Impact of anisotropic mesh adaptation for turbulent flows in aeronautics Towards the certification of numerical solutions ?

Frédéric Alauzet

Projet GammaO - Inria-Onera team - INRIA Saclay - France

Tetrahedron Workshop VII - Barcelona - 2023

1. Motivations
2. Mesh-Adaptive Solution Platform
3. Numerical Results
4. Conclusions

Acknowledgments ...

... to all the people who contribute or exchange on this work

My colleagues at Inria:

F. Clerici, L. Frazza,
P.L. George, A. Loseille,
M. Maunoury, L. Marechal,
C. Tarsia Morisco,
L.-M. Tenkes, J. Vanharen

My colleagues at Onera:

G. Puigt

My colleagues at MSU:

D. Marcum

My colleagues at Safran Tech:

L. Billon, D. Papadogiannis,
E. Parente, M. Philit,
A. Remigi

My colleagues at Boeing:

T. Michal, P. Spalart

My colleagues at Lemma:

O. Allain, A. Dervieux, D. Guegan

Thanks to the sponsors who believe in this work
Safran Tech, Boeing, Ariane Group, Lemma

Scientific Motivations

Errors in the numerical simulation can be catastrophic !!!

Roissy Terminal 2E
roof collapse

[Feghaly, SC 2008]

Sleipner-A offshore
platform sank
in the North sea

[Jakobsen, SEI 1994], [Collins, CI 1997]

Falcon 5X
production stops
due to Silvercrest issues

Scientific Motivations

Flow solver (and model) verification and validation:

- Verification: yields the right results ("it solves the equation right")

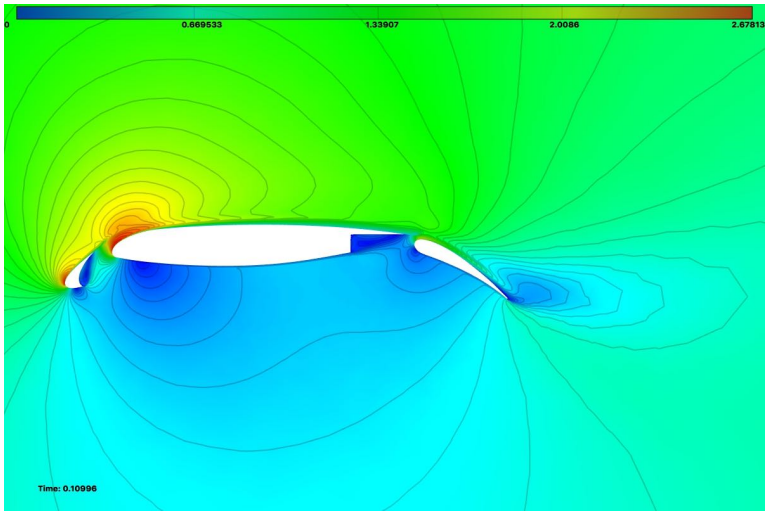
method of manufactured solutions (analytical solution, order of convergence), comparison with other flow solvers (NASA Turbulence Modeling Resource)

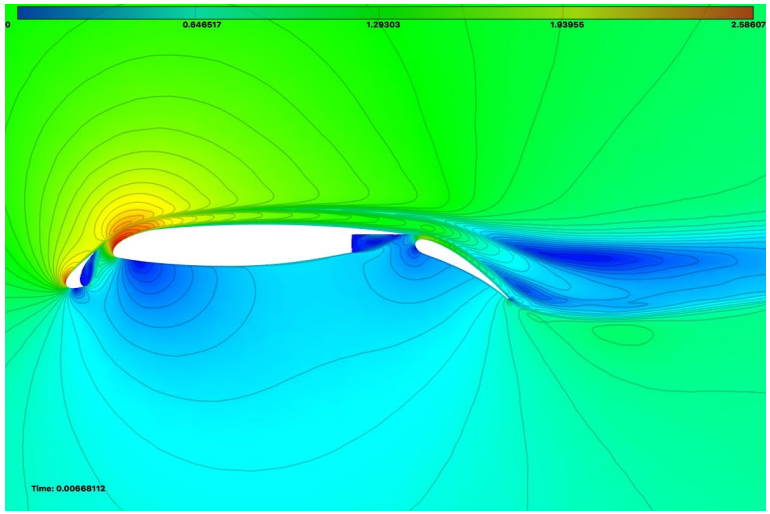
- Validation: models with accuracy real world problem ("it solves the right equations")

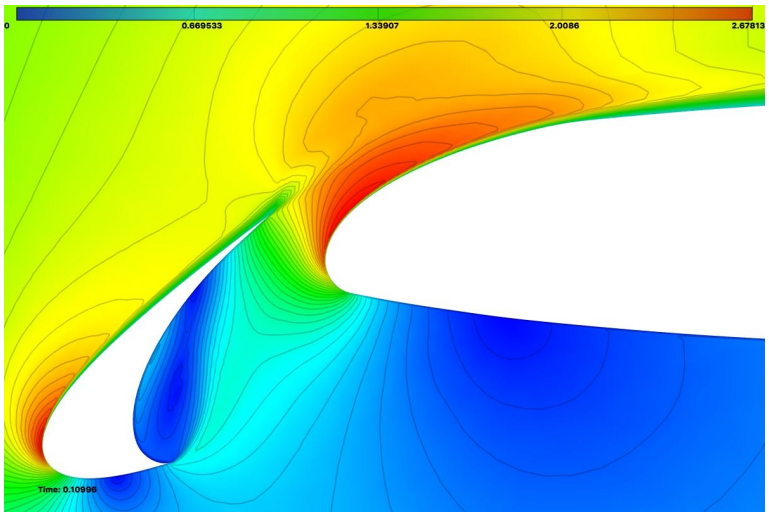
comparison to experimental measurements, AIAA CFD Prediction Workshops

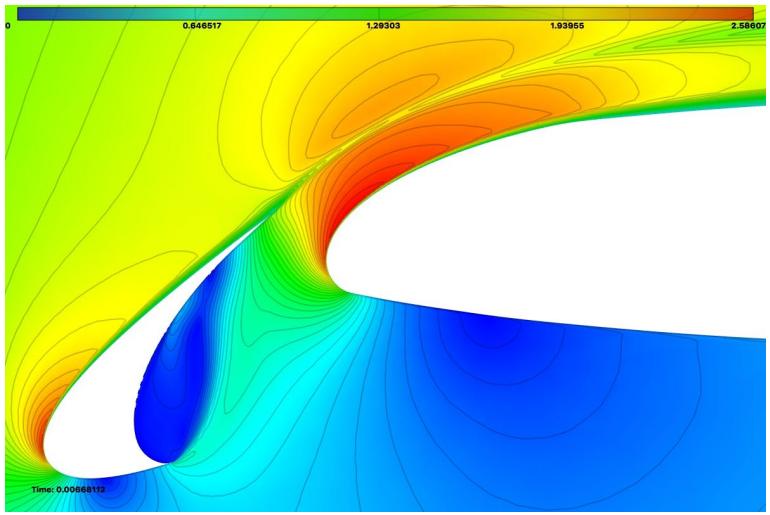
But, we have no guarantee that numerical solutions are correct for any industrial simulations

Is this the same flow ?

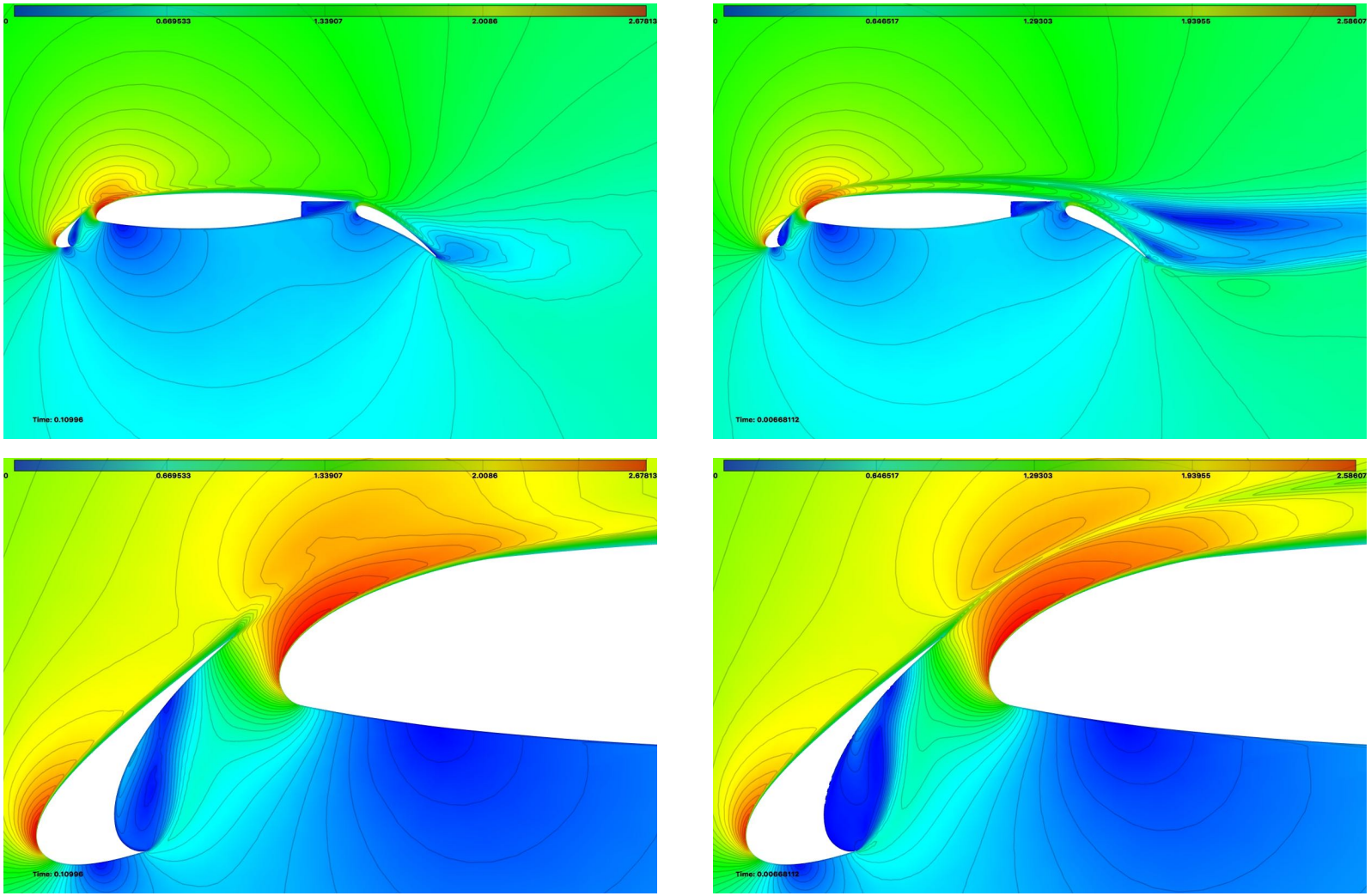






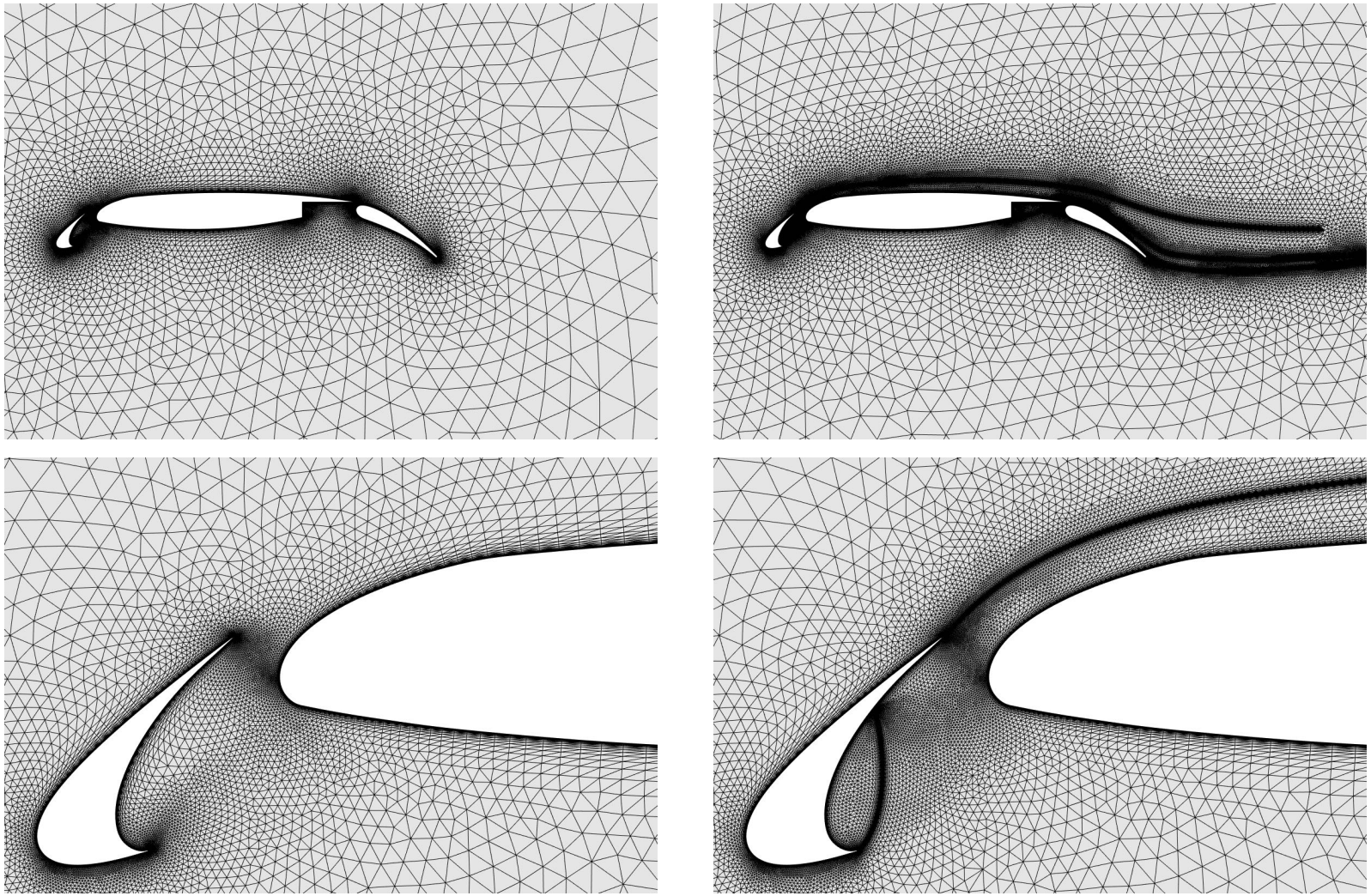


Is this the same flow ?



The answer is YES: $M = 0.175, \quad \alpha = 16.21, \quad Re = 15.1 \times 10^6$

Is this the same mesh ?



The answer is NO: manually adapted using `AFLR` by inserting seeding layers knowing the flow field

Scientific Motivations

Flow solver (and model) verification and validation:

- Verification: yields the right results ("it solves the equation right")

method of manufactured solutions (analytical solution, order of convergence), comparison with other flow solvers (NASA Turbulence Modeling Resource)

- Validation: models with accuracy real world problem ("it solves the right equations")

comparison to experimental measurements, AIAA CFD Prediction Workshops

But, we have no guarantee that numerical solutions are correct for any industrial simulations

Geometry and numerical solution certification:

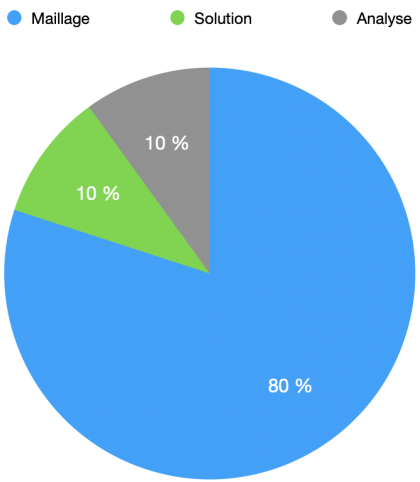
- Certification: geometric or solution discretization error control, numerical solution UQ, mesh-converged solution

$\Longrightarrow$ *This is the purpose of error estimate, mesh adaptation and correctors*

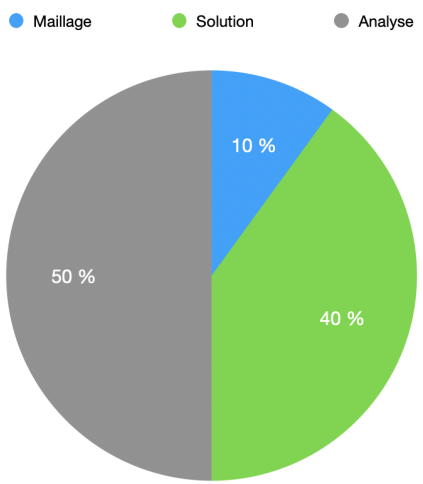
Industrial Motivations

Mesh generation represents **a very high cost** in human resources
Several men weeks each time

- Meshing generally remains a manual process
- Several mesh-computation iterations are necessary to obtain a "good" mesh
- The geometries and physical phenomena considered are increasingly complex



Best-practice mesh paradigm



Mesh-adaptive paradigm

Engineers are paid for analysis and design **not for meshing**

Why ?

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

Typical High-Resolution Viscous Application

High-resolution of viscous flows for complex geometries involves

- Large variation in length scales,
- Thick BL regions,
- Shear layers,
- Detached vortices, ...
- Complex geometric configuration flap, slat, engine, ...
- Small gaps,
- Small details often crucial to predicting key flow physics,

What we propose

A new paradigm for numerical simulations

- Mesh and solutions are computed at the same time, they are strongly coupled. Even more, **mesh and solution are indissociable**

- Different simulations lead to different meshes

It requires an automated process and flexibility in the meshing process
$\Longrightarrow$ Full tetrahedra meshes

A numerical simulation platform with **mesh adaptation**

- Control of the numerical errors (error estimates)
- Quantification of the numerical errors (non-linear correctors)
- Certification of numerical solutions (mesh convergence)
- High-fidelity numerical solutions (high-order, massively parallel)

What we propose

A new paradigm for numerical simulations

- Mesh and solutions are computed at the same time, they are strongly coupled. Even more, **mesh and solution are indissociable**

- Different simulations lead to different meshes

It requires an automated process and flexibility in the meshing process
$\Longrightarrow$ Full tetrahedra meshes

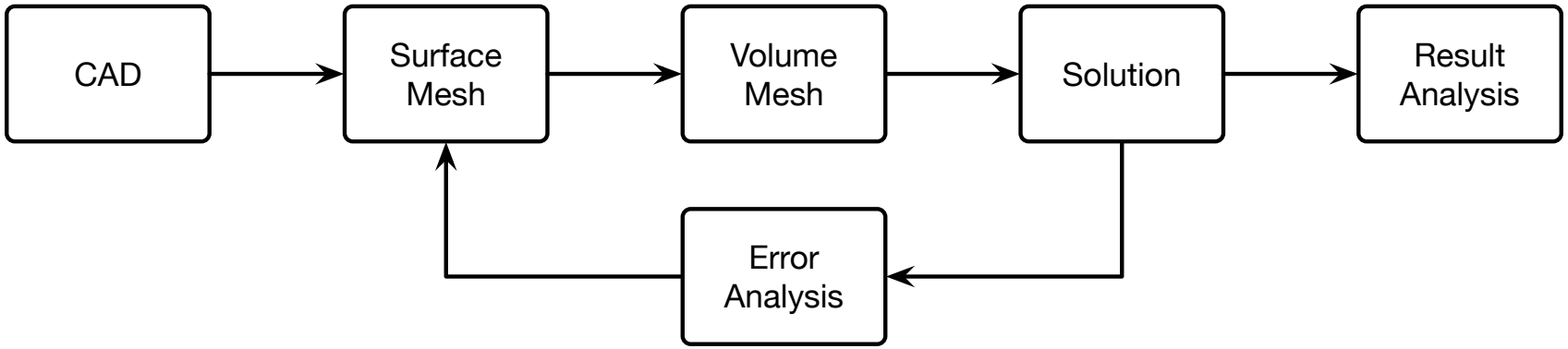
This new paradigm is dedicated to problems involving:
- complex geometries
- physics with singularities (mainly anisotropic physics)

It provides new insights on turbulence models

Outline

Mesh Adaptation Loop

Mesh-Adaptive Numerical Simulation Pipeline:



No step of the process can be neglected

All of them are source of improvements

All of them need to handle extreme anisotropy ($1 : 10^6$)

This will govern our methodological choices

Today, I will focus on

- Flow solver
- Metric-based error estimate

Outline

The Flow Solver: `Wolf`

Compressible Turbulent Navier-Stokes Equations:

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = \mathcal{S}(W) + \mathcal{Q}(W)$$

Turbulence model: Spalart-Allmaras

Mixed-Element-Volume Method (MEV): $\Leftarrow$ **flexibility and robustness**

- Convective and source terms by Finite Volume method
- Diffusive terms by Finite Element method

Vertex-centered: low complexity, perfectly fitted for **adapted meshes with extreme anisotropy**

- Median cells

Solve on each FV cell: $|C_i| \frac{dW_i}{dt} + \mathbf{F}_i = \mathbf{S}_i + \mathbf{Q}_i$

