightforward way to adapt it to optimum problems. We present now the main lines of this procedure. 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.

Multidimensional Agglomeration

We start from a fine triangulation or tetrahedrization, with nodes 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 with the following weighted scalar product:

$$\forall \; u_h \; and \; v_h \in E_h \quad (u_h, v_h)_h = \sum_i^{n_h} (u_h)_i \; (v_h)_i \; \mathcal{M}eas \; (i) \tag{9}$$

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

$$I^f = I_1 \cup \cdots \cup I_J \cup \cdots \cup I_{n_{2h}} \quad (n_{2h} \ll n_h) \tag{10}$$

where any I_J involves the indices of a few neighbouring nodes. An algorithm for building such a partition can be found in [27]. For any I_J, a coarser basis 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

$$\forall u_{2h} \in E_{2h} \quad \overline{P} u_{2h} = u_{2h} \;\in\; E_h \; . \tag{11}$$

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 9 by:

$$(\overline{P}^* u_h)_J = \frac{\sum_{j_m \subset J}^{n_h} (u_h)_{j_m} \mathcal{M}eas(j_m)}{\mathcal{M}eas(J)} \tag{12}$$

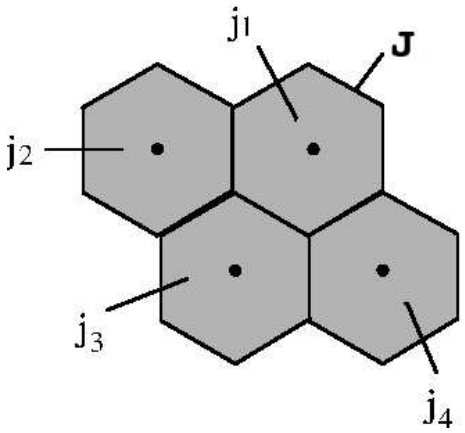

Figure 7: Sketch of the agglomeration of four fine (2D) cells, $C_{j1}, C_{j2}, C_{j3}, C_{j4}$ into a coarser one $\bar{C}_J = C_{j1} \cup C_{j2} \cup C_{j3} \cup C_{j4}$

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 .

In contrast with the usual Galerkin nested finite element sequence, the previous sequence of spaces generated with the transfers $\bar{P}$ and $\bar{P}^*$ does not have enough regularity for Sec.3 theory. 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 [24].

Let us then define **smoother transfers**. The necessary smoothness is recovered by introducing a "smoothing operator" L and its adjoint L^* with respect to the discrete L^2 scalar product. This smoothing operator is an average between a node and its neighbors. It writes:

$$(Lu)_i = (1-\theta)u_i + \theta \frac{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathcal{M}eas(j)\ u_j}{\sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathcal{M}eas(j)} \tag{13}$$

where $\mathcal{N}(i)$ represents the set of neighbors of cell i and θ is the smoothing parameter. The adjoint L^* of L is defined by:

$$(L^*u)_i = (1-\theta)u_i + \theta \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{u_j\ \mathcal{M}eas(j)}{\sum_{k \in \mathcal{N}(j) \cup \{j\}} \mathcal{M}eas(k)}\ . \tag{14}$$

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

$$V_k \quad = \quad L_k\ \bar{P}_k\ (V_{k+1}) \tag{15}$$

Starting from the initial fine space that we denote V_N, we define the analog of the BPX projection operator (2.4.2) to the directly coarser level as follows:

$$Q_N \quad = \quad L_N\ \bar{P}_N\ \bar{P}_N^*\ L_N^* \tag{16}$$

and for any coarser level:

$$Q_k \quad = \quad \Pi_{i=k+1}^{i=N} L_i\ \bar{P}_i\ \Pi_{j=N}^{j=k+1}\ \bar{P}_j^*\ L_j^*. \tag{17}$$

>From this we get:

$$B_a \quad = \quad \sum_{k=0}^{N} \left(\left(\frac{1}{2^a} \right)^k \quad - \quad \left(\frac{1}{2^a} \right)^{k-1} \right) Q_k. \tag{18}$$

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

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

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

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

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

Agglomeration for a surface in 3D

We return to the notation of the previous sections. Let Σ_0 be the tesselated initial 3D surface, made of triangles. The generic discrete surface Σ_γ is defined by a γ-length translation of the vertices of Σ_0 along a vector $\vec{n}$, which is (approximately) normal to Σ_0 and defined at each vertex:

$$\vec{x}_i^\gamma \text{ is a vertex of } \Sigma_\gamma \quad \Leftrightarrow \quad \vec{x}_i^\gamma = \vec{x}_i^o + \gamma(i)\ \vec{n}_i. \tag{19}$$

Here, i is the index of the vertex and $\vec{x}_i^o$ 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 (10, 11, 12), 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:

$$(L\ \vec{x})_i = (1-\theta)\vec{x}_i + \theta\ \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}\vec{x}_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}} \tag{20}$$

where w_{ij} are the weights defined by :

$$w_{ij} = \max\ (\mathcal{A}rea(i)\vec{n}_i \cdot \mathcal{A}rea(j)\vec{n}_j\ ,\ 0) \quad \|\vec{n}_i\| = 1\ \ \forall\ i\ . \tag{21}$$

With this formula, the smoothing is cancelled on geometry sharp angles (e.g. on wing trailing edges). The rest of preconditioner definition is the same as in previous section. The smoothing parameter θ is set to $\frac{1}{2}$ according to the analysis of [30].

Let us focus on the adaptation of the proposed multi-level preconditioner to an optimal shape design loop.

2.5 Application to the shape design problem

As explained above, gradient methods in optimal shape design need functional preconditioning, due to a loss of regularity in the gradient. In the case of the Euler equations, the loss of derivative, can be derived in a formal way from Hadamard-type formulas, see [42], being equal to 1. To test the proposed ideas, we consider again a shape optimization example for an ONERA M6 wing, checking several options. The question is to obtain it in the most efficient way, and in particular to analyse the impact of our preconditioner.

Next, we show that the optimal parameter $a = 1$ predicted by the theory can be numerically verified. We first study this with a coarse mesh discretized geometry, involving 2203 nodes for the 3D mesh, but

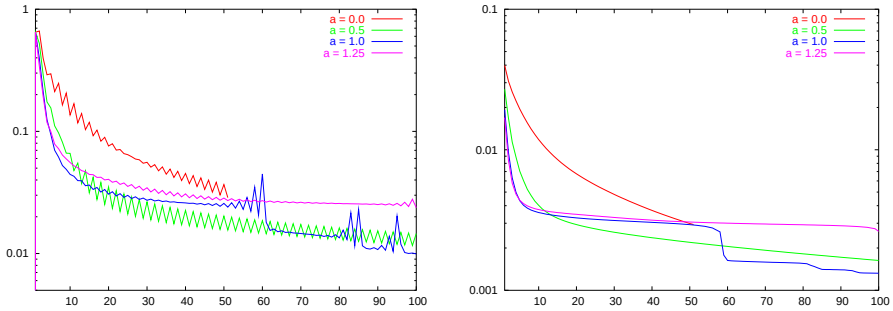

Figure 8: ONERA M6 shape optimization with a gradient method: Left, Gradient convergence. Right, Cost function, ns = 2203

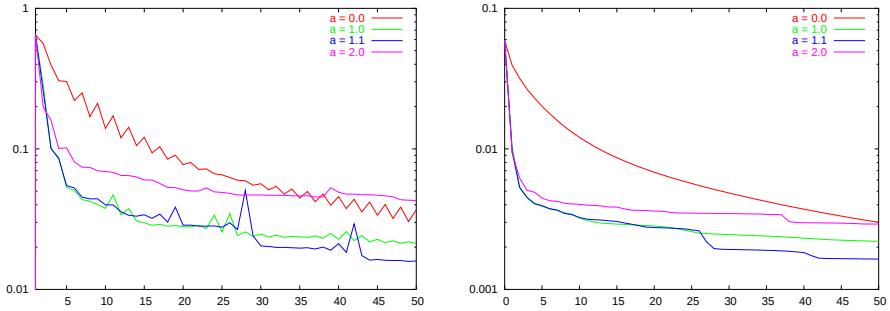

Figure 9: ONERA M6 shape optimization with a conjugate gradient method: Left, Gradient convergence. Right, Cost function, ns = 2203

yet 780 shape parameters. Two optimization algorithms are applied, a gradient (Fig.8) and a conjugate gradient (Fig.9). The 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 optimization 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.

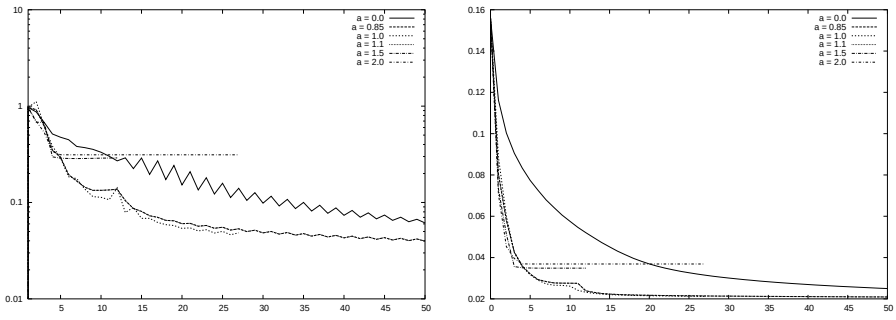

Figure 10: ONERA M6 shape optimization with a gradient method: Left, Gradient convergence. Right, Cost function, ns = 77315

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, shown in Fig.10 and Fig.11. 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. Indeed, it is

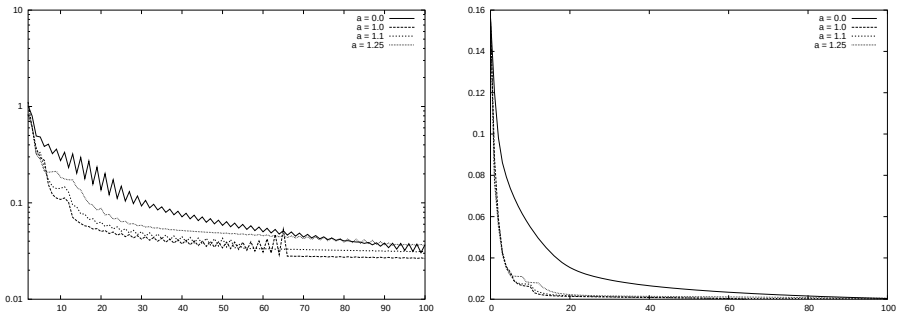

Figure 11: ONERA M6 shape optimization with a conjugate gradient method: Upper Gradient convergence, Lower Cost function, ns = 77315

equivalent to a Laplace-Beltrami smoothing. All the above results show that that this option is not bad, although not as good as the optimal one.

