POST-SAROD Workshop, Bangalore, dec. 2005

OPTIMIZATION LOOPS FOR SHAPE AND ERROR CONTROL

A. Dervieux[1], L. Hascoet[1],
M. Vázquez[1,2], B. Koobus[1,3]

[1] INRIA, Centre de Sophia-Antipolis, BP 93, 2004 Route des Lucioles, 06902 Sophia-Antipolis Cedex, France, alain.dervieux@sophia.inria.fr, laurent.hascoet@sophia.inria.fr
[3] Université de Montpellier II, Dept Mathématiques, CC.051 34095 Montpellier Cedex 5, France, koobus@math.univ-montp2.fr
[2] Universitat de Girona, Av. LLuìs Santaló s/n - 17071 Girona, Spain, mariano.vazquez@udg.es

1. LARGE-SCALE DESIGN

Simulation: from a particular design, compute its performances via a model system

Design: find the optimal design which minimizes a performance-based objective functional

Large-scale Design: the design variable set is large-scale, the model system is large-scale, the optimization loop cannot use the objective functional as a black box.

- derive sensitivity of existing model,

- build adequate algorithms.

What is available?

The simulation software involves:

- the assembling part: computes from parameters and arbitrary state variables a residual reflecting how the state variables satisfy the state equation:

$$\text{state-assembler:}(\gamma,W) \rightarrow \Psi(\gamma,W)$$

- the solution algorithm (which calls state-assembler):

$$\text{state-solution-algorithm:}\gamma \rightarrow W(\gamma) \text{ , such that } \Psi(\gamma,W(\gamma) = 0.$$

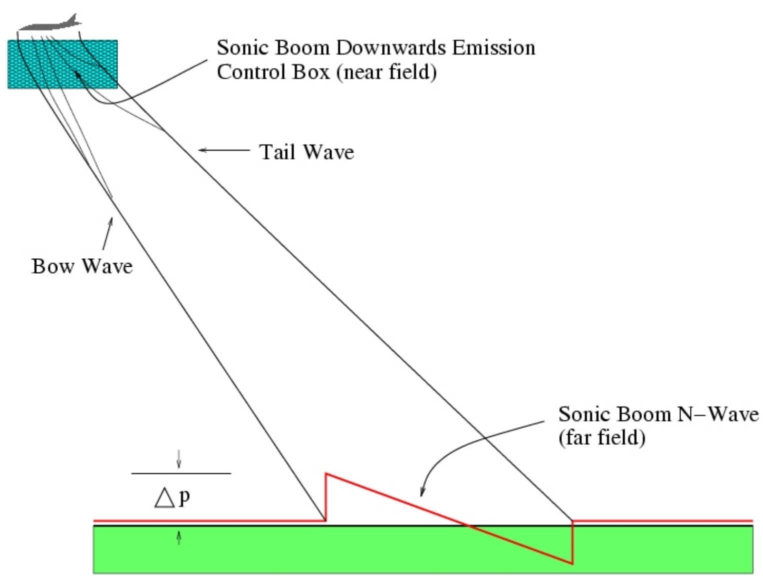
- the post-evaluation software which we assume to be already written as an objective/cost function, involving no solution algorithm:

$$\text{cost-assembler:}(\gamma,W) \rightarrow J(\gamma,W).$$

2. EXAMPLE 1: SHAPE DESIGN

Control: shape γ of the aircraft
State: 3D Euler.
Cost: We measure the "sonic boom downwards emission" with a volume integral of squared pressure gradient.

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



Discretized problem

The discrete CFD model uses an upwind Euler solver applying to unstructured tetrahedrizations.

The shape is changed by moving the nodes on the boundary of the mesh along normals to that boundary.

Their displacement is taken into account by a transpiration condition in order to avoid costly remeshings.

3. OPTIMAL DESIGN SYSTEM ASSEMBLY

Apply Automatic Differentiation (AD):

$$u \mapsto v = \Phi(u)$$

$$\begin{aligned}\Phi'(u) \quad = \quad & (\phi'_p \circ \phi_{p-1} \circ \phi_{p-2} \circ \cdots \circ \phi_1(u)) \\ . \; & (\phi'_{p-1} \circ \phi_{p-2} \circ \cdots \circ \phi_1(u)) \\ . \; & \ldots \\ . \; & (\phi'_1(u)) \; .\end{aligned}$$

the **tangent mode** of AD produces a routine computing the directional derivative:

$$u,\dot{u} \mapsto \frac{\partial \Phi}{\partial u}(u) \; \dot{u} \; .$$

only one output if Φ is a functional.