Wolf : Spatial Discretization

Edge-based formulation with upwind element for convective terms: 1D splitting

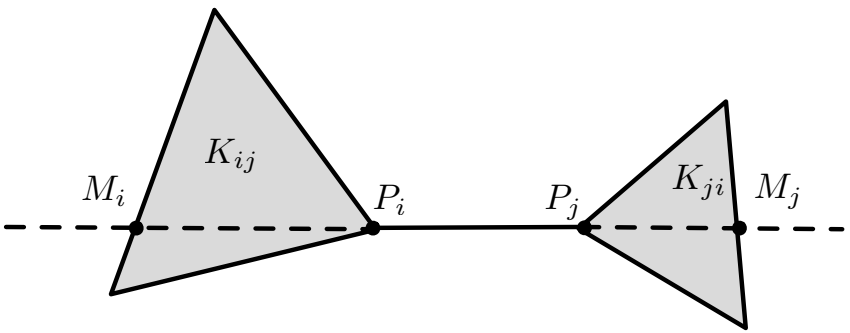
- Cells are implicitly defined via the edges and not stored

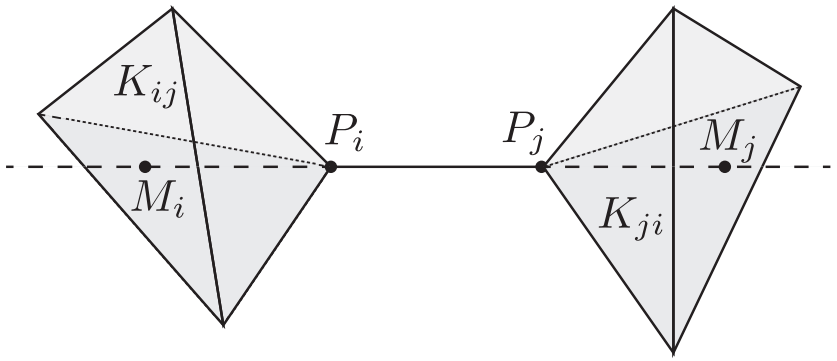
Convective flux: $\mathbf{F}_i = \sum_{j \in \mathcal{V}(i)} F_{|\partial C_{ij}} \cdot \int_{\partial C_{ij}} \mathbf{n}_i \mathrm{d}\gamma = \sum_{j \in \mathcal{V}(i)} \Phi_{ij}(W_i, W_j, \mathbf{n}_{ij})$

- Φ_{ij} with the HLLC Riemann solver $\Leftarrow$ **the most robust and accurate**

2^{nd} order scheme using a MUSCL-like reconstruction at the cell interface:

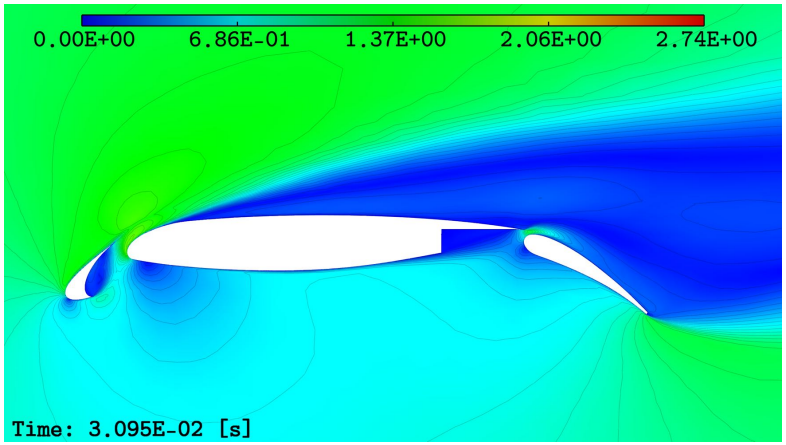
- Directional MUSCL using upwind and downwind elements
- **Low dissipation reconstruction**: V4-scheme (V6-scheme also exists)
  Combination of centered and upwind-downwind gradients
  Specific limiters: Piperno or **Gamma**



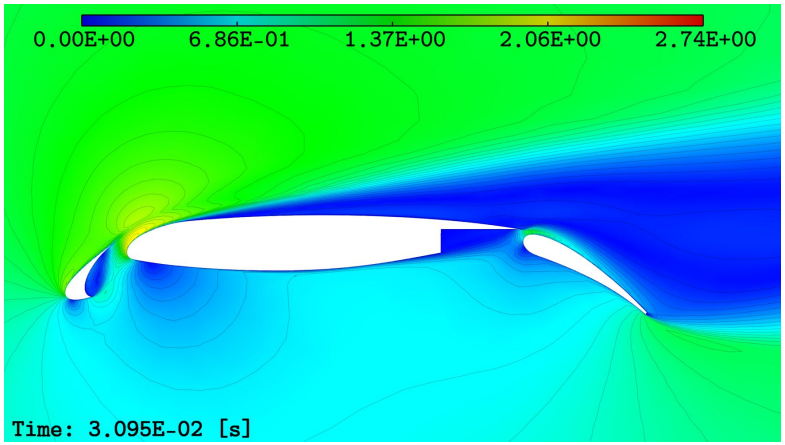


Low dissipation schemes with appropriate limiters are advantageous

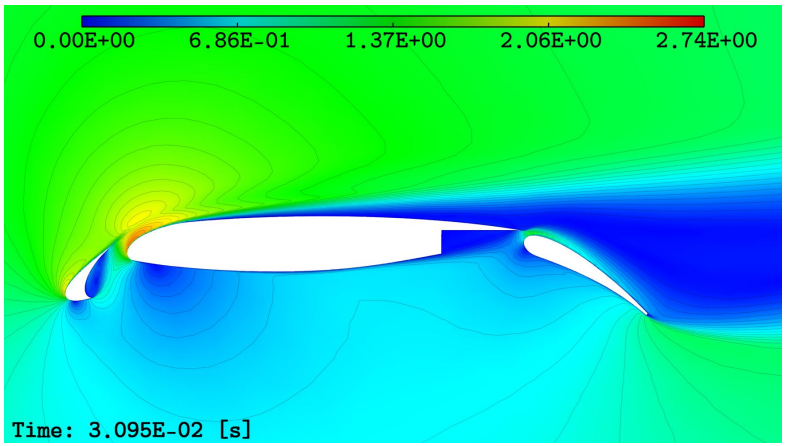
Test on an adapted mesh composed of 12 229 vertices and 22 048 triangles



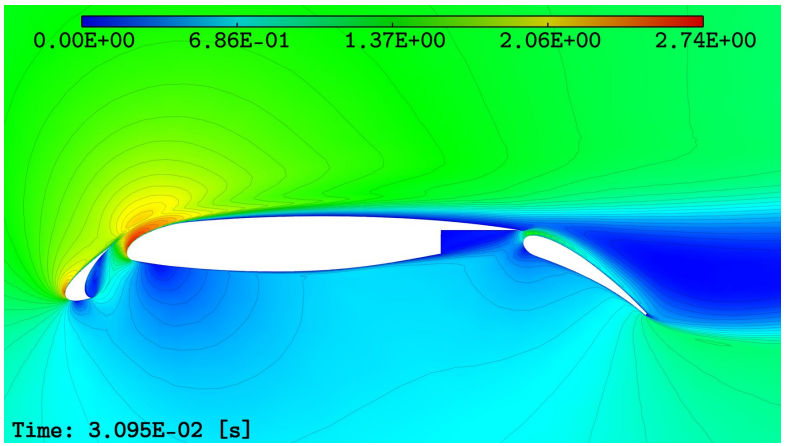
V3-scheme - MinMod limiter



V3-scheme - VanAlbada limiter



V4-scheme - Piperno limiter



V4-scheme - New limiter

Element-based classical FEM formulation for viscous terms

$$\int_{\partial C_i} \mathbf{S}(W_i) \cdot \mathbf{n} \, \mathrm{d}\gamma \;=\; \sum_{P_j \in \mathcal{V}(P_i)} \int_{\partial C_{ij}} \mathbf{S}(W_i) \cdot \mathbf{n} \, \mathrm{d}\gamma \;=\; -\sum_{K \ni P_i} \int_K \mathbf{S}(W_i)|_K \cdot \nabla \phi_i \, \mathrm{d}\mathbf{x} \, .$$

Turbulence model discretization:

- Convective term: non-linear convection (positive, maximum principle, ...)
- Diffusive and dissipative terms: FEM formulation
- Production and destruction terms: FV source term formulation

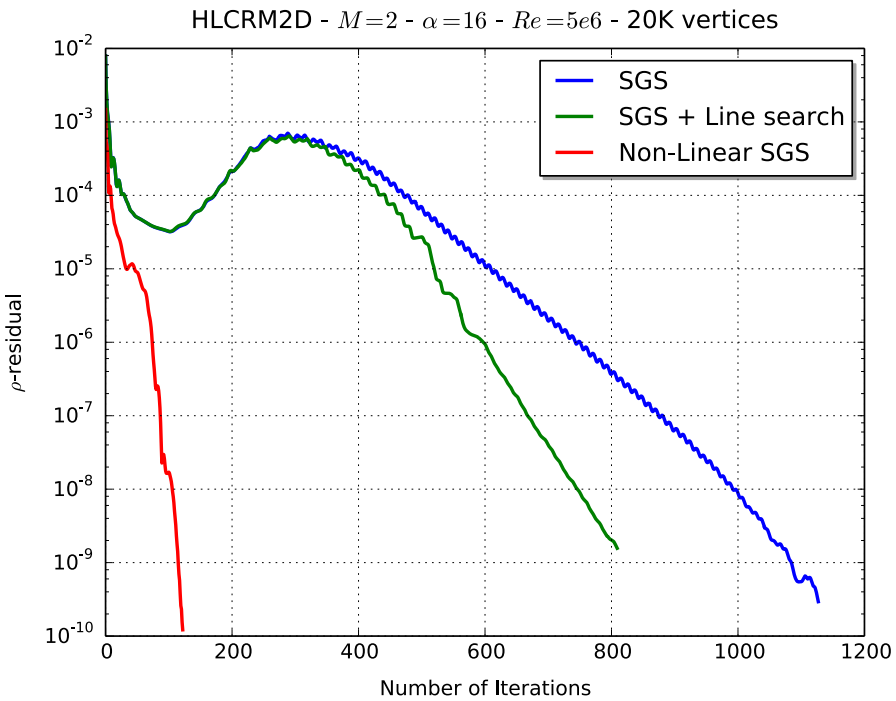
Wolf : Temporal Discretization

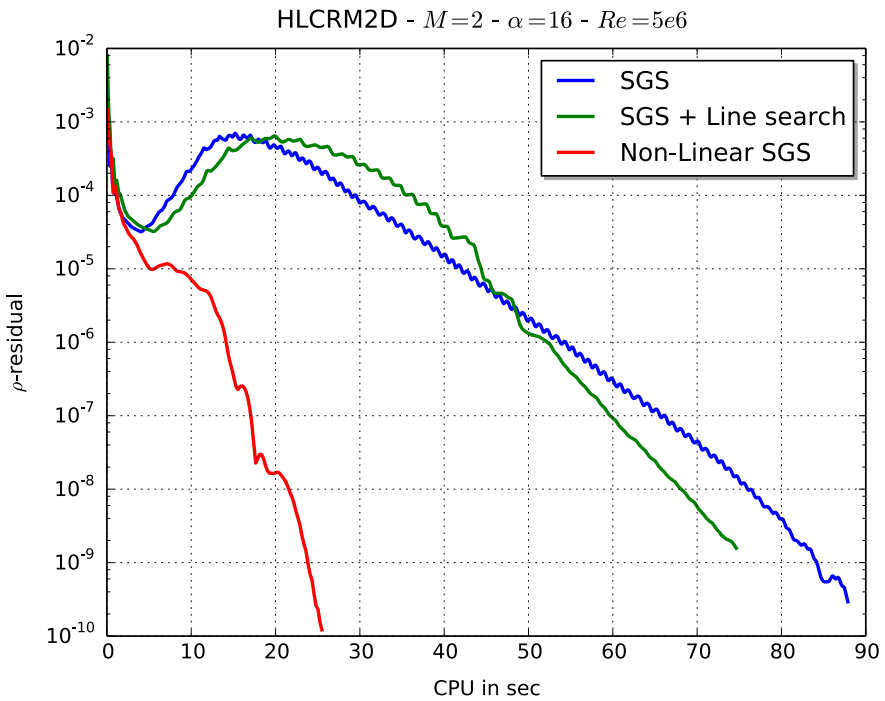
Implicit scheme from order 1 to order 2 in time

$$|C_i|\frac{\delta W_i^n}{\delta t} = \beta \mathbf{R}_i^{n+1} + (1-\beta)\mathbf{R}_i^n \quad \Longrightarrow \quad \left(\frac{|C|}{\delta t}\mathbf{I} - \beta\frac{\partial \mathbf{R}^n}{\partial W}\right)\delta W^n = \mathbf{R}^n \,,$$

- BDF1, BDF2, Crank-Nicolson
- SGS iterative solver - Coupling with multigrid and line solver available
- All terms are fully differentiated
  Only the convective terms are differentiated at order 1 but uses 2nd order solution
- Strong implicit solver can be source of great improvement in robustness, efficiency and accuracy
  - Under-relaxation coupled with CFL law
  - Line search
  - Non-linear SGS

Numerical illustration on the 2D HLCRM from HLPW4





Convergence to machine zero on this laptop (8 cores 2,3 GHz Intel Core i9)

Adapted mesh # of vertices	# of iterations	CPU in sec
20 K	123	25
40 K	178	80

Adjoint linear system

$$\left(\frac{\partial \mathbf{R}^n}{\partial W}\right)^T \delta W^* = \frac{\partial \mathbf{J}}{\partial W},$$

- Same as the flow solver (call the same functions) - manual differentiation
- GMRES with SGS preconditionner
- Important to converge the linear system at least at 10^{-12}
- Using extrapolated values for convective terms provides better adjoint

Outline

Concept of Metric-Based Mesh Adaptation

Main idea: change mesh generator distance and volume computation

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

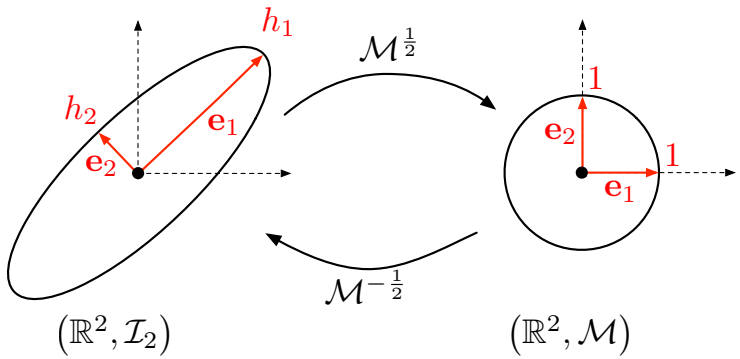
Fundamental concept: The notion of **metric** and Riemannian metric space

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

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

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

Concept of Metric-Based Mesh Adaptation

Main idea: change mesh generator distance and volume computation

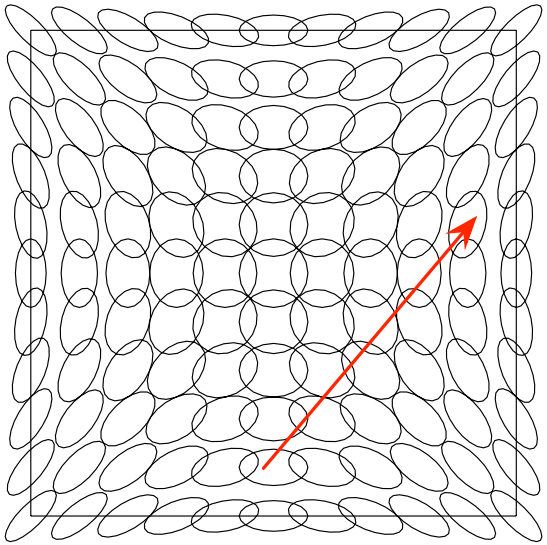
[George, Hecht and Vallet., Adv. Eng. Software 1991]

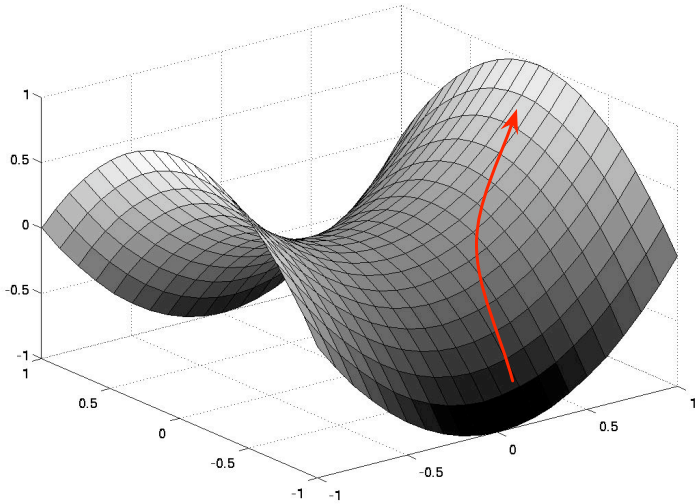
Fundamental concept: The notion of **metric** and Riemannian metric space

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

$\Updownarrow$

Computing geometric quantities on $\mathcal{S}$





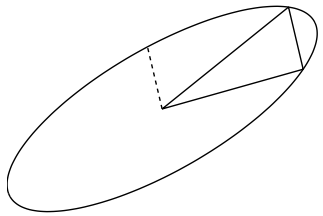
Concept of Metric-Based Mesh Adaptation

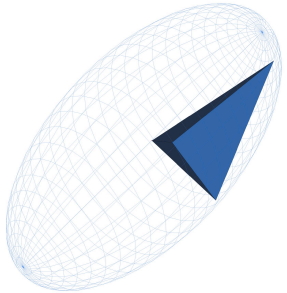
Main idea: change mesh generator distance and volume computation

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

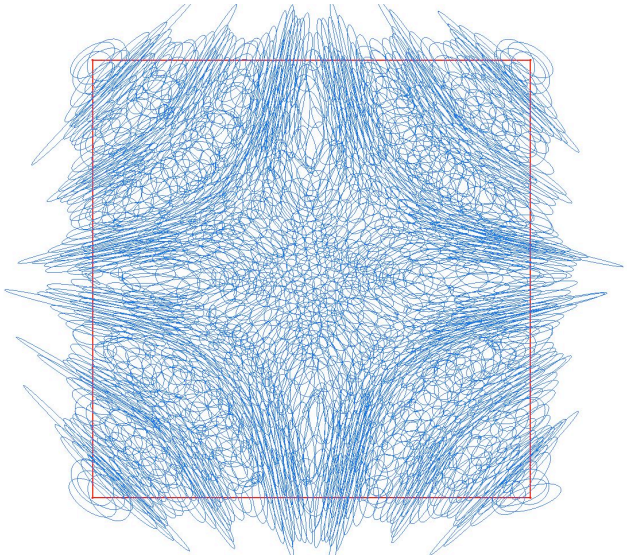
Fundamental concept: Generate a **unit mesh** w.r.t $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

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

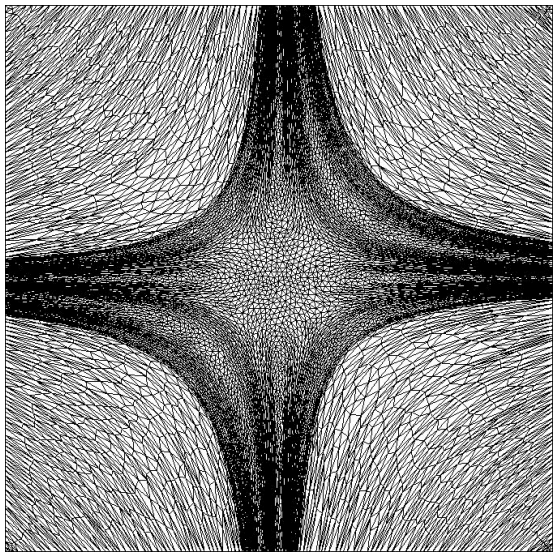




Inputs $(\mathcal{H}_0, \mathcal{M}_i)_{i\in\mathcal{H}}$



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 $\alpha \left(\det \mathcal{M}\right)^{-\frac{1}{2}}$
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 ($u(\mathbf{x}) = \frac{1}{2}\, {}^t\mathbf{x}\, H_u\, \mathbf{x}$):

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

Working in this framework enables us to use powerful mathematical tool

Feature-Based Error Estimates

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

Deriving the **best mesh** to compute the characteristics of a given solution **w**

- Solve optimization problem:

$$\text{Find } \mathbf{M}_{L^p} \text{ such that } \mathbf{E}_{L^p}(\mathbf{M}_{L^p}) = \min_{\mathbf{M}} \left(\int_\Omega \left(w(\mathbf{x}) - \pi_{\mathcal{M}} w(\mathbf{x}) \right)^p \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}}$$

$$\text{under constraint } \mathcal{C}(\mathbf{M}) = \int_\Omega d_{\mathcal{M}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = N.$$

- Optimal continuous mesh minimizing the interpolation error in L^p-norm:

$$\mathcal{M}_{L^p}(H_w) = D_{L^p}(N) \left(det|H_w| \right)^{\frac{-1}{2p+d}} |H_w|$$

- The choice of the L^p norm is up to the user depending on the case of application

Goal-Oriented Error Estimate

Goal-oriented anisotropic mesh adaptation [Loseille et al., JCP2010, Belme et al., JCP2018, ...]