We complete the above convergence curves by Tabs. 2.5 and 2.5 which compare the efficiency of the preconditioned algorithm to the standard one in the practical case where convergence of optimization 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 [42] the proposed preconditioner is applied to more complex geometries.

3 NUMERICAL ERROR REDUCTION

In this section, we apply the control theory derived formalism to mesh adaption with the target of reducing the numerical error in discretized problem solutions.

3.1 A few remarks : adaptive convergence, barriers

Let us assume that we have computed a sequence of adaptive solutions $(\mathcal{M}_N, U_N)$ where the total number of nodes N belongs to a sequence of positive integers tending to infinity. Symbol U_N denotes a numerical solution of a certain state equation computed on the mesh $\mathcal{M}_N$, and $\mathcal{M}_N$ is a mesh adapted to the solution U_N . Then the mesh convergence order α can be expressed in terms of N and of the dimension of geometrical space d (here $d = 1$):

$$\epsilon \leq C \; N^{\alpha/d},$$

where ϵ is the error upper bound and C is a constant. In the case of a discontinuous function u, sequences of uniform meshes cannot show an order better than one half. A higher order convergence property of adaptive strategies is asserted by the following lemma:

Lemma 1
Consider an interpolator of real-valued functions of one variable, with theoretical order of accuracy α. Let u a piecewise regular function with one discontinuity. Then, uniform refinement convergence can be of order at most 1/2 in L^2. On the contrary, when an **adaptive strategy** *is applied, the convergence order can be as high as α.*

In short, an adaptive strategy can force the successive meshes to be $2^{(\alpha+1)}$ finer near the discontinuity while just twice finer in regular parts. Then the global error is 2^α times smaller, while the number of nodes is essentially doubled. It follows that the method is of order α.

3.1.1 Early capturing

Many authors have noted that in the case of smooth functions with locally high gradients, uniform refinement may show half order convergence until a great number of nodes are considered in such a way that local mesh size is small enough for the good representation of the local high gradients. Only then, the higher order convergence appears. See for example [36]. In contrast, the number of nodes necessary for adaptive methods to show higher order convergence is much smaller number. We call this property "early capturing".

In order to illustrate this property, we will apply a mesh adaptation stategy that is a rather popular combination of Hessian evaluation and anisotropic mesh reconstruction with control on the mesh size [7, 4]. Iteration between PDE solution and mesh adaptation is applied for a fixed number N of nodes. Next, we study the capture of a boundary layer in a supersonic turbulent flow past a flat plate. The flow model is $k - \varepsilon$ with a wall law parameterized in such a way that only the fully turbulent part of the boundary layer is captured by the mesh. We consider the friction velocity on a point on the plate after the stagnation point. A very fine and accurate adapted anisotropic computation with 200,000 nodes plays the role of the reference exact solution.

A series of uniform refinements with meshes as large as $100,000$ nodes failed to produce a measurable convergence to solution. We compare then, as shown in Fig. 12, two adaptive strategies: an isotropic one in which triangles are not too different from equilateral, and an anisotropic one in which stretching is applied. We observe that the isotropic strategy allows early capturing, i.e. second-order convergence with a rather small number of nodes (meshes finer than 15,000 nodes). On the other hand, the anisotropic strategy gives second-order convergence already for 6,000 nodes. Other examples are presented in [29].

3.1.2 Barriers for refinement strategies

These results point out the remarkable difference of performances between isotropic and anisotropic mesh adaptation. In some cases, it is possible to quantify this difference, thanks to the statement of a theoretical accuracy upper limit or barrier for isotropic mesh adaptation methods:

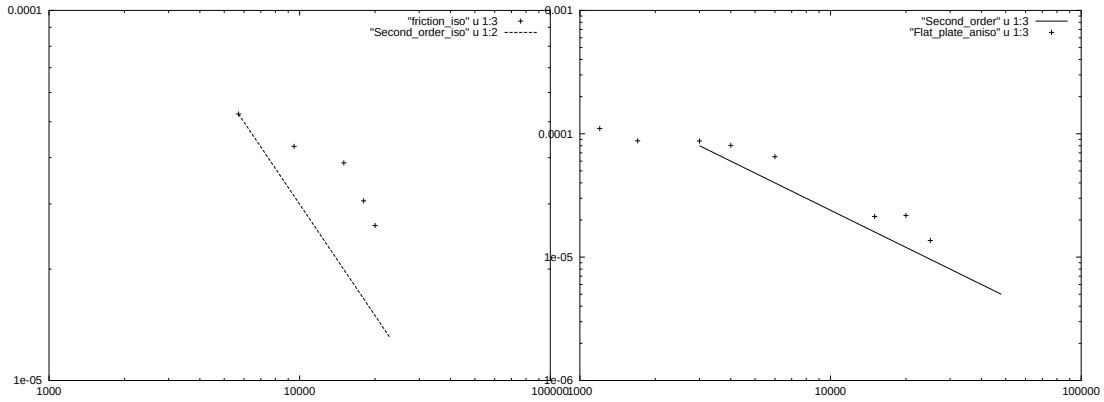

Figure 12: Mesh convergence of friction velocity for a flat plate computation with anisotropic adaptation. Number of nodes is read on x-axis, L^2 approximation error on y-axis. Dash line correspond to second-order convergence. On left side, crosses correspond to a series of adapted isotropic computations, and second-order convergence is observed only for mesh sizes larger than 10000. On right side, crosses correspond to a series of adapted anisotropic computations, and second-order convergence is already observed for mesh sizes larger than 3000.

Lemma 2
The order of convergence α in L^p of an isotropic adaptive mesh method [9], applied in dimension d to $\mathcal{P}_1$-interpolate a discontinuous function is bounded by:

$$\alpha \leq \frac{d/p}{d-1}.$$

Therefore, isotropic mesh refinement is a rather inefficient procedure for interpolating discontinuous functions: the error estimate with order 3/4 for isotropic 3D adaptation is a severe limit. Many engineers and scientists remarked that the price of isotropic mesh adaptation such as AMR (Adaptive Mesh refinement) is reasonable in 2D, but prohibitive in 3D, as soon as large 2D surfaces of discontinuity are involved. Our conjecture -and numerical experience- is that the above limit does not apply to the best anisotropic mesh adaptation methods.

3.2 Metric and interpolation

3.2.1 Continuous metric in an interval

In the one-dimensional case, a mesh is uniquely specified by a metric. After some definitions concerning the metric, we recall an estimate of the interpolation error and then show how an optimal metric can be derived.

A **metric** on a given set allows to define the distance between two of each elements. We shall call a metric on the interval $[a, b]$ a (strictly) positive continuous function $\mathcal{M} : x \to \mathcal{M}(x)$ defined on $[a, b]$. It specifies, for any c and d within this interval, the length of segment cd as follows:

$$L_{\mathcal{M}}(cd) = \int_c^d \sqrt{\mathcal{M}}(s) \ \ ds \tag{22}$$

Let us consider a mesh of interval $[a, b]$ with N nodes. It is a subdivision $x_0 = a < x_1, ..., x_i < x_{i+1}, ..., x_{N-1} < x_N$ b of this interval. A consequence of the above definition is that a metric can prescribe a particular class of meshes. Indeed, we shall say that a mesh conforms to metric $\mathcal{M}$ if and only if the following relation for the unitary element length holds:

$$\int_{x_i}^{x_{i+1}} \sqrt{\mathcal{M}} dx = 1, \quad \text{for any element} \quad [x_i, x_{i+1}]$$

Now if we introduce the local continuous mesh size $m_{\mathcal{M}} = \mathcal{M}^{-1/2}$ we have:

$$\int_{x_i}^{x_{i+1}} \frac{1}{m_{\mathcal{M}}}\, dx = 1, \quad \text{for any element } [x_i, x_{i+1}]$$

which shows that when $m_{\mathcal{M}}$ is a constant function, it is nothing other than the element size.

Another way to understand this is to introduce the local continuous node density $d = 1/m$:

$$\int_{x_i}^{x_{i+1}} d\, dx = 1, \quad \text{for any element } [x_i, x_{i+1}]$$

It can be verified that the number of nodes (or equivalently intervals) of the mesh is specified by the metric. It is given by:

$$C(\mathcal{M}) = \int_a^b \sqrt{\mathcal{M}} dx = \int_a^b 1/m dx = \int_a^b d\, dx. \tag{23}$$

If $C(\mathcal{M})$ is a positive integer, only one mesh is decribed by it, if $C(\mathcal{M})$ is not an integer, no mesh is described by it.

3.2.2 Interpolation error

We propose now to model the error related to a given metric. Since the metric determines the local mesh size, it also determines the interpolation error resulting of the application of an interpolator on the mesh. We can model this error as a continous function of the metric.

Let us consider :

- A uniform mesh, i.e.: $x_0 < x_1 < ... < x_N$ with $x_i = x_0 + \frac{i}{N-1}(x_N - x_0)$,
- A function u, regular enough, defined on a segment $[a, b]$, $a = x_i, b = x_{i+1}$,
- h is not necessarily small,
- $\Pi_h u$ the $\mathcal{P}_1$ interpolation of u on every interval $[a, b]$ of the mesh.
- Assuming $\Pi_h u(a) = u(a)$ and $\Pi_h u(b) = u(b)$, the approximation error is defined by:

$$e = u - \Pi_h u. \tag{24}$$

It is shown in [12] that there exists a bounded number M such that $\forall \xi \in I$:

$$|e(\xi)| = |(u - \Pi_h u)(\xi)| \leq \frac{(b-a)^2}{8} M. \tag{25}$$

This estimate, after being modelled in terms of continuous functions, will contribute to the continuous problem statement.

