Management and control of the anisotropic unstructured meshes Applications in aerodynamics

Institute of Aerodynamics and FLow Technology

(DLR)

Youssef Mesri

INRIA, project-team Smash, 2004 Route des Lucioles, BP 93

06902 Sophia Antipolis Cedex, France

July 16th, 2007

Table of content

Overview

Make a fast turbulent Navier-Stokes computations over Falcon$7X$ mesh.

$\Downarrow$

- ☐ Multigrids.
- ☐ Parallelism.

FIG. 1 – Falcon$7X$ (Dassault-Aviation constructor)

Review of multigrid method convergence characteristics

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

$$A\,u = -\Delta\,u = f \text{ in } [0,1]^d \subset \mathbb{R}^d \Longrightarrow \text{(by usual 2nd-order F.D.) } A_h\,u_h = f_h$$

$$A_h = \frac{1}{h^2}\,\mathrm{Trid}_x(-1,2,-1) \oplus \mathrm{Trid}_y(-1,2,-1) \oplus \ldots$$

$$h = \frac{1}{N}\,, \quad \mathrm{N} = N^d$$

- **Approximation error :**

$$\|u_h - u\| = O(h^2)$$

- **Modal analysis :** eigensystem made of discrete Fourier modes ; 1D :

$$s_i^{(m)} = \sqrt{2h}\,\sin i\alpha^{(m)}\,,\ \alpha^{(m)} = \frac{m\pi}{N} = m\pi h\,,\ \lambda^{(m)} = \frac{2 - 2\cos\alpha^{(m)}}{h^2}$$

- **Condition number**

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{4}{tan^2\frac{\pi}{2N}} = O(N^2)$$

Basic relaxation method : Jacobi iteration

$$u_h^{n+1} = u_h^n - \tau^* \left(A_h \, u_h^n - f_h \right) \, , \; \tau^* = \left(\frac{\lambda_{\max}+\lambda_{\min}}{2}\right)^{-1}$$

- **Approximation error :** controlled by grid size h :

$$\|u_h - u\| \sim C_A \times h^2$$

- **Iterative error** for Jacobi-type solution of the system, controlled by iteration count n ; if $\kappa = \lambda_{\max}/\lambda_{\min}$ (condition number of matrix A_h) :

$$\|u_h^n - u_h\| \sim C_I \times \rho(h)^n \, , \quad \rho(h) = \frac{\kappa - 1}{\kappa + 1} = 1 - \frac{2}{\kappa} + \ldots = 1 - c\,h^2 + \ldots$$

- **Termination criterion and computation cost estimate :**
  - cost of one iteration (matrix-vector product) : N^d
  - best attainable error $\mathcal{O}(h^2)$, no reason to iterate more :

$$\rho(h)^n \sim h^2 \Longrightarrow n \sim N^2 \log N \Longrightarrow$$

$$\text{COST}_{\text{JACOBI}} \sim N^{d+2} \, \log N = \text{N} \times N^2 \, \log N$$

Nested Iteration, or progressive mesh enrichment

Kronsjö-Dahlquist (1971). *BIT* 12, 1972.

Fine :				↗	$\mathcal{M}_{h/4}$	. . .
Medium :		↗	$\mathcal{M}_{h/2}$			
Coarse :	$\mathcal{M}_h$					
	Solve	Prolongate	**Smooth**	Prolongate	**Smooth**	. . .

- Efficiency : depends on the design of :
  - appropriate smoothers,
  - adequate iterative termination criterion, and
  - accurate-enough interpolation operators
- Cost reduction : only by a factor of $\log N$ (i.e. number of levels)
- In nonlinear context : improved robustness

- **Ideal Two-Grid Algorithm** : $\rho \leq B_1$; example : *V-cycle*

Fine Grid $\mathcal{M}_h$: smoothing only ↘ Coarse Grid $\mathcal{M}_{2h}$: exact solve ↗ Fine Grid $\mathcal{M}_h$: smoothing only

- **Multigrid** : by extension : $\rho_{\text{MG}} \leq B$

$$\rho_{\text{MG}}^n \sim h^2 \Longrightarrow n \sim \log N \Longrightarrow$$

$$\boxed{\text{COST}_{\text{MG}} \sim \text{N} \times \log N}$$

Full Multigrid Method, “FMG”

Combines Nested Iteration with MG-cycle :

Fine :				↗	$\mathcal{M}_{h/4}$ $\downarrow\uparrow$	
Medium :		↗	$\mathcal{M}_{h/2}$ $\downarrow\uparrow$		$\mathcal{M}_{h/2}$ $\downarrow\uparrow$	
Coarse :	$\mathcal{M}_h$		$\mathcal{M}_h$		$\mathcal{M}_h$	
	Solve (1-grid)	Prolongate	**MG-cycle** (2-grid)	Prolongate	**MG-cycle** (3-grid)	...

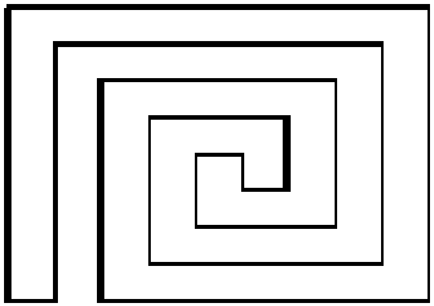
- ☐ Linear complexity : $O(\mathrm{N})$ $(\mathrm{N} = N^d)$ ONLY !
- ☐ Introductory Texts : Briggs (SIAM, 1991), Wesseling (John Wiley, 1991), Désidéri. (Editions Hermès, Paris, 1998, *in French*)

Mesh generation & mesh adaptation

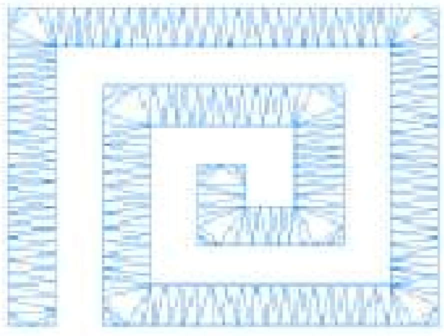
INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE
INRIA

2D example :

- From a surface (boundary) mesh (CAD)



- Construct a mesh for all domain

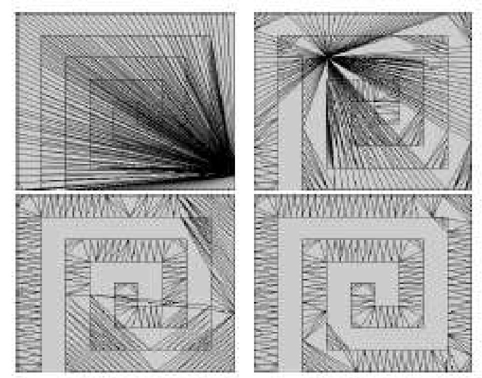


For unstructured 3D meshes (tetrahedron) :

Mesh generator	Advantages	Disadvantages
Octree	- good inside - tree structure	- many nodes - little anisotropy
Frontal	- excellent quality	- non convergence
Delaunay	- very fast	- border difficult to genera - degenerate elements
By optimization	- very robust - independent dimension (4D)	- slower

We use a topological mesh generator : MTC (developed at CEMEF)

$$\left\{\begin{array}{l}\sum_{T\in\mathcal{T}}|T| > |\Omega| \\ c_0\frac{|T|}{h(T)^d} \simeq 0\end{array}\right.$$

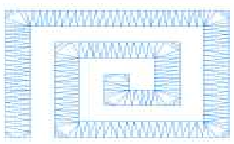


Local topological optimization

$$\Downarrow$$

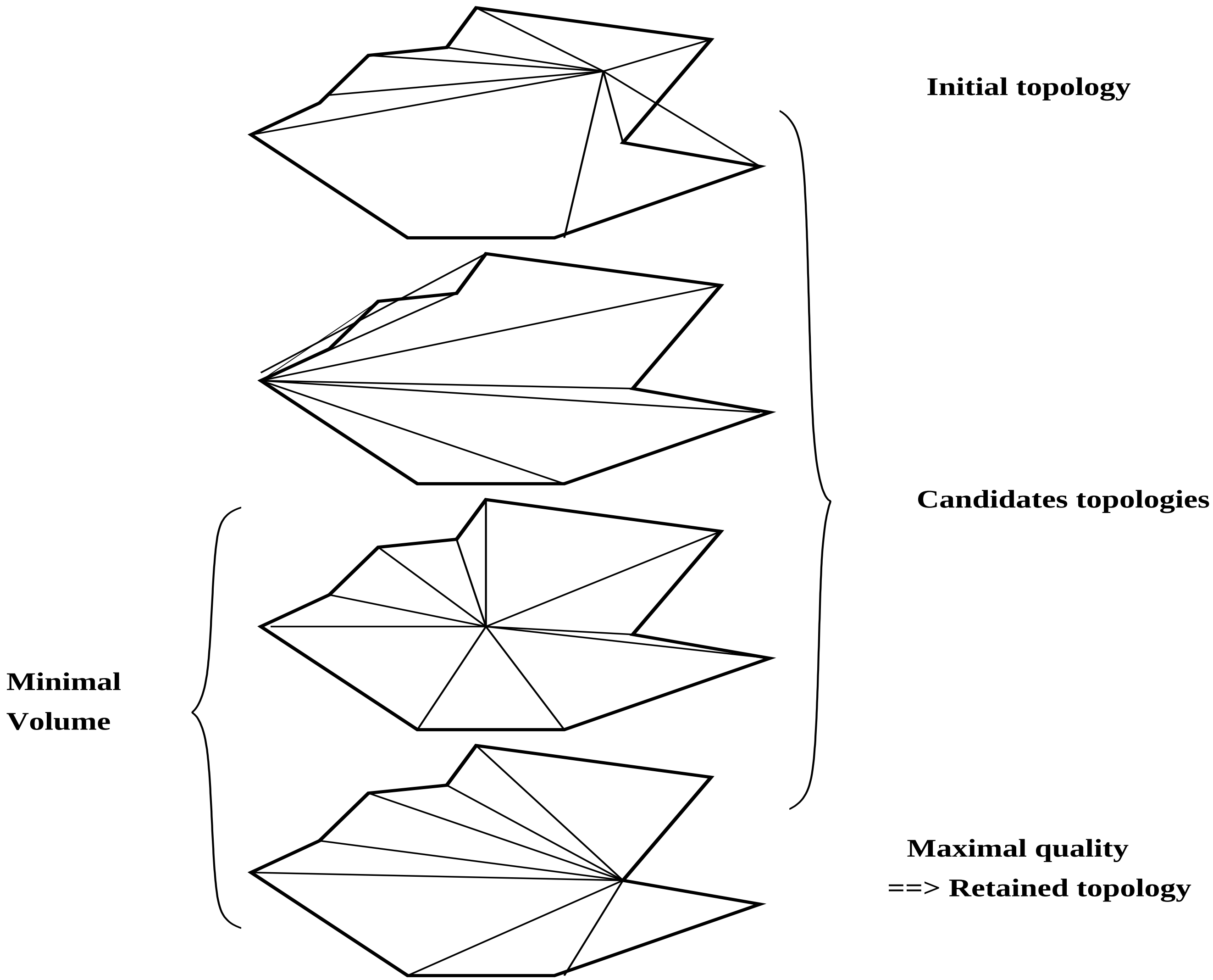
Theorem : The equality $\sum_{T\in\mathcal{T}}|T| = |\Omega|$ is reached only by the valid meshes.

$$\left\{\begin{array}{l}\sum_{T\in\mathcal{T}}|T| = |\Omega| \\ c_0\frac{|T|}{h(T)^d} \simeq 1\end{array}\right.$$



Local topological optimization

Around each node and edge :



MTC : mesh generator

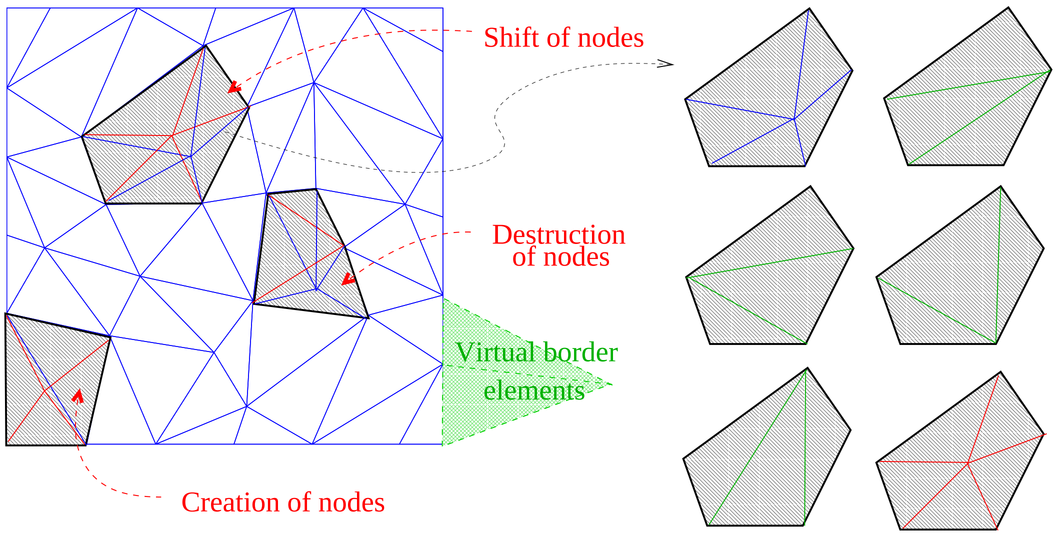
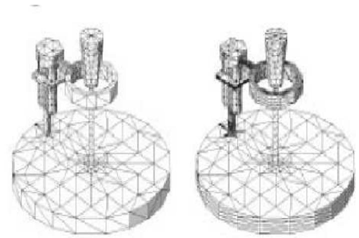


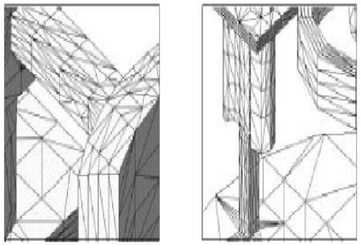
FIG. 2 – Local mesh optimization process in MTC

State of art : without metrics

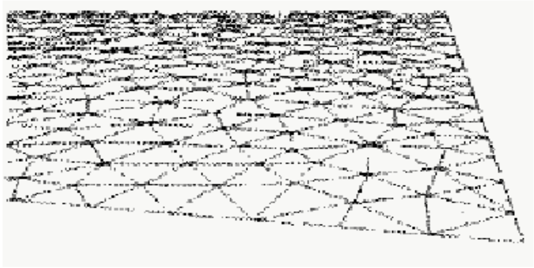
For anisotropic mesh, one can proceed :

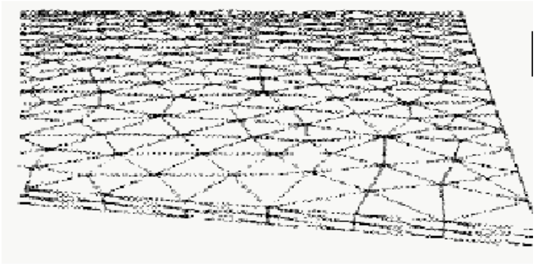
- By one-directional refinement [Garimella, 1998]

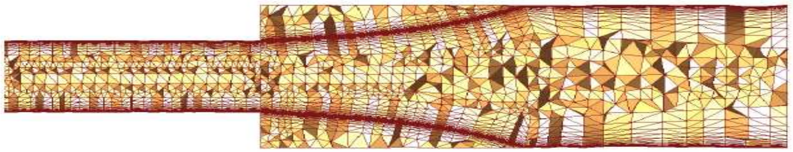




- By extrusion from the surface (border or median) [Knockaert,2001] [Garimella & Shephard,1998]





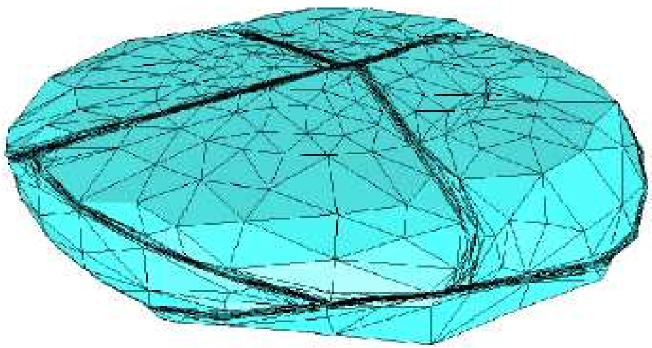


→ Problem of connections in the curved zones

Adaptation to a metric field is more flexible :

In 3D, a metric = a rotation R + 3 lengths $h_1 h_2 h_3$ (sizes desired in the principal directions of R)

$$M = R^T . \begin{pmatrix} h_1^{-2} & & \\ & h_2^{-2} & \\ & & h_3^{-2} \end{pmatrix} .R \qquad (1)$$



Modify mesh generator with respect to metric :

- The form criteria : $c_0 \frac{|T|}{h(T)^d}$ becomes $c_0 \frac{|T|_M}{h_M(T)^d}$

Where, we change the volume computation : $|T|_M = |T|\sqrt{(}det(M))$

the lengths computation : $h_M(T) = h(T)$ in the norm $|.|_M$

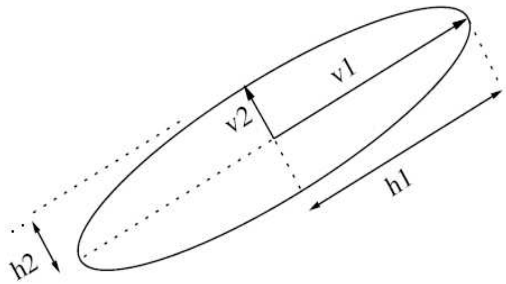
- and just all ! the optimization strategy is the same.

