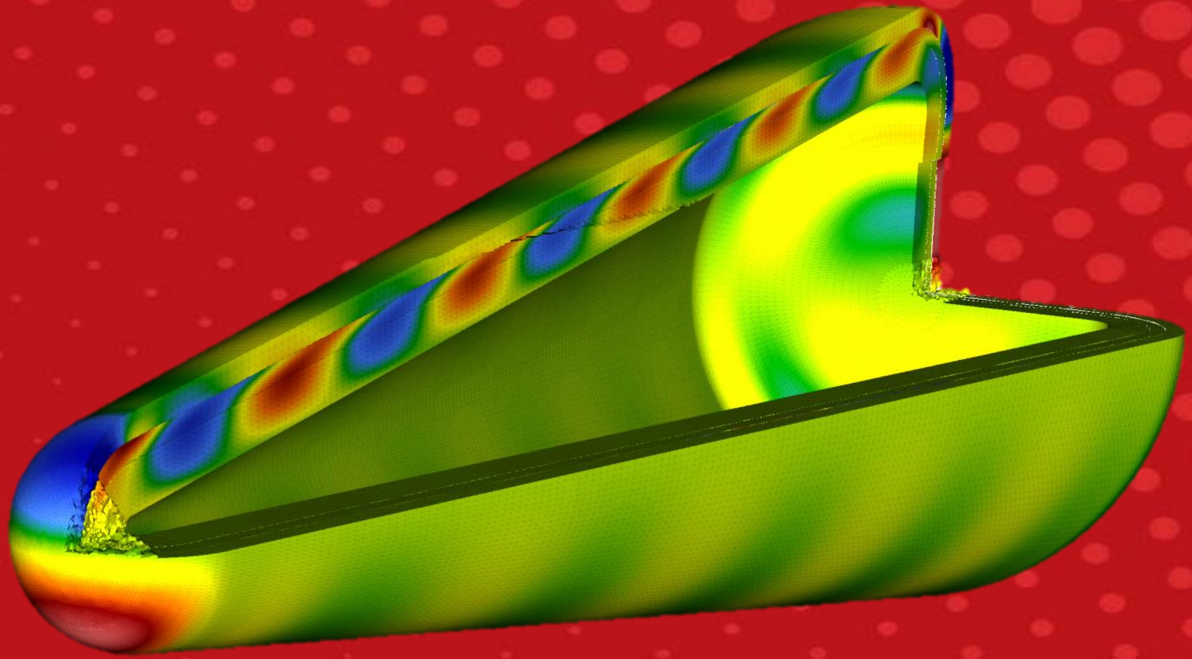
Framework for automatic meshing applied to 2D and 3D Electromagnetic simulations

FROM RESEARCH TO INDUSTRY



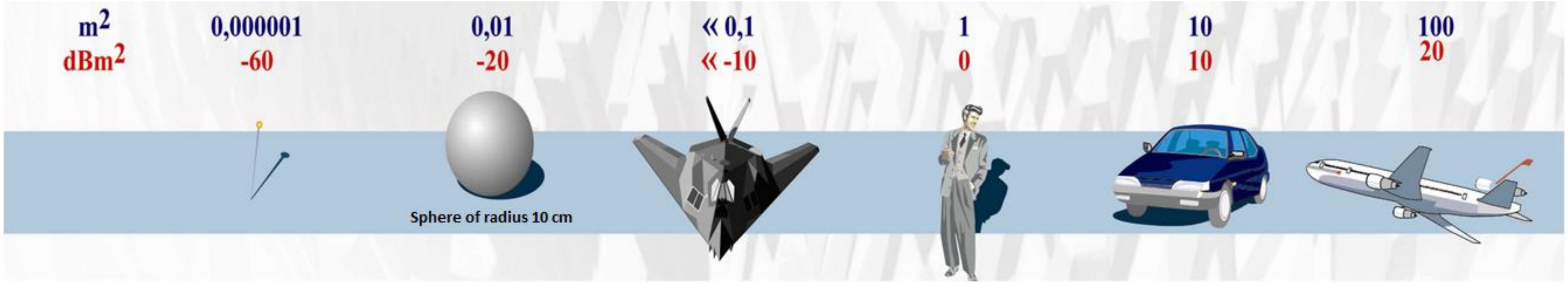
Fabien VIVODTZEV, Thierry HOCQUELLET, Matthieu LECOUVEZ

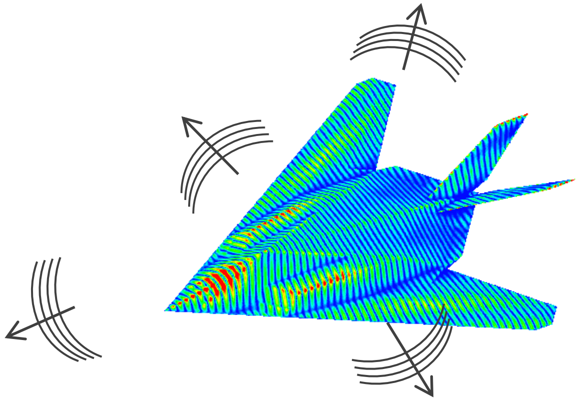
Eccomas Congress 2020 & 14th WCCM
Virtual Congress 11-15 January, 2021

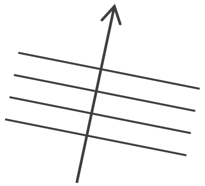
Computation of the Radar Cross Section (RCS) of 3D objects

Computation of the RCS of stealth objects is numerically expensive and challenging

3D objects covered by multiple (usually thin) layers of materials







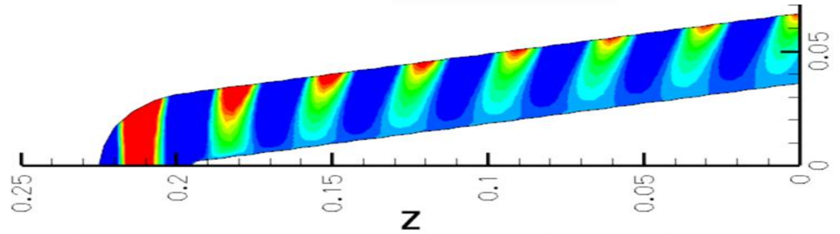
- Requires highly accurate solution of harmonic Maxwell's equations in an unbounded domain
- Objects may be electrically large
- Materials may have large permittivity/permeability

