Dynamic mesh adaptation for turbulent flows around immersed geometries

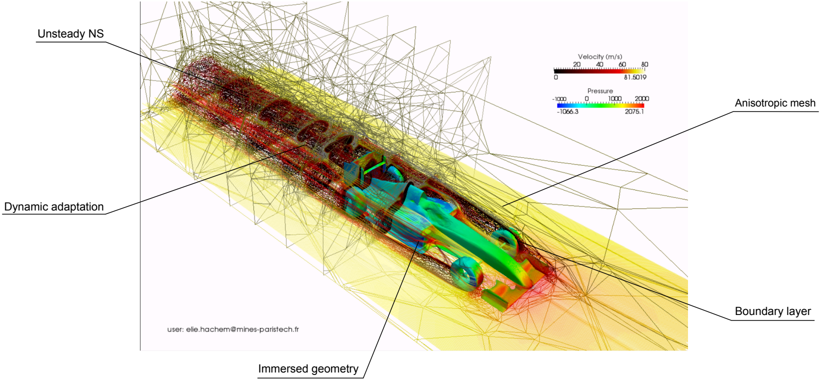
Laure BILLON, Youssef MESRI, Elie HACHEM

CFL: Computing & FLuids Research Team
Mines-ParisTech, CEMEF, Sophia Antipolis, France

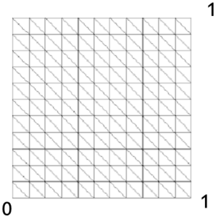
laure.billon@mines-paristech.fr

13 April 2016
ANR MAIDESC

Numerical simulation of turbulent flows is one of the great challenges in Computational Fluid Dynamics (CFD)

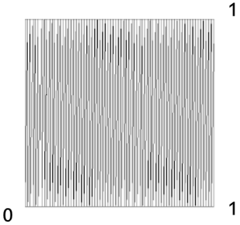


What's a metric ?



$$+ \quad M = \begin{pmatrix} \dfrac{1}{0.01^2} & \\ & \dfrac{1}{1^2} \end{pmatrix} \quad =$$

Metric (directions, size)



Dimensionless wall distance for the first cell :

$$y_0^+ = h_{min} \frac{u_\tau}{\nu}$$

with u_τ the friction velocity, and τ_ω the wall shear stress :

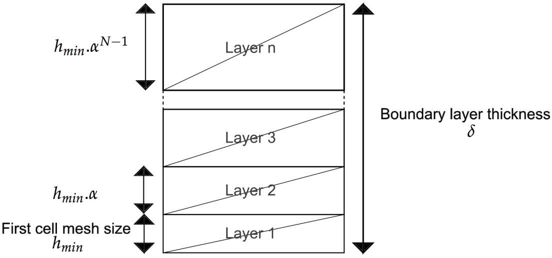
$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \qquad \tau_\omega = C_F \frac{\rho U_\infty^2}{2}$$

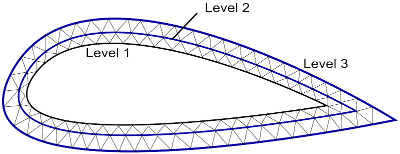
Minimum wall mesh size :

$$h_{min} = \frac{y_0^+}{\frac{Re}{L}\sqrt{\frac{C_F}{2}}}$$

Proposition :

- Boundary Layer Metric (y_0^+, Re, L)
- Fitting the geometry curvature
- Fixed mesh suitable for the whole simulation





