Contributions of Applied Mathematics to Meshing Technologies and their Applications to Aerospace Simulations

Frédéric Alauzet

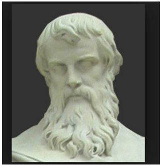
INRIA - Gamma3 Project - Saclay, France

Future CFD Technology Workshop - 2018

1. Contribution of computational geometry
2. Contribution of differential geometry
3. Contribution of set theory
4. Aerospace examples

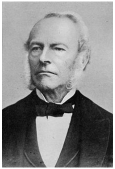
INVENTEURS DU MONDE NUMÉRIQUE

Meshing historical principles after some famous people:



-300
Euclidean Geometry

1785 - 1836



1819 - 1903

Boundary Layer

1826 - 1866
Riemannian Geometry
Anisotropy

1838 - 1916
Shock Wave
Mesh Adaptation

1862 - 1943
HPC

1868 - 1908
Diagram

1890 - 1980
Triangulation

1910 - 1999

1930 -

CAD
High-Order

Outline

1. Contribution of Computational Geometry
   -  
2. Contribution of Differential Geometry
3. Contribution of Set Theory
4. Anisotropic Mesh Adaptation for Steady Flows
5. Anisotropic Mesh Adaptation for Unsteady Flows
6. Conclusions and Remaining Challenges

Scientific Context

Numerical Simulation Pipeline

CAD $\longrightarrow$ **MESH** $\longrightarrow$ SOLVER $\longrightarrow$ VISU / ANALYSIS

1. *no mesh = no simulation*

Structured Mesh Generation Methods:

- Cartesian (grid) or IJK meshes
- Structured by block hex mesh generation
- Medial axis hex mesh generation
- Octree hex mesh generation (automatic)
- ...

(Too) difficult to handle complex geometries
Semi-automatic $\longrightarrow$ manual intervention is time-consuming
Automatic methods do not put structure where needed
Mesh adaptation is limited

Numerical Simulation Pipeline

CAD $\longrightarrow$ **MESH** $\longrightarrow$ SOLVER $\longrightarrow$ VISU / ANALYSIS

1. *no mesh* $=$ *no simulation*

Automated Unstructured Tetrahedra Mesh Generation Methods:

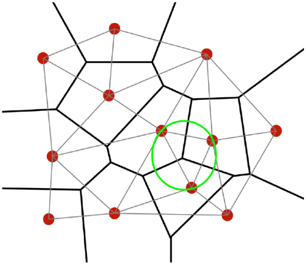
- Octree-like [Yerry and Shephard, IJNME 1984], ...
- Advancing front [Lohner and Parikh, IJNMF 1988], [Peraire et al., IJNME 1988], [Jin and Tanner, IJNME 1991], ...
- Delaunay [Hermeline, RAIRO AN 1982], [Baker, AIAA 1987], [George, Hecht and Saltel, ICSE 1990], [Weatherhill, CMA 1992], ...
- Minimal volume [Coupez, REEF 2000], ...
- Coupled Delaunay-frontal [Marcum and Weatherhill, AIAA 1995], ...

At the end of 90's

3D powerful and mature mesh generation methods become available

Properties:

- Dual of the Voronoï Diagram (nearest neighbor diagram)
  Delaunay triangulation connect the site that share a diagram edge

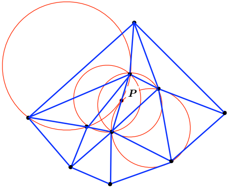


- Unicity
- Empty circle/sphere property
- Maximize the smallest angle
- Locally Delaunay everywhere $\Longleftrightarrow$ Globally Delaunay
- ...

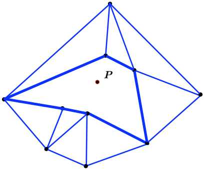
Locally Delaunay everywhere $\Longleftrightarrow$ Globally Delaunay
Insertion of P (incremental Delaunay context)

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

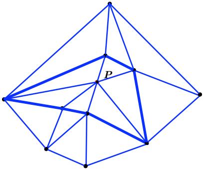
[Bowyer, CJ 1981], [Watson, CJ 1981], [Hermeline, RAIRO 1982], ...



$\mathcal{H}_k$



$\mathcal{H}_k - \mathcal{C}_p$



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

Delaunay criteria principle

a) $\mathcal{H}_k$ is Delaunay
b) Circumcircles of elements of $\mathcal{C}_P$ contain point P $\Longrightarrow$ $\mathcal{H}_{k+1}$ is Delaunay

Delaunay-Based Meshing

But Delaunay criteria is not suitable for meshing:

- In 2D, the Delaunay criteria is NOT a quality criteria for highly anisotropic meshes
  We want to minimize the maximal angle [Barth, AIAA 1991]
- In 3D, the Delaunay criteria is NOT a quality criteria (slivers)

Robust extension to meshing $\Longrightarrow$ Constraint cavity and cavity correction

[George et al, ICSE 1990], [George et al, CMAME 1991], [George et al, IJNME 1992], ...