3.2.3 Optimal metric: Optimality condition for norm L^α

We can now *model* the error resulting from using a $\mathcal{P}_1$ interpolation on a mesh respecting a given metric $\mathcal{M}$. The analysis of previous section suggests (neglecting higher-order terms) to take the following model:

$$|e_{\mathcal{M}}(x)| = (d_{\mathcal{M}}(x))^{-2} |u''(x)|. \tag{26}$$

where $d_{\mathcal{M}}(x)$ is the node density of the mesh, or equivalently the inverse local mesh size, i.e. the inverse of $m_{\mathcal{M}}(x)$. We want now to find the minimum with respect to metric $\mathcal{M}$ of the L^α $(0 < \alpha < \infty)$ norm of the error $e_{\mathcal{M}}$:

$$\min_{\mathcal{M}} \; (|e_{\mathcal{M}}(x)|)_{L^\alpha}^\alpha = \min_{\mathcal{M} \in K} \; \frac{1}{2} \int_a^b (d_{\mathcal{M}}(x)^{-2} |u''(x)|)^\alpha ds. \tag{27}$$

In order to get a non-trivial infinitely fine solution, the space of admissible metrics is chosen in order to prescribe the number of nodes:

$$C(M) = \bar{C}(d) = \int_a^b d(x)dx \ = N. \tag{28}$$

which gives a linear constraint. Additionally, to get (at least formally) an optimality condition, we can derive the functional of (27) with respect to d:

$$-2\alpha \int_a^b d^{-2\alpha-1} \left(|u''|\right)^\alpha \delta d \ ds \geq 0 \, , \, \forall \delta d : \int_a^b \delta d = 0. \tag{29}$$

Thus,

$$d_{opt}(x) = Cte.|u''(x)|^{\frac{\alpha}{2\alpha+1}} \tag{30}$$

and taking into account the constraint $\bar{C}(d) = N$, we get:

$$d_{opt}(x) = \frac{N}{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds} \ |u''(x)|^{\frac{\alpha}{2\alpha+1}}, \tag{31}$$

or in terms of the local mesh size:

$$m_{opt}(x) = \frac{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds}{N} \ |u''(x)|^{\frac{-\alpha}{2\alpha+1}}. \tag{32}$$

The minimum of the functional writes:

$$(\mathcal{E}_\alpha^{opt})^\alpha \ = \ 1/2 \left(\frac{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds}{N} \right)^{2\alpha} \ \int_a^b |u''(s)|^{\frac{\alpha}{2\alpha+1}} ds. \tag{33}$$

Remark 2
For identifying the above formally optimal metric as a rigorous minimum of the initial problem, several questions remain open.
First, it is not clear whether there is a minimum. probably, it is not reasonable to seek it without any smoothness assumption, for example it seems useful to assume that the metric is bounded in a Sobolev space. This would correspond to an assumption on the smoothness of the underlying meshes: no element with vanishing measure, no sudden variation of elements shape and size.
Second, in the case we make an assumption leading to a sufficient compacity of admissible set, the question of uniqueness should be addressed.
And third, a local analysis is possible, the Hessian of the functional is positive, which shows that the formal optimum is at least a local optimum in any smooth enough subset of the set of constraints.

Remark 3
The local mesh size, naturally inversely proportional to the number of nodes, is defined by (32) only if the second derivative u'' never vanishes. In practice, we replace $|u''|$ by $\max(\varepsilon, |u''|)$, with a small positive ε.

3.2.4 Examples

- For the case of L^1 norm: This is a rather classical norm in image processing,

$$m_{opt}(x) \ = \ \frac{\int |u''|^{1/3} ds}{N} \ |u''(x)|^{-1/3} \tag{34}$$

- For the case of the L^2 norm: This is the natural option for PDE's,

$$m_{opt}(x) \ = \ \frac{\int |u''|^{2/5} ds}{N} |u''(x)|^{-2/5} \ . \tag{35}$$

- For the case of L^∞ norm: We can make the exposant tend to infinity and *formally* get:

$$m_{opt}(x) \;=\; \frac{\int |u''|^{1/2} ds}{N} |u''(x)|^{-1/2} \; . \tag{36}$$

Remark 4
In the last case, introducing d_{opt}:

$$d_{opt}(x) = \frac{N}{\int |u''|^{\frac{1}{2}} ds} \; |u''(x)|^{\frac{1}{2}}. \tag{37}$$

in (26) gives a uniform error,

$$|e_{\mathcal{M}}(x)| = (d(x))^{-2} |u''(x)| = \frac{(\int |u''|^{\frac{1}{2}} ds)^2}{N^2}, \tag{38}$$

which is nothing else that the usual error repartition formula. We also remark that the optimal error magnitude is determined by the $\mathcal{L}^{1/2}$ norm of the second derivative.

If we take the α-th root of expression (33), we get:

$$\mathcal{E}_\alpha^{opt} \;=\; 1/2 \left(\frac{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds}{N} \right)^2 \; (\int_0^1 |u''(s)|^{\frac{\alpha}{2\alpha+1}} ds)^{1/\alpha}, \tag{39}$$

and passing to the limit:

$$\mathcal{E}_\infty^{opt} \;=\; \frac{1}{N^2} \left(\int |u''|^{\frac{1}{2}} ds \right)^2 . \tag{40}$$

Remark 5
The power $1/2$ of the second derivative $|u''|$ is related with the accuracy order of the interpolation. The same analysis can be done with a more accurate interpolation, that is typically with an error model of κ-th order:

$$|e_{\mathcal{M}}(x)| \;=\; (d_{\mathcal{M}}(x))^\kappa |u^{(\kappa)}| \; ; \tag{41}$$

in that case the optimal power is $\frac{\alpha}{\kappa\alpha+1}$ and always smaller than $1/\kappa$.

3.2.5 Convergence order of the continuous metric model

Let us examine how to look for an optimal metric in the *discontinuous case*. When a $\mathcal{P}_1$ interpolation is choosen, we model the error as :

$$\int_0^1 |e_{\mathcal{M}}(x)|^\alpha ds = \int_0^1 (m^2 |\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))|)^\alpha ds, \tag{42}$$

where δ is smaller than m. We observe that the differential quotient:

$$\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))$$

either it is close to $\frac{\partial^2 u}{\partial x^2}$ where u is regular (a) or it is of the order of δ^{-2} at singularities of u (b). Moreover, since u is bounded,

$$||\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))||_{L^{1/2}} \;\; is \;\; bounded \;\; independently \;\; of \;\; \delta. \tag{43}$$

The calculus of variations gives now:

$$m_N(x) = Cte.|(|\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))|(x))|^{\frac{-2}{5}}, \tag{44}$$

and the resulting optimal error in L^2 writes:

$$\text{Error} = \frac{2}{N^2} \; \left(\int |\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))|^{\frac{2}{5}} \right)^{\frac{5}{2}} \;\; < \;\; \frac{K}{N^2}$$

where K is a bounded constant, due to (43). We deduce that the adaptive strategy is formally of second-order accuracy.

3.3 The 2D case

In this section, we propose an extended model for the interpolation error and then apply again a variation calculus.

3.3.1 Definition

Main notations:

Let u be a twice continuously differentiable function from a subset Ω of R^2 in R. To any triangulation $\mathcal{T}_h$ of Ω, it corresponds an interpolation $\mathcal{P}^1$ of u that we denote by $\Pi_h u$.

For the local error analysis:

- we consider $K = [a, b, c]$, a triangle whose diameter h_{max} does not tend to zero.
- we consider u, a function from R^2 to R that is $\mathcal{C}^2$,
- we denote by $\Pi_h u$, the linear interpolate of u over K,
- we assume that u and $\Pi_h u$ coincide in a, b and c.

We want to find a good majorant to the error $e = u - \Pi_h u$ on $K = [a, b, c]$.

Hessian matrix $\mathcal{H}$:

The Taylor formula writes:

$$u(\vec{x} + \delta\vec{x}) \quad = \quad u(\vec{x}) + \frac{\partial u}{\partial x_i}(\vec{x})\delta x_i + \frac{1}{2}\frac{\partial^2 u}{\partial x_i \partial x_j}(\vec{x})\delta x_i \delta x_j + \ldots$$

with:

$$\delta\vec{x} = \begin{pmatrix} \delta x \\ \delta y \end{pmatrix} \tag{45}$$

The Hessian of u is denoted by:

$$\mathcal{H} = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial x \partial y} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix} \tag{46}$$

$\mathcal{H}$ is diagonalizable and can be written:

$$\mathcal{H} = \mathcal{R} * \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} * \mathcal{R}^{-1} \tag{47}$$

We introduce the rotation matrix $\mathcal{R}$, which goes from the usual (x, y) coordinate system to a system (ξ, η):

$$\mathcal{H} \equiv \mathcal{R}\widehat{\mathcal{H}}\mathcal{R}^{-1}, \tag{48}$$

meaning that

$$\lambda_1 = \frac{\partial^2 u}{\partial \xi^2} \tag{49}$$

and

$$\lambda_2 = \frac{\partial^2 u}{\partial \eta^2}. \tag{50}$$

3.3.2 Metric $\mathcal{M}$

The family of metrics we shall consider contains a tensor field depending on (x, y) and defined, from two rotations $\mathcal{S}$ and $\mathcal{S}^{-1}$, as follows:

$$\mathcal{M}(x,y) = \mathcal{S}^{-1} \begin{pmatrix} \frac{1}{m_\xi}^2 & 0 \\ 0 & \frac{1}{m_\eta}^2 \end{pmatrix} \mathcal{S}, \tag{51}$$

where $\mathcal{S}, m_\xi$, and m_η depend on x and y. The coefficients m_ξ, and m_η are the local mesh sizes in each of the two directions ξ and η defined by the rotation $\mathcal{R}$. Similarly to the 1D case, the length $L_{\vec{u}}$ of the vector $\vec{u}$ in metric $\mathcal{M}$ is defined as follows:

$$L_{\vec{u}} = \int_0^1 \sqrt{\vec{u}.\mathcal{M}.\vec{u}}\ \ ds. \tag{52}$$

The quantities $\frac{1}{m_\xi}$ and $\frac{1}{m_\eta}$ represent the number of mesh elements by unit length running along axes ξ and η respectively.

3.3.3 Complexity $C(\mathcal{M})$

In a similar way to the 1D case, we associate to a metric $\mathcal{M}$ the complexity or total number of nodes computed from an integral of mesh density, i.e. from ε, m_ξ and m_η, on an area of ε^2 surface.

$$C(\mathcal{M}) = \int_\Omega \sqrt{\mathcal{M}} ds = \int_\Omega \frac{1}{m_\xi}\frac{1}{m_\eta}\ dxdy. \tag{53}$$

Let $d(\xi, \eta)$ be the local density of nodes for the metric $\mathcal{M}$. It is equal to $\frac{1}{m_\xi}.\frac{1}{m_\eta}$.