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

Inviscid case:

- Formal non-linear a priori error analysis (transport equation):

$$|j(w) - j(w_h)| \approx ((\Psi - \Psi_h)(w), w^*)$$

- Application to compressible Euler equations:

$$\begin{aligned} |j(W) - j(W_h)| &\approx \| |\nabla W^*| \cdot |\mathcal{F}^E(W) - \Pi_h \mathcal{F}^E(W)| \|_{L^1(\Omega_h)} \\ &\approx \left\| \left| \frac{\partial \mathcal{F}^E}{\partial W} \cdot \nabla_x W^* \right| |W - \Pi_h W| \right\|_{L^1(\Omega_h)} \end{aligned}$$

- Optimal continuous mesh minimizing the weighted interpol. error in L^1-norm:

$$\mathcal{M}_{L^1}\left(\sum_{k=1}^{5} \left(\left| \frac{\partial \mathcal{F}_k^E}{\partial W} \cdot \nabla_x W_k^* \right| |H(W_k)| \right) \right)$$

Goal-Oriented Error Estimate

Goal-oriented anisotropic mesh adaptation [Loseille et al., JCP2010, Belme et al., JCP2018, ...]

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

Viscous case:

- Treat viscous term by integration by part and linearization:

$$
\begin{aligned}
&|J(W) - J(W_h)| \\
&\quad \approx \int_\Omega |W^*| \left| \nabla \cdot (\mathcal{F}^E(W) - \mathcal{F}^E(\Pi_h W)) - \nabla \cdot (\mathcal{F}^V(W) - \mathcal{F}^V(\Pi_h W)) \right| \mathrm{d}\Omega \\
&\quad \leq \left| \int_\Omega \sum_i \nabla_{x_i} \left[\frac{\partial \mathcal{F}_i^E}{\partial W} (W - \Pi_h W) \right] W^* + \sum_{i,j} \nabla_{x_i} \left[\frac{\partial \mathcal{F}_i^V}{\partial \nabla_{x_j} W} \nabla_{x_j} (W - \Pi_h W) \right] W^* \, \mathrm{d}\Omega \right| \\
&\quad \leq \int_\Omega \left| \sum_i \frac{\partial \mathcal{F}_i^E}{\partial W} \nabla_{x_i} W^* + \sum_{i,j} \frac{\partial \mathcal{F}_i^V}{\partial \nabla_{x_j} W} \nabla_{x_i, x_j} W^* \right| |W - \Pi_h W| \, \mathrm{d}\Omega
\end{aligned}
$$

- Optimal continuous mesh minimizing the weighted interpol. error in L^1-norm:

$$
\mathcal{M}_{L^1} \left(\sum_{k=1}^{5} |G_k(W, W^*)| \, |H(W_k)| \right)
$$

Outline

Anisotropic Mesh Adaptation Algorithm

Mesh adaptation is a non-linear problem

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

Convergence study is free if mesh complexity is increased during the process

In that case, a "multigrid effect" is obtained (faster convergence)

Advantageous to converge on coarse meshes

Anisotropic Mesh Adaptation Algorithm

Initial mesh $\mathcal{H}_0^0$, solution S_0^0, adjoint $(S^*)_0^0$, and complexity $\mathcal{C}^0$

//--- Outer loop to perform the convergence study

For j=0,ncmp

  //--- Inner loop to converge the mesh adaptation at fixed complexity

  For i=1,nadap

    $\mathcal{M}_{i-1}^j = \texttt{ComputeErrorEstimateMetric}(\mathcal{C}^j, \mathcal{H}_{i-1}^j, \mathcal{S}_{i-1}^j, (\mathcal{S}^*)_{i-1}^j)$

    $\mathcal{H}_i^j = \texttt{GenerateAdaptedMesh}(\mathcal{H}_{i-1}^j, \mathcal{M}_{i-1}^j)$

    $\mathcal{S}_{i,0}^j = \texttt{InterpolateSolution}(\mathcal{H}_{i-1}^j, \mathcal{S}_{i-1}^j, \mathcal{H}_i^j)$

    $(\mathcal{S}_i^j, (\mathcal{S}^*)_i^j) = \texttt{SolveStateAndAdjoint}(\mathcal{S}_{0,i}^j, \mathcal{H}_i^j)$

    If (Res < Tol) i = ni else i = i+1

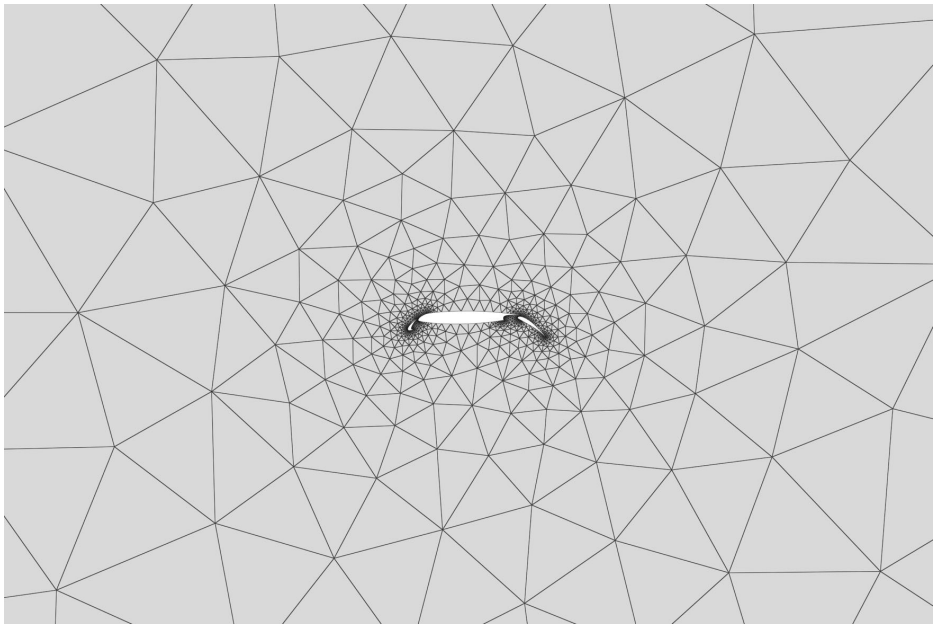
  End for

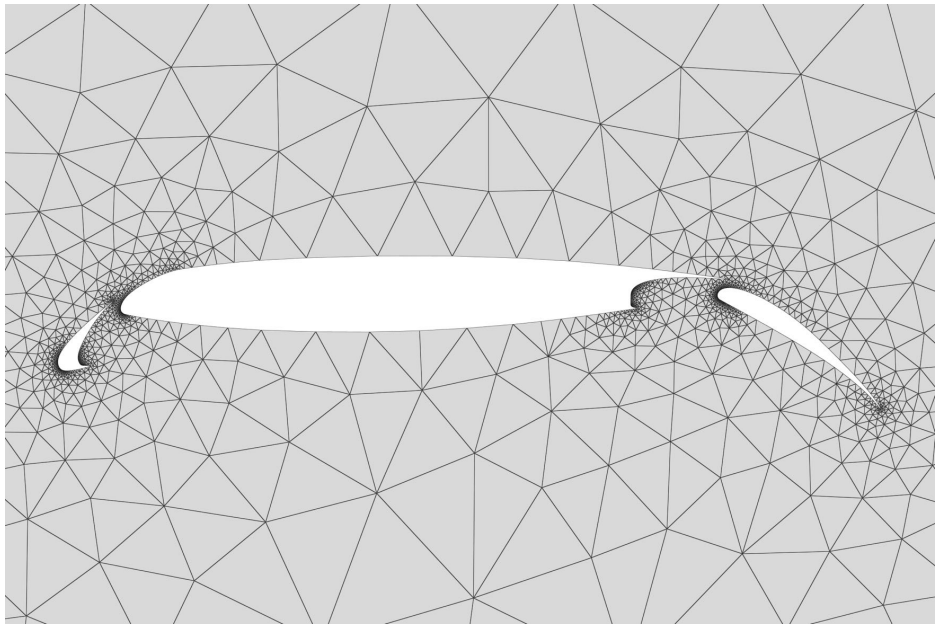
  $\mathcal{H}_0^{j+1} = \mathcal{H}_{ni}^j$; $\mathcal{S}_0^{j+1} = \mathcal{S}_{ni}^j$; $(\mathcal{S}^*)_0^{j+1} = (\mathcal{S}^*)_{ni}^j$; $\mathcal{C}^{j+1} = \alpha \mathcal{C}^j$

End for

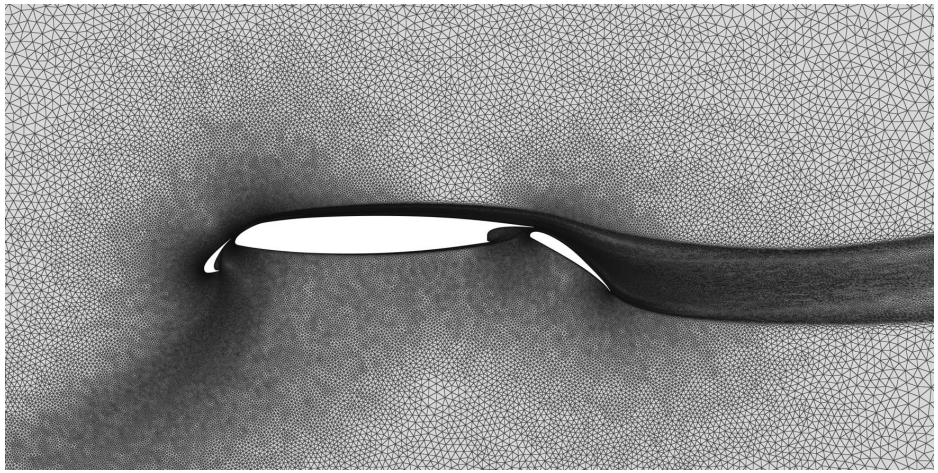
2D AIAA HLPW4 Common Research Model

- Geometry: 2D Section of the HLPW4 Common Research Model
- Initial mesh: $\#Vertices = 2\,651$ and $\#Triangles = 4\,616$
- Flow characteristics: $Mach = 0.2$, $\alpha = 16°$, $Re = 5 \times 10^6$
- 9 complexities: from 4 000 to 1 280 000 multiplied by 2 each time
- Maximal number of adaptations is 15 per complexity