Computationally challenging

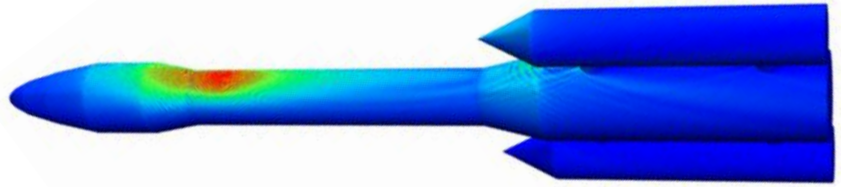
The numerical methods implemented at CEA/Cesta lead to different mesh requirements

Depending on the nature of the problem to solve, the CEA has implemented different numerical methods

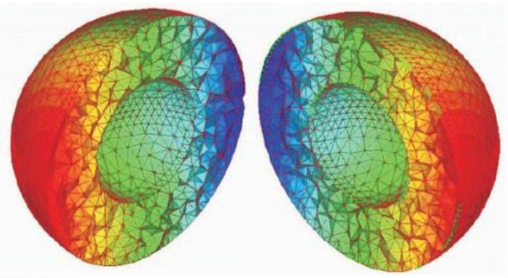
- **Axisymmetric finite element code**
  - Requires 2D meshes (triangles) with a description of the materials for each cell



- **Method of Moments code**
  - Requires 3D surface meshes (triangles) with a description of the interfaces of the materials



- **Full 3D finite element code**
  - Requires 3D volume meshes (tetrahedrons) with a description of the materials for each cell



The complexity for generating the meshes greatly increases with the complexity of the code capabilities (longer computational time but <u>also engineering time</u>)

For some applications, the geometry is axisymmetric and only the materials properties are fully 3D

➔ How to optimize the generation those meshes, respecting the mesh constrains imposed by the physical phenomenon ?

Mesh constrains for electromagnetic problems

It is well known in the literature that the cell size h must be inversely proportional to the frequency f of the wave

Speed of light in free space — Wavelength in the material

Usually, for finite element codes we target $$h \approx \frac{c}{f \alpha N} = \frac{\lambda_{mat}}{N}$$

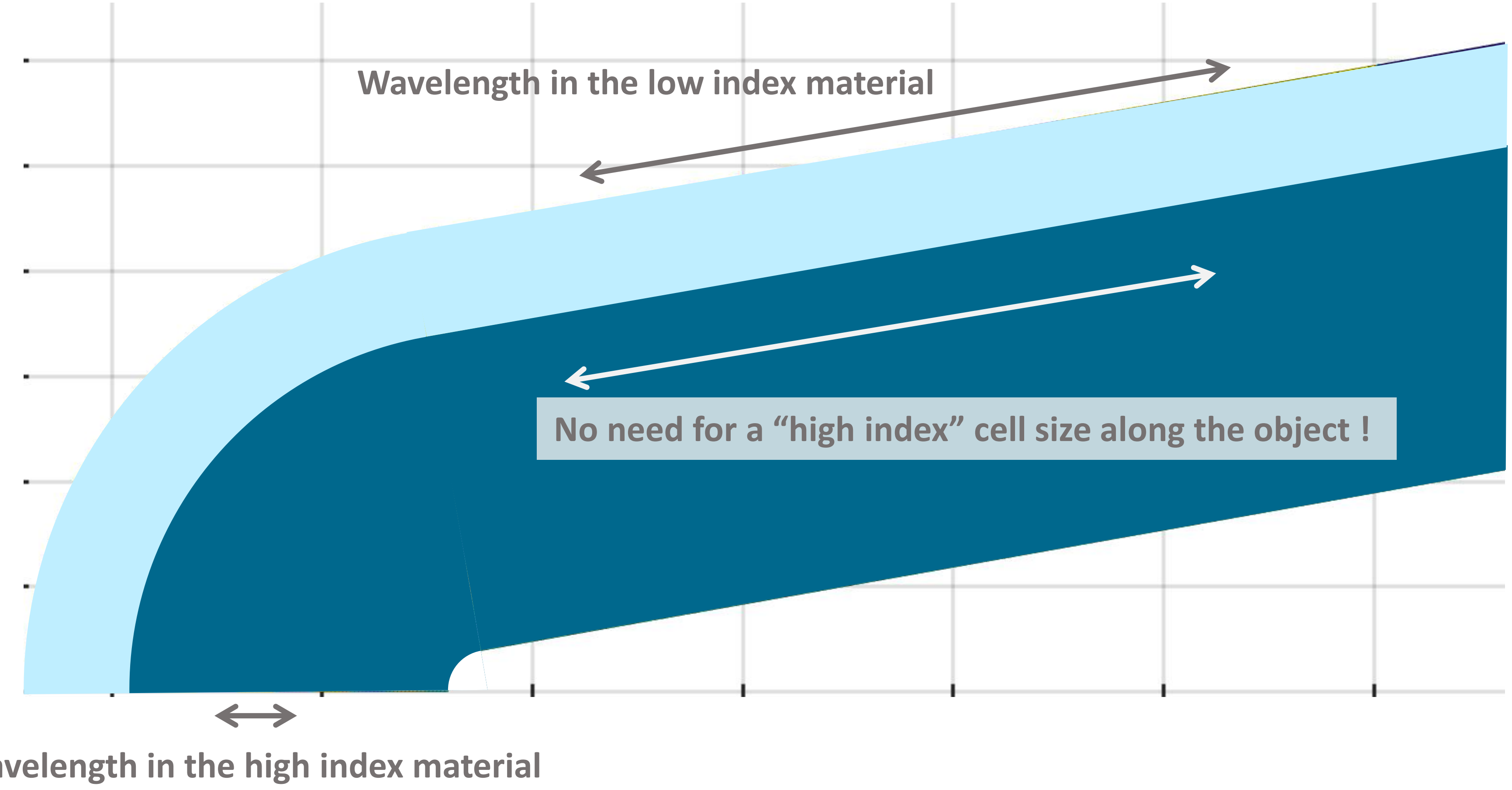
Frequency of the wave — Number of mesh points per wavelength

EM index of the material

▶ **The cell size depends on the electromagnetic properties of the material (high index materials require finer meshes)**

However, for large objects with thin layer of high index materials, one can take advantage of the properties of the solution to adapt the mesh

Example for two layers
- One low index
- One high index



We gain a lot on the size of the mesh by using **anisotropic** meshes

How to generate easily “good” meshes for our EM applications?

