Embedded and high-order meshes : two alternatives to linear body-fitted meshes

Rémi Feuillet

Unité de mathématiques appliquées, ENSTA-CNRS-INRIA
Équipe-projet GAMMA, INRIA

10 December 2019

Supervisors: Frédéric Alauzet, Patrick Ciarlet & Adrien Loseille

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Fondation mathématique FMJH Jacques Hadamard

informatiques mathématiques Inria

ENSTA UMA

ENSTA IP PARIS

Introduction

The main topic of this thesis is about meshes

What is it ?

- A set of elementary entities, like triangles, quadrilaterals, tetrahedra, hexahedra, ...
- A discrete representation of an object on a computer

What for ?

- Two main applications
- Computer graphics
  $\Longrightarrow$ Use for animated movies, computer games, ...
- Engineering
  $\Longrightarrow$ Use for numerical computations, engineering design, ...

Introduction

The main topic of this thesis is about meshes

What is it ?

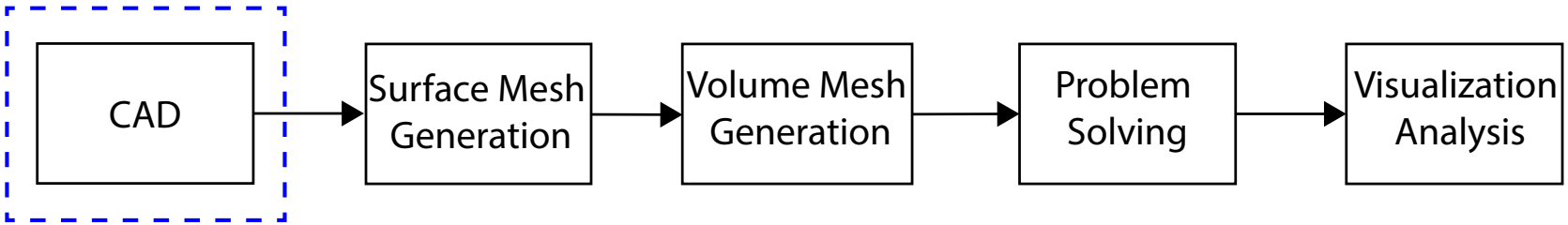
- A set of elementary entities, like triangles, quadrilaterals, tetrahedra, hexahedra, ...
- A discrete representation of an object on a computer

What for ?

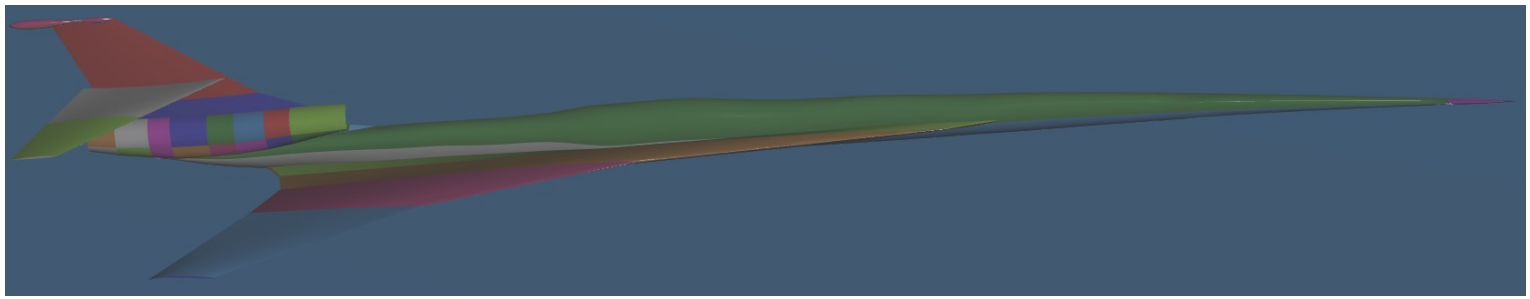
- Two main applications
- Computer graphics
  $\Longrightarrow$ Use for animated movies, computer games, ...
- Engineering
  $\Longrightarrow$ Use for numerical computations, engineering design, ...

Introduction

In a numerical simulation, mesh generation is part of the computational pipeline

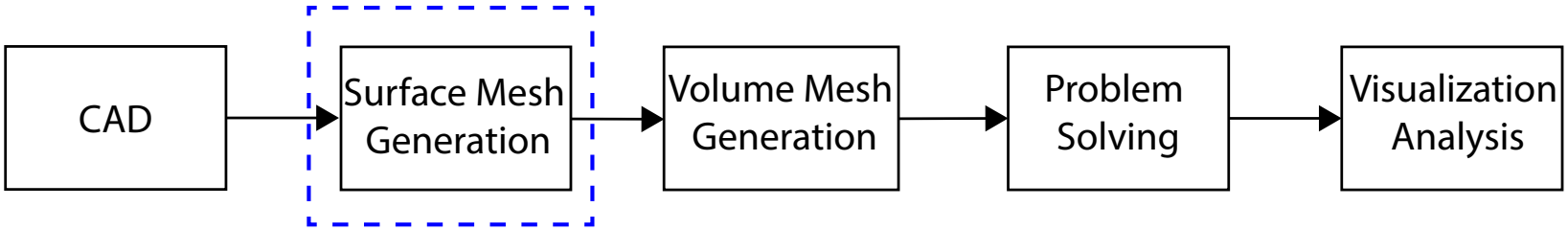


- The studied geometry is defined using a CAD model

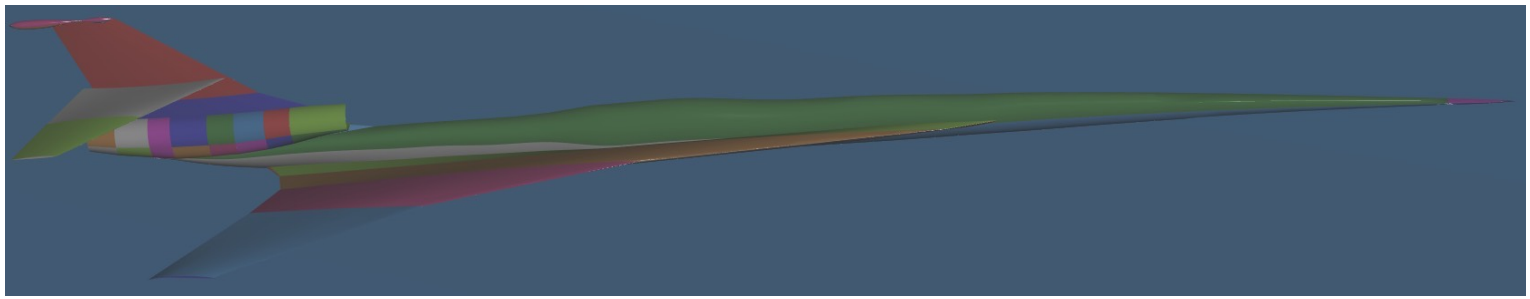


Introduction

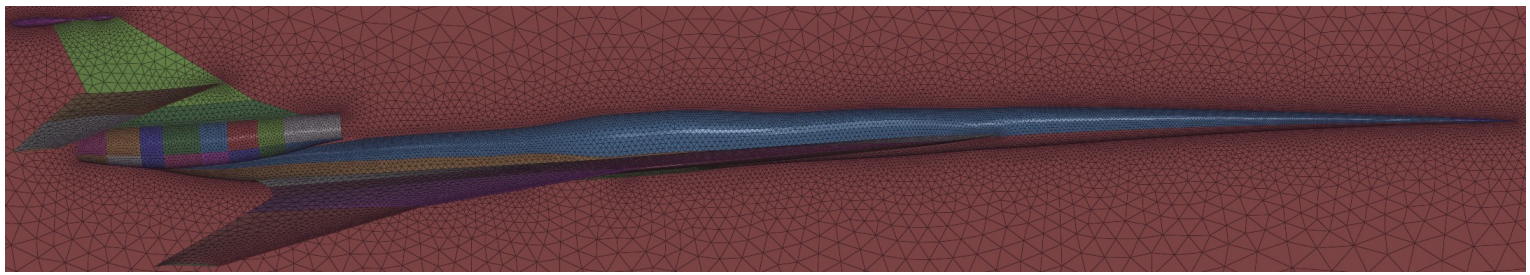
In a numerical simulation, mesh generation is part of the computational pipeline



- The studied geometry is defined using a CAD model

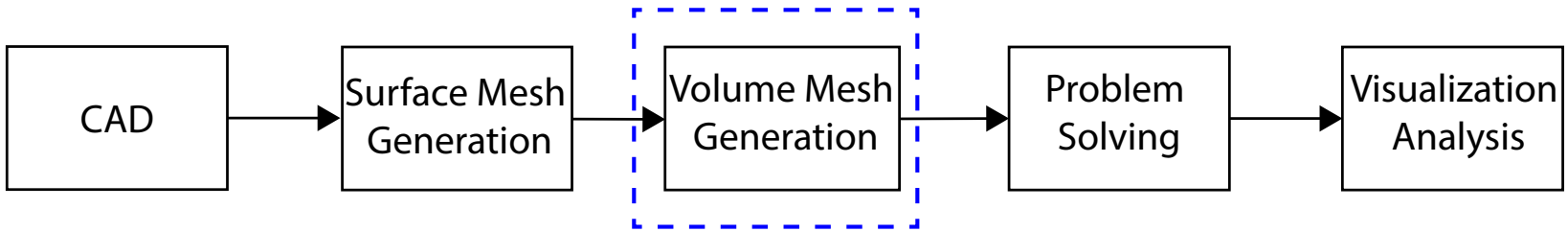


- A surface mesh is generated from the CAD

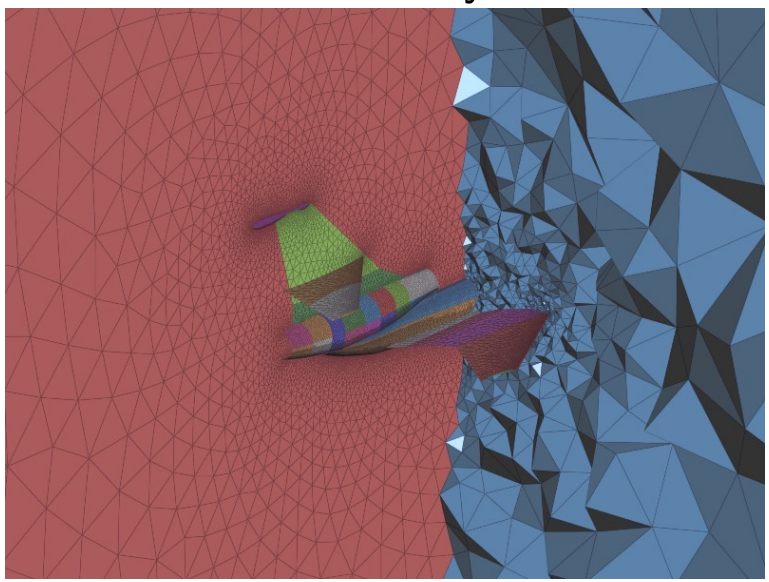


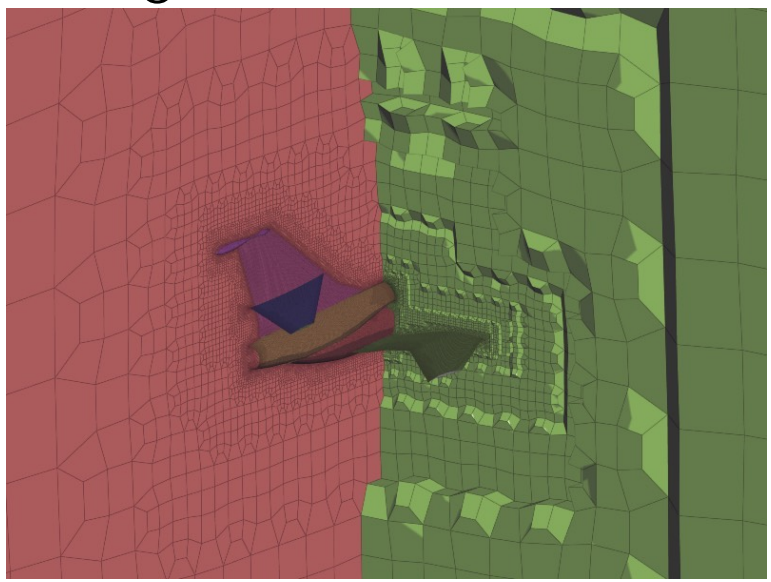
Introduction

In a numerical simulation, mesh generation is part of the computational pipeline



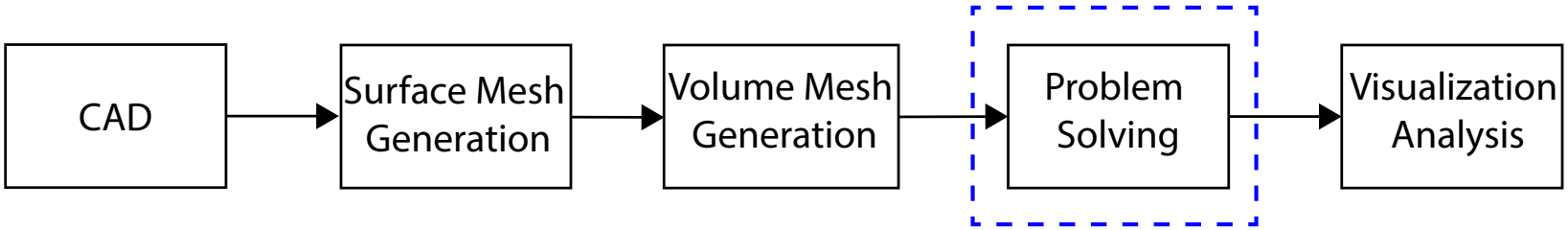
- The surface mesh may be used as input to generate a volume mesh



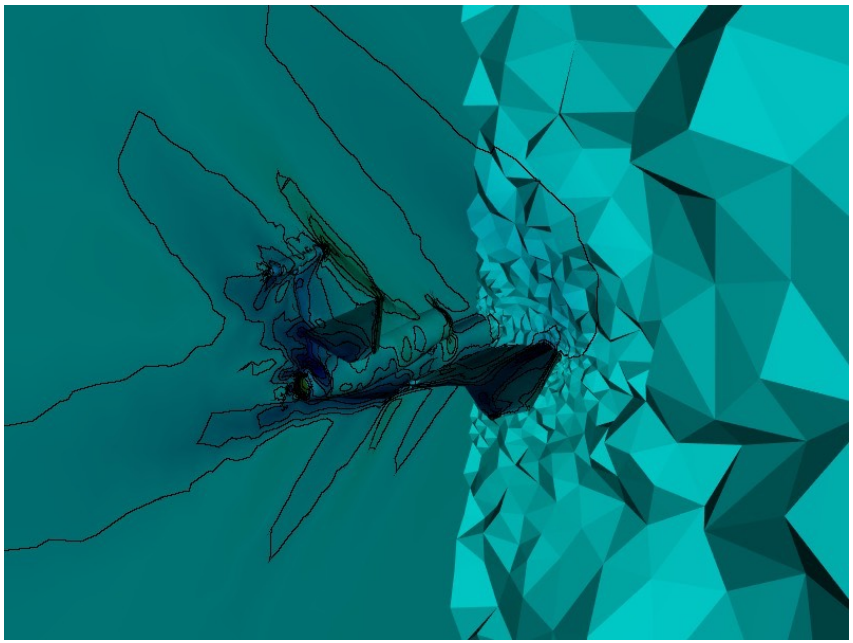


Introduction

In a numerical simulation, mesh generation is part of the computational pipeline



- A numerical computation is then performed on the volume mesh



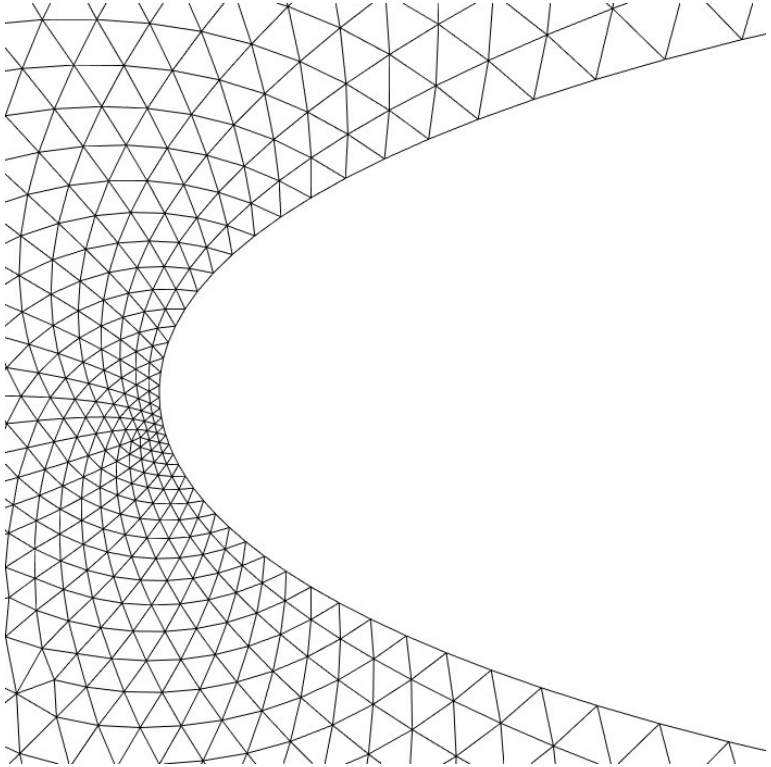
Introduction

How do we represent the geometry ?

Introduction

How do we represent the geometry ?

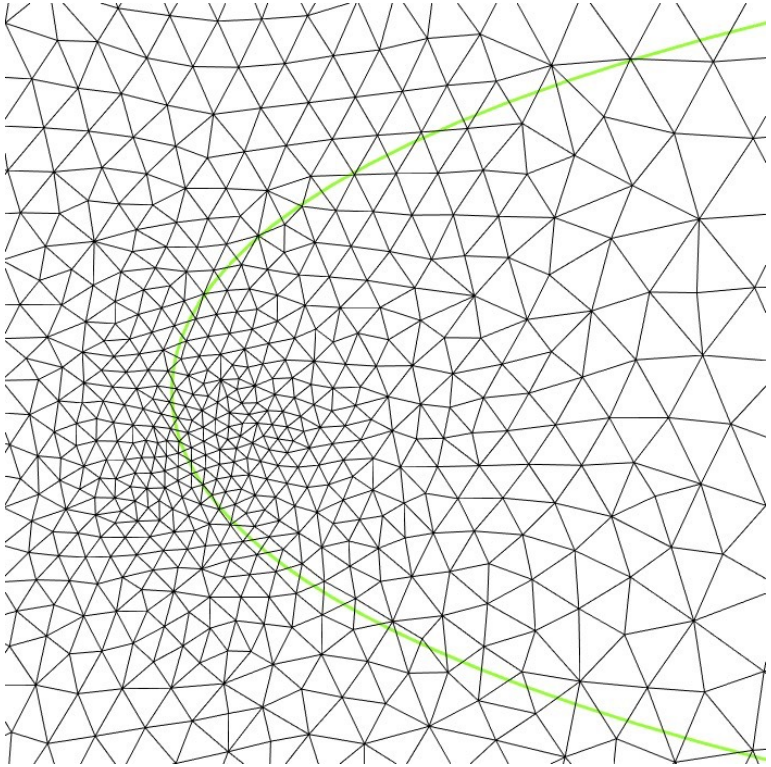
- Linear body-fitted mesh



Introduction

How do we represent the geometry ?

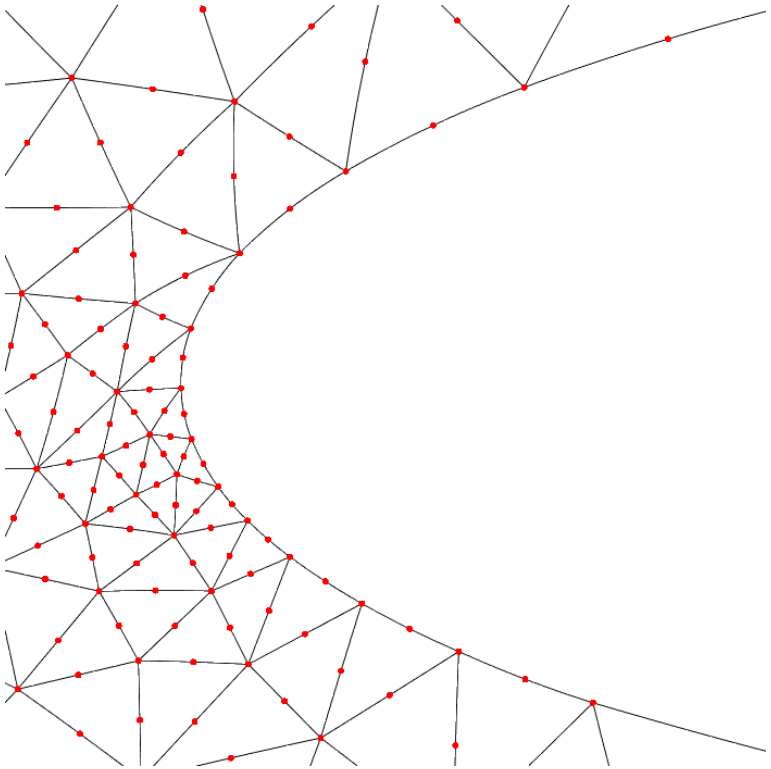
- Linear body-fitted mesh
- Embedded mesh



Introduction

How do we represent the geometry ?

- Linear body-fitted mesh
- Embedded mesh
- High-order body-fitted mesh

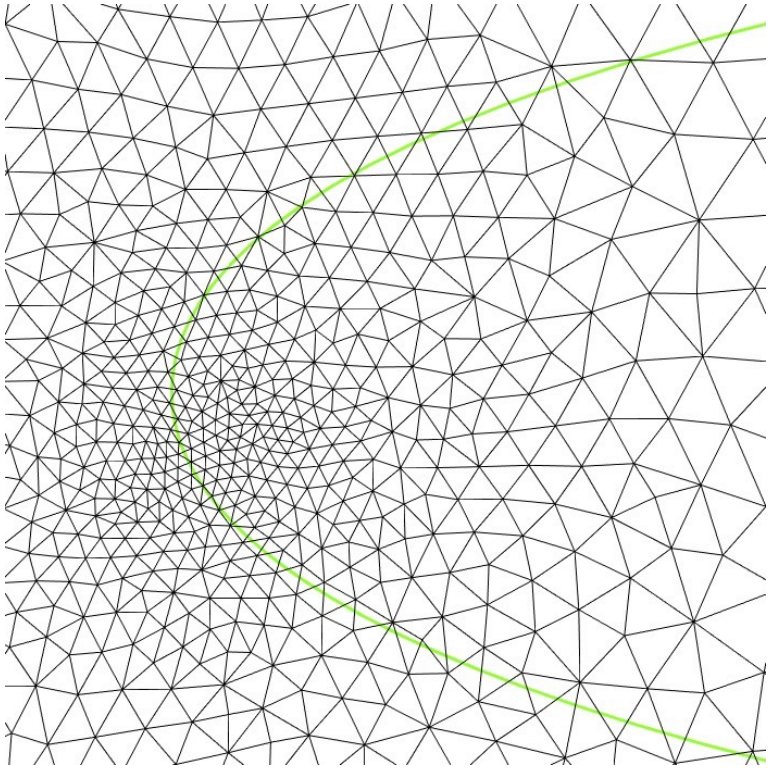


Introduction

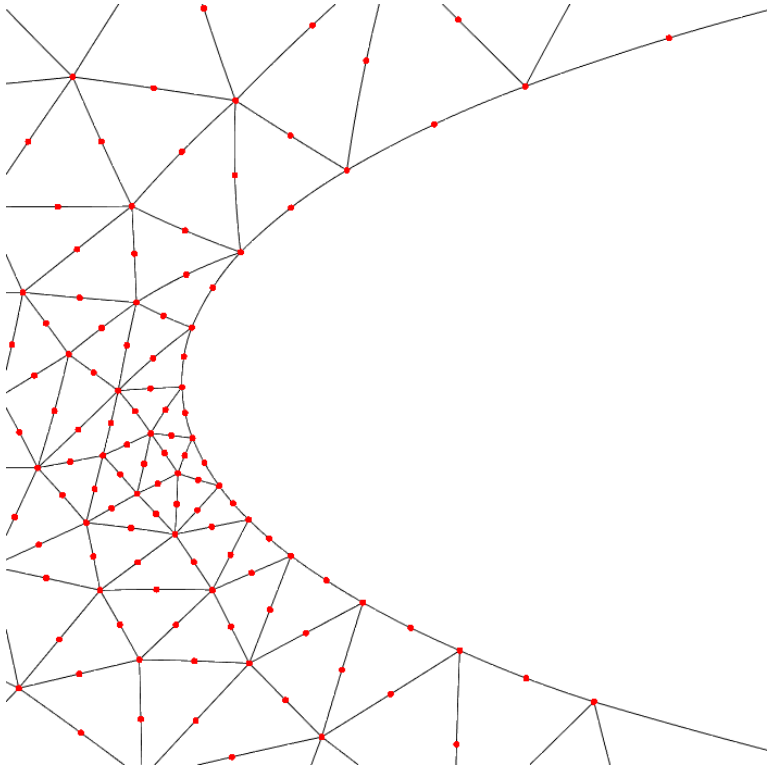
How do we represent the geometry ?

