On a unique mesh modification operator for mesh adaptation

Adrien Loseille [1]

[1] EPI GAMMA3
INRIA Paris-Rocquencourt

informatics mathematics

Why Mesh Adaptation ?

Flow characteristics

- Phenomena are concentrated in small regions of the computational domain
  - $\rightsquigarrow$ uniform meshes are not optimal in term of sizes
- Phenomena are anisotropic: shock waves, boundary layers, ...
  - $\rightsquigarrow$ uniform meshes are not optimal in term of directions
- These regions are moving if the flow is unsteady
  - $\rightsquigarrow$ require an uniformly fine mesh in all evolution regions

Why Mesh Adaptation ?

In the real world we face:

- 3D problems
- Complex geometries
- Complex flows

$\Longrightarrow$ Problem solution is *a priori* unknown

$\Longrightarrow$ Simulation requires a large number of degrees of freedom

$\Downarrow$

Development of methods in order to reduce the complexity

one among them mesh adaptation

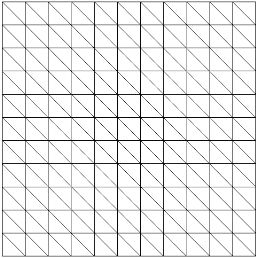
Idea: Modify discretization of Ω to control solution accuracy

Outline

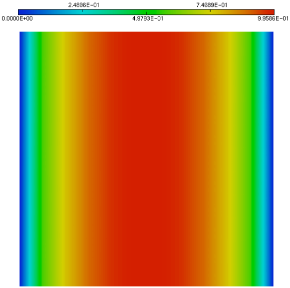
A simple 2D example

We have:

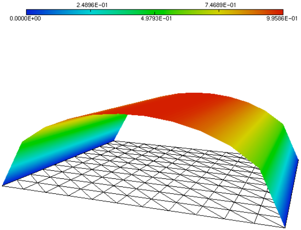
1. a mesh $\mathcal{H}_1$ of $\Omega = [-1, 1] \times [-1, 1]$ with $|\mathcal{H}_1| = N = 144$ vertices
2. the function (half-cylinder) $f(x, y) = \sqrt{1 - x^2}$



$\mathcal{H}_1$



Iso-values



$\Pi_h f$ of f on $\mathcal{H}_1$

A simple 2D example

We define the interpolation error by $e(f) = \|f - \Pi_h f\|_{\mathbf{L}^p(\Omega)}$

For instance, for $\mathcal{H}_1$:

$\mathbf{L}^1$	$\mathbf{L}^2$	$\mathbf{L}^\infty$
0.029	0.059	0.133

Problematic

How to reduce the interpolation error with the same number of vertices ?

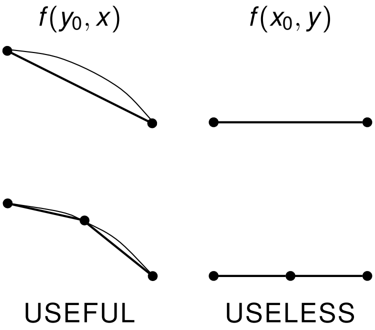
$$\begin{cases} \text{Find } \mathcal{H} = \text{Argmin}\|f - \Pi_h f\|_{\mathbf{L}^p} \\ |\mathcal{H}| = N = 144 \end{cases}$$

A simple 2D example

Two local remarks:

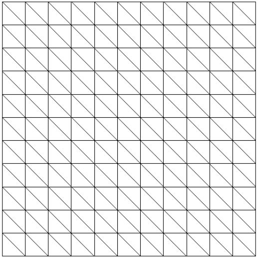
1. the largest variation (curvature) is in x-direction
2. the function f does not depend on y

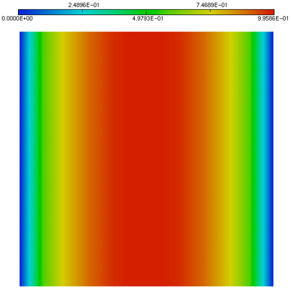
Function f has anisotropic properties

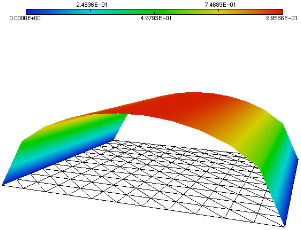


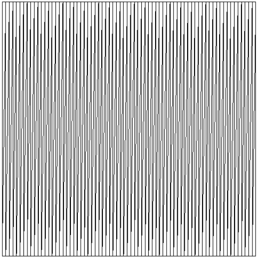
$\Longrightarrow$ All vertices on the lines $x = 1$ and $x = -1$

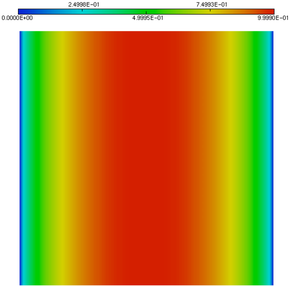
A simple 2D example

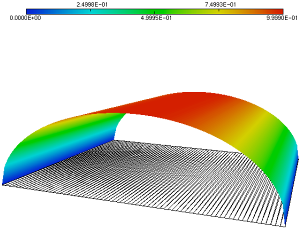












A simple 2D example

Interpolation error on both meshes:

$\mathbf{L}^1$	$\mathbf{L}^2$	$\mathbf{L}^\infty$	Mesh
0.029	0.059	0.133	$[-1:(2/11):1]\times[-1:(2/11):1]$
0.008	0.005	0.014	$[-1:(2/72):1]\times[-1:2:1]$

$\Longrightarrow$ Manual use of a local information

the anisotropy

of f to improve its representation

Mesh adaptation

$\Longrightarrow$ set up automatically this process

1. how to communicate with an automatic mesh generator ?
2. how to measure or quantify mesh size and anisotropy ?

The fundamental concept of **metric**

What is a Metric ?

- Canonical Euclidean space:

$$\langle \mathbf{u}\,,\,\mathbf{v}\rangle = {}^t\mathbf{u}\;\mathbf{v} \quad\Longrightarrow\quad \ell(\mathbf{a},\mathbf{b}) = \sqrt{{}^t\mathbf{ab}\;\mathbf{ab}}$$

- Euclidean metric space:
  $\mathcal{M}$: $d \times d$ symmetric definite positive matrix

$$\langle \mathbf{u}\,,\,\mathbf{v}\rangle_{\mathcal{M}} = {}^t\mathbf{u}\mathcal{M}\mathbf{v} \;\Longrightarrow\; \ell_{\mathcal{M}}(\mathbf{a},\mathbf{b}) = \sqrt{{}^t\mathbf{ab}\;\mathcal{M}\;\mathbf{ab}}$$

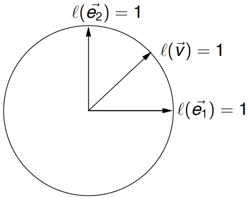
$$|K|_{\mathcal{M}} \quad = \quad \sqrt{\det \mathcal{M}}\,|K|$$

$$\cos(\theta) \quad = \quad \frac{\langle \mathbf{u}\,,\,\mathbf{v}\rangle_{\mathcal{M}}}{\|\mathbf{u}\|_{\mathcal{M}} + \|\mathbf{v}\|_{\mathcal{M}}}$$

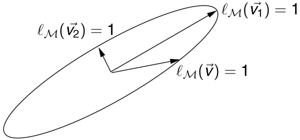
Geometric Representation:

- Unit ball:

$$\mathcal{E}_{\mathcal{M}(\mathbf{a})} = \left\{ \mathbf{b} \,\middle|\, \sqrt{{}^t\mathbf{ab}\,\mathcal{M}(\mathbf{a})\,\mathbf{ab}} = 1 \right\}$$



$\mathcal{M} = Id$



$\mathcal{M} > 0$

What is a Metric ?

- Euclidean metric space:
  $\mathcal{M}$: $d \times d$ symmetric definite positive matrix

$$\langle \mathbf{u}\,,\,\mathbf{v}\rangle_{\mathcal{M}} = {}^t\mathbf{u}\mathcal{M}\mathbf{v} \implies \ell_{\mathcal{M}}(\mathbf{a},\mathbf{b}) = \sqrt{{}^t\mathbf{ab}\ \mathcal{M}\ \mathbf{ab}}$$

- Riemannian metric space:
  $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

$$\ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 \sqrt{{}^t\mathbf{ab}\ \mathcal{M}(\mathbf{a}+t\mathbf{ab})\ \mathbf{ab}}\,\mathrm{d}t$$

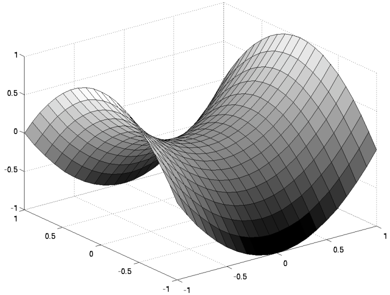
$$|K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}}\,\mathrm{d}K$$

