On a unique operator for mesh generation and adaptation

Adrien Loseille

INRIA - Projet Gamma3 - Saclay, France

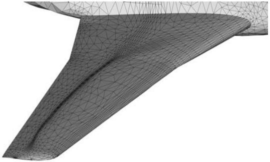
Tetrahedron Workshop
Liège, July 2016

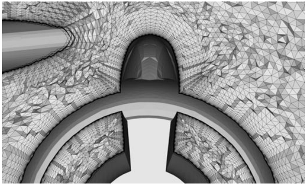
1. Motivations and context
2. **Topology** : unique cavity-based operator

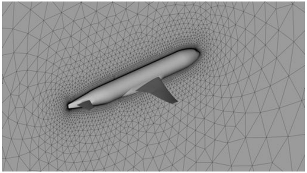
   Applications : Regular Adaptation, Parallel, CAD projection
3. **Geometry** : Generalized Advancing-Point

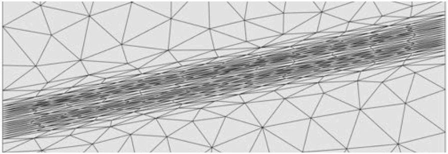
   Applications: Metric-aligned, Metric-orthogonal, ...

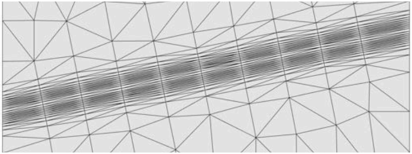
Scope of the talk











- How to generate frontal-like, adapted, aligned, meshes . . .
  within a unique framework ?
- **Topology** : We use a single mesh modification operator
- **Geometry** : We define a single advancing-point strategy

Context : metric-based framework

Fundamental concept: The mesh generator distance and volume computation are computed in a Riemmanian metric space

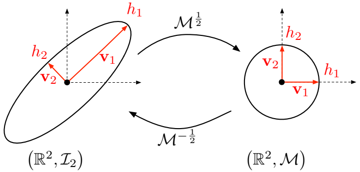
[George, Hecht and Vallet., Adv. Eng. Software 1991]

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

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

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

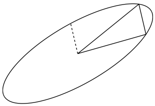
Context : metric-based framework

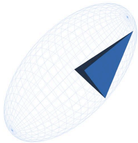
Fundamental concept: The mesh generator distance and volume computation are computed in a Riemmanian metric space

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

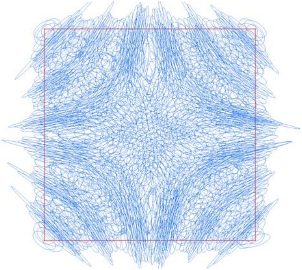
Scope of the meshing algorithm: Generate a **unit mesh** w.r.t $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

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

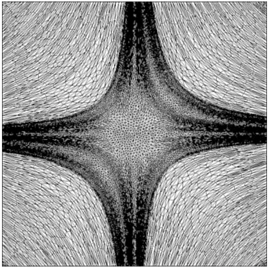




Inputs $(\mathcal{H}_0, \mathcal{M}_i)_{i\in\mathcal{H}}$



Output $\mathcal{H}$



Local Adaptive Remeshing

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 $Q_{\mathcal{M}}$

Local Adaptive Remeshing

Numerous 3D local adaptive remeshers have been developed:

- `EPIC` [Michal and Krakos, AIAA 2012]
- `Feflo.a` [Loseille and Lohner, AIAA 2010]
- `Forge3d` [Coupez, REEF 2000]
- `Fun3d-Refine1/2` [Jones et al., AIAA 2006], [Park and Carlson, AIAA 2010]
- `MAdLib` [Compere et al., IJNME 2010]
- `MeshAdap` [Li et al., IJNME 2005]
- `Mmg3d` [Dobrzynski and Frey, IMR 2008]
- `Mom3d` [Tam et al., CMAME 2000]
- `Pragmatic` [Rokos et al.,Springer 2013]
- `Tango` [Bottasso, IJNME 2004]
- `Libadaptivity` [Pain et al., CMAME 2001]
- ....