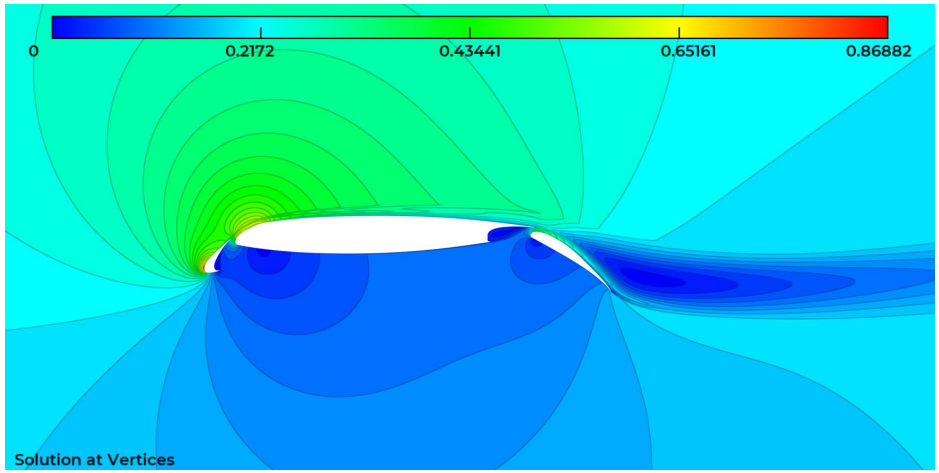




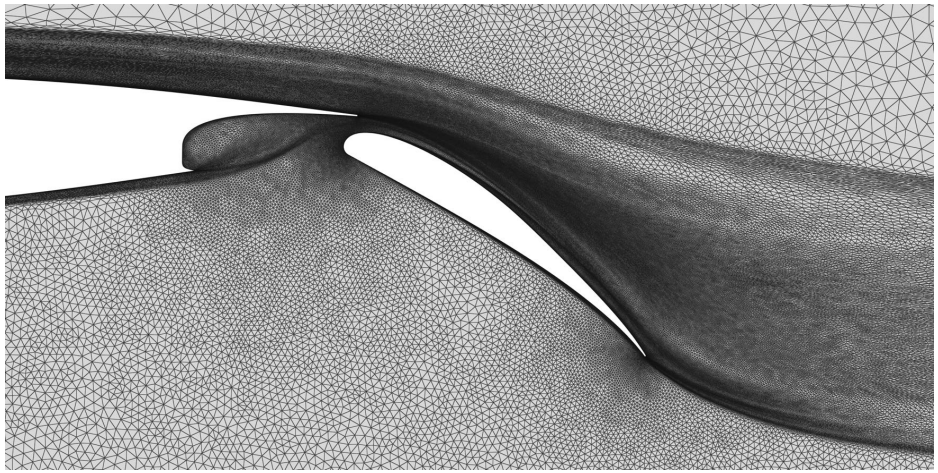
2D AIAA HLPW4 Common Research Model



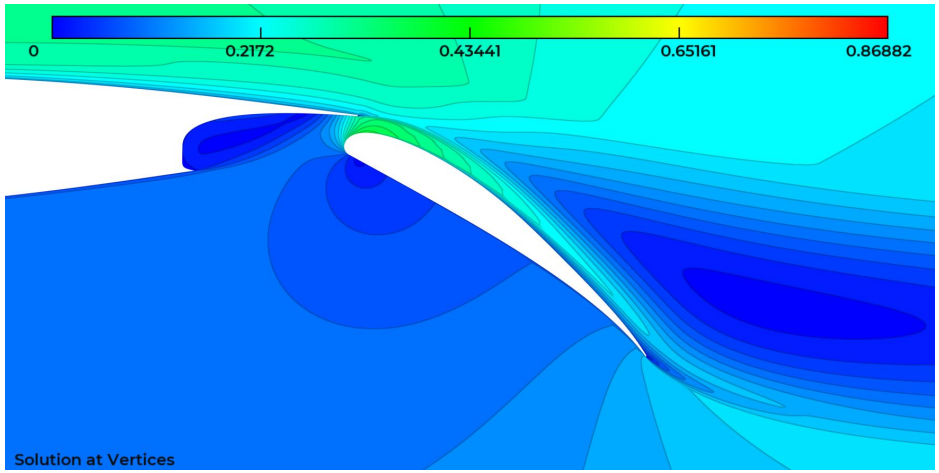
Overall view



Local Mach number

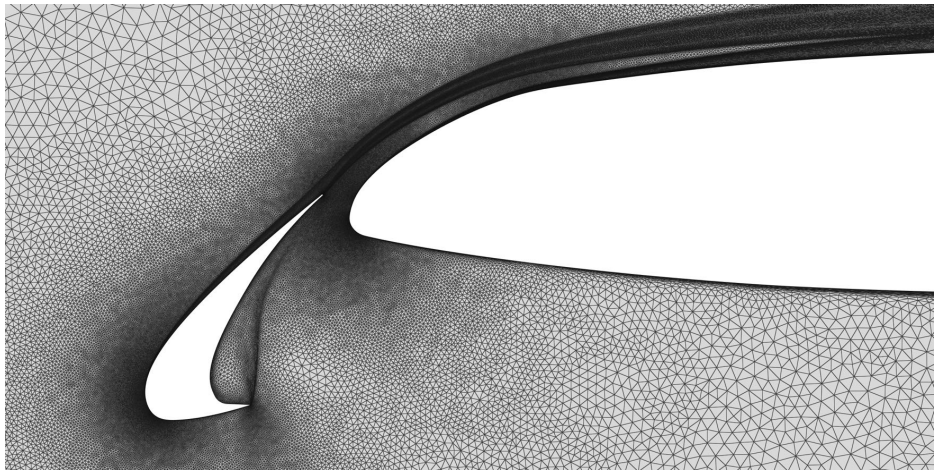


Flap view

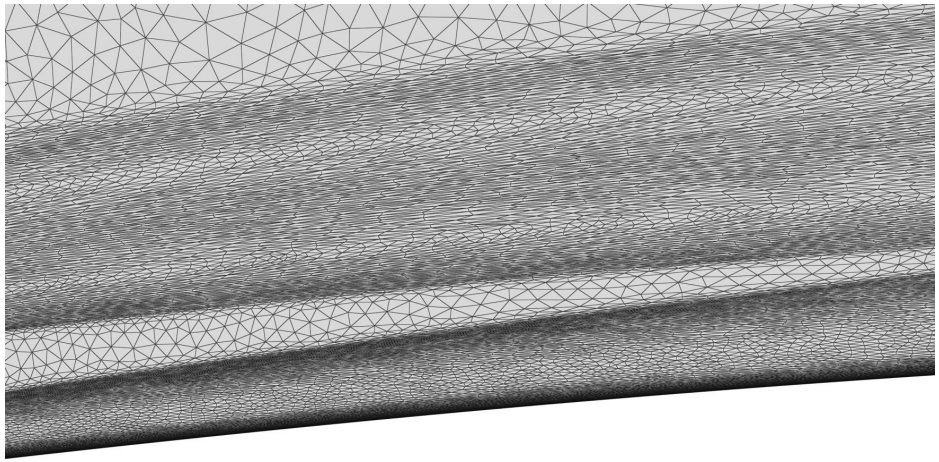


Local Mach number

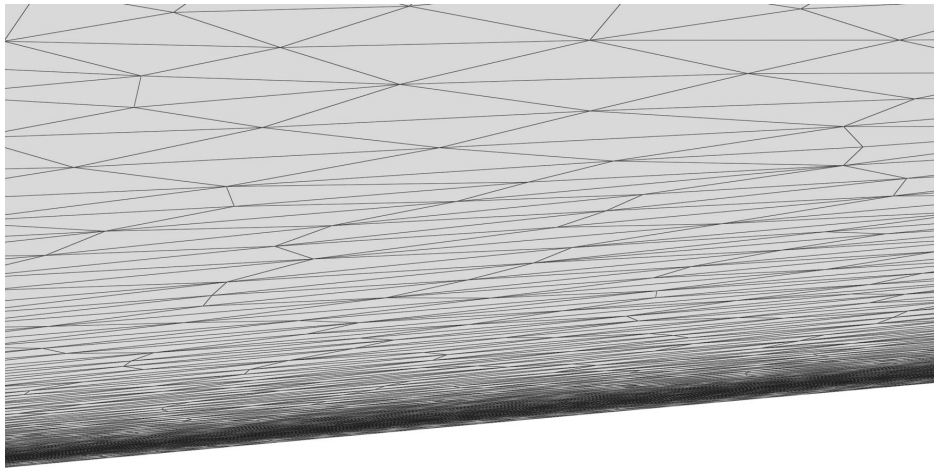
2D AIAA HLPW4 Common Research Model



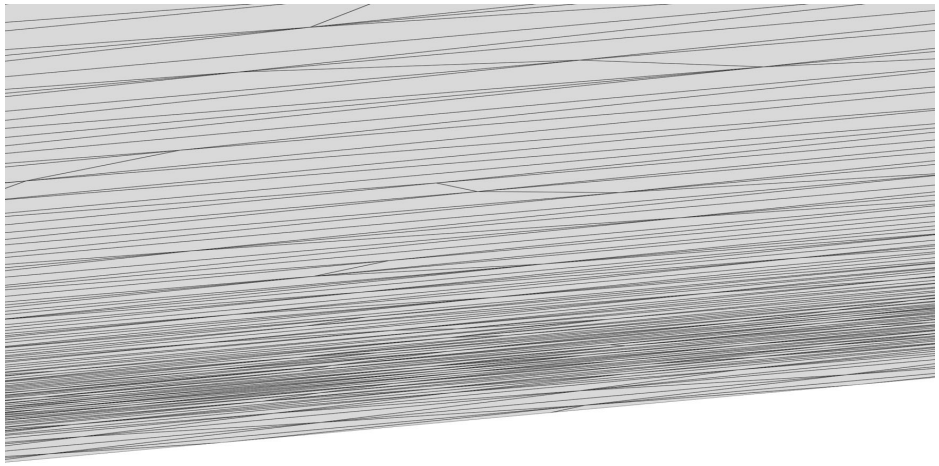
Slat view



Above main wing



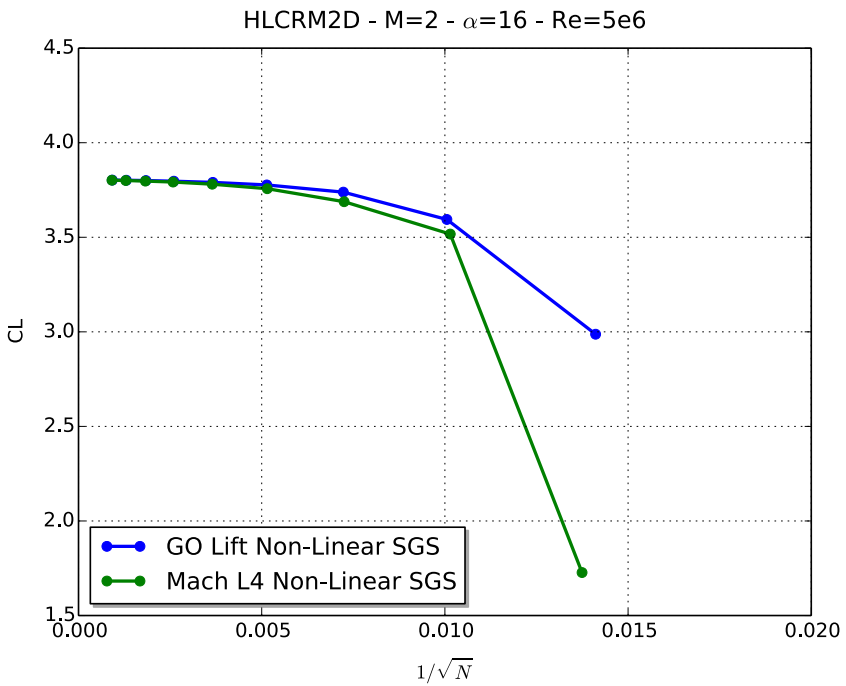
Above main wing far BL

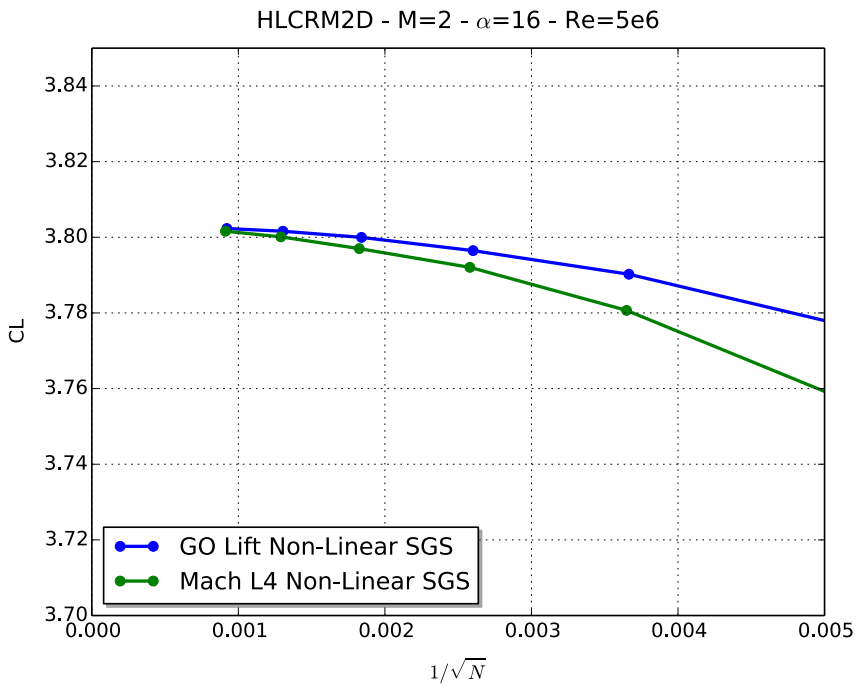


Above main wing BL

2D AIAA HLPW4 Common Research Model

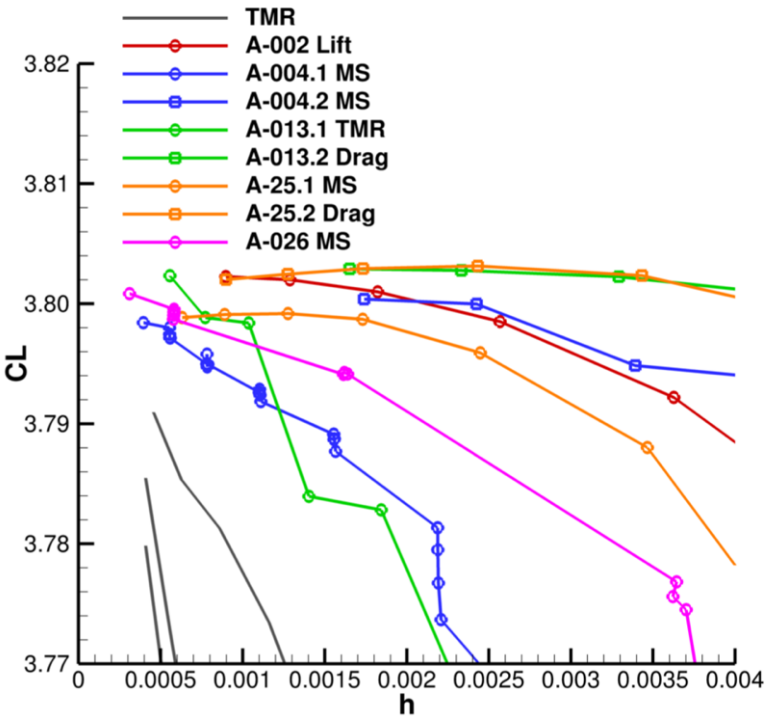
Lift convergence

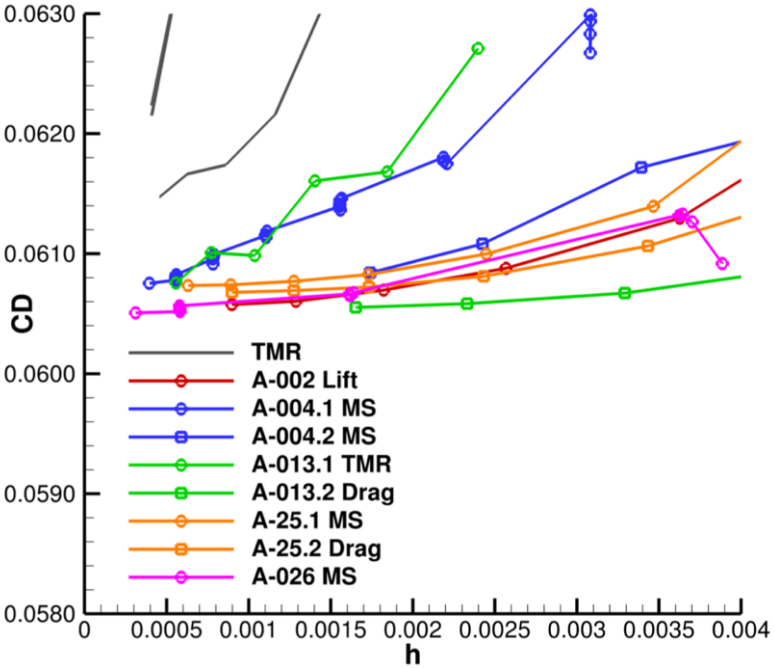




Group	Method	CL	CD	CDv	CM
Boeing	GGNS-EPIC-L4	3.80170	0.060654	0.009391	-0.38290
Boeing	GGNS-EPIC-Lift	3.80195	0.060673	0.009375	-0.38296
MIT	SANS-EPIC-MOES	3.80284	0.060718	0.009376	-0.38314
NASA	SFE-REFINE-MS	3.80038	0.060841	0.009370	-0.38280
Inria	Wolf-Fefloa-L4 NL-SGS 2021	3.80160	0.060601	0.009395	-0.38285
Inria	Wolf-Fefloa-Lift NL-SGS 2021	3.80232	0.060565	0.009385	-0.38298

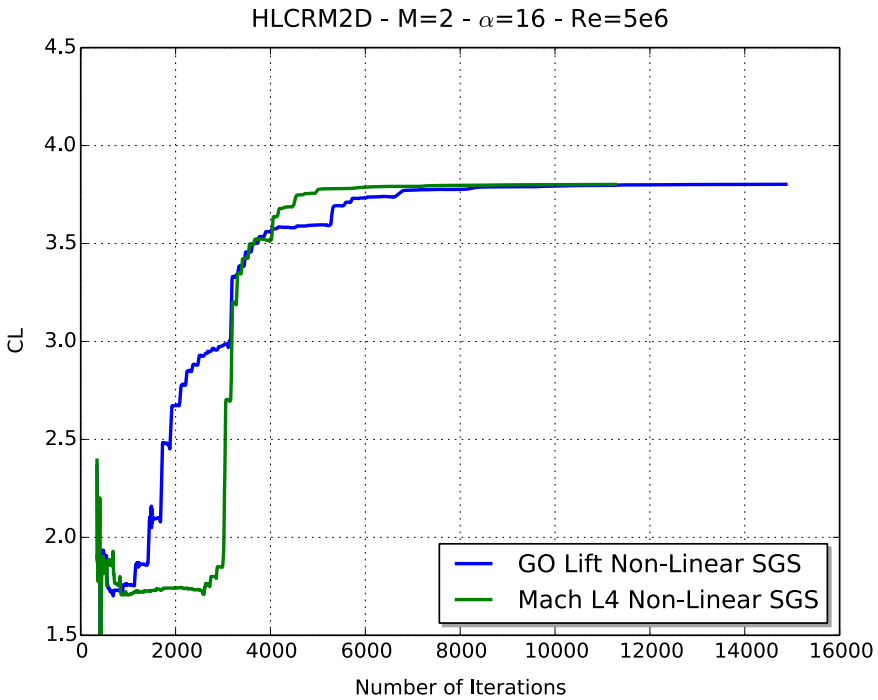
2D AIAA HLPW4 Common Research Model

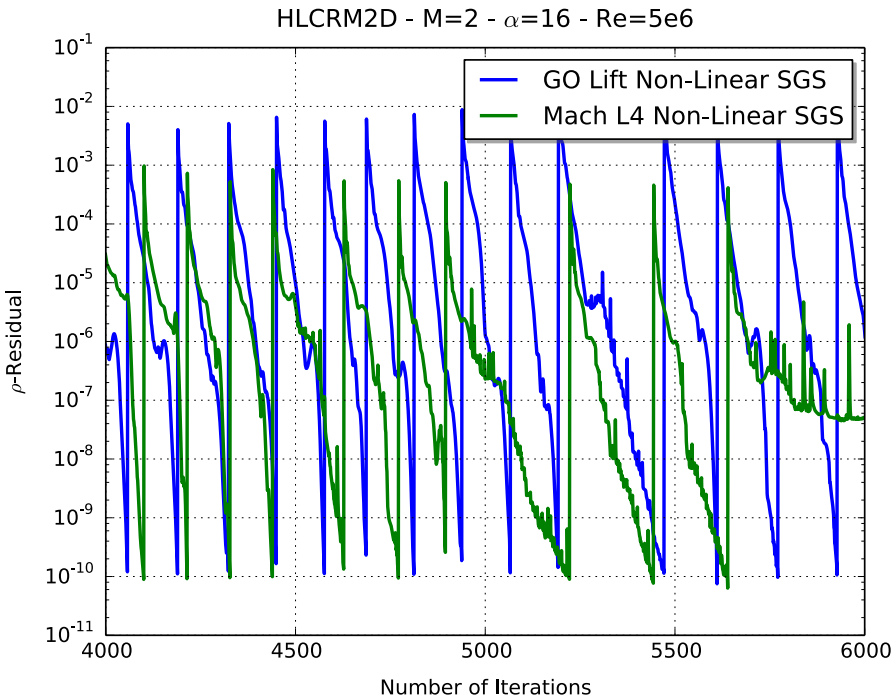




2D AIAA HLPW4 Common Research Model

Convergence history over the whole process

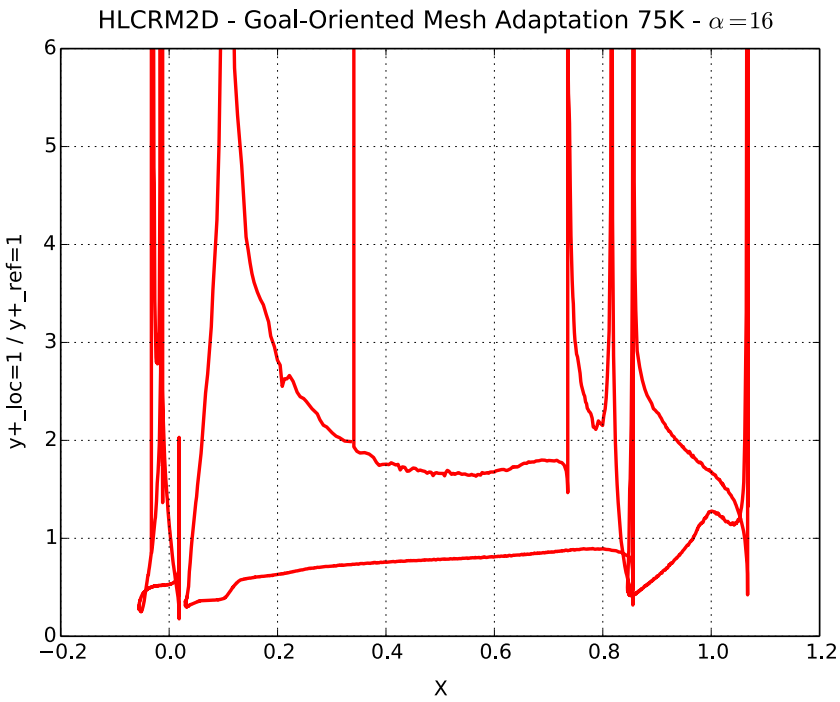




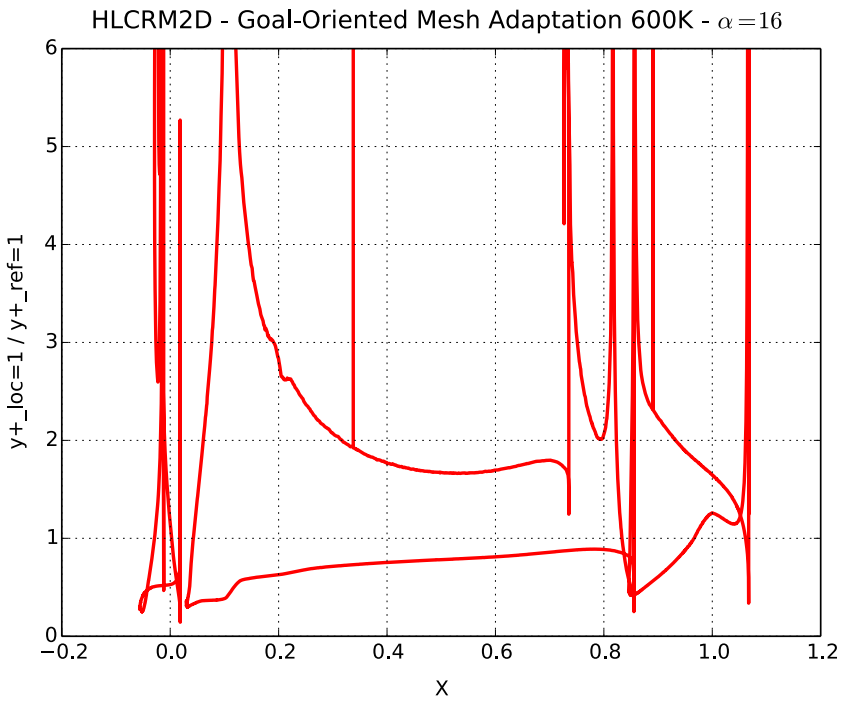
Convergence to "machine zero" is required at each computation of the mesh adaptation process

2D AIAA HLPW4 Common Research Model

$y^+ = 1$ ratio between the local value and the reference value (from the Reynolds number)



75K vertices

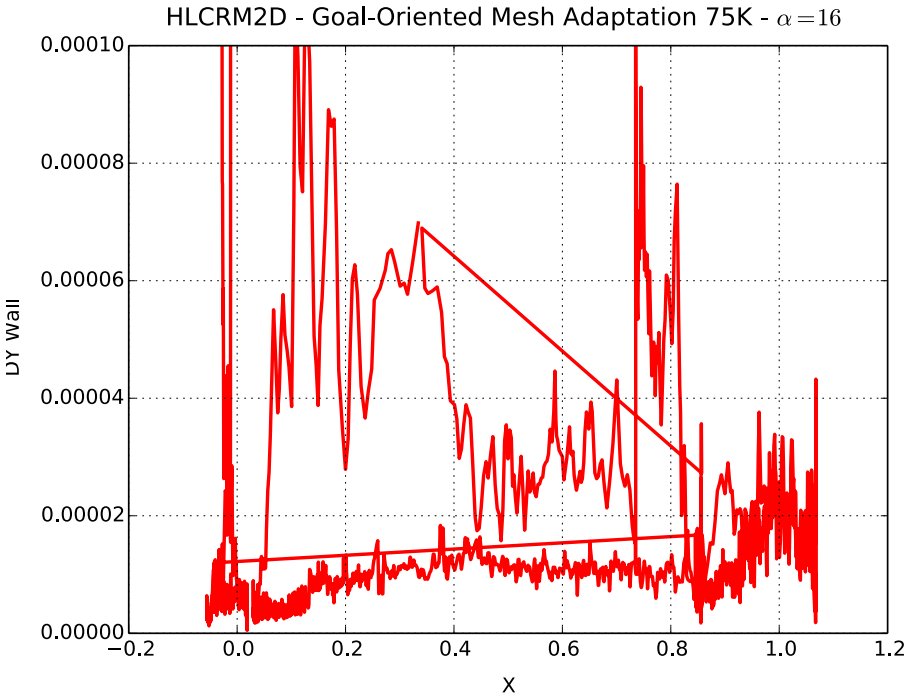


600K vertices

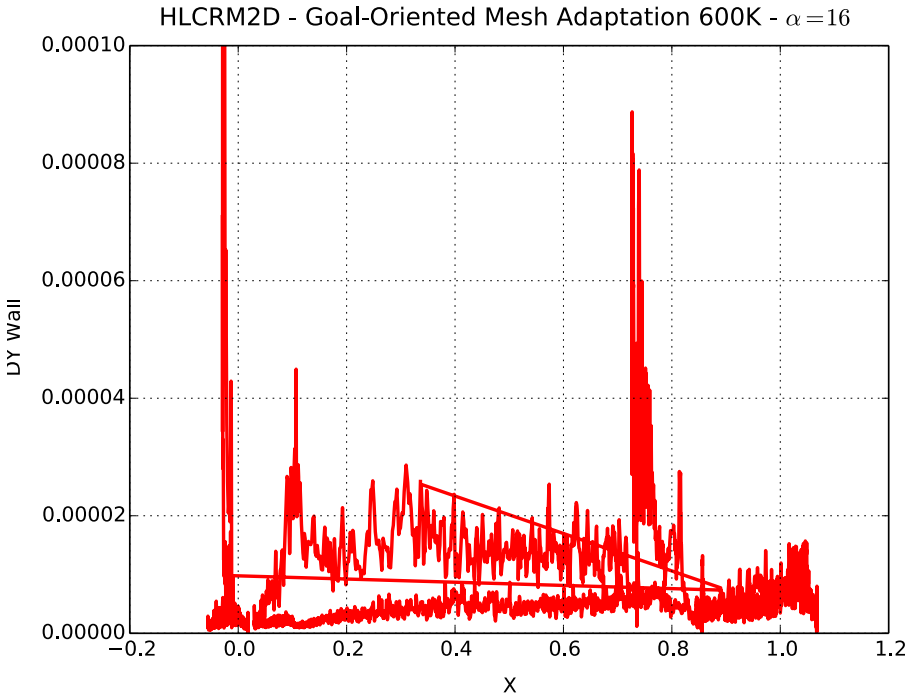
The $y^+ = 1$ ratio given by the mesh adaptation process is consistent whatever the mesh complexity