Reverse mode:

$$u \mapsto v = \Phi(u)$$

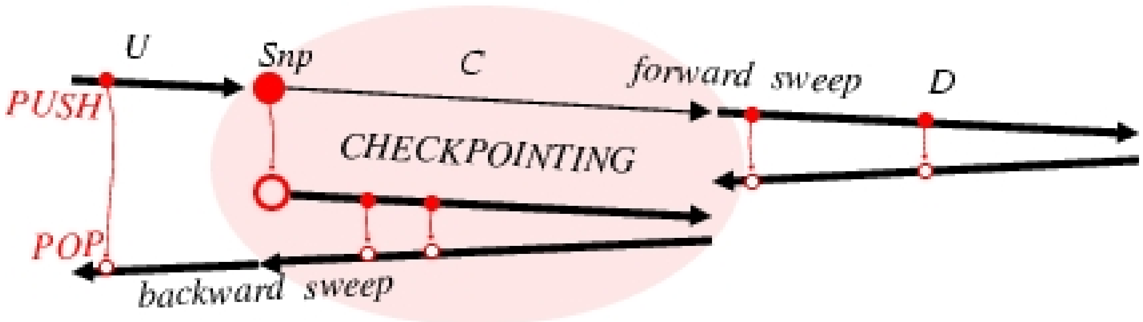
$$\Phi'^{*}(x).\overline{y} = \phi_1'^{*}(x_0).\phi_2'^{*}(x_1).\ldots.\phi_p'^{*}(x_{p-1}).\overline{y}$$

The **reverse mode** of AD produces a routine computing from u and any array $\overline{v}$ of the same dimension as v the following product which has the dimension of u:

$$u,\overline{v} \mapsto \left(\frac{\partial \Phi}{\partial u}(u)\right)^{*} \overline{v} \ .$$

Reverse mode (cont'd):

Reverse mode is two-way and needs recovering data (stored or recomputed):



In the case of a functional of n variables, gives n outputs and thus can be $n/2$ times more efficient than usual AD.

Reverse mode of AD is available in AD tools such as TAPENADE.

Hascoet-Pascual: Tapenade manual.http://www-sop.inria.fr/tropics

Functional's gradient/KKT optimality system

$$
\left\{
\begin{array}{rcl}
\Psi(Y,u) & = & 0 \\
& & \text{(State)} \\
\left(\dfrac{\partial \Psi}{\partial Y}(Y,u)\right)^{*} \Pi & = & \left(\dfrac{\partial J}{\partial Y}(Y,u)\right) \\
& & \text{(Adjoint state)} \\
j'(u) = \left(\dfrac{\partial J}{\partial u}(Y,u)\right) - < \left(\dfrac{\partial \Psi}{\partial u}(Y,u)\right)^{*} \Pi \ , \ 1 > & = & 0 \\
& & \text{(Stationarity)}
\end{array}
\right.
$$

Computation of functional's gradient

Either:

-(a) apply reverse AD only to assembly routine.

-(b) apply reverse AD to the whole simulation+cost sequence,

(b) gives an exact derivative if (at least):
- the linearised fixed-point state-solver is stable.

(a) is completed by applying the state iteration algorithm to adjoint system solution.

It remains to introduce a minimization iteration.

4. OPTIMAL DESIGN ALGORITHM: A. One-shot

- Sequential Quadratic Programming (SQP) methodology. Allows inexact solution of linear systems.

- **one-shot** -or progressive, or simultaneous- algorithms (Taasan, Dadone-Grossman, Courty-Dervieux), 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).

One-shot decreases impressively the cost of one global iteration.

Optimal Design Algorithm: B. Preconditioner

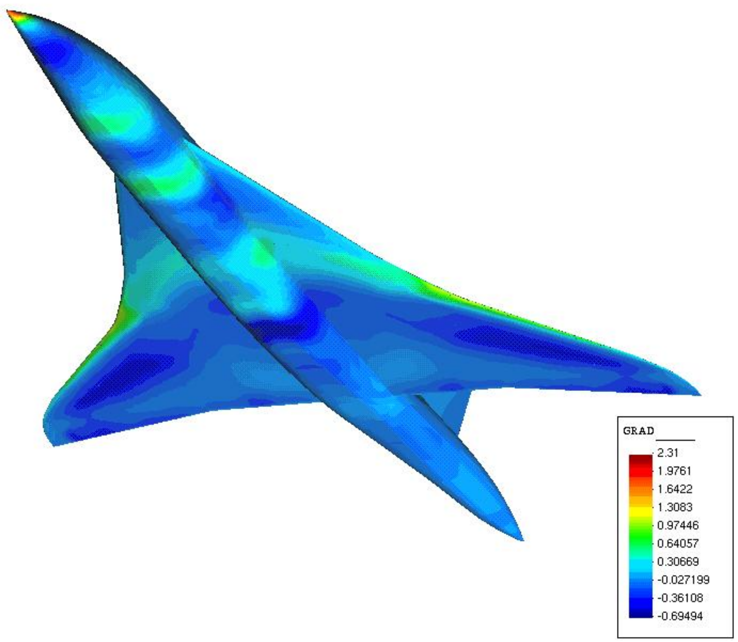
$$\gamma^{n+1} = \gamma^n - \rho B g(\gamma_n), \tag{1}$$

$g(\gamma_n)$ is generally less regular than $_n$.
For example g involves spatial differentiation operators of degree n, positive, which are then unbounded operators.
In discrete case, these operators have highest eigenvalues associated with high frequencies and mesh size. If no preconditioner is used ($B =$Identity), convergence of discrete version will be slow and mesh dependant.
In order to get *essentially mesh-independent rates*, the preconditioner B should recover the degree of regularity lost by the operator g.