Lemma 3
If a tensorial mesh with $N+1$ nodes in each direction satisfies ideally the metric $\mathcal{M}$ then :

$$\int_0^1 m_\xi^{-1}.m_\eta^{-1}\,(s)\,ds = \sum_i \int_{x_i}^{x_{i+1}} \int_{y_i}^{y_{i+1}} d\,(s)\,ds = \sum_{i=1}^{N}\sum_{i=1}^{N}(1) = N^2. \tag{54}$$

3.3.4 Minimization of the interpolation error (I)

We consider first the optimal anisotropic metric. The local mesh size is defined as a unique scalar field, $m(x, y)$ or equivalently the node density by area unit $d(x, y) = 1/m^2(x, y)$. The total number of nodes is given by:

$$C(\mathcal{M}) = \int_\Omega d(x,y)\ dxdy. \tag{55}$$

For error modelling, we get inspiration from a rough estimate and write:

$$e_{\mathcal{M}}(x,y)\ = m^2(x,y)M(x,y)\ =\ d^{-1}(x,y)M(x,y)\ . \tag{56}$$

where M, above defined is taken equal to the largest absolute value of eigenmodes of the Hessian of u.

Let us minimize the $\mathcal{L}^\alpha$ norm of this error under the constraint of a number of nodes equal to N:

$$\min_{\mathcal{M}}\ \frac{1}{2}\int_\Omega M^\alpha\ d^{-\alpha}\ dxdy \tag{57}$$
$$\text{under the constraint } C(\mathcal{M}) = N\ .$$

The optimality conditions are:

$$-\,\alpha \int_\Omega M^\alpha\ d^{-\alpha-1}\ \delta d\ dxdy\ \leq\ 0 \tag{58}$$
$$\text{for any } \delta d \text{ such that } \int_\Omega \delta d\ dxdy = 0\ ,$$

or taking into account the constraint,

$$d_{opt}(x) = \frac{N}{\int_\Omega M^{\frac{\alpha}{-\alpha-1}} dxdy}\ M(x,y)^{\frac{\alpha}{\alpha+1}}. \tag{59}$$

Remark 6
Again the case $\alpha = +\infty$ gives $d = M$, that is the error isorepartition.

In terms of the local mesh size m it gives:

$$m_{opt}(x) = \frac{(\int_\Omega M^{\frac{\alpha}{-\alpha-1}} ds)^{1/2}}{N}\ M(x,y)^{\frac{-\alpha}{2\alpha+2}}. \tag{60}$$

In particular for $\alpha = 2$, the exponent in (60) is $1/3$.

3.3.5 Minimization of the interpolation error (II)

Considering an anisotropic family of metrics, we return to the previous general notations.

Optimization problem

We assume that the function u has bounded second derivatives $\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2}$. We admit that the best metric should be chosen among metrics that are aligned with the Hessian of u. We then consider metrics $\mathcal{M}$:

$$\mathcal{M}_{x,y} = \mathcal{R}_{\mathcal{M}}^{-1} \begin{pmatrix} (m_\xi)^{-2} & 0 \\ 0 & (m_\eta)^{-2} \end{pmatrix} \mathcal{R}_{\mathcal{M}} \tag{61}$$

such that

$$\mathcal{R}_{\mathcal{M}} \ = \ \mathcal{R}_u\ , \tag{62}$$

where $\mathcal{R}_u$ is the rotation which diagonalizes the Hessian of u:

$$\mathcal{H}_u = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x . \partial y} \\ \frac{\partial^2 u}{\partial x . \partial y} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix} = \mathcal{R} * \begin{pmatrix} \frac{\partial^2 u}{\partial \xi^2} & 0 \\ 0 & \frac{\partial^2 u}{\partial \eta^2} \end{pmatrix} * \mathcal{R}^{-1} \tag{63}$$

As above, for simplicity, the second derivatives $\frac{\partial^2 u}{\partial \xi^2}$ and $\frac{\partial^2 u}{\partial \eta^2}$ are assumed to be of a strictly positive absolute value.

An anisotropic error estimate given in [13] leads to consider the following *anisotropic error continuous model*:

$$\mathcal{E}_\alpha = \int \left(|\frac{\partial^2 u}{\partial \xi^2}| . m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}| . m_\eta^2 \right)^\alpha \ dxdy\ . \tag{64}$$

The optimal metric minimizes the functional $\mathcal{E}_\alpha$ under the constraint $C(\mathcal{M}) = N^2$:

$$\min_{\mathcal{M}} \int \left(|\frac{\partial^2 u}{\partial \xi^2}| . m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}| . m_\eta^2 \right)^\alpha \ dxdy \tag{65}$$
$$\text{under the constraint} \int {m_\xi}^{-1} {m_\eta}^{-1}\ dxdy = N^2.$$

The optimality system then writes

$$\begin{aligned} \mathcal{E}'_\alpha(\mathcal{M}) \delta\mathcal{M} &= 0, \\ \forall \delta\mathcal{M},\ \ C'(\mathcal{M}_{opt}).\delta\mathcal{M} &= 0. \end{aligned} \tag{66}$$

The second equation can be used for writing a relation between $\mathcal{M}$ and $\mathcal{C}$:

$$\begin{aligned} C'(\mathcal{M}_{opt}).\delta\mathcal{M} &= 0, \\ &\Updownarrow \\ \int \frac{-1}{m_\xi} . \frac{\delta m_\eta}{m_\eta^2} + \frac{-1}{m_\eta} . \frac{\delta m_\xi}{m_\xi^2} &= 0, \\ &\Updownarrow \\ \int \frac{1}{m_\xi} . \delta m_\eta + \frac{1}{m_\eta} . \delta m_\xi &= 0. \end{aligned} \tag{67}$$

Therefore, it can be written that

$$\begin{pmatrix} \delta m_\xi \\ \delta m_\eta \end{pmatrix} = \zeta \begin{pmatrix} -m_\xi \\ m_\eta \end{pmatrix}. \tag{68}$$

Equation (66) will be verified for any pair $(\delta m_\xi, \delta m_\eta)$ such that (68) holds at least for one scalar function ζ of (x, y).

Let us expand equation (66):

$$\int \left(|\frac{\partial^2 u}{\partial \xi^2}| m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}| . m_\eta^2 \right)^{\alpha-1} \left(|\frac{\partial^2 u}{\partial \xi^2}| m_\xi \delta m_\xi + |\frac{\partial^2 u}{\partial \eta^2}| m_\eta \delta m_\eta \right) dxdy = 0. \tag{69}$$

Due to statement (68) we can replace δm_ξ and δm_η :

$$\int \left(\left| |\frac{\partial^2 u}{\partial \xi^2}| m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}| . m_\eta^2 \right|^{\alpha-1} \right) \zeta \left(-|\frac{\partial^2 u}{\partial \xi^2}| m_\xi m_\xi + |\frac{\partial^2 u}{\partial \eta^2}| m_\eta m_\eta \right) dxdy = 0\,. \tag{70}$$

Since m_η, m_ξ and the second derivatives of u do not vanish, this will be zero for any function ζ if

$$|\frac{\partial^2 u}{\partial \xi^2}| . m_\xi^2 = |\frac{\partial^2 u}{\partial \eta^2}| . m_\eta^2. \tag{71}$$

>From which we can derive the ratio between m_ξ and m_η

$$\frac{m_\xi}{m_\eta} = \sqrt{\frac{|\left(\frac{\partial^2 u}{\partial \eta^2}\right)|}{|\left(\frac{\partial^2 u}{\partial \xi^2}\right)|}} \tag{72}$$

In the sequel, it will be simpler to express metric $\mathcal{M}$ in terms of the node density d, number of nodes located in a unit area, and of the local aspect ratio μ:

$$\mathcal{M} = \frac{1}{d}\, \mathcal{R}_{\mathcal{M}}^{-1} \begin{pmatrix} \mu & 0 \\ 0 & \frac{1}{\mu} \end{pmatrix} \mathcal{R}_{\mathcal{M}}\ , \tag{73}$$

More precisely we set $m_\xi = \sqrt{\frac{\mu}{d}}$ and $m_\eta = \sqrt{\frac{1}{\mu d}}$. The constraint (67) then becomes

$$\int \delta d = 0. \tag{74}$$

The condition (66) can now be written

$$\int \left(|\frac{\partial^2 u}{\partial \xi^2}| \frac{\mu}{d} + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu d} \right)^{\alpha-1} \left(|\frac{\partial^2 u}{\partial \xi^2}| (\frac{\delta \mu}{d} - \frac{\mu \delta d}{d^2}) - |\frac{\partial^2 u}{\partial \eta^2}| \frac{d\delta\mu + \mu\delta d}{\mu^2 d^2} \right) = 0,$$
$$\forall \delta d \quad \text{such that} \quad \int \delta d = 0 \quad \text{and} \quad \forall \delta \mu.$$

Let us expand now equation (66) in terms of $\delta\mu$ and δd. First, for $\delta\mu$:

$$\int (*)^{\alpha-1} \left(|\frac{\partial^2 u}{\partial \xi^2}| \frac{1}{d} - |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu^2 d} \right) \delta\mu \ = \ 0 \quad \forall \delta\mu\ , \tag{75}$$

where $(*)$ stands for $|\frac{\partial^2 u}{\partial \xi^2}| \frac{\mu}{d} + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu d}$ and which, by assumption, never vanishes. We deduce that

$$|\frac{\partial^2 u}{\partial \xi^2}| \frac{1}{d} - |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu^2 d} = 0. \tag{76}$$

We get then that

$$\mu = \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|} \right)^{1/2} \tag{77}$$

which is (72).