2D AIAA HLPW4 Common Research Model

Size prescription at wall by the mesh adaptation process



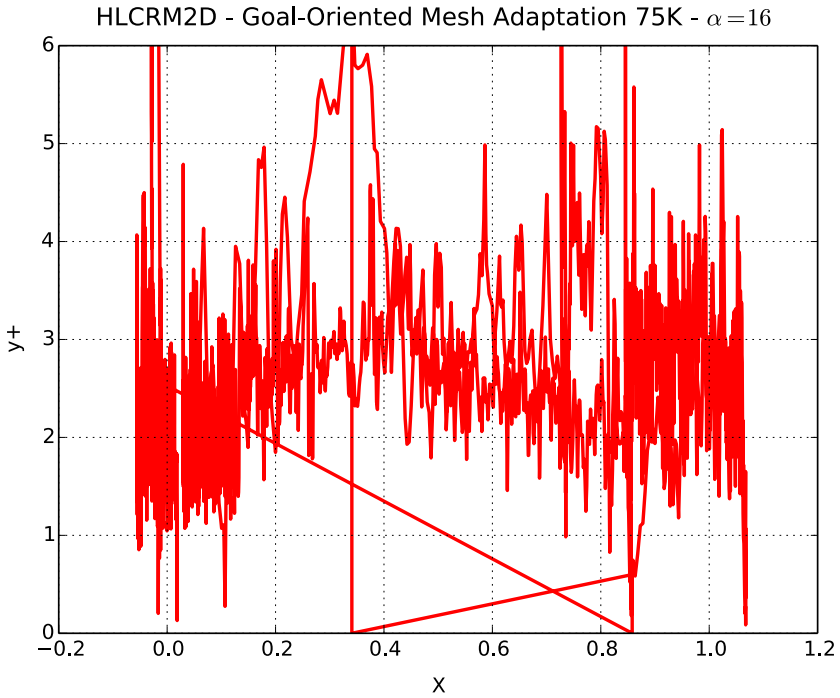
75K vertices



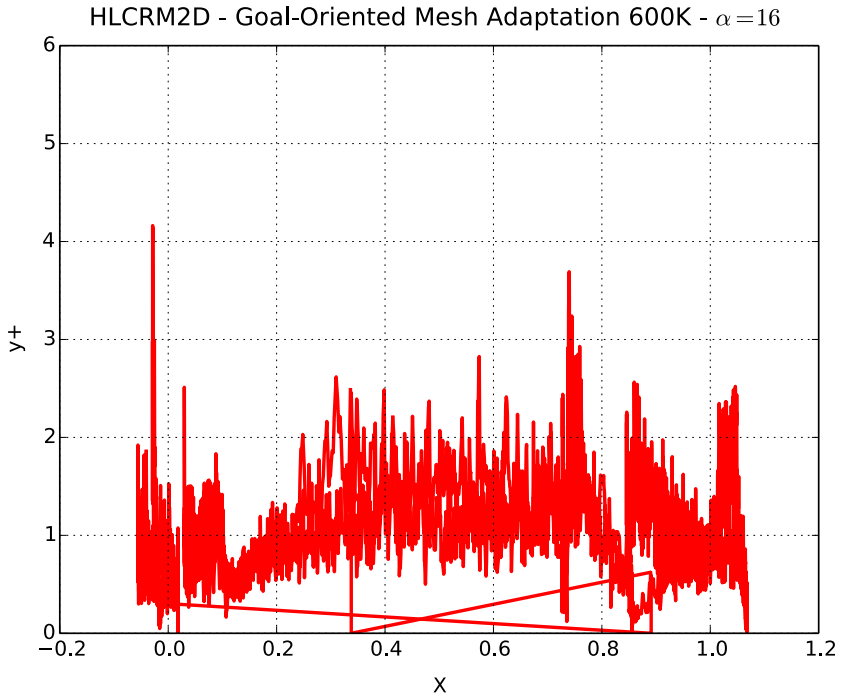
600K vertices

2D AIAA HLPW4 Common Research Model

y^+ prescription at wall by the mesh adaptation process



75K vertices

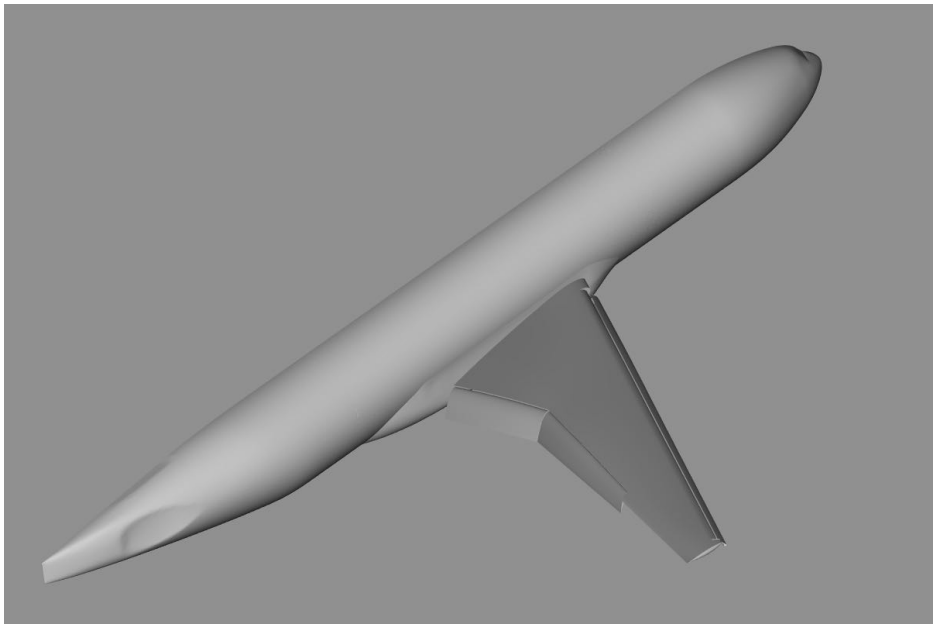


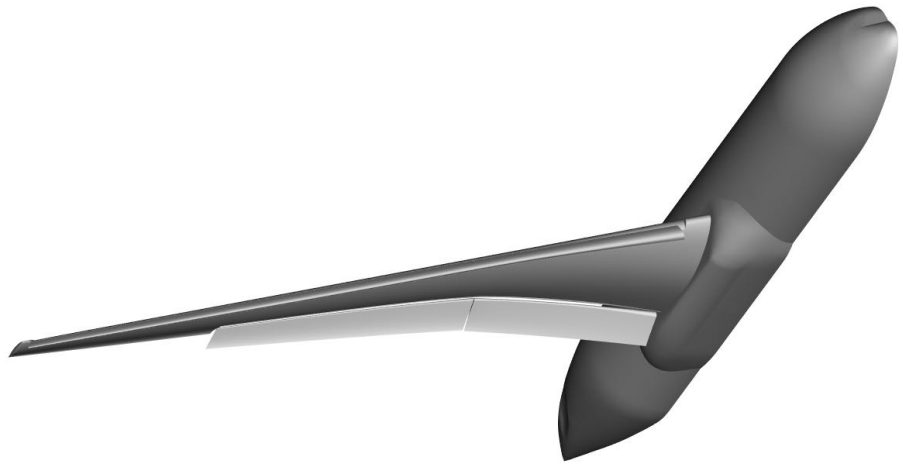
600K vertices

Outline

3D AIAA HLPW3 Common Research Model

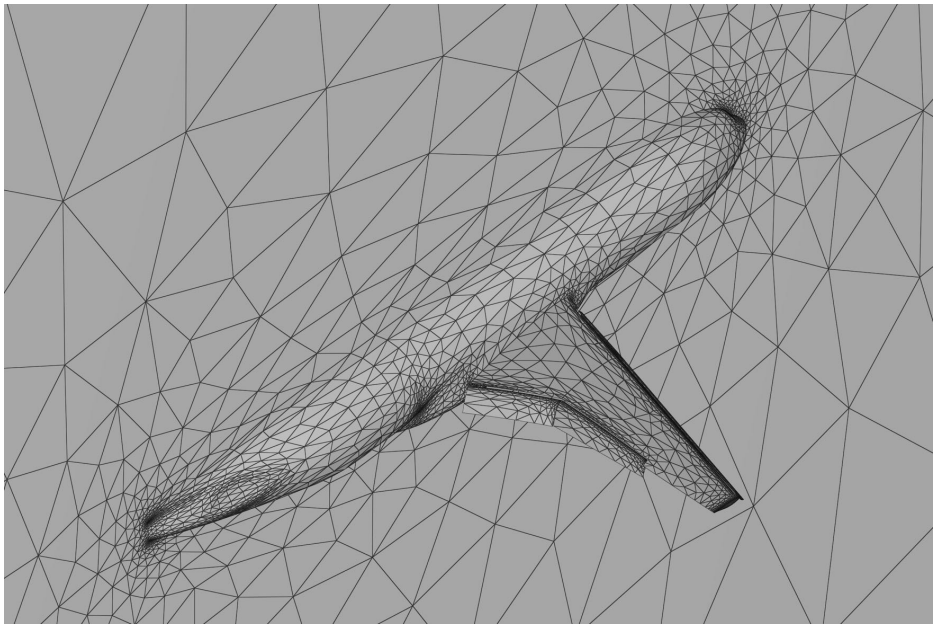
- Geometry: HLPW3 Common Research Model
- Initial mesh: $\#Vertices = 229\,263$ and $\#Tetrahedra = 726\,852$
- Flow characteristics: $Mach = 0.2$, $\alpha = 16°$, $Re = 3.26 \times 10^6$
- 7 complexities : from 600 000 to 20 000 000 multiplied by 2 each time
- Maximal number of adaptations is 15 per complexity

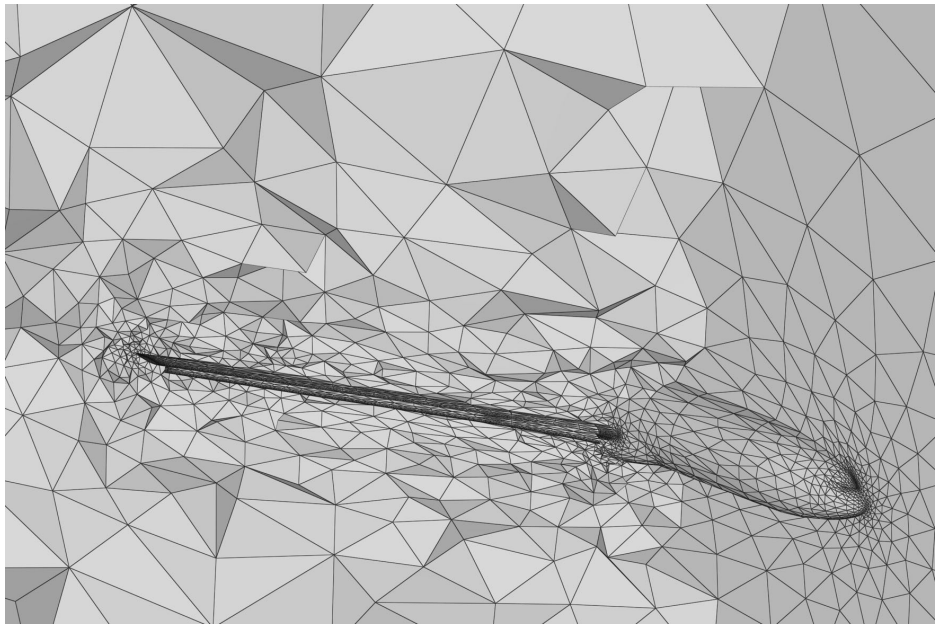




3D AIAA HLPW3 Common Research Model

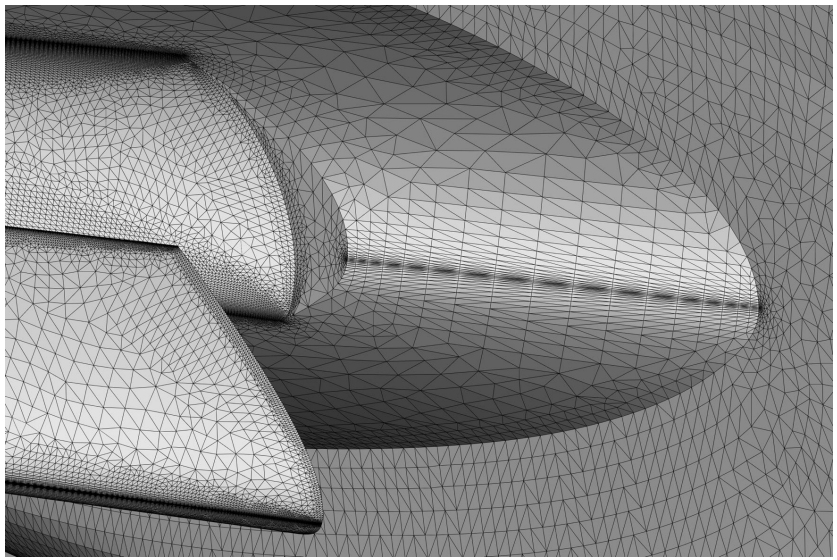
- Geometry: HLPW3 Common Research Model
- Initial mesh: $\#Vertices = 229\,263$ and $\#Tetrahedra = 726\,852$
- Flow characteristics: $Mach = 0.2$, $\alpha = 16°$, $Re = 3.26 \times 10^6$
- 7 complexities : from 600 000 to 20 000 000 multiplied by 2 each time
- Maximal number of adaptations is 15 per complexity