5. APPLICATION TO SHAPE OPTIMISATION

γ : design parameter aircraft shape,
(20,000 unknowns).

$W(\gamma)$: state variable, 3D Euler ($5\times$ 170,000 variables).



Functional's gradient visualization

Functional Analysis:

The derivative of a PDE solution with respect to boundary can be obtained from the Hadamard variational formula involving in that case first-order spatial derivatives. For example for a Dirichlet problem,

$$\Delta u_\gamma = 0,\ u_\gamma = \delta\gamma \tfrac{\partial u}{\partial n}$$

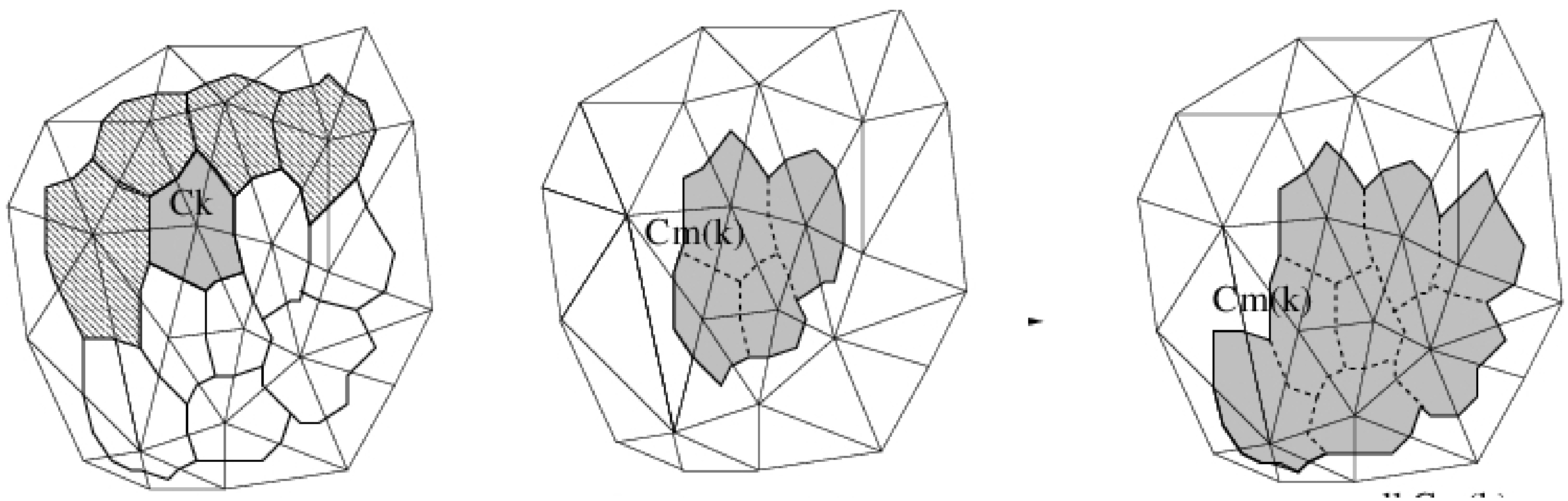
Gradient $G(\gamma)$ is also unbounded with respect to shape:

$$G(\gamma){:}C^{l+\alpha}(\Gamma) \rightarrow C^{l-1+\alpha}(\Gamma)$$

with a loss of 1 **derivation**.