- Linear body-fitted mesh
- Embedded mesh
- High-order body-fitted mesh

The aim of this thesis is to study the last two approaches

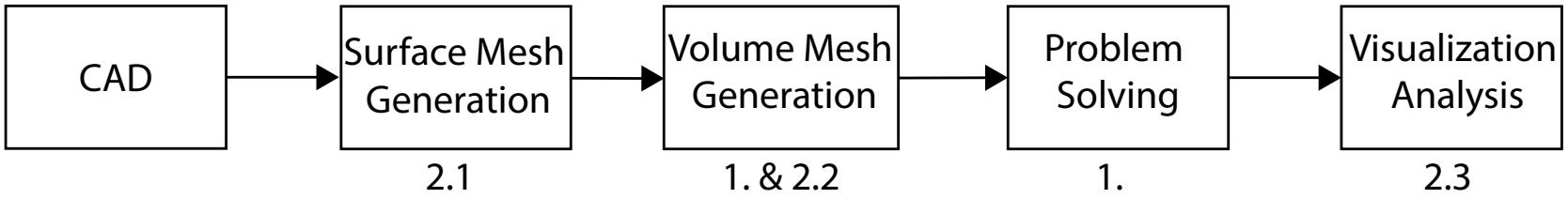


Embedded



High-order

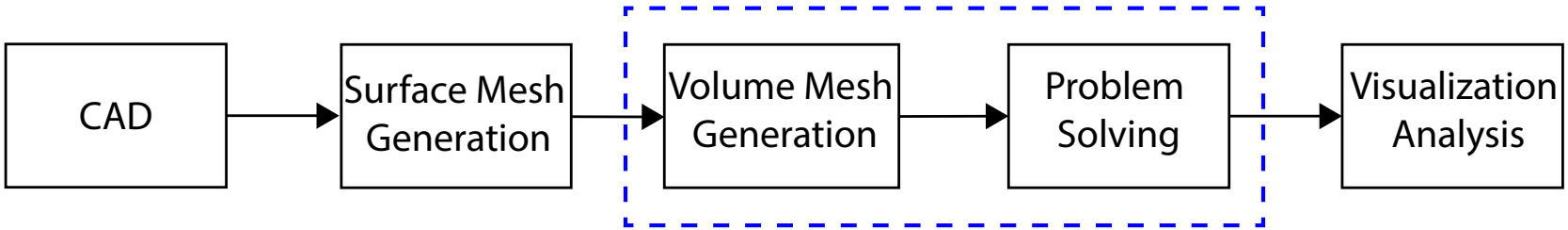
Outline



1. Mesh adaptation for the embedded boundary method

2. High-order
   - 2.1 High-order surface meshing
   - 2.2 Optimization of quadratic meshes and applications
   - 2.3 Visualization of high-order meshes and solutions

Outline

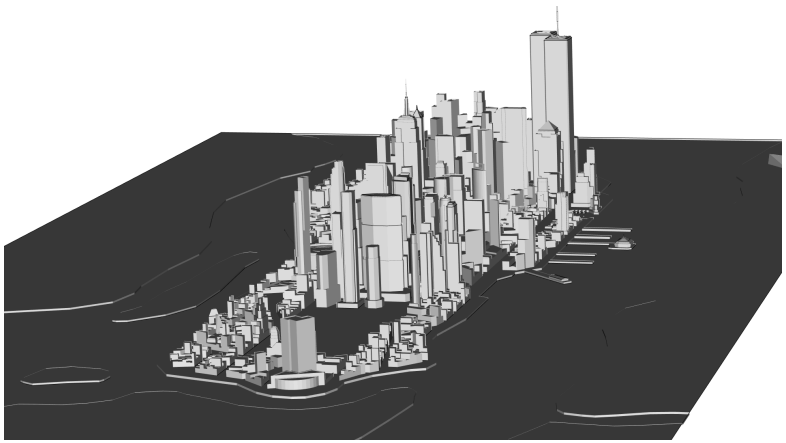


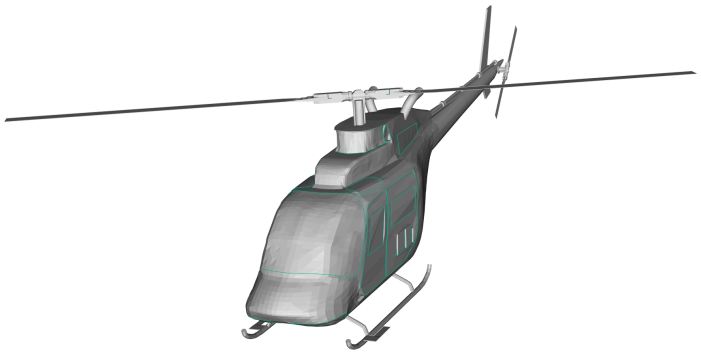
Mesh adaptation for the embedded boundary method

Embedded boundary method: motivations

Motivations

- CAD systems provide dirty and nonconforming geometries
- Some geometries are only implicitly described
- Aim at reducing preprocessing time to cleanup them





1 to 3 months of preprocessing are mandatory in these two cases

$\Longrightarrow$ The embedded boundary method avoids these issues [Yang et al, Aeronautical Journal, 1997], [Löhner et al, IJNME, 2004], [Quan et al, EWC, 2014]

Embedded boundary method: implementation

Input

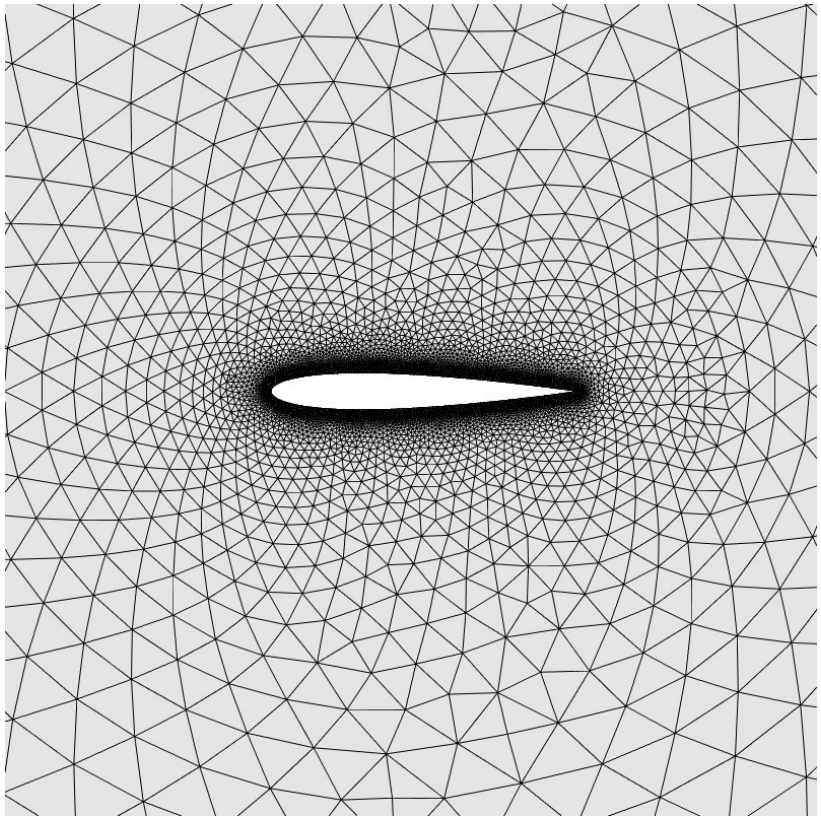
- Linear surface mesh, not necessarily conform → Embedded geometry
- Linear volume mesh → Computational mesh
- Settings of the flow solver (Euler equations, inviscid flow)

Strategy (2D and 3D)

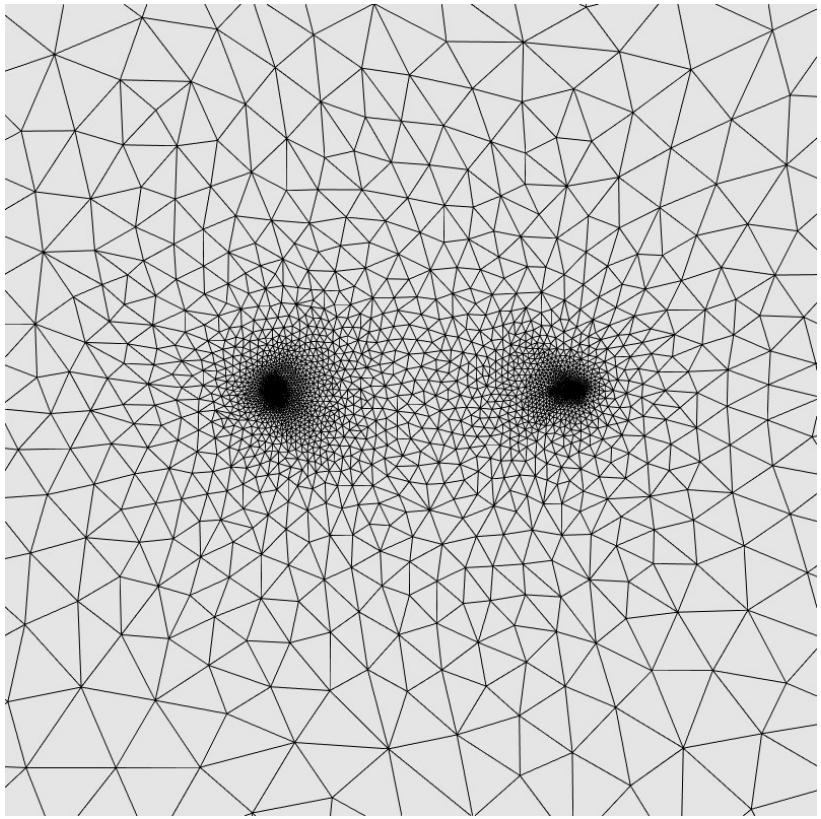
- Localize vertices of the embedded geometry inside the computational mesh
- Solve the intersection of the embedded geometry with the computational mesh
- Set an implicit slipping boundary condition that simulates the presence of the object using the intersection result
- Set a constant value to the vertices covered by the geometry (if any)

Embedded boundary method: results

Case of an embedded supersonic NACA0012 airfoil.



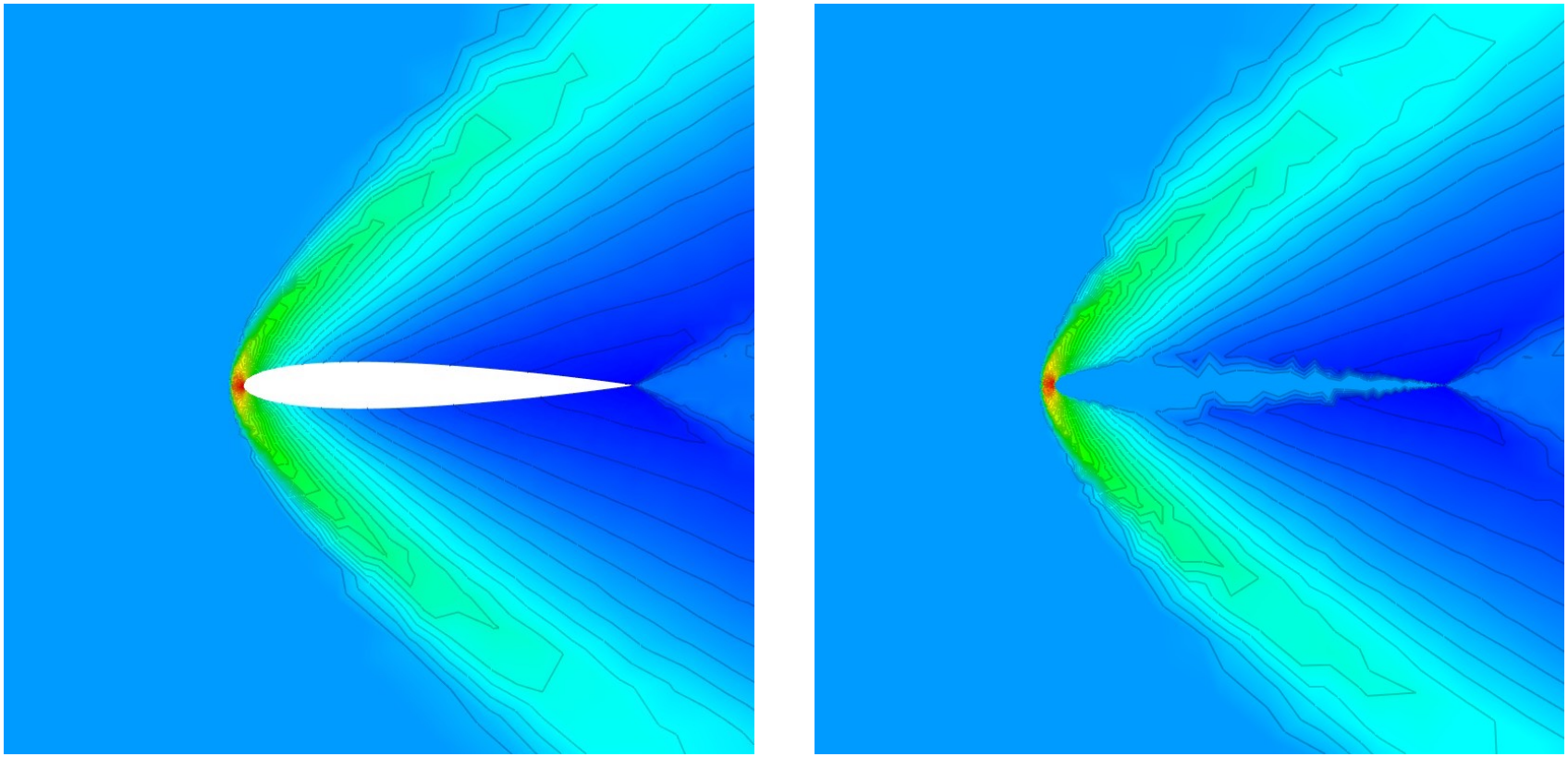
5041 vertices



4812 vertices

Embedded boundary method: results

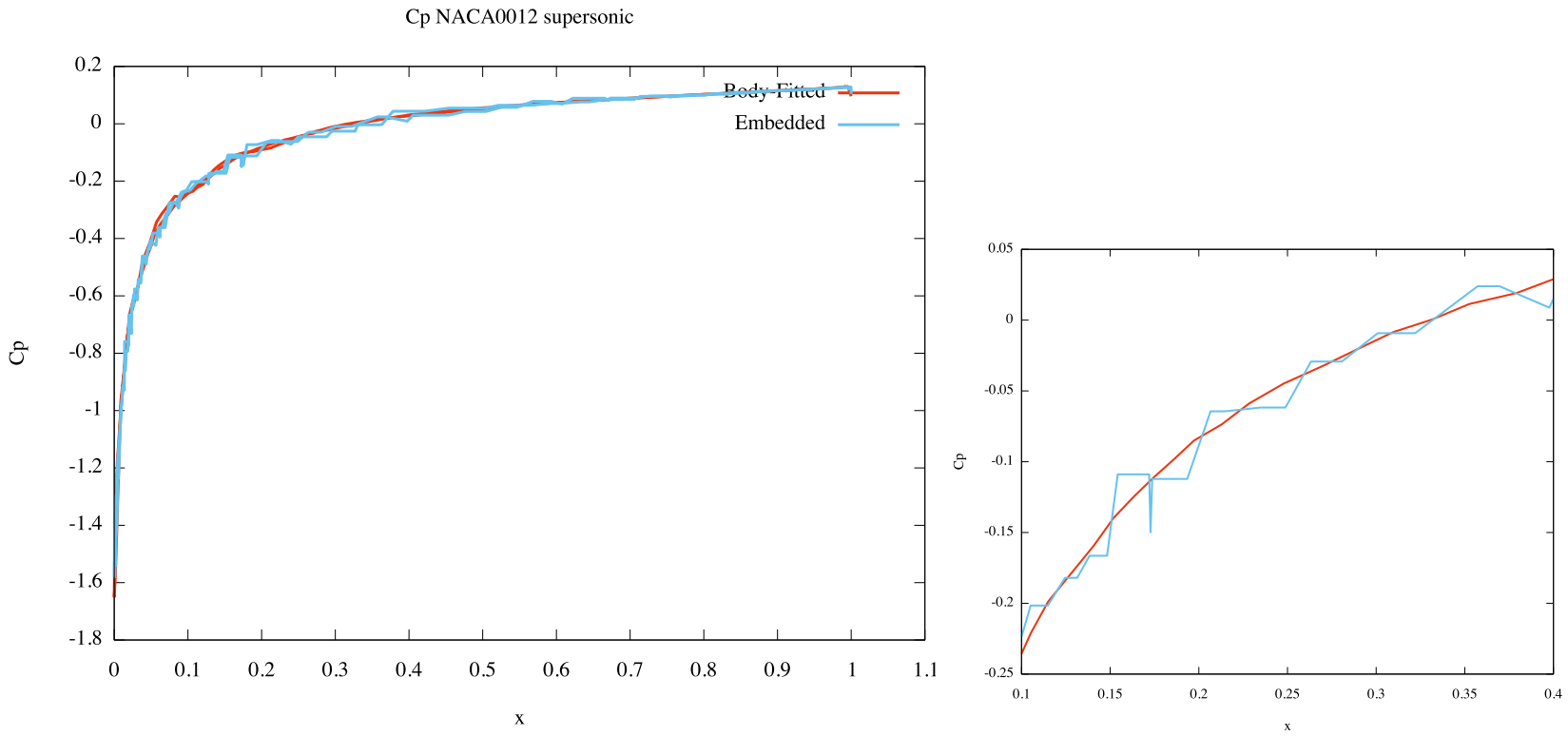
Case of an embedded supersonic NACA0012 airfoil.



Density flow

Embedded boundary method: results

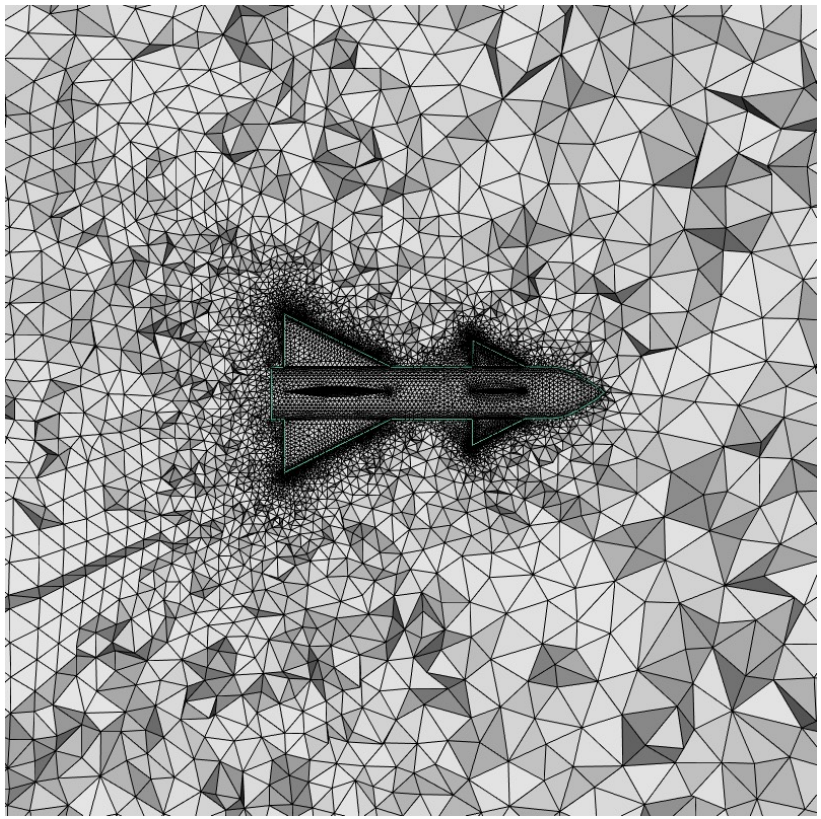
Case of an embedded supersonic NACA0012 airfoil.



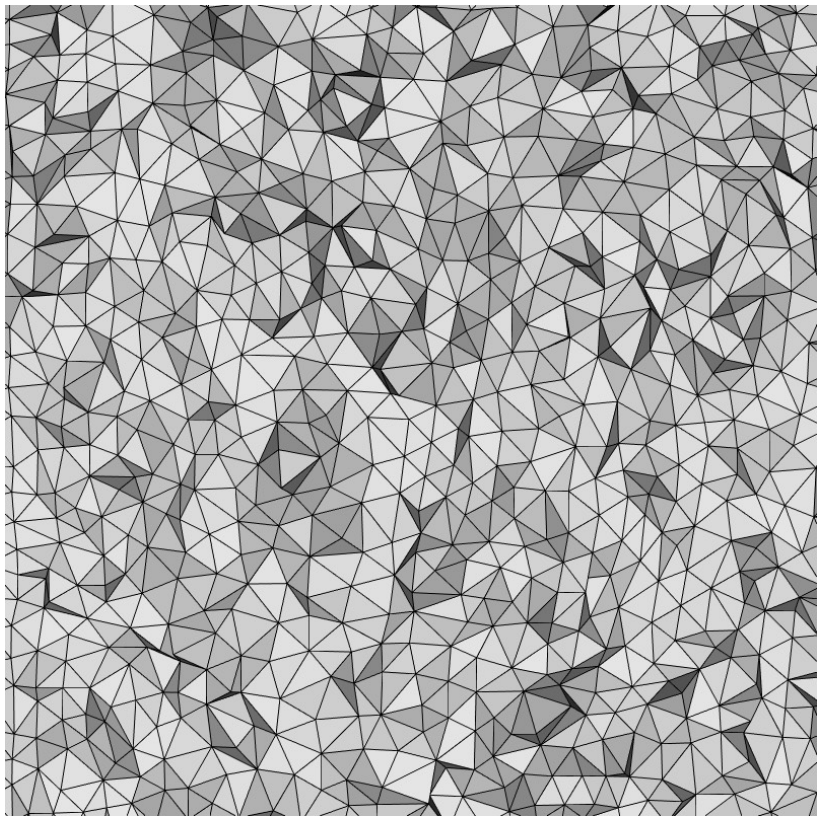
Pressure coefficient

Embedded boundary method: results

Case of an embedded supersonic missile



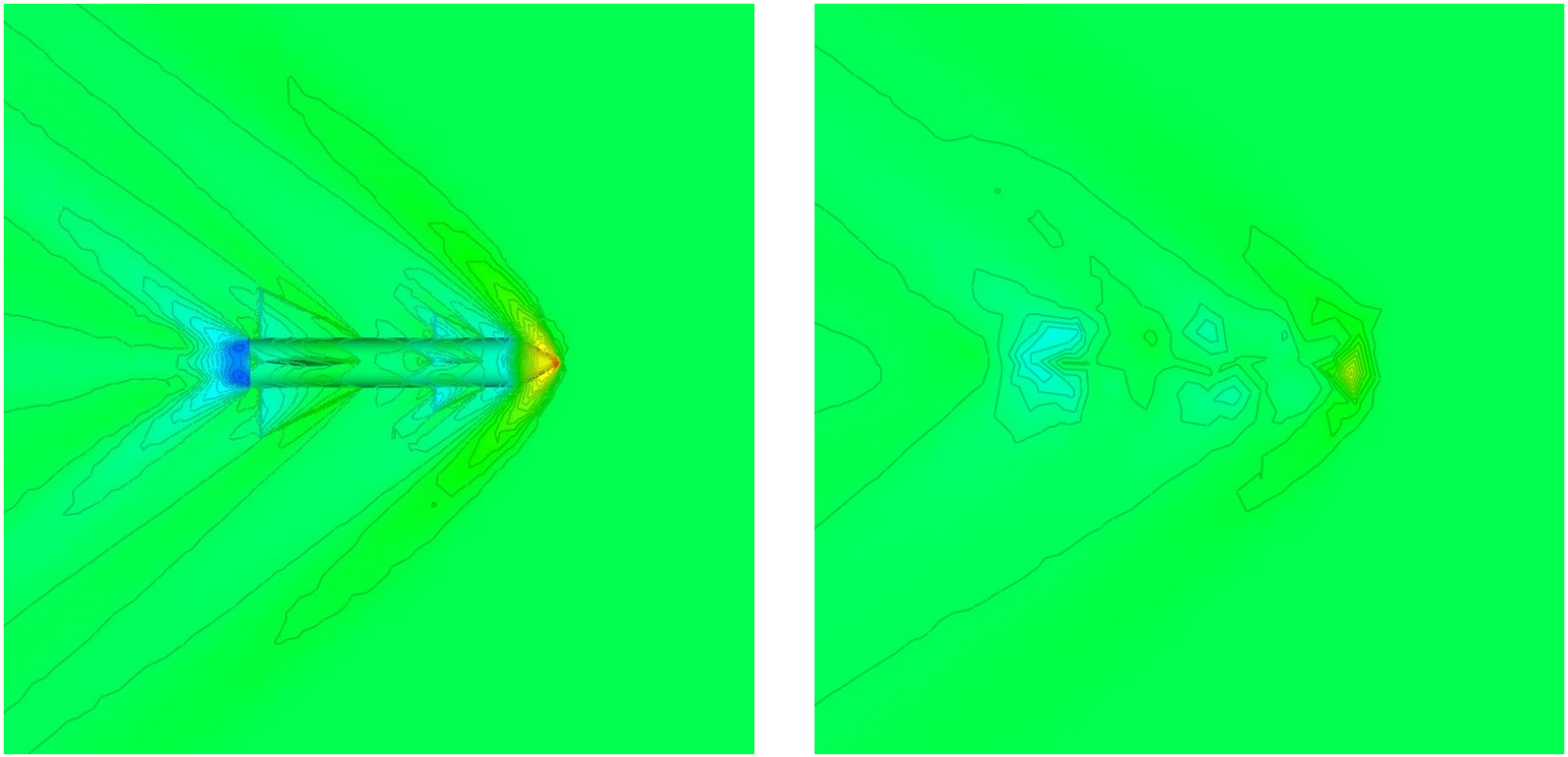
784 847 vertices



90 078 vertices

Embedded boundary method: results

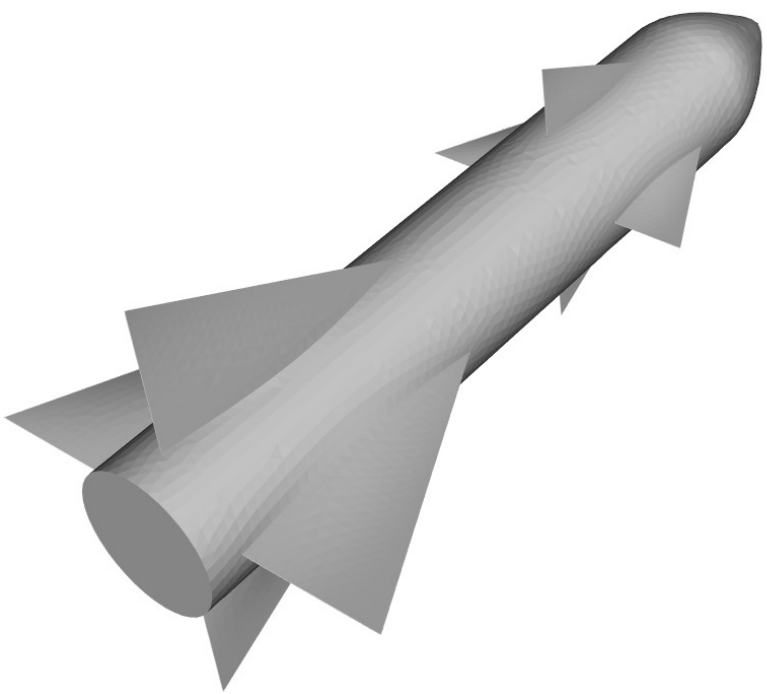
Case of an embedded supersonic missile



Density flow

Embedded boundary method: results

Case of an embedded supersonic missile



Body-fitted solver

Embedded solver

Geometry detected by the solver

Embedded boundary method: results

Observations

- Embedded solution is far less accurate than the body-fitted one
- Due to a coarse detection of the geometry by the flow solver