In 2001, R. Löhner (GMU) in his book wrote:

"**The best way to avoid slivers is by relaxing the Delaunay criterion** ... This fundamental departure from the traditionnel Delaunay criterion, first proposed by George et al. (1990) to the chagrin of many mathematicians and computational geometers has allowed this class of unstructured grid generation algorithms to produce reliably quality grids. **It is a simple change, but has made the difference between a theoretical exercise and a practical tool.**"

Unique Cavity-Based Operator

Difficulties:

- Many mesh modification operators: split, collapse, swap, relocation and all possible combinations
- Many kinds of elements: triangles, quads, tetrahedra, prisms, pyramids, hex, ...
- Many kinds of meshes: surface, volume, boundary layer, structured, curved, ...
- Severals kinds of geometry: manifold, non manifold, ...

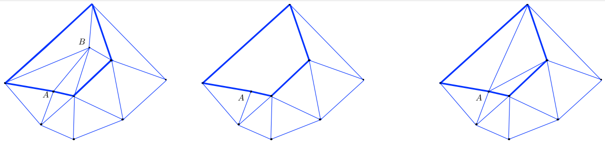
Conclusion:
It becomes too difficult to (i) maintain the code and (ii) gather all functionalities

We propose a **unique cavity-based operator** inspired from the Delaunay method

Unique meshing operator

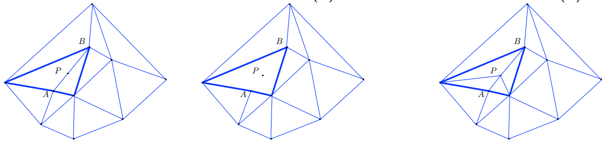
Each operator $\equiv$ a node (re)insertion: $\mathcal{H}^{k+1} \equiv \mathcal{H}^k - \mathcal{C} + \mathcal{R}$

Collapse



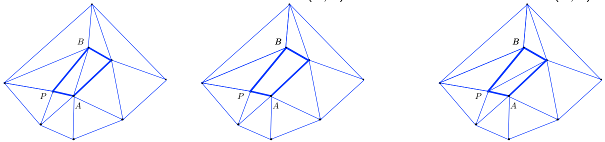
$\mathcal{H}^k$ | $\mathcal{H}^k - \mathcal{C}_{ball(B)}$ | $\mathcal{H}^{k+1} = \mathcal{H}^k - \mathcal{C}_{ball(B)} + \mathcal{R}_A$

Insertion



$\mathcal{H}^k$ | $\mathcal{H}^k - \mathcal{C}_{shell(A,B)}$ | $\mathcal{H}^{k+1} = \mathcal{H}^k - \mathcal{C}_{shell(A,B)} + \mathcal{R}_P$

Swap



$\mathcal{H}^k$ | $\mathcal{H}^k - \mathcal{C}_{shell(A,B)}$ | $\mathcal{H}^{k+1} = \mathcal{H}^k - \mathcal{C}_{shell(A,B)} + \mathcal{R}_P$

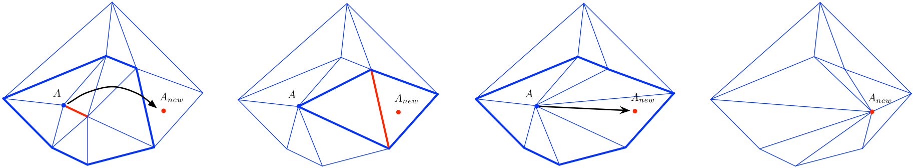
Unique meshing operator

Each operator $\equiv$ a node (re)insertion: $\mathcal{H}^{k+1} \equiv \mathcal{H}^k - \mathcal{C} + \mathcal{R}$

Cavity correction(s) to create combination of meshing operators

Example: Relocate vertex A to new position A_{new} requires

1 edge collapse + 1 edge swap + 1 vertex relocation



Collapse   Swap   Relocation 3   Final

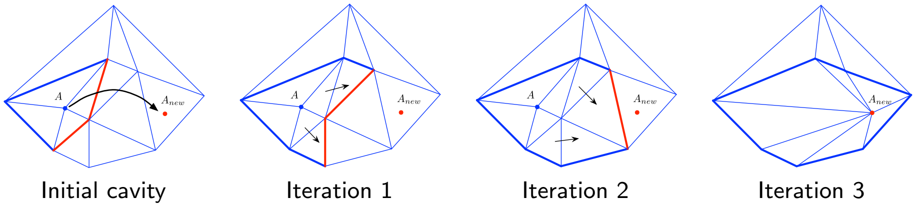
Unique meshing operator

Each operator $\equiv$ a node (re)insertion: $\mathcal{H}^{k+1} \equiv \mathcal{H}^{k} - \mathcal{C} + \mathcal{R}$

Cavity correction(s) to create combination of meshing operators

Example: Relocate vertex A to new position A_{new} requires

1 node reinsertion with the appropriate cavity definition



Red edges have no visibility w.r.t $A_{new} \Longrightarrow$ Correct (enlarge) the cavity
Final cavity bold blue edges

Outline

Scientific Context

Numerical Simulation Pipeline

CAD $\longrightarrow$ **MESH** $\longrightarrow$ SOLVER $\longrightarrow$ VISU / ANALYSIS

1. *no mesh = no simulation*
2. *a "bad" mesh implies a wrong or inaccurate solution*

Let's give two catastrophic examples

Numerical Simulation Pipeline

CAD $\longrightarrow$ **MESH** $\longrightarrow$ SOLVER $\longrightarrow$ VISU / ANALYSIS