Outline

1. Motivations and context
2. **Topology:** Unique cavity-based operator
3. **Geometry** : Generalized Advancing-Point

Unique Cavity-Based Operator

Issues:

- 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

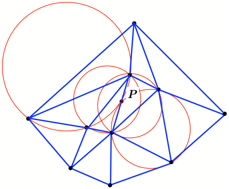
We propose a **unique cavity-based operator** inspired from the Delaunay method [Loseille and Löhner, AIAA 2010, Loseille et Menier, IMR 2013]

Delaunay Kernel

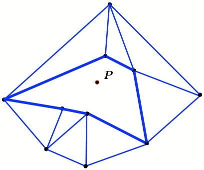
Insertion of P (incremental Delaunay context)

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

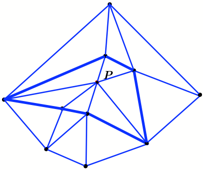
[Bowyer, CJ 1981], [Watson, CJ 1981], [Hermeline, RAIRO 1982], ...



$\mathcal{H}_k$



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



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

Validity principle

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

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

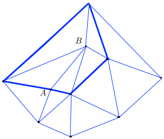
[George et al, ICSE 1990], [George et al, CMAME 1991], [George et al, IJNME 1992], ...

Unique Cavity-Based Operator [Loseille and Löhner, AIAA 2010], ...

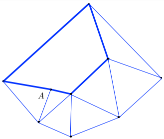
Unique meshing operator

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

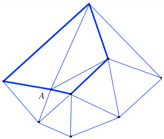
Collapse



$\mathcal{H}^k$

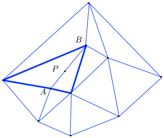


$\mathcal{H}^k - \mathcal{C}_{ball(B)}$

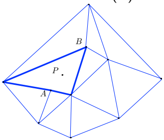


$\mathcal{H}^{k+1} = \mathcal{H}^k - \mathcal{C}_{ball(B)} + \mathcal{R}_A$

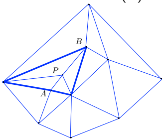
Insertion



$\mathcal{H}^k$

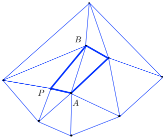


$\mathcal{H}^k - \mathcal{C}_{shell(A,B)}$

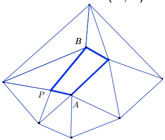


$\mathcal{H}^{k+1} = \mathcal{H}^k - \mathcal{C}_{shell(A,B)} + \mathcal{R}_P$

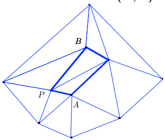
Swap



$\mathcal{H}^k$



$\mathcal{H}^k - \mathcal{C}_{shell(A,B)}$



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

Unique Cavity-Based Operator [Loseille and Löhner, AIAA 2010], ...

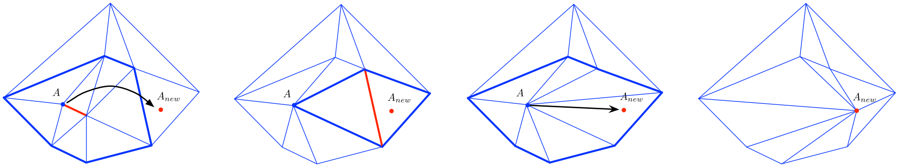
Unique meshing operator

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 Cavity-Based Operator [Loseille and Löhner, AIAA 2010], ...

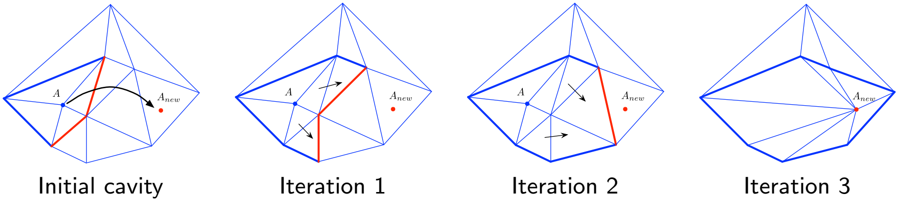
Unique meshing operator

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

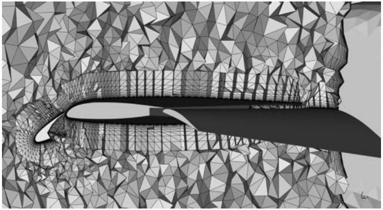
Unique Cavity-Based Operator [Loseille and Löhner, AIAA 2010], ...