The existing tools can handle 2D anisotropic meshes, although it can be cumbersome to describe them in CAD tools

- It becomes very complicated for 3D meshes

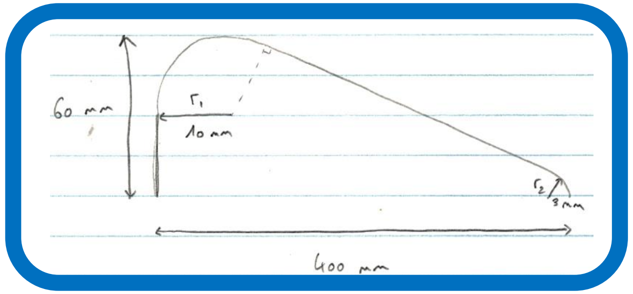
We describe in this talk a toolchain for generating 3D anisotropic meshes, respecting the requirements for a high accuracy when used in our 3D finite element code. This chain includes the following steps

- Generation of the geometry
- Build of a 2D anisotropic mesh
- Extrusion of the mesh in 3D
- Local remeshing of the generated mesh to increase its quality
- Mesh verification tool

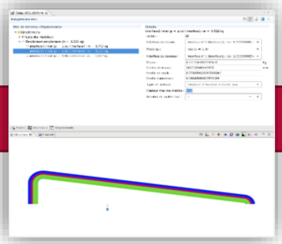
We finally present in conclusion some results by the 3D EM code using the improved meshes

Generation of the geometry

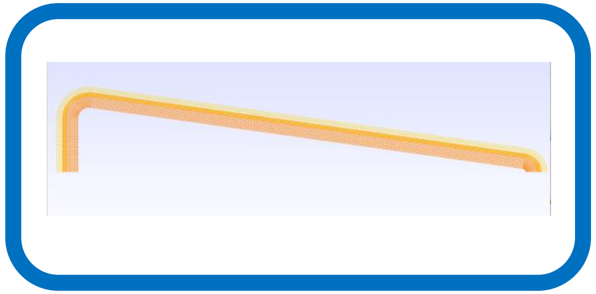
- **Implementation of a software to design draft geometries of 1D-shape (FORME)**
  - Modeling tool adapted to the design of 1D-shapes with thin stacked layers and material properties
  - Software architecture : JAVA, Eclipse, VTK, EMF modeling
- **To precisely handle geometric primitives required for the EM and others**
  - Domain specific geometric descriptors (types of splines, circle arc, segment …)
  - Analytic description of the profils
  - Ease the geometric exchange between physics based on a unique format in order to optimize the iterative design
- **To help the**
  - Weighted stretching of section to constrain global lenght
  - Automated sizing of sections between interfaces
  - Precise mass computation of thin layer of materials