1. *no mesh = no simulation*
2. *a "bad" mesh implies a wrong or inaccurate solution*

Examples of catastrophic events:

- Roissy Terminal 2E roof collapse
  [Feghaly, SC 2008]

- Sinking of Sleipner-A offshore platform
  [Jakobsen, SEI 1994], [Collins, CI 1997]

Due to errors in numerical simulations

Scientific Context

Numerical Simulation Pipeline

CAD $\longrightarrow$ **MESH** $\longrightarrow$ SOLVER $\longrightarrow$ VISU / ANALYSIS

1. *no mesh = no simulation*
2. *a "bad" mesh implies a wrong or inaccurate solution*

- Address ever increasing geometrical complexity
- Address ever increasing physical complexity
- Address the large variety of numerical schemes
- Address convergence studies in 3D

Require **tailored** meshes to address and certify numerical results

$\downarrow$

Modify discretization of Ω to control numerical solution accuracy

How to prescribe size in any directions

Main idea: change mesh generator distance and volume computation

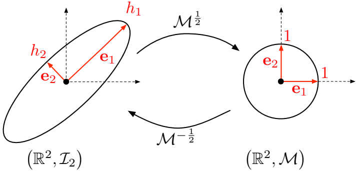
[George, Hecht and Vallet., Adv. Eng. Software 1991]

Fundamental concept: The notion of **metric** and Riemannian metric space

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

$$\langle \mathbf{u}\,,\,\mathbf{v}\rangle_{\mathcal{M}} = {}^t\mathbf{u}\mathcal{M}\mathbf{v} \quad \Longrightarrow \quad \ell_{\mathcal{M}}(\mathbf{a},\mathbf{b}) = \sqrt{{}^t\mathbf{ab}\ \mathcal{M}\ \mathbf{ab}}$$
$$|K|_{\mathcal{M}} \quad = \quad \sqrt{\det \mathcal{M}}\,|K|$$

Distance unit ball is an ellipse



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

$$\ell_{\mathcal{M}}(\mathbf{ab}) \quad = \quad \int_0^1 \sqrt{{}^t\mathbf{ab}\ \mathcal{M}(\mathbf{a} + t\mathbf{ab})\ \mathbf{ab}}\,\mathrm{d}t$$
$$|K|_{\mathcal{M}} \quad = \quad \int_K \sqrt{\det \mathcal{M}}\,\mathrm{d}K$$

How to prescribe size in any directions

Main idea: change mesh generator distance and volume computation

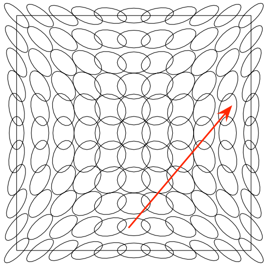
[George, Hecht and Vallet., Adv. Eng. Software 1991]

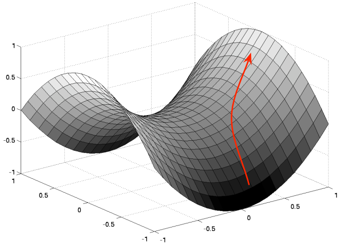
Fundamental concept: The notion of **metric** and Riemannian metric space

Computing geometric quantities in Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

$$\Updownarrow$$

Computing geometric quantities on $\mathcal{S}$





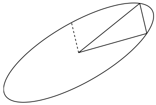
How to prescribe size in any directions

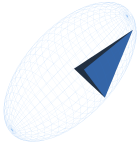
Main idea: change mesh generator distance and volume computation

[George, Hecht and Vallet., Adv. Eng. Software 1991]

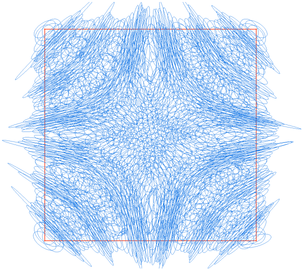
Fundamental concept: Generate a **unit mesh** w.r.t $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

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

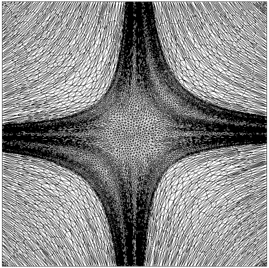




Inputs $(\mathcal{H}_0, \mathcal{M}_i)_{i\in\mathcal{H}}$



Output $\mathcal{H}$



Continuous Mesh Framework

We proposed a continuous mesh framework to theorize mesh adaptation

[Alauzet et al., IMR 2006], [Alauzet, IJNMF 2008], [Loseille and Alauzet, SINUM 2010]

Discrete	**Continuous**
Element K	Metric tensor $\mathcal{M}$
Volume $\lvert K \rvert$	Volume $\alpha\,(\det \mathcal{M})^{-\frac{1}{2}}$
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 $\pi_{\mathcal{M}} u$

Local interpolation error duality

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

$$\|u-\Pi_h u\|_{L^1(K)} = \frac{\sqrt{2}}{240} \underbrace{\det(\mathcal{M}^{-\frac{1}{2}})}_{\textit{mapping}} \underbrace{\operatorname{trace}(\mathcal{M}^{-\frac{1}{2}}\, H_u\, \mathcal{M}^{-\frac{1}{2}})}_{\textit{anisotropic term}} = \|u-\pi_{\mathcal{M}} u\|_{L^1(\mathcal{B}(\mathcal{M}))}$$