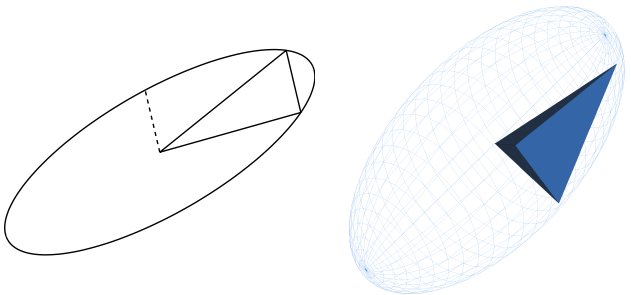
Idea

- Boundary conditions make naturally appear variations of the solution in the vicinity of the embedded boundary
- Perform feature-based mesh adaptation to retrieve the geometry and thus the accuracy of the solution

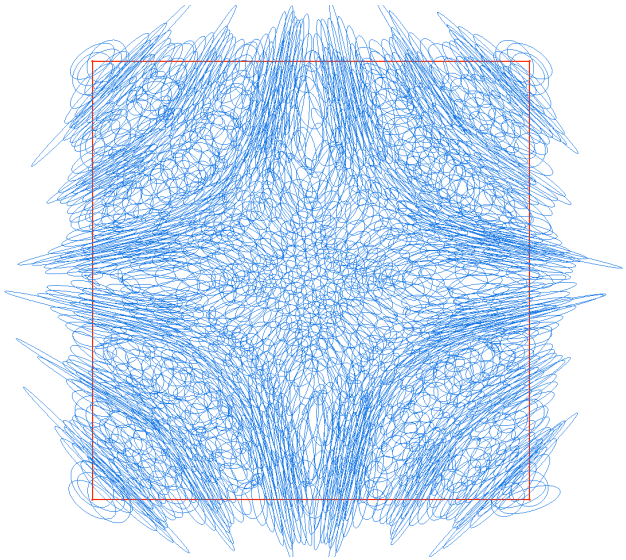
Generating adapted meshes

- Main idea: change mesh generator distance and volume computation [George et al., Adv. Eng. Software 1991]
- Fundamental concept: Generate a local unit mesh w.r.t $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$

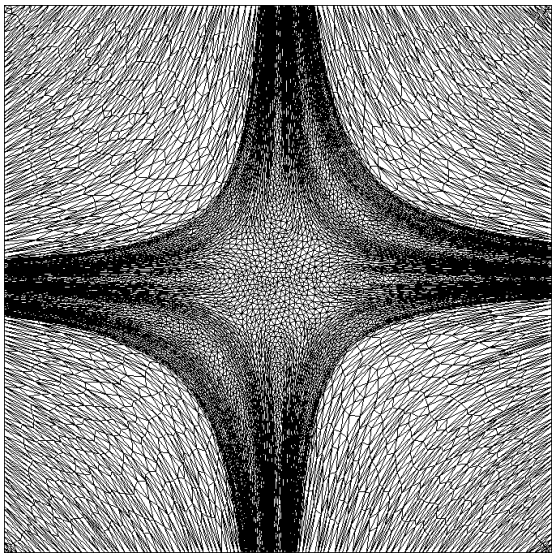
$$\forall \mathbf{e},\ \ell_{\mathcal{M}}(\mathbf{e}) \approx 1 \text{ and } \forall K,\ |K|_{\mathcal{M}} \approx \begin{cases} \sqrt{3}/4 & \text{in 2D} \\ \sqrt{2}/12 & \text{in 3D} \end{cases}$$



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

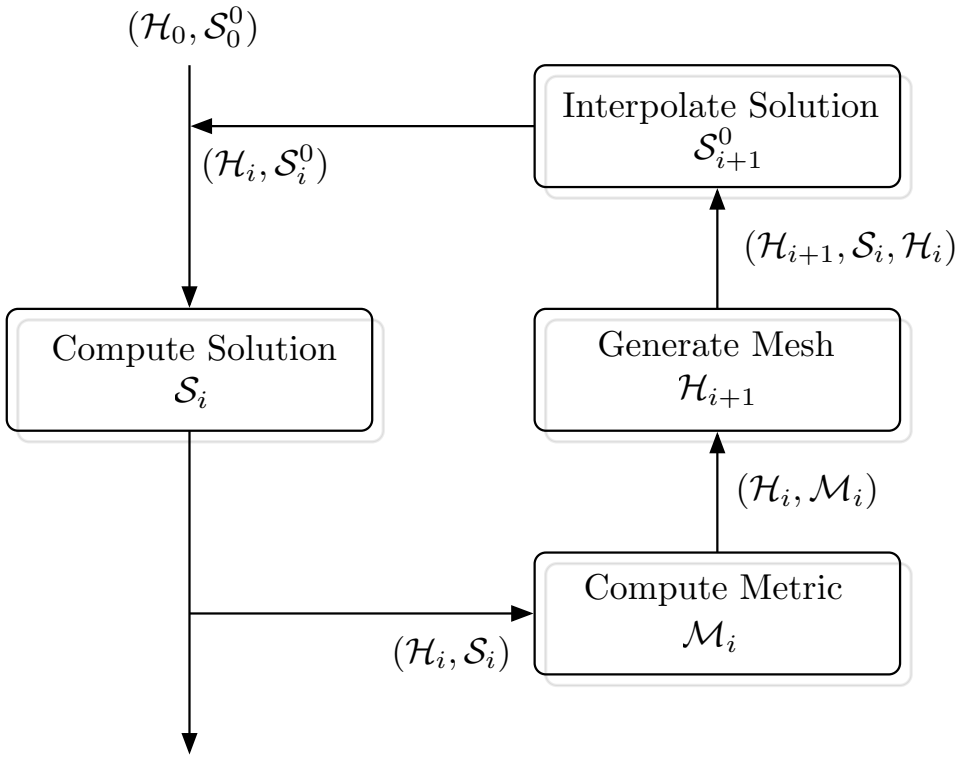


Output $\mathcal{H}$



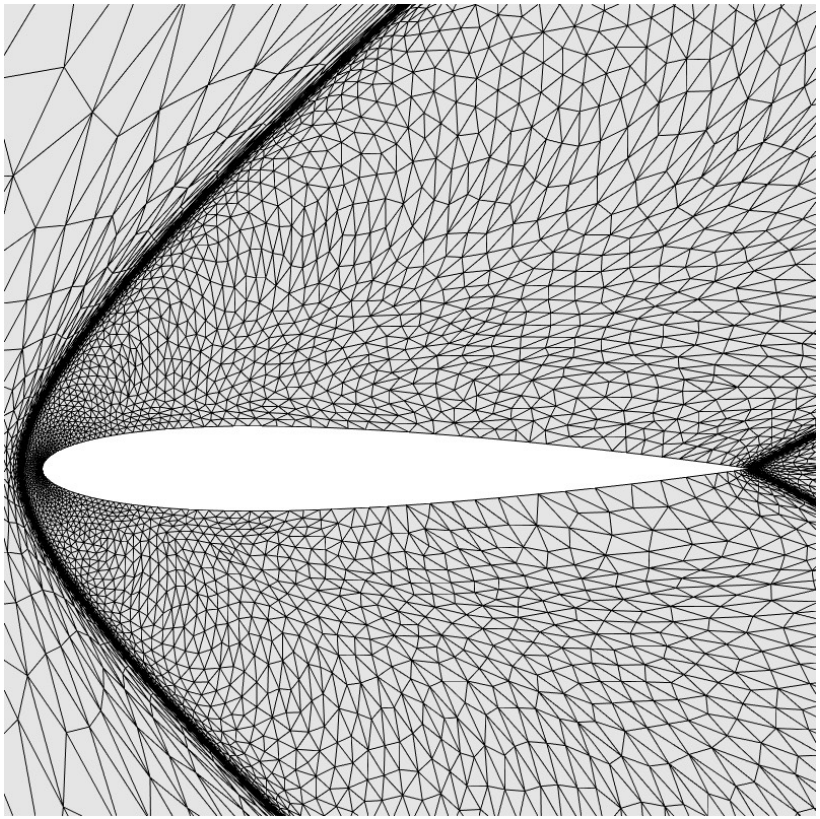
Mesh adaptation loop

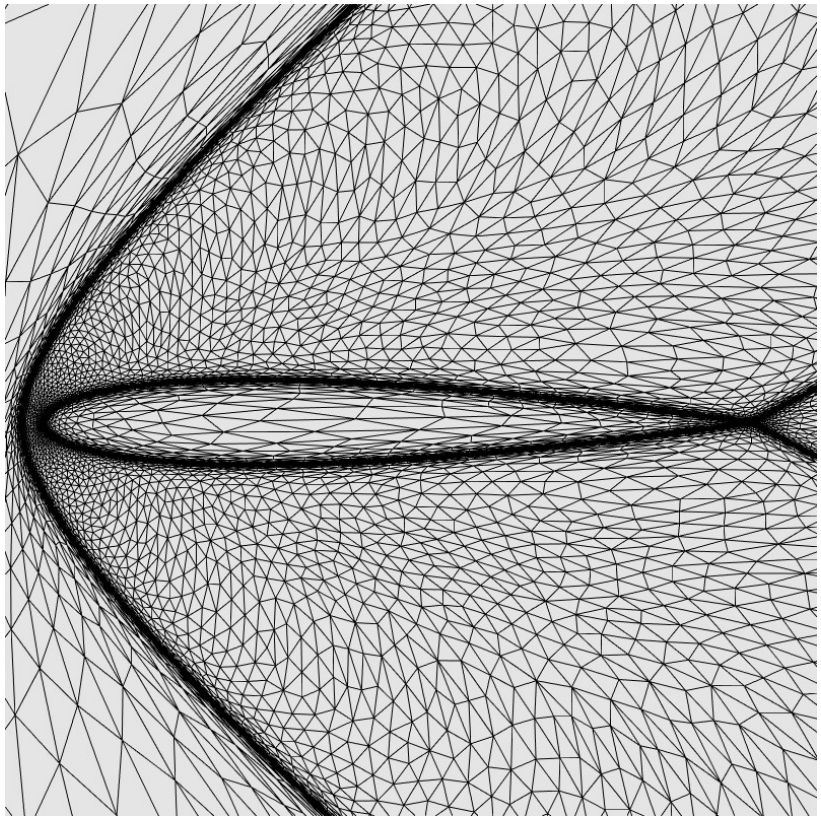
The principle of mesh adaptation is to find the optimal mesh-solution couple for a targeted complexity ($\sim$ number of points)



Mesh adaptation: results

Case of an embedded supersonic NACA0012 airfoil.

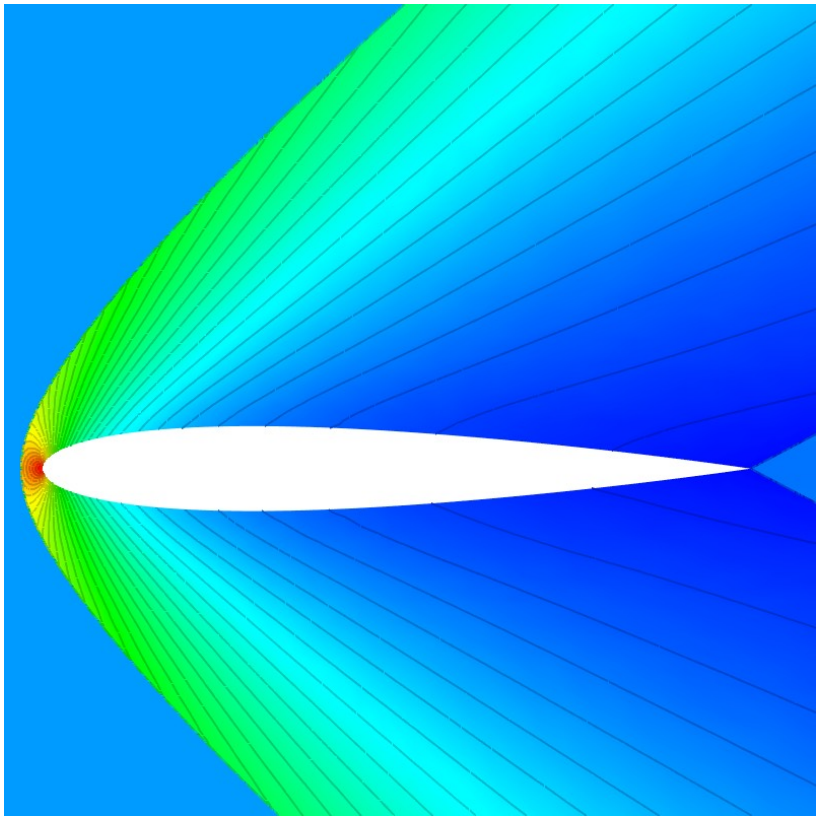


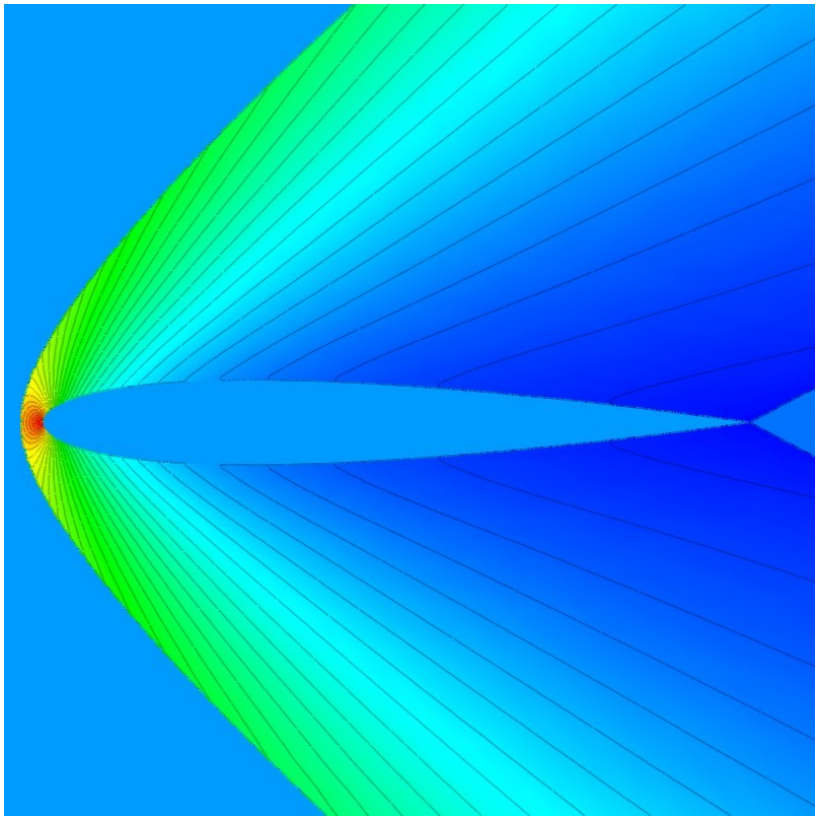


Targeted complexity of 16 000

Mesh adaptation: results

Case of an embedded supersonic NACA0012 airfoil.

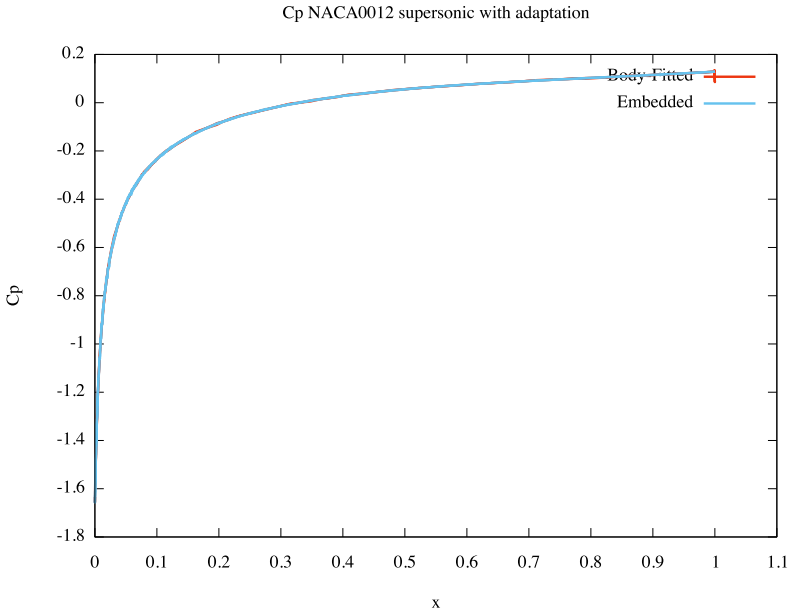




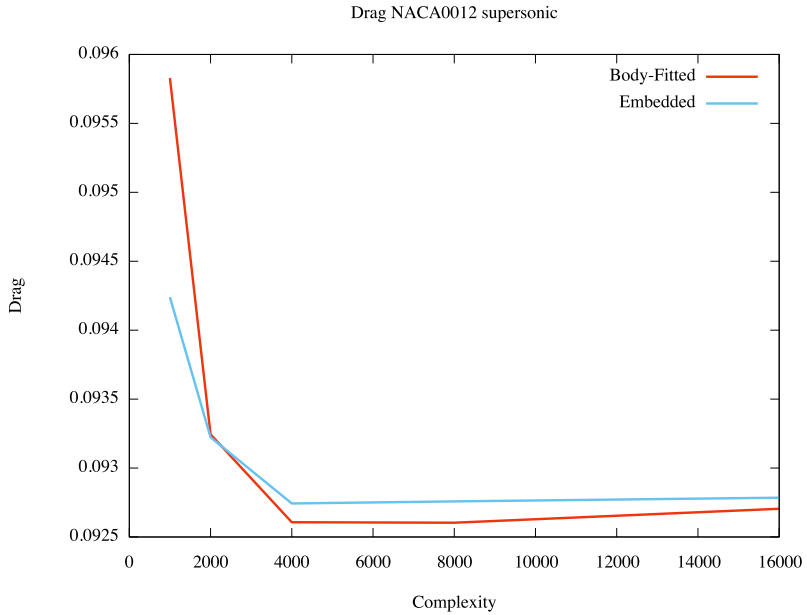
Density flow

Mesh adaptation: results

Case of an embedded supersonic NACA0012 airfoil.



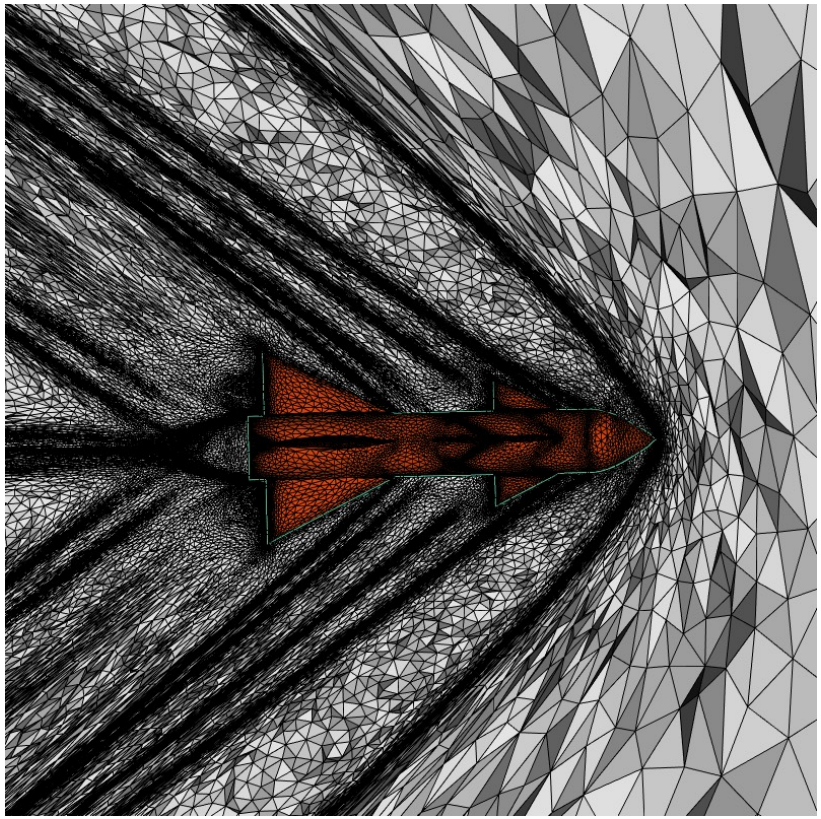
Pressure coefficient at 16 000

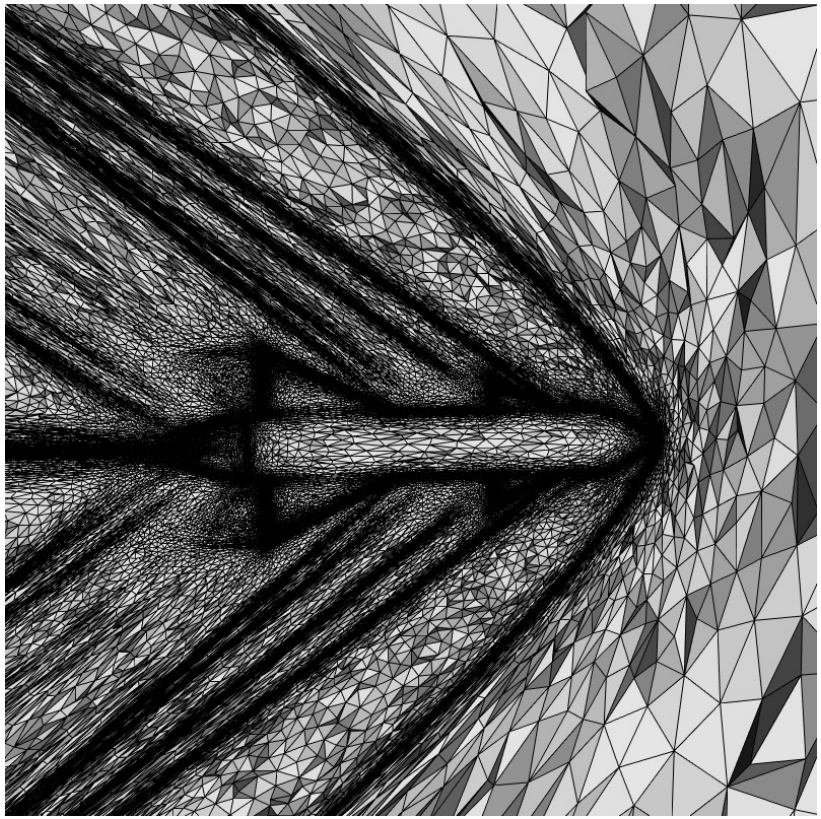


Drag evolution

Mesh adaptation: results

Case of an embedded supersonic missile.

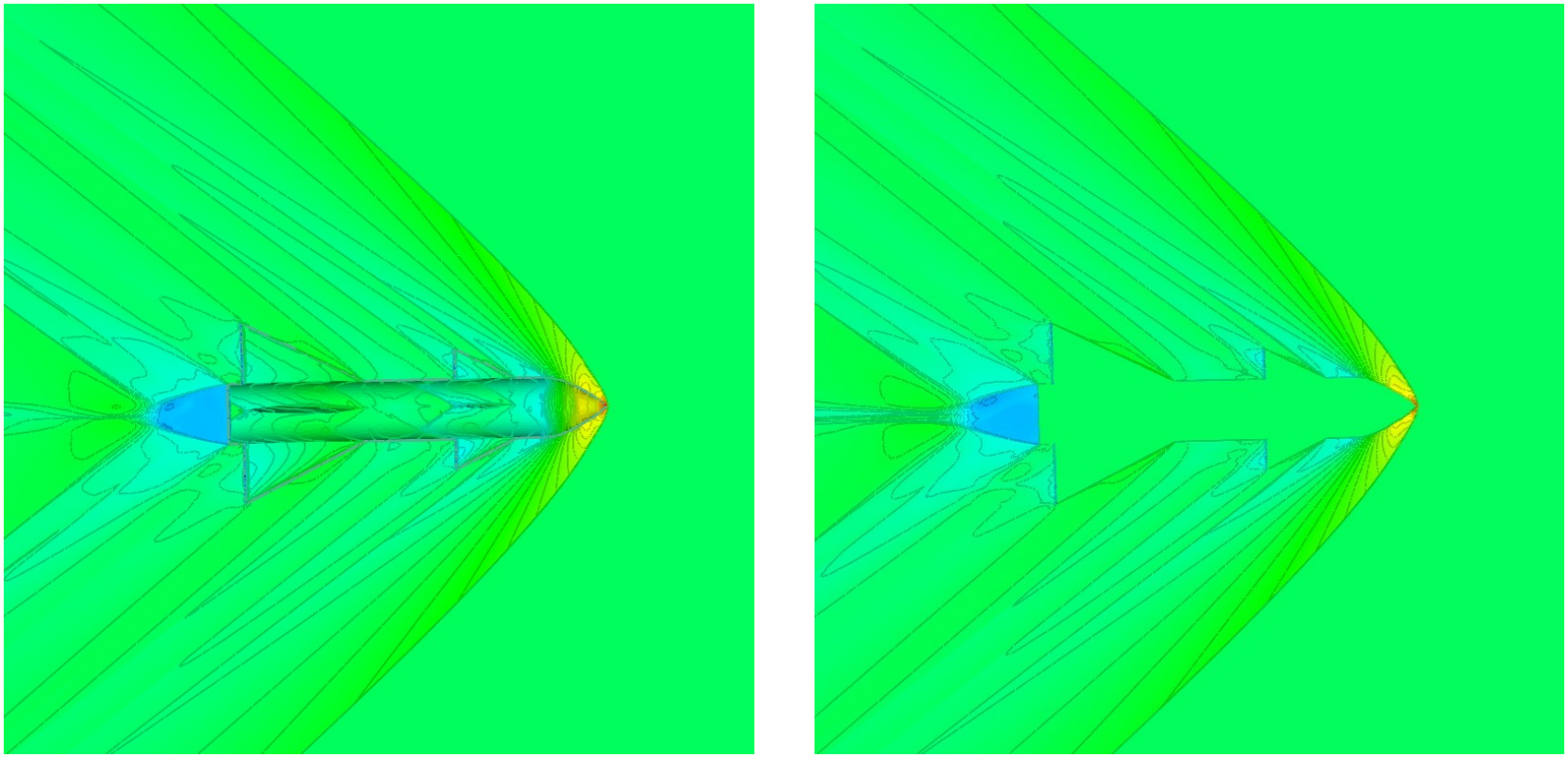




Targeted complexity of 128 000

Mesh adaptation: results

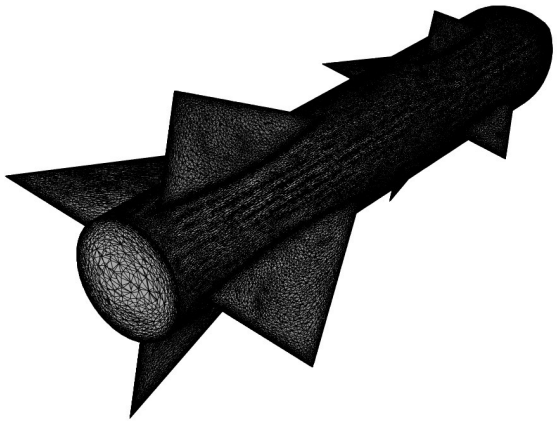
Case of an embedded supersonic missile.