1D drawing



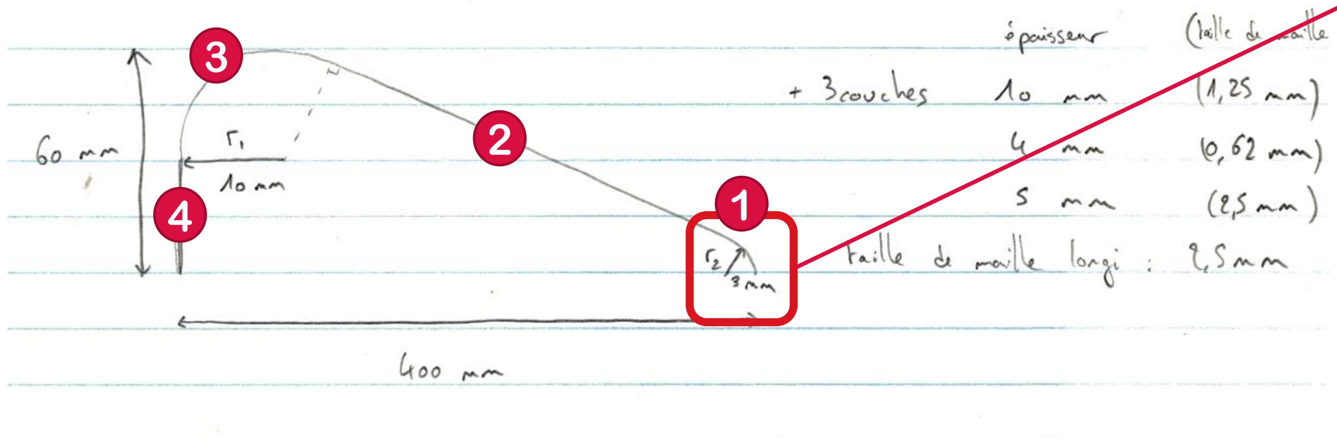
FORME

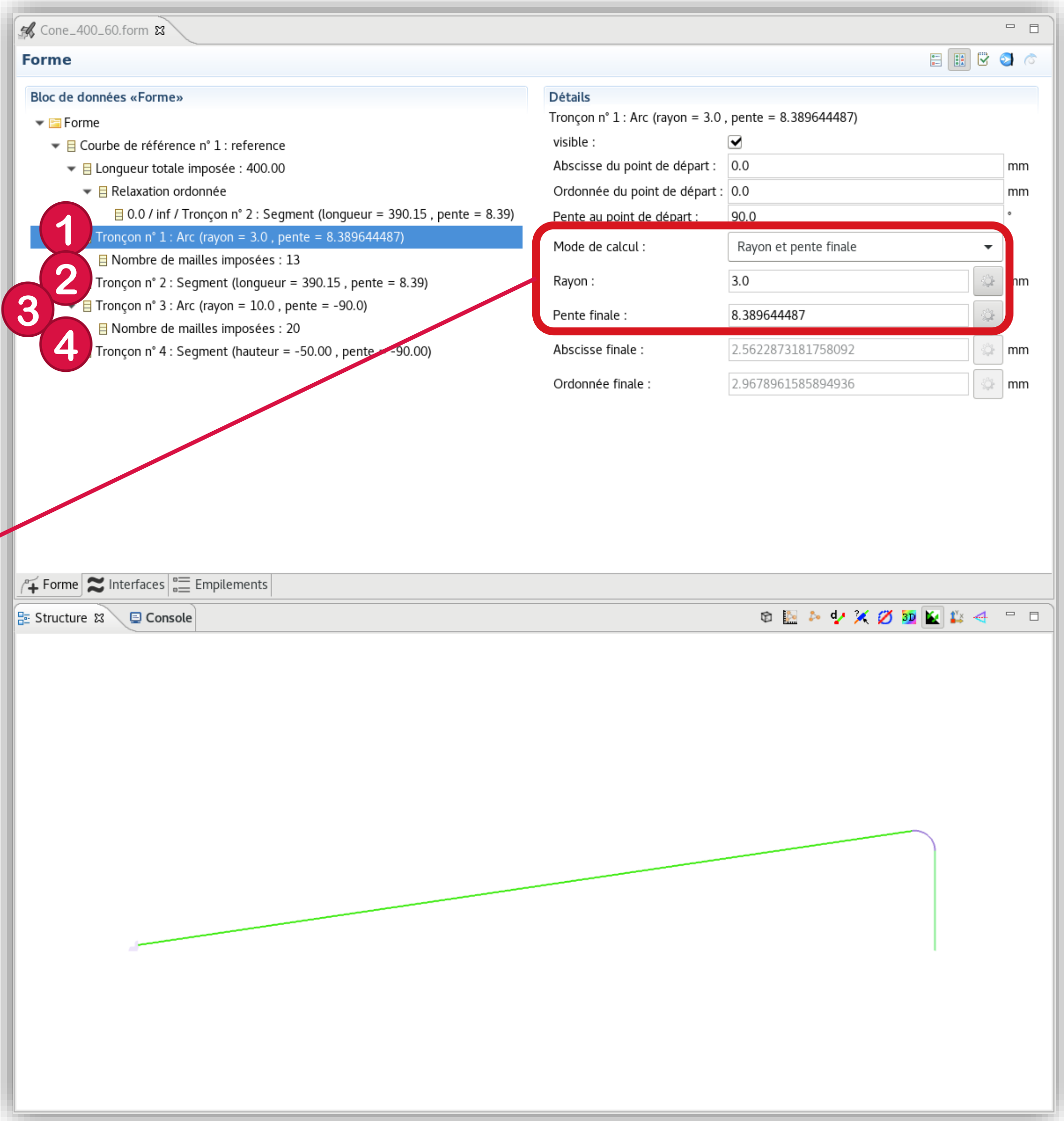


2D mesh

Use case on an axysymetric sphere-cone geometry :

- Length : 400 mm
- Width : 60 mm
- Back radius : 10 mm
- Front radius : 3 mm
- 3 layers of materials
  - Layer 1 : 10 mm (cell size : 1,25 mm)
  - Layer 2 : 4 mm (cell size : 0,62 mm)
  - Layer 3 : 5 mm (cell size : 2,5 mm)
- Cell size along the axis : 5,2 mm