Second, for δd:

$$\int (*)^{\alpha-1} \left(|\frac{\partial^2 u}{\partial \xi^2}| \frac{-\mu}{d^2} + |\frac{\partial^2 u}{\partial \eta^2}| \frac{-1}{\mu d^2} \right) \delta d \;=\; 0 \; . \tag{78}$$

We get then

$$(*)^{\alpha-1} \frac{1}{d^2} \left(|\frac{\partial^2 u}{\partial \xi^2}|(\mu) + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu} \right) = \text{constant}, \tag{79}$$

or, in other words,

$$\frac{1}{d^{\alpha+1}} \left(|\frac{\partial^2 u}{\partial \xi^2}|(\mu) + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu} \right)^{\alpha} = \text{constant}. \tag{80}$$

Let us replace μ by its value:

$$d^{\alpha+1} = \text{constant} \; \left(|\frac{\partial^2 u}{\partial \xi^2}| \; |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2}} . \tag{81}$$

We thus get

$$d = C_\alpha \; \left(|\frac{\partial^2 u}{\partial \xi^2}| \; |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2\alpha+2}} . \tag{82}$$

Here, the constant C_α is given by:

$$C_\alpha \;=\; \left(\int \left(|\frac{\partial^2 u}{\partial \xi^2}| \; |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2\alpha+2}} \; dxdy \right)^{-1} \; N^2 \; . \tag{83}$$

Finally, the square local mesh sizes are given by

$$\begin{aligned} m_\xi^2 &= C_\alpha^{-1} |\frac{\partial^2 u}{\partial \xi^2}|^{\frac{-2\alpha-1}{2(\alpha+1)}} |\frac{\partial^2 u}{\partial \eta^2}|^{\frac{1}{2(\alpha+1)}} \\ m_\eta^2 &= C_\alpha^{-1} |\frac{\partial^2 u}{\partial \xi^2}|^{\frac{1}{2(\alpha+1)}} |\frac{\partial^2 u}{\partial \eta^2}|^{\frac{-2\alpha-1}{2(\alpha+1)}} \end{aligned} \tag{84}$$

which means that metric $\mathcal{M}_{opt}$ is defined by:

$$\mathcal{M}_{opt} = C_\alpha^{-1} \; \left(|\frac{\partial^2 u}{\partial \xi^2}| \; |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{-\alpha}{2\alpha+2}} \; \mathcal{R}^{-1} \; \begin{pmatrix} \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|} \right)^{1/2} & 0 \\ 0 & \left(\frac{|\frac{\partial^2 u}{\partial \xi^2}|}{|\frac{\partial^2 u}{\partial \eta^2}|} \right)^{1/2} \end{pmatrix} \; \mathcal{R} \; . \tag{85}$$

In the case of the $\mathcal{L}^2$ norm, this becomes:

$$\mathcal{M}_{opt,2} = C_2^{-1} \; \mathcal{R}^{-1} \; \begin{pmatrix} |\frac{\partial^2 u}{\partial \xi^2}|^{-5/6} |\frac{\partial^2 u}{\partial \eta^2}|^{1/6} & 0 \\ 0 & |\frac{\partial^2 u}{\partial \eta^2}|^{-5/6} |\frac{\partial^2 u}{\partial \xi^2}|^{1/6} \end{pmatrix} \; \mathcal{R} \; . \tag{86}$$

The case of the $\mathcal{L}^\infty$ norm can be *formally* derived by passing to the limit:

$$\mathcal{M}_{opt,\infty} = C_\infty^{-1} \; \mathcal{R}^{-1} \; \begin{pmatrix} |\frac{\partial^2 u}{\partial \xi^2}|^{-1} & 0 \\ 0 & |\frac{\partial^2 u}{\partial \eta^2}|^{-1} \end{pmatrix} \; \mathcal{R} \; . \tag{87}$$

And we again get an isorepartition of the integrand of (66).

3.3.6 Accuracy order

It is possible to extend the error analysis done above in order to get insight on the accuracy. In the case of an isotropic adaptation, the value at the optimum is

$$\mathcal{E} = \frac{C}{N} \int_\Omega M^{\frac{\alpha}{\alpha+1}} dxdy. \tag{88}$$

Then, second order accuracy will be satisfied if the above integral is bounded. In the case of the 2D Heavyside function equal to unity for positive x and zero otherwise, the largest eigenvalue is necessary that corresponding to x. Similarly to the 1D case, this second derivative is integrable at best at power one half. For $\alpha = 2$, the power in the optimal error is $2/3$. Thus second order accuracy is not obtained.

In the anisotropic case, the analog computation shows that in order to obtain second order accuracy with the Heavyside function, it is enough that the second $x-$derivative is integrable at power $1/4$, which is true since it is in $L^{1/2}$.

It is then remarkable that the proposed model suggests that isotropic optimal mesh adaption will not produce a second order accurate method in L^2 while the anisotropic optimal mesh adaption will produce such an accuracy order. These predictions are in accordance with the results in [9, 19].

3.4 A few numerical experiments

Several issues in our theory deserve numerical illustrations in order to be more convincing. Firstly, the conclusions of our analysis are valid only if the models are close enough to the discrete context. It is crucial to check that this does not hold only for extremely fine meshes, that could not be used in practice.

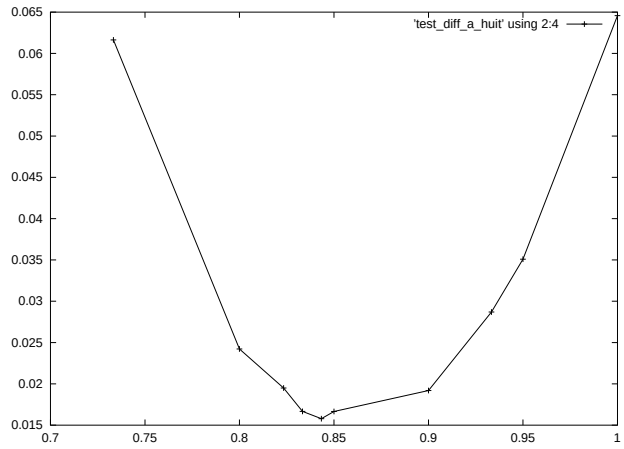

Figure 13: Optimality of the proposed metric: abscissas are values of the β parameter in the adaptation criterion, ordinates are values of the resulting L^2 interpolation error.

Secondly, the ideal meshes that are described as optimal in the case of L^∞ error norm are the ones generally proposed in the literature for solutions of elliptic problems. Now, usual error estimates for elliptic problems rely on a Hilbertian theory. We have shown that in L^2, a different family of optimal meshes is found already for the case of an interpolation error. Then it is important to validate our assertion that the second family is somewhat optimal, and to study what qualitative differences appearing when we shift from L^∞ to L^2.

In order to do this, we have to pass to a discrete context. For us, the discrete problem is just a discretization of the continuous optimality system, which expresses the optimal metric as a function of the Hessian of the continuous function to interpolate. These are the steps for building it:

- building the Hessian, either from analytic differentiation, or, preferably for the sequel, by using a background initial mesh, that is fine enough,

- deriving a metric (continuous or on the background mesh).

>From this, we get a discrete solution (the metric on the background mesh) of the continuous problem ("find the metric"). The error of the discretization is committed in computing the metric and is related to the quality of the background mesh.

Once the discrete metric is obtained, building the adapted mesh is just a post-treatment. Of course, the main interest is in the adapted mesh and the interpolation on it. We shall focus on the verification of its optimal properties. Before this, we describe how the adapted mesh is built from the metric.

3.4.1 Mesh adaptation tool

All the presented experiments were done thanks to the BAMG software [7]. Given a "background" mesh and an analytic function, BAMG evaluates on the mesh the Hessian of the function by a discrete derivation formula.

We have modified BAMG in order that the metric could be computed from the Hessian according to the above formula. Once the metric is obtained (on nodes of the background mesh), it is used in a mesh re-generator for rebuilding a new mesh following the metric. The mesh re-generator relies on a Delaunay reconnection in a space mapped by the metric and on vertex addition, again according to the metric. The number of nodes is adjusted when necessary by trial-and-error through the modification of the multiplicative coefficient of the metric. Many experiments with BAMG can be found in [28].

3.4.2 Some optimality

The purpose of this section is to give some numerical experiments showing some optimality of the variational solution proposed.

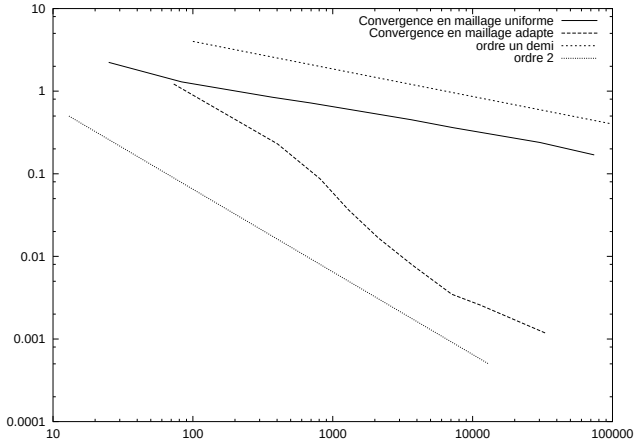

Figure 14: Convergence to exact function for $\beta = 0.66$, abscissas are numbers of nodes in the meshes used,, ordinates are values of the resulting L^2 interpolation error.

We consider the interpolation on the unit disk of the plane of the following function:

$$f(x,y) \;=\; 10x^3 \;+\; y^3 \;\;+\;\; \mathrm{atan}\left(\frac{\epsilon}{\sin(5y)-2x}\right) \tag{89}$$

We consider also a series of meshes, indexed by a positive β, that all have about 2100 nodes and are adapted according to the following formula:

$$\mathcal{M}_{opt} = \left(|\frac{\partial^2 u}{\partial \xi^2}|.|\frac{\partial^2 u}{\partial \eta^2}|\right)^{\beta} \mathcal{R}^{-1} \begin{pmatrix} \left(\frac{|\frac{\partial^2 u}{\partial \xi^2}|}{|\frac{\partial^2 u}{\partial \eta^2}|}\right)^{1/2} & 0 \\ 0 & \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|}\right)^{1/2} \end{pmatrix} \mathcal{R} \; . \tag{90}$$

This family of metrics involves the usual isorepartition metric, for $\beta = 1$ and the theoretically optimal metric for the L^2 norm, for $\beta = 5/6$ and $\epsilon = 0.0001$. We then compute the L^2 error. Outputs are depicted in Fig. 3.4. The error norm related to $\beta = 5/6$ is the lowest one, and is about three times smaller the those resulting from $\beta = 0.7$ or from $\beta = 1$.

3.4.3 Second-order accuracy

In [36] it is noted that for a smooth function with *steep gradients*, uniform mesh refinement shows numerical second order convergence only for very fine meshes, and that, in contrast, best mesh adaptation methods show a property of "early capturing of details", according to which second-order numerical convergence is observed with meshes with a much smaller number of nodes. The purpose of this section is to show a typical example for which the early capturing of details occurs.

In order to evaluate this phenomenon, we have considered the function defined in (89), with $\epsilon = 0.001$. We do not observe mesh convergence with uniform refinement, even with meshes with almost $100,000$ nodes. We now adapt the meshes by applying the proposed metric. The effect is a much faster convergence, essentially second order, observable for meshes as coarse as of a few hundred nodes (Fig.14).