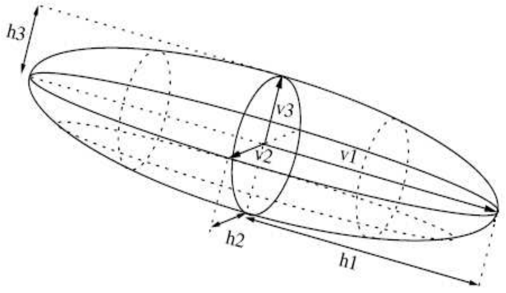
Meshes and Metrics

Definition :
A Metric (also Tensor) $\mathcal{M}$ in $\mathcal{R}^d$ is a symmetric definite positive real matrix with d the dimension of space. It is then a $d \times d$ invertible matrix, its eigenvalues are real and positive, and its eigenvectors form a orthogonal basis of $\mathcal{R}^d$. such that

$$\mathcal{M} = V(\mathcal{M})^t \Lambda(\mathcal{M}) V(\mathcal{M})$$

where $V(\mathcal{M})$ denotes the orthonormal matrix corresponding to the eigenvectors of $\mathcal{M}$ while $\Lambda(\mathcal{M})$ is the diagonal matrix of its eigenvalues.





Natural metrics : metric associated with a simplicial element

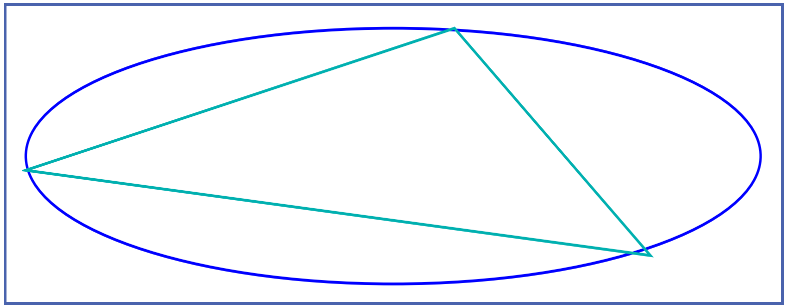
We seek for a metric in which the edges length of an element T are equal :

$$x_{kl}^T M_T x_{kl} = 1 \forall (k,j) \in T$$

Proposition :

$$M_T = C_0 \left(\sum_{(k,l) \in E(T)} x_{kl} . x_{kl}^T \right)^{-1}$$

where $C_0 = (d+1)/2$



Nodal metric : metric associated to the mesh nodes

- Define a P1 metric field on the background mesh
- The anisotropic mesh generators currently in use [Coupez, George] take as input a metric defined only over nodes
- It is easier to transport P^1 fields during mesh generation process than P^0 fields.
- Define from a P^0 tensor field a P^1 tensor field
  - Conserve length of edges and the stretching directions.

Nodal metric (1)

Interpolation of element metrics to node :

$$M_i = \Big(\frac{1}{card(T(i))} \sum_{T_k \in T(i)} M_{T_k}^{-\frac{1}{2}}\Big)^{-2} \tag{2}$$

$T(i)$: the set of elements that contain the node i

Corrected interpolation of element metrics to node :

$$M_i^* = V_i . D_i^* . V_i \tag{3}$$

where V_i : the eigenvectors matrix of the original matrix M_i and

$$D_i^* = \Big(\frac{1}{card(T(i))} \sum_{T_k \in T(i)} D_{T_k}^{-\frac{1}{2}}\Big)^{-2} \tag{4}$$

is the modified diagonal matrix.

Interpolation as computing geodesics :

the intrinsic mean of points in A is defined as a minimum of the function j_A, that is,

$$\mu = \text{argmin}_{x \in \mathcal{E}} j_A(x) \tag{5}$$

where

$$j_A(x) = \frac{1}{2n} \sum_{i=1}^{n} d(x, x_i)^2$$

and d is geodesic distance on $\mathcal{E}$.

$d(M,N) = Trace(diag(M^{-1/2} N M^{-1/2}))$

The gradient of j_A is given by

$$\nabla j_A(x) = -\frac{1}{n} \sum_{i=1,n} Log_x(x_i)$$

where $Log_x(x_i) = x^{1/2} log(x^{-1/2}.x.x_i^{-1/2}) x^{1/2}$

Nodal metric (3)

and thus the intrinsic mean of a set of tensors can be computed by the following gradient descent algorithm [Pennec, 1999] :

Algorithm 3 : Intrinsic Mean of Tensors

Input : $M_1, \cdots, M_n \in SPD$ space

Output : $R \in SPD$ space, the intrinsic mean

$$R_0 = M_1$$

Do

$$X_i = \frac{1}{n} \sum_{k=1,n} Log_{R_i}(M_k)$$
$$R_{i+1} = Exp_{R_i}(X_i)$$

While $(X_i, X_i)_{R_i} > \epsilon$

Nodal metric (4)

Interpolation in Log-Euclidean space [Arsigny,2005] :

Definition Let $M_1, M_2 \in Sym_*^+(n)$ and $\lambda \in \mathcal{R}$. a logarithmic product of M_1, M_2 is defined as follow :

$$M_1 \odot M_2 := exp(log(M_1) + log(M_2))$$

and a logarithmic scalar multiplication $\otimes$ is defined by :

$$\lambda \otimes M_1 := exp(\lambda.log(M_1)) = M_1^\lambda$$

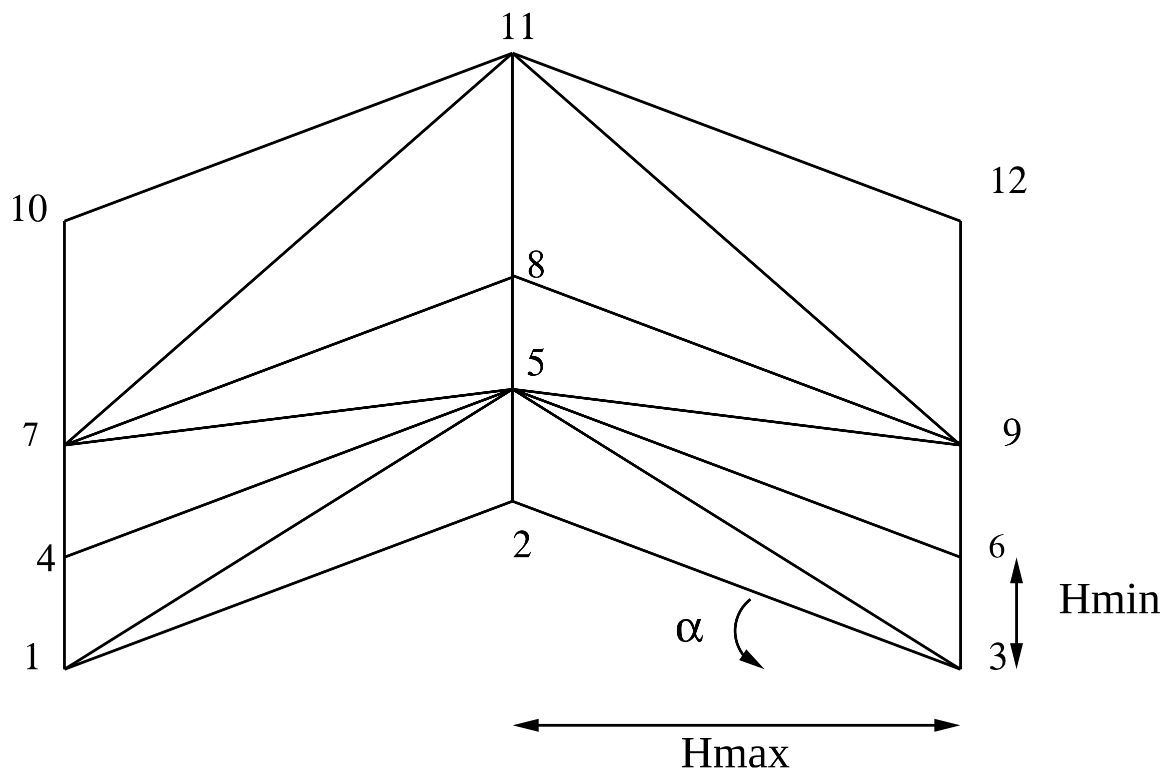
Proposition. : $(Sym_*^+(n), \odot, \otimes)$ is a vector space structure.

Proposition. : Let $(M_i)_1^N$ be a finite number of SPD matrices. Then their Log-Euclidean Fréchet mean exists and is unique. It is given by :

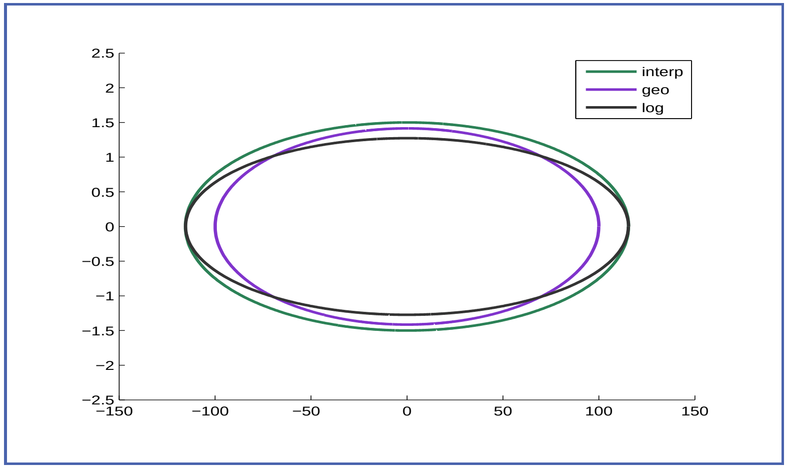
$$M_{LE}(M_1, \cdots, M_N) = exp(\frac{1}{N}\sum_{i=1}^{N} log(M_i))$$

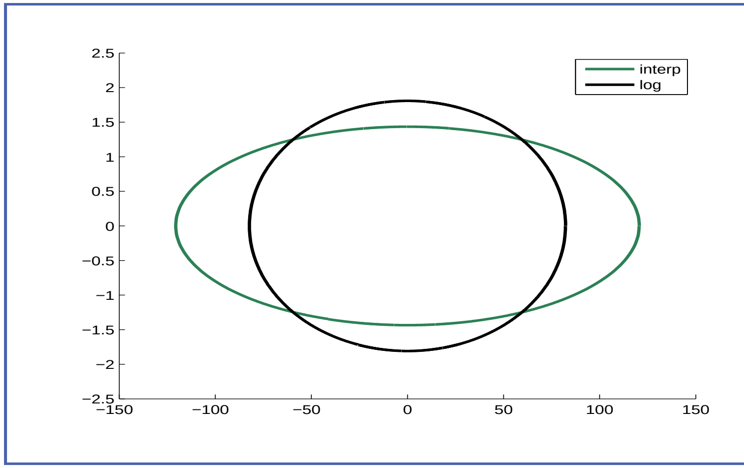
Comparison between the interpolation strategies (1)

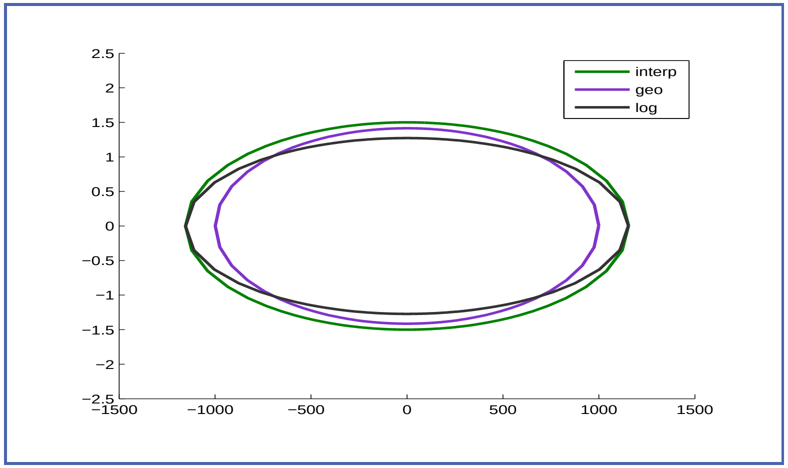
Recalling that our main problem concerns highly stretched meshes used in boundary layers, we first compare these three strategies on a 2-D model boundary layer mesh.

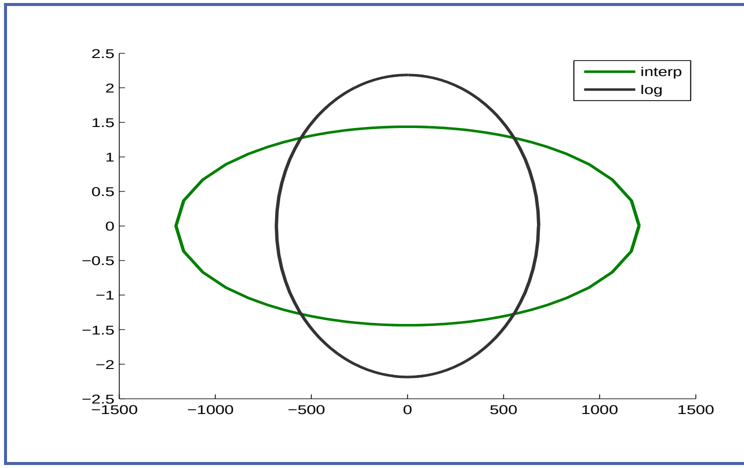


Comparison between interpolation strategies (2)









Anisotropic coarsening algorithm

The target metric is constructed with respect to the following stages :

- Generate on each node of the finest mesh an initial nodal metric that reflects the size and stretching of elements belonging to this mesh.

- Modify the initial metric to establish a corresponding coarsened mesh metric. This is done by modifying the eigenvalue λ_i associated to the metric, which will modify the mesh size in the desired direction which is the corresponding eigenvector V_i.

- Provide the background mesh and the desired metric field to the MTC mesher.