Let's define $N = -\frac{\nabla\phi(P)}{|\nabla\phi(P)|}$

$$\mathcal{S} = -\nabla_T N = (I - N.N^T)Hes(\phi)(P)$$

Eigenvalues = Curvatures

$$\kappa_i = tr(S) \pm \sqrt{\left(\frac{tr(S)}{2}\right)^2 - Z(S)}$$

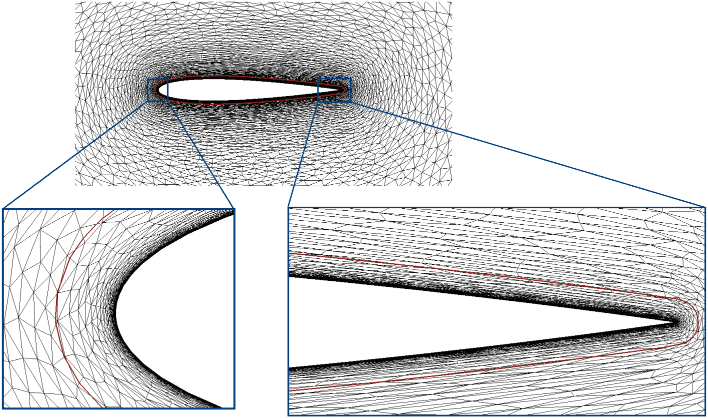
$$h_{ti} = f(\kappa_i, h_{min})$$

Associated eigenvectors = Directions

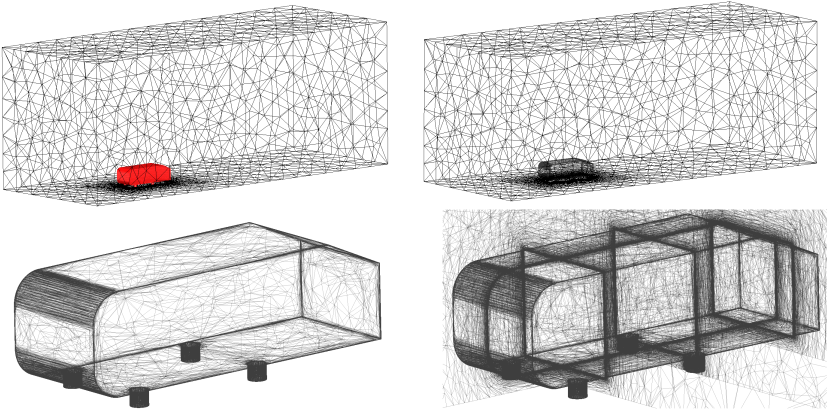
$$\mathcal{M} = \begin{pmatrix} N \\ T_1 \\ T_2 \end{pmatrix} \begin{pmatrix} \frac{1}{h^2} & 0 & 0 \\ 0 & \frac{1}{{h_{t_1}}^2} & 0 \\ 0 & 0 & \frac{1}{{h_{t_2}}^2} \end{pmatrix} \begin{pmatrix} N & T_1 & T_2 \end{pmatrix}$$

[Billon et al., 2016]

Curvature

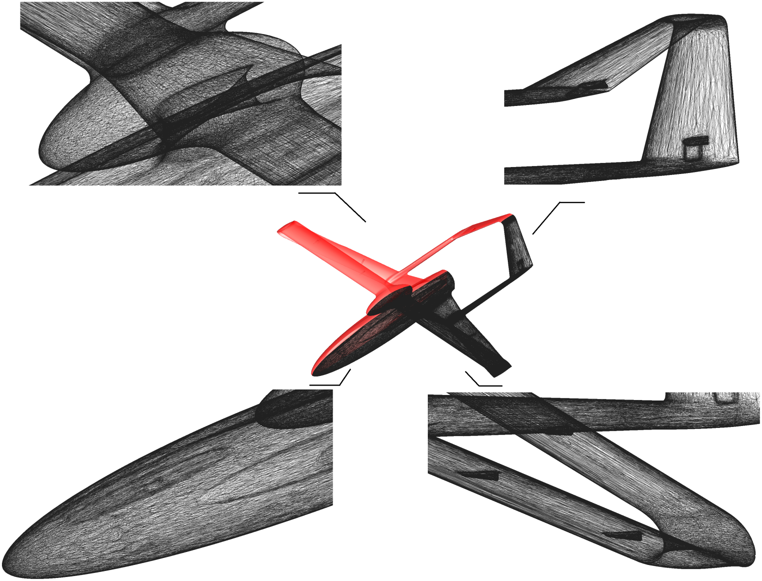


Simple 3D geometry : Ahmed body



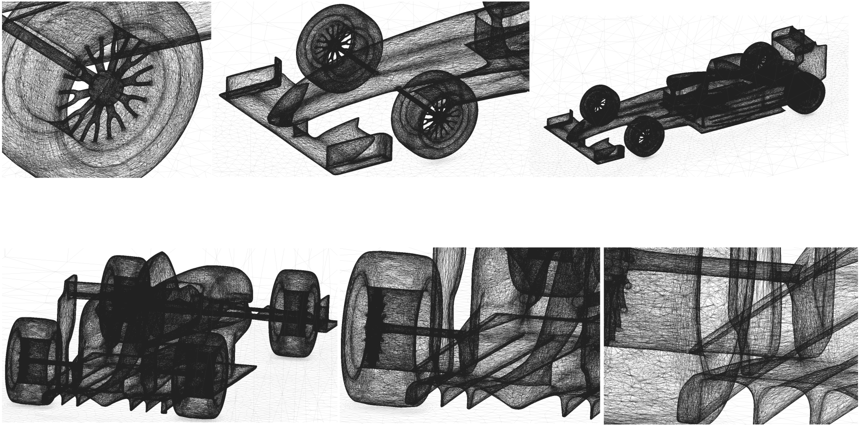
2167057 elements - 378259 nodes - 20 cores