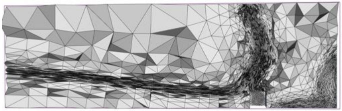
- Do either line or surface or volume
- Non manifold geometries
- Metric-based framework
- Easy to maintain
- Multiple operators for free (collapse + swap, many swaps, split + collapse, ...)
- Easy to set-up complex meshing operator
  Simple cavity mesh modification $\Leftrightarrow$ Complex multiple operators
  (generally too difficult to envision)
- Speeds intel Core i7 at 2.7Ghz Laptop
  Cavity-based collapse 20 000 points removed/sec
  Cavity-based insertion 20 000 points inserted/sec

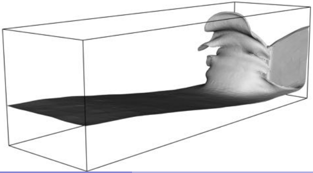
Overview of applications (in serial)

Anisotropic mesh generation [Loseille and Menier, IMR 2013], boundary layer and hybrid [Loseille and Löhner, IMR 2012], coarse-grained parallel [Loseille, Menier and Alauzet, IMR 2015], CAD projection with volume.

- Aerodynamics
- Unsteady flows
- Boundary-layer mesh generation
- Slug flows
- Free surface flows

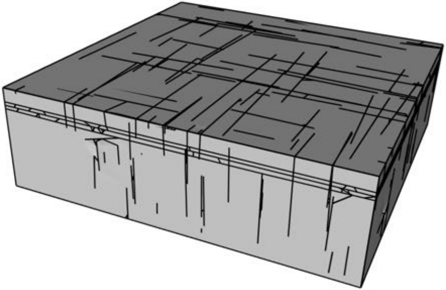


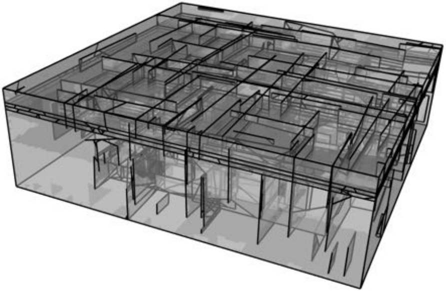




Non Manifold Case

- Fractured Reservoir engineering
- Highly complex non-manifold geometries
- Maximal edge connectivity of 6 - 1783 internal patches
- Geometry courtesy of Distene



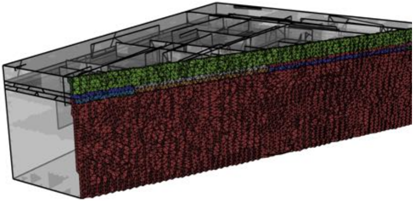


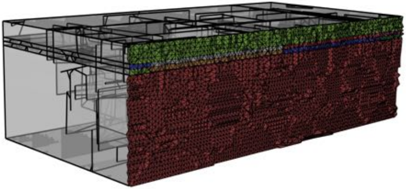
Once the cavity-inserter support non-manifold components:

$\Longrightarrow$ Works for unit-mesh phases, Optimization, Surface projection, . . .

Non Manifold Case

- Fractured Reservoir engineering
- Highly complex non-manifold geometries
- Maximal edge connectivity of 6 - 1783 internal patches
- Geometry courtesy of Distene





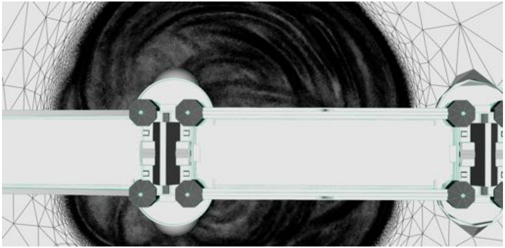
- Remeshing is conformed to the surface (definition)
- Conservation of internal boundary faces/ non manifold edges

verts : 425822 # tets: 243158

Parallel case: Blast on a London Tower Bridge

- High-resolution of the shock wave propagation
- Initial mesh : 3 837 269 vertices 477 852 tris and 22 782 603 tets
- Final mesh : 185 360 428 vertices 4 567 306 tris and **1 108 515 564** tets
- Parallel remeshing time is 14 min on 120 cores
  The total CPU time is 23 min (IOs, final gathering)