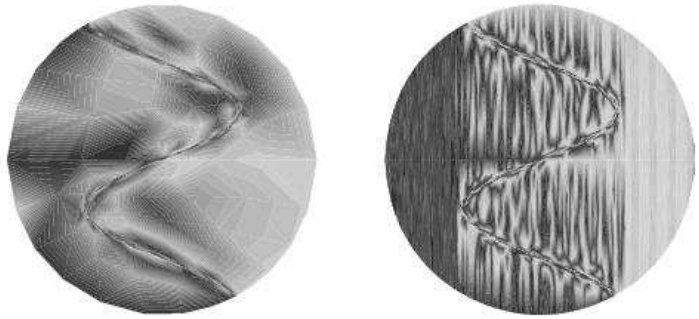

Figure 15: Representation of a function with the two options, L^∞ (left) and L^2(rigth): contours.

3.4.4 The influence of the choice of the norm

As mentioned earlier, the proposed variational analysis takes into account the functional space L^α in which we minimize the interpolation error. The influence of the functional norm will now be studied in relation with the application of a method for image compression.

Indeed, given a function defined on a fine (uniform or not) mesh, a compression could consist of storing it in a smaller mesh, accepting some loss in the accuracy of its definition, as far as the new file is sufficiently small. Mesh adaptive interpolation is an answer to this problem, already used in image processing [31].

The first example will illustrate the better ability of the L_p option with small p to adapt the mesh to the small amplitude details of a function. Let us start with the sum of an arctangent function of amplitude 1., combined this time with a sine function of a ten times lower amplitude.

$$f(x,y) \quad = \quad 0.1\,\sin(50x) \quad + \quad \mathrm{atan}\left(\frac{0.001}{\sin(5y) - 2x}\right) \tag{91}$$

We compare two adapted meshes with about 2000 nodes each. Meshes are depicted in Fig.16 and the resulting contours of interpolated functions are presented in Fig. 15. The first one is adapted following the error isorepartition principle, in other words, by minimizing the L^∞ error functional. The second one is adapted according to the minimization of the L^2 error norm. We observe that L^2 option restitutes the low amplitude sine oscillation while the L^∞ oes not show it at all.

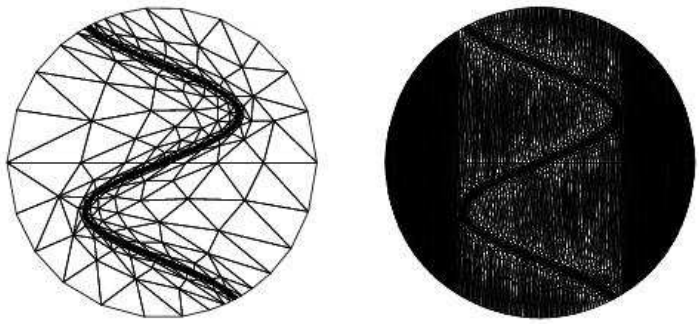

Figure 16: Representation of a function with the two options, L^∞ (left) and L^2(right): corresponding meshes.

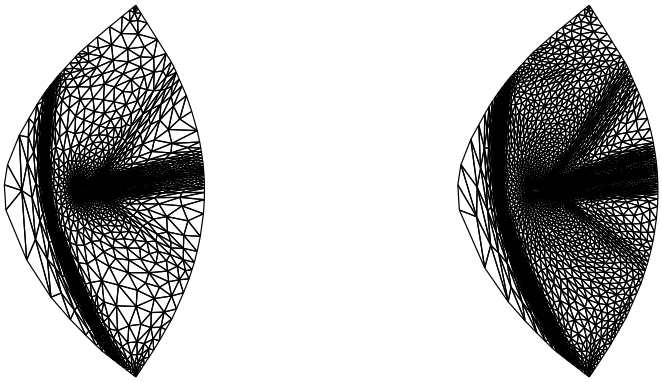

Figure 17: Mesh-based compression of Mach contours: view of the meshes for L^∞ and L^2 compressions.

Mesh-based image compression is particularly useful for storing images produced by numerical finite element computations. As an illustration we consider the compression of the Mach contours of a flow analysis. Conditions of flow calculation is not important for our subject. In Fig. 17 we compare the L^∞ and L^2 adapted meshes. The ramp-like shocks starting from the middle of the airfoil correspond to a much smaller amplitude of variation than the vertical bow shock at left part. We observe that they are nearly ignored by the L^∞ option while it is well followed by the L^1 option.

3.5 An application to PDE adaption

Certified convergence in CFD

In the previous section, we get a mesh convergence at the asymptotical order. We examine now how we can derive from it an evaluation of the accuracy of the result, and therefore a kind of "certification" of its quality.

In the numerical experiments presented so far, we compared our calculations with a reference, "very accurate" solution, which in practice should not be available. Instead, we can measure the numerical order of convergence by comparing three results obtained with meshes involving N nodes, $4N$ nodes and $16N$ nodes.

As a test case, we consider the very easy computation of a laminar flow around a NACA0012 airfoil. Farfield Mach number is 1.2 and Reynolds number is 73. A summary of results is presented in Table 3.5. We start with uniform refinement with a coarse mesh of rather good quality, with 3000 nodes (i.e. vertices). It is uniformly divided first into a 12,000-node mesh which, in turn, is divided uniformly in a 48,000-node mesh. We then compute the numerical order of convergence, which gives only a first-order numerical convergence (see Table 3.5).

Conversely, if each successive mesh is built with anisotropic adaptation, a convergence order of $1.6-1.8$ is already observed with a coarser sequence of 800-3,000-12,000-node meshes. Second-order convergence is confirmed by a finer sequence of adapted meshes with the two previous adapted meshes, 3,000-12,000-node, complemented by a 48,000-node adapted mesh. Extrapolation strategies can then be applied to specific outputs (lift, drag, etc.), see for example [39].

Flow variables	Order for (a) Uniform (fine)	Order for (b) Adaptive (coarse)	Order for (c) Adaptive (fine)
Density, ρ	0.948	1.62	2.15
Horizontal moment, ρU	1.016	1.78	1.75
Vertical moment, ρV	1.024	1.85	1.87

Table 3: Numerical convergence order in L^2 norm for an airfoil flow: mesh sequence (a) is obtained by uniform refinement, with meshes of 3000, 12000, 48000 nodes, mesh sequence (b) is obtained with coarser anisotropic adaptated meshes of 800, 3000, 12000 nodes, mesh sequence (c) is obtained with anisotropic adaptated meshes of 3000, 12000, 48000 nodes.

Let us denote by u_N the solution computed with the adapted anisotropic mesh of N vertices. >From the computation we get:

$$||u_{3000} - u_{12000}||_{L^2} \;=\; 1,802\ 10^{-4}\ ,$$

Relying on the second-order convergence we can *certify* the order of magnitude of the error on the medium mesh as follows:

$$||u_{12000} - u_{exact}||_{L^2} \;\leq\; (1,802/3)\ 10^{-4} = 0,600\ 10^{-4}\ . \tag{92}$$

As expectable, this estimate is coherent with the finest calculation which gives:

$$||u_{12000} - u_{48000}||_{L^2} \;=\; 5,637\ 10^{-5}\ .$$

3.6 Metric and PDE

An important difficulty in mesh adaptation is the choice of the quantities that will prescribe the fineness of the adapted mesh. Indeed, the approximation error is a complex non-local function of mesh fineness. The main idea to solve this difficulty is to consider the error as the solution of a continuous or discrete linear error system with a right hand side, the truncation error that should depend only on the local quality of the mesh. On these grounds, two families of mesh adaptation methods can be distinguished.

In a first set of works, a particular function of the dependant variable, the **sensor**, is chosen and its derivatives are considered as a good local truncation error indicator. This can be justified in the case of Hessian based adaptation (see for example [7], [23]). Indeed, the proposed indicator is derived from interpolation error, and, in the case of finite-element approximations, a classical *a priori* error analysis shows that approximation error is smaller than the interpolation error. The paramount interest of this approach for anisotropic adaptation is a very sound indication of the necessary stretching of the mesh. Another important feature is that it involves the research of an ideal mesh.

In a second series of works, several authors propose a rather accurate definition of the final goal of the adaptative process: to minimize a norm of the error or a real-valued functional to compute accurately. Then adjoint-based *a posteriori* error estimates were systematically derived from this specification. This allows to compute an error level and to refine the regions where the local truncation error is larger than

a tolerance [3, 43]. The fact that effective *a posteriori* errors are controlled is an advantage of this kind of methods. These works are progressing towards anisotropic adaptation. One difficulty on this way is that what is handled is a correction with respect to an existing mesh instead of the specification of an ideal mesh which should not depend on the current mesh.

We propose to extend the first approach, based on *a priori* estimates, with the goal of obtaining an ideal mesh, by considering a purely continuous model for the mesh adaptation problem. The "continuous metrics method" idea was elaborated in [13] for the simpler problem of getting the best possible mesh to interpolate a given analytic function. In the present paper, we propose an extension of these ideas to mesh adaptation for PDE. This extension keeps the same choice of stretching direction as in [13]. It takes also some inspiration from adjoint-based methods. But it goes a little further in the direction of setting the mesh adaption issue as a standard **optimal control problem**. New efficient algorithms are being developed in the litterature for solving optimal control problems (let us refer to two recent contributions of ours: [18] and [10]). Finally, we shall discuss in the last section how an optimal control of the mesh can be combined with an optimal design loop.

Finite-Element model

We concentrate on the usual Poisson problem in a polyhedral n-dimensional domain:

$$\Delta u \;=\; f \;\text{ on }\; \Omega \;\;;\;\; \mathrm{u} = 0 \;\text{ on }\; \partial\Omega. \tag{93}$$

the variational form of which is written

$$a(u,v) \;=\; \int \nabla u.\nabla v \; dx \;=\; < \; f \; , \; v > \qquad \forall \; v \in V \tag{94}$$

where V holds for the Sobolev space $H^1(\Omega) = u \in L^2(\Omega), \nabla u \in (L^2(\Omega))^n$. Let τ_h be a mesh of Ω made of simplexes, let V_h be the subspace of V of continuous functions that are $\mathcal{P}_1$ on each element of the mesh. The discrete variational problem is

$$a(u_h, v_h) \;=\; < \; f \; , \; v_h > \qquad \forall \; v_h \in V_h \tag{95}$$

A posteriori error estimates have been derived by many authors for this approximation. These analyses propose different forms of the main term of the error. On the other hand, *a priori* estimates have been derived much more earlier, in $H^1(\Omega)$ ("projection property"), and in $L^2(\Omega)$ (Aubin-Nitsche analysis), but only by means of inequalities. It is worth to mention that the main part of the error is generally not computed.