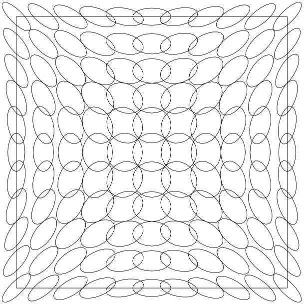
Computing geometric quantities on $\mathcal{S}$

$$\Updownarrow$$

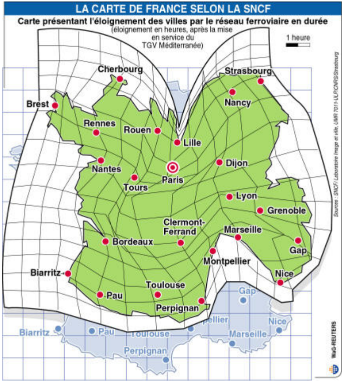
Computing geometric quantities in Riemannian metric space

$$\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$$





Geometric Illustrations



- **Main idea:** change the mesh generator distance computation
  [George, Hecht and Vallet., Adv. Eng. Software 1991]

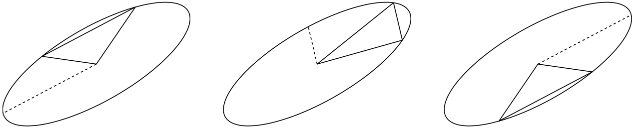
- **Fundamental concept:** **Unit mesh**

Adapting a mesh

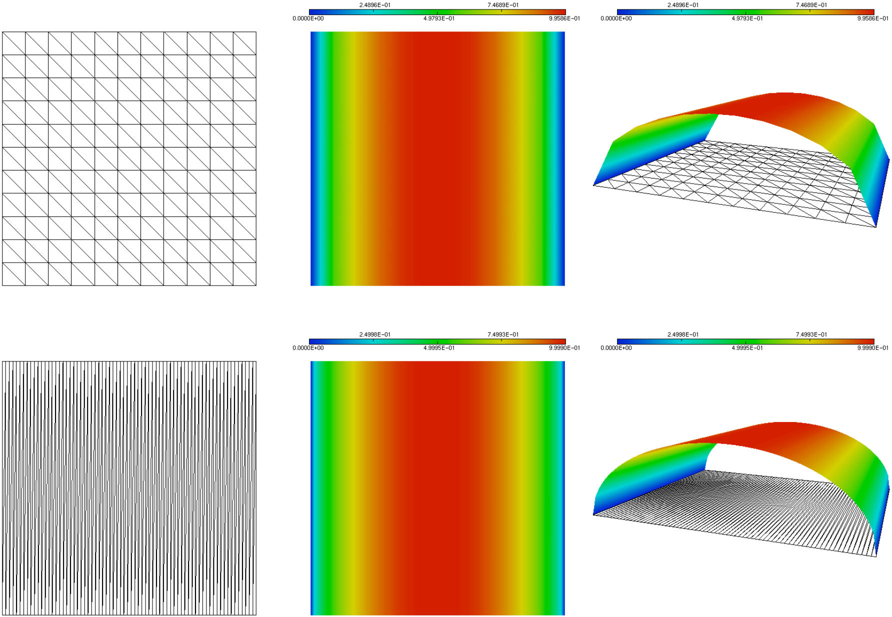
$\Updownarrow$ Work in adequate Riemannian metric space

Generating a uniform mesh w.r. to $\mathcal{M}(\mathbf{x})$

$$\mathcal{H} \text{ unit mesh} \iff \forall \mathbf{e},\ \ell_{\mathcal{M}}(\mathbf{e}) \approx 1 \text{ and } \forall K,\ |K|_{\mathcal{M}} \approx \begin{cases} \sqrt{3}/4 & \text{in 2D} \\ \sqrt{2}/12 & \text{in 3D} \end{cases}$$



Coming Back to the Introducing Example

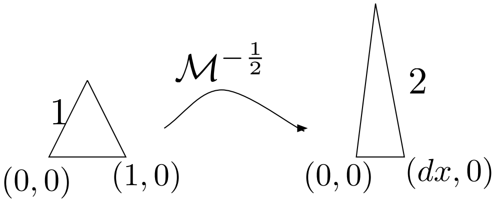


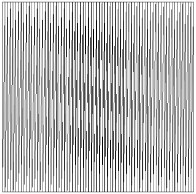
Coming Back to the Introducing Example

To generate $\mathcal{H}_2$, mesh generator works in $([-1,1]\times[-1,1],\mathcal{M})$

$$\mathcal{M} = \begin{pmatrix} \frac{1}{dx^2} & 0 \\ 0 & \frac{1}{2^2} \end{pmatrix}$$

$$\begin{cases} \|{}^t(x,0)\|^2_{\mathcal{M}} = x^2/dx^2 & = 1 \ \text{ if } x = dx \\ \|{}^t(0,y)\|_{\mathcal{M}} = y^2/2^2 & = 1 \ \text{ if } y = 2 \end{cases}$$





Triangles area $|K|_{\mathcal{M}} = \frac{\sqrt{3}}{4}$

Mesh adaptation

$\Longrightarrow$ set up automatically this process

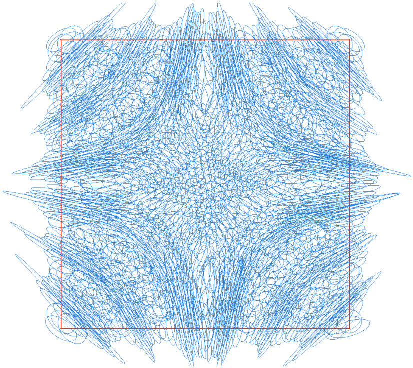
✓ how to communicate with an automatic mesh generator ?

2. how to measure or quantify mesh size and anisotropy ?

Use appropriate error estimates

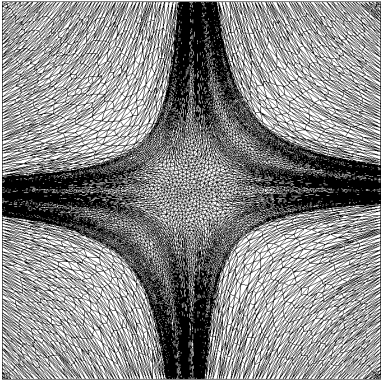
Generation of Adapted Meshes

Inputs $(\mathcal{H}_0, \mathcal{M}_i)_{i \in \mathcal{H}}$



Part ❷

Output $\mathcal{H}$



Parts: ❸ , ❹ , ❺

$\mathcal{H}$ unit mesh $\iff \forall \mathbf{e},\ \ell_{\mathcal{M}}(\mathbf{e}) \approx 1$ and $\forall K,\ |K|_{\mathcal{M}} \approx \begin{cases} \sqrt{3}/4 & \text{in 2D} \\ \sqrt{2}/12 & \text{in 3D} \end{cases}$

Outline

An ill-posed problem

Find $\mathcal{H}_{opt}$ having N vertices such that

$$\mathcal{H}_{opt}(u) = \operatorname{Arg\,min}_{\mathcal{H}} \|u - \Pi_h u\|_{\mathcal{H},\mathbf{L}^p(\Omega)}$$

Continuous Mesh Framework

We proposed a continuous mesh framework to solve this problem [Loseille and Alauzet, SINUM 2010]

Discrete	Continuous
Element K	Metric tensor $\mathcal{M}$
Mesh $\mathcal{H}$ of Ω_h	Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$
Number of vertices N_v	Complexity $\mathcal{C}(\mathbf{M}) = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\, d\mathbf{x}$
Linear interpolate $\Pi_h u$	Continuous linear interpolate $\pi_{\mathcal{M}} u$

Working in this framework enables us to use powerful mathematical tool

Definition

- function $\mathbf{M} : \mathbf{a} \in \Omega \mapsto \mathcal{M}(\mathbf{a})$,
- density: $d = \dfrac{1}{h_1 h_2 h_3} = \sqrt{\lambda_1 \lambda_2 \lambda_3}$,
- n anisotropic quotients $r_i = \dfrac{h_i^3}{h_1 h_2 h_3}$
- complexity $\mathcal{C}$:

$$\mathcal{C}(\mathbf{M}) = \int_\Omega d(\mathbf{a})\, d\mathbf{a} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{a}))}\, d\mathbf{a}.$$

Matrix rewriting

$$\mathcal{M}(\mathbf{a}) = d^{\frac{2}{3}}(\mathbf{a})\, \mathcal{R}(\mathbf{a}) \begin{pmatrix} r_1^{-2/3}(\mathbf{a}) & & \\ & r_2^{-2/3}(\mathbf{a}) & \\ & & r_3^{-2/3}(\mathbf{a}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{a}).$$

Local interpolation error [Loseille and Alauzet, SINUM 2010]

For all K unit for $\mathcal{M}$ and for all u quadratic positive form $(u(\mathbf{x}) = \frac{1}{2}\,{}^t\mathbf{x}\, H_u\, \mathbf{x})$:

$$\begin{aligned} \|u - \Pi_h u\|_{\mathbf{L}^1(K)} &= \frac{|K|}{40} \sum_{i=1}^{6} {}^t\mathbf{e}_i |H_u| \mathbf{e}_i \\ &= \frac{\sqrt{2}}{240} \underbrace{\det(\mathcal{M}^{-\frac{1}{2}})}_{\textit{mapping}} \underbrace{\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}\, H_u\, \mathcal{M}^{-\frac{1}{2}})}_{\textit{anisotropic term}} \end{aligned}$$

Discrete-continuous duality

$$\begin{aligned} \forall \mathbf{a} \in \Omega\,, \quad |u - \pi_{\mathcal{M}} u|(\mathbf{a}) &= 2\, \frac{\|u - \Pi_h u\|_{\mathbf{L}^1(K)}}{|K|} \\ &= \frac{1}{10} \mathrm{trace}\big(\mathcal{M}(\mathbf{a})^{-\frac{1}{2}}\, |H_u(\mathbf{a})|\, \mathcal{M}(\mathbf{a})^{-\frac{1}{2}}\big) \end{aligned}$$