Additive multilevel preconditioner

Volume agglomeration



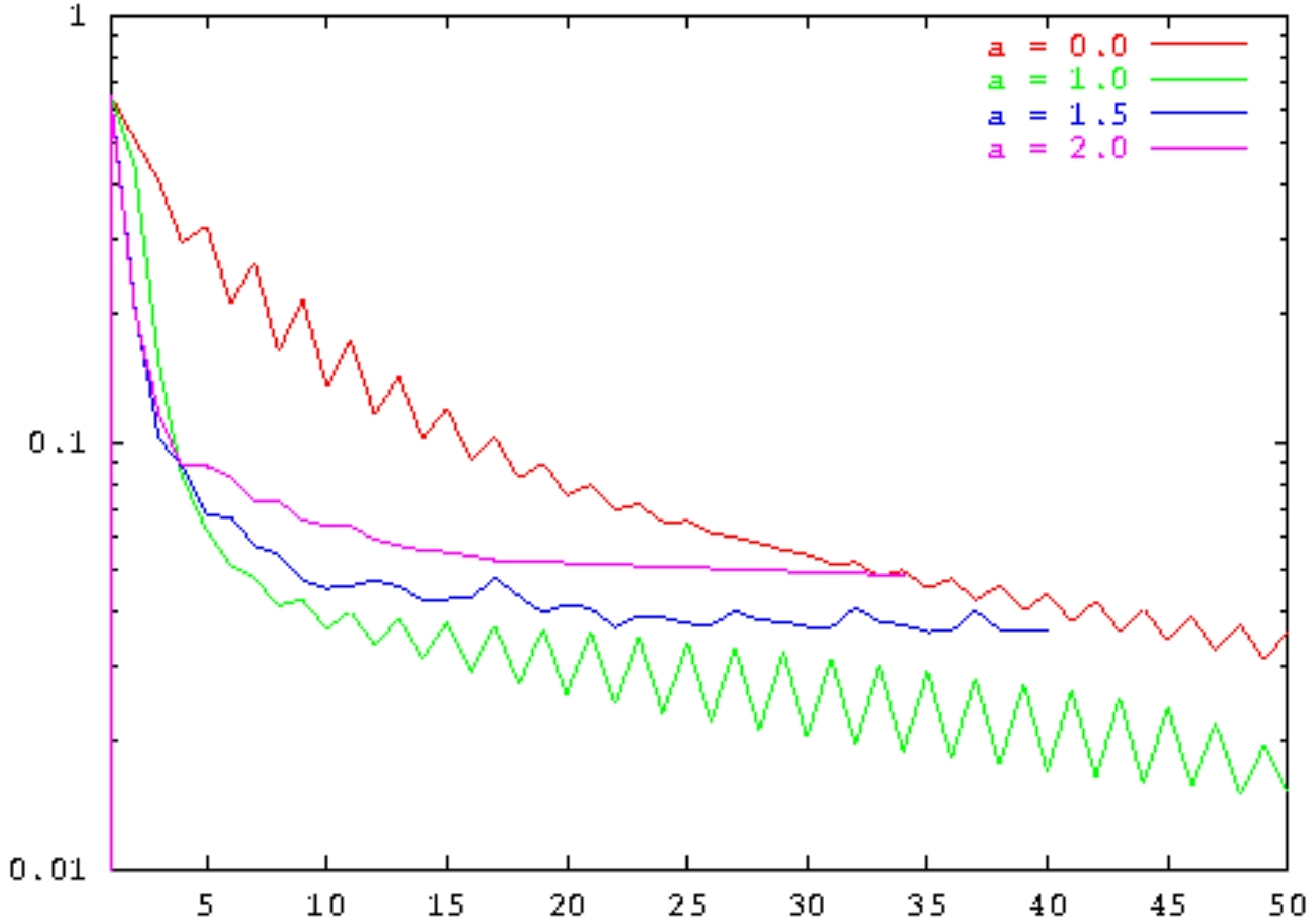
- $B_a = \sum_{k=1}^{\infty} 2^{-k(a+\alpha)}(Q_k - Q_{k-1})$,

where Q_k is a projection on V_k is for $\alpha > 0$,$a > 0$,

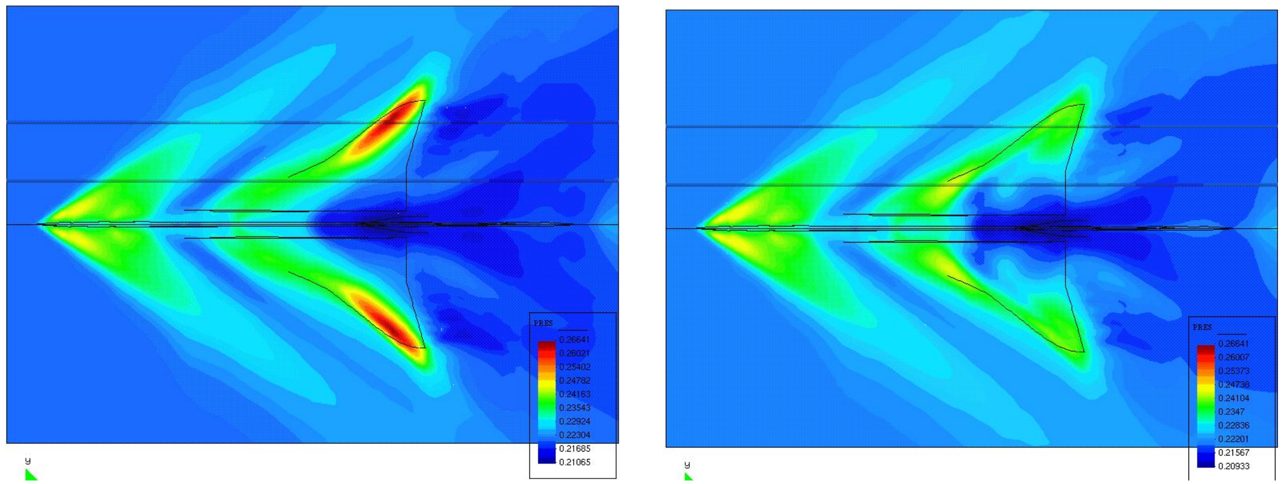
a **compact operator of degree $-a$** .

Courty-Dervieux, Vazquez-Koobus-Dervieux INRIA reports.