Level	% done	# of tets in interface	# of tets inserted	CPU time (sec.)	# of cores used
1	96%	44 490 595	1 051 662 221	415	120
2	100%	4 418 356	1 104 097 208	110	59
3	100%	10 084	1 108 515 564	36	1



⟹ Constrained cavity operators for coarse-grained parallelization

Outline

- Uniform speed for any operators
- Speed (cavity) versus quality (advancing-front)
- Combine both worlds : (try to) puts the point at the right place

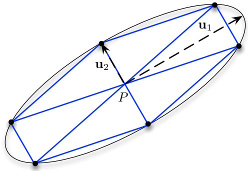
$\Longrightarrow$ Advancing point

Local Remeshing by Advancing Point

[Lohner and Onate, CNME 1998], [Remacle et Al., IJNME 2012], [Loseille, IMR 2014], [Marcum et Alauzet, IMR 2014]

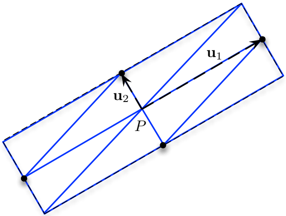
Init:

1. Initialize a list of points (front) $\{P_i\}_i$ from the boundary (line or surface)
2. From $\{P_i\}_i$,create a heap list of size/direction $(P_i, h_k, \mathbf{u}_k)_{ik}$.



Align edges along the shortest size eigen direction

6 (2D) and 24 (3D) points are proposed



Align edges along eigen directions

4 (2D) and 6 (3D) points are proposed

Local Remeshing by Advancing Point

[Lohner and Onate, CNME 1998], [Remacle et Al., IJNME 2012], [Loseille, IMR 2014], [Marcum et Alauzet, IMR 2014]

Init:

1. Initialize a list of points (front) $\{P_i\}_i$ from the boundary (line or surface)
2. From $\{P_i\}_i$,create a heap list of size/direction $(P_i, h_k, \mathbf{u}_k)_{ik}$.

Advancing-point: [Create all the vertices to be inserted]

1. Pop the first heap list entry, creates $P_{new} = P_i \pm h_k\, \mathbf{u}_k$
2. Update length/position according of P_{new} to Riemannian metric field.
2. Metric-based length filtering, add P_{new} for insertion
3. Update the heap list with $(P_{new}, h_k, \mathbf{u}_k)_k$
4. If the heap list is not empty goto 2

Local Remeshing by Advancing Point

[Lohner and Onate, CNME 1998], [Remacle et Al., IJNME 2012], [Loseille, IMR 2014], [Marcum et Alauzet, IMR 2014]

Init:

1. Initialize a list of points (front) $\{P_i\}_i$ from the boundary (line or surface)
2. From $\{P_i\}_i$,create a heap list of size/direction $(P_i, h_k, \mathbf{u}_k)_{ik}$.

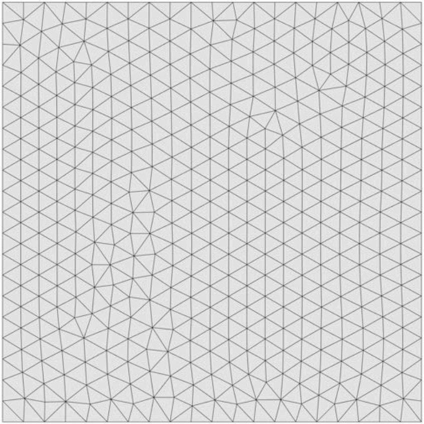
Advancing-point: [Create all the vertices to be inserted]

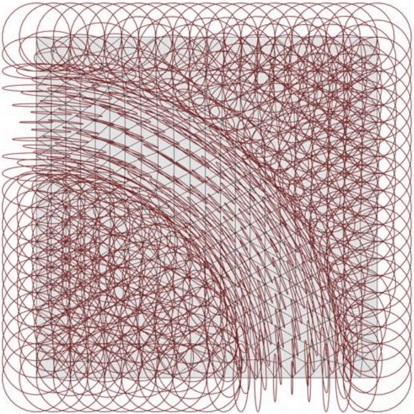
1. Pop the first heap list entry, creates $P_{new} = P_i \pm h_k\, \mathbf{u}_k$
2. Update length/position according of P_{new} to Riemannian metric field.
2. Metric-based length filtering, add P_{new} for insertion
3. Update the heap list with $(P_{new}, h_k, \mathbf{u}_k)_k$
4. If the heap list is not empty goto 2