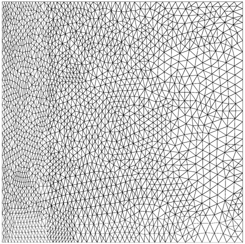
Examples

Sequence of 2D embedded continuous meshes $\mathbf{M}(\alpha) = (\mathcal{M}_\alpha(\mathbf{x}))_{\mathbf{x}\in\Omega}$

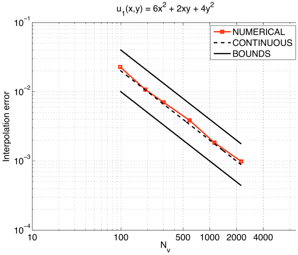
$$\mathcal{M}_\alpha(x,y) = \alpha \begin{pmatrix} h_1^{-2}(x,y) & 0 \\ 0 & h_2^{-2}(x,y) \end{pmatrix} \text{ with } \begin{array}{l} h_1(x,y) = 0.1(x+1) + 0.05(x-1) \\ h_2(x,y) = 0.2 \end{array}$$

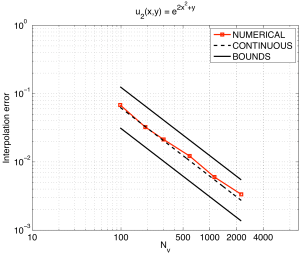
Analyze the interpolation error of functions:

$$u_1(x,y) = 6x^2 + 2xy + 4y^2 \text{ and } u_2(x,y) = e^{(2x^2+y)}$$



$\alpha = 32$





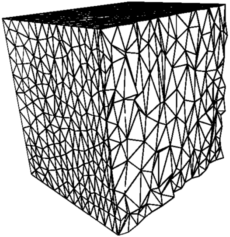
Sequence of 3D embedded continuous meshes

$\mathbf{M}(\alpha) = (\mathcal{M}_\alpha(\mathbf{x}))_{\mathbf{x}\in\Omega}$ defined on $\Omega = [0,1]\times[0,1]\times[0,1]$ by:

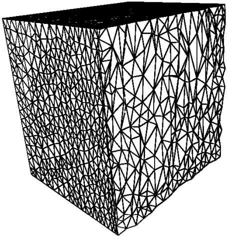
$$\mathcal{M}_\alpha(x,y,z) = \alpha \begin{pmatrix} h_1^{-2}(x,y,z) & 0 & 0 \\ 0 & h_2^{-2}(x,y,z) & 0 \\ 0 & 0 & h_3^{-2}(x,y,z) \end{pmatrix},$$

where $h_1(x,y,z) = 0.1(x+1) + 0.05(x-1)$, $h_2(x,y,z) = 0.2$, $h_3(x,y,z) = 0.2(z+2)$

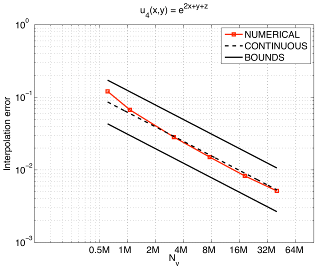
Interpolation error on : $u_3(x,y,z) = e^{2x+y+z}$



$\alpha = 4$



$\alpha = 8$



An ill-posed problem

Find $\mathcal{H}_{opt}$ having N vertices such that

$$\mathcal{H}_{opt}(u) = \text{Arg}\min_{\mathcal{H}} \|u - \Pi_h u\|_{\mathcal{H},\mathbf{L}^p(\Omega)}$$

A well-posed problem

Find $\mathbf{M}_{\mathbf{L}^p} = (\mathcal{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ of complexity N such that

$$\begin{aligned} E_{\mathbf{L}^p}(\mathbf{M}_{\mathbf{L}^p}) = \min_{\mathbf{M}} E_{\mathbf{L}^p}(\mathbf{M}) &= \min_{\mathbf{M}} \|u - \pi_{\mathcal{M}} u\|_{\mathbf{L}^p(\Omega)} \\ &= \min_{\mathbf{M}} \left(\int_\Omega |u(\mathbf{x}) - \pi_{\mathcal{M}} u(\mathbf{x})|^p \, \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}} \end{aligned}$$

Solved by a calculus of variations

Minimizing the Interpolation Error in $\mathbf{L}^p$-norm

Optimal metric

$$\mathcal{M}_{\mathbf{L}^p} = \underset{❶}{D_{\mathbf{L}^p}} \quad \underset{❷}{(\det|H_u|)^{\frac{-1}{2p+3}}} \quad \underset{❸}{\mathcal{R}_u^{-1}} \quad \underset{❹}{|\Lambda|} \quad \mathcal{R}_u$$

❶ Global normalization: to reach the constraint complexity N

$$D_{\mathbf{L}^p} = N^{\frac{2}{3}} \left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+3}} \right)^{-\frac{2}{3}} \quad \text{and} \quad D_{\mathbf{L}^\infty} = N^{\frac{2}{3}} \left(\int_\Omega (\det|H_u|)^{\frac{1}{2}} \right)^{-\frac{2}{3}}$$

❷ Local normalization: sensitivity to small solution variations, depends on $\mathbf{L}^p$ norm

❸ Optimal directions equal to Hessian eigenvectors

❹ Diagonal matrix of absolute values of Hessian eigenvalues

It verifies the following properties: [Loseille and Alauzet, SINUM 2010]

- $\mathbf{M}_{\mathbf{L}^p}(u)$ is unique
- $\mathbf{M}_{\mathbf{L}^p}(u)$ is locally aligned with the eigenvectors basis of H_u and has the same anisotropic quotients as H_u
- $\mathbf{M}_{\mathbf{L}^p}(u)$ provides an optimal explicit bound of the interpolation error in $\mathbf{L}^p$ norm:

$$\|u - \pi_{\mathcal{M}_{\mathbf{L}^p}} u\|_{\mathbf{L}^p(\Omega)} = 3\, N^{-\frac{2}{3}} \left(\int_{\Omega} (\det |H_u|)^{\frac{p}{2p+3}} \right)^{\frac{2p+3}{3p}}$$

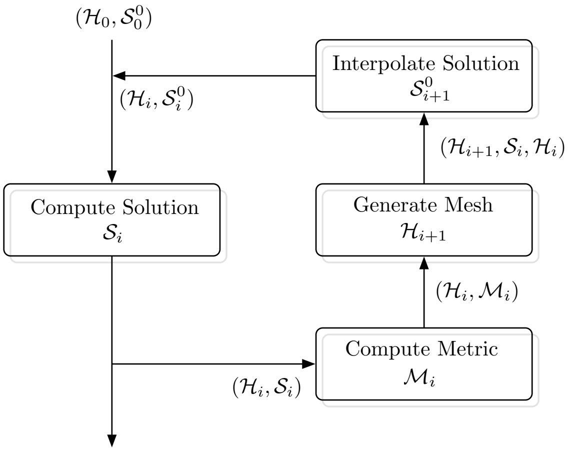
- For a sequence of embedded continuous meshes $(\mathbf{M}^N_{\mathbf{L}^p}(u))_{N=1\ldots\infty}$, the asymptotic order of convergence verifies:

$$\|u - \pi_{\mathcal{M}^N_{\mathbf{L}^p}} u\|_{\mathbf{L}^p(\Omega)} \leq \frac{Cst}{N^{2/3}}.$$

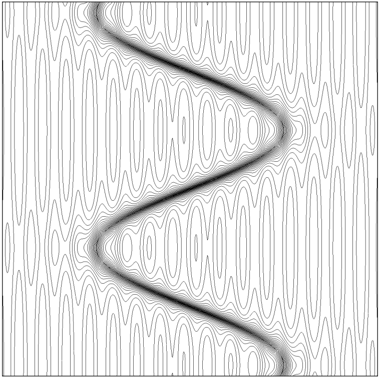
Thus, we may expect a global second order of mesh convergence for the mesh adaptation process

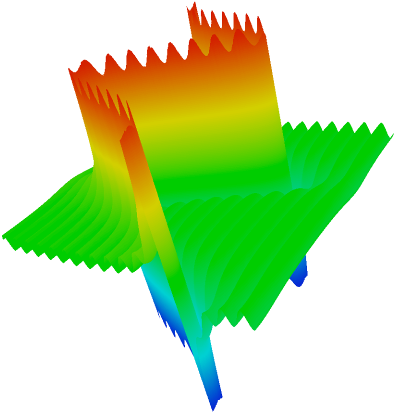
Mesh adaptation is a non-linear problem

$\Longrightarrow$ an iterative process is required to converge the couple mesh-solution

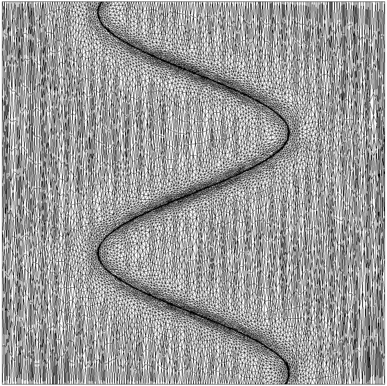


Example on a non-regular solution:

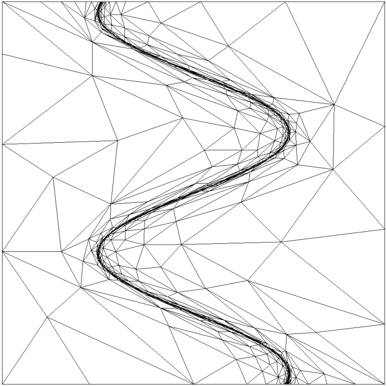




Example on a non-regular solution:



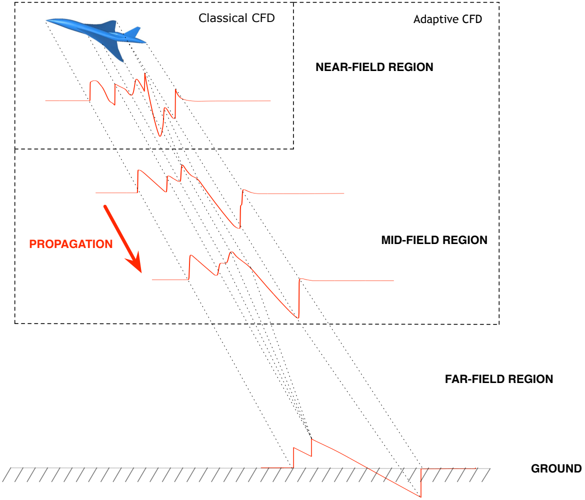
L^2-adaptation



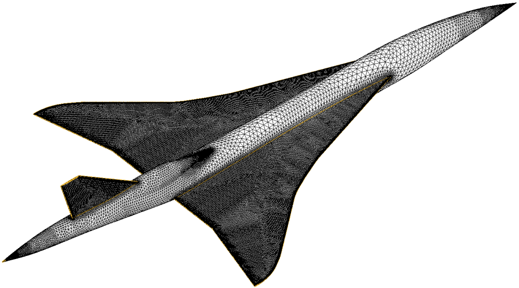
L^∞-adaptation

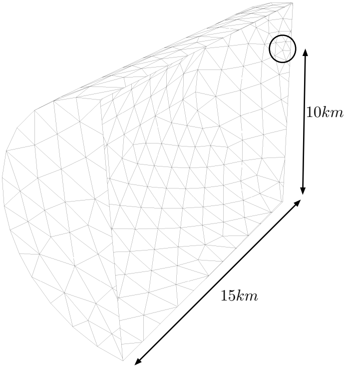
A more challenging computation

A full scale supersonic simulation



A full scale supersonic simulation

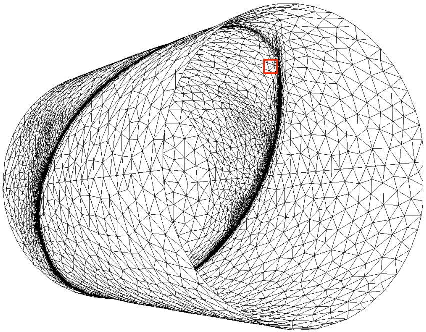
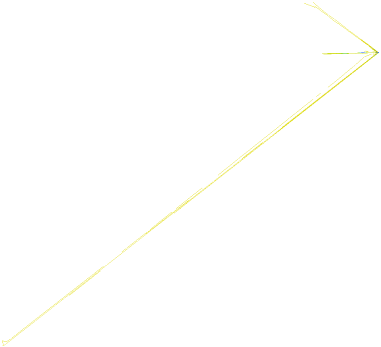




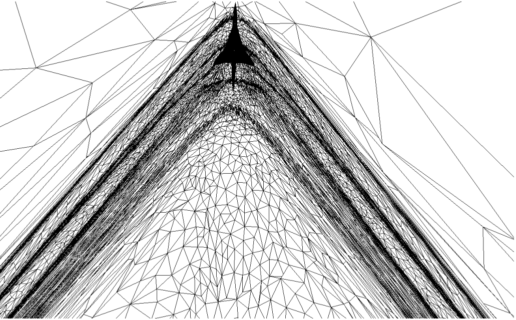
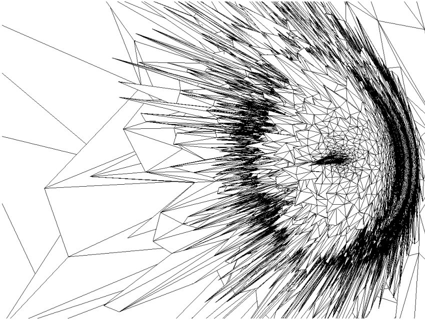
- initial mesh: frontal mesh generation, # vert. 415 535, # tets 2 397 666
- volume $[5.4e^{-11}, 4.7e^{10}]$
- $h_{min}/h_{max} = 1.0^{-9}$

Iteration	Complexity	Ratio	Quotient	# Vertices	# Tet.	CPU time
5	80 000	200	10 964	432 454	2 254 826	1 h 10 mn
10	160 000	383	30 295	608 369	3 294 197	2 h 54 mn
15	240 000	698	81 129	1 104 910	6 243 462	6 h 9 mn
20	400 000	1 089	177 295	1 757 865	10 125 724	11 h 15 mn
25	600 000	1 575	340 938	2 572 814	14 967 820	18 h 47 mn
30	800 000	1 907	503 334	3 299 367	19 264 402	28 h 35 mn

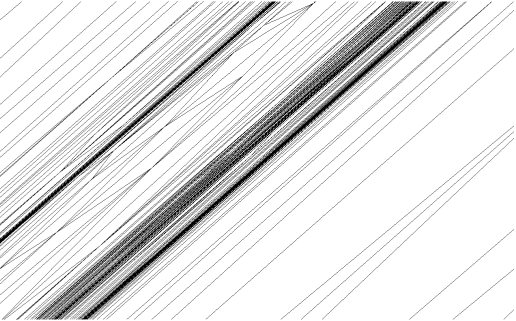
- 8 Cpu Mac Intel Xeon with 20 GB of memory
- total CPU time is around 28 h 35 mn
- 75 % FEFLO, 35 % in the remeshing, interpolation and error estimate.

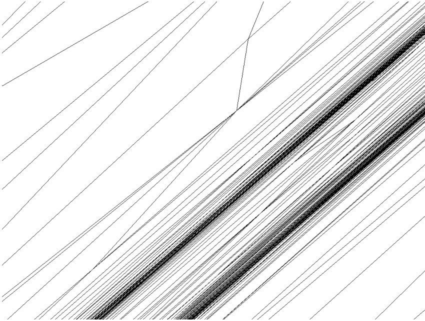
3 299 367 vertices and 19 264 402 tets.

3 299 367 vertices and 19 264 402 tets.

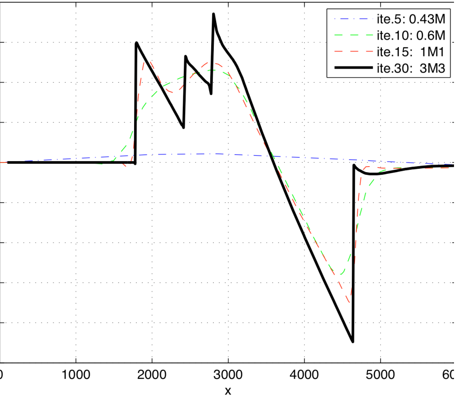


$R = 5$ km



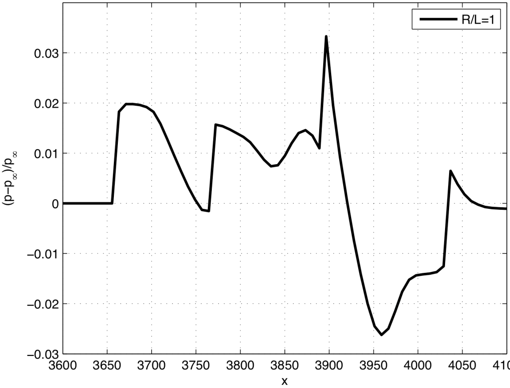
$R = 9$ km

3 299 367 vertices and 19 264 402 tets.



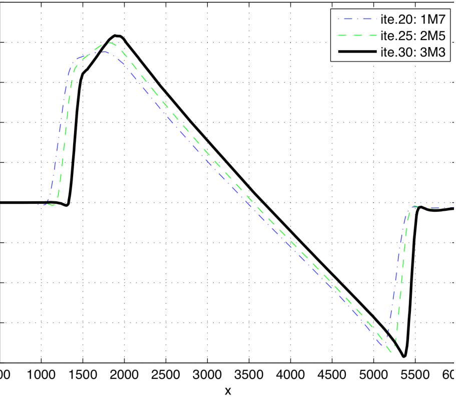
ite.5: 0.43M
ite.10: 0.6M
ite.15: 1M1
ite.30: 3M3
1000
2000
3000
4000
5000
x

$R = 1$ km

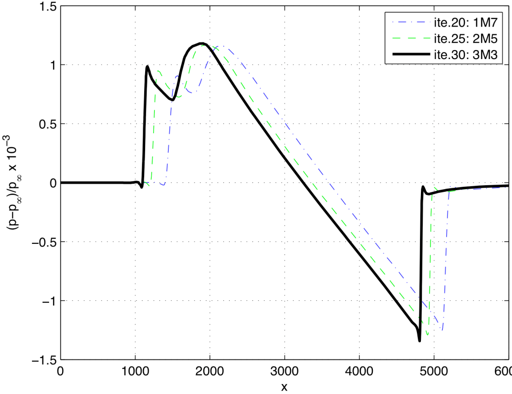


R/L=1
(p−p∞)/p∞
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
3600
3650
3700
3750
3800
3850
3900
3950
4000
4050
x