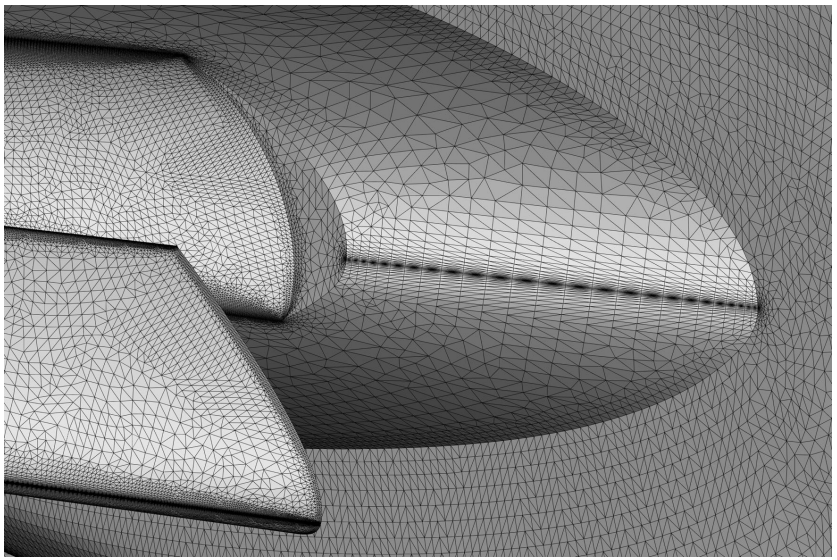


3D AIAA HLPW3 Common Research Model

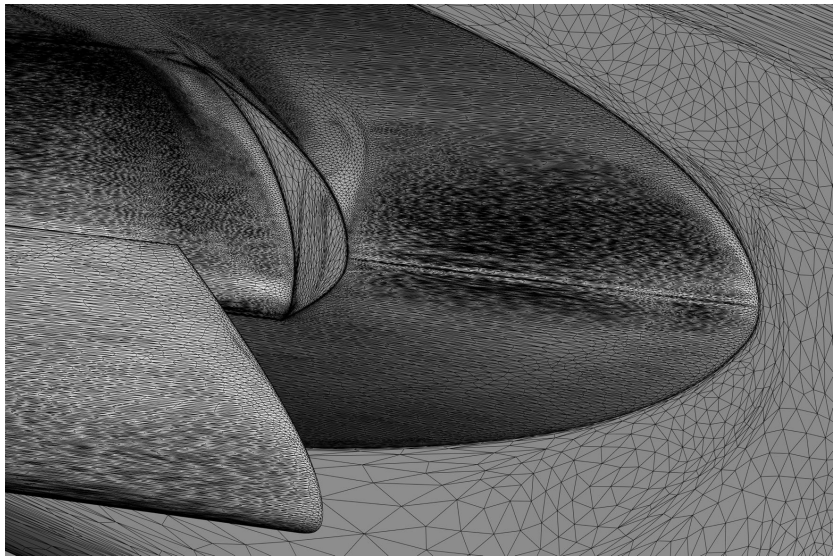
26 491 087 vertices



69 890 932 vertices

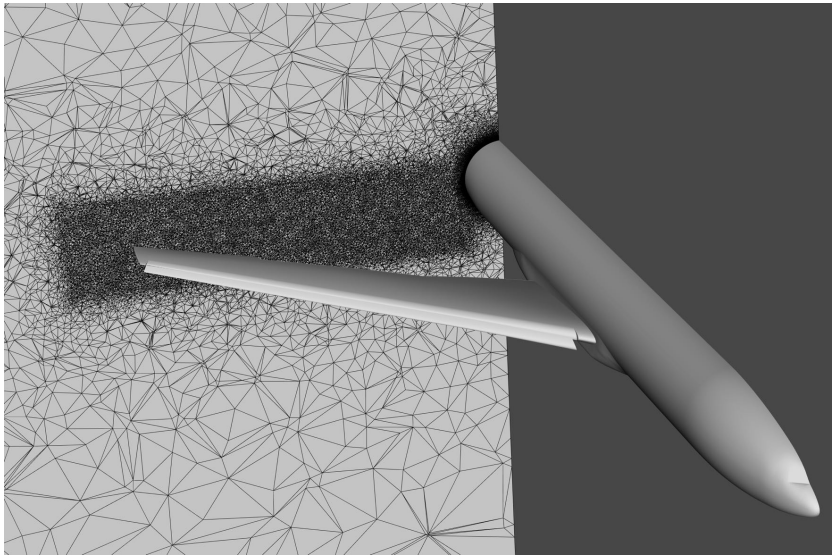


5 318 448 vertices

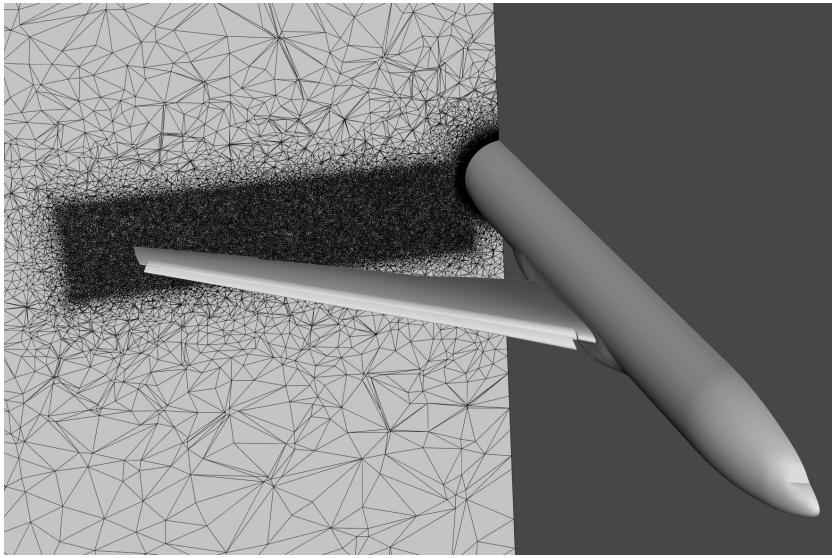


3D AIAA HLPW3 Common Research Model

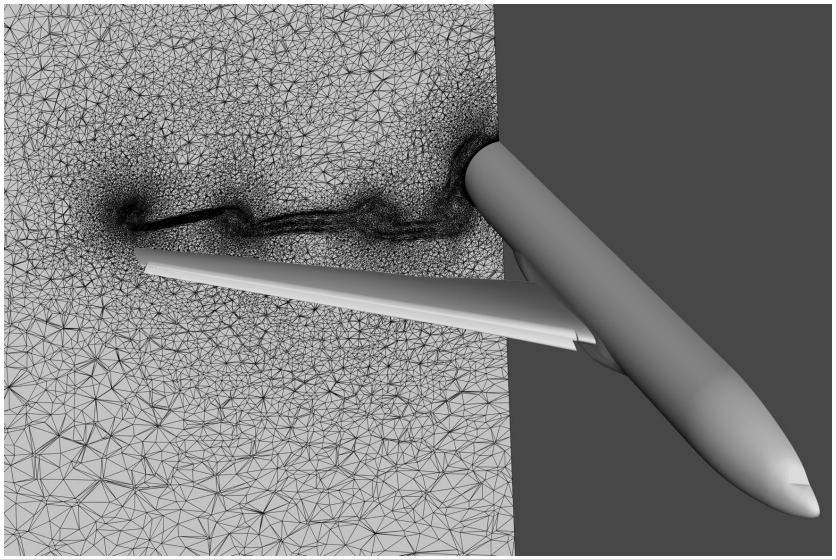
26 491 087 vertices



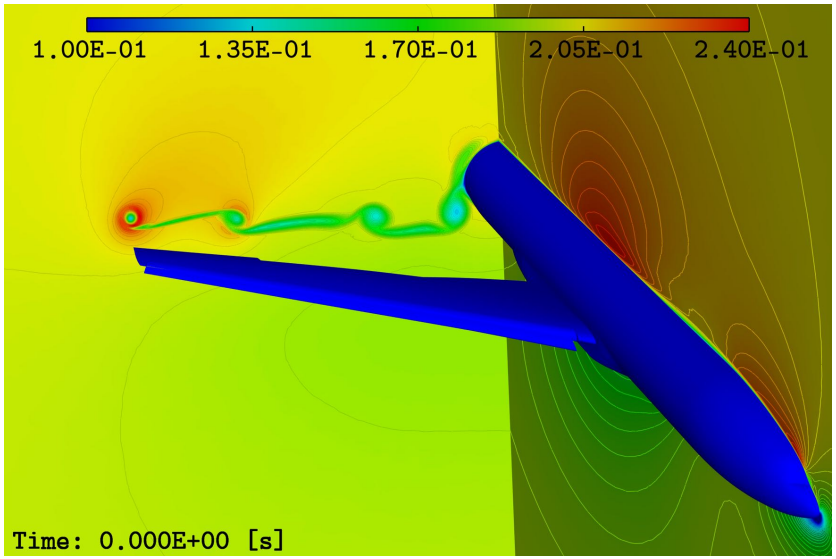
69 890 932 vertices



5 318 448 vertices

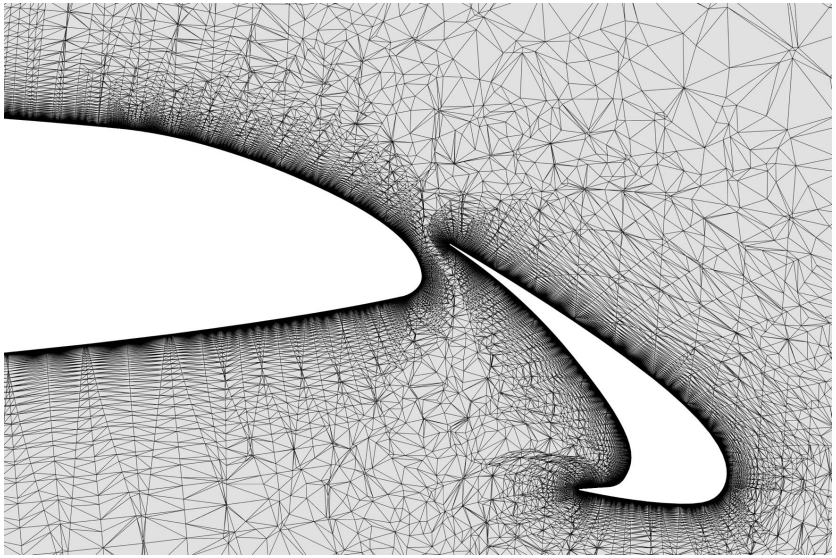


Local Mach Number

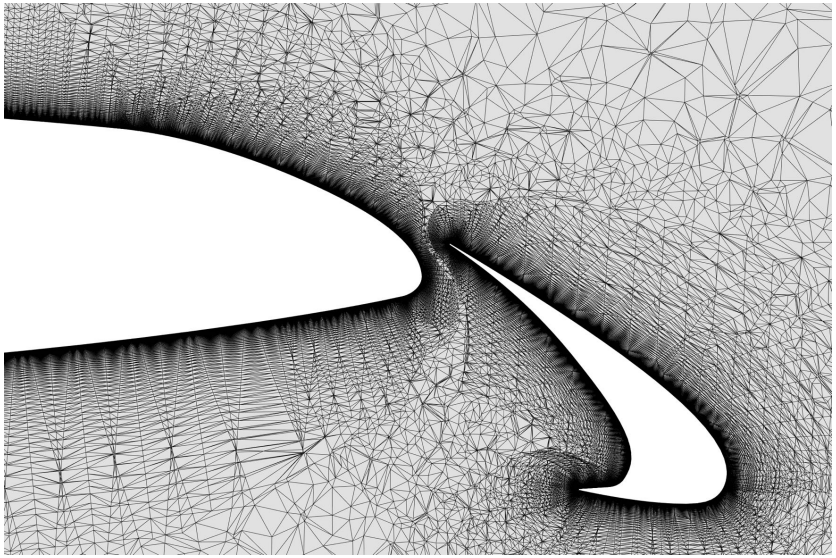


3D AIAA HLPW3 Common Research Model

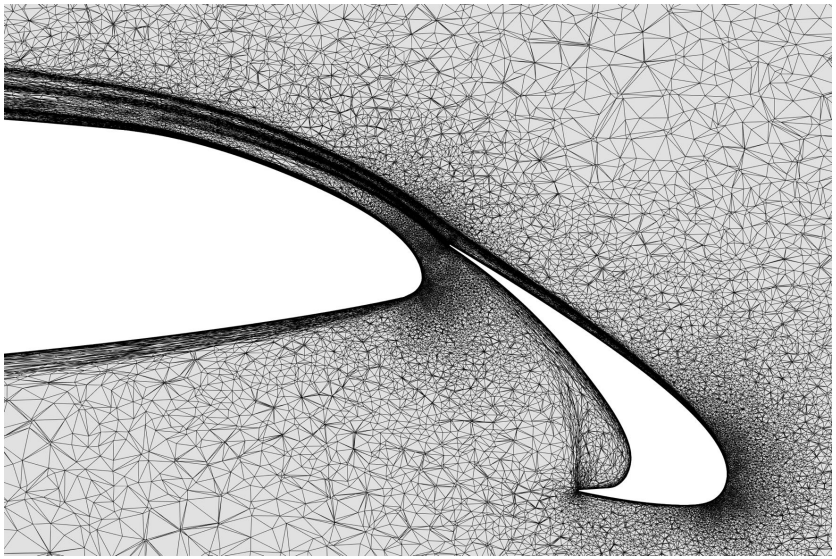
26 491 087 vertices



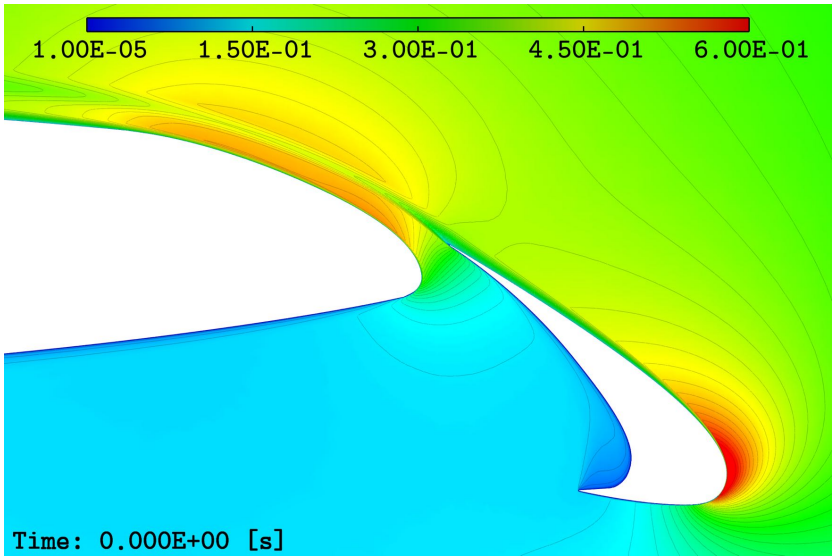
69 890 932 vertices



5 318 448 vertices

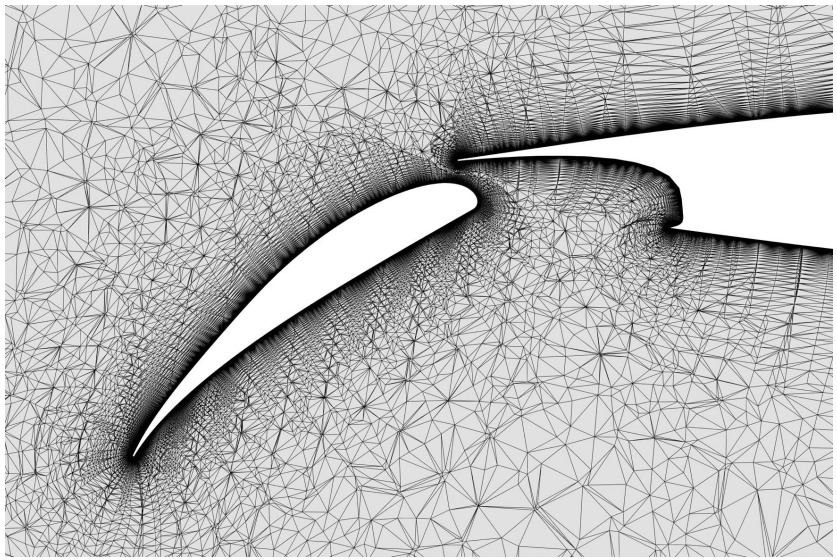


Local Mach Number

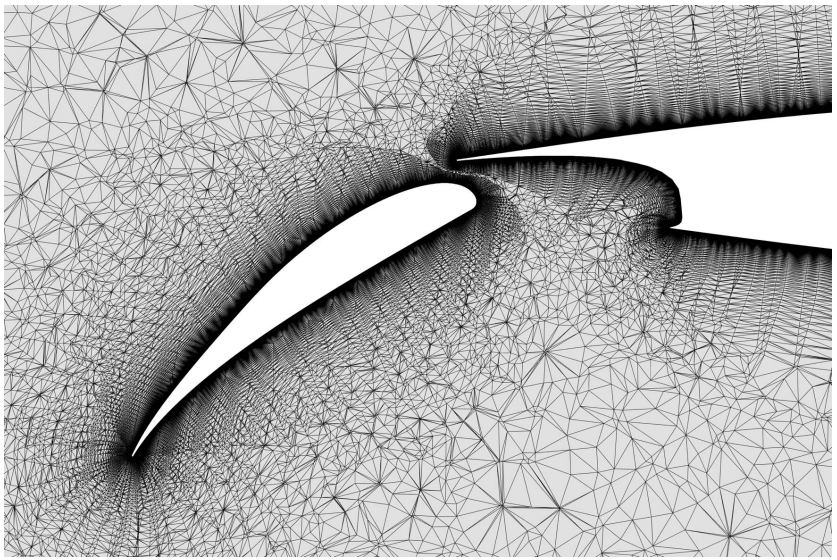


3D AIAA HLPW3 Common Research Model

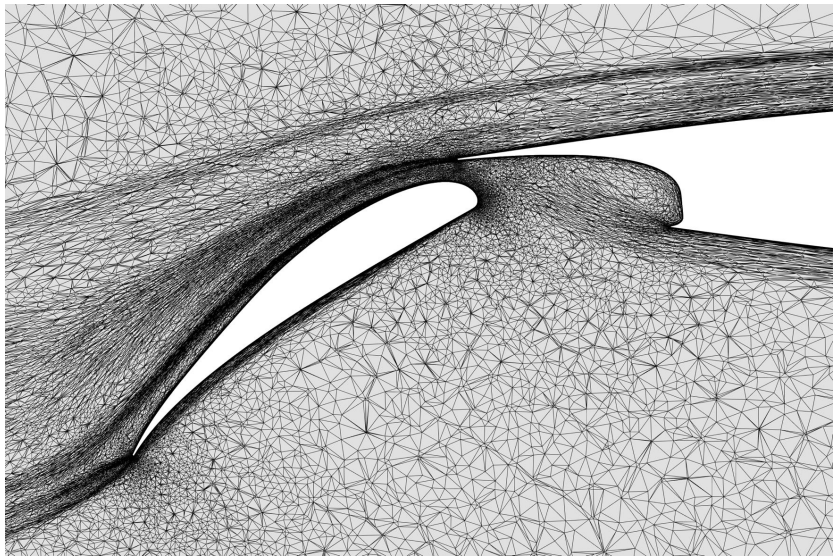
26 491 087 vertices



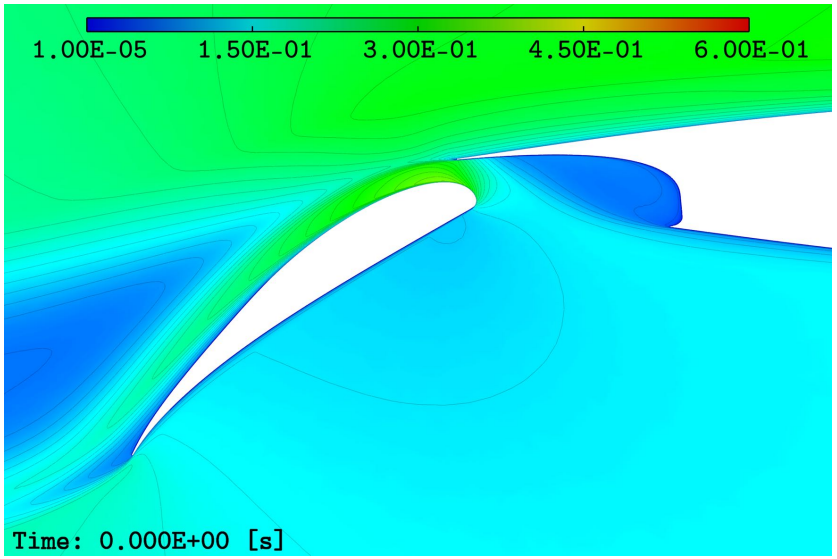
69 890 932 vertices



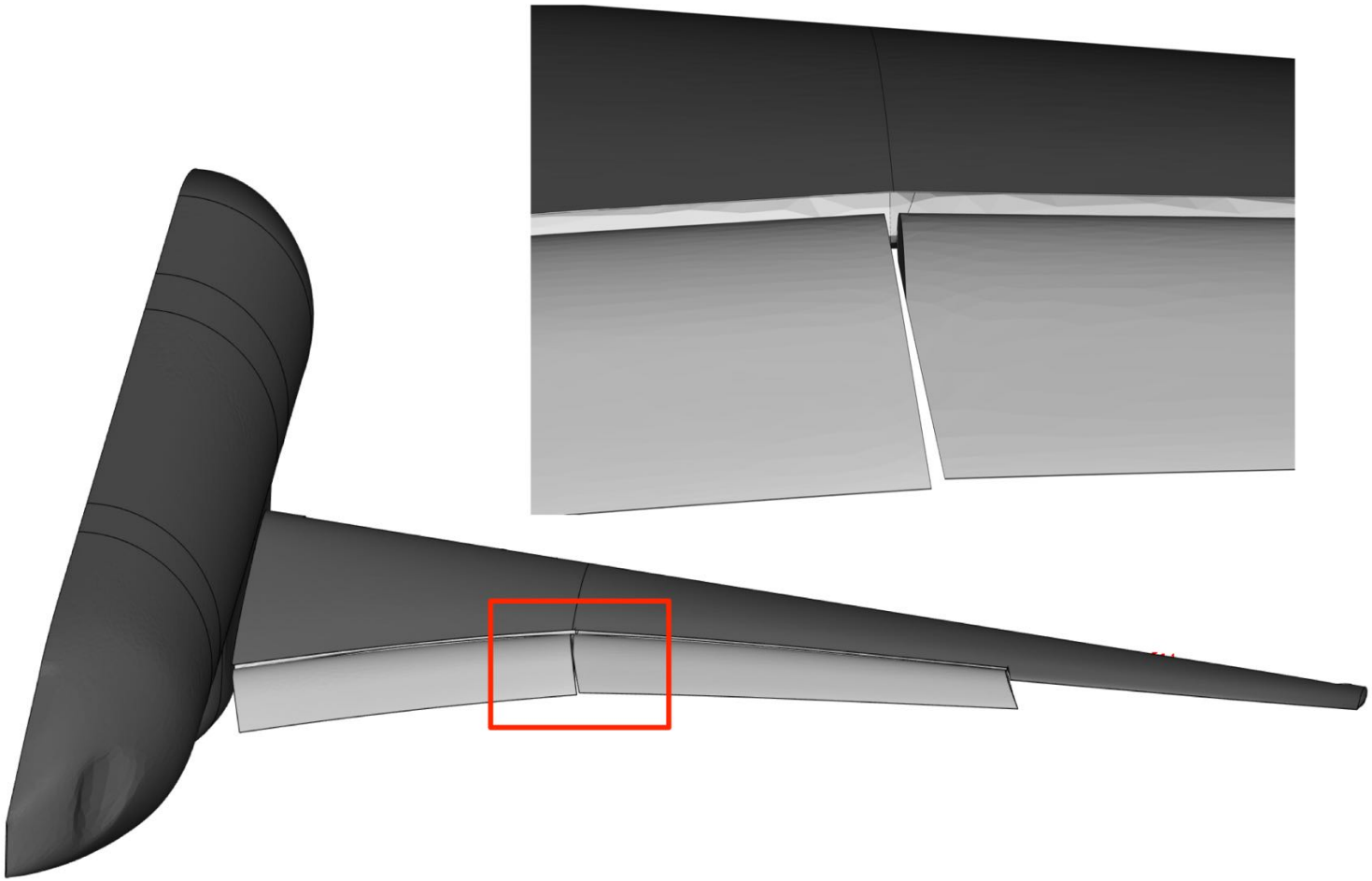
5 318 448 vertices



Local Mach Number

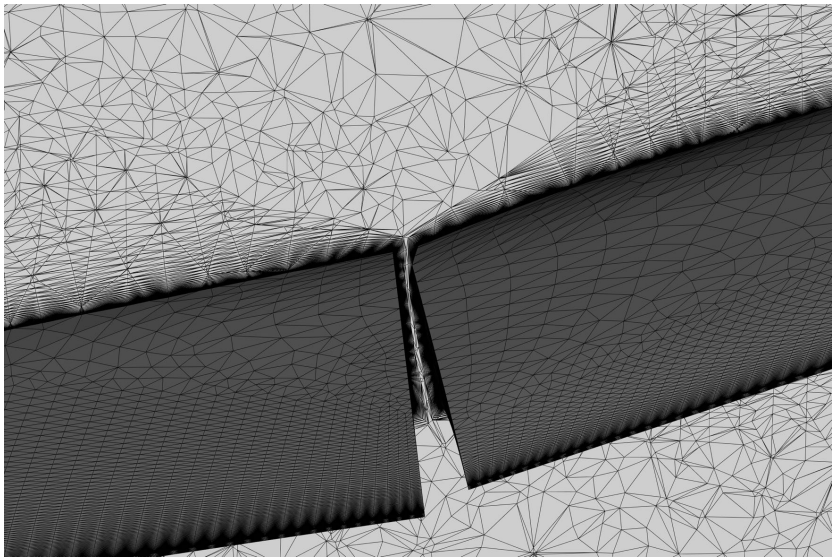


3D AIAA HLPW3 Common Research Model

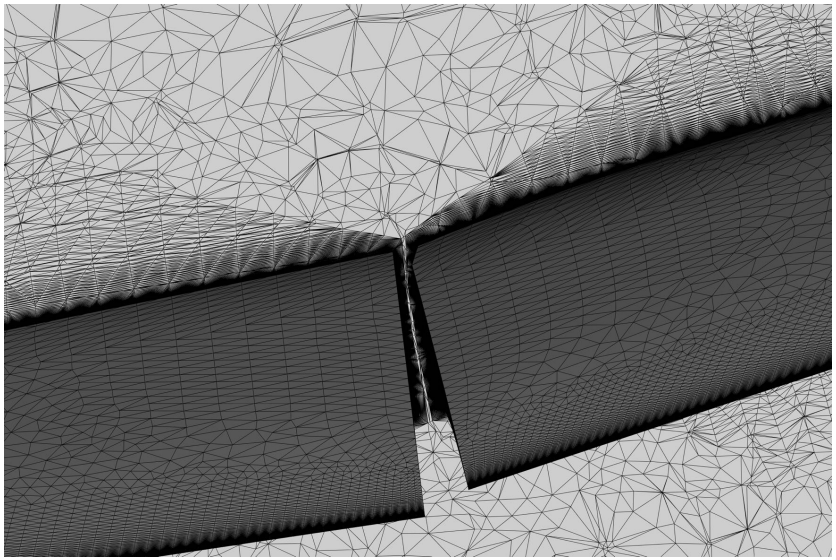


3D AIAA HLPW3 Common Research Model

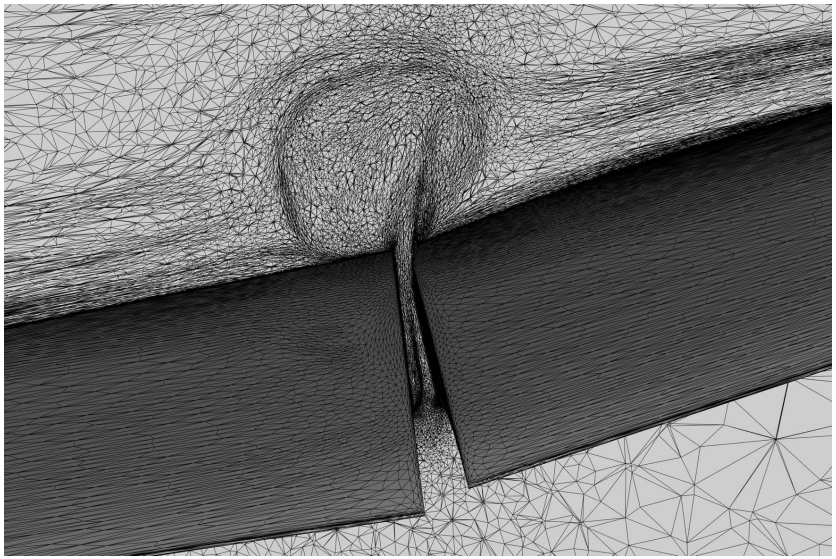
26 491 087 vertices



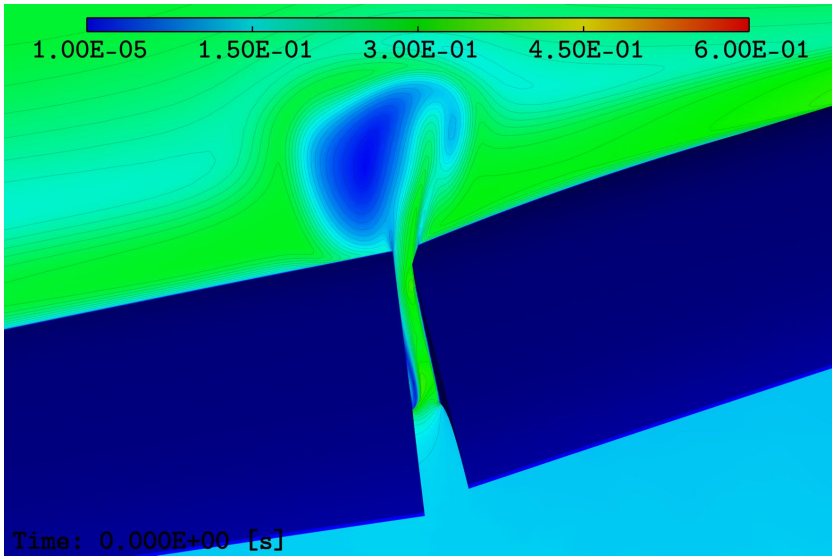
69 890 932 vertices



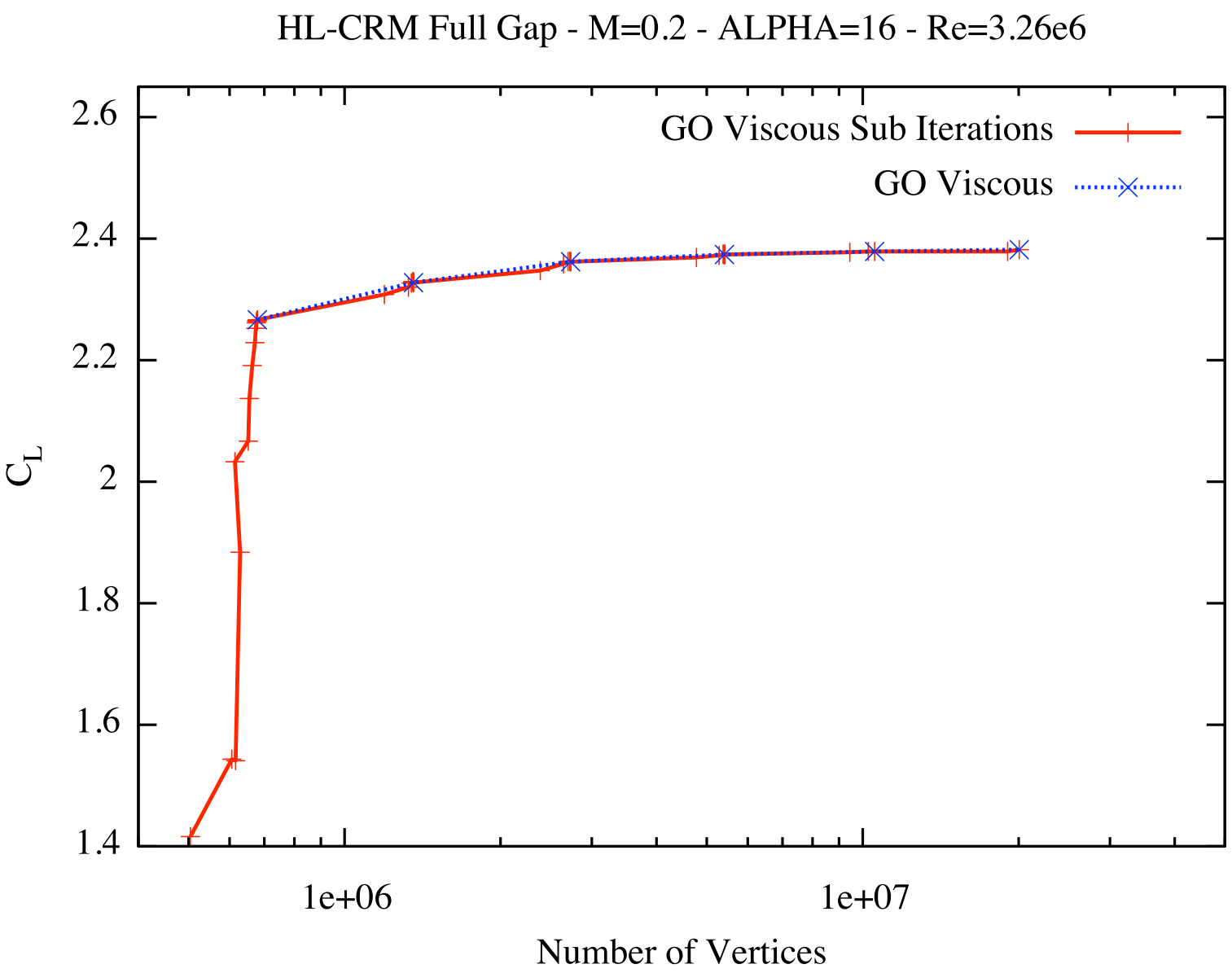
5 318 448 vertices



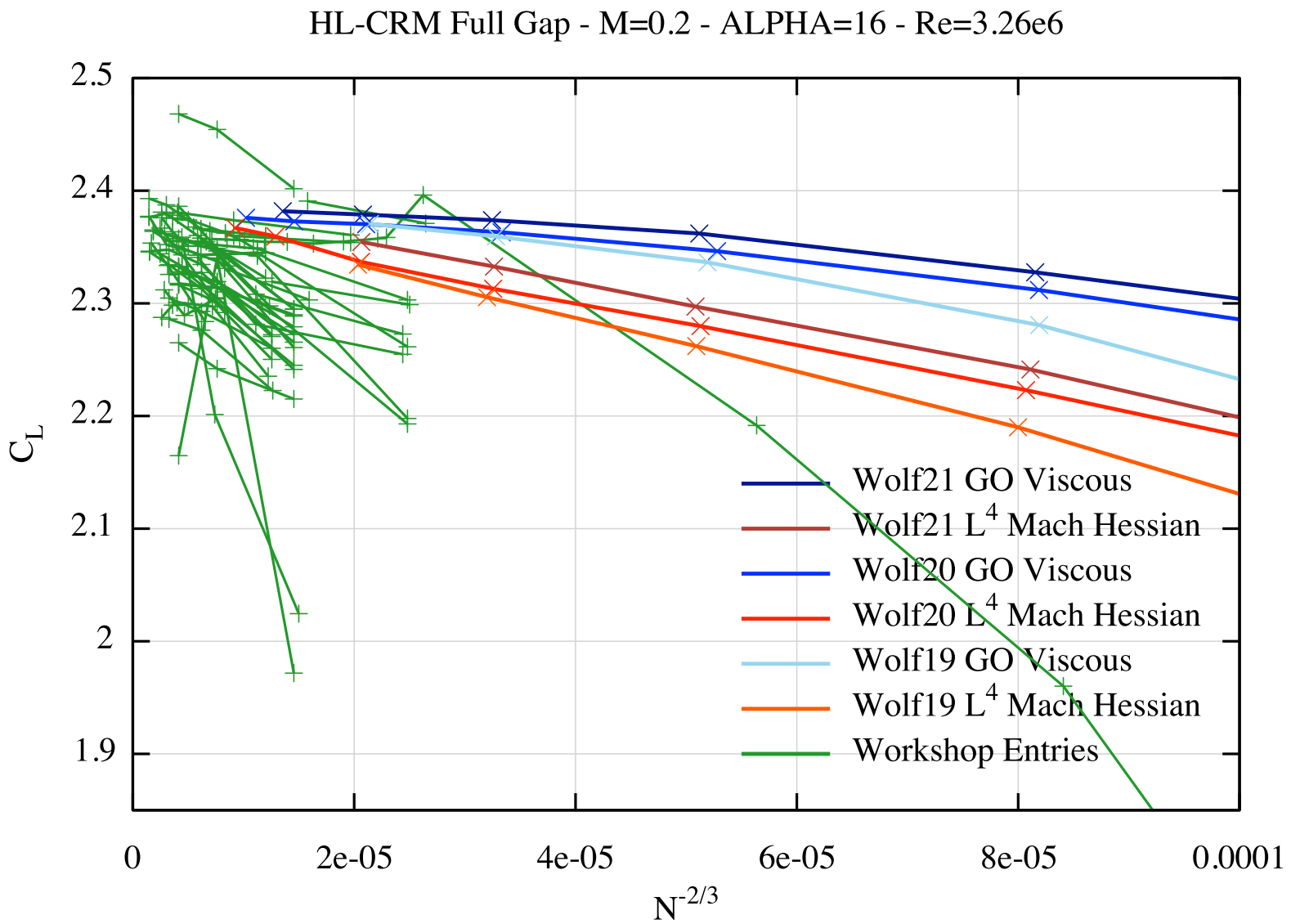
Local Mach Number



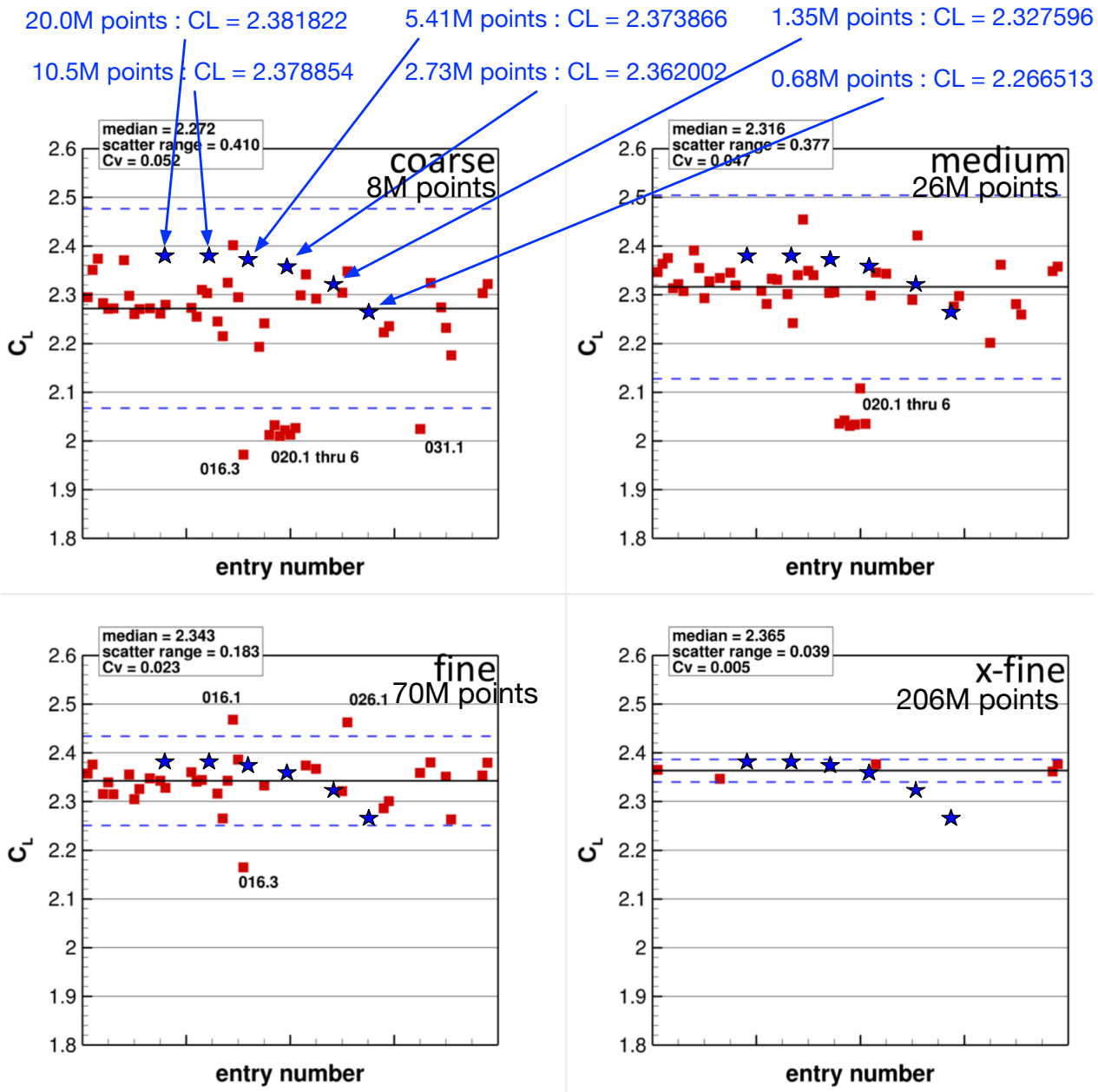
3D AIAA HLPW3 Common Research Model



3D AIAA HLPW3 Common Research Model



3D AIAA HLPW3 Common Research Model



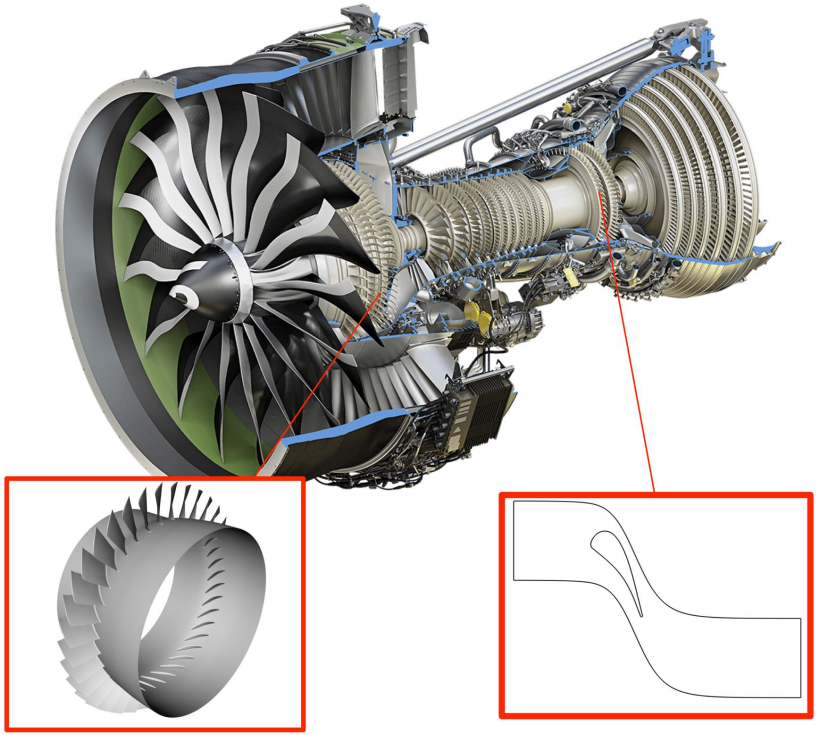
Outline