Insertion:

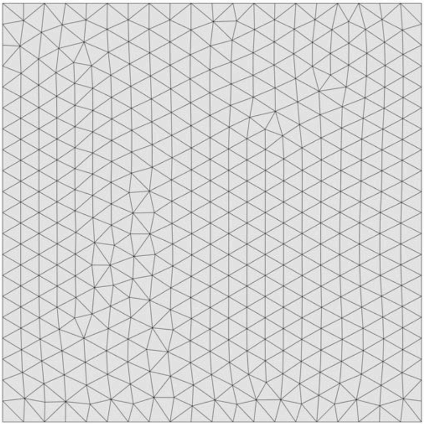
1 Use cavity-based operators

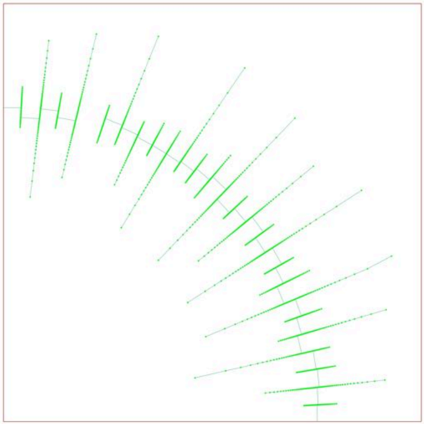
Local Remeshing by Advancing Point





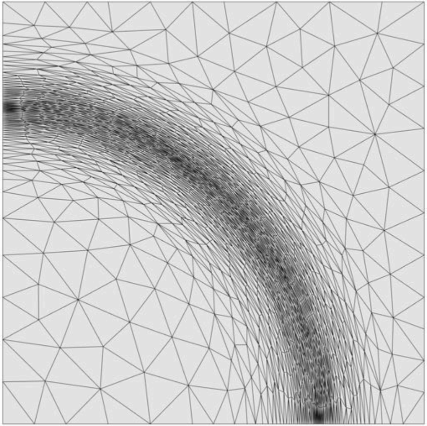
Local Remeshing by Advancing Point



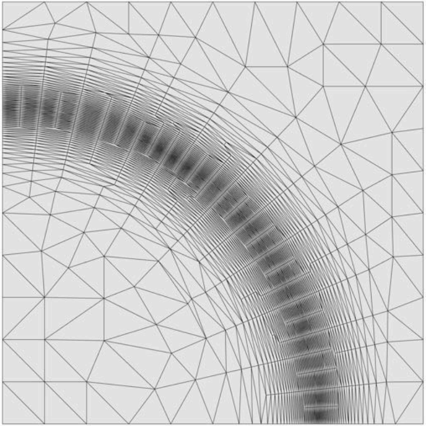


- Sorting by minimal sizes $\Longrightarrow$ avoid transition in highly anisotropic areas

Local Remeshing by Advancing Point



Standard



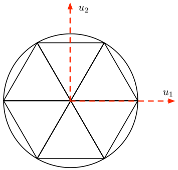
Advancing-point

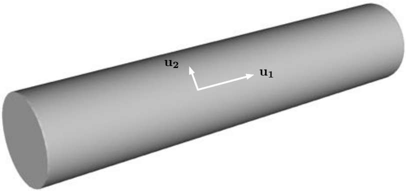
- Reduced large angles
- Local alignement/orthogonality

Unicity of advancing-point scheme

For isotropic metric, we do not have privileged directions

- Use the local surface features (principal direction)