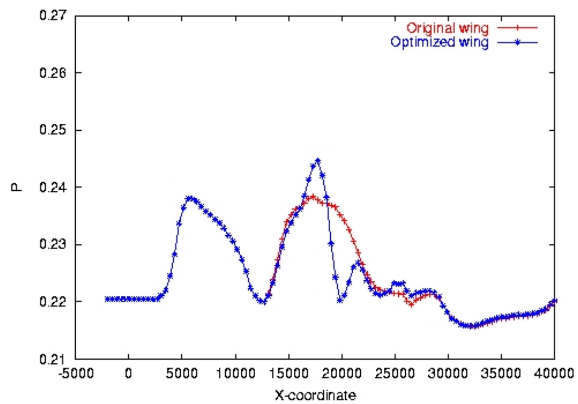
Application of preconditioner B_a for different degrees $-a$



Wing optimization case



Pressure in a plan under aircraft (10 optimization iterations)



A cut of above plan parallel to aircraft axis

Extension to multidiscipline: jig shape optimization

An inportant advantage of the CAD-free shape parameterization in the possible extension to optimization loop where the effect of elastic deformation is also taken into account.

Corrected shape optimization (Euler) and

fluid-Structure steady interaction are solved in turn in the jig-shape optimization.

Vazquez-Koobus-Dervieux, Comp. Engg.05

Back to single-point. Global algorithm:

Part-(ii) aeroelastic inverse problem can be solved simultaneously by relaxation with step-(i):

(a) Solve the fluid+structure system to get an "aeroelastic deformation correction",

(b) To find γ_1 *solution of a single-discipline optimization, corrected by the frozen "aeroelastic deformation correction",*

Step-(b) still needs a two-field adjoint: flow, mesh.

Transformation of Step-(i)

Euler flow model : transpiration can be used.
Step-(b) transforms in:

(b1)- transpiration-based optimization, taking into account the previous aeroelastic deformation-correction,

(b2)- a post-remeshing by mesh deformation producing a mesh fitted with the current domain at rest, ready for aeroelastic analysis.

Then most transpiration error (due to perturbation amplitude) is discarded.

Global algorithm: main options, cont'd

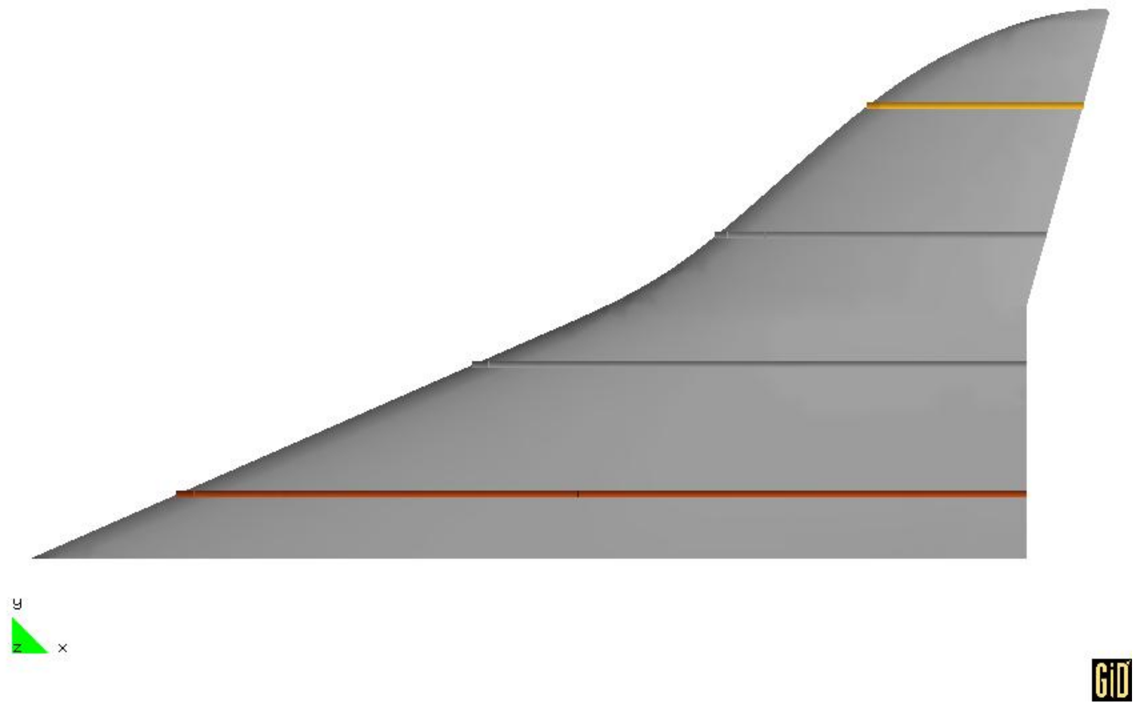
1- Start with an initial shape,

2- Aeroelastic analysis including remeshing (analyser unchanged),

3- Transpiration-based optimization (with correction) $+$ post-remeshing.