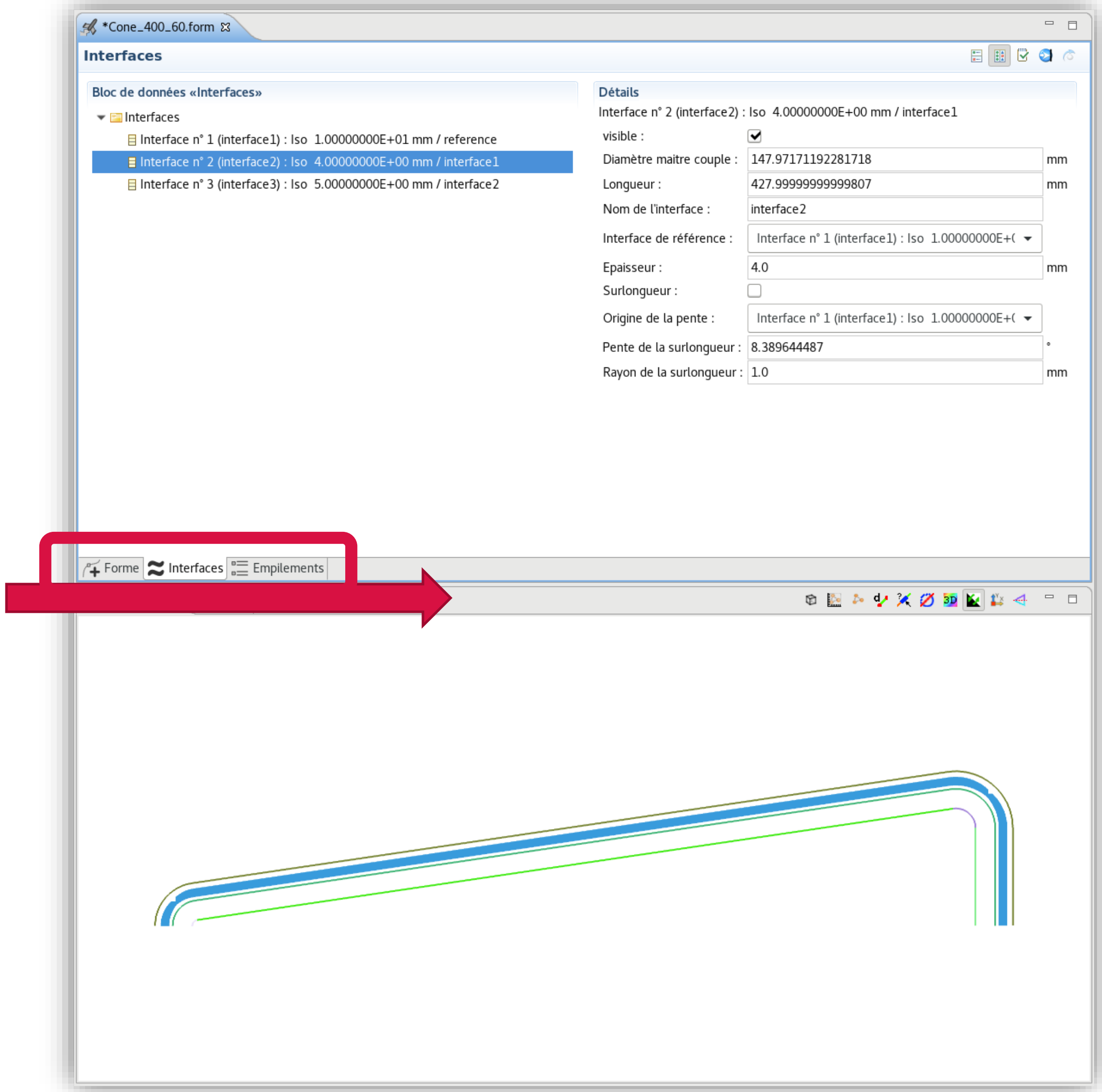




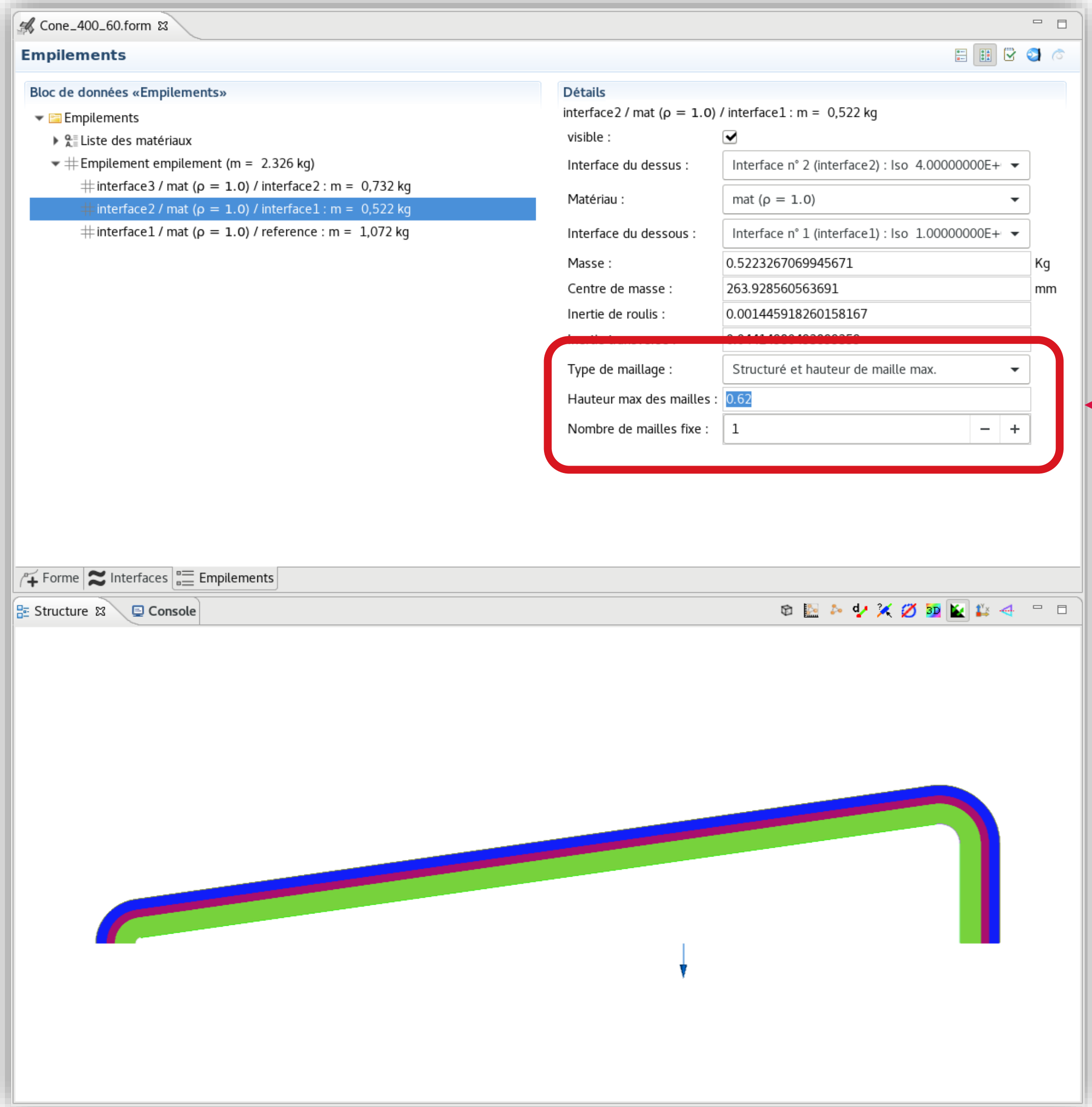
Generation of the geometry : interfaces based on the reference

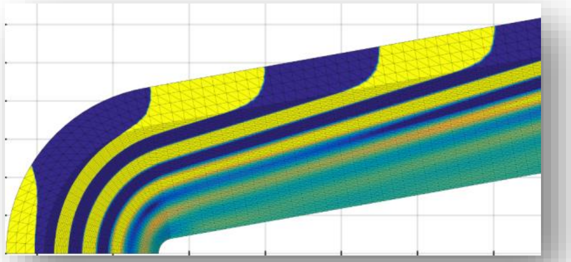
Definition of the interfaces in order to build the layers

Generation based on thickness laws between layers



Generation of the geometry : material stacking and mesh rules

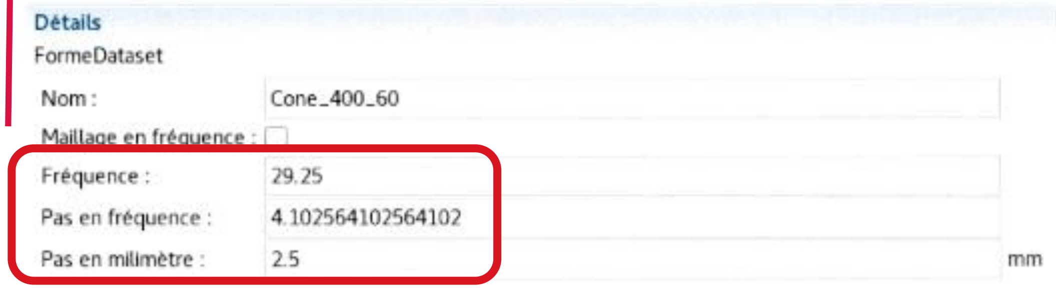




Mesh rule in the thickness

- Max cell size in the thickness
- Fixed number of cells in the thickness
- Free mesh