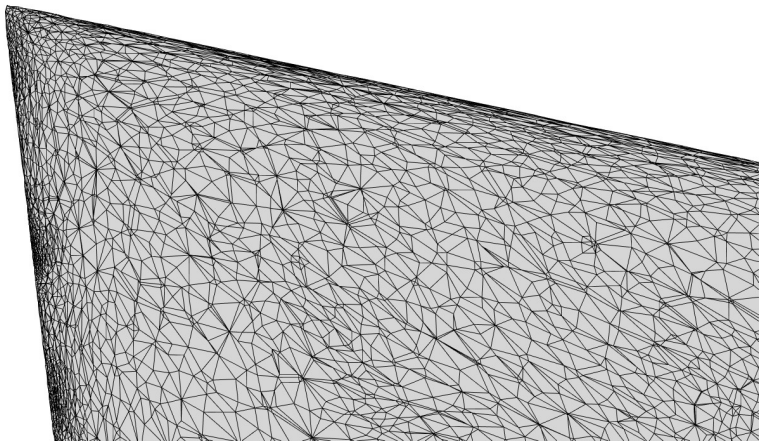


Density flow

Mesh adaptation: results

Case of an embedded supersonic missile.

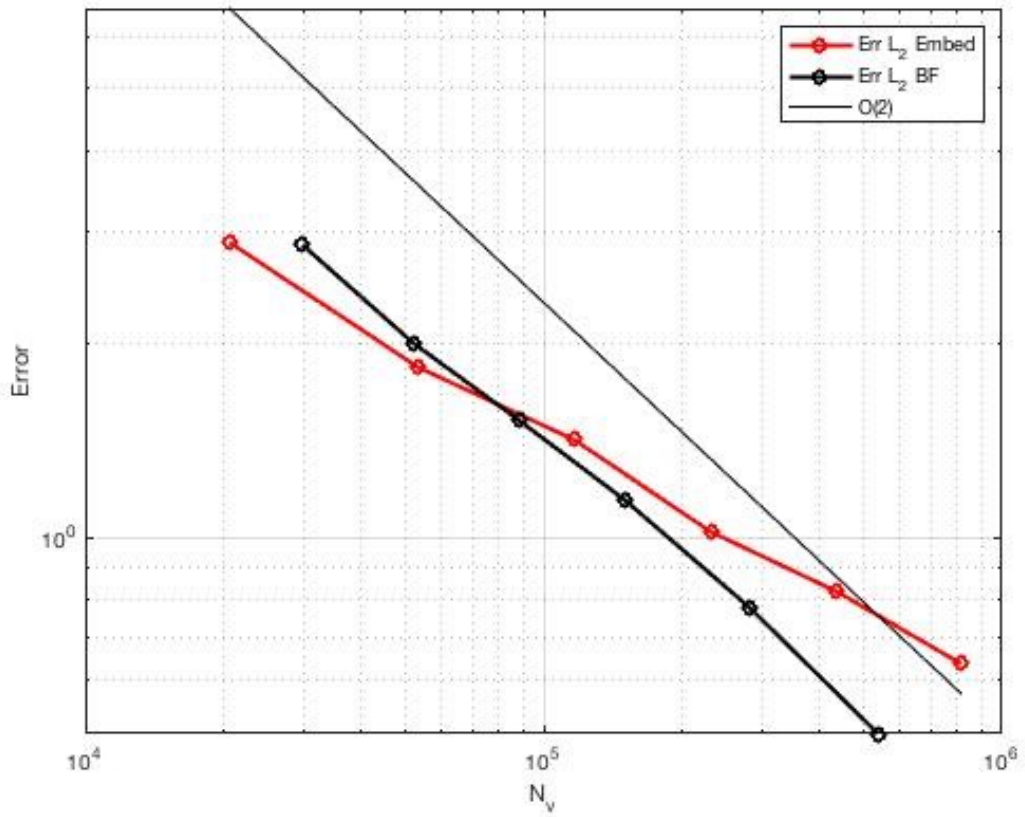




Automatic recovery of sharp features

Mesh adaptation: results

Case of an embedded supersonic missile.



Convergence study using L^2 error

Embedded boundary method: conclusion

Conclusion

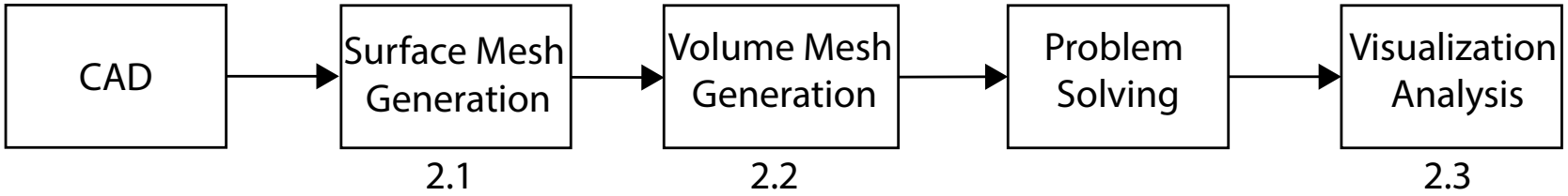
- Implementation of the embedded boundary method in a vertex-centered Finite Volume solver
- Coupling with mesh adaptation enables to get an accuracy comparable to the body-fitted case
- No easy generalization to moving objects

Perspectives

- Use of a high-order embedded geometry
- Computation of a level-set metric
- Extension to Navier-Stokes equations

[1] Feuillet, Loseille and Alauzet, *Mesh adaptation for the embedded boundary method in CFD*, INRIA Research Report, 2019

Outline

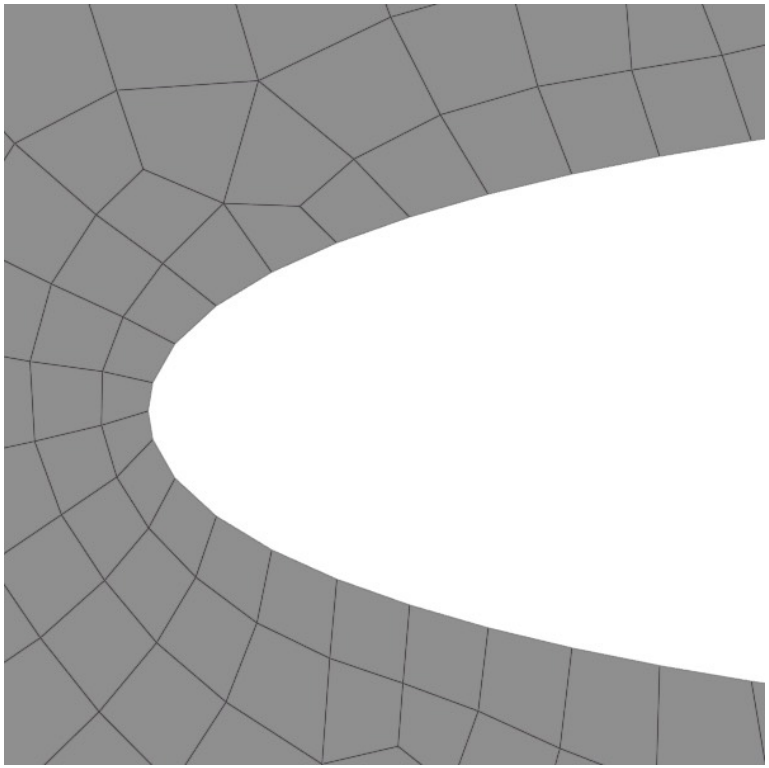


2. High-order
   - 2.1 High-order surface meshing
   - 2.2 Optimization of quadratic meshes and applications
   - 2.3 Visualization of high-order meshes and solutions

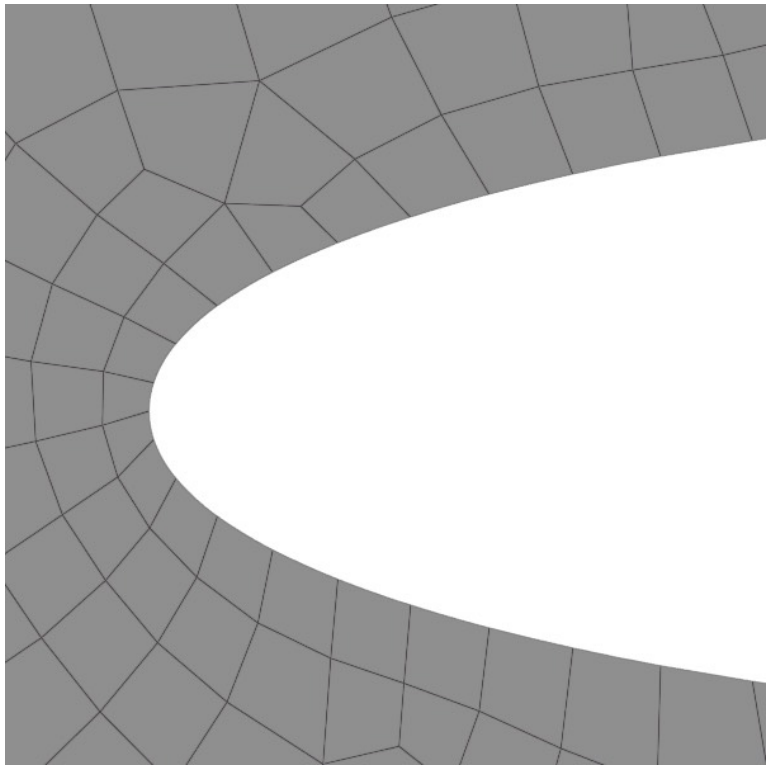
High-order: motivations

Motivations for high-order meshing

- High-order methods : SUPG, DG, FR, SD, WENO, ...
- These methods require high-order discretization of the boundary (meshes of degree k, P^k meshes) to be fully efficient [Bassi et al., JCP 1997]



Degree 1 mesh



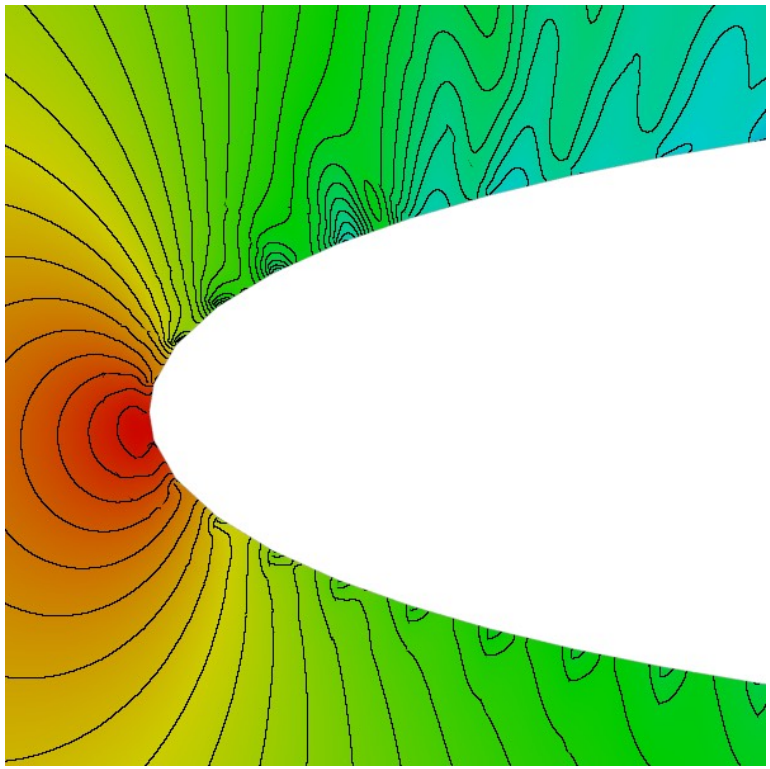
Degree 2 mesh

Courtesy of C. Peyret (ONERA)

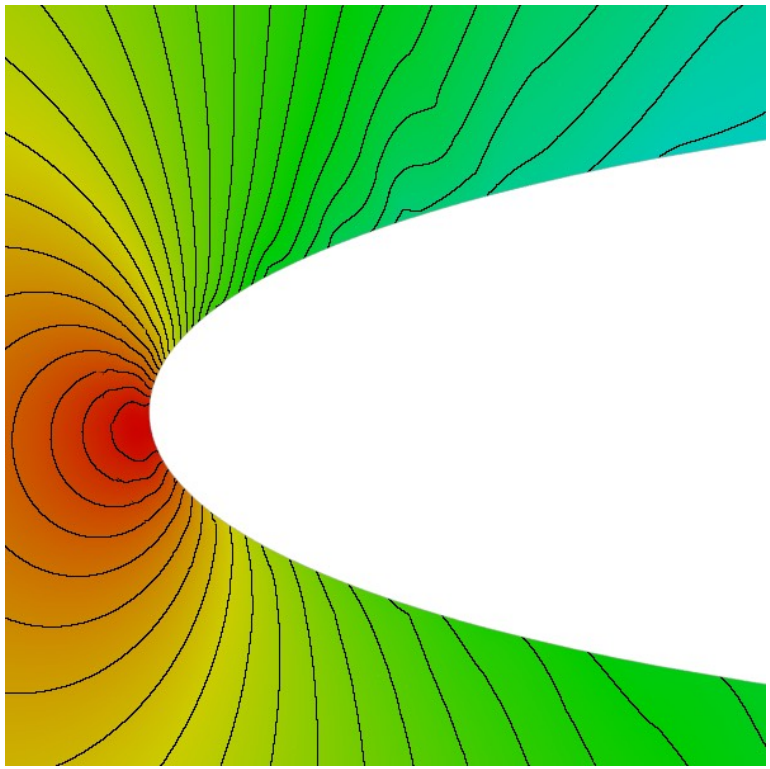
High-order: motivations

Motivations for high-order meshing

- High-order methods : SUPG, DG, FR, SD, WENO, ...
- These methods require high-order discretization of the boundary (meshes of degree k, P^k meshes) to be fully efficient [Bassi et al., JCP 1997]



Degree 1 mesh



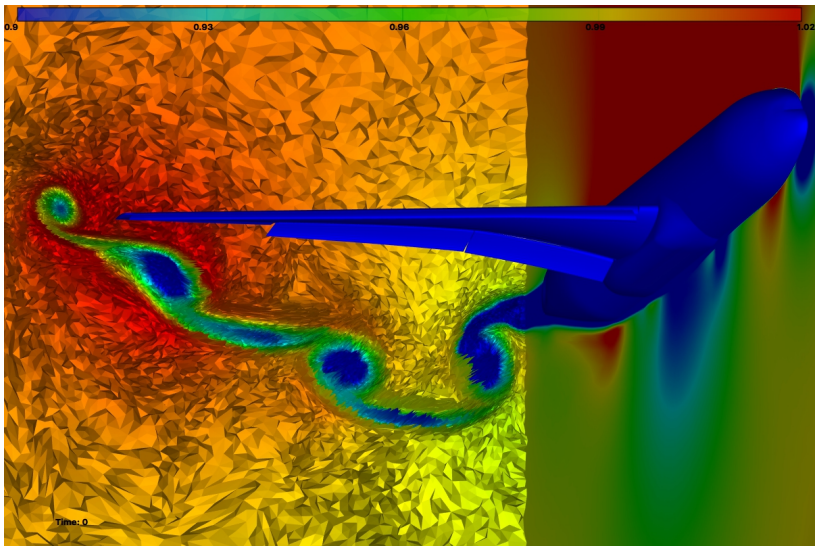
Degree 2 mesh

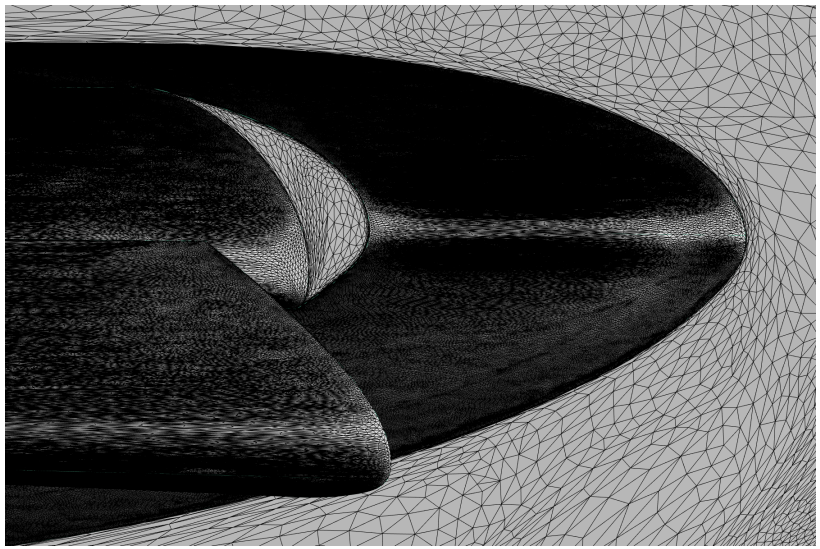
Courtesy of C. Peyret (ONERA)

High-order: motivations

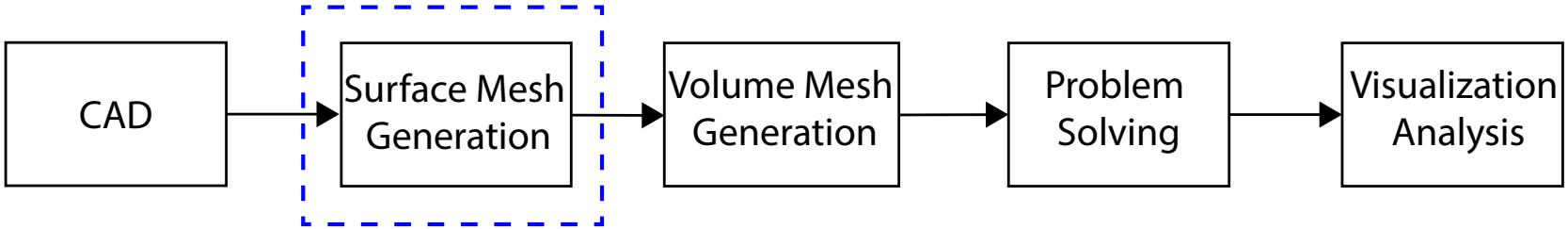
Motivations for surface remeshing

- CAD queries are expensive and not fail-safe
- Surface remeshing using a high-order geometry as CAD surrogate rather than a linear one





Outline



High-order surface meshing

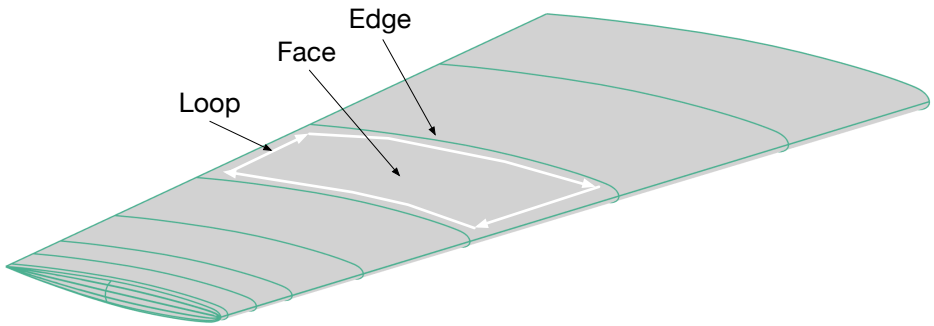
High-order surface meshing

Related work

- Seminal work on a direct high-order mesh generation technique [Dey et al., IMR 1999]
- Relocation of the DOF on an initial linear mesh [Ruiz-Gironès et al., IMR 2015 & 2016], [Toulorge et al., JCP 2016], [Turner, PhD Thesis 2018]
- CAD is used for projection only
  - $\Longrightarrow$ No explicit control of "high-order" approximation of the surface
  - $\Longrightarrow$ No extension of linear anisotropic meshing to high order anisotropic meshing on CAD surfaces

High-order surface meshing

- Focus on parametric surfaces defined by a Boundary Representation (BREP)



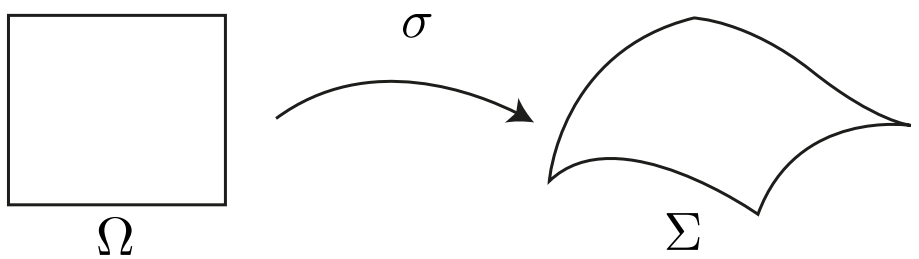
- The linear case is well established [de Cougny et al., IMR 1996], [Tristano et al., IMR 1998], [Miranda et al., IMR 2002], [Siqueira et al., IMR 2010], [Laug, FEAD 2010], [Aubry et al. CAD 2014]

Scope of the work

- Extension of standard linear parametric surface mesh generation techniques
- Based on the development of high-order and intrinsic error estimates

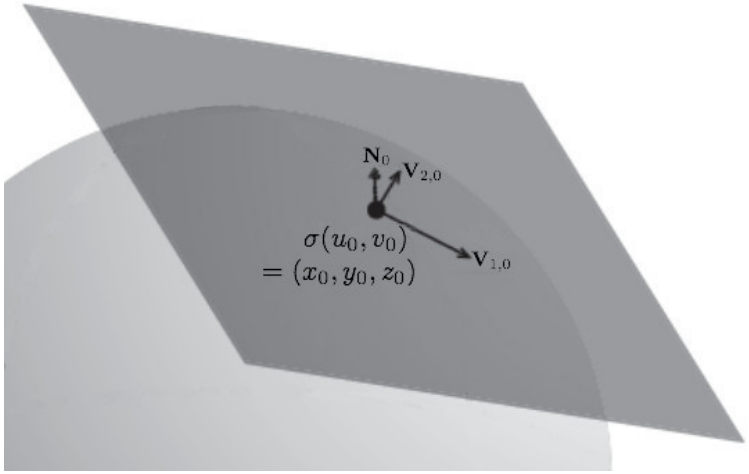
Study of the Linear Case

- The meshing is performed in a hierarchical fashion: edges and then faces
- It consists in an anisotropic meshing in the parametric space of both curves and surfaces
- The key feature is the computation of the metrics



Meshing a surface

- The principal curvature κ_i, their associated principal directions $\mathbf{V}_{i,0}$ and the normal vector $\mathbf{N}_0 = \mathbf{V}_{1,0} \times \mathbf{V}_{2,0}$ is defined for each regular point ($\sigma'(u_0, v_0)$ invertible) of a parametric surface $(u, v) \to \sigma(u, v)$



- These data are intrinsic [Do Carmo, Differential Geometry 2016]
- From this, a 3D metric is deduced ($\rho_i = \frac{1}{\kappa_i}$, $c_i, \lambda \in \mathbb{R}$):

$$\mathcal{M}_{surf} = \begin{pmatrix} \mathbf{V}_{1,0} \\ \mathbf{V}_{2,0} \\ \mathbf{N}_0 \end{pmatrix}^T \begin{pmatrix} \frac{1}{(c_1\rho_1)^2} & 0 & 0 \\ 0 & \frac{1}{(c_2\rho_2)^2} & 0 \\ 0 & 0 & \lambda \end{pmatrix} \begin{pmatrix} \mathbf{V}_{1,0} \\ \mathbf{V}_{2,0} \\ \mathbf{N}_0 \end{pmatrix}$$

- In the parametric space, the used metric is:

$$\tilde{\mathcal{M}}_{surf} = \begin{pmatrix} \sigma_u^T & \sigma_v^T \end{pmatrix} \mathcal{M}_{surf} \begin{pmatrix} \sigma_u \\ \sigma_v \end{pmatrix}$$

Extension to high-order

Requirements

- Direct extension of the linear case
- Independence with respect to any parameterization

Extension to high-order

Requirements

- Direct extension of the linear case
- Independence with respect to any parameterization

Observations

- Principal curvatures are terms of order 2 of Taylor expansions of the gap to the tangent plane in the intrinsic local frame

Extension to high-order

Requirements

- Direct extension of the linear case
- Independence with respect to any parameterization