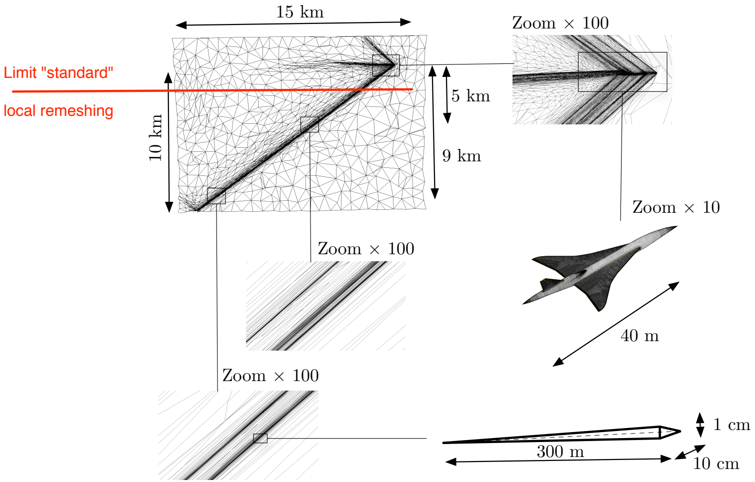
$R = 43$ m


ite.20: 1M7
ite.25: 2M5
ite.30: 3M3
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
x

$R = 9$ km


ite.20: 1M7
ite.25: 2M5
ite.30: 3M3
1.5
1
0.5
0
−0.5
−1
−1.5
(p−p∞)/p∞ x 10−3
0
1000
2000
3000
4000
5000
x

$R = 5$ km



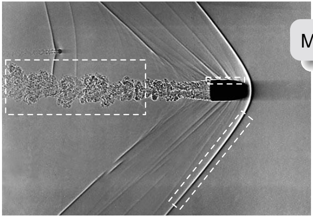
- Error estimate: $\mathbf{L}^2$ estimates $\Longrightarrow$ no h_{min} and small scales
- Solver : Implicit time-stepping
- Adaptation: anisotropy and quality $\Longrightarrow$ accuracy and stability

Outline

Mesh Generation Algorithms

3. Volume cavity-based operators
4. Surface cavity-based operators
5. Hybrid cavity-based operators

Problematics : Mesh Generation



Many phenomena $\Longrightarrow$ Many kinds of meshes

- Turbulent flow: isotropic,structured, ...
- Shock waves: anisotropic $O(1/100 - 1000)$
- Boundary-layers: quasi-structured $O(1/10^4 - 10^6)$

Frontal	High-Quality	Small Anisotropy
Delaunay	Robust	Anisotropy but Bad Quality
Octree-based	Robust	Surface mesh not constrained
Cartesian	Robust	Low Anisotropy, viscous effects
BL Extrusion		Closure of the domain, adaptivity
Local Refinement	Robust	Slow, High Anisotropy but Bad Quality

$\Longrightarrow$ No Unique Technology
$\Longrightarrow$ Robustness decreases with Geometry Complexity

[Coupez, Forge3D, CEMEF], [George et al., GHS3D, INRIA], [Ito,UAB-JAXA], [Löhner, Gen3D, GMU]
[Marcum et al., AFLR, MSU], [Oubay et al., Swansea U.], [Rassineux, UTC], [Remacle et al., Louvain U.]
[Shepard et al., MeshAdapt, SCOREC], [Schöberl et al., NETGEN, JKU], [Si, TetGen, WIAS], [Yvinec et al., CGAL, INRIA]

Scope and overview of the approach

Robustness is the primary concern

1. Local mesh modification operators
   - adaptivity is an iterative procedure
   - no mesh $\Longrightarrow$ no solution
   - always a valid mesh on output
   - use of simplicial meshes

Handling all types of meshes is the secondary concern

2. Unique operator
   - mesh adaptation : surface-volume
   - mesh optimization: edge-face swaps, point smoothing
   - boundary layer mesh generation: hybrid entities insertion

Outline

Cavity-based operators

- Generalization of edge-based operators
- Each operator is either an insertion or a re-insertion

insertion, collapse, edge-face swaps,
smoothing, surface projection,
quasi-structured layers generation, . . .

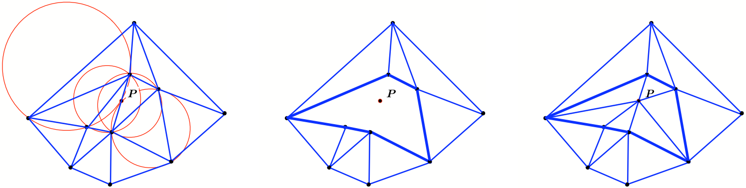
3. Volume operators
4. Surface operators
5. Hybrid operators (Boundary Layer)

Outline

Volume operators: Generality

Insertion of P (incremental Delaunay context)

$$\mathcal{H}_{k+1} = \mathcal{H}_k - \mathcal{C}_P + \mathcal{B}_P$$



$\mathcal{H}_k$ $\quad$ $\mathcal{H}_k - \mathcal{C}_p$ $\quad$ $\mathcal{H}_{k+1}$

[see authors: Baker, Borouchaki, Chen, Chrisochoides, George, Miller, Rivara, Shewchuck, Shimada, Si, Simpson, Wang, Weatherill, CG community. . .]

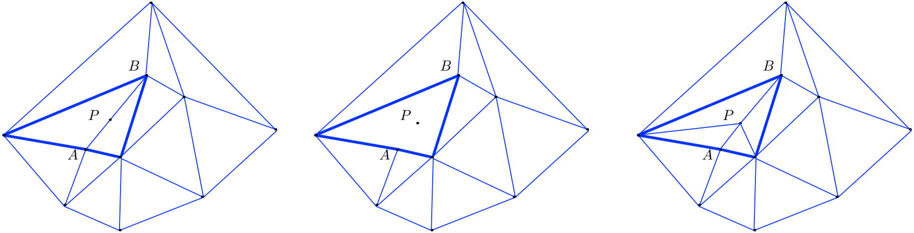
Validity principle

a) $\mathcal{H}_k$ is valid
b) $\mathcal{C}_P$ is connected by faces $\quad \Longrightarrow \mathcal{H}_{k+1}$ is valid
c) P visible from external face of $\mathcal{C}_P$

Volume operators: Initial choice of cavity

Different choices of $\mathcal{C}_P$ lead to different operators

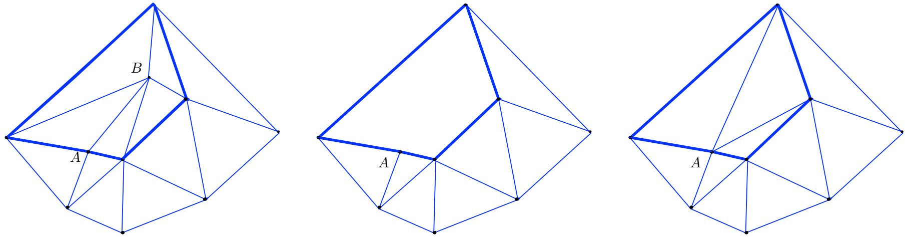
- edge-insertion:
  - $P \in [A, B]$
  - $\mathcal{C}_P = \text{shell}(A, B)$
  - insert P



$\mathcal{H}_k$ | $\mathcal{H}_k - \text{shell}(A, B)$ | $\mathcal{H}_{k+1} = \mathcal{H}_k - \text{shell}(A, B) + \text{star}(P)$

Different choices of $\mathcal{C}_P$ lead to different operators

- edge-insertion
- edge-collapse:
  - $\mathcal{C}_A = \text{ball}(B)$
  - re-insert A



$\mathcal{H}_k$ $\qquad$ $\mathcal{H}_k - \text{ball}(B)$ $\qquad$ $\mathcal{H}_{k+1} = \mathcal{H}_k - \text{ball}(B) + \text{star}(A)$

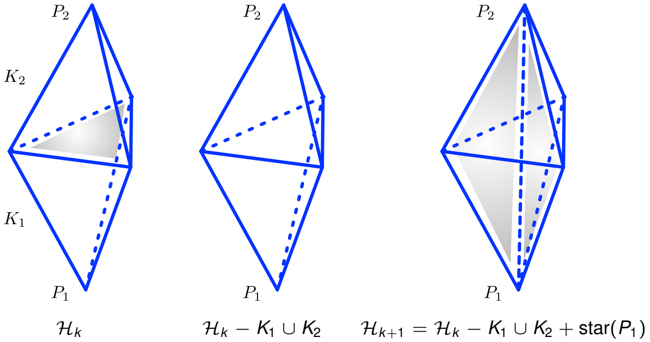
Volume operators: Initial choice of cavity

Different choices of $\mathcal{C}_P$ lead to different operators

- edge-insertion
- edge-collapse
- point smoothing/moving:
  - $\mathcal{C}_A = \text{ball}(A)$
  - re-insert A