The approximation error in the sequel will be split into an interpolation error and an *implicit error*:

$$u_h \;-\; u \;=\; u_h \;-\; \Pi_h u \;+\; \Pi_h u \;-\; u \tag{96}$$

where the interpolation error $\Pi_h u \;-\; u$ is explicit (in function of u) but infinite-dimensional, and the *implicit error* $u_h \;-\; \Pi_h u$ is finite-dimensional, but implicitly defined from u, that is with the help of the inversion of a system.

Majoration of the implicit error

Classically we can perform this by writing that the finite-element solution is better than a direct projection of the continuous unknown. The new idea is that we proceed with equalities, as it is next explained. We assume for simplicity that the right-hand side f is exactly integrated. The discrete system writes:

$$a(u_h, v_h) \;=\; (\; f, v_h) \;\forall \; v_h \in V_h \tag{97}$$

For any v in V its projection $\bar{\Pi}_h v$ is in V_h, thus

$$a(u_h, \bar{\Pi}_h v) \;=\; (\; f, \bar{\Pi}_h v \;) \;\forall \; v \in V$$

then appears the continuous differential operator:

$$(Au_h , \bar{\Pi}_h v)_{D',D} = (f, \bar{\Pi}_h v) \forall v \in V \quad (98)$$

we also use the adjoint of the projector:

$$(\bar{\Pi}_h^* A u_h , v) = (\bar{\Pi}_h^* f , v) \forall v \in V.$$

Thus

$$\bar{\Pi}_h^* A u_h = \bar{\Pi}_h^* f \text{ in } V'$$

This allows the following error analysis:

$$\begin{aligned} \bar{\Pi}_h^* A u_h - \bar{\Pi}_h^* A \bar{\Pi}_h u &= \bar{\Pi}_h^* f - \bar{\Pi}_h^* A \bar{\Pi}_h u \text{ in } V' \\ \bar{\Pi}_h^* A (u_h - \bar{\Pi}_h u) &= \bar{\Pi}_h^* A (u - \bar{\Pi}_h u) \text{ in } V' \end{aligned} \quad (99)$$

For u and ϕ smooth, and by a density argument, the above calculation extends to the usual interpolation Π_h. This is an *a priori* analysis since we express the approximation error $u_h - \Pi_h u$ as a function of the unknown u.

Remark:
We can already note that the implicit error will be small when the interpolation one will be. This pleads for the choice of a mesh stretching designed for the interpolation error, i.e., typically for a Hessian based stretching.

3.6.1 Error modeling

For the sake of clarity, we describe our error model in a two-dimensional context. In contrast to *a posteriori*-based adaptation, we do not try to repair an **existing** mesh, but to find, in a set of regular enough meshes, the **ideal** mesh for maximizing the accuracy on the problem to solve. This allows to apply the second idea of the method, which consists of representing the mesh with a continuous field.

In an extremely regular mesh, the main part of error can show compensations between two neighboring elements. For a smooth function u, the approximate gradient $\nabla \Pi_h u$ can be of first order accuracy on a given isocele triangle $T_1 = ABC, CA = CB$, while being (weakly) of second order accuracy on the union of this triangle with the triangle T_2 symmetric with respect to the basis AB (Fig.18).

Let I be the center of AB, the basis is assumed to be of length Δx and the height of length Δy. Then, for any smooth ϕ, the following estimate holds:

$$\frac{1}{meas(T_1 \cup T_2)} \int_{T_1 \cup T_2} (\nabla_x u - \nabla_x \Pi_h u) . \nabla_x \Pi_h \phi \, dx =$$

$$- \frac{1}{48} u_{xxx}(I) \phi_x(I) (\Delta x)^2 + \frac{1}{12} u_{xyy}(I) \phi_x(I) (\Delta y)^2 + o(Max(\Delta x, \Delta y)^2)$$

Due to the smoothness assumptions on the ideal mesh, we are placed in a context of superconvergence at second-order.

Mesh modeling

In this paper we make the (indeed not so arbitrary) choice of a meshes family, being all of them stretched in the same direction as the Hessian of function u. In short, this means that the Hessian diagonalizes as follows:

$$\nabla \nabla u = \mathcal{R}^{-1} \mathcal{D} \mathcal{R}$$

where $\mathcal{D}$ is diagonal, and that the mesh family under investigation is defined by the following set of metrics:

$$\mathcal{M} = \mathcal{R}^{-1} \mathcal{N} \mathcal{R}$$

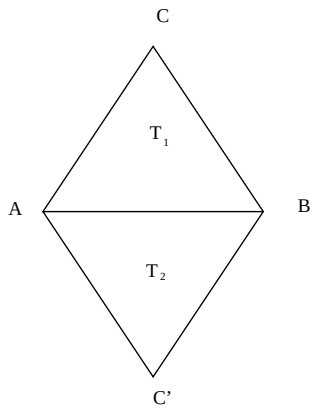

Figure 18: Interpolation error analysis

where

$$\mathcal{N} \;=\; diag(m_\xi, m_\eta)\ .$$

is the unknown mesh descriptor. According to [7] the edges of mesh that is represented by a metric are of lenght unity. In practice, this means that the direction of stretching is defined by the rotation $\mathcal{R}$, with m_ξ and m_η defined as the local mesh sizes in the two stretching directions.

Optimal Control model

Given a metric $\mathcal{M}$ of the above family, we can define a continuous implicit error model as follows:

$$Y \;\in\; H_0^1(\Omega)\ ,\ \text{and}\ \forall\ \ \phi \;\in\; H_0^1(\Omega)\ ,$$

$$\int_\Omega \nabla Y.\nabla\phi\ dx \;=$$

$$\frac{-1}{48}\int_\Omega u_{\xi\xi\xi}\phi_\xi(m_\xi)^2\ dx \;+\; \frac{1}{12}\int_\Omega u_{\xi\eta\eta}\phi_\xi(m_\eta)^2\ dx \;+$$

$$\frac{-1}{48}\int_\Omega u_{\eta\eta\eta}\phi_\eta(m_\eta)^2\ dx \;+\; \frac{1}{12}\int_\Omega u_{\eta\xi\xi}\phi_\eta(m_\xi)^2\ dx$$

This is a Dirichlet problem that can be interpreted as follows:

$$-\Delta Y \;=\; \left(\frac{1}{48}u_{\xi\xi\xi}(m_\xi)^2 \;-\; \frac{1}{12}\,u_{\xi\eta\eta}(m_\eta)^2\right)_\xi \;+\; \left(\frac{1}{48}u_{\eta\eta\eta}(m_\eta)^2 \;-\; \frac{1}{12}\,u_{\eta\xi\xi}(m_\xi)^2\right)_\eta\ .$$

$$Y \;=\; 0\ \text{ on }\ \partial\Omega$$

We recall that the complete error is derived from the addition of (a) the implicit error and (b) the interpolation error on u. We made some numerical comparison between Y and $\Pi_h u - u$ with $u(x,y) = (x^2 - x)(y^2 - y)$ for a 100×100 uniform mesh and verified in that case the accuracy of this model, as can be seen Fig. 19.

At this point, we are able to look for the best mesh for minimizing a given error. This is formulated as an Optimal Control problem. The control is the metric, the state variable is the solution $Y(\mathcal{M})$ of the above system and represents the approximation error. Let J be a functional depending on both control and state and let us define:

$$j(\mathcal{M}) \;=\; J\left(\mathcal{M}, Y(\mathcal{M})\right).$$

Then, the optimization problem to solve writes:

$$\text{Find}\ \ \mathcal{M}_{opt} \;=\; \text{ArgMin}\ j(\mathcal{M})\ .$$

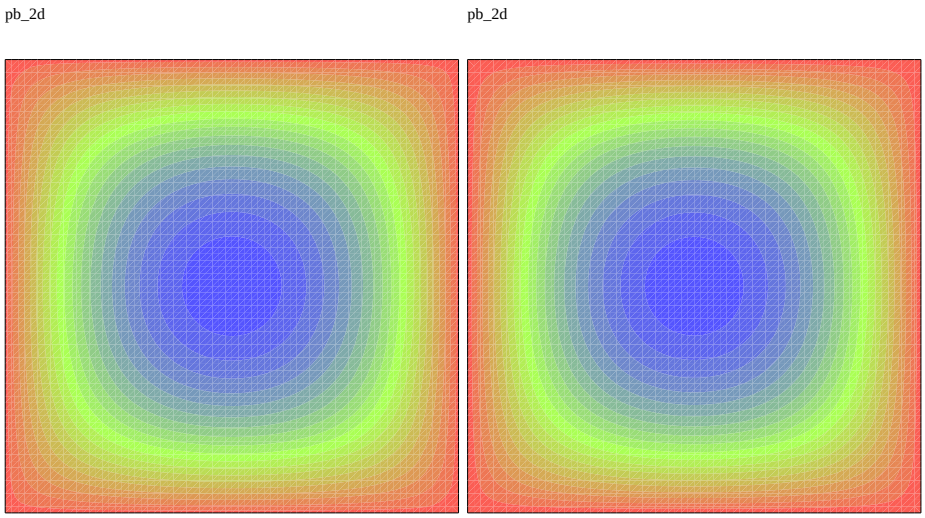

Figure 19: Comparison between approximation error and its model, function Y

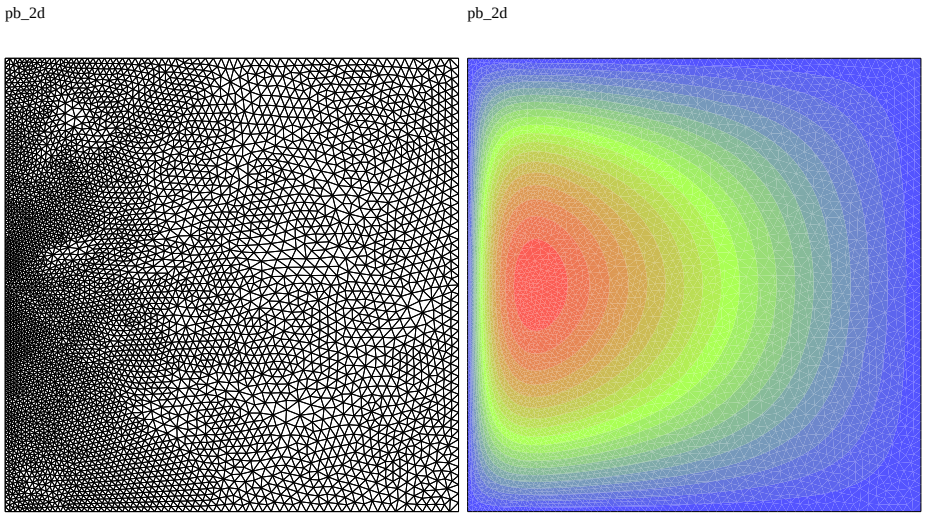

Figure 20: New mesh and new solution after optimization of the metric

3.7 Next steps

Experiments are currently done for the evaluation of this approach. The first functional under investigation is the H^1 semi-norm of the implicit error model Y. The minimisation problem is discretized and solved by a gradient-type method with a fixed mesh. After convergence of the optimization loop, we obtain an "optimal metric" which can be used for specifying a new adapted mesh. Preliminary outputs are depicted in Fig. 20.