Coarsening algorithm

Algorithm 1 : Anisotropic coarsening algorithm

Input : $\lambda_1, \cdots, \lambda_d$ eigenvalues of a $\mathcal{M}_k$

Output : Update $\lambda_1, \cdots, \lambda_d$ coarsened eigenvalues

For each $i \in \mathcal{N}$

Set : $h_k^i = (\lambda_k^i)^{-1/2}$, $k = 1, \ldots, d$, $h_1^i \leq \cdots \leq h_d^i$ and $h_0 = C_{cf}.h_1$.

$$h_k^i = \max\big(h_k^i, \min\left(C_{cf} \cdot h_k^i, h_{k-1}^i\right)\big) .$$

Define new eigenvalues $\lambda_k \leftarrow h_k^{-2}$.

End for.

Modified anisotropic coarsening algorithm (1)

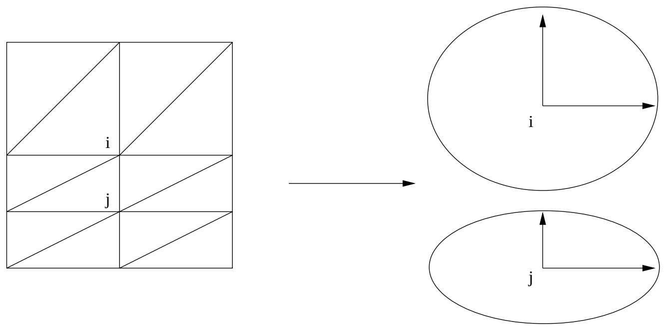


FIG. 4 – metrics at interface points between two different mesh anisotropy zones

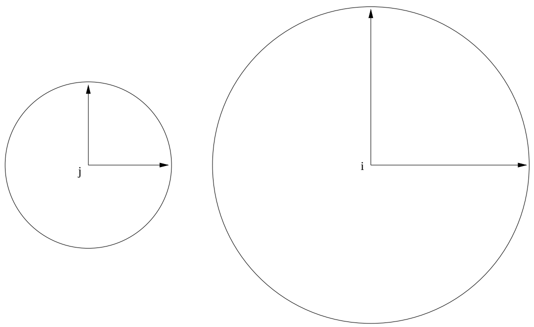


FIG. 5 – Conflict metrics

Metric smoothing : [Borouchaki & George, 1997]

Modified anisotropic coarsening algorithm (2)

Algorithm 2 : Modified anisotropic coarsening algorithm (MACA)

Input : $\lambda_1, \cdots, \lambda_d$ eigenvalues of a $\mathcal{M}_k$

Output : Update $\lambda_1, \cdots, \lambda_d$ coarsened eigenvalues

For each $i \in \mathcal{N}$

Set : $h_k^i = (\lambda_k^i)^{-1/2}$, $k = 1, \ldots, d$, $h_1^i \leq \cdots \leq h_d^i$.

Set $h_0 = C_{CF} \cdot h_1$.

Store the initial size : $h_{k,\text{old}}^i = h_k^i$.

// compare the current node size specifications with its neighbors

if $\max\left(h_d^i, \min\left(C_{cf} \cdot h_d^i, h_{d-1}^i\right)\right) = C_{cf} \cdot h_d^i$

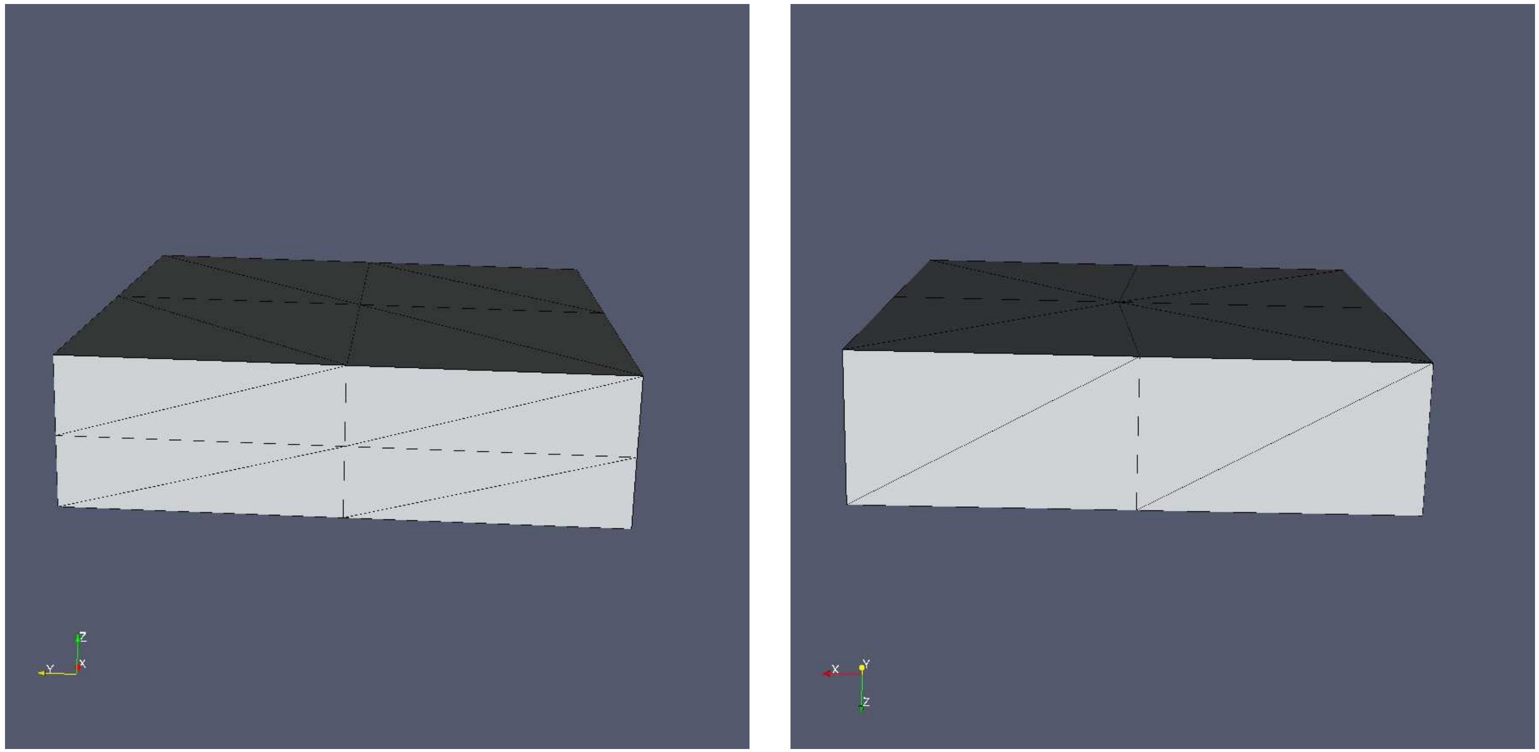
and $\exists j_0 < i \in V(i)$ such as $h_1^{j_0} < h_1^i$. then $h_k^i = h_{k,old}^i, k = 1, \ldots d$

Define new eigenvalues $\lambda_k \leftarrow h_k^{-2}$.

End for

Applications

Coarsening synthetic mesh



(a)Baseline mesh (27 nodes) (b)Coarsened mesh (18 nodes)

FIG. 6 – Semi-coarsening strategy on a synthetic boundary layer mesh

M6 wing test case (1)

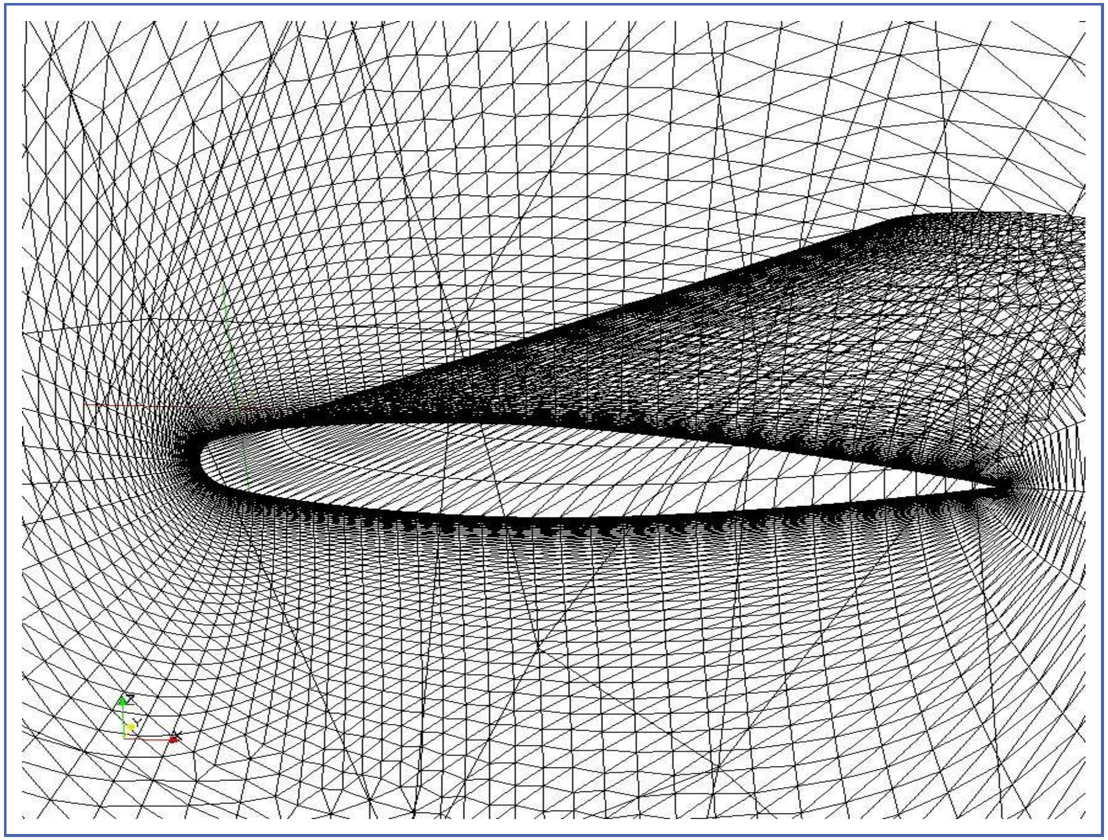


FIG. 7 – finest mesh

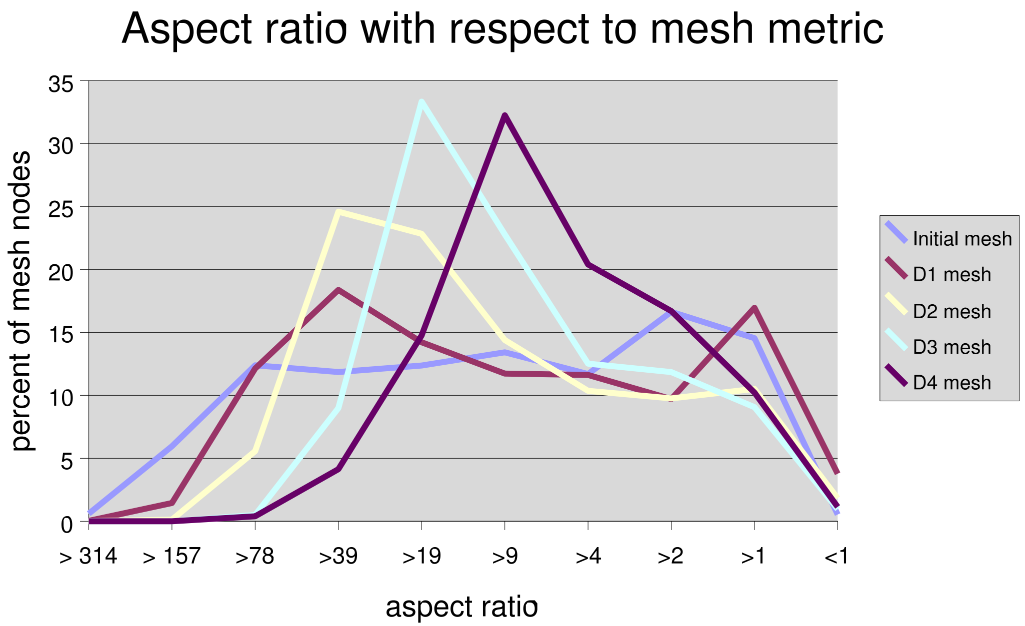
M6 wing test case (3)

	h-interpolation		log-euclidean	
	Num. of nodes	$\mathcal{C}$ ratio	Num. of nodes	$\mathcal{C}$ ratio
initial mesh	67866			
1st coarsened mesh	30961	2.19	39010	1.74
2nd coarsened mesh	15123	1.99	22264	1.75
3rd coarsened mesh	8507	1.77	13246	1.68
4th coarsened mesh	5213	1.63	8323	1.59

Comparison of h-interpolation and log-euclidean methods on the M6 mesh.

$\Rightarrow$ We use in the sequel only the h-interpolation

M6 wing test case (2)



Falcon test case (1)

Falcon$7X$ mesh provided by Dassault-Aviation as a representative example of the mesh used for turbulent Navier-Stokes computations.

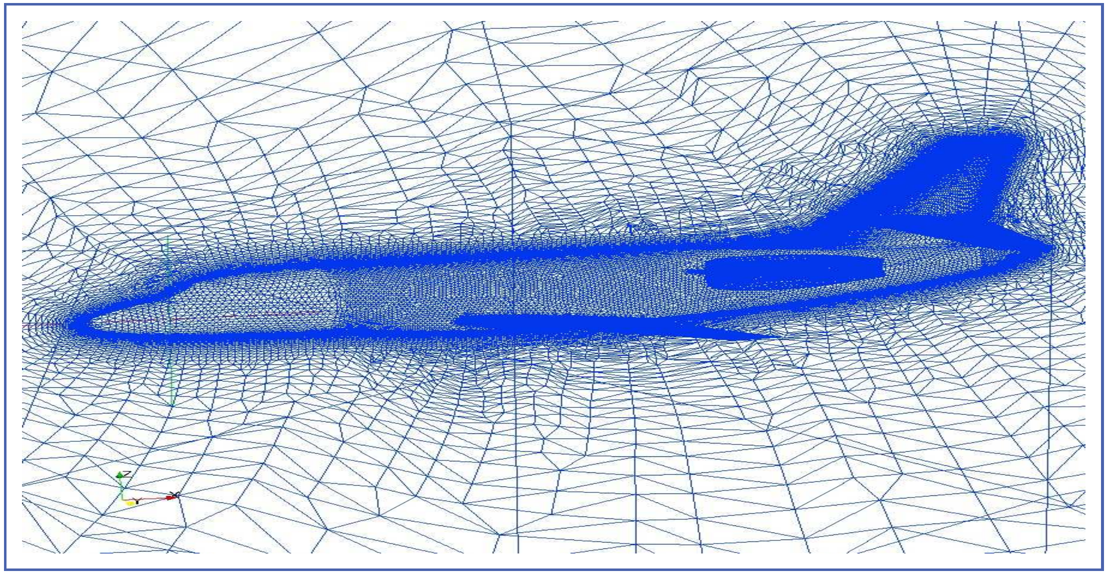


FIG. 8 – finest mesh (2M nodes, 11M elements and more than 500K as aspect ratio) !