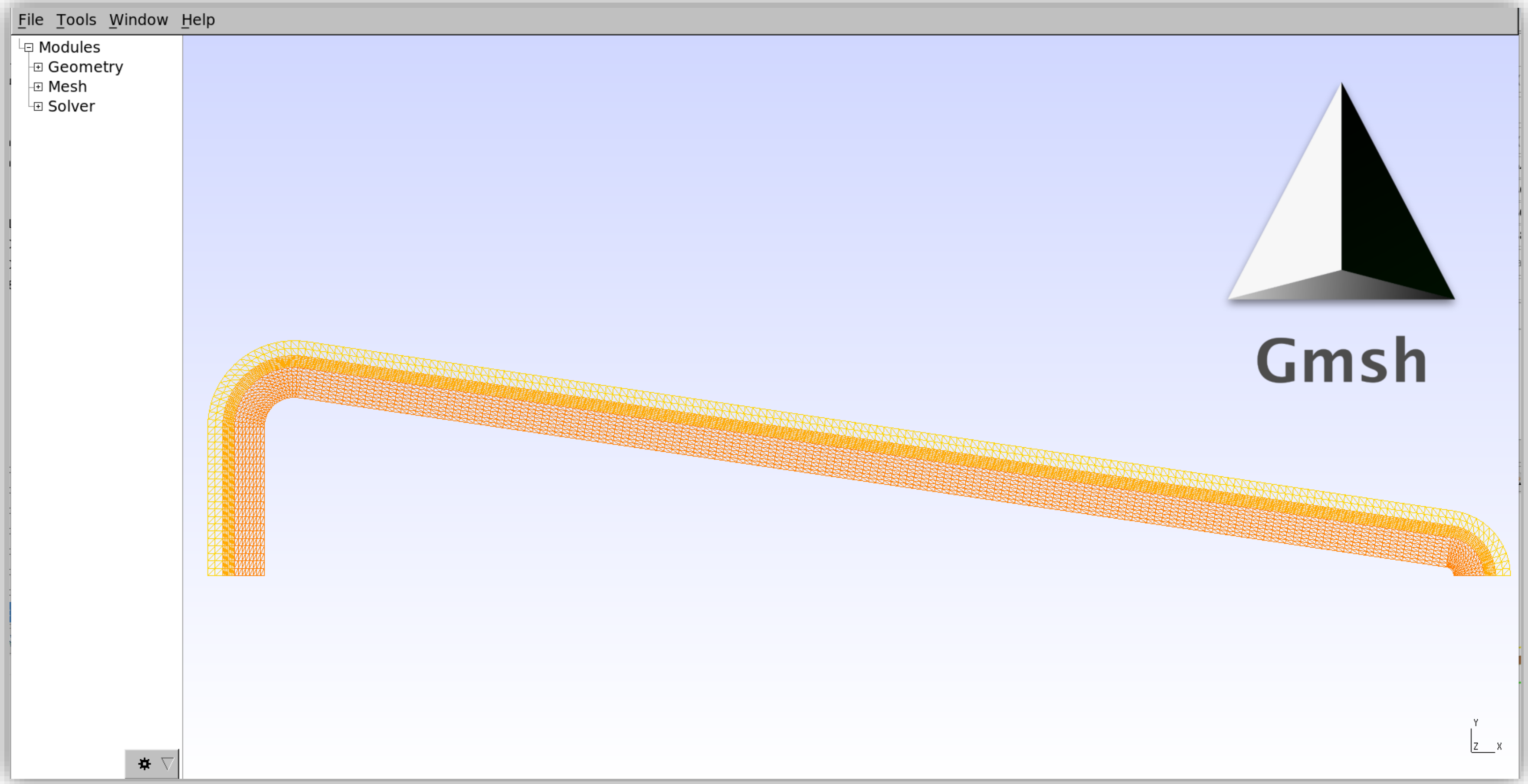
Mesh rule along the axis



Automated generation of the 2D axisymmetric ruled mesh

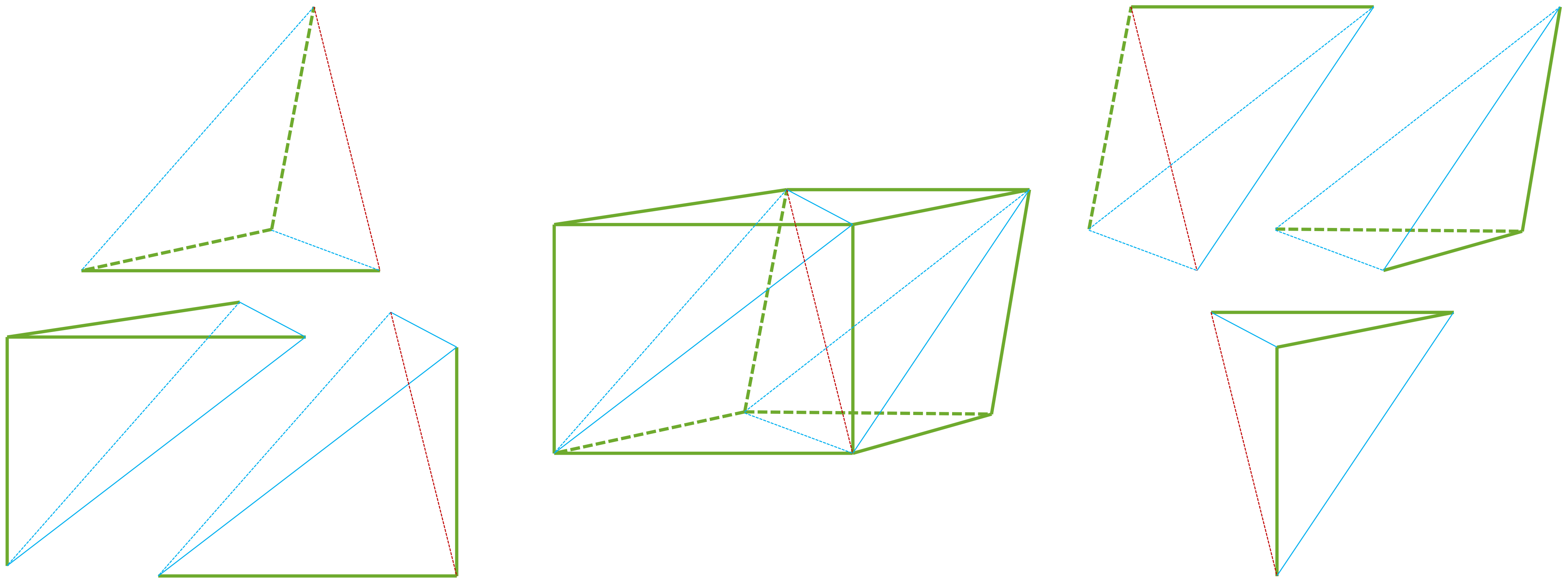
- GMSH geometry generated from the modeling tool FORME with specific mesh rules of the EM code