Observations

- Principal curvatures are terms of order 2 of Taylor expansions of the gap to the tangent plane in the intrinsic local frame
- If we note $\sigma(u_0, v_0) = (x_0, y_0, z_0)$ in $(\mathbf{V}_{1,0}, \mathbf{V}_{2,0}, \mathbf{N}_0)$, then

$$z = z_0 + \frac{1}{2}(\kappa_1(x - x_0)^2 + \kappa_2(y - y_0)^2) + \mathcal{O}(\|(x - x_0, y - y_0)\|^3)$$

Extension to high-order

Strategy

- Extend the previous Taylor expansion to higher orders
- Deduce from these expansions "high-order curvatures"
- Build new metrics using these curvatures
- Generate high-order meshes using these metrics

Case of a surface

- We assume that σ is at least C^{k+1}
- In the local frame, we have

$$\begin{aligned}(X, Y) &= (x - x_0, y - y_0) \\ &= \left((\sigma(u, v) - \sigma(u_0, v_0), \mathbf{V}_{1,0}\right), \left(\sigma(u, v) - \sigma(u_0, v_0), \mathbf{V}_{2,0}\right)) \\ &= \phi(u - u_0, v - v_0)\end{aligned}$$

- If (u_0, v_0) is a regular point, then, using the inversion function theorem ϕ is invertible in (u_0, v_0) and there exists ψ such that $\psi(X, Y) = (u - u_0, v - v_0)$ in the vicinity of (u_0, v_0)
- If ϕ is C^{k+1}, then ψ is C^{k+1}

Case of a surface

- ϕ is an analytical function whose derivatives up to order $k+1$ are known [Piegl et al., NURBS Book 1997]
- Using an inversion formula, the derivatives of ψ can be deduced up to order $k+1$ [Encinas et al., Applied Mathematics Letters 2003]
- We then deduce a Taylor expansion of parameters (u, v) in the vicinity of (u_0, v_0) with respect to (X, Y):

$$\begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} = \sum_{n=1}^{k+1} \sum_{i+j=n} A_{ij}^n X^i Y^j + \mathcal{O}(||(X, Y)||^{k+2}),$$

- As σ is C^{k+1}, Z has a Taylor expansion in $(u - u_0, v - v_0)$ up to order $k+1$

Case of a surface

- By composition, it comes the intrinsic Taylor expansion:

$$Z = F_k(X, Y) + R_{k+1}(X, Y) + \mathcal{O}(||(X, Y)||^{k+2})$$

  F_k is a polynomial of degree $\leq k$ and R_{k+1} is an homogeneous polynomial of degree $k + 1$

- Using the log-simplex algorithm [Coulaud et al., IMR, 2016], it comes:

$$|R_{k+1}(X, Y)| \leq \left(\frac{1}{2} (X \ Y) Q_{k+1} (X \ Y)^T \right)^{\frac{k+1}{2}}$$

  where Q_{k+1} is a symmetric matrix

- "High-order principal curvatures" κ_i^{k+1} are chosen as the eigenvalues of Q_{k+1} and the "high-order principal directions" $\mathbf{V}_i^{k+1}$ are deduced from the eigenvectors

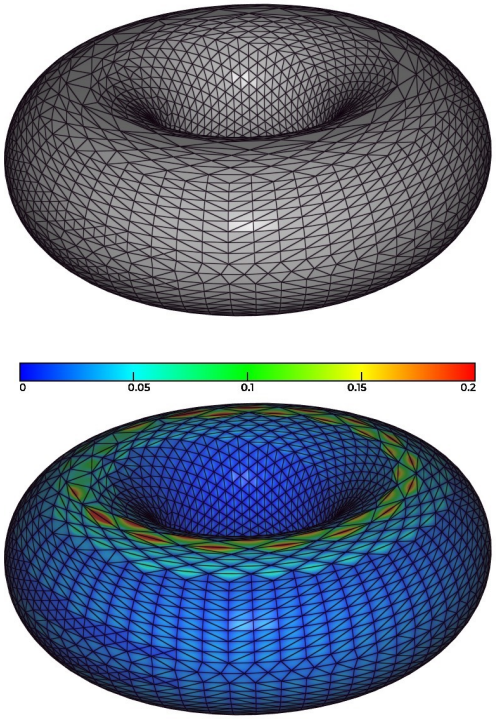
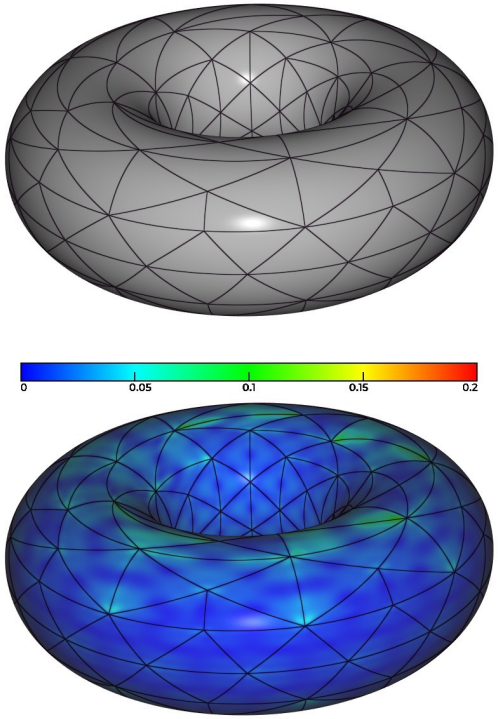
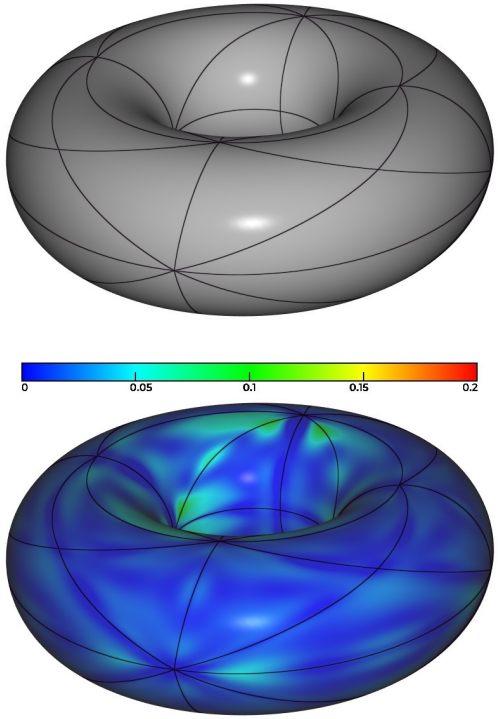
Surface meshing process

Meshing steps

- Perform line meshing using a high-order metric
- Perform surface meshing in the parametric space using a high-order metric
- Project high-order nodes onto the CAD geometry

Torus

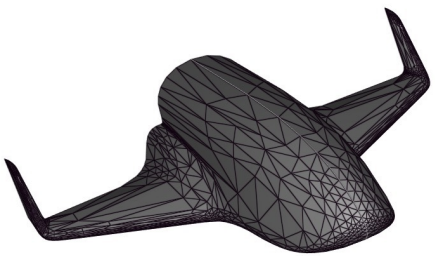
- 2 NURBS of degree 5 both defined by 6 control points and 12 knots.
- Results for an equivalent level of error (err $\sim 10^{-1}$)

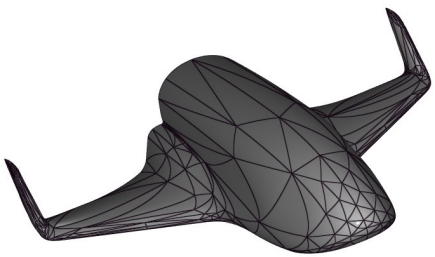
P^1	P^2	P^3
		
2787 DOF	666 DOF	252 DOF

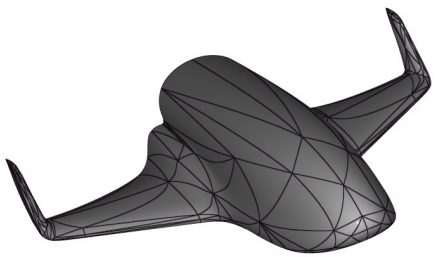
Shuttle

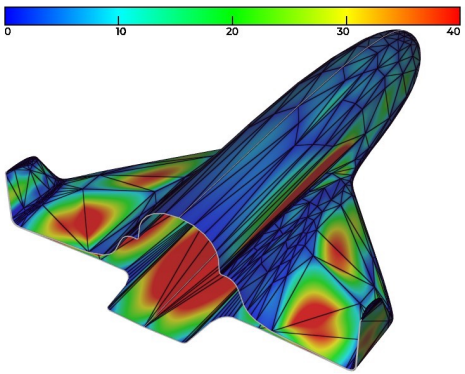
- 2 NURBS of degree 3 defined by 8 (resp. 13) control points and 12 (resp. 17) knots with strong variation in the parametric space
- Results for an equivalent number of degrees of freedom (DOF $\sim$ 1700)

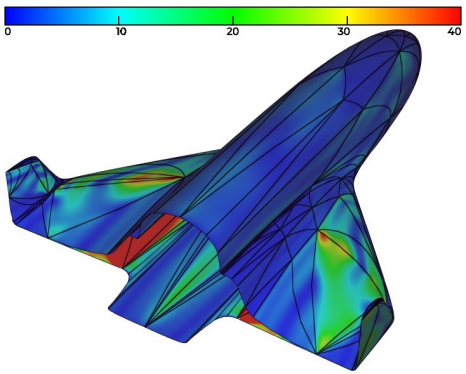
P^1 P^2 P^3

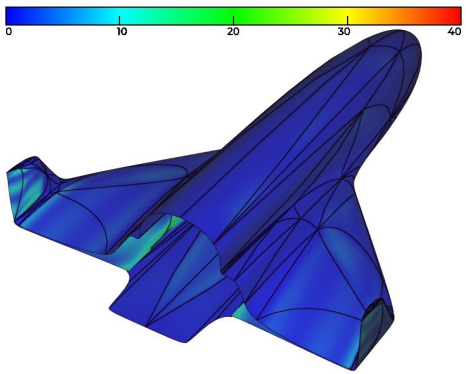












err = 197.814 err = 58.3507 err = 22.217

High-order surface meshing: conclusion

Conclusion

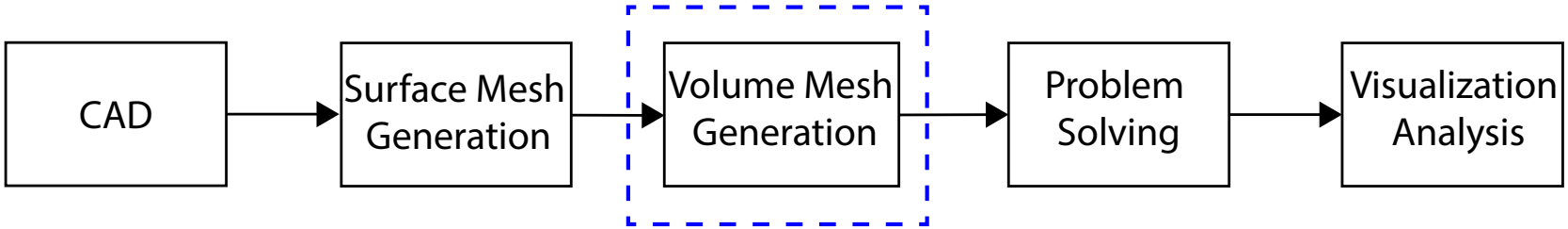
- A novel approach for high-order surface mesh generation
- It is based on error estimates issued of an intrinsic Taylor expansion
- Readily usable in a surface mesh generation process

Perspectives

- High-order implicit surface meshing
- Coupling with curvilinear mesh adaptation

[2] Feuillet, Coulaud and Loseille, *Anisotropic error estimate for high-order parametric surface mesh generation*, 28th International Meshing Roundtable, 2019

Outline



Optimization of quadratic meshes and applications

P^2 mesh optimization and applications

Related work

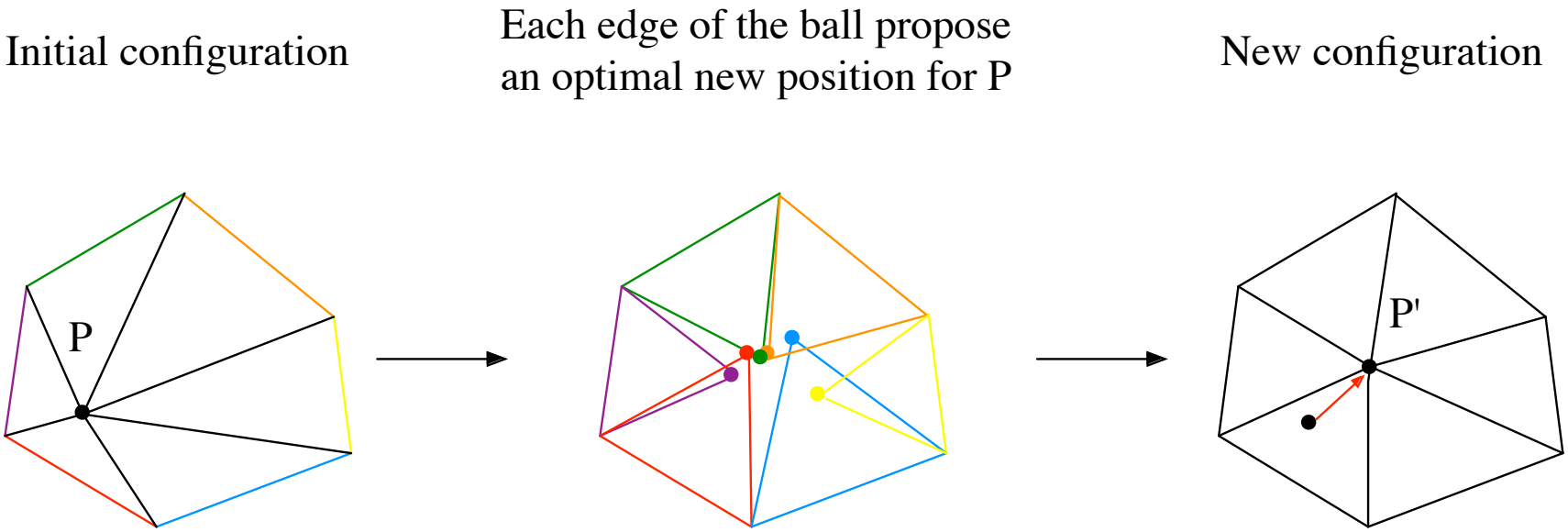
- High-order volume meshing with high-order surface as an input
  $\Longrightarrow$ Local approach: high-order mesh untangling [George et al., IJNME 2012], [Toulorge et al., JCP 2013]
  $\Longrightarrow$ Global approach: PDE/variational models [Roca et al., IMR 2012], [Moxey et al., IMR 2014]

Intent

- Expose P^2 quality-based mesh optimization operators for simplicial meshes
- Use P^2 quality function proposed in [George et al., 2019]
  $\Longrightarrow$ It varies between 1 and infinity, the best element (equilateral) has a quality of 1
- Applications to P^2 mesh generation, P^2 moving mesh techniques and P^2 boundary layer mesh generation

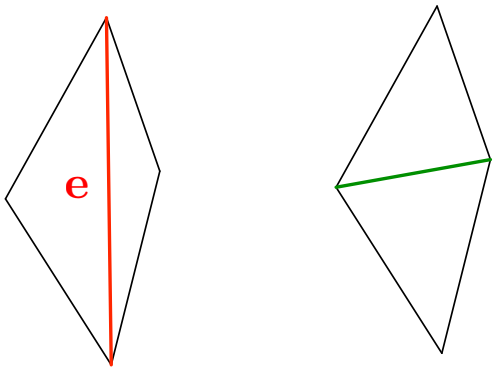
P^1 operators

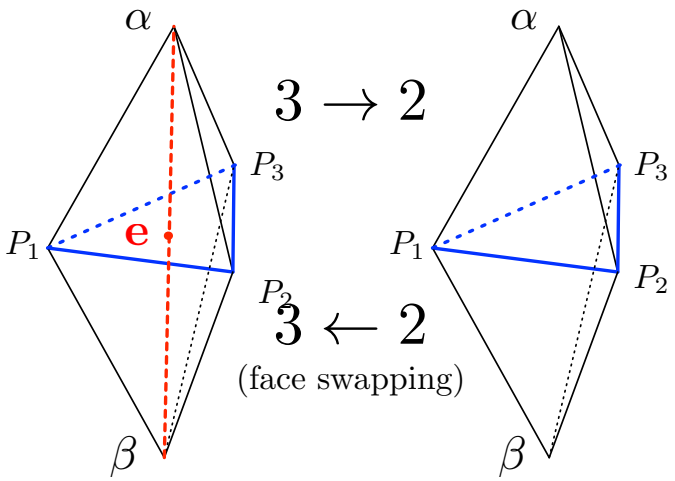
- Two classical operators of P^1 mesh optimization [Brière de l'Isle et al., 1995]
- Mesh smoothing



P^1 operators

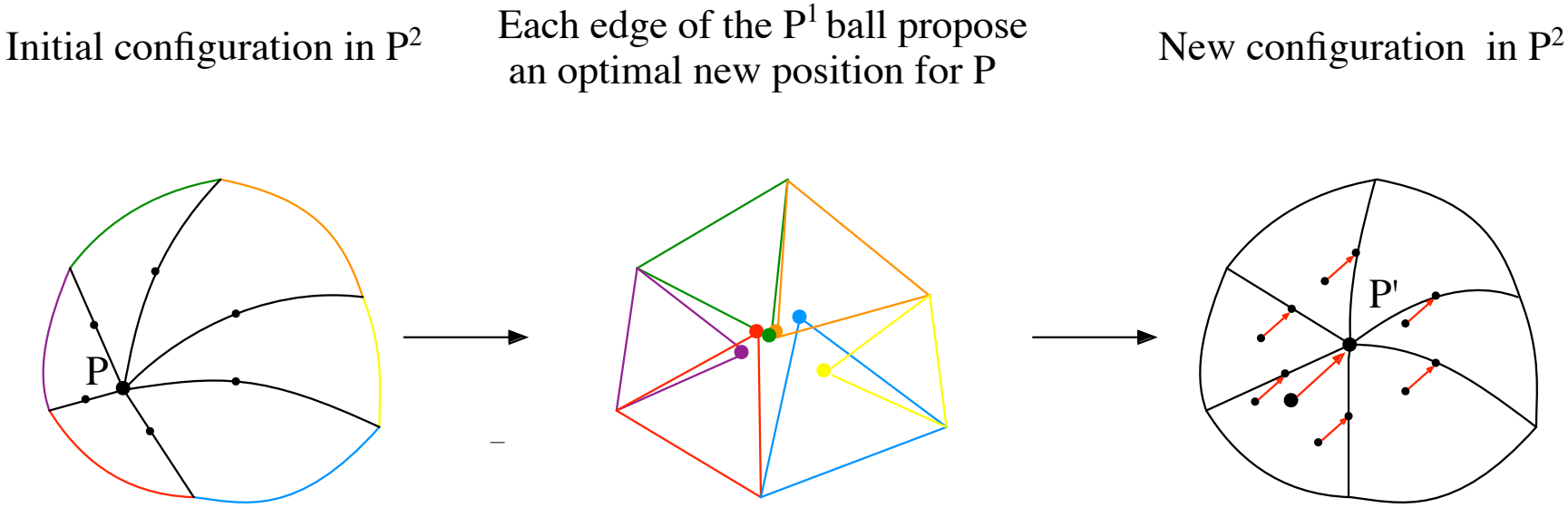
- Two classical operators of P^1 mesh optimization [Brière de l'Isle et al., 1995]
- Generalized edge swapping
- Delete an edge and replace it by a new triangulation of its shell





P^2 Mesh smoothing

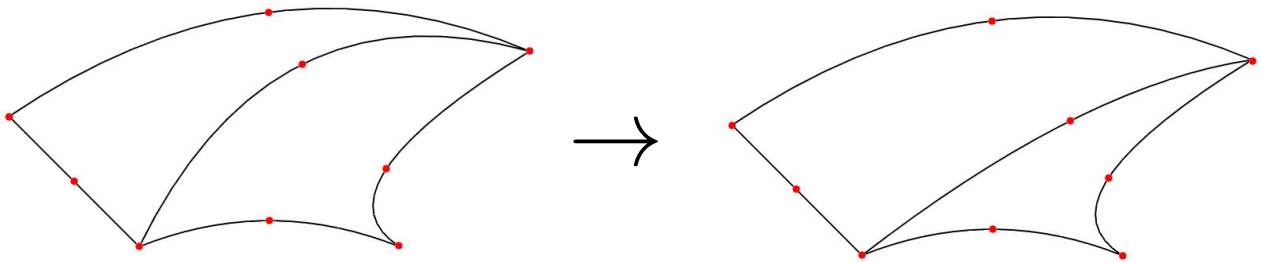
- P^2 mesh smoothing is done in two steps
- Vertex smoothing



- Optimization of the position of the vertices using P^1 smoothing technique
- Extrapolation for the nodes of the ball so that P^2 straight meshes stay consistent with P^1 meshes

P^2 Mesh smoothing

- P^2 mesh smoothing is done in two steps
- Node smoothing

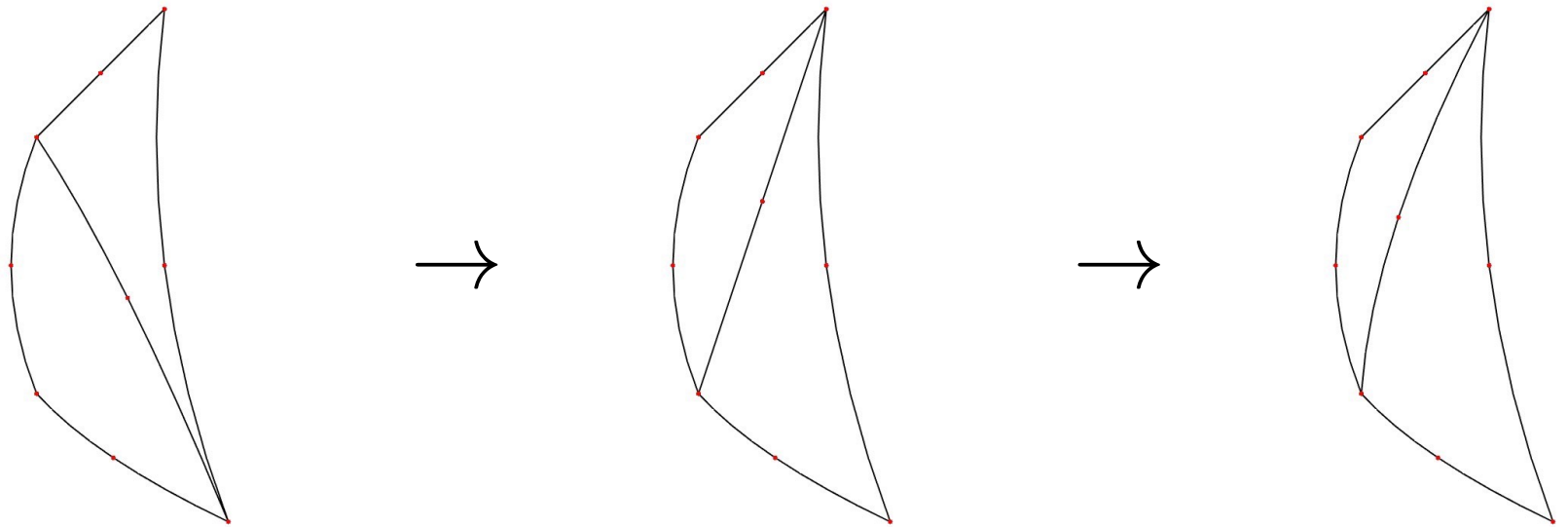


$Q_{ini} = 11.34\ (4.69)$ $Q_{end} = 8.36\ (2.65)$

- Resolution of an optimization problem

P^2 Edge/Face swapping

- Generalization of edge swapping to P^2 meshes in 3 steps



$Q_{ini} = 10.45\ (6.04)$ $Q_{mid} = 5.78\ (4.48)$ $Q_{end} = 4.78\ (3.72)$

- Resolution of the same optimization problem as with the node smoothing

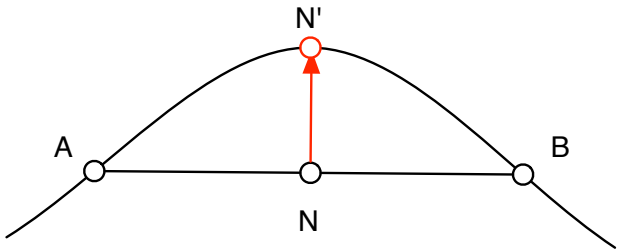
Applications

P^2 Mesh Generation, Moving-Mesh and BL Mesh Generation

- Methods are based on a PDE model
- Finite element resolution of linear elasticity equation
- Resolution at the order of the studied mesh
- Dirichlet boundary conditions are used

High-order mesh generation

- The initial boundary deformation is given by the input high-order surface mesh



- This is interpreted as a Dirichlet boundary condition
- Resolution of the FE system gives volume deformation
- High-order resolution provides a model in which all DOF are intrinsically represented.
- Optimization operators in post and preprocessing ensure robustness to the process