Falcon test case (2)

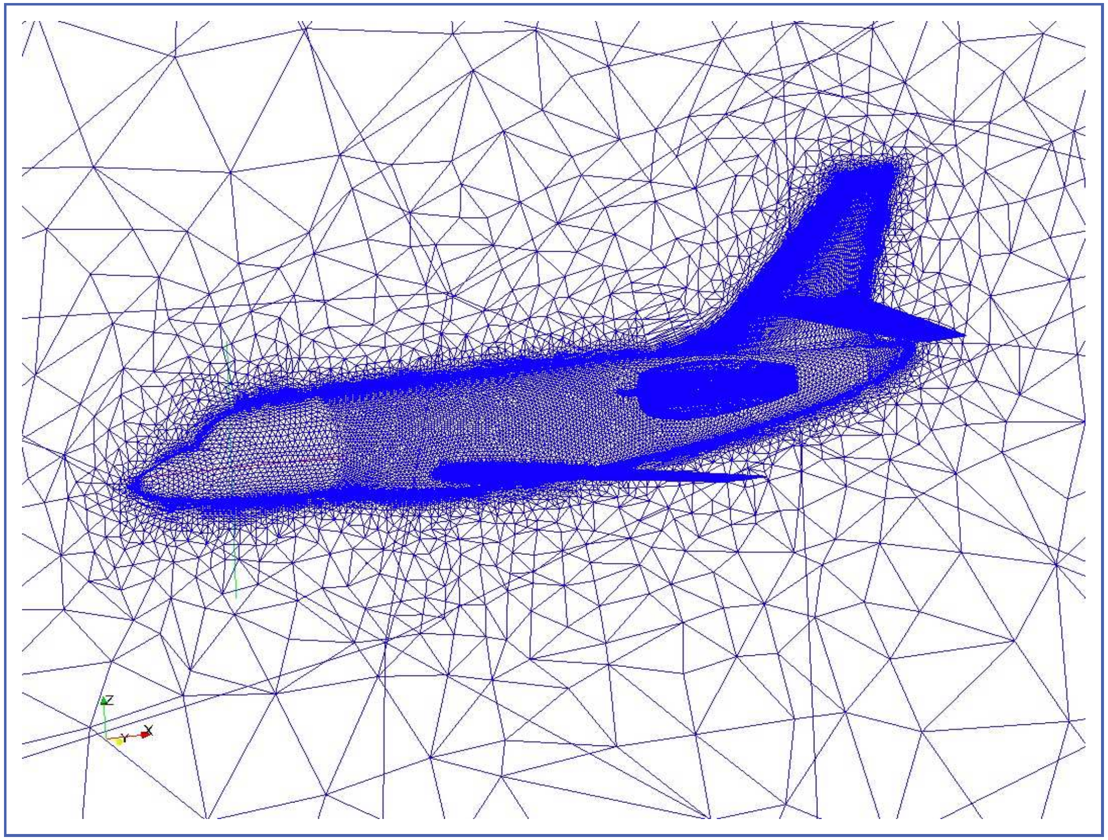


FIG. 9 – first coarsened mesh

Falcon test case (3)

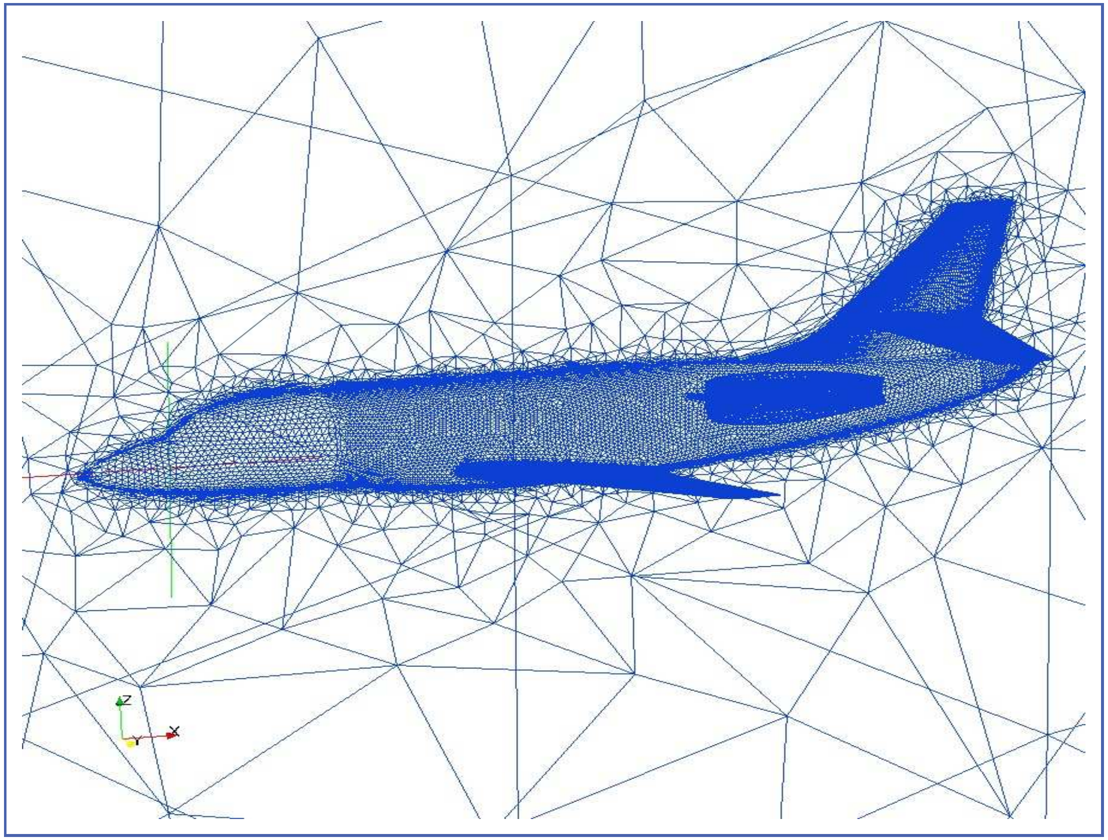


FIG. 10 – second coarsened mesh

Falcon test case (4)

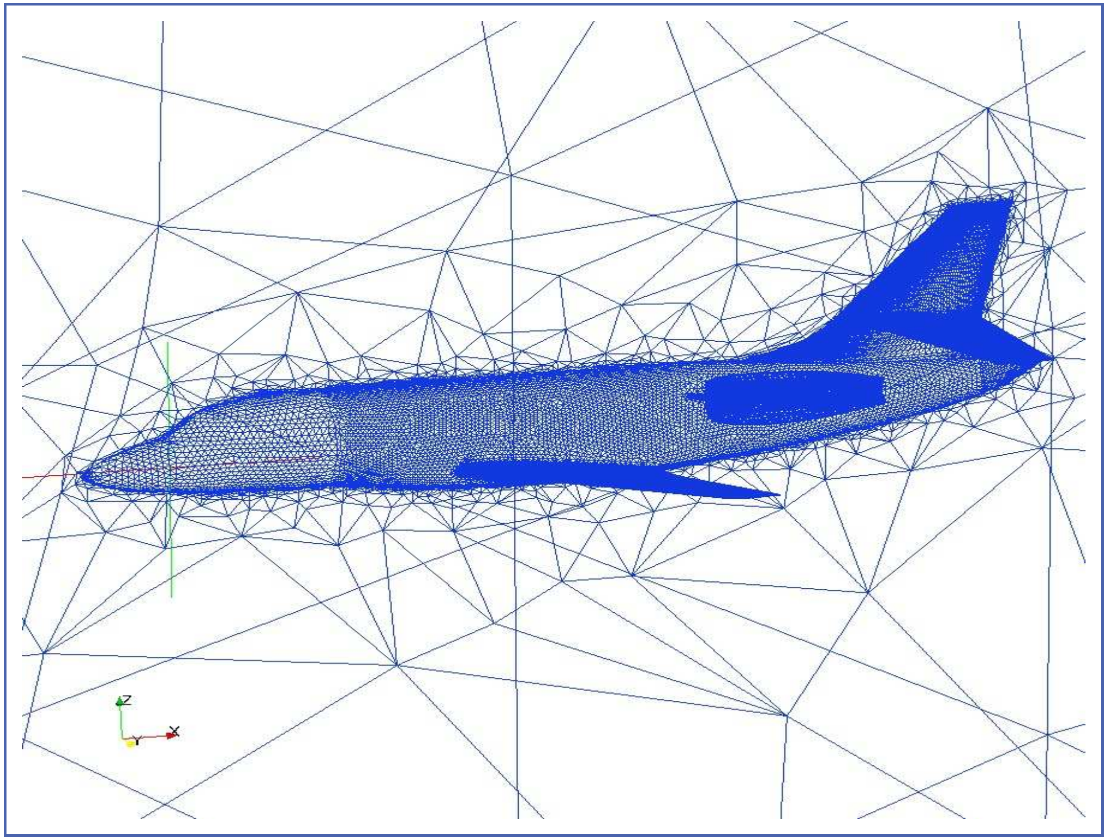


FIG. 11 – third coarsened mesh

Falcon test case (5)

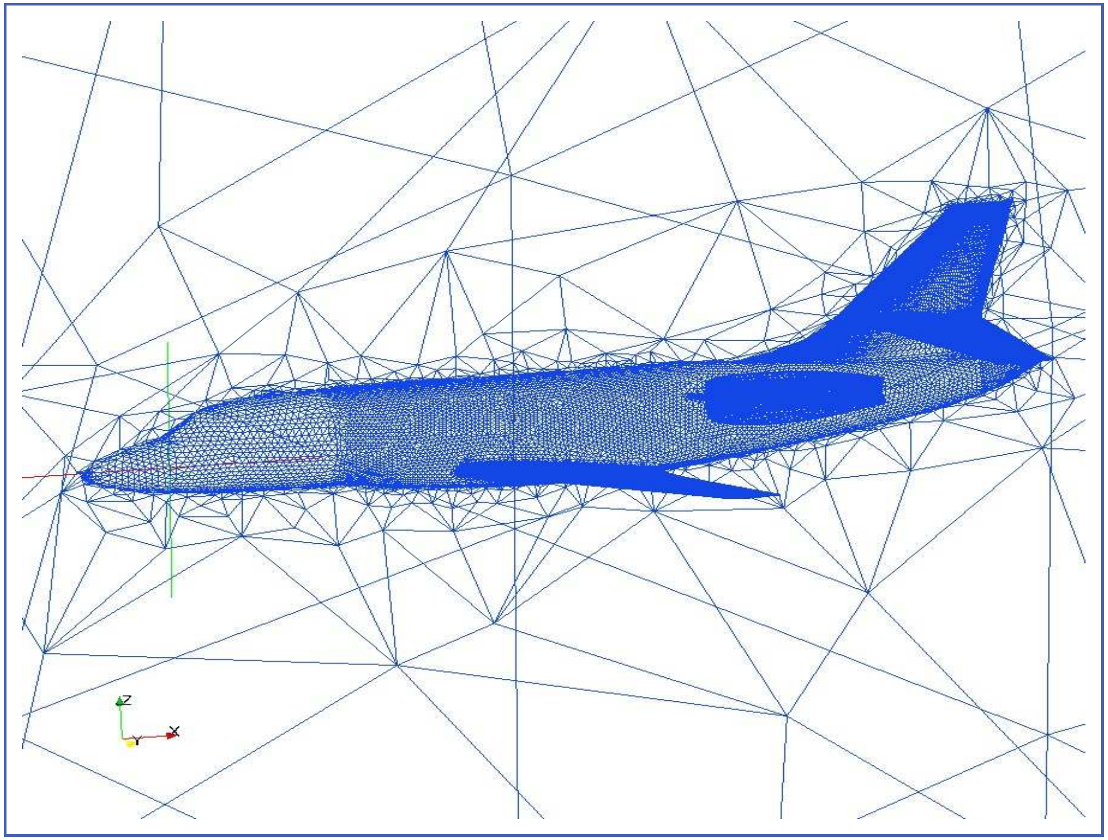


FIG. 12 – fourth coarsened mesh

Falcon test case (6)

	Number of nodes	coarsening ratio
initial mesh	2008248	
first coarsened mesh	779558	2.57
second coarsened mesh	338762	2.3
third coarsened mesh	218850	1.54
fourth coarsened mesh	139522	1.56

Table of initial and coarsened meshes, number of nodes and coarsening ratio

Falcon test case (5)

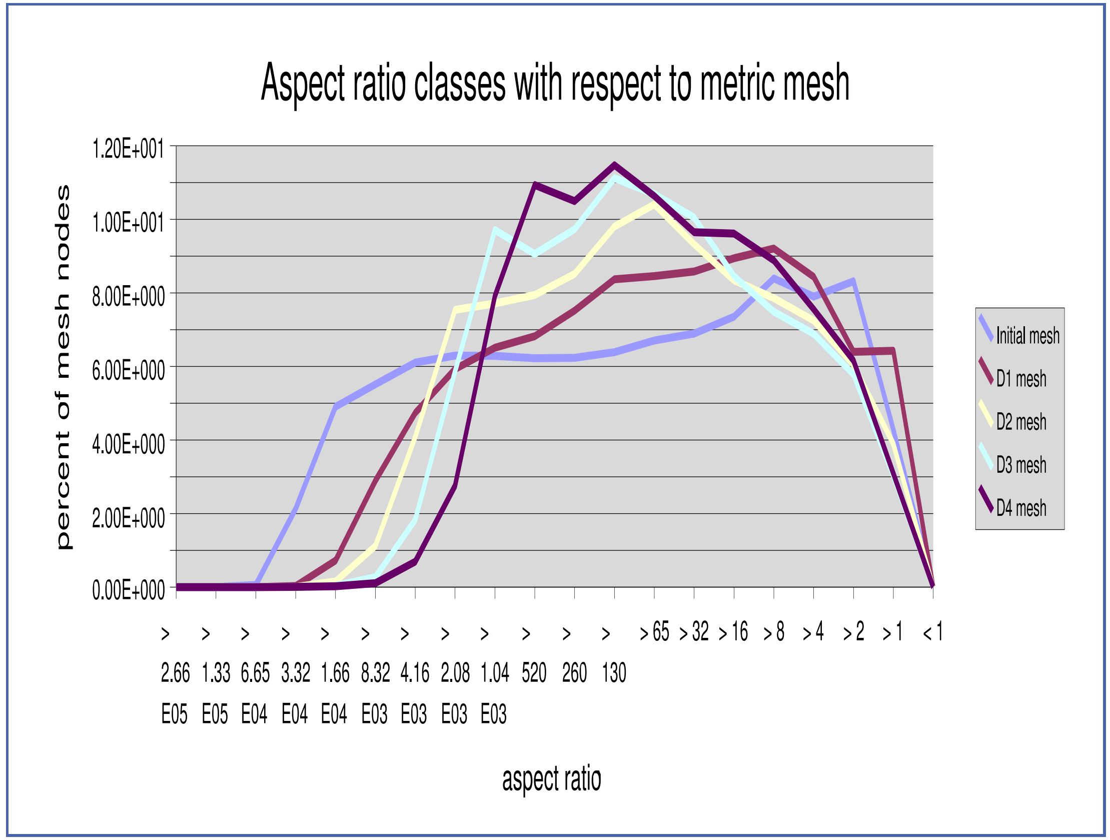


FIG. 13 – Number of vertices (in % of the total number of nodes) belonging to the correspond Aspect ratio class

Synthesis

For mesh generation and adaptation :

- Generate unstructured anisotropic 3D meshes for multigrid methods is not a challenge :
  - Local optimization method : robust
  - Complex geometries with height aspect ratio (10^5) are manageable
- It is possible to construct metrics :
  - Natural metrics to localize anisotropic specifications of the fine mesh.
  - Target metrics to construct a new coarsened mesh.
  - A C++/MPI parallel code is delivered to the industrial partner (Dassault-Aviation)

Table of content

Make an efficient shape optimization process over supersonic jets.

⇓

- ☐ Mesh adaptation.
- ☐ Shape optimization.
- ☐ Both.