- Automated generation of the mesh with GMSH

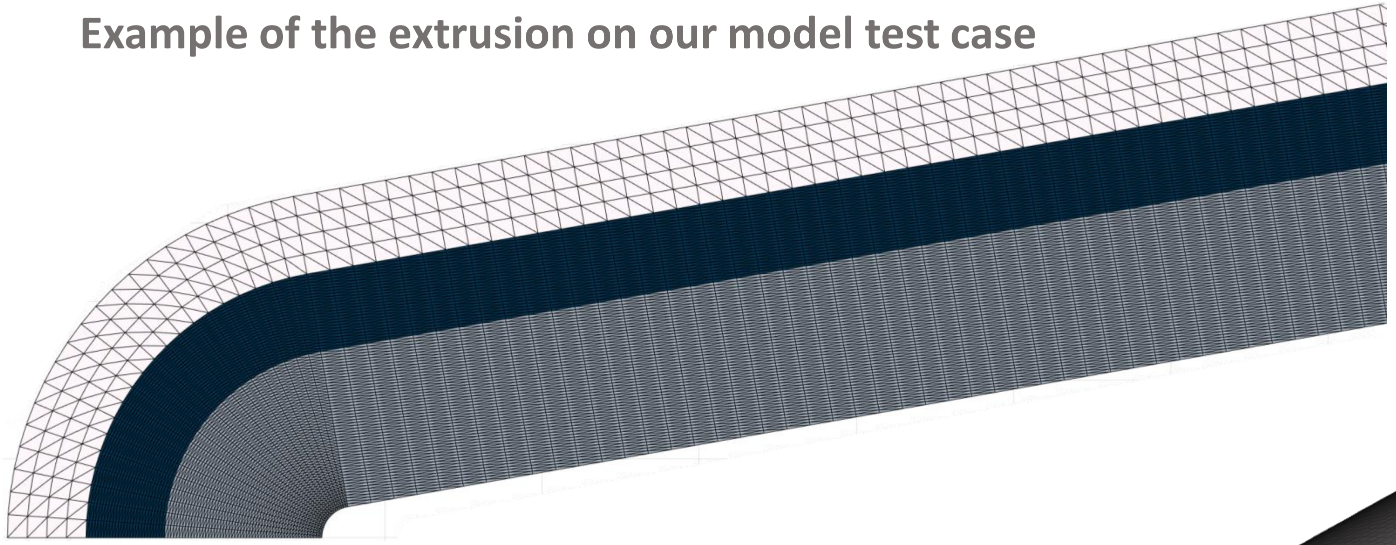


The easiest way to extrude a 2D ruled mesh is by meridian:

- The triangles are paired in order to create quadrangles
- The whole 2D mesh is extruded by an angle θ
- Each quadrangle generate a hexahedron (except quadrangles on the axe)
- Each hexahedron is split into 6 tetrahedrons in a consistent way

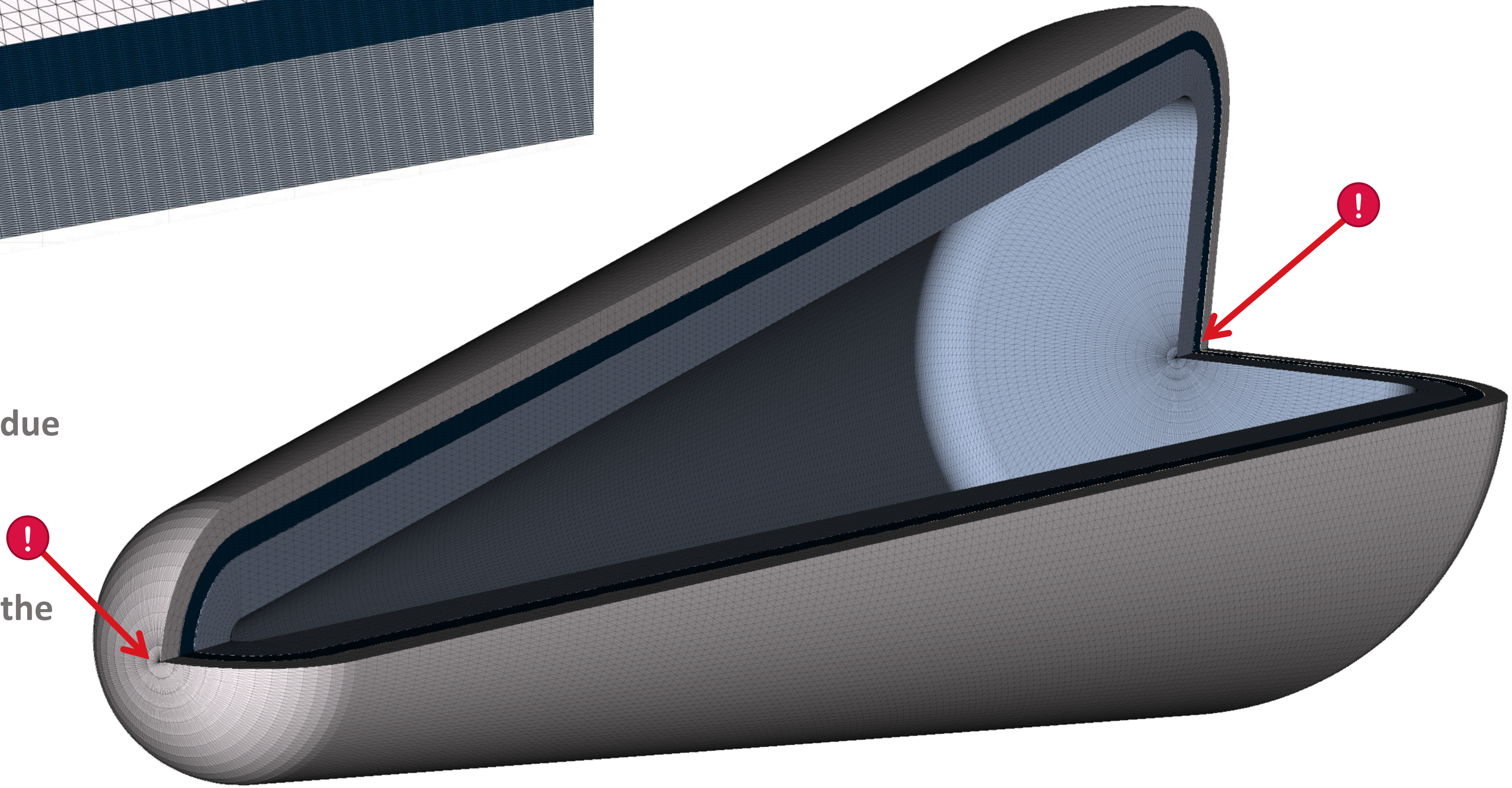


Example of the extrusion on our model test case



Around the poles, the mesh quality in strongly degraded due to the convergence of all the meridians

- This impact the quality of the solution as well

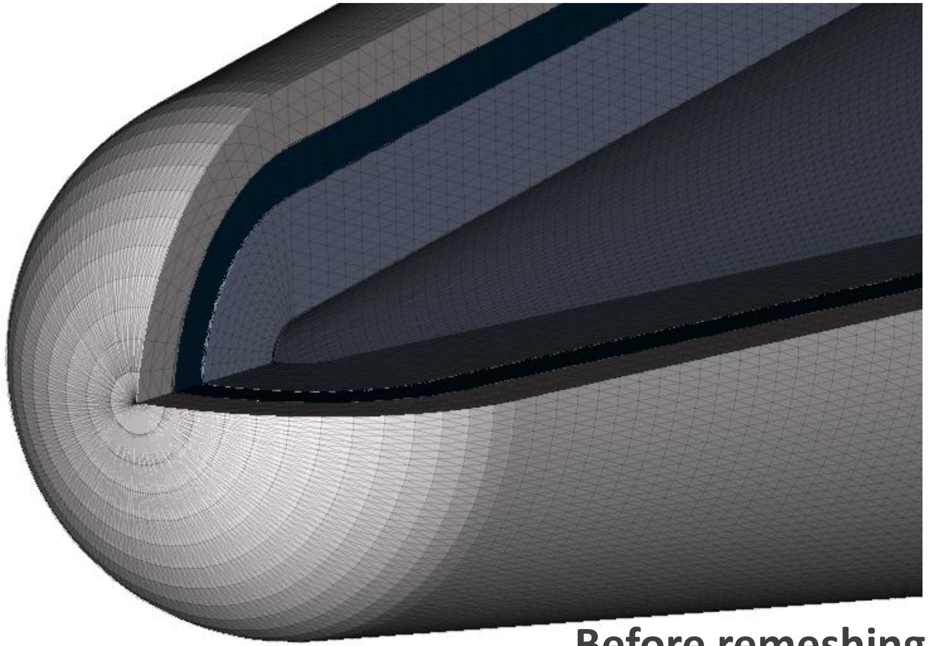


To improve the mesh quality around the poles, we use the automatic remeshing tool MMG