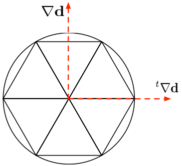


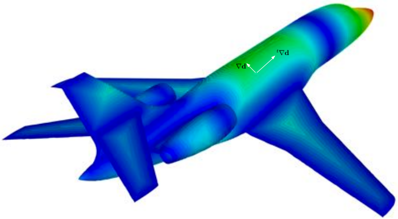


Unicity of advancing-point scheme

For isotropic metric, we do not have privileged directions

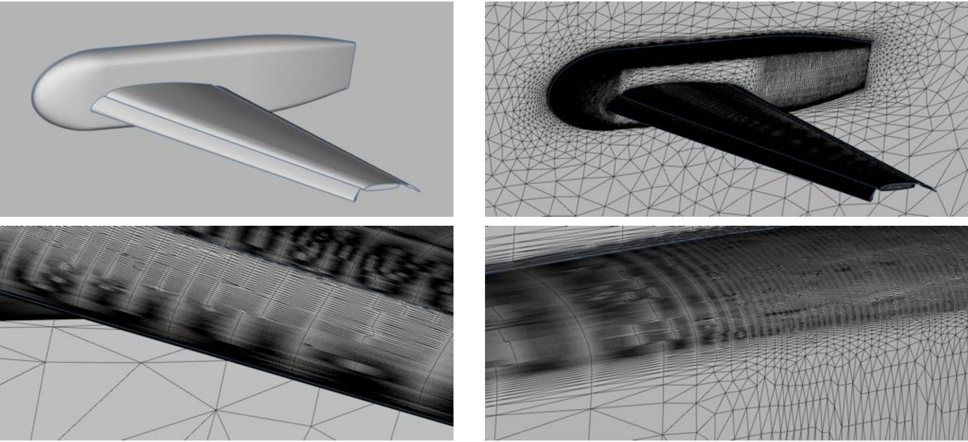
- Use the local surface features (principal direction)
- Use the distance to the body





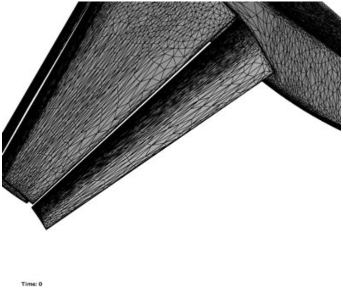
Anisotropy and Alignment: Surface Remeshing

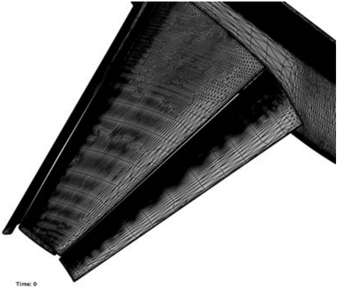
- High Lift geometry : 194 810 verts 389 616 tris
- Highly anisotropic surface meshes $O(1 - 10000)$
- Surface mesh is far from best-practice grids



Anisotropy and Alignment: Surface Remeshing

- High Lift geometry : 194 810 verts 389 616 tris
- Highly anisotropic surface meshes $O(1 - 10000)$
- Surface mesh is far from best-practice grids





- Metric-alignment and orthogonality produces high quality meshes
- Alignement reduces the error of the surface approximation

Anisotropy and Alignment: RANS adaptation

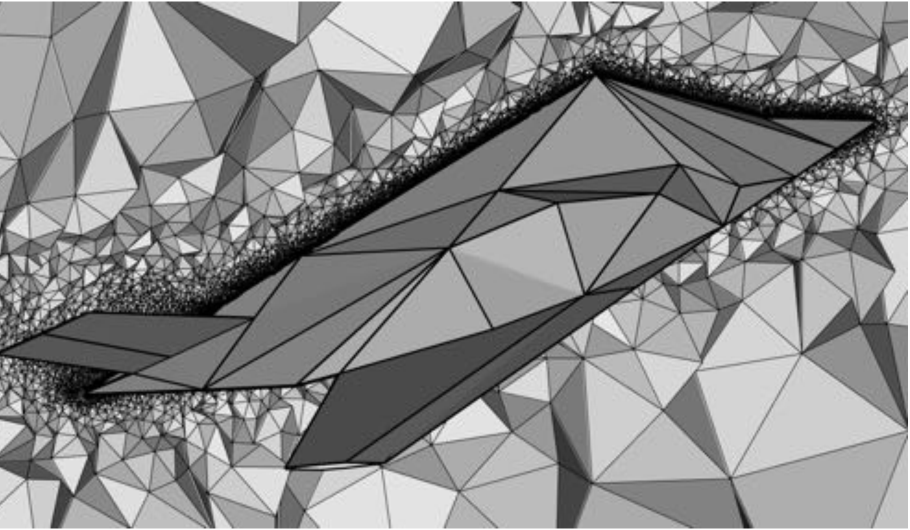
- F117 geometry
- RANS-SA at Mach 0.8, Reynolds $= 5 \times 10^6$