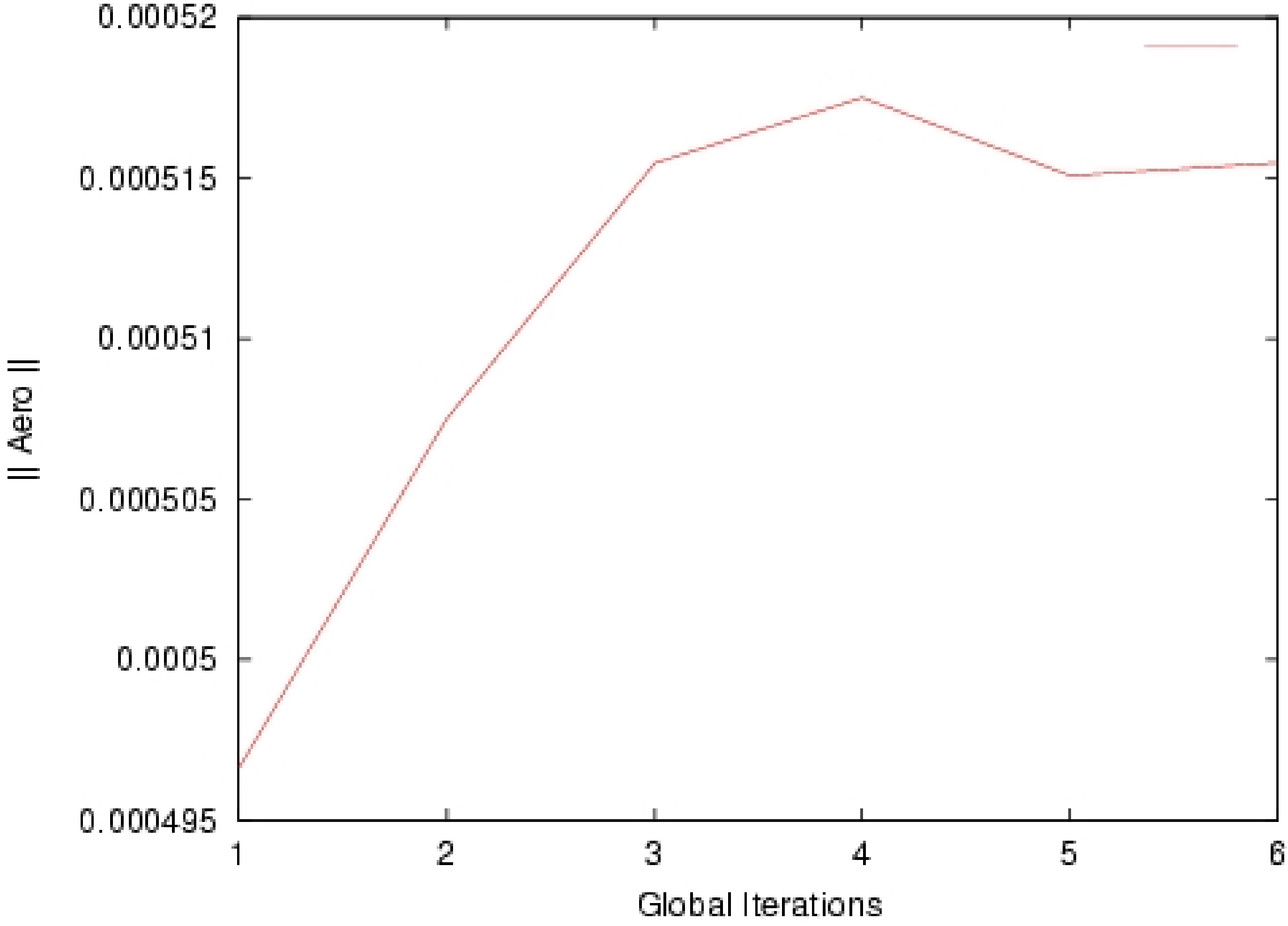
4- If residuals of 2 and 3 are not small, go to 2.