Different choices of $\mathcal{C}_P$ lead to different operators

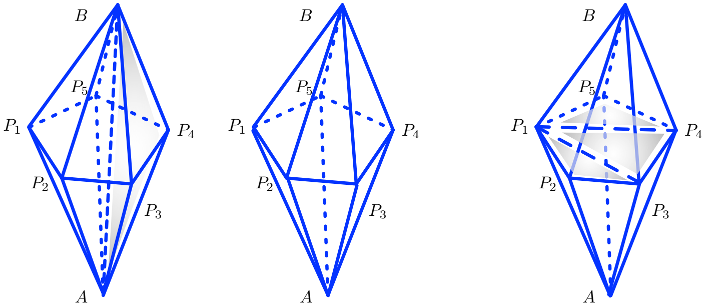
- edge-insertion, edge-collapse, point smoothing/moving
- face-swap
  - $\mathcal{C}_{P_i} = K_1 \cup K_2$
  - re-insert P_i



Volume operators: Initial choice of cavity

Different choices of $\mathcal{C}_P$ lead to different operators

- edge-insertion, edge-collapse, point smoothing/moving
- face-swap
- edge-swap [A,B]
    - $\mathcal{C}_{P_i} = \text{shell}(A, B)$
    - re-insert P_i



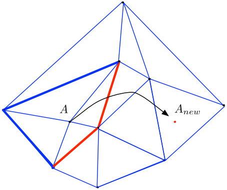
$\mathcal{H}_k$ $\qquad$ $\mathcal{H}_k - \text{shell}(A, B)$ $\qquad$ $\mathcal{H}_{k+1} = \mathcal{H}_k - \text{shell}(A, B) + \text{star}(P_1)$

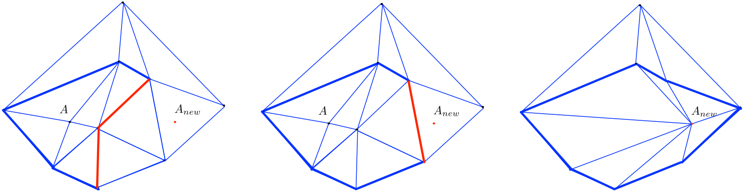
Volume operators: Cavity correction

if $\mathcal{B}_P$ is valid from these initializations
$\Longrightarrow$ we recover edge-based/standard operators

when $\mathcal{B}_P$ is invalid $\Longrightarrow$ Cavity corrections
$\Longrightarrow$ we define generalized operators

- Moving A to A_{new} is rejected





Iteration 1 Iteration 2 Iteration 3

Volume operators: Final considerations

Operators implemented in a metric-based framework
$(\mathcal{H}, \mathcal{M})$, *redefinition of length and quality*

Step 1: Generate a unit-mesh

- Collapse all edges of size lower than $1/\sqrt{2}$
- Split all edges of size greater than $\sqrt{2}$

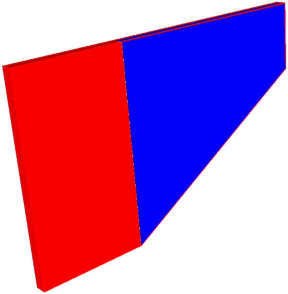
Step 2: Mesh optimization

- Perform point smoothing to improve $Q_{\mathcal{M}}$
- Perform edge and face swaps to improve $Q_{\mathcal{M}}$

3D double mach reflection

Ensuring optimality for unsteady simulations
anisotropy <> quality <> minimal time step

- 8-processors 64-bits MacPro with an IntelCore2 chipsets with a clockspeed of 2.8GHz with 32Gb of RAM
- Unsteady multi-scale error estimate [Alauzet et Olivier, AIAA 2011]
- Feflo compressible flow solver [Löhner, see AIAA from 1996 to 2013]
- 30 mesh adaptations, 5 fixed point iterations, 21 metric intersection in time
- Simultion CPU time 8h55m (Computation: first 1m30s and last 1h56m)
- 80% Solver, 20% mesh adaptation



Mesh of size N with an accuracy of h:

$$\frac{h}{2} \rightsquigarrow 8N \quad \text{and} \quad dt \sim h_{min} \rightsquigarrow \frac{dt}{2} \quad \Longrightarrow \quad \text{CPU} \times \mathbf{16}$$

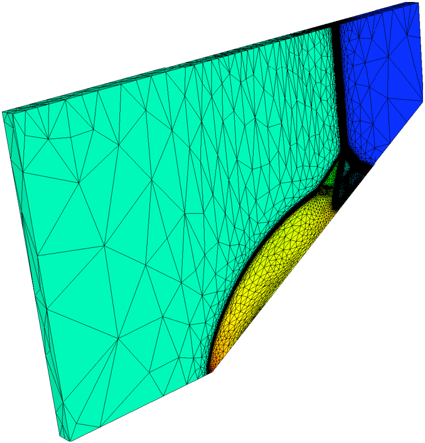
$$\frac{h}{4} \rightsquigarrow 64N \quad \text{and} \quad dt \sim h_{min} \rightsquigarrow \frac{dt}{4} \quad \Longrightarrow \quad \text{CPU} \times \mathbf{256}$$

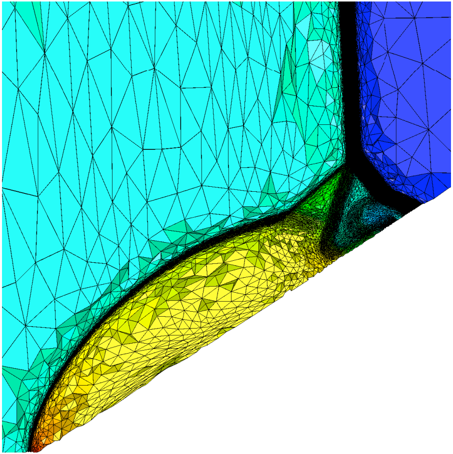
the quality of generated mesh must be perfect
$\Rightarrow$ NO bad element

$$dt \sim h_{min} \Longrightarrow h_{min} = 0.01 h_{target} \rightsquigarrow \text{CPU} \times \mathbf{100}$$

3D double mach reflection

Ensuring optimality for unsteady simulations
anisotropy <> quality <> minimal time step

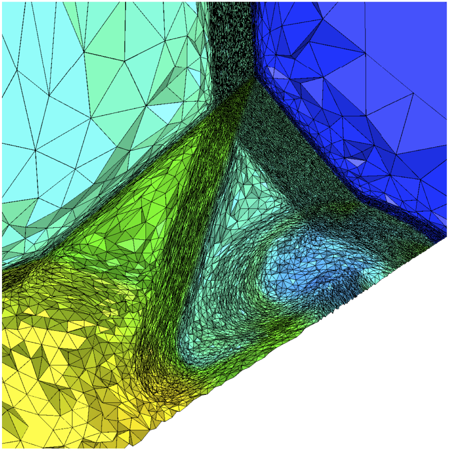


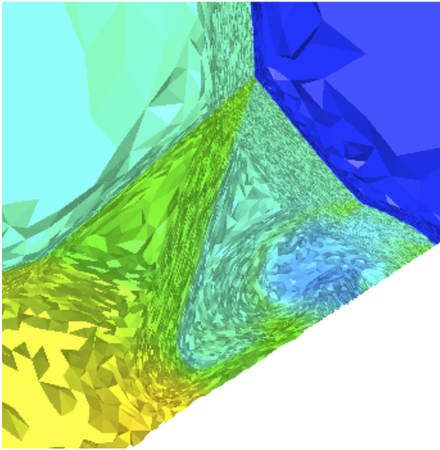


235 095 vertices, 1 310 082 tetrahedra and 57 864 boundary faces

3D double mach reflection

Ensuring optimality for unsteady simulations
anisotropy <> quality <> minimal time step





235 095 vertices, 1 310 082 tetrahedra and 57 864 boundary faces

Volume operators

Features of *generalized* cavity-based volume operators

- Embed (multiples) collapses/swaps in one call of the operator
  Improve locally the mesh quality
- Additional more restrictive control possible (tetrahedra altitude)
  Ensure optimal CPU for the flow solver
- No more threshold based on quality
  Faster convergence
- Only ONE operator throughout the code

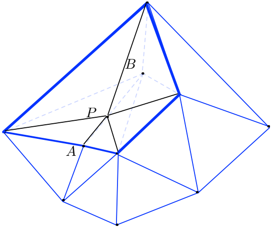
Outline

Surface cavity

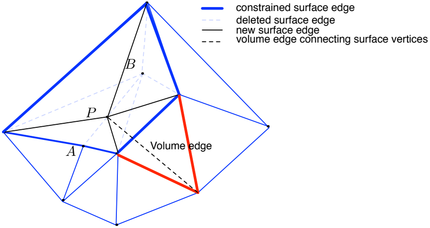
- Check geometric surface approximation
- Check topology conformity (patches, lines, ridges, corners)
- Check manifold components

Volume cavity

- $\mathcal{C}_P = \bigcup_{edge\ P_i, P_j} \text{shell}(P_i, P_j)$
- Apply previous volume corrections
- Check cavity conformity between surface-volume



$\Longrightarrow$ After correction(s)



Surface operators (2/2)

Curvature-based metric

[Borouchaki and Frey, 1997, Frey, IMR 2000, Aubry et al., JCP 2013]

$$\mathcal{M}_S(\varepsilon) = (\mathbf{u}_S, \mathbf{v}_S, \mathbf{n}_i) \begin{pmatrix} \frac{\lambda_{1,S}}{\varepsilon} & 0 & \\ 0 & \frac{\lambda_{2,S}}{\varepsilon} & 0 \\ 0 & 0 & h_{max}^{-2} \end{pmatrix} {}^t(\mathbf{u}_S, \mathbf{v}_S, \mathbf{n}_i)$$

- Local surface approximation : no global parameter, . . .
- Background surface mesh, background discrete metric
- Projection on background surface mesh

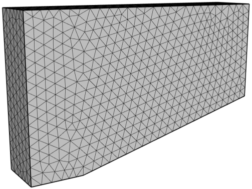
Cavity enhancements to limit volume cavity growth

- Steiner point to ease surface point insertion
- On-the-fly cavity retrianglulation
- On-the-fly cavity optimization (2-3 face swaps)

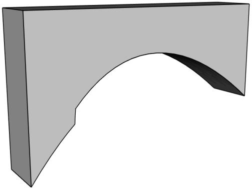
Why *generalized* operator(s) ?