NASA Common Research Model

Mesh size: 32 479 vertices - 118 012 tets - 45 956 triangles

Total number of swaps: 12 703

$Q \in [1, \infty)$

P^1 mesh

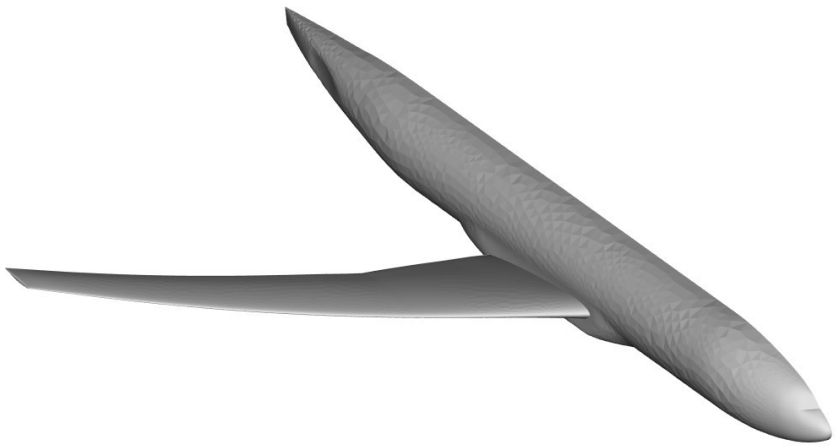
$Q_{avg} = 2.5$

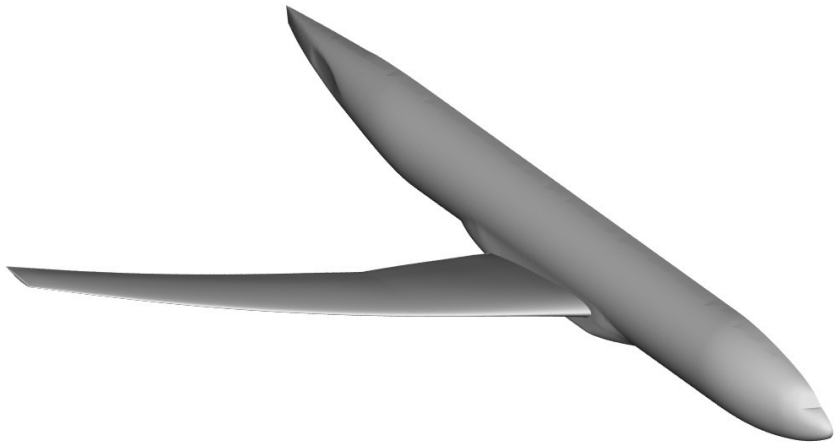
$Q_{wst} = 271$

P^2 mesh

$Q_{avg} = \infty \xrightarrow{optim} 2.13$

$Q_{wst} = 10 \ inv. \xrightarrow{optim} 271$





NASA Common Research Model

Mesh size: 32 479 vertices - 118 012 tets - 45 956 triangles

Total number of swaps: 12 703

$Q \in [1, \infty)$

P^1 mesh

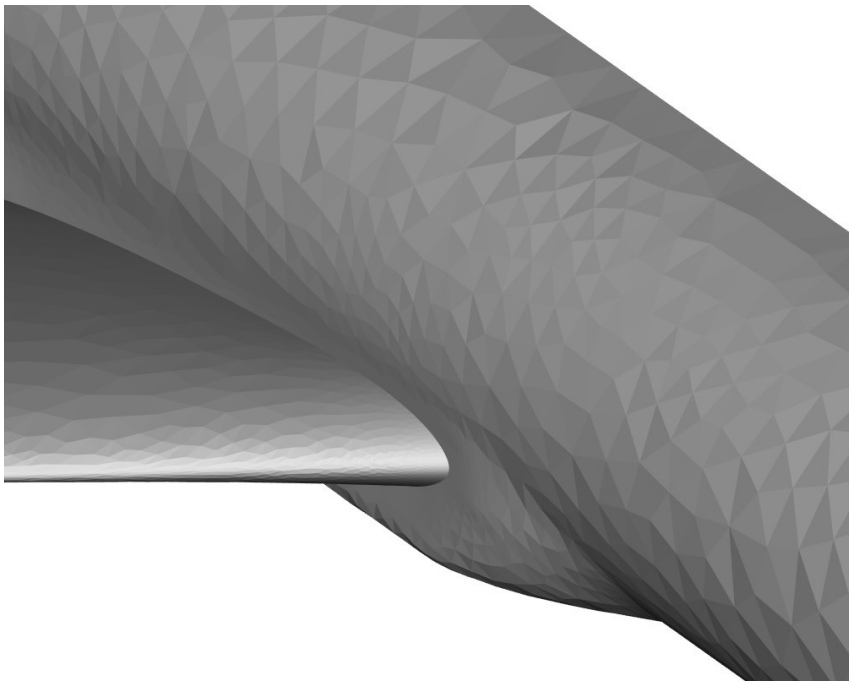
$Q_{avg} = 2.5$

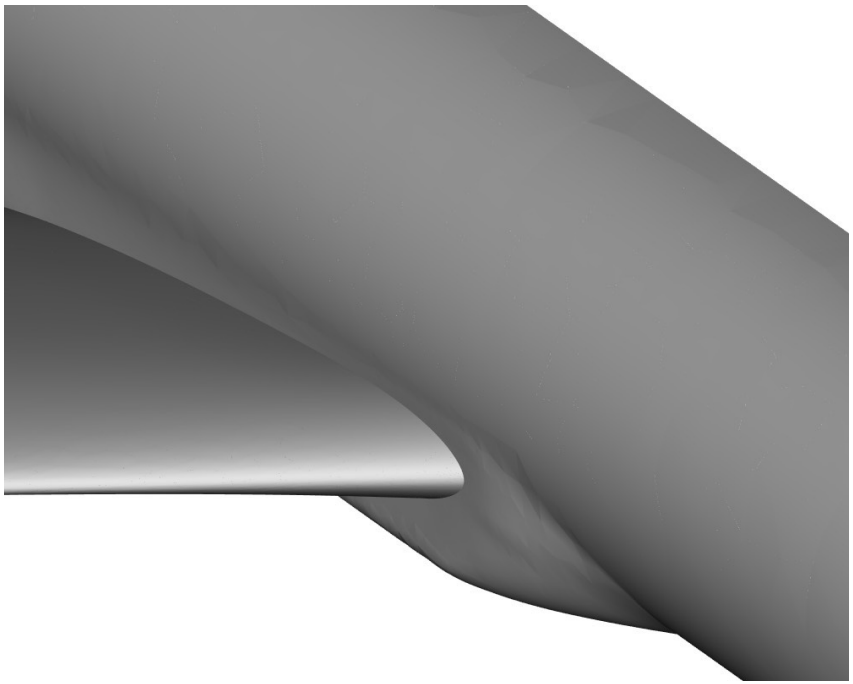
$Q_{wst} = 271$

P^2 mesh

$Q_{avg} = \infty \xrightarrow{optim} 2.13$

$Q_{wst} = 10 \ inv. \xrightarrow{optim} 271$





NASA Common Research Model

Mesh size: 32 479 vertices - 118 012 tets - 45 956 triangles

Total number of swaps: 12 703

$Q \in [1, \infty)$

P^1 mesh

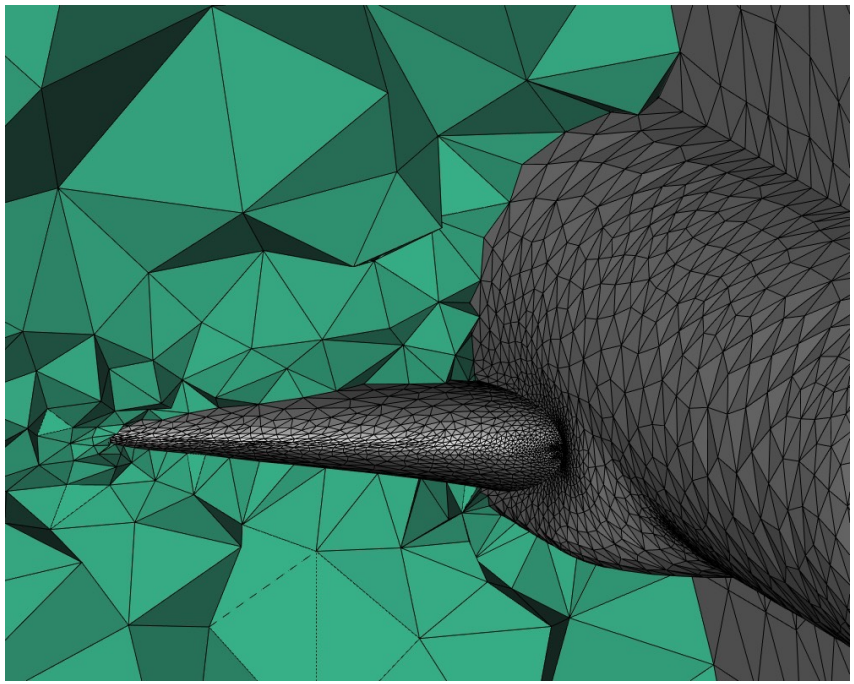
$Q_{avg} = 2.5$

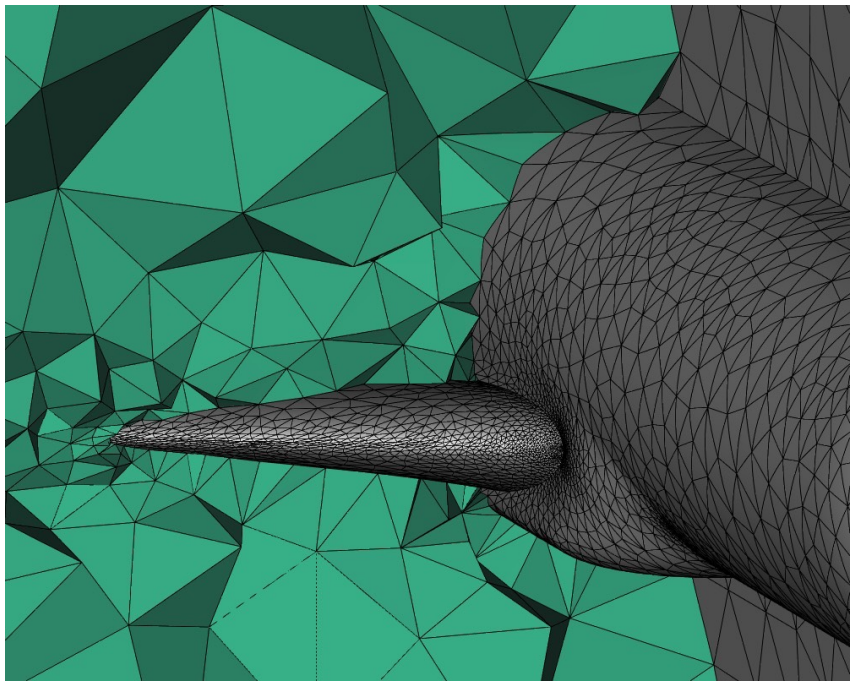
$Q_{wst} = 271$

P^2 mesh

$Q_{avg} = \infty \xrightarrow{optim} 2.13$

$Q_{wst} = 10\ inv. \xrightarrow{optim} 271$





Basic P^2 Moving Mesh Algorithm

- Start from a P^2-mesh

- Compute moving body displacement providing boundary nodes/vertices positions
  It sets Dirichlet boundary conditions for the linear elasticity solver

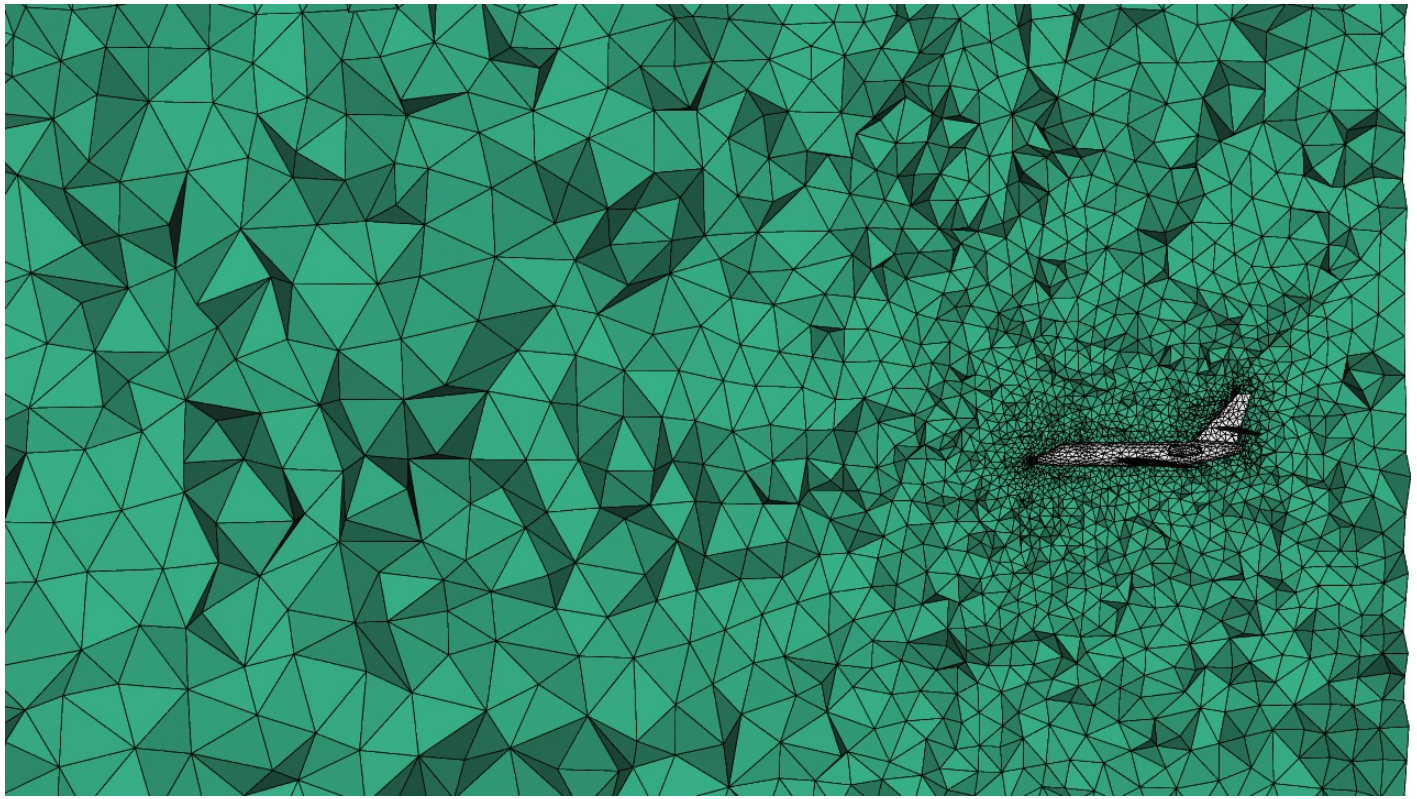
- Compute high-order solution, it provides new positions to all nodes/vertices
  Move all nodes/vertices to their new position

- Perform high-order mesh optimization to correct invalid elements and improve mesh quality

- Iterate until displacement is performed

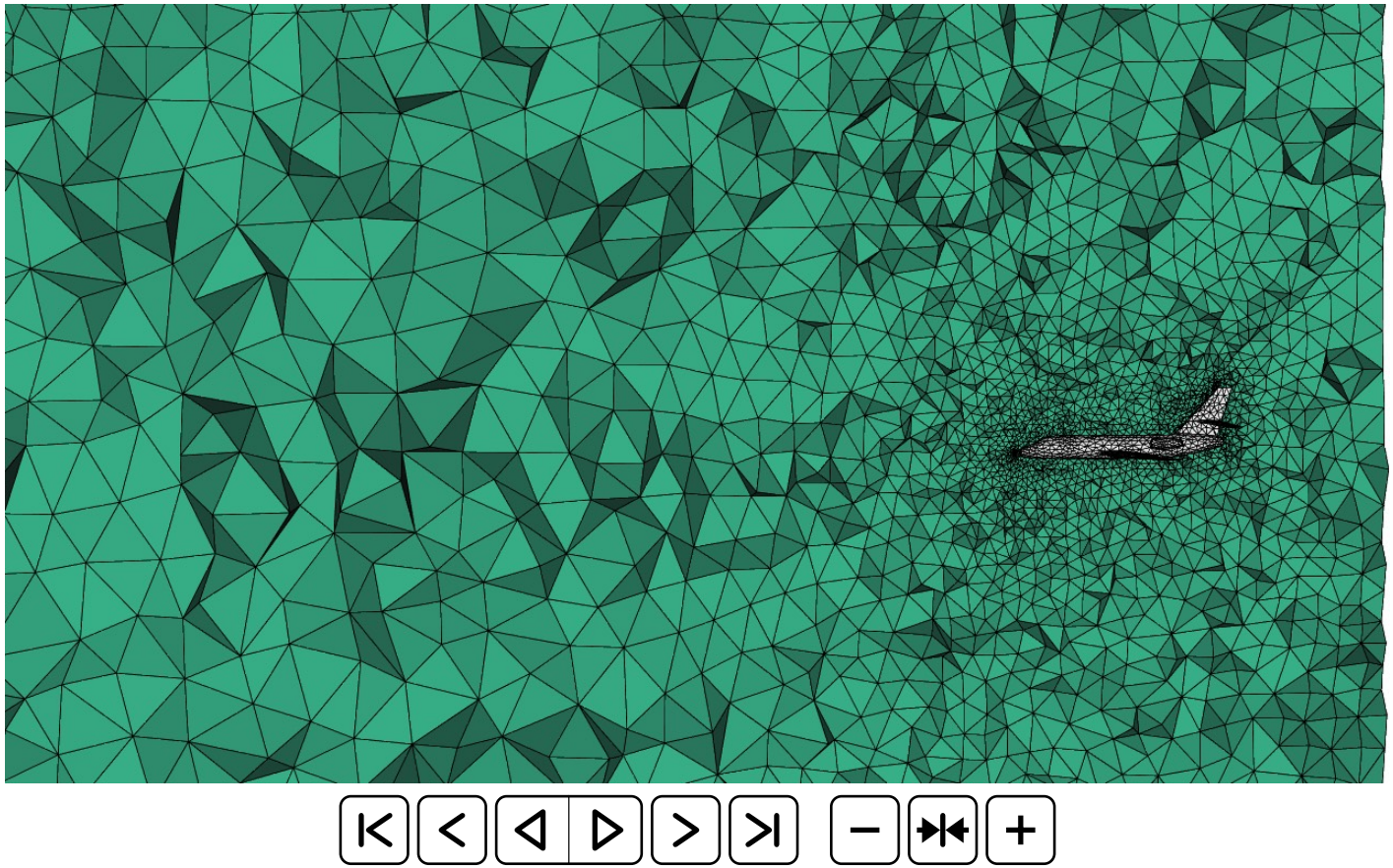
P^2 Moving mesh

- Translating and rotating falcon in a mesh of 498 181 tetrahedra
- Result with no optimization



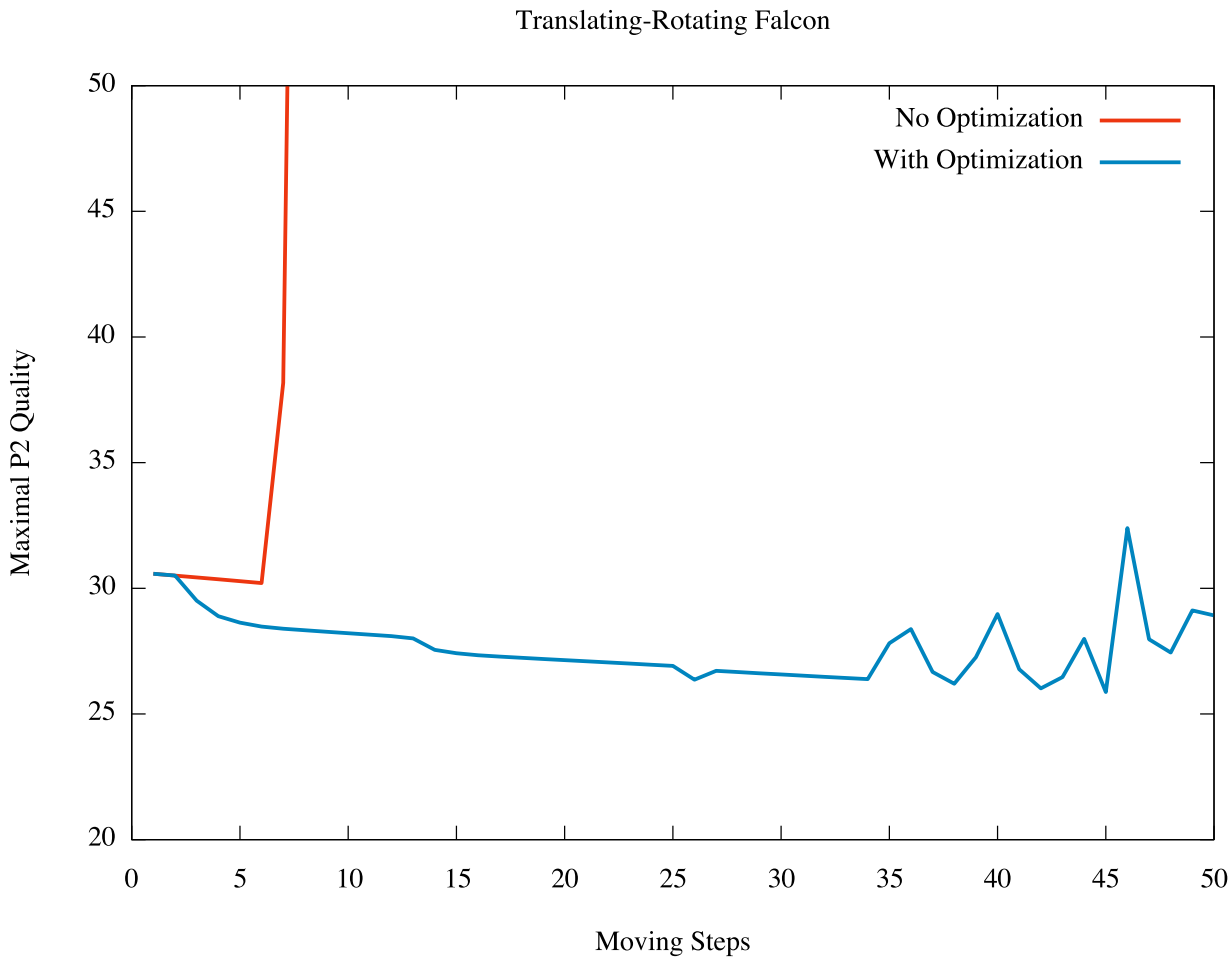
P^2 Moving mesh

- Translating and rotating falcon in a mesh of 498 181 tetrahedra
- Result with swap and vertex/node smoothing



P^2 Moving Mesh

- Translating and rotating falcon in a mesh of 498 181 tetrahedra
- Evolution of the worst quality



Application

P^2 Boundary Layer mesh generation

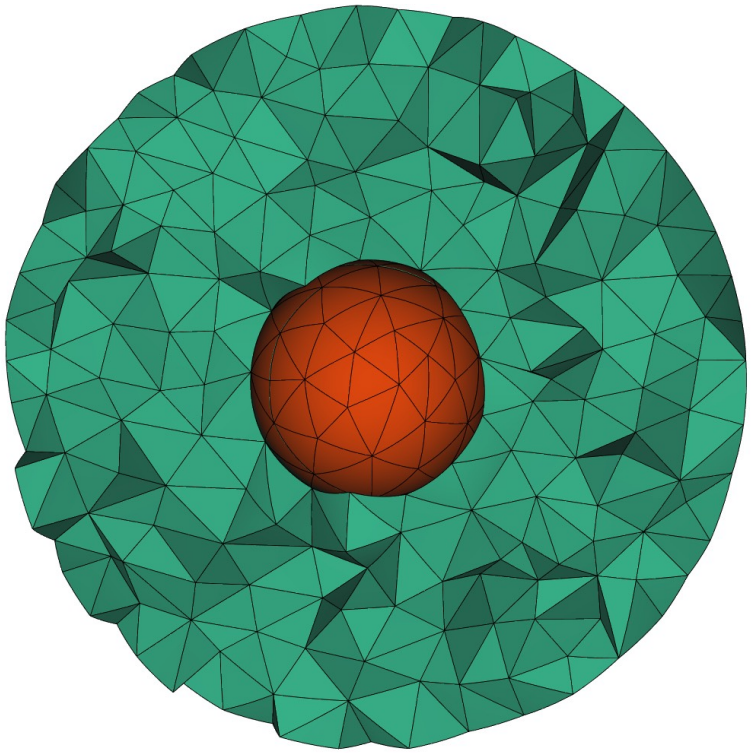
- Extension to P^2 of a closed-advancing layer method [Alauzet et al., EWC, 2015]

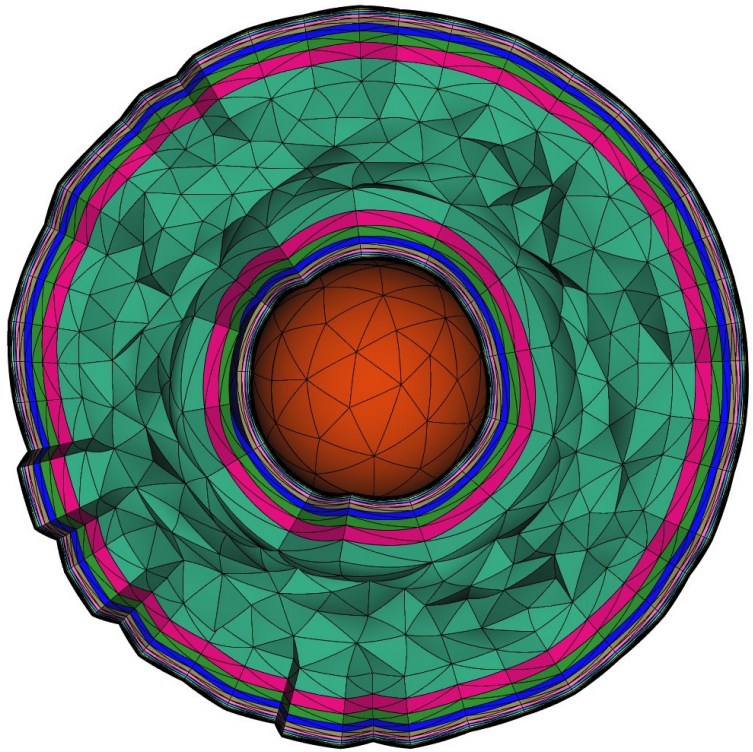
Features

- Start from an existing P^2 mesh
- Inflate the boundary while proposing a P^2 BL mesh
- Use of a connectivity-change moving-mesh method during the inflation
- Backtracking if necessary
- Guarantee of a final valid mesh