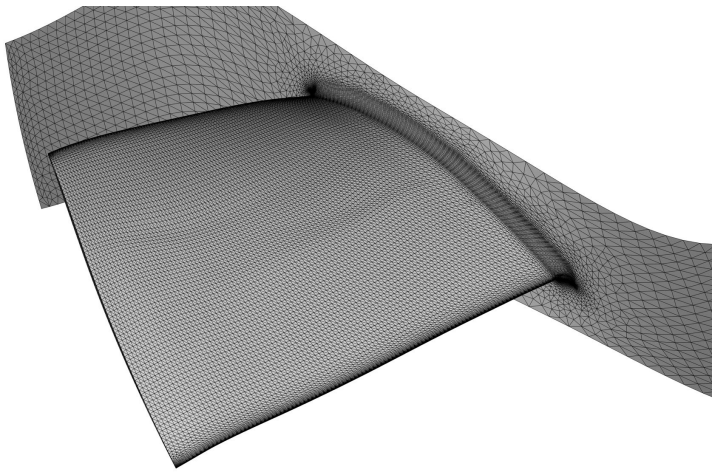
Industrial Context

High-fidelity numerical simulation in turbomachinery:

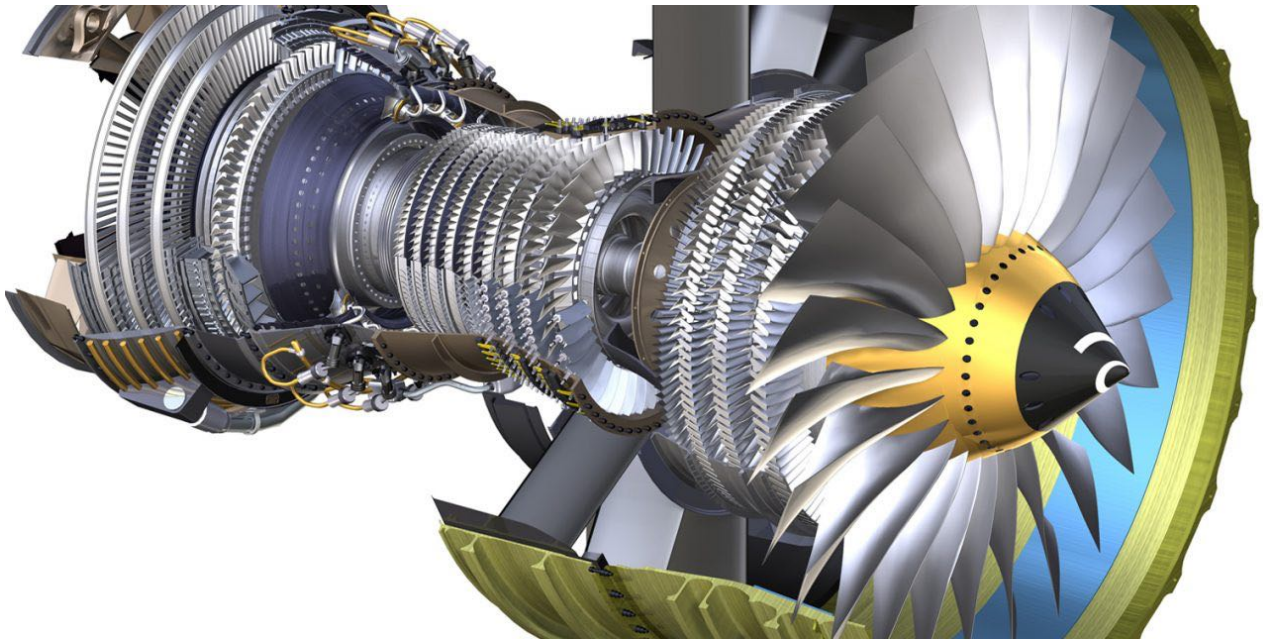
- **Extremely complex geometry:**
  Periodicity, multi-stage,
  many technological effects, ...

- **Extremely complex physics:**
  Shock waves, boundary layers,
  off-body vortices, turbulence,
  thermal effects, ...





For turbomachinery, one of the main difficulties is to take into account the **periodicity**



Metric-Based Local Remeshing

Dealing with the periodicity:

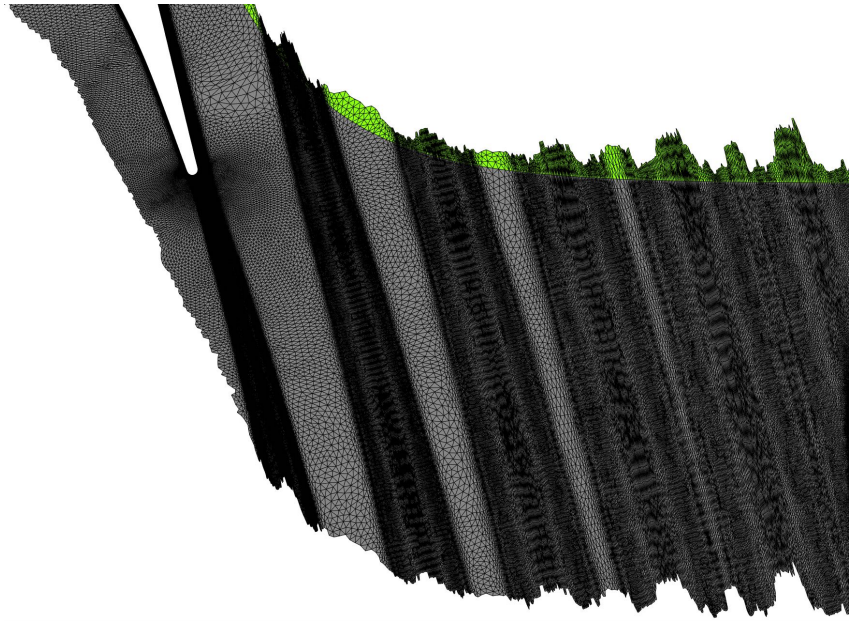
1. Directly manage the periodicity in the local remesher
   Pros: 1 call thus efficient ; preserve the original geometry
   Cons: hard to code (lot of modifications) ; maintenance ; cavity with double constraints

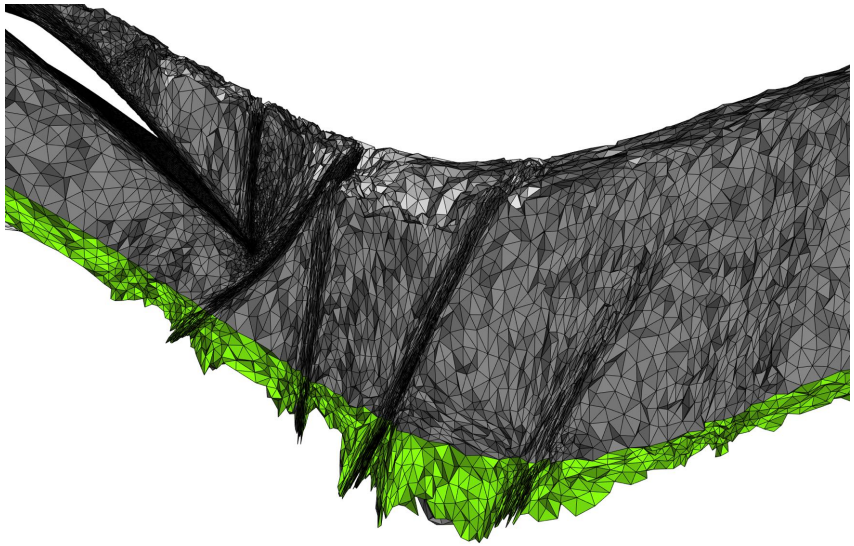
2. Add a migration step and keep the original local remesher [Dobrzynski et al., IJNME 2011]
   Pros: easy, use the same remesher !
   Cons: do not preserve the original geometry ; 2 calls and 1 migration

3. Two migration steps and keep the original local remesher
   Pros: easy, use the same remesher ! ; preserve the original geometry
   Cons: 2 calls and 2 migrations ; requires to manage no-manifold geometries

Dealing with the periodicity:

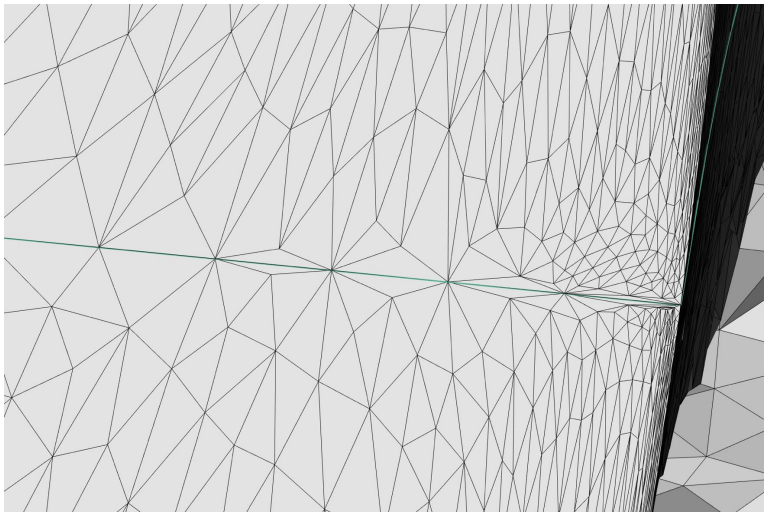
1. Adapt surface and volume mesh leaving periodic boundary unchanged
2. Migrate N layers of elements from one side of the periodic boundary to the other
   Transport the metric field accounting for rotations: $\bar{\mathcal{M}} = R\mathcal{M}R^T$
3. Adapt the new domain with the periodic surface as constraint in the volume
4. Migrate back the elements to recover the initial geometry of the domain