- 5 759 136 vertices, 339 530 triangles and 33 967 205 tetrahedra
- 30 Layers, grow rate of 1.3, $y^+ = 0.03$
- Uniform surface mesh

- Multi-scale multi-field metric: density, velocity, pressure in $\mathbf{L}^2$ *norm*
- Metric-aligned approach on surface and volume
- 15 iterations, with 5 iterations at fixed complexity

$$[750\,000, 1\,500\,000, 3\,000\,000]$$

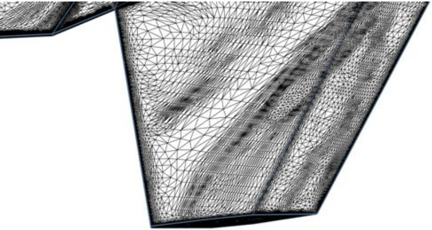
Anisotropy and Alignment: RANS adaptation

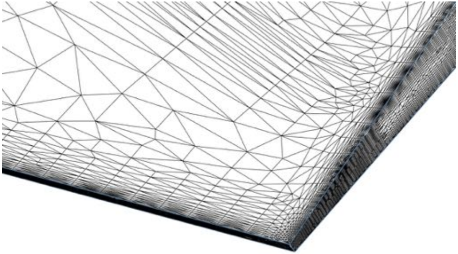


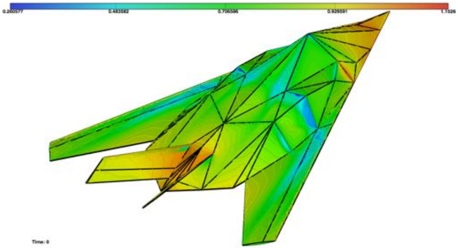
Initial mesh

Anisotropy and Alignment: RANS adaptation

- Automatic quasi-structured meshes of leading/trailing edges

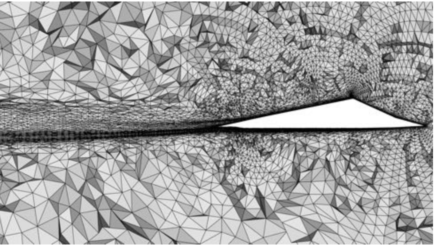


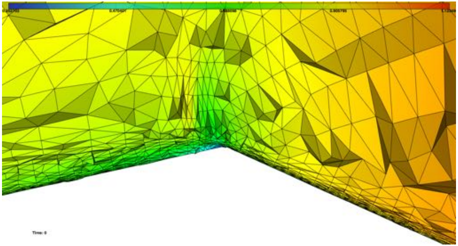


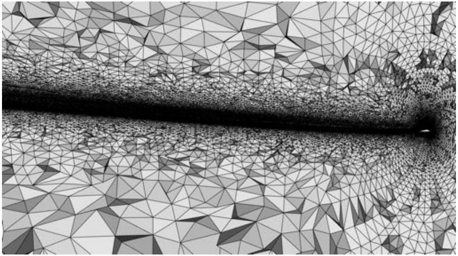


- 277 829 vertices
- 555 650 triangles

Anisotropy and Alignment: RANS adaptation

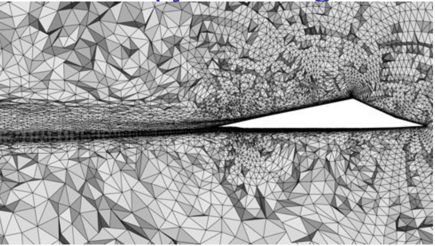


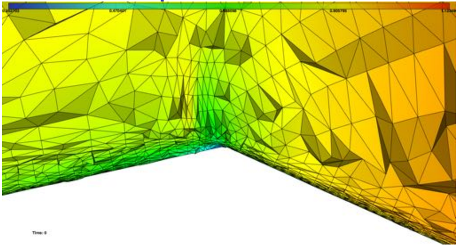