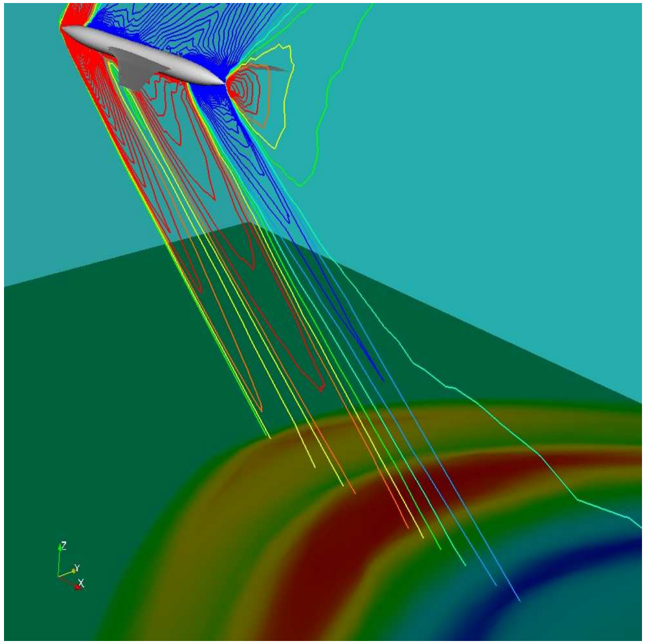


FIG. 14 – Pressure contours over the HISAC geometry

Design method

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

Review of Design methods (1)

- Theory of control of PDE (of the flow) by boundary control (the shape).
  - Continuous Adjoint approach. [Jameson,1988]
    $\Rightarrow$ Continuous PDE $\longrightarrow$ continuous gradient $\longrightarrow$ discrete gradient
  - Discrete exact gradient. [Giles,2001]
    $\Rightarrow$ Continuous PDE $\longrightarrow$ discrete PDE $\longrightarrow$ discrete exact gradient

- Shape variation :
  1. Free form deformation. [Barr,1984]
  2. Torsional springs or Elliptic operators. [Farhat & Degand, 2002]
  3. Transpiration (Hadamard).

Review of Design methods (2)

- The discrete exact gradient developed by automatic differentiation reverse mode.
- Differentiator tool : Tapenade (L. Hascoet & V. Pascual).[1]
- Shape deformation by transpiration boundary conditions.
- Mesh deformation by torsional springs.

[1] TAPENADE 2.1 *user's guide,RT-0300,INRIA,2004*

Design using Euler equations (1)

The three-dimensional Euler equations may be written as

$$\Psi(W) \quad = \quad \frac{\partial W}{\partial t} + \frac{\partial F_i(W)}{\partial x_i} = 0$$

where

$$W = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix}, \quad F_i(W) = \begin{pmatrix} \rho u_i \\ \rho u_i u_1 + P\delta_{i1} \\ \rho u_i u_2 + P\delta_{i2} \\ \rho u_i u_3 + P\delta_{i3} \\ (\rho E + P)u_i \end{pmatrix}$$

and δ_{ij} is the Kronecker delta function. Also,

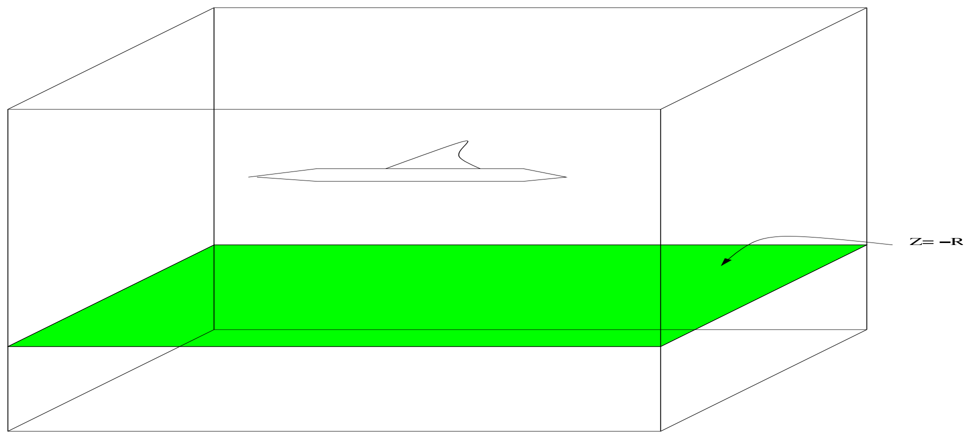
$$P = (\kappa - 1)[\rho E - \frac{1}{2}\rho(u_1^2 + u_2^2 + u_3^2)] \quad \text{with} \quad \kappa = 1.4$$

Design using Euler equations (2)

In order to design a shape which will lead to a desired pressure distribution, a natural choice is to set

$$j(\gamma) \quad = \quad J(\gamma, W(\gamma)) = 1/2 \int_C (P - P_{target})^2 dS$$

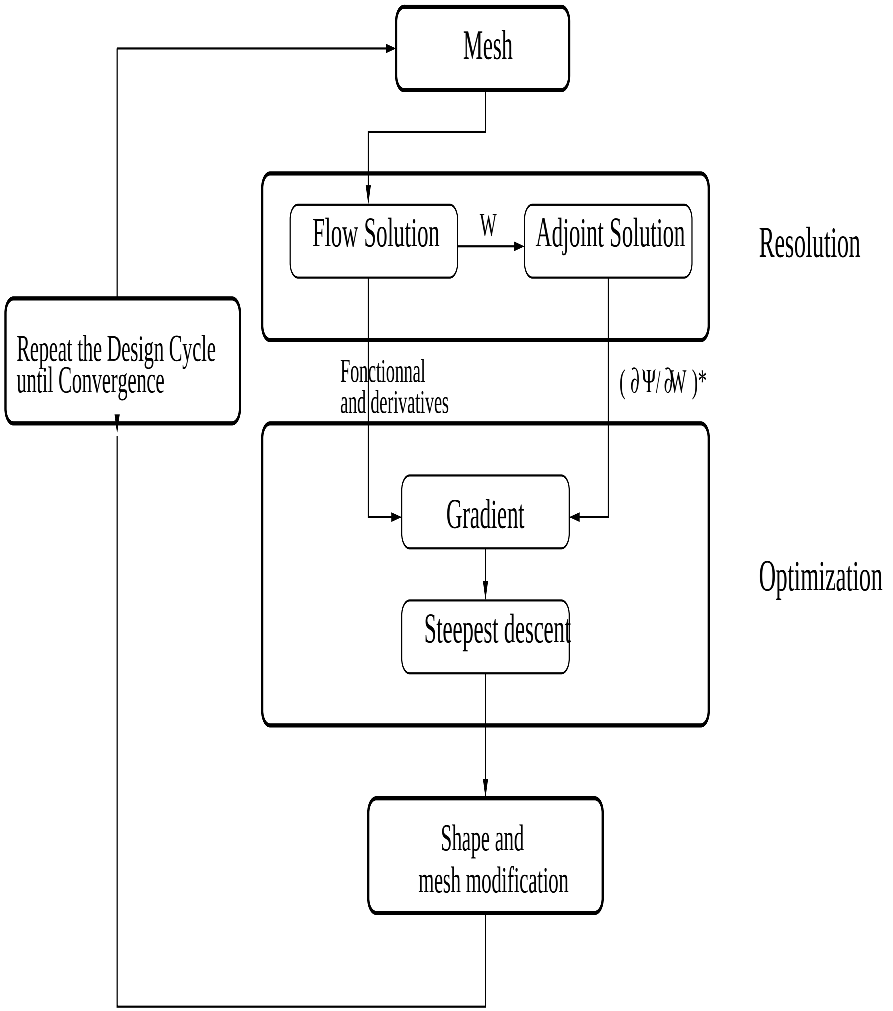
where P_{target} is the target surface pressure and the integral is evaluated over a surface area.



Design Cycle

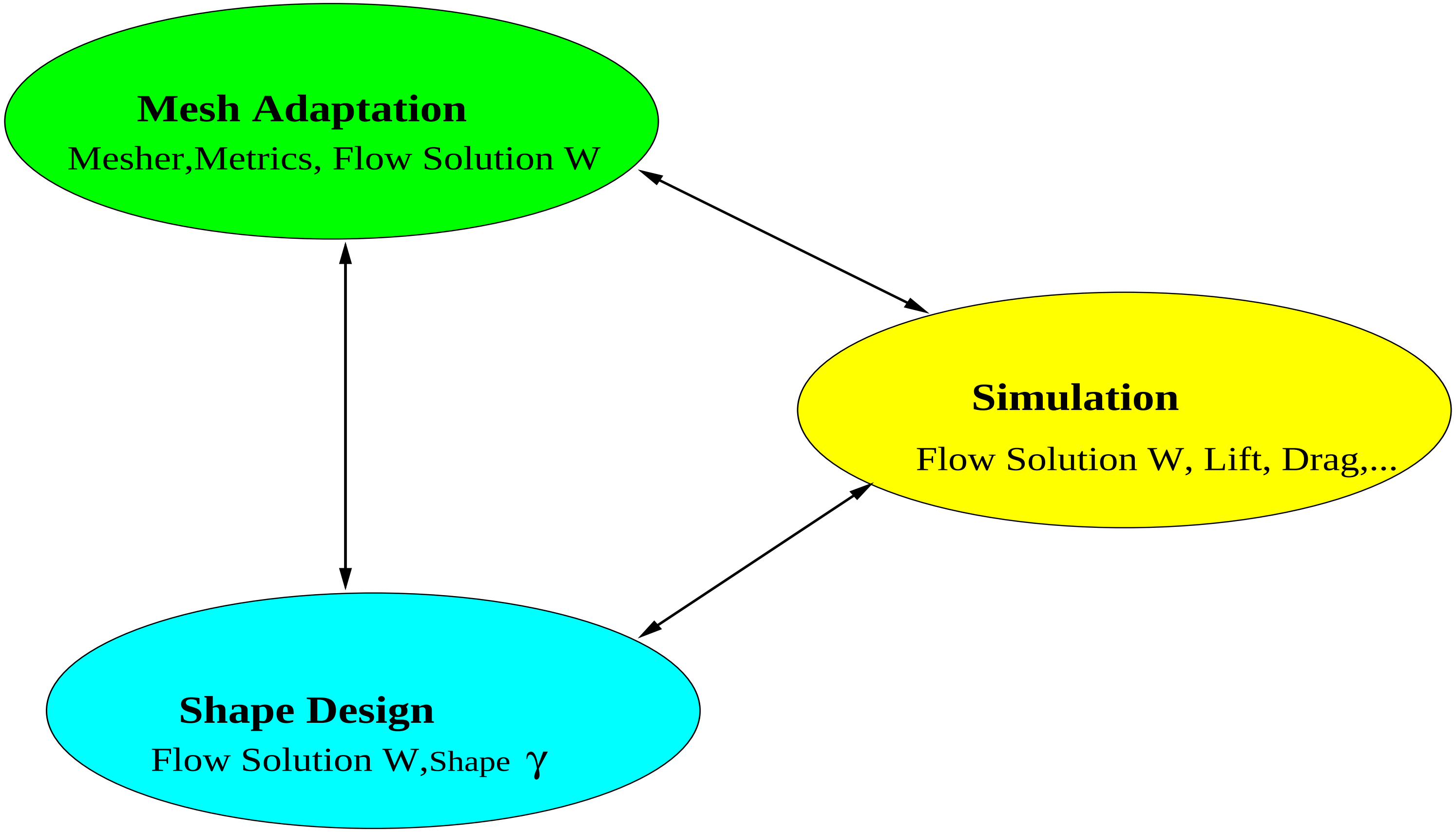
The design procedure can be summarized as follows :

1. Solve the flow equations
2. Solve the adjoint equations
3. Evaluate the gradient
4. Update the shape based on the direction of steepest descent
5. Return to 1. until convergence is reached



Mesh adaptation & Design

Mesh adaptation & Design



Review of adaptation strategies with respect to the criterion

- * Metric-based adaptation [Hecht 1997, ALauzet 2003]
  $\Rightarrow$ Reduce approximation error $(W - W_h)$ on W.

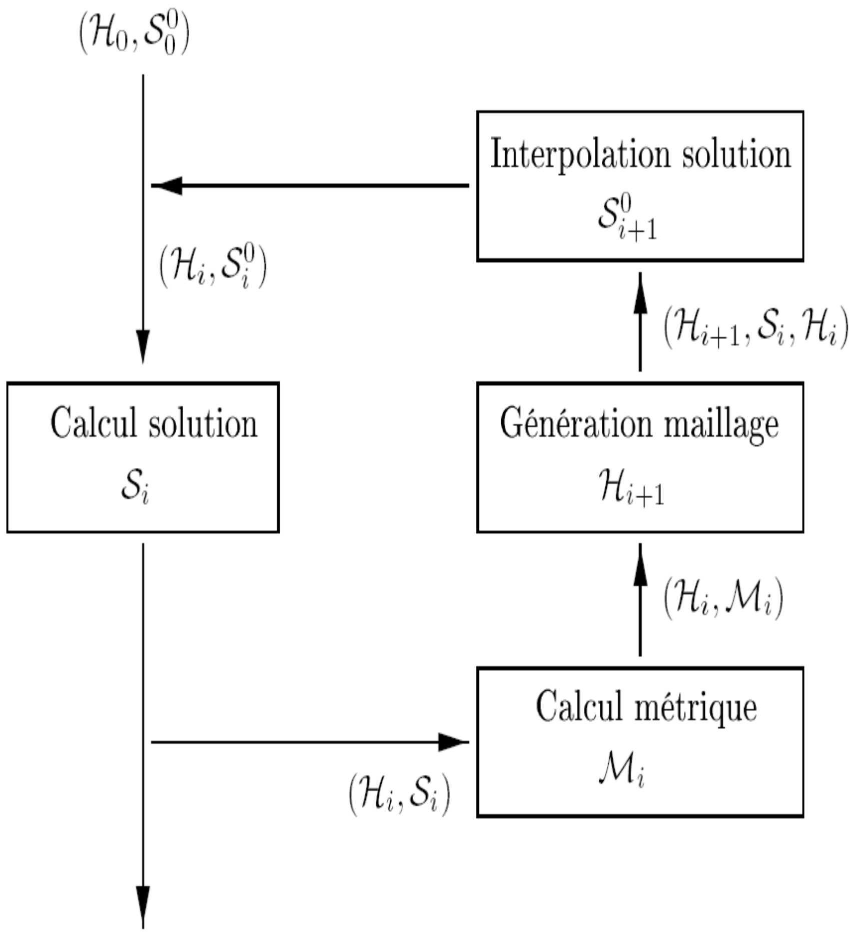
- * Goal-oriented adaptation
  - ☐ Super-convergence(adjoint based error estimate).[Giles & Pierce, 2001]
  - ☐ A posteriori adaptation. [Becker,Kapp & R. Rannacher,2001]

  $\Rightarrow$ Reduce error on $j(W)$.

- * Mesh adaptation for optimization.
  $\Rightarrow$ Adapt the mesh to an optimum parameter γ_{opt} [Becker, 2001]

Classical mesh adaptation algorithm.

- Find a local error model $e_{\mathcal{M}}$ based on interpolation error.
- Compute the L^p-optimum metric.
- Re-meshing takes into account the error analysis.



Review of continuous Metric [Dervieux & al. 2004,2006]

- Problematic
  - A theoretical analysis of anisotropic efficiency on unstructured meshes is really difficult to carry out.
  - No simple Hilbert structure for non-isotopological meshes required in any variational study is available.
  - Two different meshes may give the same interpolation error $\Rightarrow$ classes of equivalence ?