Sphere example

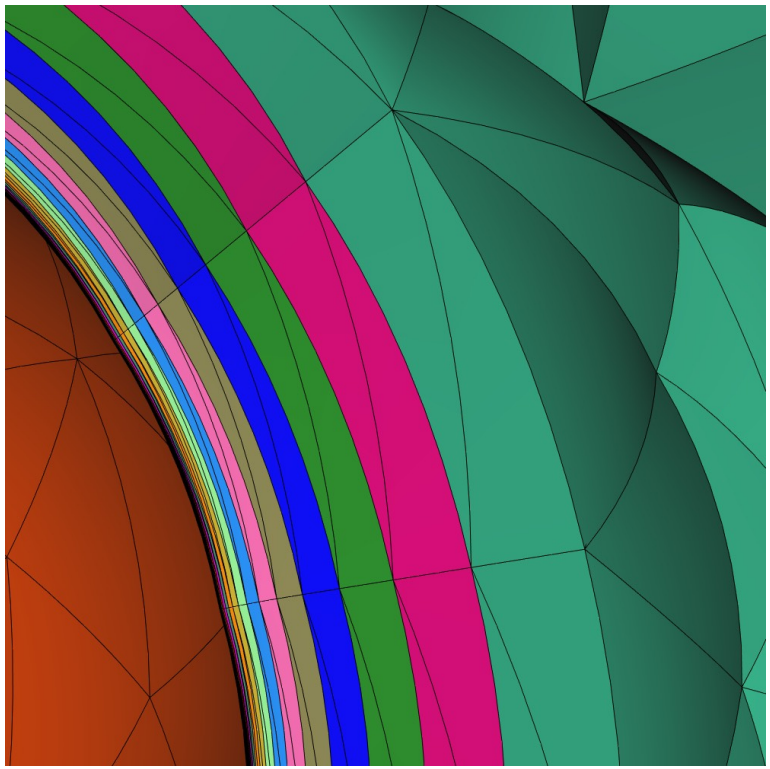
- Initial BL spacing is set to 0.0001
- Growth rate is 1.2
- 21 layers are generated

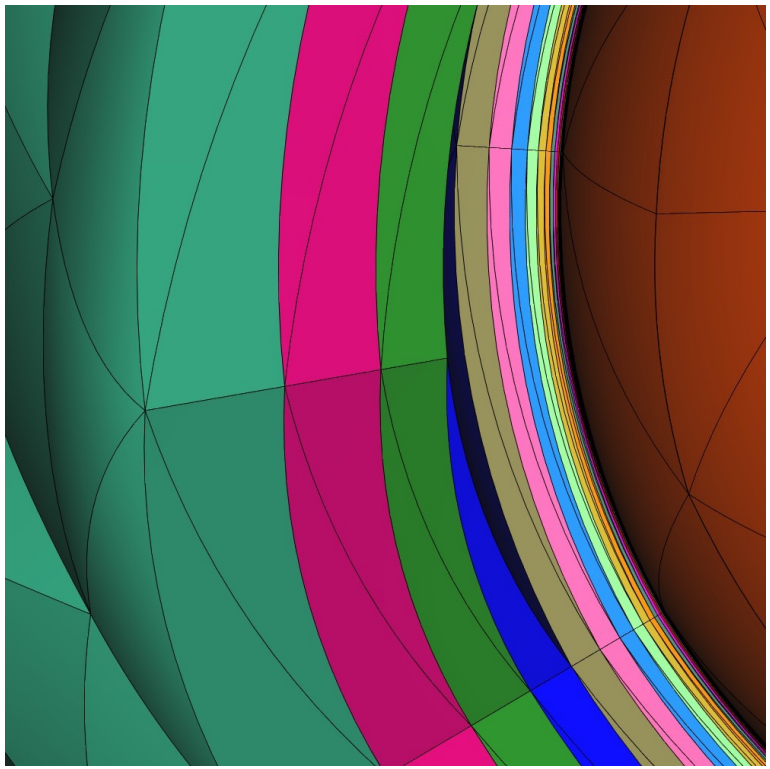




Sphere example

- Initial BL spacing is set to 0.0001
- Growth rate is 1.2
- 21 layers are generated





Quadratic mesh optimization: conclusion

Conclusion

- Development of vertices/nodes smoothing and connectivity-change **quality-based** optimization operators
- Insurance of a robust high-order Euler mesh generation
- Possibility of a high-order connectivity-change moving mesh methods
- Application to P^2 boundary layer mesh generation
- The methodology is as robust as for the P^1 context

Perspectives

- Extension to higher orders
- Investigation of mesh curving using a local approach
- Extension to anisotropy

---

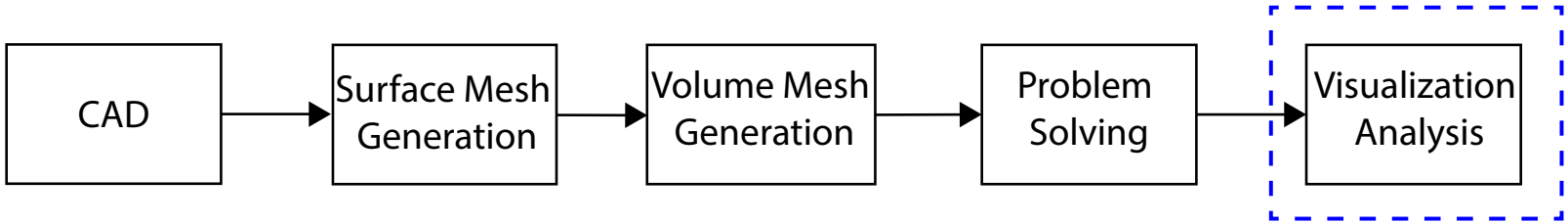
[3] Feuillet, Loseille and Alauzet, *P^2 mesh optimization operators*, 27th International Meshing Roundtable, 2018

[4] Feuillet, Loseille, Marcum and Alauzet, *Connectivity-change moving mesh methods for high-order meshes: Toward closed advancing-layer high-order boundary layer mesh generation*, AIAA, 2018

[5] Feuillet, Marcum and Alauzet, *A closed advancing-layer method for generating curved boundary layer mesh*, AIAA, 2019

[6] Feuillet, Loseille and Alauzet, *Optimization of P^2 meshes and applications*, under review, 2019

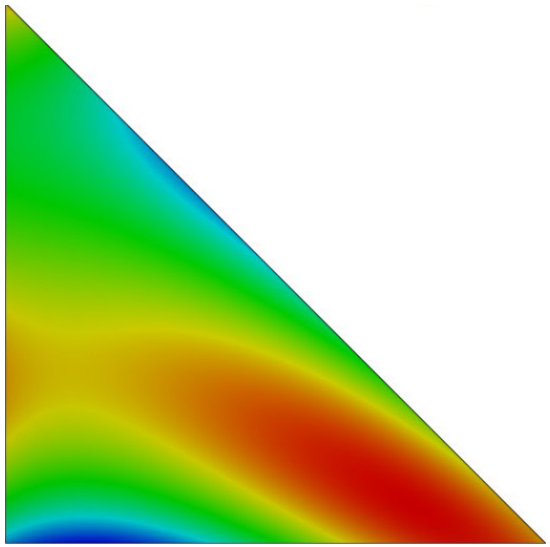
Outline

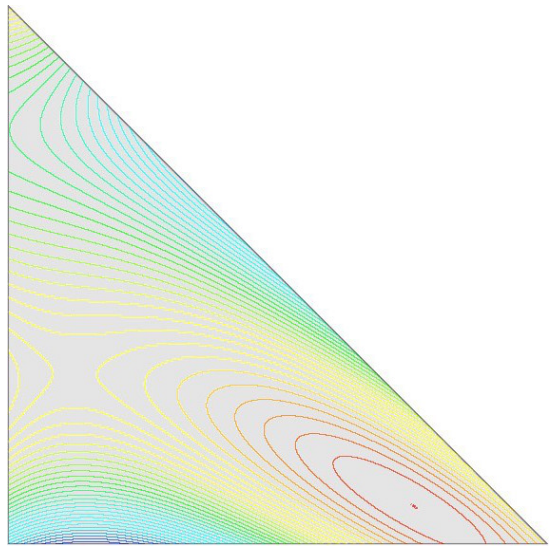


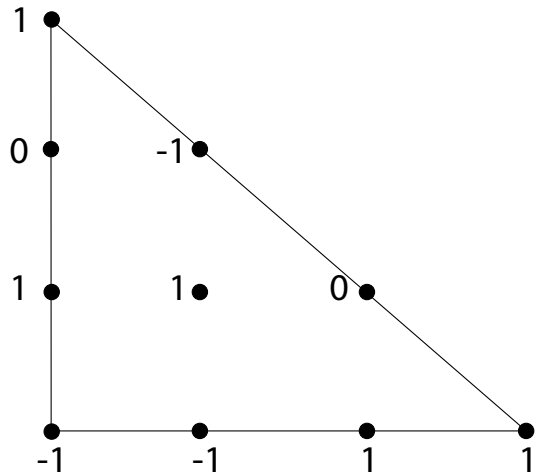
Visualization of high-order meshes and solutions

Motivations

- High-order entities (meshes and solutions) are not properly handled
- Entities are subdivided on CPU during a preprocessing step



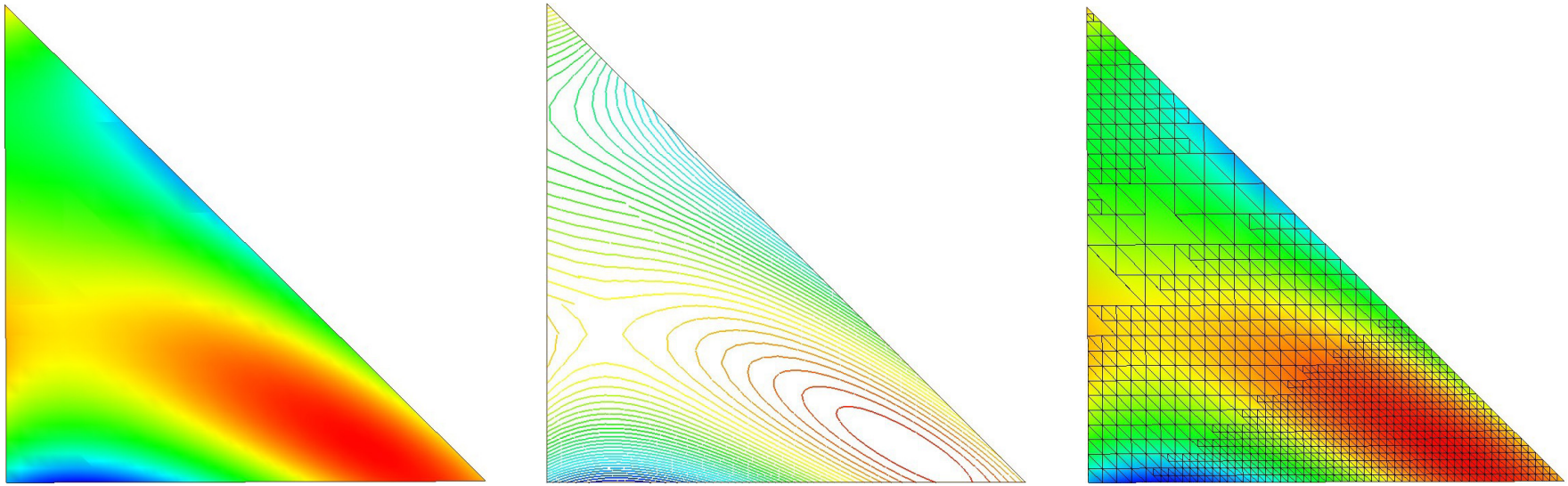




Visualization

Motivations

- High-order entities (meshes and solutions) are not properly handled
- Entities are subdivided on CPU during a preprocessing step



1858 triangles are generated in this case

Visualization

Focus on high-order finite elements representations

Related work

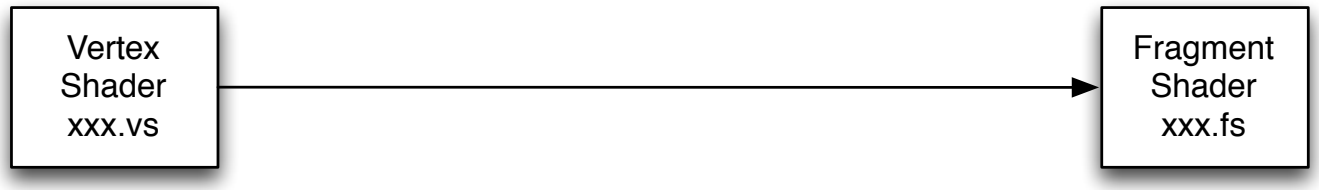
- Ray tracing techniques for pixel-exact solution rendering [Peiro et al., 2015]
  - $\Longrightarrow$ Resolution of nonlinear problems
  - $\Longrightarrow$ Greedy computations
- Adaptive subdivision [Schroeder et al., IEEE, 2006], [Remacle et al., IJNME, 2006]
  - $\Longrightarrow$ Still end-up with linear interpolation ...

Objectives

- Compromise between two approaches
- Solution that does not subdivide straight entities at all and subdivides curved entities on the fly on GPU
- Reliable rendering of high-order solutions

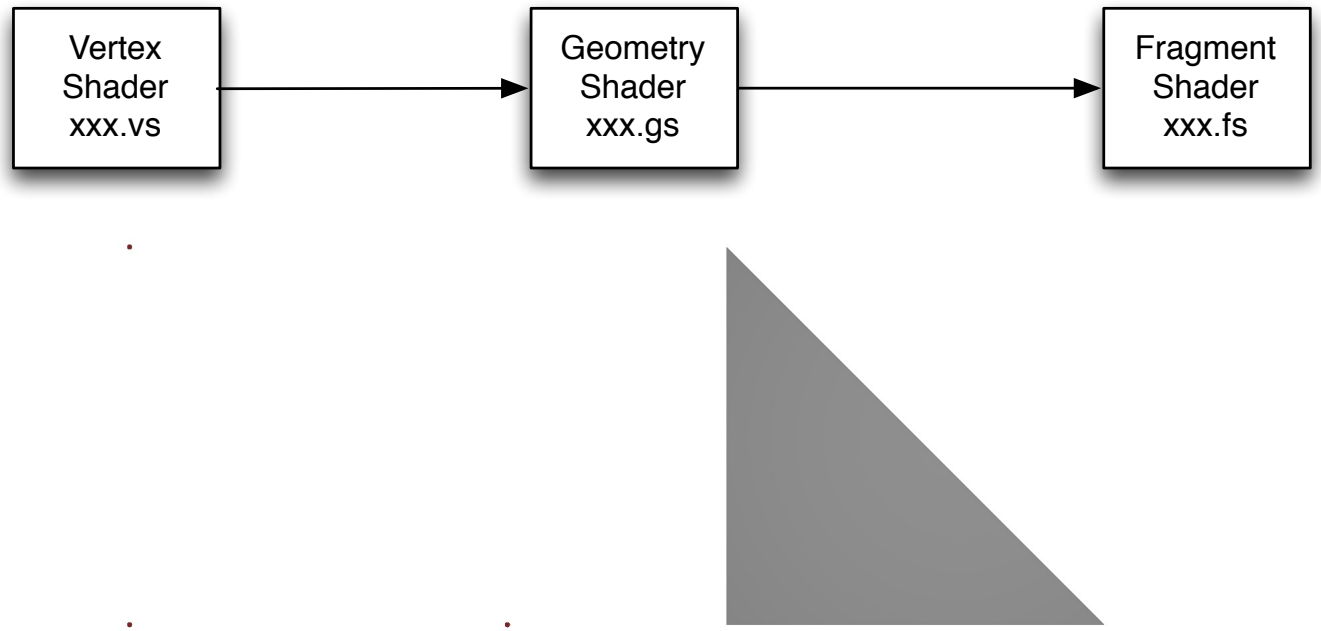
Curved surface elements visualization

- The visualization process relies on the use of the OpenGL 4.0 graphic pipeline that can be customized with up to five different shader stages
- Shaders are parts of code that are directly compiled on GPU
- Display of a point



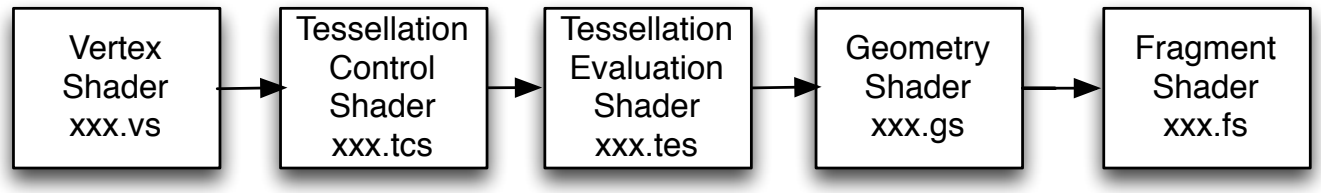
Curved surface elements visualization

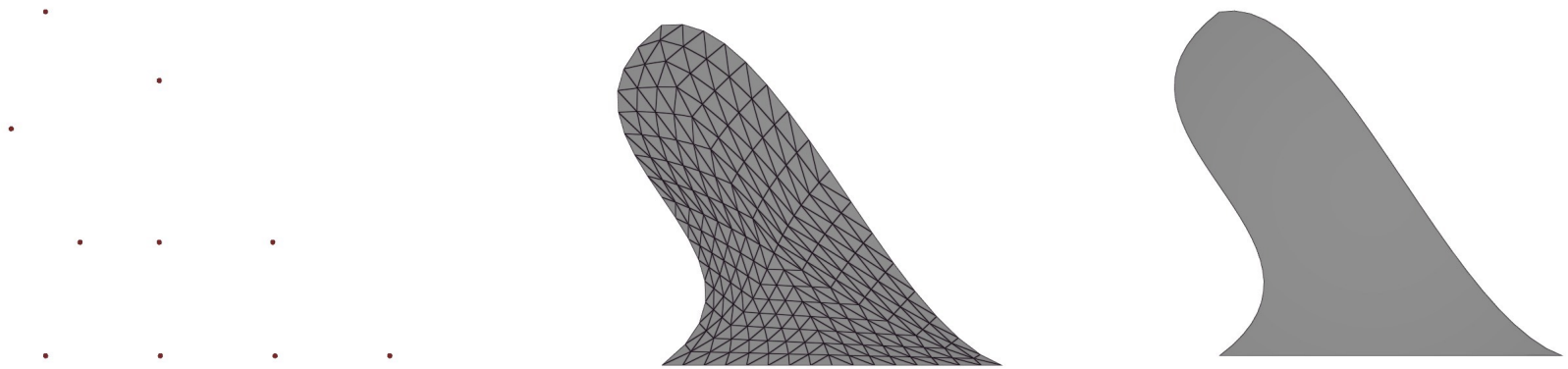
- The visualization process relies on the use of the OpenGL 4.0 graphic pipeline that can be customized with up to five different shader stages
- Shaders are parts of code that are directly compiled on GPU
- Display of a simple geometry (edg, tri)



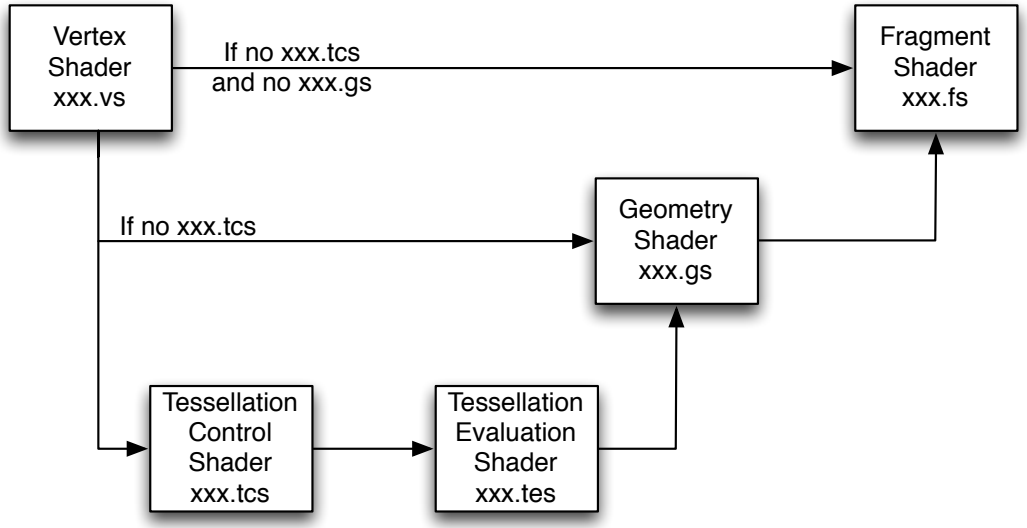
Curved surface elements visualization

- The visualization process relies on the use of the OpenGL 4.0 graphic pipeline that can be customized with up to five different shader stages
- Shaders are parts of code that are directly compiled on GPU
- Display of a complex geometry (high-order elements)





Curved surface elements visualization

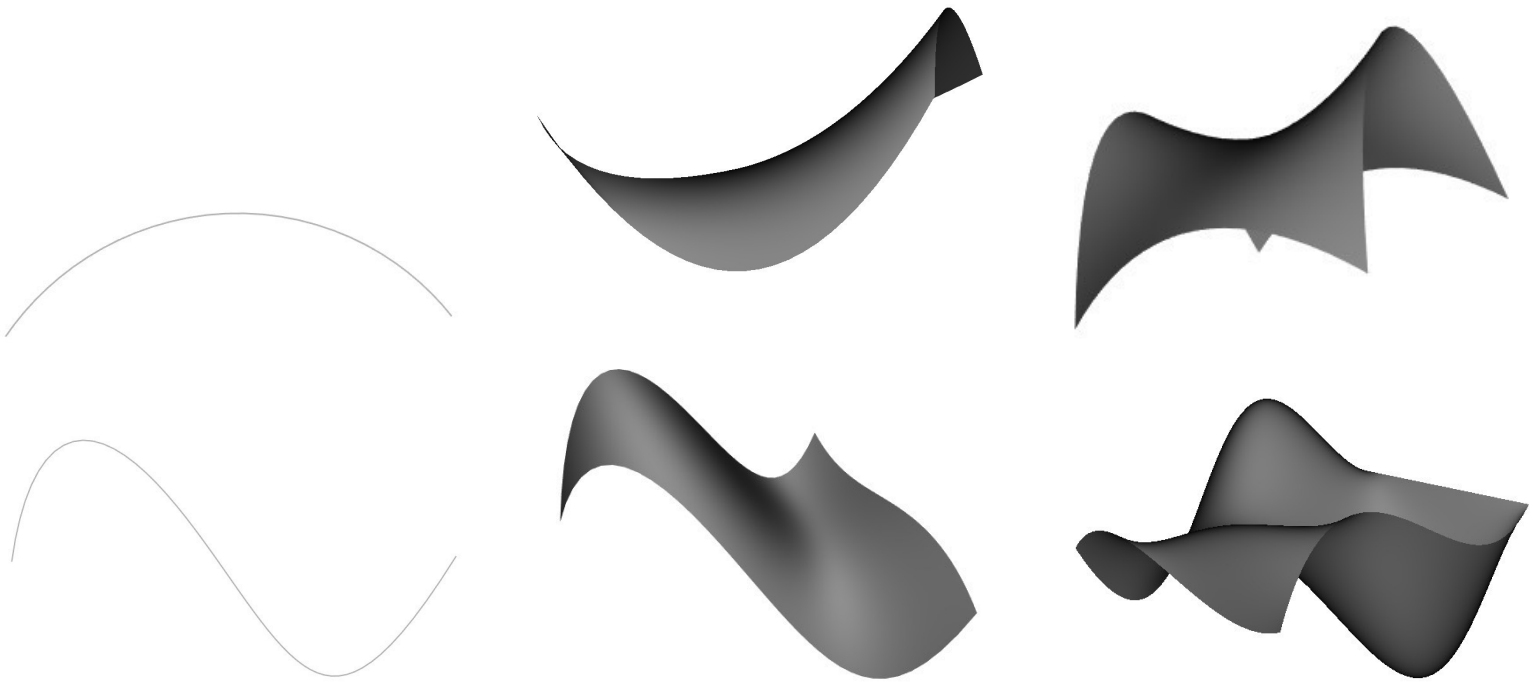


- The accuracy of the geometric approximation can be controlled thanks to a tesselation shaders
- Among built-in variables, we can pass trough our own variables:

$\Longrightarrow$ for a pixel we then know: (x, y, z), or (u, v), or primitive ids

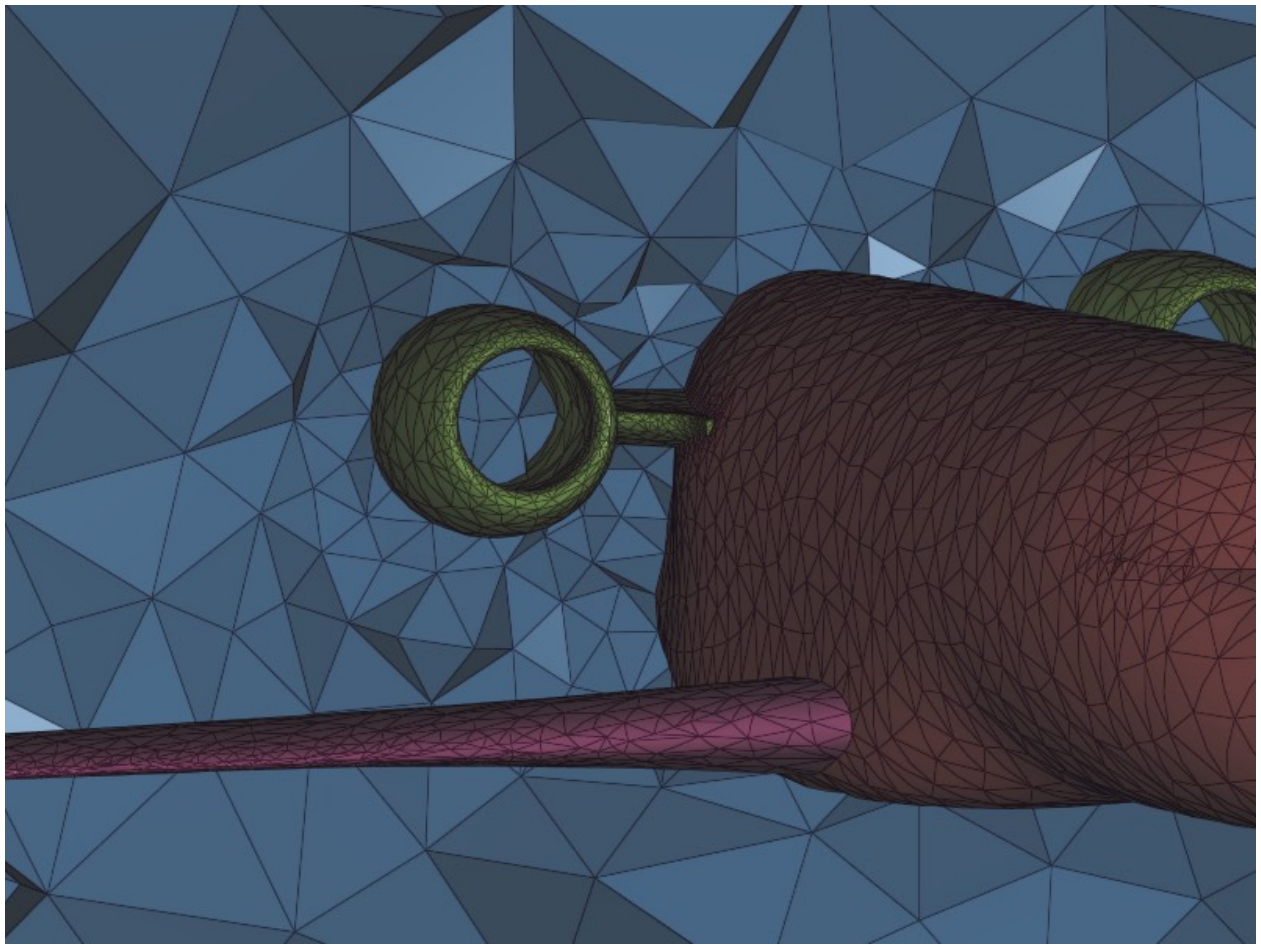
Curved surface elements visualization

- Display of elements of degree 2 (up) and degree 3 (bottom)
- From left to right: edge, triangle and quadrilateral