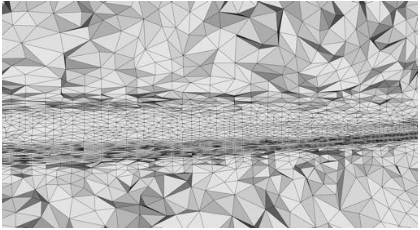


- 4 052 725 vertices
- 555 650 triangles
- 23 816 402 tetrahedra

Anisotropy and Alignment: RANS adaptation



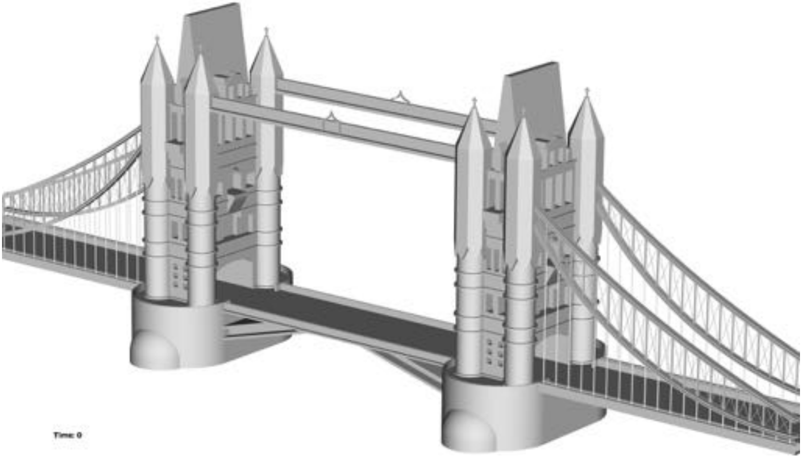




- 4 052 725 vertices
- 555 650 triangles
- 23 816 402 tetrahedra

Quasi-structured elements are generated when the metric features orthogonality components

Anisotropy and Alignment: Blast example

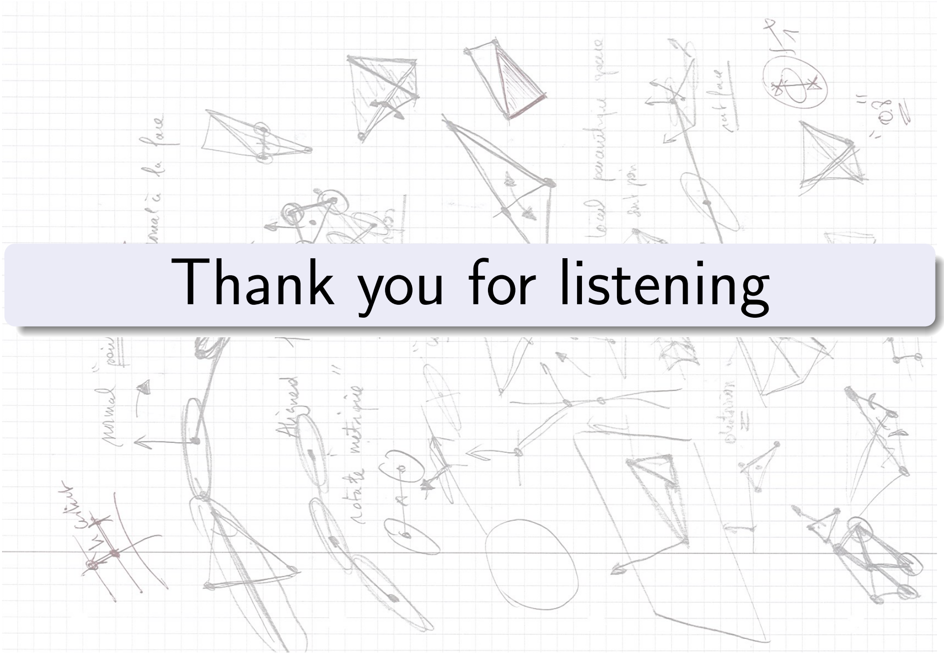


Conclusion

- Unique cavity-based operator
  - Cavity initialization defines the initial operation
  - Multiple operators defined naturally
  - Ease of maintenance and robustness
  - Highly flexible remeshing process
- Metric-orthogonal, Metric-aligned
  - Favor alignment and the creation of quasi-structured elements
  - Work in 3D, for non manifold geometries

On going work

- Implement the very good ideas we had yesterday night
- Wait tonight for maybe even better ideas



Thank you for listening