Complex geometry : Tyrix drone



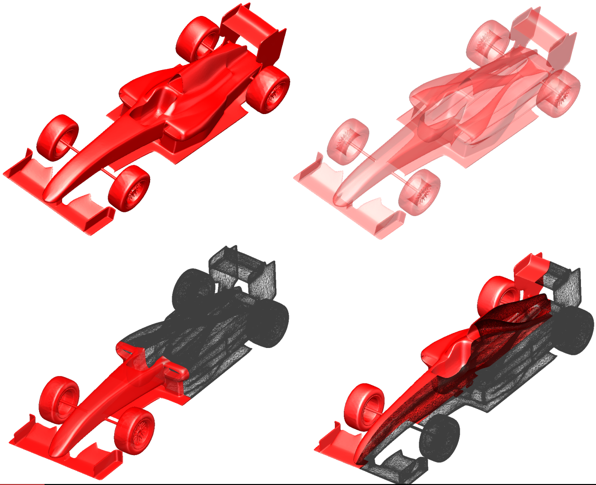
8049721 elements - 1378131 nodes - 20 cores

To go further : Formula 1

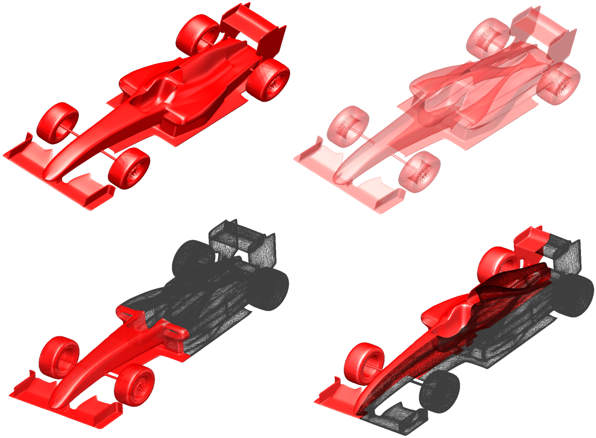


21026520 elements - 3602483 nodes - 40 cores

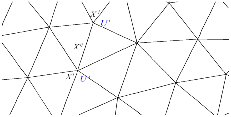
To go further : Formula 1



To go further : Formula 1



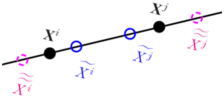
Edge based metric in details



$$\mathbf{X}^{ij} = \mathbf{X}^j - \mathbf{X}^i$$

Lengths distribution tensor :

$$\mathbb{X}^i = \frac{d}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}$$



$$\widetilde{\mathbf{X}^{ij}} = s_{ij} \mathbf{X}^{ij}$$

Drive these stretching factors using an edge based error estimator :

$$s_{ij} = \left(\frac{\varepsilon}{e_{ij}} \right)^{\frac{1}{p}}, \; \varepsilon = \left(\frac{\sum_i \sum_{j \in \Gamma_i} e_{ij}^{\frac{p}{p+2}}}{N_a} \right)$$

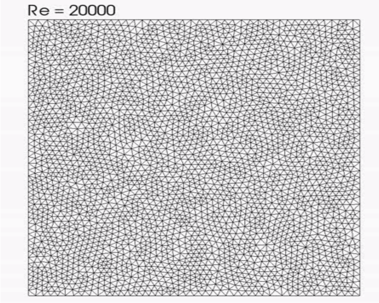
[Coupez, 2011] [Coupez and Hachem, 2013]

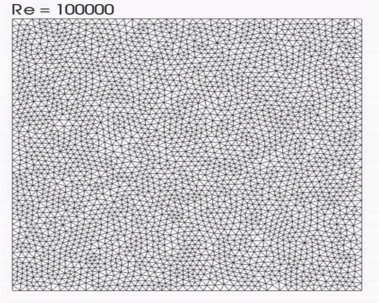
Edge based metric :

$$\widetilde{\mathcal{M}^i} = \frac{|\Gamma(i)|}{d} \left(\widetilde{\mathbb{X}^i} \right)^{-1}$$

Multi-criterias :

$$u(X^i) = \left\{ \frac{V^i}{|V^i|}, \frac{|V^i|}{\max_j |V^j|} \right\}$$



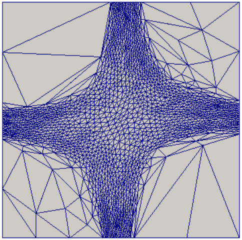


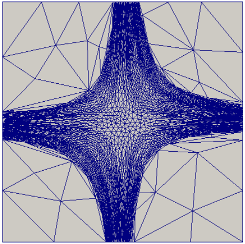
Stretching factors computation :