Some important potential advantages of the proposed methodology are:

- As in Hessian-based methods, an ideal mesh is seeked.
- As in *a posteriori* methods, needs in accuracy can be specified by user under the form of a functional. The functional can be an error, as shown in last section, or it can be a particular (scalar) output, to be computed accurately. That second aspect opens the door to a combination of adaptation with an optimal design loop, as described in the next section.

- As in [13], the continuous model can be extended to the case where the solution presents some singularities. In that case, error order is modelled and can be predicted as second-order for optimal meshes.

There are two main difficulties to be solved. Firstly, an approximation for the third or fourth derivatives of the unknown have to be introduced in order to treat a generic mesh adaptation problem. Secondly, further analysis is needed in order to have a rigorous analysis of the connection between the model and discrete solutions.

4 TOWARDS OPTIMAL DESIGN WITH ERROR CONTROL

This section describes the idea of integrating the above method of accuracy control by adaptivity into an optimal design process. The utility of this idea is clear, because with some clever implementation work, an integrated scheme can render the optimal design with the best attainable accuracy.

Let γ be a design parameter. For example, it can be the shape of the embedded boundary of a membrane. >From this parameter we can deduce the displacement $u(\gamma)$ as solution of a state equation described by mapping Ψ. The optimal design problem in finding:

$$\bar{\gamma} \quad = \quad \text{ArgMin } j$$

with

$$j(\gamma) \quad = \quad \mathbf{J}(\gamma, u(\gamma))$$

$$\text{where} \quad u(\gamma) \quad \text{is the solution of} \quad \Psi(\gamma, u(\gamma)) = 0 \ .$$

Suppose now that a certain discretisation method is used for solving this system. Lacking of a good theory concerning the acccuracy of an approximation of the solution $\bar{\gamma}$ of the optimum, we restrict our accuracy analysis to the search of the proper accuracy for the functional j.

The problem of mesh adaptation for improving the accuracy of a real-valued functional has been considered in recent works relying on *a-posteriori* estimates. In some case, some higher-order accuracy (also refered as superconvergence) for the functional can be put in evidence ([3]), in some other case some adjoint-based correction can be introduced for obtaining this superconvergence ([22]). In both cases, mesh improvements can be defined from the error term [3, 43].

Following the standpoint of the present paper, we introduce an extra parameter, the mesh metric $\mathcal{M}$. Departing from the above shape design functional and for a given γ, we can derive a first order error functional

$$\bar{\mathbf{J}}_\gamma(\mathcal{M}, Y(\mathcal{M})) \quad = \quad |J(\mathcal{M}, \gamma, W(\mathcal{M}))'.(Y_\gamma(\mathcal{M}))|^2$$

where the error $Y_\gamma(\mathcal{M})$ is solution of an error state system:

$$Y \ = \ Y_\gamma(\mathcal{M}) \Leftrightarrow \Psi_{ERROR}(\mathcal{M}, \gamma, Y) = 0.$$

This error system has to be derived from the state system, for example using the ideas developed in the above sections. It remains to find the optimal mesh metric:

$$\bar{\mathcal{M}}_\gamma \quad = \quad \text{ArgMin } \bar{\mathbf{J}}_\gamma(\mathcal{M}, Y_\gamma(\mathcal{M})) \ .$$

We observe that, until a true mesh update is applied, the metric optimisation and the design optimisation can be done in a loosely coupled manner since the metric does not influence the flow evaluation. Conversely, it is not efficient to push too far the metric optimisation before the optimum shape is nearly found.

The combination with the shape design loop can be typically organised by the alternation of two phases: (a) the mesh is frozen and a better shape is found by minimizing J, (b) the shape is frozen, the mesh metric is found by minimising $\bar{J}$, and the resulting metric is used to regenerate a better mesh. Now, what we recommand from the various remarks made in these notes, is to reiterate this composite loop in order to find a kind of best mesh for a given number N of nodes. Then this converged result can be used as a starting condition for mesh size convergence as shown below, with the purpose of verifying that the convergence order is the one expected, which would garanty a small error level, as in [39].

5 Conclusions

As a preliminary condition for a stronger development of automated design optimisation in CFD applications, it seem of paramount importance that one controls the error made in the evaluation of the design. This means that one controls the error in the computation of the cost functional (and when applicable the state constraints). An interesting strategy for errro control is to formulate is as a genuine Optimal Control problem with a tractable parametrization of the mesh. Our suggestion is to use the metric parametrization for isotropic and anisotropic meshes. This programme involves an approach combining error control and shape control, and needs efficient tools for the fast solution of large scale problems of constrained optimisation with computer-intensive CFD state equation models.

References

[1] I. Babuska and T. Strouboulis. *The Finite Element method and its reliability.* Oxford Science Publication, 2001.

[2] 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.

[3] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[4] H. Borouchaki, P.L. George, and B. Mohammadi. Delaunay mesh generation governed by metric specifications. part 2: Application examples. *Finite Elements in Analysis and Design*, 25:85–109, 1997.

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

[6] 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.

[7] M.J. Castro-Diaz, H. Borouchaki, P.L. George, F. Hecht, and B. Mohammadi. Anisotropic adaptive mesh generation in two dimensions for CFD. In *Computational Fluids Dynamics '96*, volume 32, pages 181–192, 1996.

[8] 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.

[9] Y. Coudière, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. INRIA Report 4528, 2002.

[10] F. Courty and Dervieux A. *A SQP-like one-shot algorithm for optimal shape design.* Notes on Numerical Fluid Dynamics, V. Selmin Ed., Springer, to appear, 2004.

[11] F. Courty, A. Dervieux, B. Koobus, and L. Hascoet. Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation. INRIA Report 4363, January 2002. http://www-sop.inria.fr/tropics/Francois.Courty/Publications.html.

[12] F. Courty, D. Leservoisier, P.-L. George, and A. Dervieux. Continuous metrics and mesh optimization. *submitted for publication*, 2004.

[13] F. Courty, D. Leservoisier, P.-L. George, and A. Dervieux. Continuous metrics and mesh optimization. *Applied Numerical Mathematics*, 2005.

[14] A. Dadone and B. Grossman. Progressive optimization of inverse fluid dynamic design problems. *Computer and Fluids*, 29:1–32, 2000.

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

[16] A. Dervieux. Résolution de problèmes à frontière libre. *Thesis, university of Paris VI*, 1981. (in French).

[17] A. Dervieux, F. Courty, M. Vázquez, and B. Koobus. Additive multilevel optimization and its application to sonic boom reduction. In *Numerical Methods for Scientific Computing - JP60 Meeting, Variational Problems and Applications*, Jyvaskyla, Finland, 2002. http://www-sop.inria.fr/tropics/Francois.Courty/Publications.html.

[18] A. Dervieux, F. Courty, M. Vazquez, and B. Koobus. *Additive multilevel optimization and its application to sonic boom reduction*, pages 31–44. CIMNE, Barcelona, 2003.

[19] A. Dervieux, D. Leservoisier, P.-L. George, and Y. Coudière. About theoretical and practical impact of mesh adaptions on approximation of functions and of solution of pde. In *Conférence invitée à ECCOMAS-Swansea.* to appear in I.J.M.N.F., 2001.

[20] 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).

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

[22] M.-B. Giles. *Adjoint methods for aeronautical design.* Proceedings of the ECCOMAS CFD Conference, 2001.

[23] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent cfd solutions: Part I: General principles. *International Journal for Numerical Methods in Fluids*, 32:725–744, 2000.

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

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

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

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

[28] D. Leservoisier. *Strategies d'adaptation et de raffinement de maillages en Mecanique des fluides.* PhD thesis, Université Pierre et Marie Curie, 2001.

[29] D. Leservoisier, P.-L. George, O. Penanhoat, and A. Dervieux. Application of mesh-adaptive techniques to mesh convergence in complex CFD. In *Second international symposium on Finite Volumes, for Complex Applications, Problems and Perspectives, university of Duisbourg, july 1999, Finite Volumes for Complex Applications, R. Vilsmeier, F. Benkhaldoun, D. Haenel, Eds., Hermes Science Pub.*, pages 817–824, 1999.

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

[31] G. Marquant, S Pateux, and C. Labit. Mesh-based scalable image coding with rate-distortion. In *European Signal Processing Conference EUSIPCO 2000*, 2000.

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

[33] 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.

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

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

[36] B. Palmerio and A. Dervieux. Multimesh and multiresolution analysis for mesh adaptive interpolation. *Applied Numerical Mathematics*, 22:477–493, 1996.

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

[38] 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.

[39] E. Schall, D. Leservoisier l, A. Dervieux, and B. Koobus. Mesh adaptation as a tool for certified computational aerodynamics. *I.J. Num. Meth. Fluids*, 45:179–196, 2004.

[40] S. Ta'asan. One shot methods for optimal control of distributed parameter systems i: finite dimensional control. ICASE, Technical Report 91-2, NASA contractor report, 1991.

[41] M. Vazquez, A. Dervieux, and B. Koobus. Aerodynamical and sonic boom optimization of a supersonic aircraft. INRIA Report 4520, 2002. http://www-sop.inria.fr/rapports/sophia/RR-4520.html.

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

[43] D.A. Venditi and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comput. Phys.*, 187:22–46, 2003.

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

[45] 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, Leicester, GB, July 8–19, 1996.*, pages 213–302. Oxford: Clarendon Press. Numerical Mathematics and Scientific Computation., 1997.

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

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399