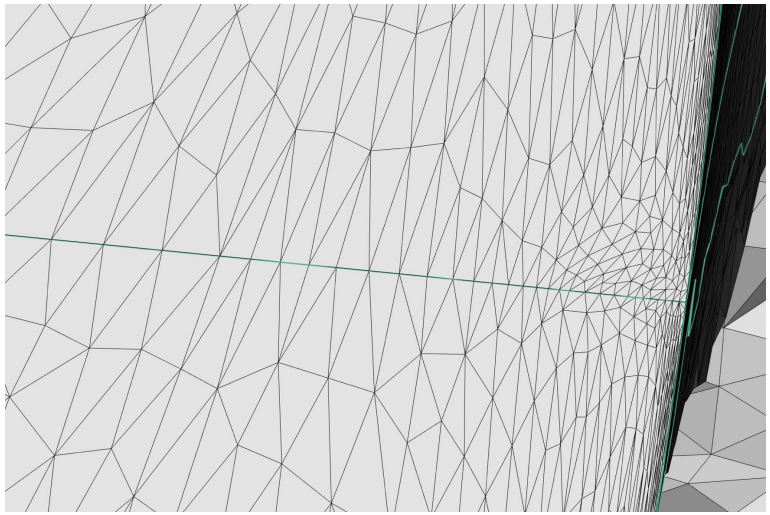




Dealing with the periodicity:

1. Adapt surface and volume mesh leaving periodic boundary unchanged
2. Migrate N layers of elements from one side of the periodic boundary to the other
   Transport the metric field accounting for rotations: $\bar{\mathcal{M}} = R\mathcal{M}R^T$
3. Adapt the new domain with the periodic surface as constraint in the volume
4. Migrate back the elements to recover the initial geometry of the domain





Un Compresseur : NASA Rotor 37

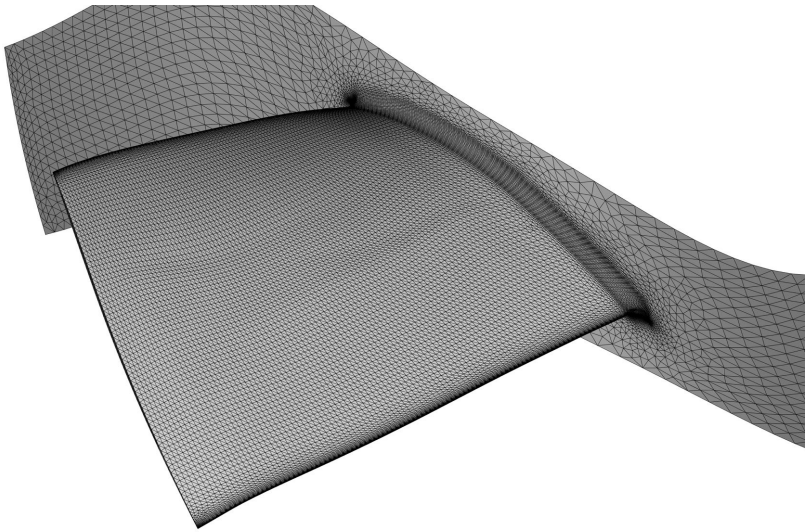
Geometry: NASA Rotor 37 - Cold geometry

Initial mesh: $\#Vertices = 158\,390$ and $\#Tetrahedra = 851\,932$

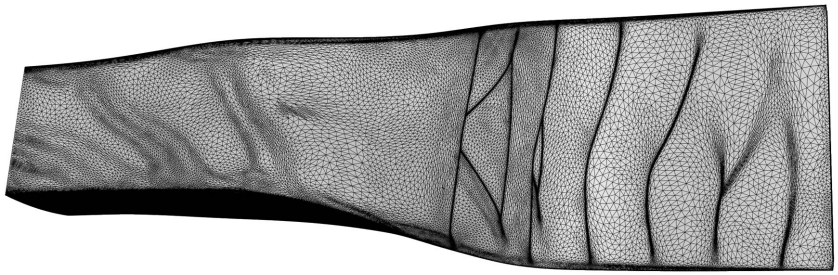
Flow characteristics:

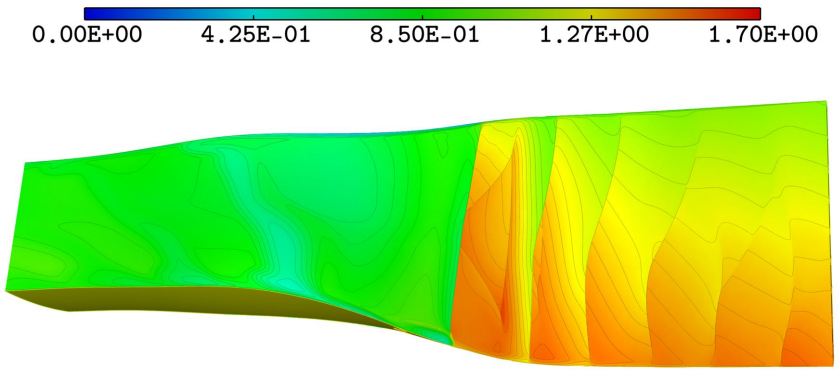
- Reference state: $\rho^{ref} = 1.225\ kg\ m^{-3}$, $\mu^{ref} = 1.789 \times 10^{-5}\ kg\ m^{-1}\ s^{-1}$, $q^{ref} = 400\ m\ s^{-1}$
- Inlet: $P_t^{in} = 1.01325 \times 10^5$ Pa, $T_t^{in} = 288.15$ K
- Outlet: $P^{out} = \beta P_t^{in}$ with at least 16 functioning points

  $\beta \in \{1, 1.025, 1.05, 1.075, 1.1, 1.125, 1.15, 1.175, 1.2, 1.225, 1.25, 1.26, 1.27, 1.275, 1.28, 1.29\}$
- Rotation speed: 1800 rad/s, sector: 10°

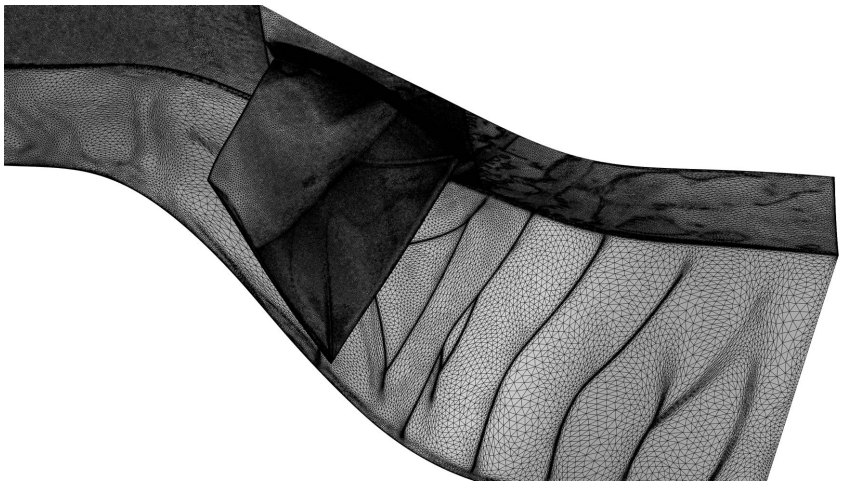


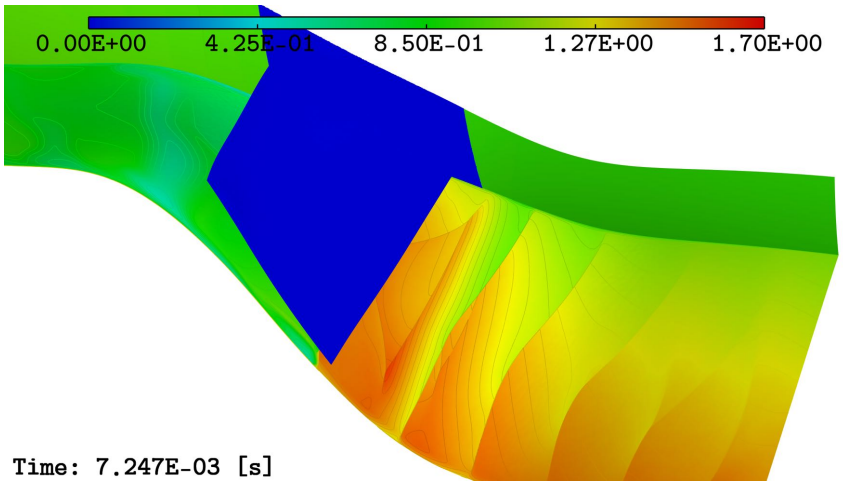
Adaptation of the periodic surfaces:





Time: 7.247E-03 [s]

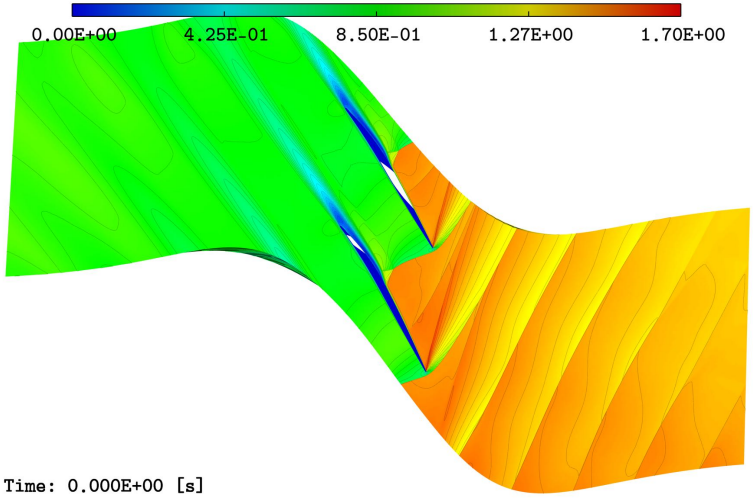


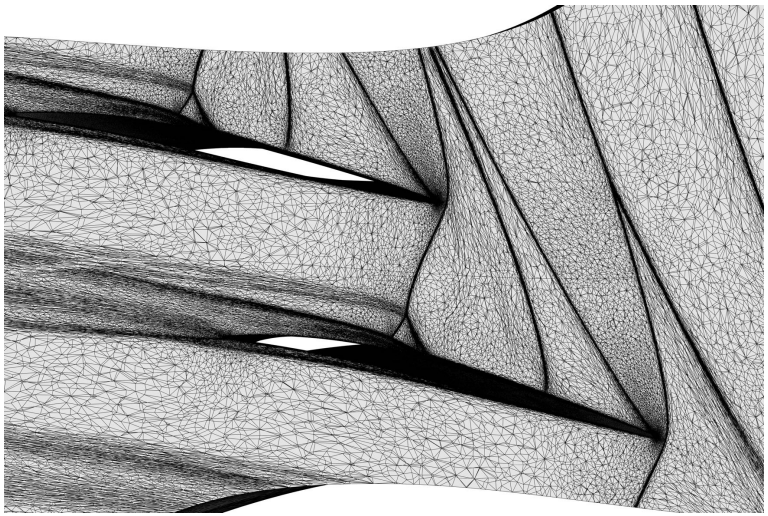


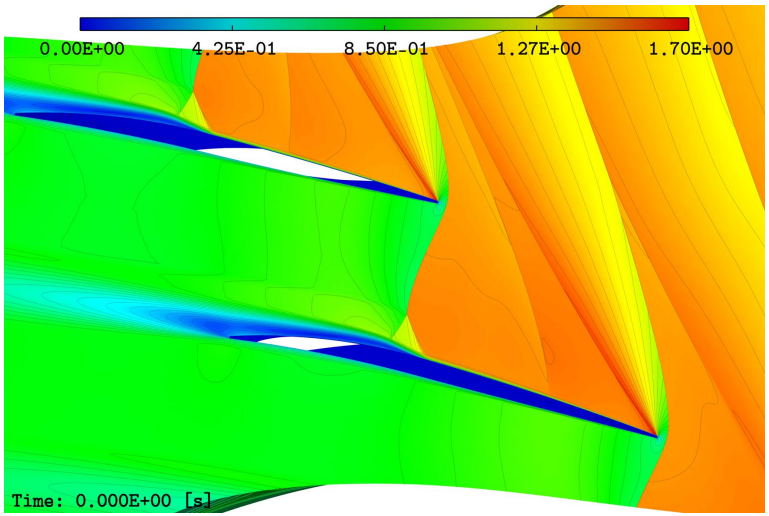
Time: 7.247E-03 [s]

Adapted mesh close to the tipwall:

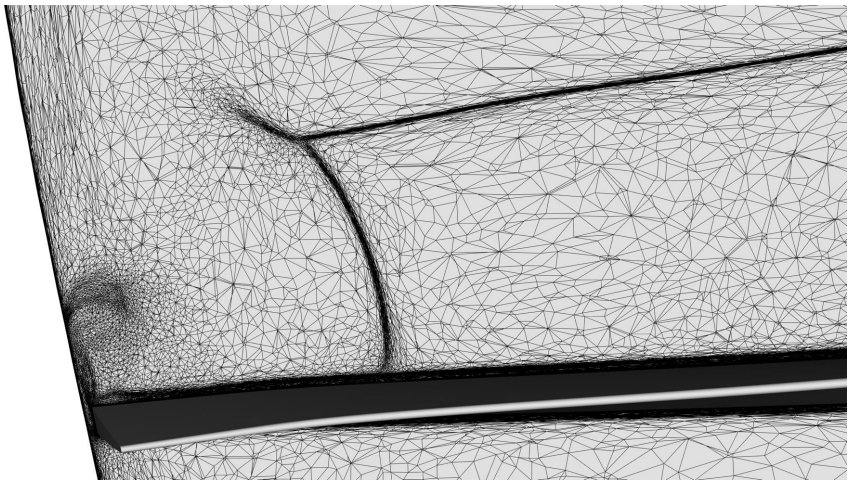
(domain is duplicated to highlight the effectiveness of the periodic adaptation)

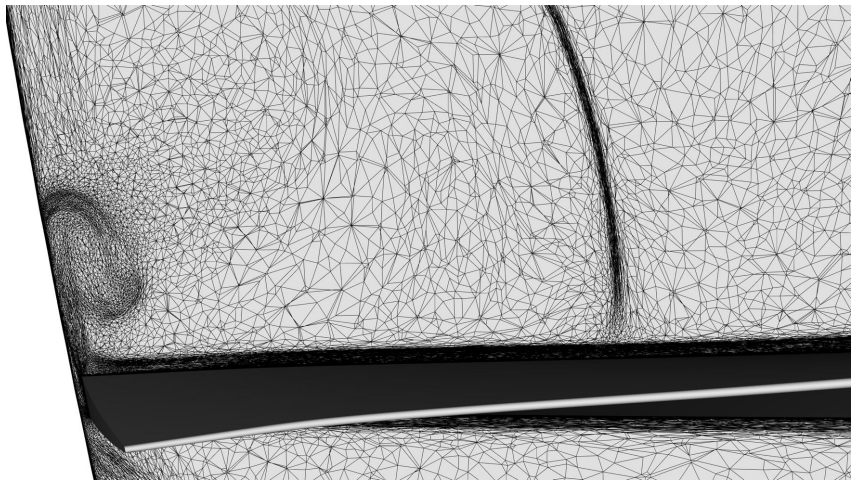


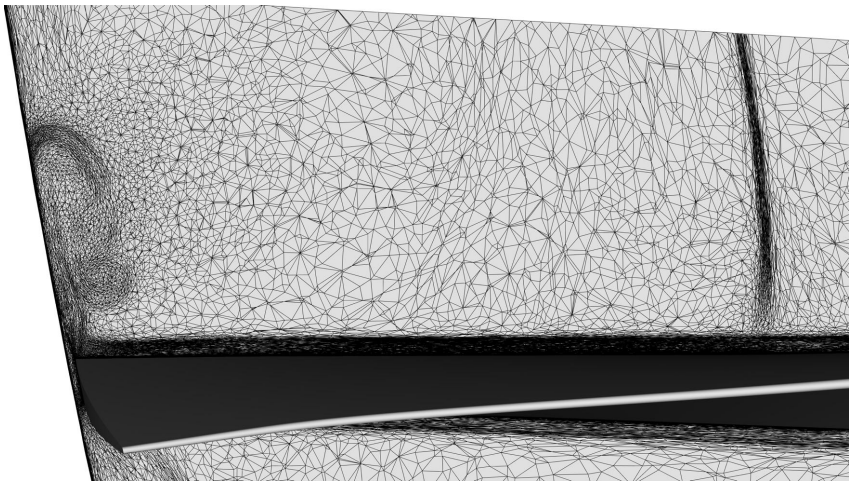


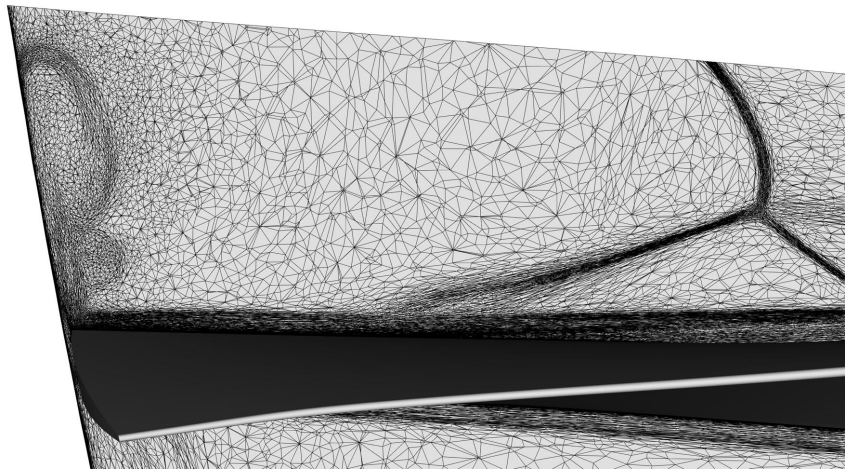


Adaptation to the tip vortices:

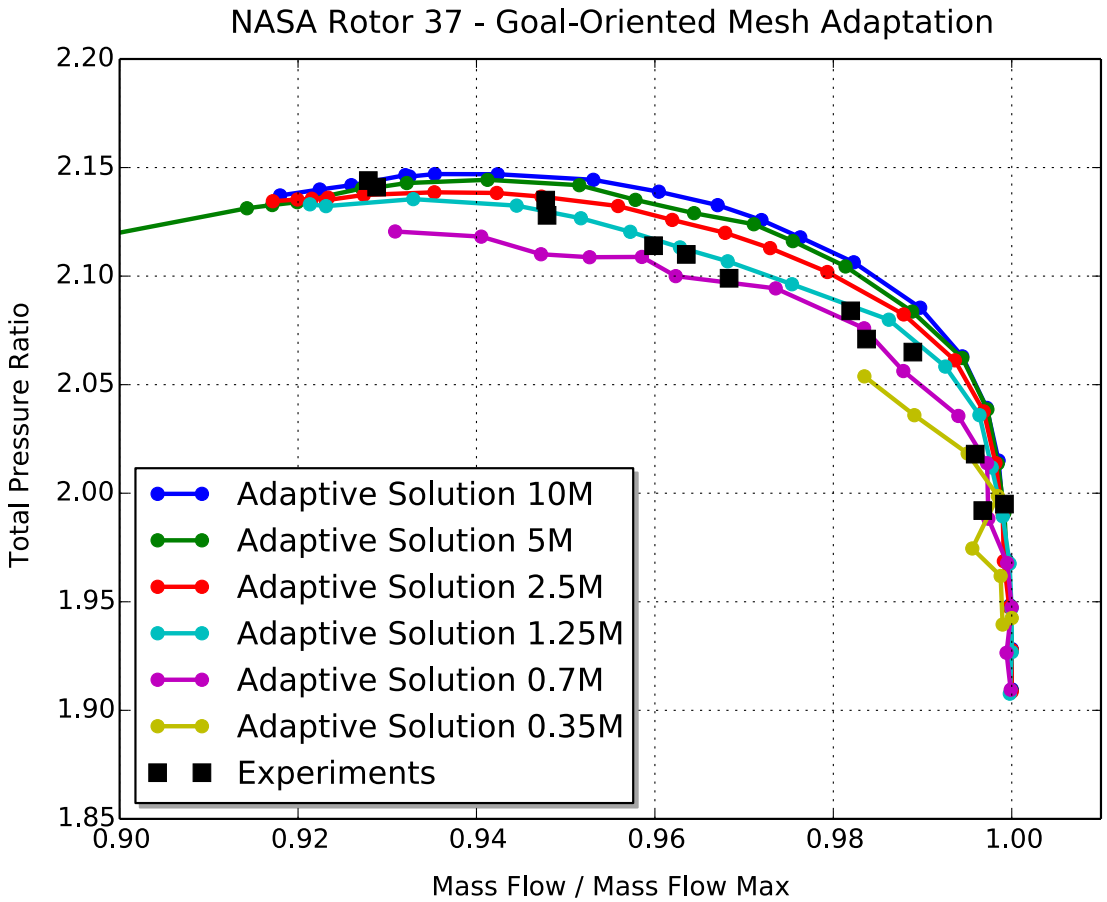








Mesh convergence of the pressure ratio characteristic:



Outline

Conclusions

- High-fidelity turbulent flow predictions are obtained with unstructured meshes composed **only of tetrahedra**

Simplify and automate the mesh generation process on complex geometries
Remove the human from the loop

- **Mesh-converged solutions** are achieved in 3D thanks to anisotropic mesh adaptation

Discretization error does not impact anymore the prediction
Provides verification results (accurate solution to the physical model)

- Results independent of the initial mesh

No boundary layer (BL). No meshing guidelines. Fully automatic (no user intervention)

Current work and perspectives

- Improved numerical scheme
- Improved turbulence model (notably RANS)
- Error quantification and results consistency
- Unsteady flows and moving geometries
- High-order meshes, error estimate and solvers

Thank you for your Attention