Application to a Delta wing



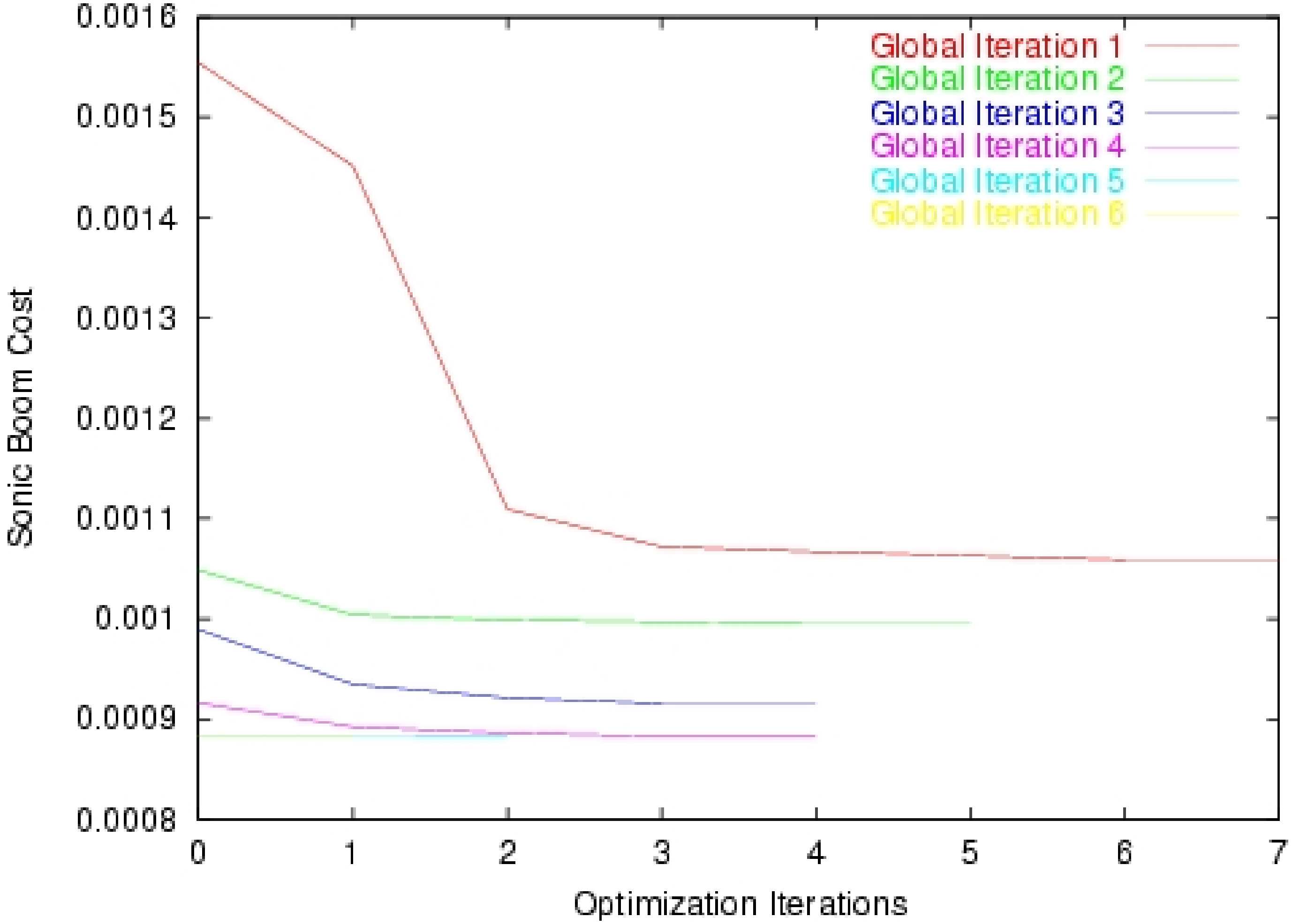
Euler flow, Mach=1.8, Initial incidence angle=3 deg. Fluid mesh (unstructured): 65418 tetrahedra and 12963 nodes, 2408 nodes on the surface (number of design parameters). Structure mesh: thin plate finite element model, 1286 triangular shell elements.

Convergence of aeroelastic phase



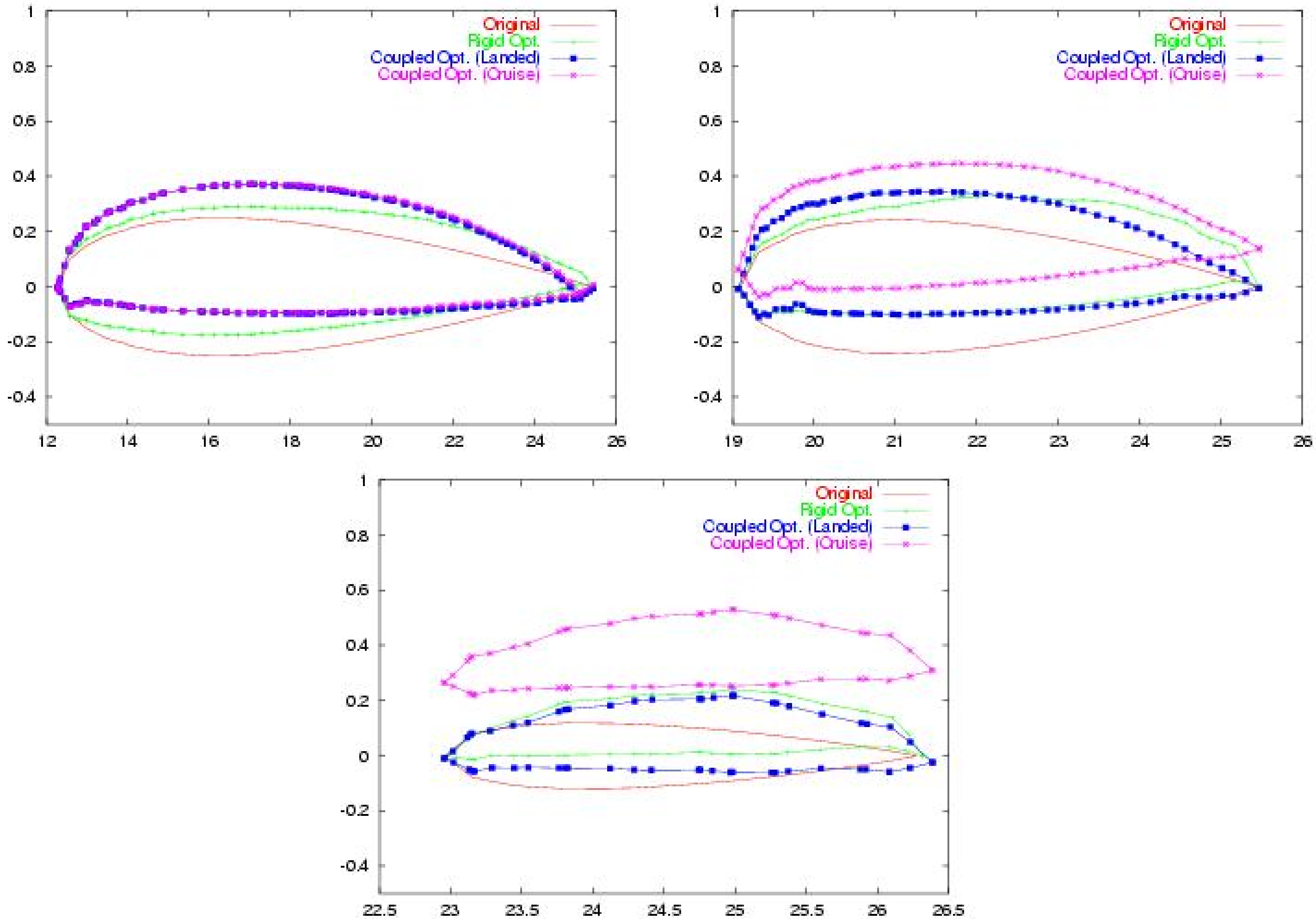
Mean aeroelastic displacement norm/vs global iteration