- Idea :
  - Interpreting metric as continuous functional of the domain $\Rightarrow$ continuous metric $\mathcal{M}$
  - A continuous metric is an analytical representative of the set of unit meshes with respect to $\mathcal{M} \Rightarrow \mathcal{M}$ define a classes of equivalence between meshes
  - Variational calculus to minimize a given error model

- Metric construction road-map :
  - Local error modeling stage :
    - find a local error model $e_{\mathcal{M}}(a)$ based on interpolation error.
  - Global calculus of variation stage :
    - minimize the error model in L^P-norm :

$$\text{find } \mathcal{M} \text{ such that } min_{\mathcal{M}} \int_{\Omega} |e_{\mathcal{M}}(x)|^p dx$$

under the constraint

$$\mathcal{C}(\mathcal{M}) = \int_{\Omega} \frac{1}{h_1 h_2 h_3} dx = \int_{\Omega} d(x) dx = N.$$

Local error modeling

- Continuous error :

$$e_{\mathcal{M}}(a) = max_{x \in \mathcal{B}(a)} |u(x) - \Pi_h u(x)|$$

- Discrete error :

$$e_{\mathcal{M}}(a) = ||u - \Pi_h u||_{K^*,\infty}$$

- Geometric error estimate :

$$e_{\mathcal{M}}(a) = max_{||\mu||_2 \leq 1} \sum_{j=1,3} (\sum_{i=1,3} \mu_i h_i (\vec{u_i}, \vec{u_j})^2 |\frac{\partial^2 u}{\partial \alpha_j^2}|)$$

- Final continuous error :

$$e_{\mathcal{M}}(a) = h_1^2|\frac{\partial^2 u}{\partial \alpha_1^2}| + h_2^2|\frac{\partial^2 u}{\partial \alpha_2^2}| + h_3^2|\frac{\partial^2 u}{\partial \alpha_3^2}|$$

Global calculus of variation

- Find the optimal functional $\mathcal{M}$ that solves the problem

$$min_{\mathcal{M}}\mathcal{E}(\mathcal{M}) = \quad min_{\mathcal{M}} \int_{\Omega} |e_{\mathcal{M}}(x)|^p dx = \\ min_{h_i} \int_{\Omega} (\sum_{i=1,3} h_i^2(x) |\frac{\partial^2 u}{\partial \alpha_i^2}|)^p dx$$

under the constraint

$$\mathcal{C}(\mathcal{M}) = \int_{\Omega} \frac{1}{h_1 h_2 h_3} dx = \int_{\Omega} d(x) dx = N.$$

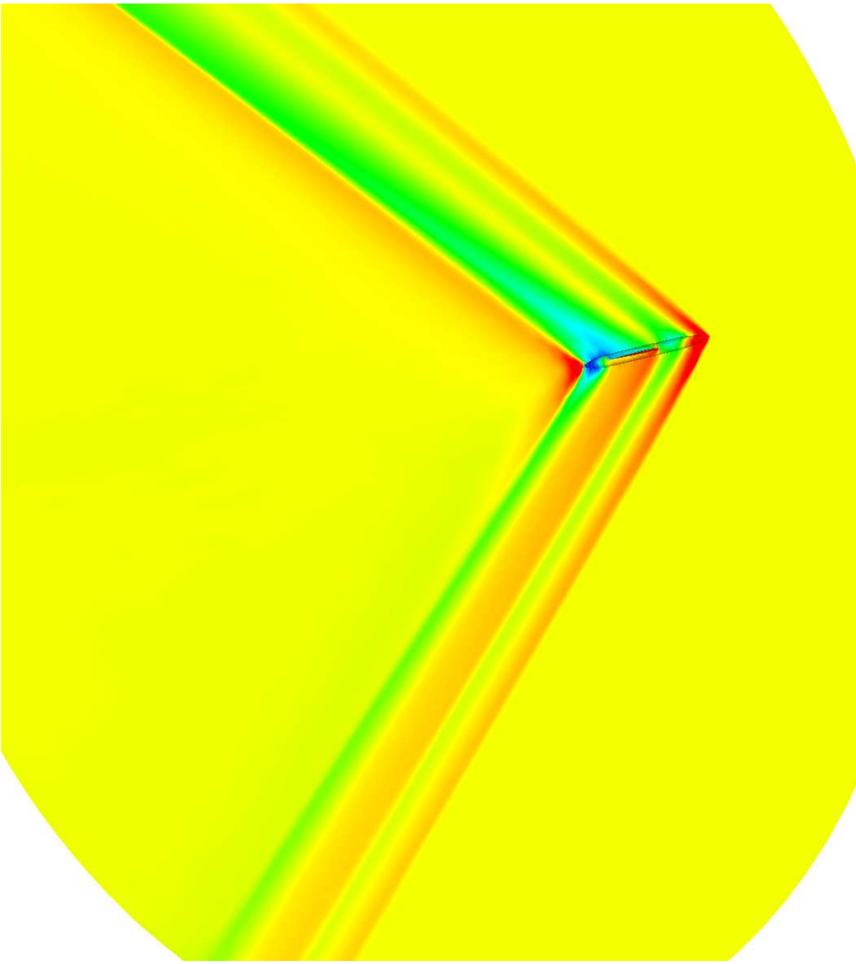
- To this end
  - perform a variable substitution with the anisotropic quotients denoted r_i and the the density d
  - get the anisotropic quotients
  - the optimal density

- Final solution
  The optimal metric is written as :

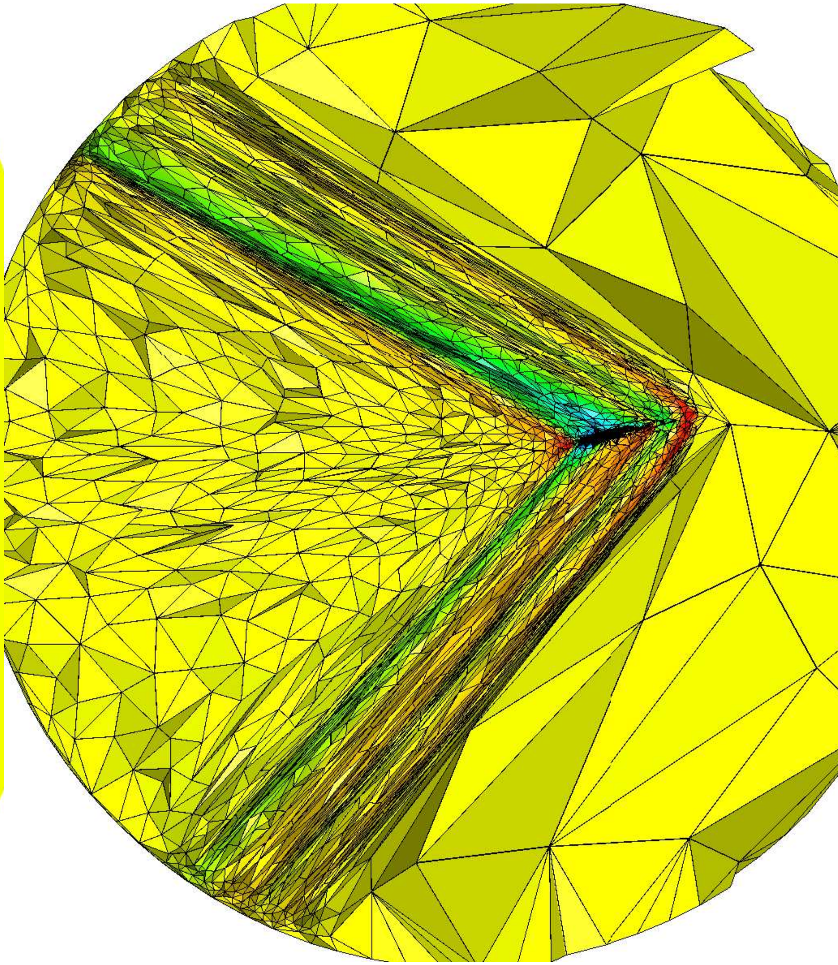
$$\mathcal{M}_{L_p} = D_{L_p} \quad \underset{1}{(det|H_u|)^{\frac{-1}{2p+3}}} \quad \underset{2}{R_u^{-1}} \quad \underset{3}{|\Lambda|} \quad \underset{4}{R_u}$$

  - 1. Global normalization $\Rightarrow$ used to reach the targeted number of points N with : $D_{L^p} = N^{\frac{2}{3}}(\int_\Omega (det|H_u|)^{\frac{p}{2p+3}})^{\frac{-2}{3}}$ and $D_{L^\infty} = N^{\frac{2}{3}}(\int_\Omega det(|H_u|)^{\frac{1}{2}})^{\frac{-2}{3}}$
  - 2. Local normalization $\Rightarrow$ refinement even with small solution variations, depends on L^p
  - 3. Optimal directions equal to Hessian eigenvectors
  - 4. Diagonal matrix of absolute values of Hessian eigenvalues

Sample : Adapted mesh corresponding to HISAC geometry



Pressure distribution



associated adapted mesh

Strong coupling : Design/Mesh adaptation

Adapted problem :

$$\mathcal{M}_{adap}(\gamma) = argmin\,\mathcal{E}(\mathcal{M},\gamma)$$

$$\bar{j}(\gamma) = j(\gamma, \mathcal{M}_{adap}(\gamma))$$

Then we shall minimize the following approximated functional :

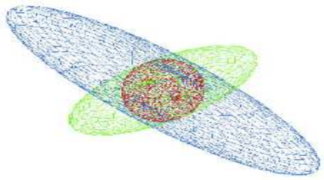
$$\text{Min}_{\gamma}\bar{j}(\gamma).$$

Problem : $\bar{j}$ is **not differentiable ! !**

Strong coupling (2)

Remedy :

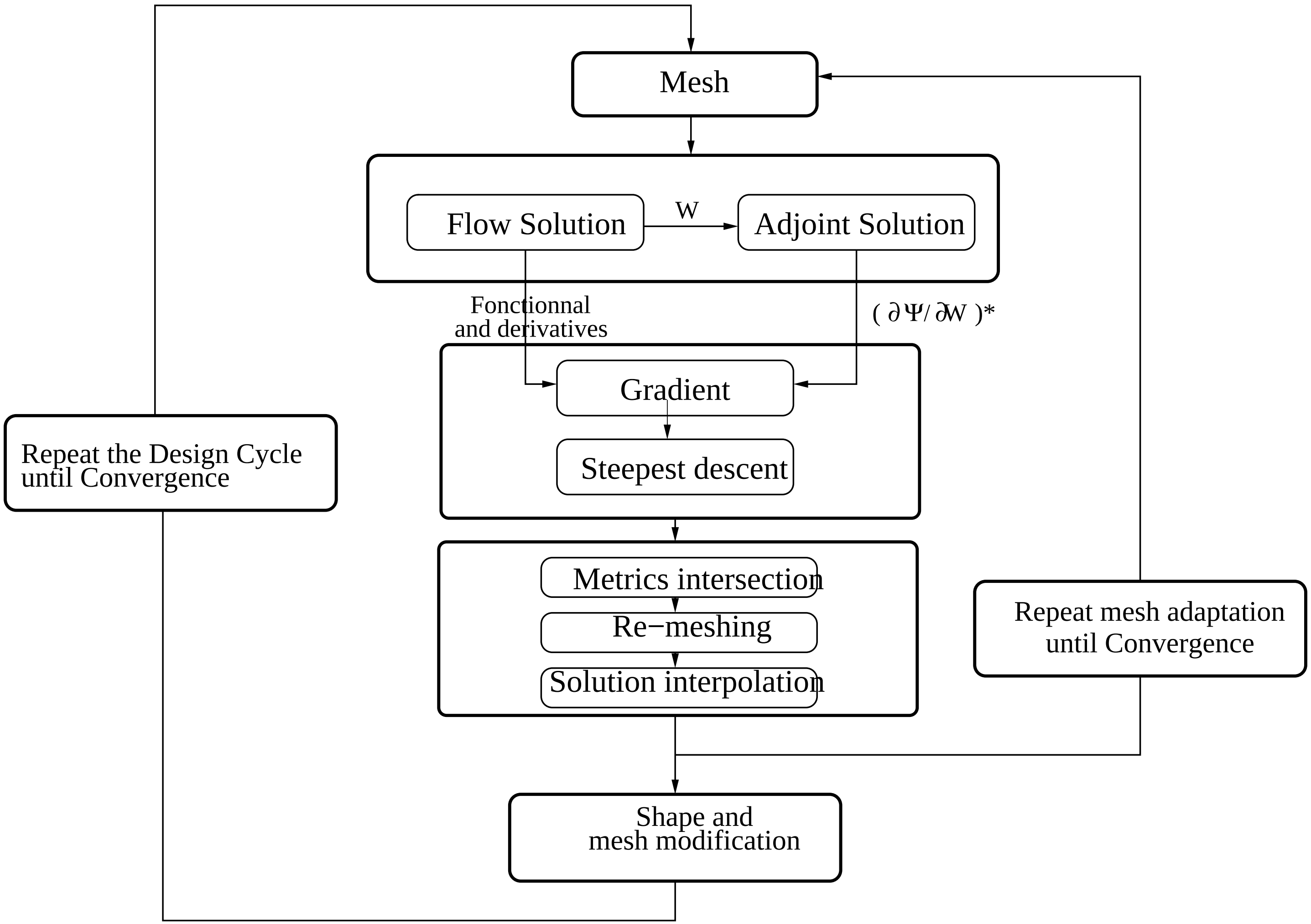
- ☐ Exact gradient for each step descent
  $\longrightarrow$ Do not adapt during an elementary descent step.
  $\longrightarrow$ But the mesh must be adapted to both solutions of descent



$\Rightarrow$ **Metrics intersection**

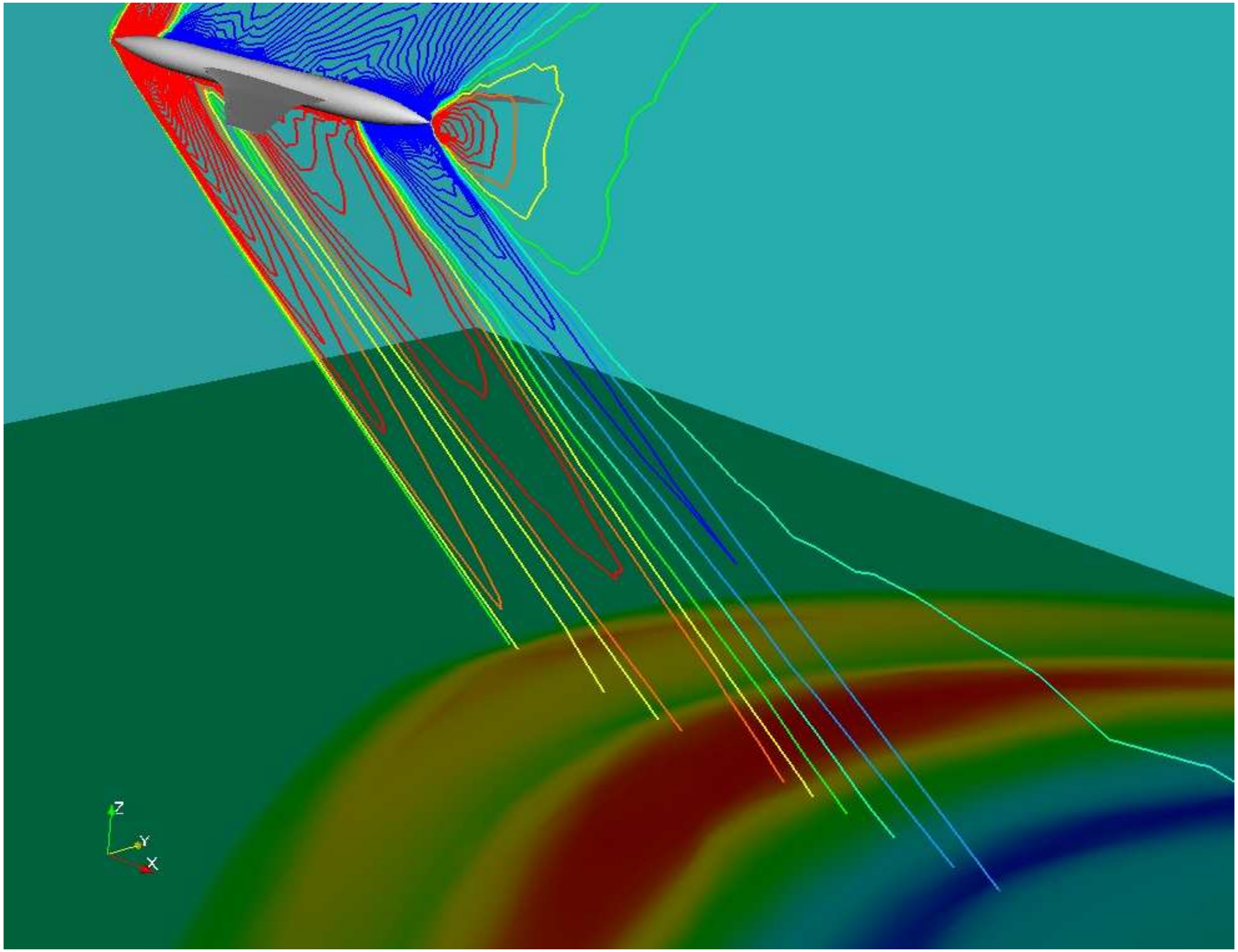
- ☐ Keep $\bar{j}$ well approximated
  $\longrightarrow$ To adapt after each shape update.