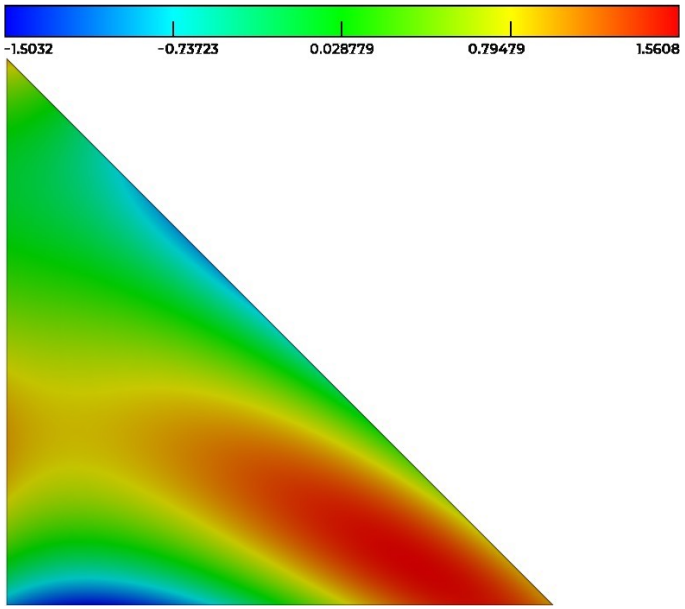
Volume element visualization

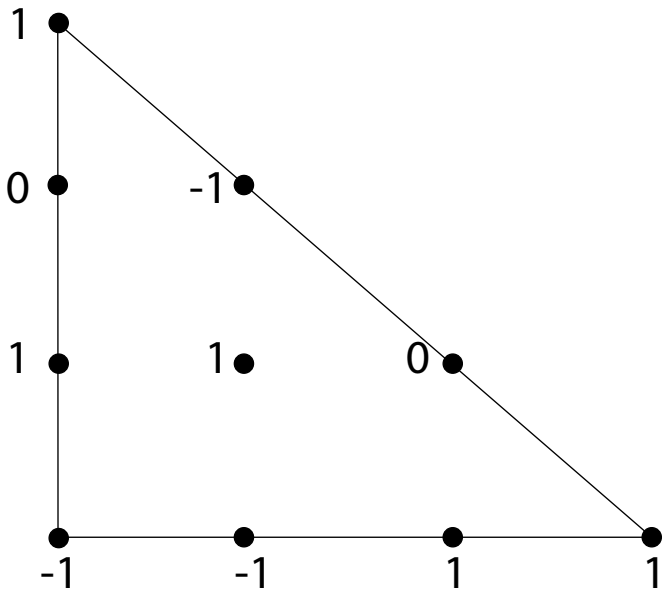
- The volume is visualized through cut planes
- Example of P^3 tetrahedral mesh



Almost pixel exact solution rendering

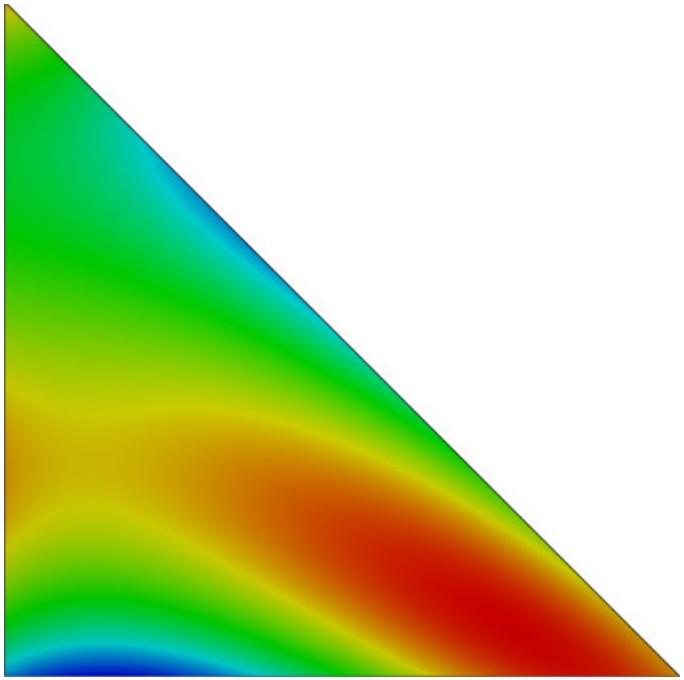
- Rendering of a P^3 solution
- No subdivision is needed

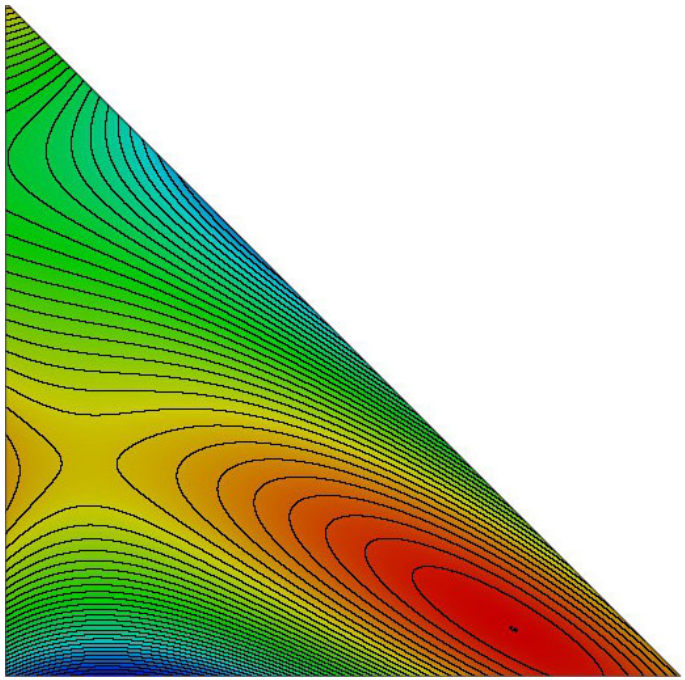




Almost pixel exact solution rendering

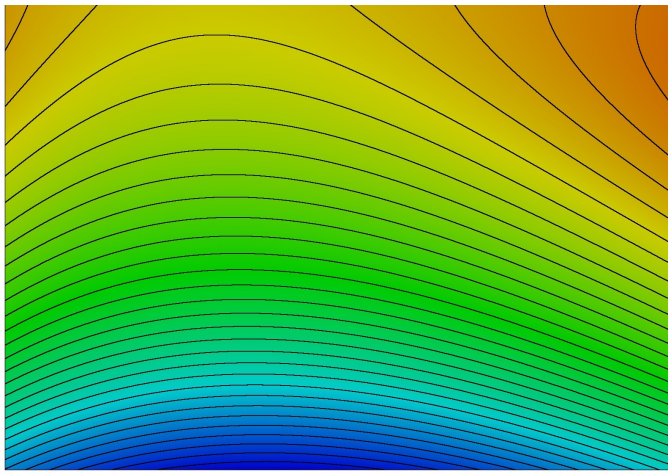
- Rendering of a P^3 solution
- No subdivision is needed

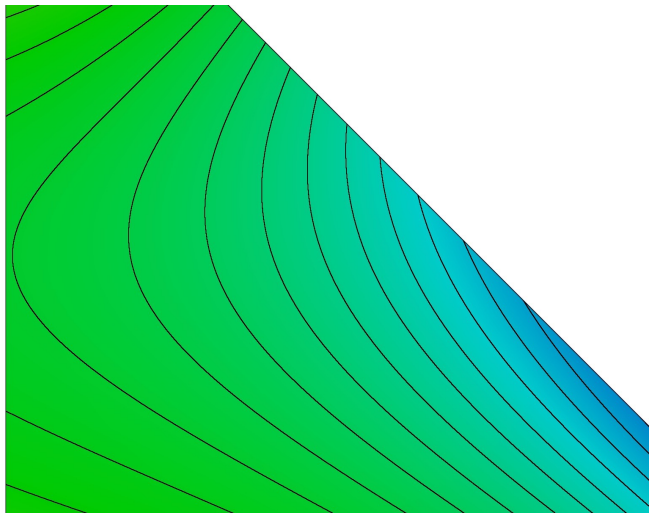




Almost pixel exact solution rendering

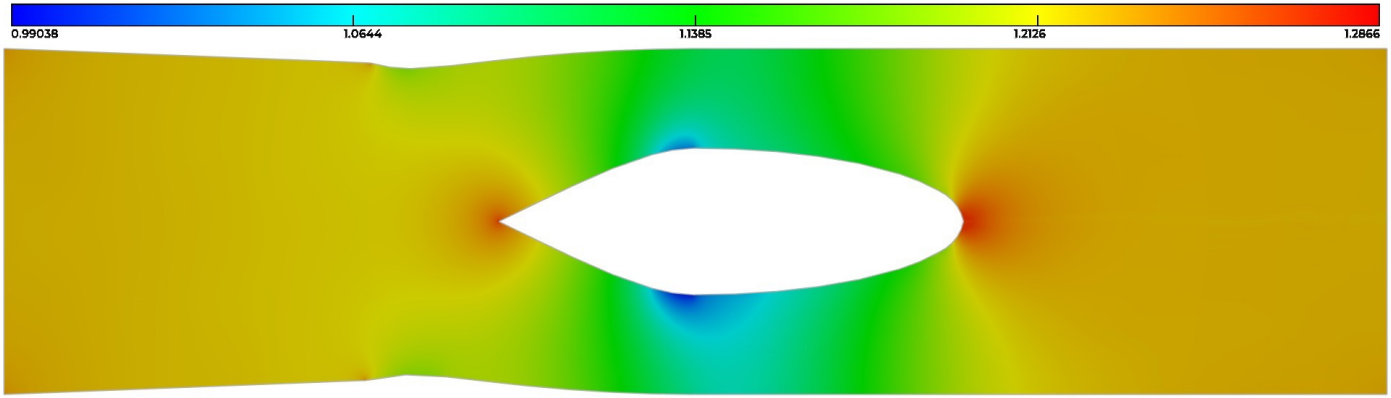
- Rendering of a P^3 solution
- No subdivision is needed

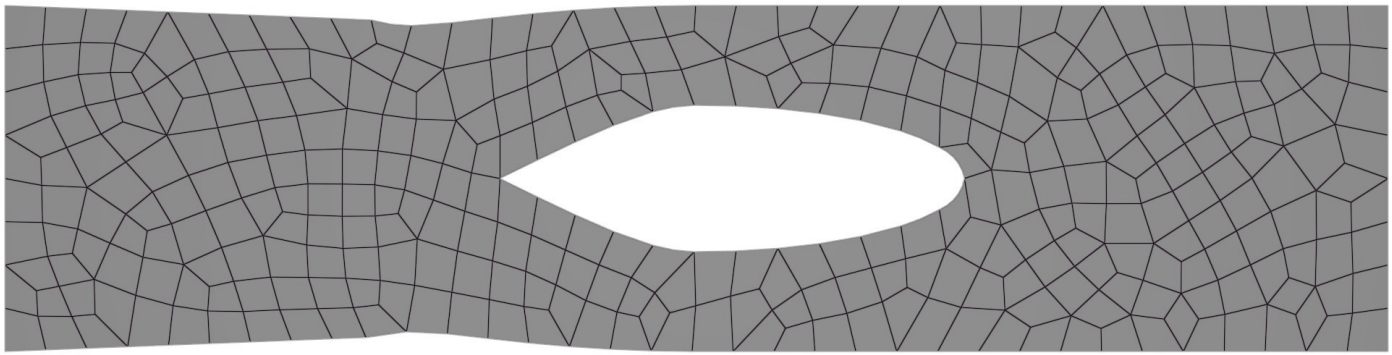




CFD example

- Rendering of a Q^{10} solution on a Q^2 mesh
- Simulation of Euler equations around a CFM56

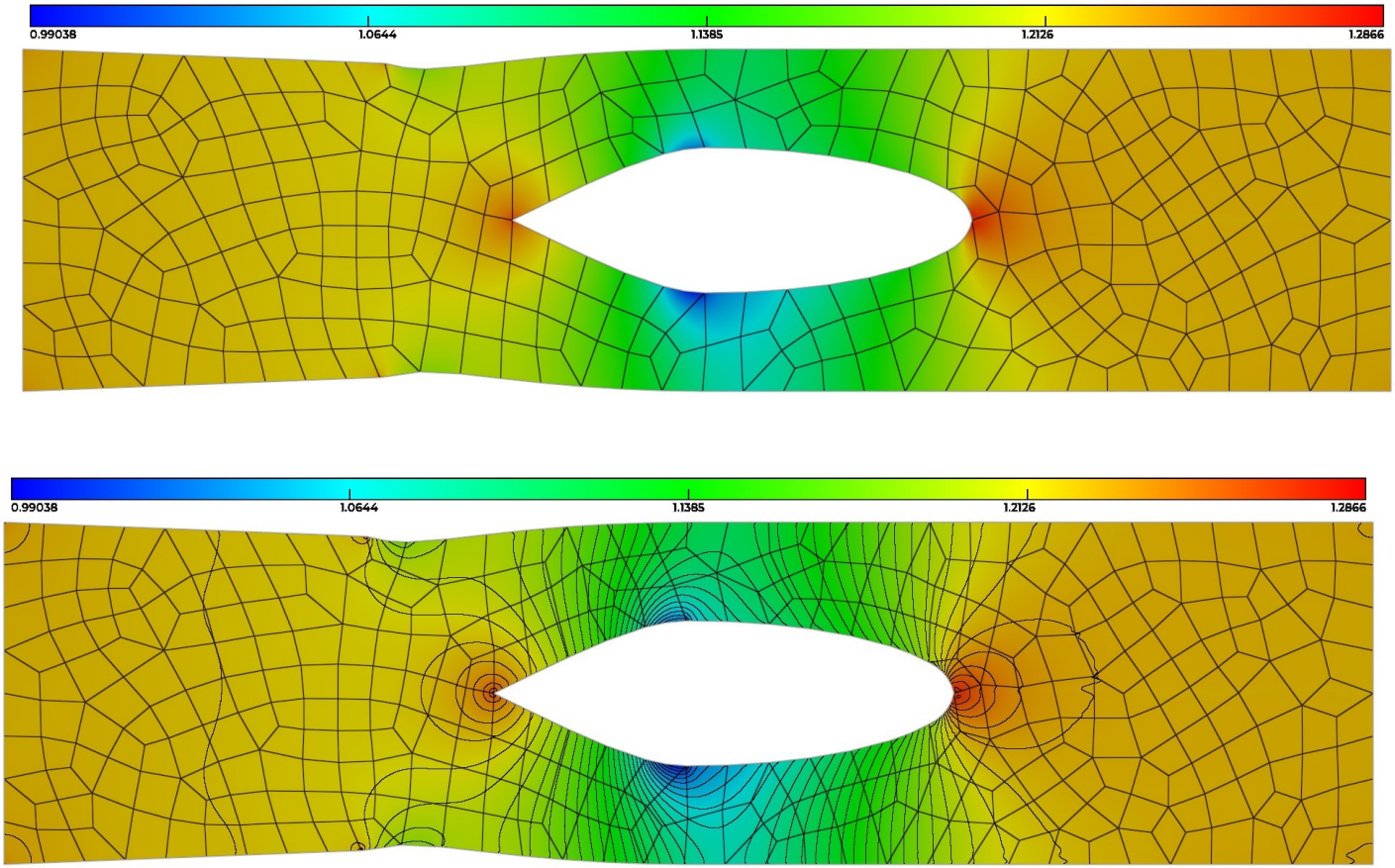




Courtesy of C. Peyret (ONERA) [Peyret, AIAA, 2017]

CFD example

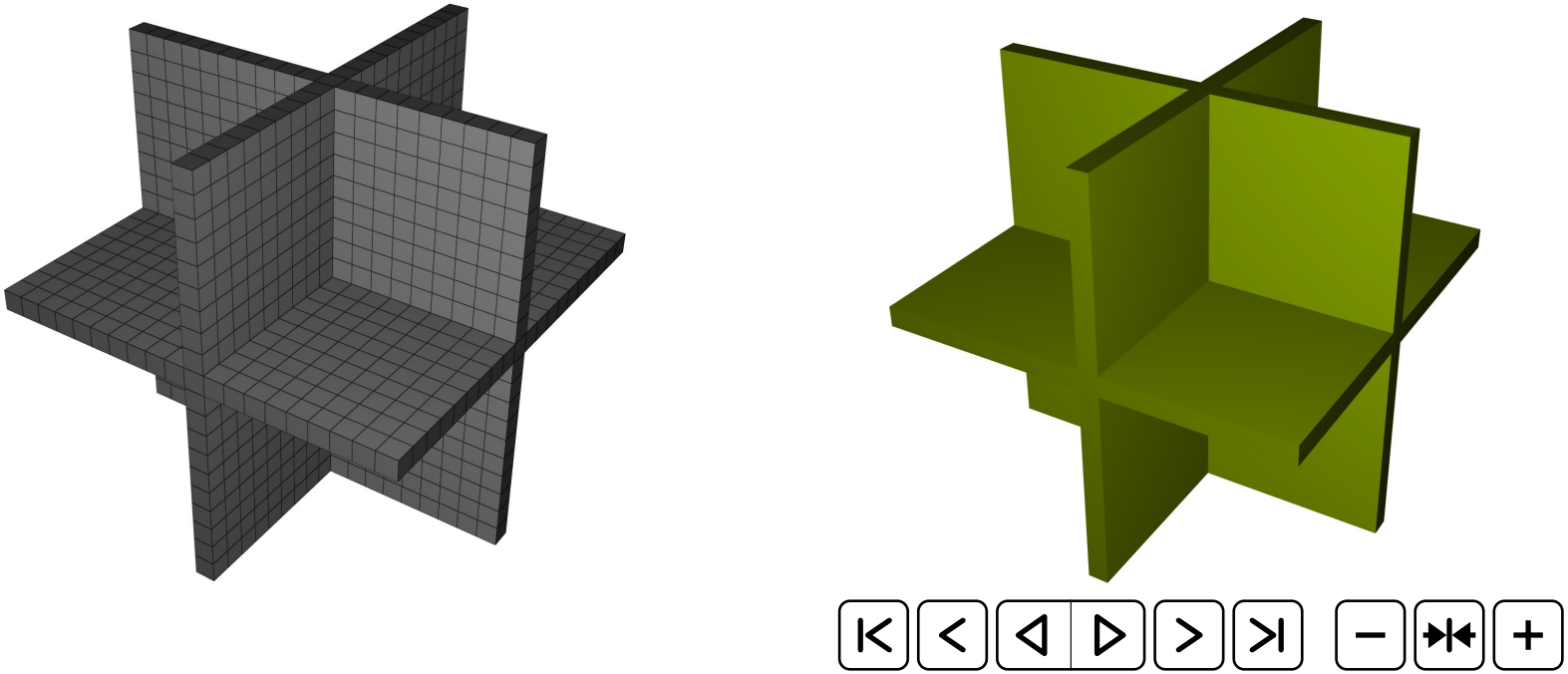
- Rendering of a Q^{10} solution on a Q^2 mesh
- Simulation of Euler equations around a CFM56



Courtesy of C. Peyret (ONERA) [Peyret, AIAA, 2017]

Wave propagation example

- Rendering of a Q^{10} solution on a Q^1 mesh
- Simulation of a wave propagation with a Perfectly Matched Layer



Courtesy of S. Impériale (INRIA) [Baffet et al., J. Sci. Comp., 2019]

Visualization: conclusion

Conclusion

- Full GPU-based rendering
- Pixel-exact rendering on straight elements (independently of the degree of the solution)
- Almost pixel exact for elements with non-linear geometrical mapping

Perspectives

- Volume rendering
- Handling of iso-surfaces
- Extraction of solutions induced on cut surfaces

[7] Loseille and Feuillet, *Vizir: high-order mesh and solution visualization using OpenGL 4.0 graphic pipeline*, AIAA, 2018

[8] Feuillet and Loseille, *On pixel-exact rendering for high-order mesh and solution*, submitted, 2019

General conclusion and perspectives

General conclusion and perspectives

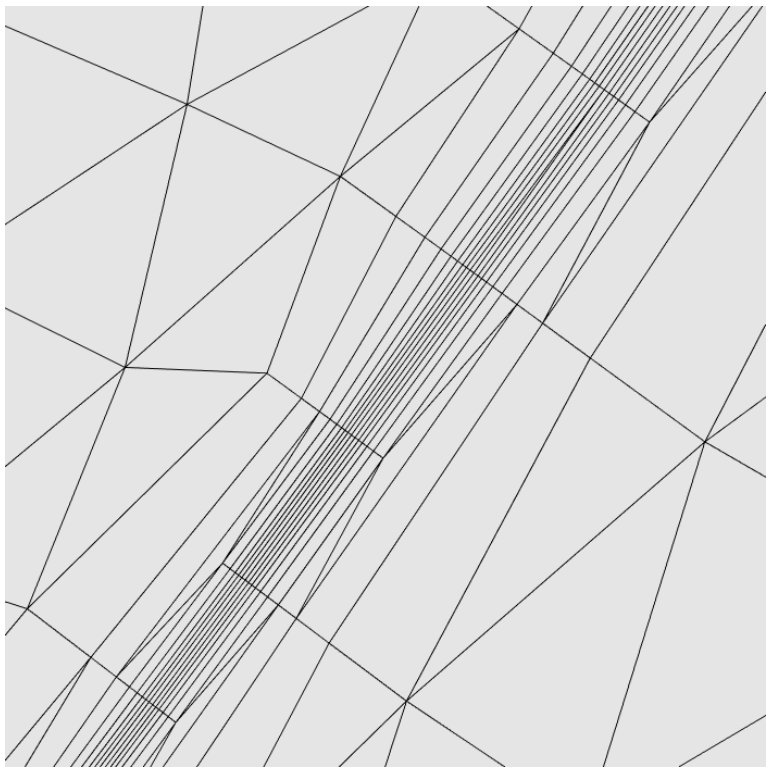
General conclusion

- Study of two alternatives to linear body-fitted representation
- The embedded approach avoids the meshing cost of complex geometries
  $\Longrightarrow$ Mesh adaptation enables to get back the loss of accuracy induced by the approach
- High-order meshes are more convenient for the solver
  $\Longrightarrow$ Presentation of a novel approach for generating high-order surface meshes
  $\Longrightarrow$ Design of optimization operators for quadratic meshes with applications to mesh curving, moving-mesh and boundary layer mesh generation
  $\Longrightarrow$ Setup of an efficient and accurate visualization process of high-order entities

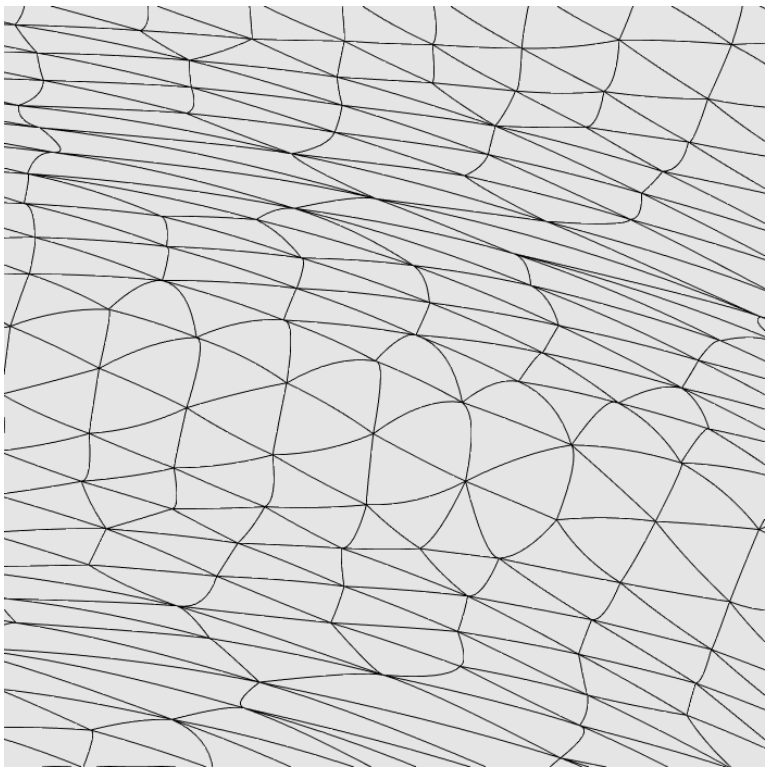
General conclusion and perspectives

Perspectives

- Hybrid metric-aligned mesh adaptation [Loseille, IMR, 2014], [Marcum et al., IMR, 2014]
- Curvilinear mesh adaptation [Zhang et al., IMR, 2018], [Aparicio-Estrems et al., IMR, 2018]
- Space-time meshes [Belda Ferrin et al., IMR, 2018], [Caplan et al., AIAA, 2019]



Hybrid mesh adaptation
(Thesis of L. Tenkès)



Curvilinear mesh adaptation
(Thesis of L. Rochery)

Thank you

Thank you for your attention !