Working in this framework enables us to use powerful mathematical tool

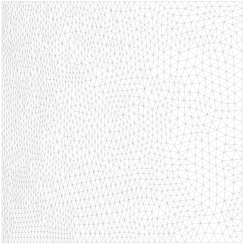
Continuous Mesh Framework

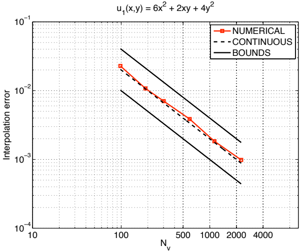
Set the sequence of 2D embedded continuous meshes $\mathbf{M}(\alpha) = (\mathcal{M}_\alpha(\mathbf{x}))_{\mathbf{x}\in\Omega}$ defined on $\Omega = [0,1]\times[0,1]$ by:

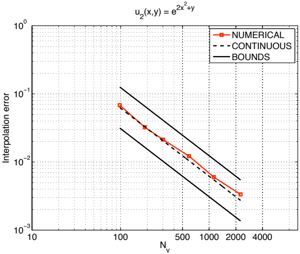
$$\mathcal{M}_\alpha(x,y) = \alpha \begin{pmatrix} h_1^{-2}(x,y) & 0 \\ 0 & h_2^{-2}(x,y) \end{pmatrix} \text{ with } \begin{array}{l} h_1(x,y) = 0.1(x+1) + 0.05(x-1) \\ h_2(x,y) = 0.2 \end{array}$$

Analyze the interpolation error of functions:

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







Application: Minimizing the Interpolation Error in L^p-norm

- An ill-posed discrete problem

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

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

- A well-posed continuous problem

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

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

$\Longrightarrow$ Solved by a calculus of variations

Continuous Mesh Framework

Optimal metric [Alauzet et al., IMR 2006], [Loseille and Alauzet, SINUM 2010]

$$\mathcal{M}^{opt}_{L^p}(\mathbf{x}) \;=\; {\color{red}N}^{\frac{2}{d}} \underbrace{\left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+d}}\right)^{-\frac{2}{d}}}_{1} \;\underbrace{(\det|H_u(\mathbf{x})|)^{\frac{-1}{2{\color{red}p}+d}}}_{2}\;\underbrace{|H_u(\mathbf{x})|}_{3}$$

- $\mathbf{M}^{opt}_{L^p}$ is unique
- $\mathbf{M}^{opt}_{L^p}$ has for optimal directions and ratios the Hessian ones
- $\mathbf{M}^{opt}_{L^p}(u)$ provides an optimal explicit bound of the interpolation error in L^p norm:

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

- Global second order of convergence for a sequence of embedded continuous meshes $(\mathbf{M}^N_{L^p}(u))_{N=1\ldots\infty}$

Error Estimates for Steady Problems

Feature-based anisotropic mesh adaptation [Loseille and Alauzet, IMR2009 & SINUM2011]

Deriving the **best mesh** to compute the characteristics of a given solution $\mathbf{w}$

To this end, optimal control of the interpolation error in L^p norm :

$$\|W - \Pi_h W\|_{L^p(\Omega_h)} \quad \Longrightarrow \quad \mathcal{M}_{L^p}(H_W) = D_{L^p}(N)\,(det|H_W|)^{\frac{-1}{2p+d}}\,|H_W|$$

Goal-oriented anisotropic mesh adaptation [Loseille et al., JCP2010], [Belme et al., submitted]

Deriving the **best mesh** to observe a given functional $\mathbf{j(w) = (g, w)}$

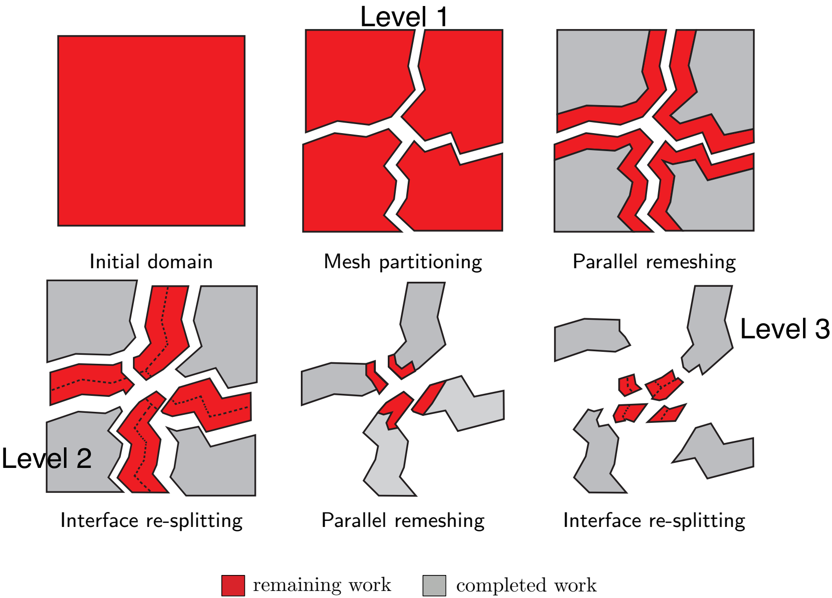
To this end, optimal control of the functional approximation error in L^1 norm