Convergence to optimum



Cost functional over 6 global iterations

Coupled optimization, results



Dassault Aviation's Supersonic Business Jet. Optimized flexible wing. Clockwise: inboard, mid-wing and outboard cross-sectional cuts

6. NUMERICAL ERROR REDUCTION

Minimize an error-functional $\bar{j}$ with respect to mesh parameter, $\mathcal{M}$.

$$\text{with } \; \Psi_{STATE}(\gamma,U) = 0$$

$$\begin{aligned} \bar{\mathcal{M}} &= \text{ArgMin } \bar{\mathbf{J}}(\mathcal{M},\gamma,U^{exact} - U) \\ \text{with } \; &\Psi_{ERROR}(\mathcal{M},\gamma,U^{exact} - U) = 0. \end{aligned}$$

$$\bar{\mathbf{J}}(U^{exact} - U(\mathcal{M})) \;=\; |J_{exacte}(U^{exact})'.(U^{exact} - U(\mathcal{M}))|^2$$

Mesh parameterization: Riemannian metric

We modelize a mesh as a **continuous medium**, with an anisotropic property, the **local metric** (*):

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

An admissible metric describes a class of equivalence of meshes, any of which satisfies the unit length criterion for every edge AB:

$$l_{\mathcal{M}}(AB) = \int_0^1 \sqrt{{}^t\overrightarrow{AB}\,\mathcal{M}(A+t(B-A))\,\overrightarrow{AB}}dt \ = \ 1\,.$$

(*)(George, Hecht,..., Fortin, Habashi, Vallet,...)

Towards implicit error reduction

An a priori error analysis leads to the model:

$$\frac{\partial \Psi}{\partial W}(W,d)Y = G(W,d),$$

W: flow variable, Y approximation error, d is the mesh local density to be ajusted for minimizing the error-functional $\bar{j}$.

$$\begin{array}{rcl}
\dfrac{\partial \Psi}{\partial W}(W,d)Y - G(W,d) & = & 0 \\
\left(\dfrac{\partial \Psi}{\partial W}(W,d)\right)^{*}\Pi - \dfrac{\partial \bar{J}}{\partial Y}(Y,d) & = & 0 \\
\bar{j}' = -\left(\dfrac{\partial(\frac{\partial \Psi}{\partial W}Y - G)}{\partial d}(W,d)\right)^{*}\Pi + \dfrac{\partial \bar{J}}{\partial d}(Y,d) & = & 0
\end{array}$$

Reverse AD applies to stationarity system assembly

$$\left(\frac{\partial}{\partial d}\left(\frac{\partial \Psi}{\partial W} Y\right)\right)^{*} \Pi$$

- can be assembled in two steps, "reverse-on-tangent fashion".

Also equal to:

$$\frac{\partial}{\partial W}\left(\frac{\partial \Psi}{\partial d}^{*} \Pi\right) Y$$

which can be assembled in two steps, "tangent-on-reverse fashion".

CONCLUSIONS

Large scale design:

combines:
program differentiation,
differentiable optimisation,
Functional Analysis based preconditioning
and the re-use of existing, physics-adapted methods and software.

Associated problems of discretization accuracy by mesh design can also be addressed following these guidelines.