Just to be able to adapt correctly the surface mesh while preserving the volume mesh

- Inserting/projecting a surface point to a new position
- Avoid dependance on the volume mesh



initial mesh

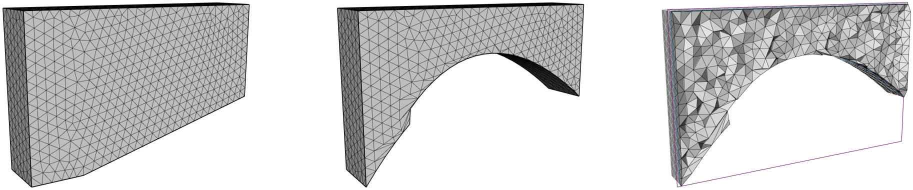


continuous geometry

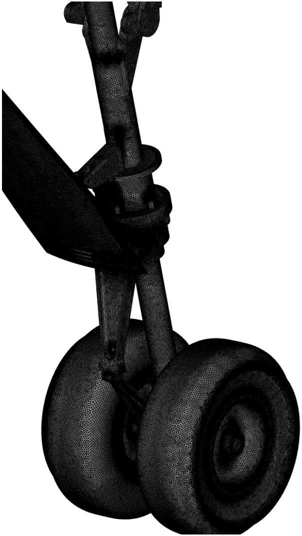
Why *generalized* operator(s) ?

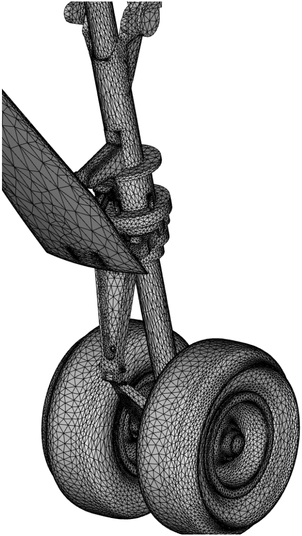
Just to be able to adapt correctly the surface mesh while preserving the volume mesh

- Inserting/projecting a surface point to a new position
- Avoid dependance on the volume mesh

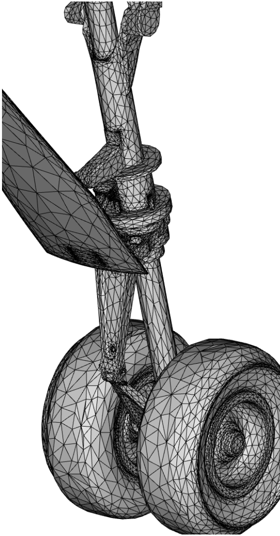


$\Longrightarrow$ Standard movement/smoothing always REJECTED
$\Longrightarrow$ One call of Generalized point smoothing
$\Longrightarrow$ Surface mesh adaptation with a boundary layer (volume) mesh





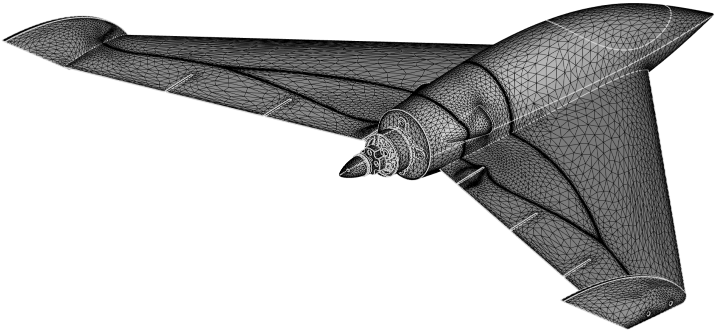
$\varepsilon = 0.01$



$\varepsilon = 0.1$

Blending surface approximation $\mathcal{M}_S$ and computational metric $\mathcal{M}_{\mathbf{L}^p}$

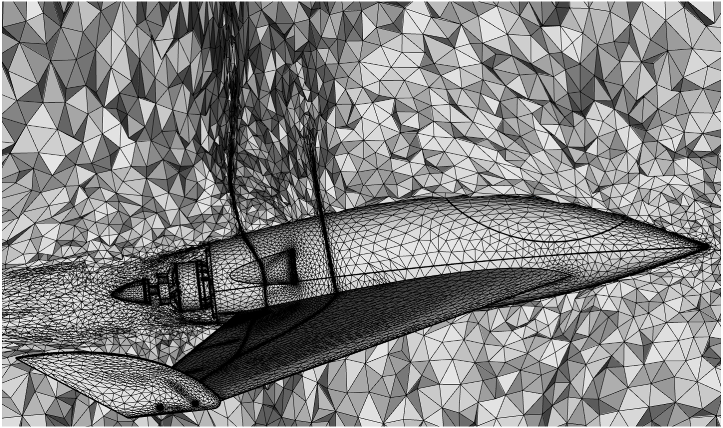
- Transsonic flight at Mach 0.7
- Multi-scale metric [Loseille and Alauzet, IMR 2009]
- Wolf flow solver [Alauzet and Loseille, JCP 2009]



43 000 surface points 85 000 triangles

Blending surface approximation $\mathcal{M}_S$ and computational metric $\mathcal{M}_{\mathbf{L}^p}$

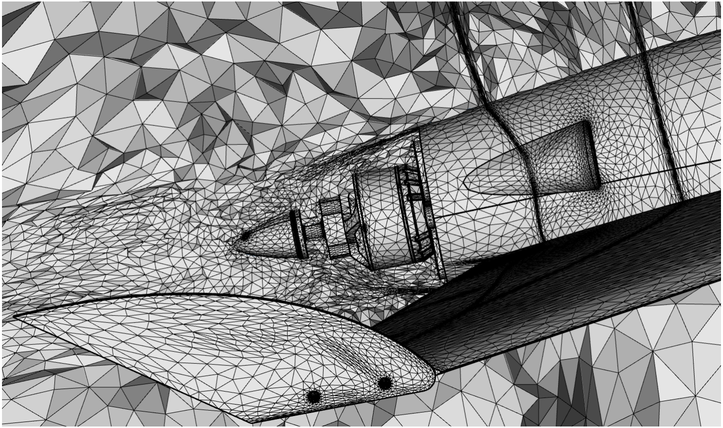
- Transsonic flight at Mach 0.7
- Multi-scale metric [Loseille and Alauzet, IMR 2009]
- Wolf flow solver [Alauzet and Loseille, JCP 2009]



345 000 vertices 85 000 triangles 2 million tetrahedra

Blending surface approximation $\mathcal{M}_S$ and computational metric $\mathcal{M}_{\mathbf{L}^p}$

- Transsonic flight at Mach 0.7
- Multi-scale metric [Loseille and Alauzet, IMR 2009]
- Wolf flow solver [Alauzet and Loseille, JCP 2009]



345 000 vertices 85 000 triangles 2 million tetrahedra

Surface operators

Features of *generalized* cavity-based surface operators

- Embed collapses/swaps in one call of the operator
  Improve locally the mesh quality
- Surface points are directly inserted to the *desired* position
  Remove the need of CPU-intensive, elasticity-based moving
- Surface remeshing becomes independent of the volume mesh
  Remeshing with boundary layers

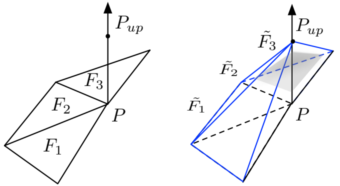
Outline

Hybrid entities insertion for quasi-structured mesh generation

- Given a starting surface mesh $\mathcal{S} = (F_i)_i$
- Given a set of normals (directions of extrusion) $(\mathbf{n}_j)_j$
- Given visibility conditions $(\mathbf{n}_i, F_{k_1}, \ldots, F_{k_n})$

Constrained insertion of P_{up} from $P, (\mathbf{n}_i, F_{k_1}, \ldots, F_{k_n})$

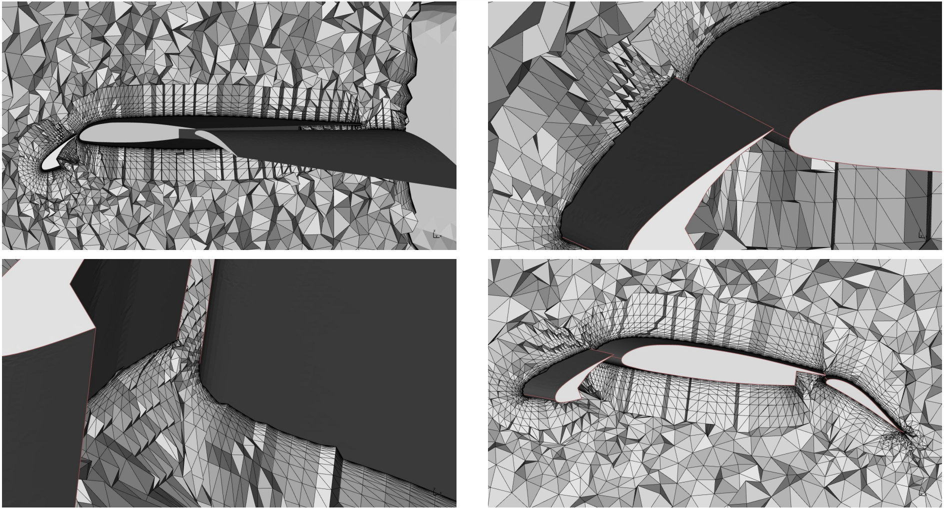
$$\mathcal{H}_{k+1} = \mathcal{H}_k - (\mathcal{C}_P - \mathcal{K}) + \mathcal{B}_P$$
$$\mathcal{S}_{k+1} = S_k - \cup_i F_{k_i} + \cup_i \tilde{F}_{k_i}$$