$$s_{ij} = \left(\frac{\varepsilon}{e_{ij}}\right)^{1/2}$$

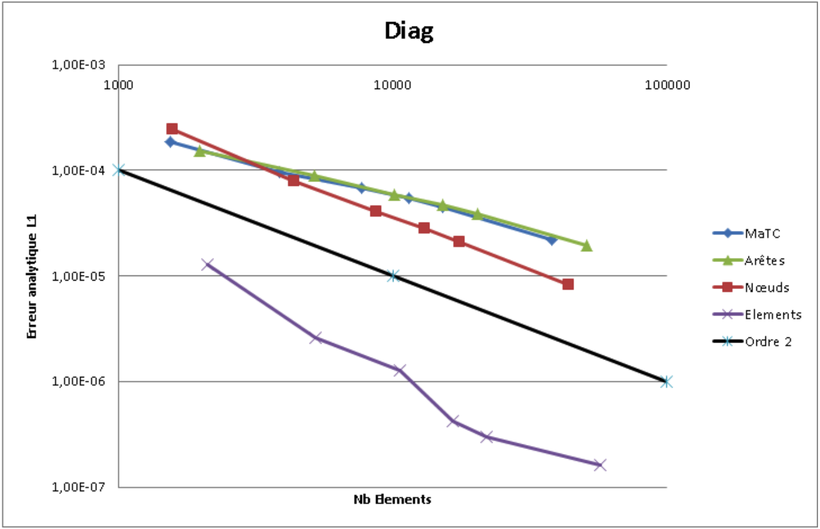
$$\varepsilon = \left(\frac{\frac{1}{C_0}\sum_T \int \sqrt{det(\mathcal{M}_T)}}{N_T}\right)^{2/d}$$

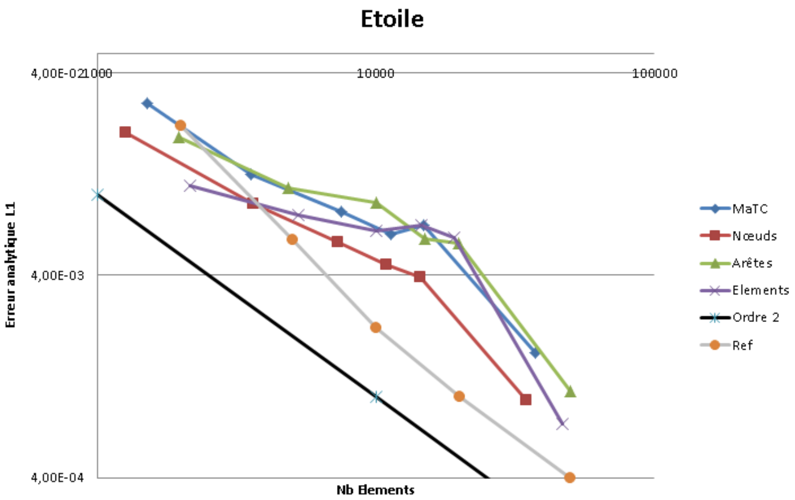
User control on the **minimum and maximum mesh sizes**



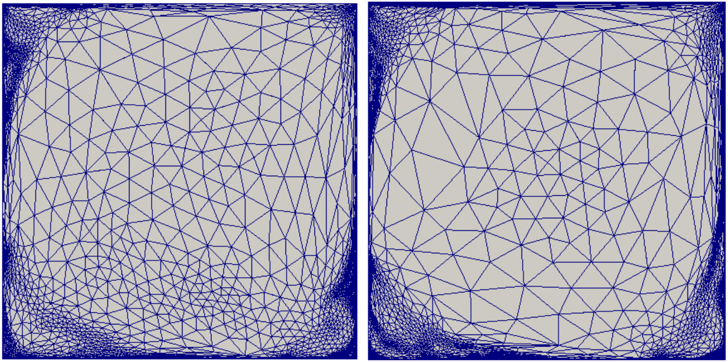


Study of analytical error :




Etoile
4,00E-02
1000
10000
100000
4,00E-03
4,00E-04
Erreur analytique L1
Nb Elements
MaTC
Nœuds
Arêtes
Elements
Ordre 2
Ref

Driven cavity comparison



Proposition :

- Method 1 : Replacing the metric inside the boundary layer zone by the Boundary layer metric
- Method 2 : Intersecting carefully both metrics inside the boundary layer region

Conclusion :

- Boundary Layer Metric with curvature
- New error estimator based on elements
- Coupling metrics

Perspectives :

- Backward facing step
- New geometries
- Moving geometry

[Khalloufi et al., 2016]

Coupez, T. (2011).
Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing.
Journal of Computational Physics, 230(7) :2391–2405.

Coupez, T. and Hachem, E. (2013).
Solution of High-Reynolds Incompressible Flow with Stabilized Finite Element and Adaptive Anisotropic Meshing.
Computer Methods in Applied Mechanics and Engineering, 267 :65–85.