$$\begin{aligned}
&\|J(W) - J(W_h)\|_{L^1(\Omega_h)} \\
&\quad \approx \int_\Omega |W_h^*| \left|\nabla\cdot(\mathcal{F}^E(W) - \mathcal{F}^E(\Pi_h W)) - \nabla\cdot(\mathcal{F}^V(W) - \mathcal{F}^V(\Pi_h W))\right| \mathrm{d}\Omega \\
&\quad \leq \int_\Omega G_{\mathcal{F}^E}\left|\mathcal{F}^E(W) - \Pi_h\mathcal{F}^E(W)\right| \mathrm{d}\Omega + \int_\Omega G_T\,|T - \Pi_h T|\,\mathrm{d}\Omega \\
&\qquad + \int_\Omega G_{u_1}\,|u_1 - \Pi_h u_1|\,\mathrm{d}\Omega + \int_\Omega G_{u_2}\,|u_2 - \Pi_h u_2|\,\mathrm{d}\Omega + \int_\Omega G_{u_3}\,|u_3 - \Pi_h u_3|\,\mathrm{d}\Omega\,,
\end{aligned}$$

where the G depend on $|\nabla W^*|$ and $|H_{W^*}|$.

Outline

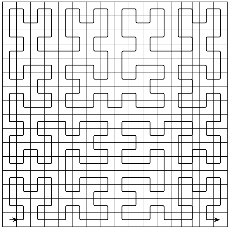
Coarse-Grained Parallel Process Overview

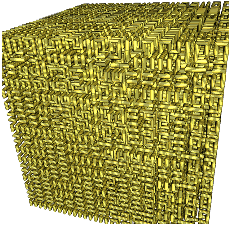


Mesh Partitioning Methods

Three methods have been considered:

1. Geometric: Hilbert space filling curves
   Ordering depends on the position in space (x, y, z)





2. Topologic: Breadth first search (BFS)
   Ordering depends on the connectivity of the mesh

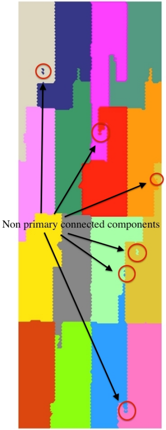
   Start from a germ, then add its neighbors, then add the neighbors of the neighbors, ... using a stack

3. Topologic: Breadth first search (BFS) with restart
   Ordering depends on the connectivity of the mesh

   Same as previously but only the first element is kept in the pile when a new partition is defined

Mesh Partitioning Methods

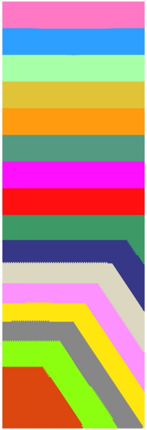
Level 1 partitioning:



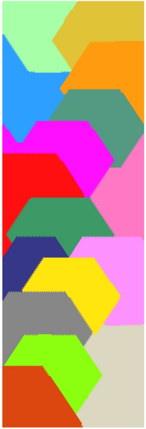
Hilbert

no correction

Hilbert



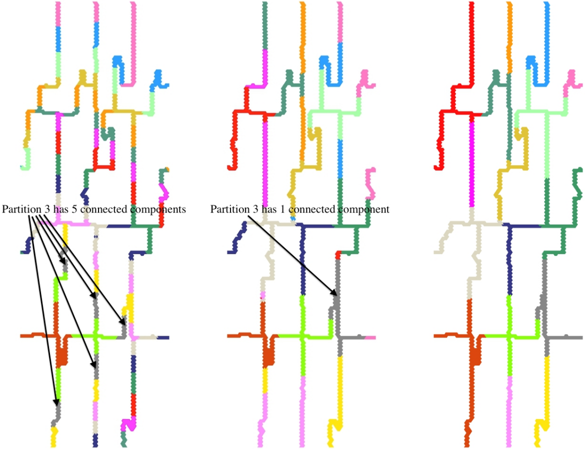
BFS



BFS restart

Mesh Partitioning Methods

Level 2 partitioning:

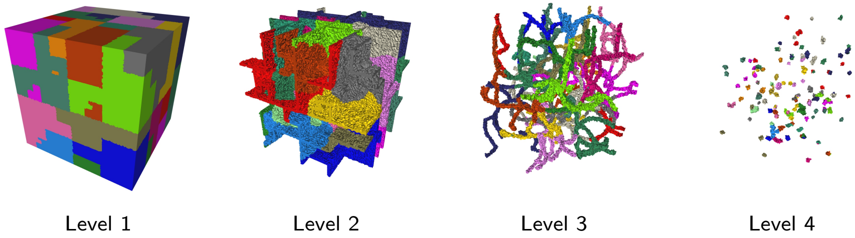


BFS no correction | BFS restart no correction | BFS restart

Hierarchical metric-based mesh partitioning

Interface size converges toward zero for hierarchical partitioning

- Level 1: Hilbert-based mesh partitioning
- Level $k > 1$: Interfaces: BFS restart partitioning
- Maximum numbers of level 5 (final level in serial)



Cluster

- 40 nodes with 48Gb of memory

  composed of two-chip Intel Xeon X56650 with 12 cores

$\Longrightarrow$ 480 procs

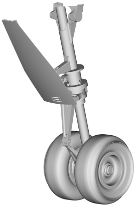
- A high-speed internal network InfiniBand (40Gb/s) connects these nodes