Strong coupling algorithm



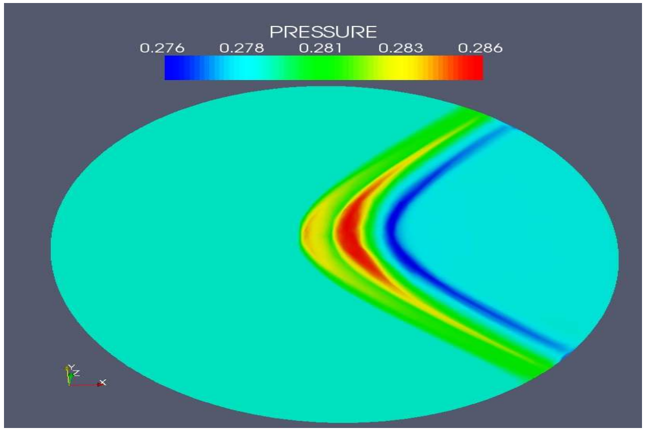
Results

Context

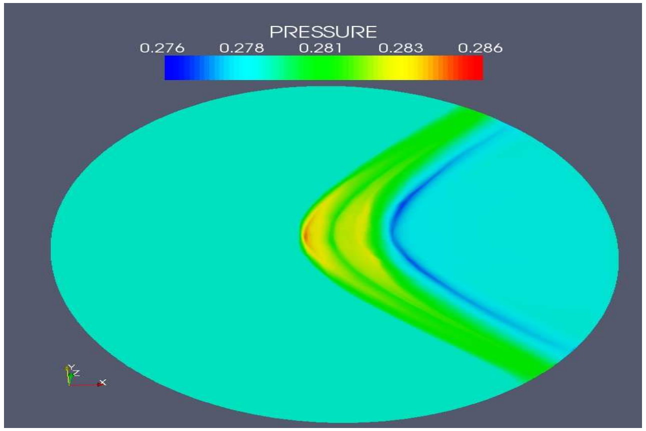


Hisac test-case : angle attack $\alpha = 3^\circ$ and a Mach number $M = 1.6$.

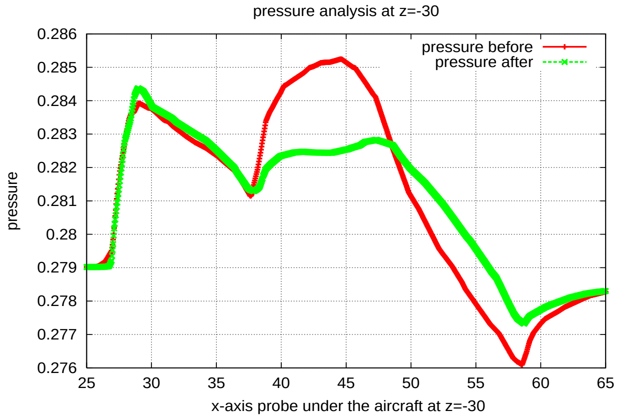
Test-case 1



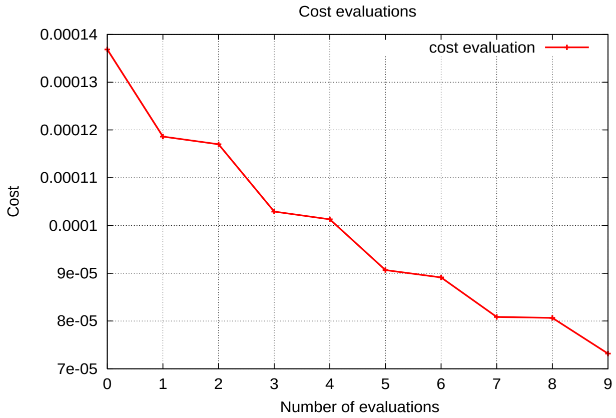
Pressure before at z=-30m



Pressure after at z=-30m



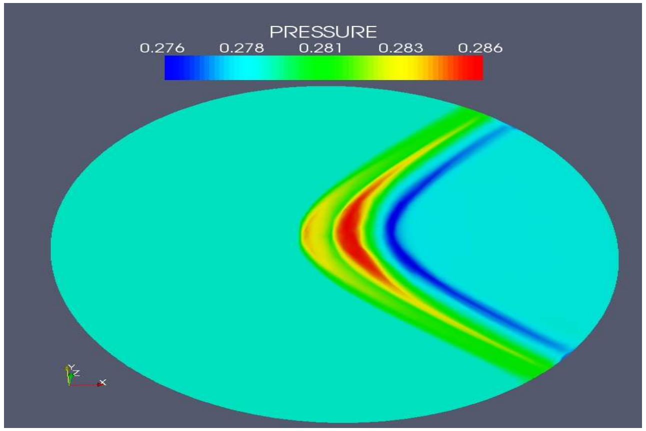
Pressure comparison



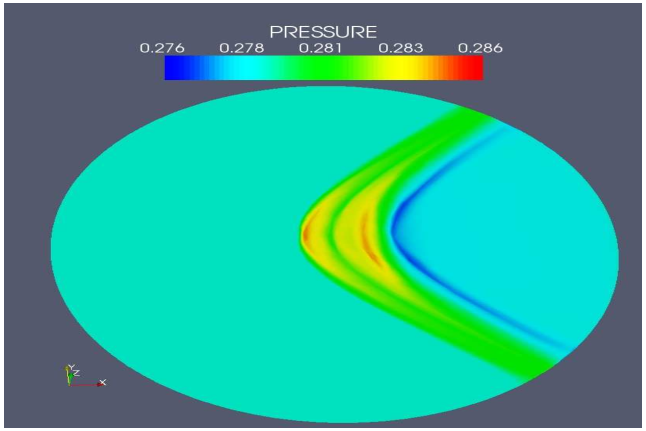
Cost evaluation

test-case 1 : $l_{front} = 25$ and $l_{back} = \mathbf{65}$.

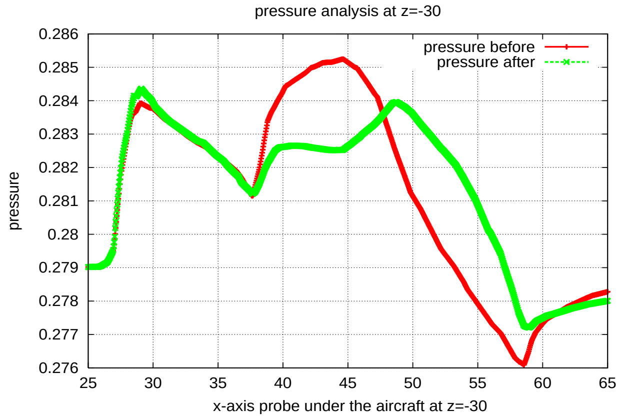
Test-case 2



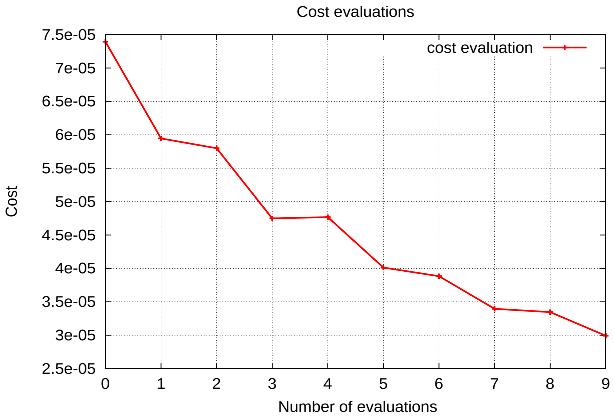
Pressure before at z=-30m



Pressure after at z=-30m



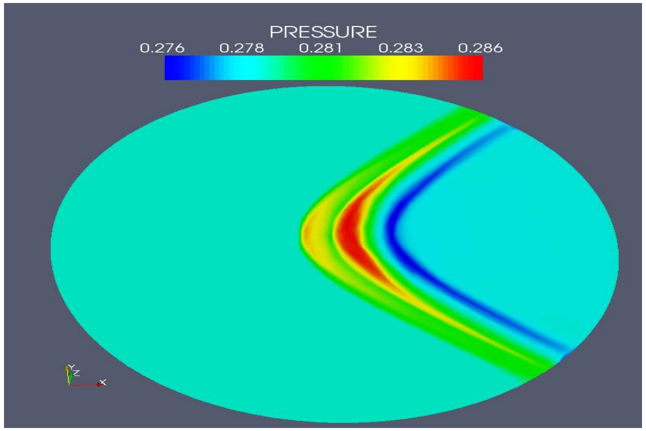
Pressure comparison



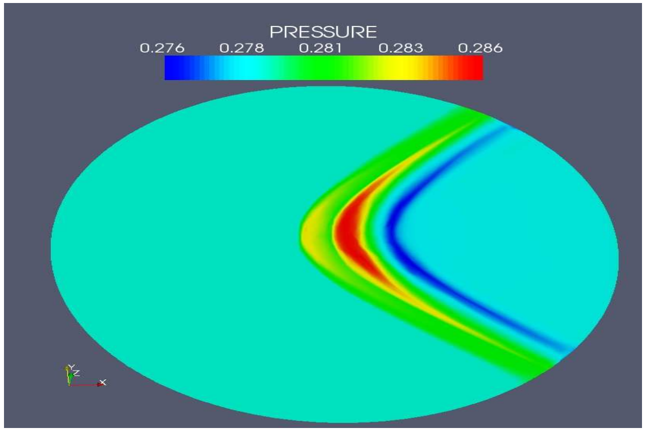
Cost evaluation

test-case 2 : $l_{front} = 25$ and $l_{back} = \mathbf{45}$.

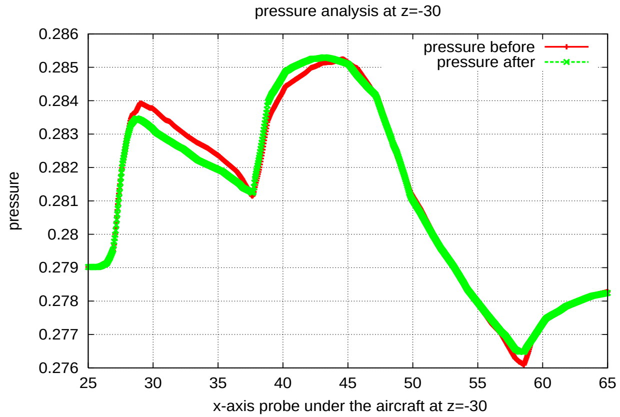
Test-case 3



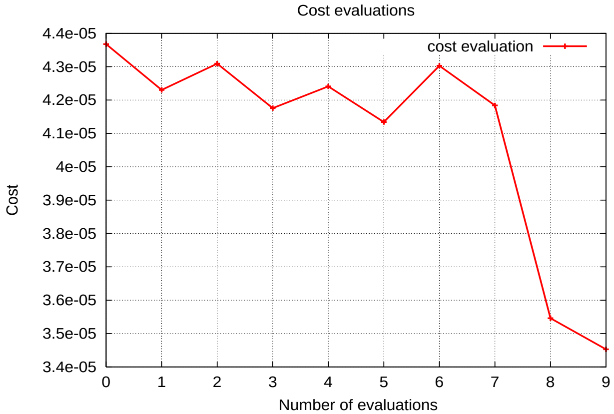
Pressure before at z=-30m



Pressure after at z=-30m



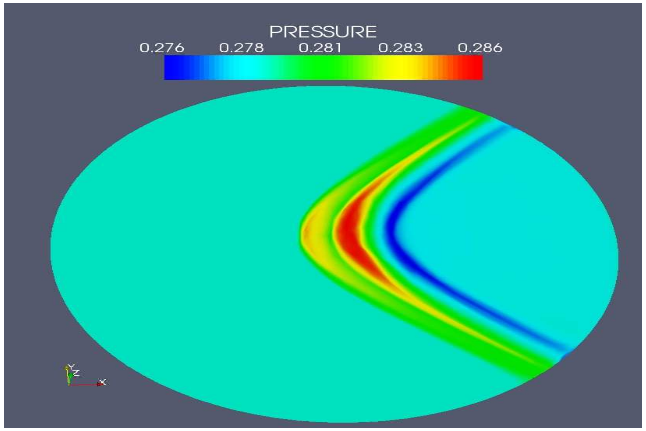
Pressure comparison



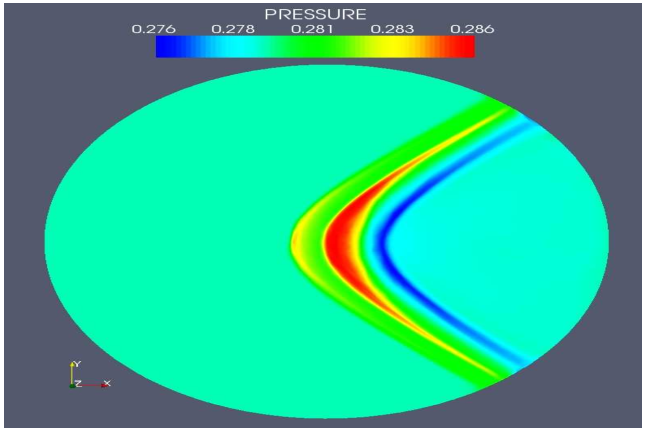
Cost evaluation

test-case 3 : $l_{front} = 25$ and $l_{back} = \mathbf{35}$.

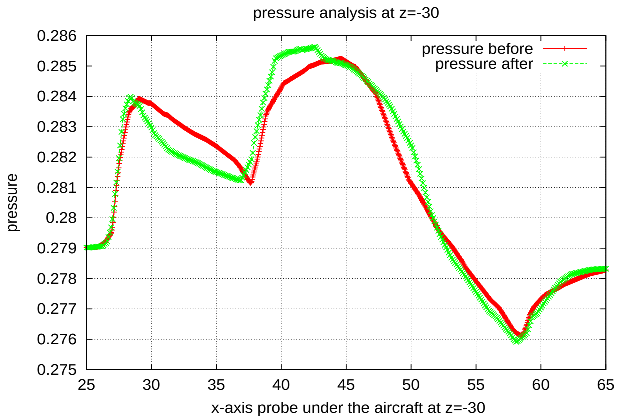
Test-case 3’