- Front surface S is updated
- $\mathcal{K}$ is updated with elements having one F_{k_i} as face
- Surface cavity checks are applied to S_{k+1}

This operator generates quasi-structured layers
hybrid entities depending on the nature of the faces

[Loseille and Löhner, IMR 2012]

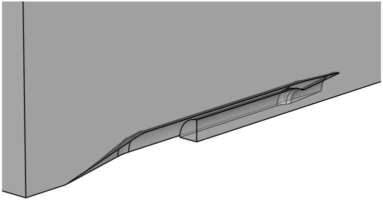


100 000 prisms/second, laptop Mac i7 @ 2,7 Ghz

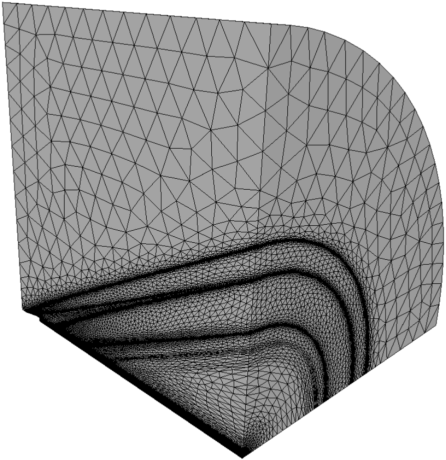
Plume exhaust example

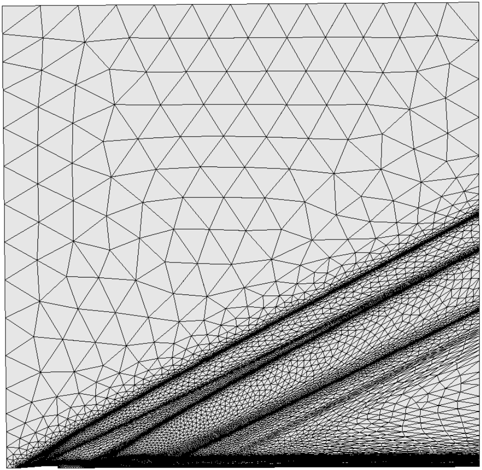
Blending $\mathcal{M}_S$, boundary layer and computational metric $\mathcal{M}_{\mathbf{L}^p}$

- Supersonic inflow at mach 2.2
- Laminar, Reynolds number of 1.8 Million.
- Surface/volume remeshing
- Mixed structured/unstructured boundary layer
- Adaptation on the density/mach
- Feflo flow solver [Löhner, see AIAA from 1996 to 2013]
- 6 hours on 8 procs (Mac book pro)



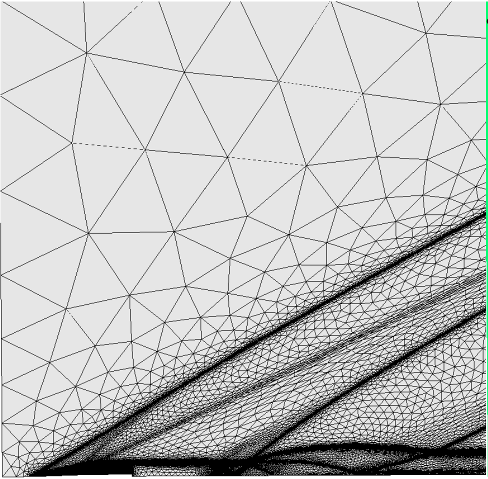
Plume exhaust example

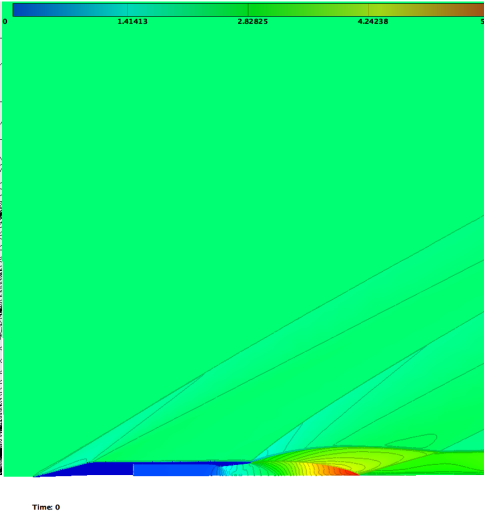




- 1,4 million of vertices, 8 millions of tets
- 10 quasi-structured layers recovered at each adaptation

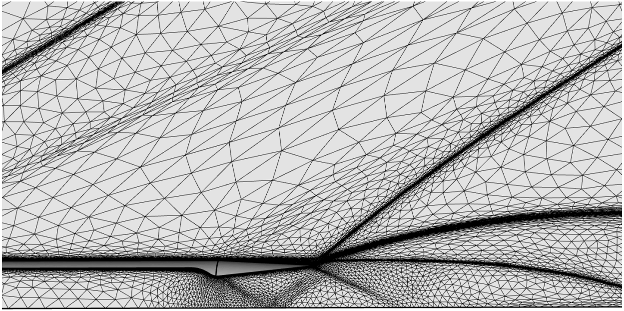
Plume exhaust example

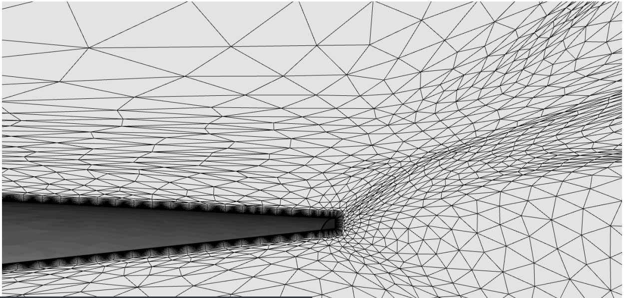




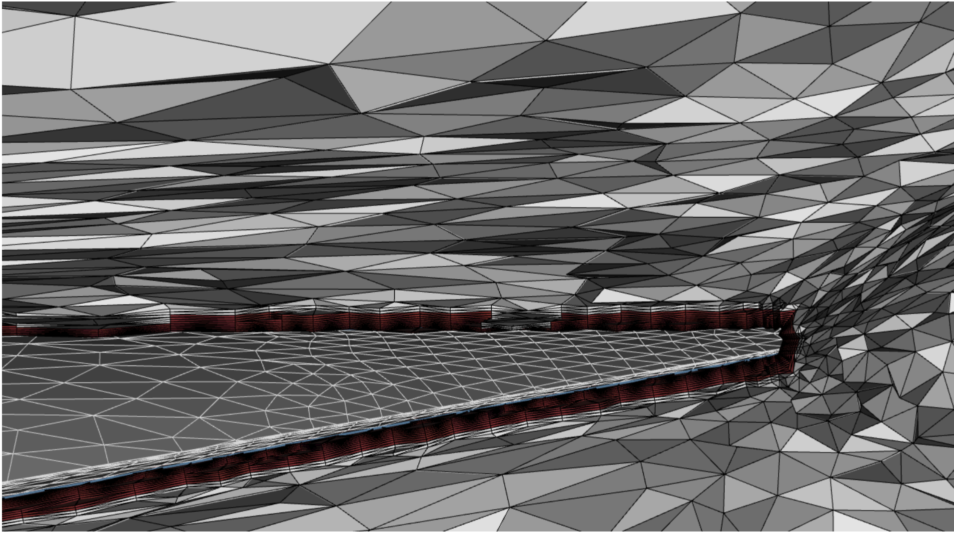
- 1,4 million of vertices, 8 millions of tets
- 10 quasi-structured layers recovered at each adaptation

Plume exhaust example





Plume exhaust example



- 1,4 million of vertices, 8 millions of tets
- 25 quasi-structured layers recovered at each adaptation

Conclusion

Cavity-based local operators

- unique operator with multiple cavity initializations/corrections
- adaptive code ($\approx$ 150 000 lines of code)
  $\Longrightarrow$ ease of code robustness/maintenance/improvements

Achievements

- Surface and volume remeshing in a adaptive robust context
- First runs of adaptive mesh adaptation with a mixed approach

Long term goals

- Fully adaptive hybrid mesh adaptation : boundary layer, cartesian, structured, anisotropic, uniform, ...
- Adaptivity for turbulent NS equations

Acknowledgments

- INRIA
  - Frédéric Alauzet
  - Alain Dervieux
  - Paul Louis George
  - Loïc Maréchal
- GMU CFD center:
  - Fernando Camelli
  - Rainald Löhner
- NRL:
  - Romain Aubry
- Boeing:
  - Todd R. Michal