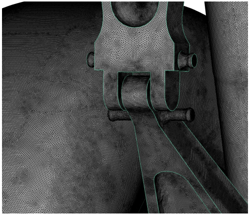
Presents results for 120 procs

- Targeted applications: Acoustic/Turbulent flows
- High-resolution of the complex geometry
- Initial mesh : 2 658 753 vertices 844 768 tris and 14 731 068 tets
- Final mesh : 183 334 265 vertices 14 263 732 tris and **1 081 733 853** tets
- Parallel remeshing time is 10min (7m36s) on 120 cores
  The total CPU time is 17 min (IOs, final gathering)
  The serial CPU time is 14h (speed-up 84 without IOs - 50 with IOs)

Speed (without IOs) is $1.8\,10^6$ ($2.37\,10^6$) elements / second

Level	% done	# of tets in interface	# of tets inserted	CPU time (sec.)	# of cores used
1	97 %	30 681 418	1 043 004 327	379	120
2	100 %	2 866 212	1 078 867 641	56	12
3	100 %	5 304	1 081 733 853	21	1



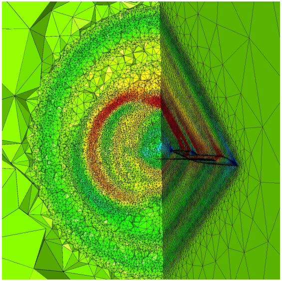


Outline

Supersonic Aircraft Simulation

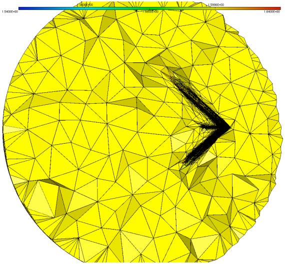
A bit of history: chronology of the results

[Alauzet, PhD 2003]
Isotropic
L^∞ error estimate



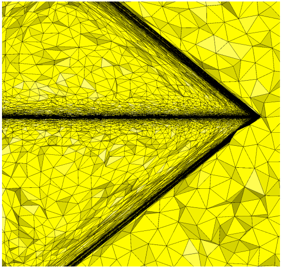
798 756 vertices
4 714 162 tetrahedra
$\frac{R}{L} = 1.25 \; (50m)$

[Alauzet, ECCOMAS 2006]
Anisotropic
L^∞ error estimate



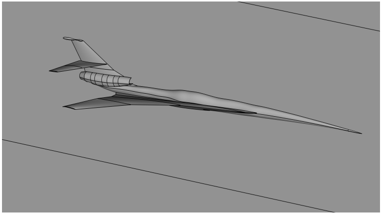
1 775 049 vertices
10 474 598 tetrahedra
$\frac{R}{L} = 3 \; (120m)$

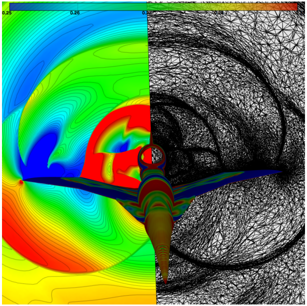
[Loseille et al., AIAA 2007]
Anisotropic
L^p error estimate

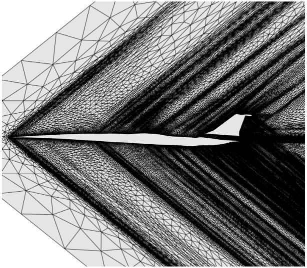


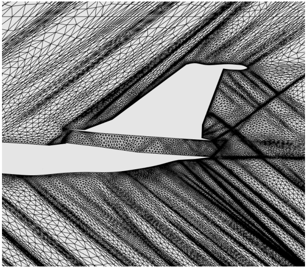
3 764 591 vertices
22 324 258 tetrahedra
$\frac{R}{L} = 40 \; (1.6km)$

AIAA Sonic Boom Workshop Results



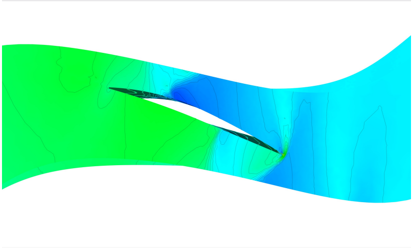


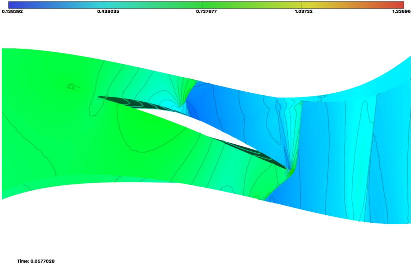


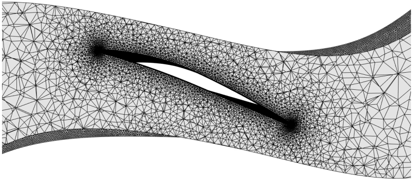


Initial mesh

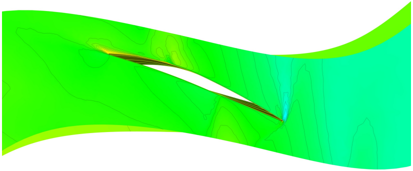
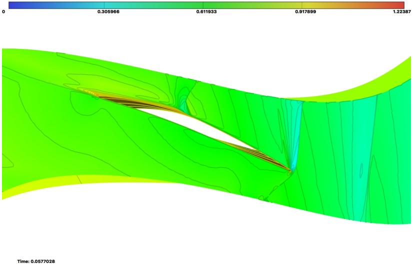
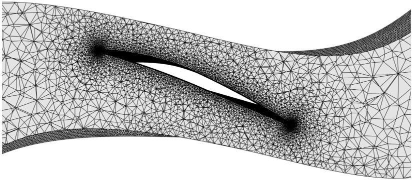
Adapted mesh