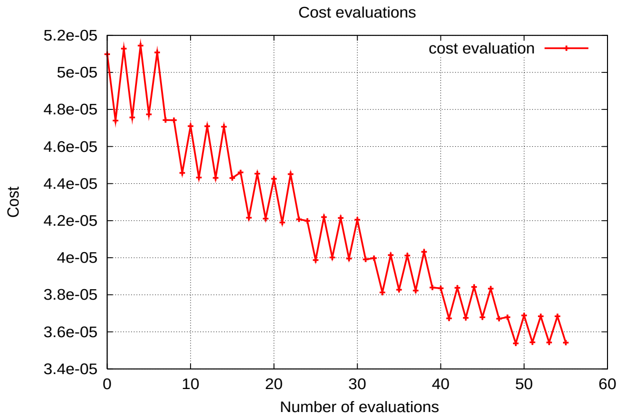
Pressure before at z=-30m



Pressure after at z=-30m



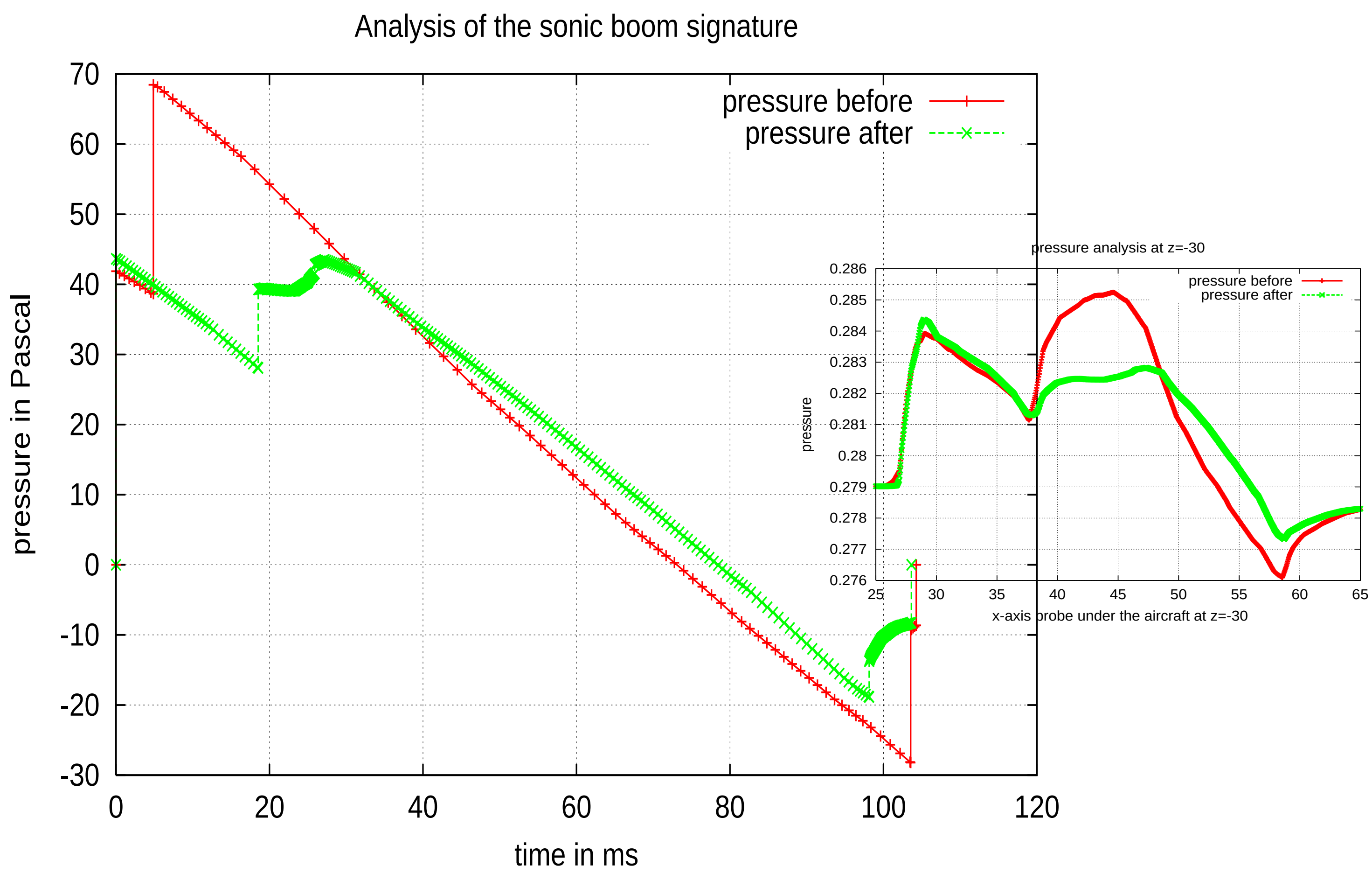
Pressure comparison



Cost evaluation

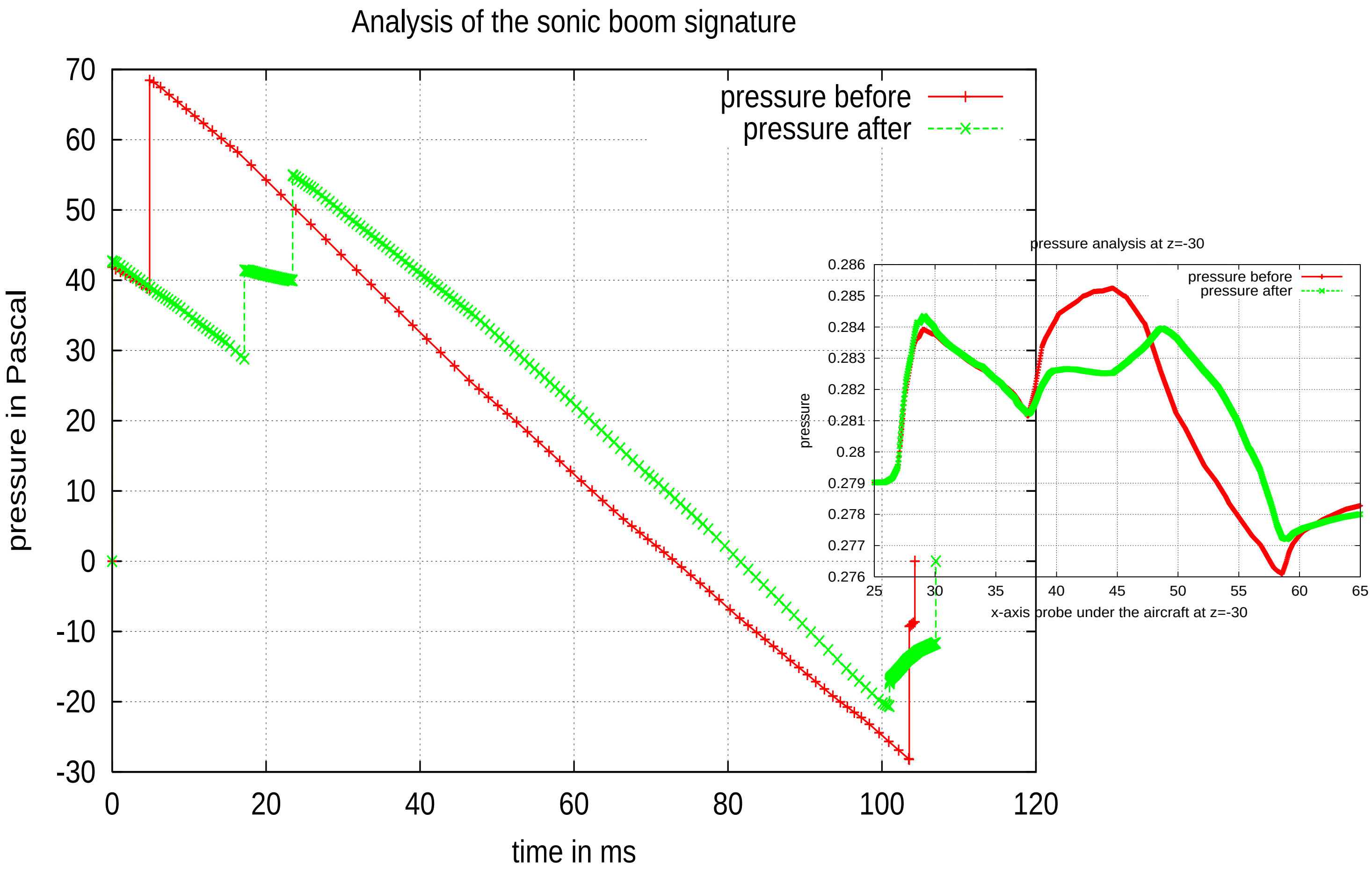
test-case 3’ : $l_{front} = 25$ and $l_{back} = \mathbf{35}$ with well converged intermediate meshes.

Analysis of the sonic boom pressure signature : test-case 1



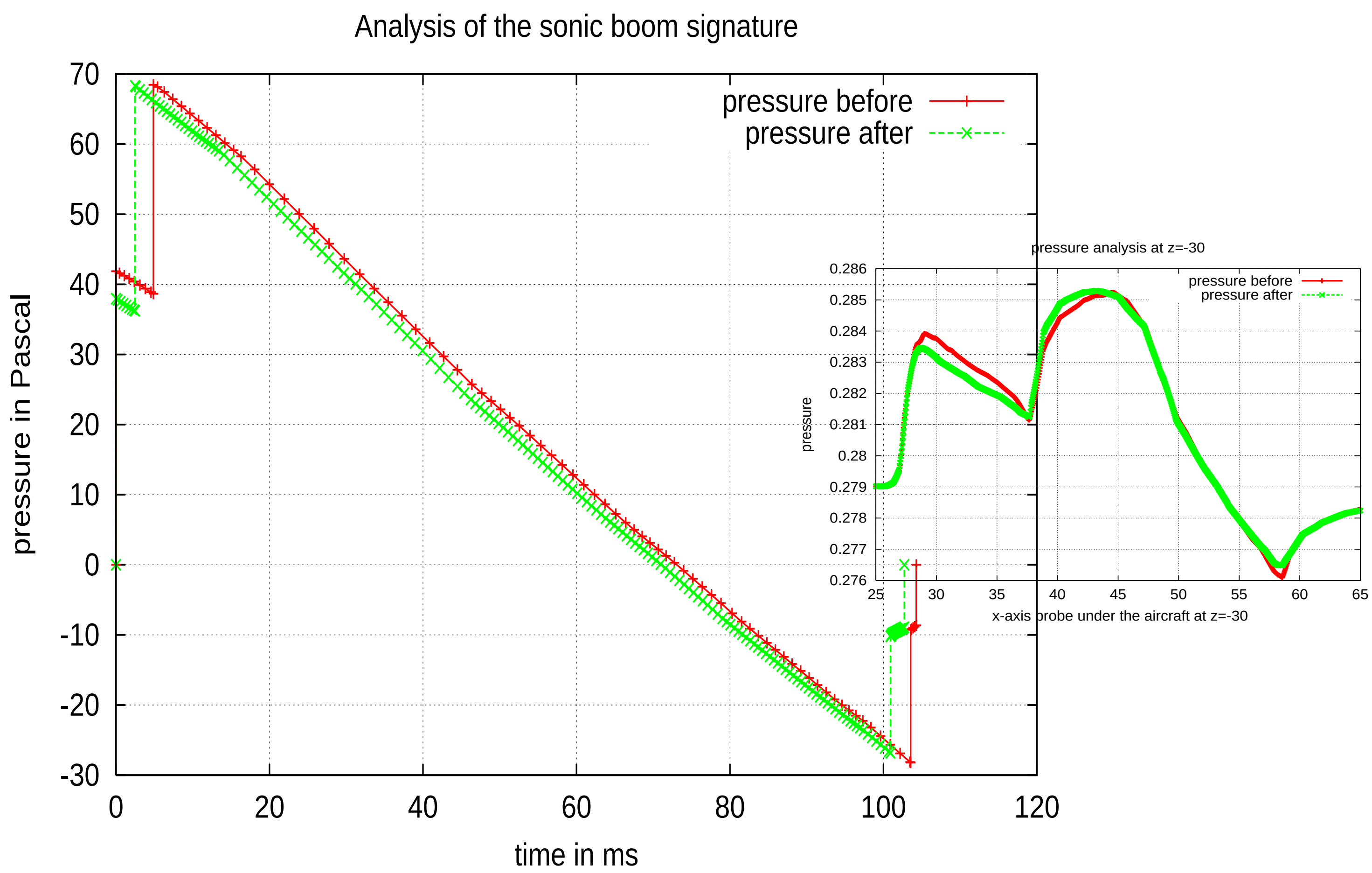
Test case 1 : $H = 45000\,ft$

Analysis of the sonic boom pressure signature : test-case 2

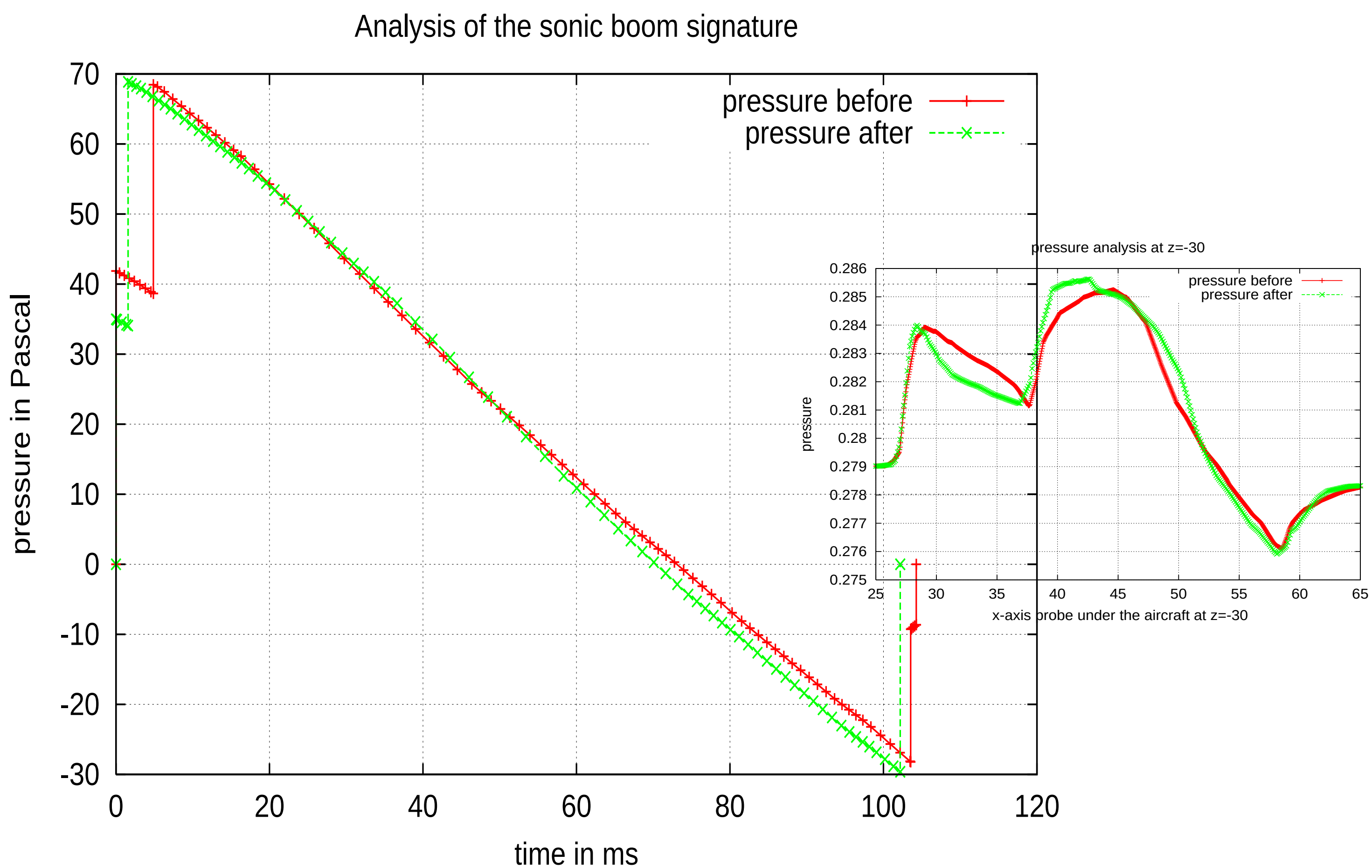


Test case 2 : $H = 45000 ft$

Analysis of the sonic boom pressure signature : test-case 3



Test case 3 : $H = 45000\,ft$



Test case 3' : $H = 45000ft$

Summary of the sonic boom analysis

case	ISPR (Pa)[2]	MSPR (Pa)[3]
Not optimized	42	68.29
Test case 1	42	43.47 (36%)
Test case 2	42	52.94 (28%)
Test case 3	37.5 (12%)	68.29
Test case 3'	35 (20%)	68.85

Initial and maximal shock pressure rise in Pascal for different cases

[2] ISPR : Initial shock pressure rise
[3] MSPR : Maximal shock pressure rise

Synthesis

- We have proposed a strongly coupled algorithm between mesh adaptation and optimization.
- Mesh adaptation improves the efficiency of the optimization loop.
- Preliminary application to sonic boom reduction have been presented.
- For difficult cases the advantage of the strongly coupled adaptation/optimization is evident :
  $\longrightarrow$ the convergence is fast and the result are reliable
- Future work
  - Further improvements are envisaged from introducing goal-oriented adaptation based on adjoint mesh