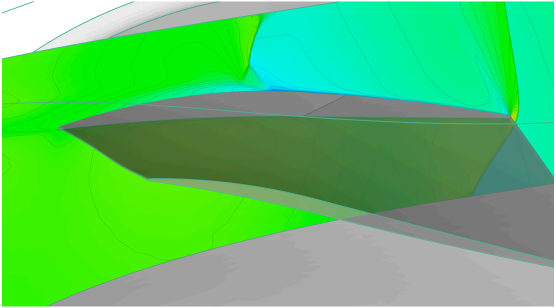


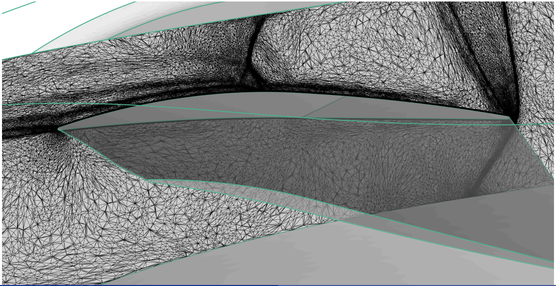


Turbomachinery Application: Turbulent NASA RO37

Initial mesh	Adapted mesh
	
	

Turbomachinery Application: Turbulent NASA RO37





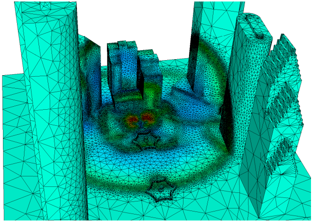
Outline

A bit of history: chronology of the results

[Frey and Alauzet, IMR 2003]

Isotropic $n_{adap} = 30$

$L^{\infty} - L^{\infty}$ error estimate



743 735 vertices

4 328 741 tetrahedra

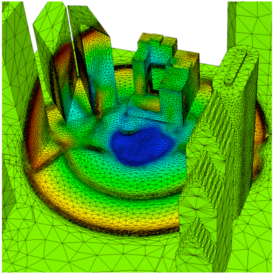
87 322 triangles

Accuracy > 30cm

[Alauzet and Olivier, AIAA 2011]

Anisotropic $n_{adap} = 40$

$L^{\infty} - L^{p}$ error estimate



185 148 vertices

1 027 537 tetrahedra

50 250 triangles

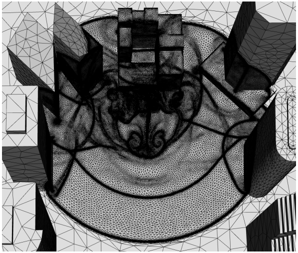
Accuracy 11cm

Mean quotient 56

[Alauzet et al., IMR 2014]

Anisotropic - $n_{adap} = 128$

$L^{p} - L^{p}$ error estimate



4 187 548 vertices

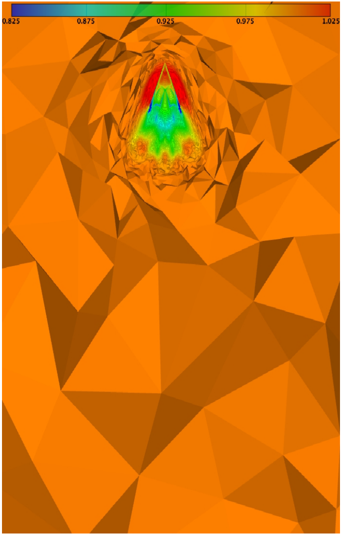
25 249 618 tetrahedra

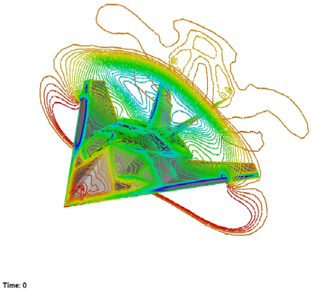
329 610 triangles

Accuracy 5mm

Mean quotient 249

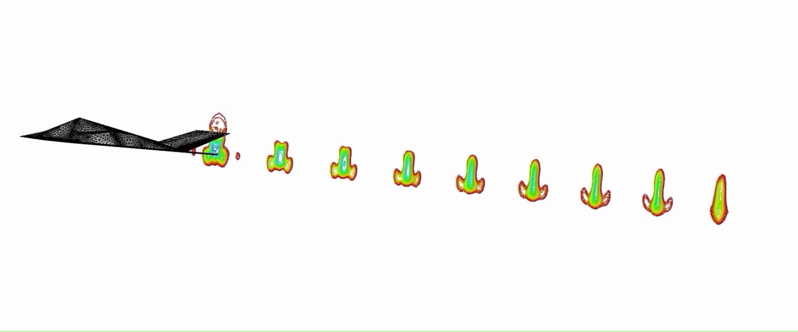
Vortical Flow Behind a F117 fighter





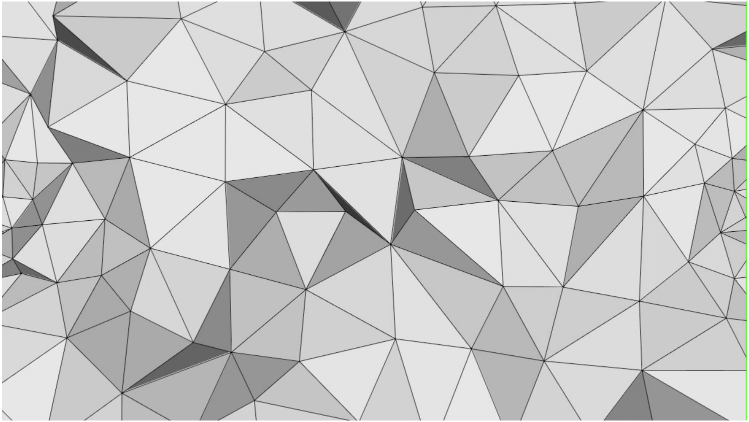
Nosing-up F117

- A nosing-up f117 creates a vortical wake
- Ascending and descending phases are split into an accelerated and a decelerated phase



Two F117s crossing flight paths

- Two aircrafts moved at Mach 0.4 inside inert air
- The planes are translated and rotated



Outline

Conclusions

Meshing technologies progresses have relied on Mathematics

- Euclidean and Riemannian geometries
- Computational geometry
- Differential geometry - calculus of variation
- Set theory - space filling curves
- Error estimate theory
- Adjoint techniques

and computer science

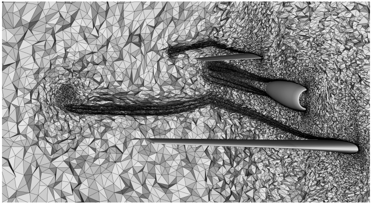
- Algorithms (hashing, coloring, pile, chain list, dynamic allocation, ...)
- HPC (MPI, p-threads, ...)

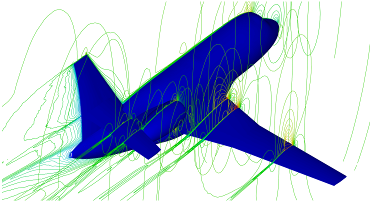
Interdisciplinary contributions are fundamental to meshing technologies

There is still a lot of work to do in meshing technologies for aerospace application. But, some recent advances are promising

Turbulent flows: RANS simulations

- Results on the ONERA M6 wing are encouraging (see the presentation of D. Kamenetskiy AIAA Paper 2018-0920 (Tuesday))
- We still have to work on the goal-oriented error estimate for the turbulence model
- Analyze the impact of the metric gradation control (blending)



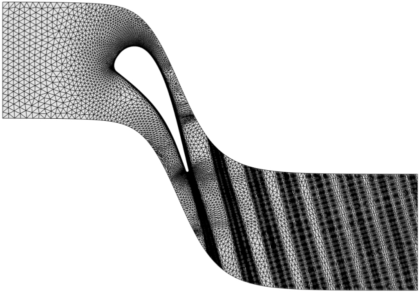


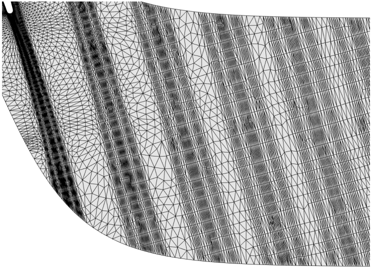
Remaining Challenges

There is still a lot of work to do in meshing technologies for aerospace application. But, some recent advances are promising

Metric-aligned and metric-orthogonal approaches

- Structured meshes with an automated unstructured method
- Still some slivers remaining in 3D
- Better metric field blending to avoid large angle 2-to-1 transition
- Convert full tet meshes to multi-elements meshes



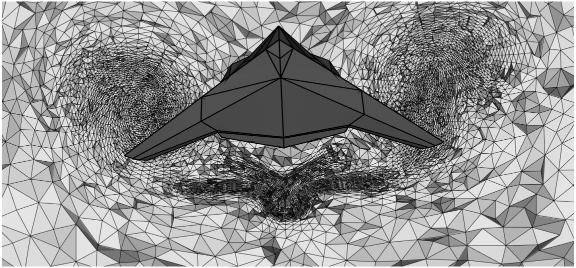


Remaining Challenges

There is still a lot of work to do in meshing technologies for aerospace application. But, some recent advances are promising

Metric-aligned and metric-orthogonal approaches

- Structured meshes with an automated unstructured method
- Still some slivers remaining in 3D
- Better metric field blending to avoid bad 2 to 1 transition
- Convert full tet meshes to multi-elements meshes



Remaining Challenges

There is still a lot of work to do in meshing technologies for aerospace application. But, some recent advances are promising

Unsteady mesh adaptation for turbulent flows (URANS - LES - DES - VMS)

- Dedicated error estimates
- Check efficiency of the fixed-point mesh adaptation algorithm

Curved mesh generation

- Mesh quality and validity to govern meshing algorithm
- Robust boundary layer meshing
- High-order error estimates
- Curved mesh adaptation

Thank you for your attention