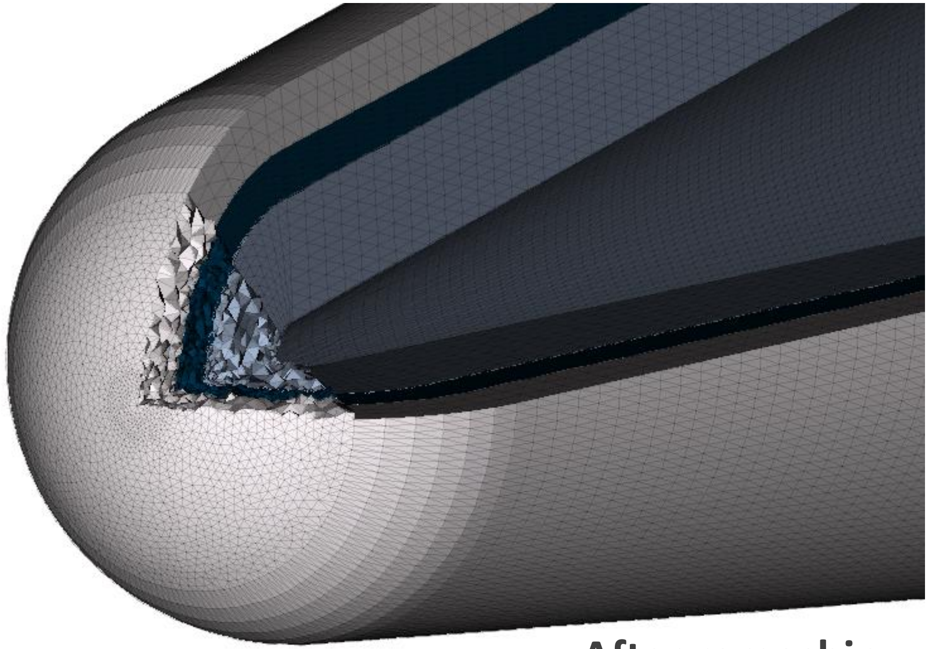
- Open source (https://www.mmgtools.org)
- C API
- Out of the box (we use no specific parameter)

We only specify:

- A mask to determine which tetrahedrons need to be remeshed
- A map of cell size based on the reference of the tetrahedrons



Before remeshing



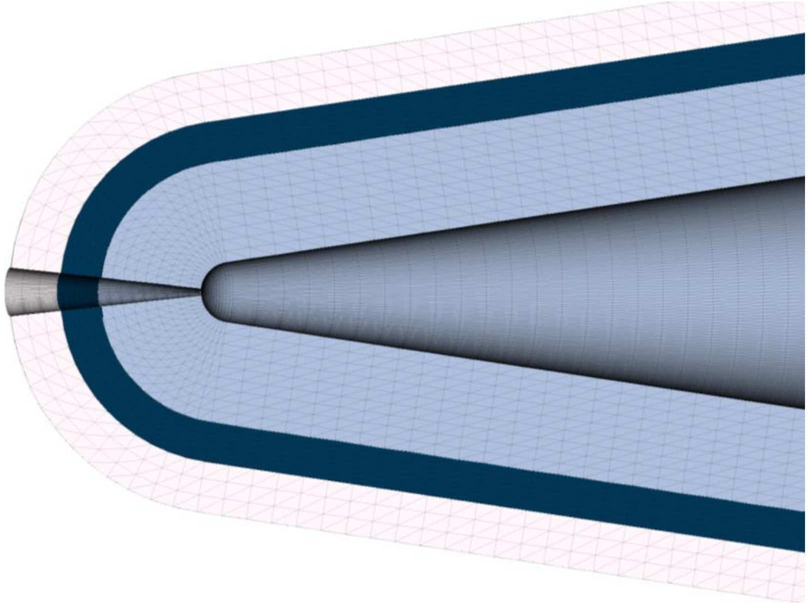
After remeshing

To remedy the poor mesh quality around the poles, we use the automatic remeshing tool MMG

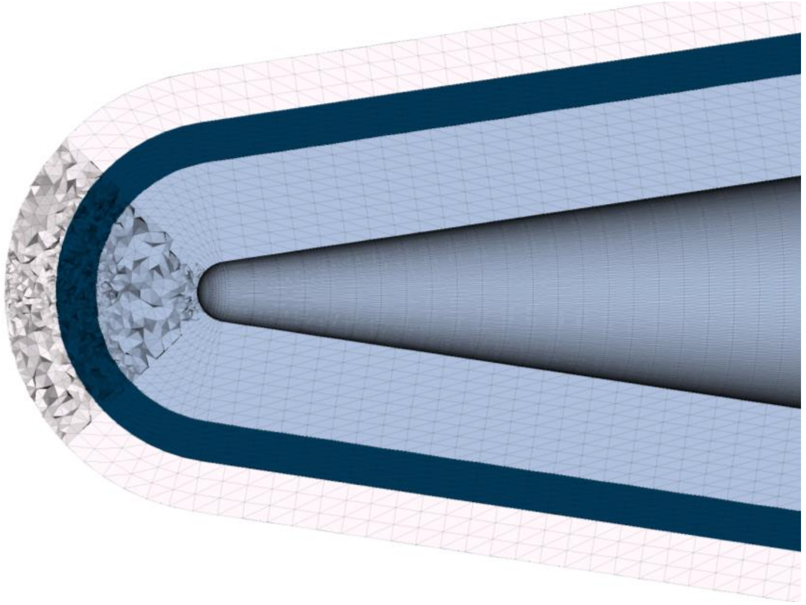
- Open source (https://www.mmgtools.org)
- C API
- Out of the box (we use no specific parameter)

We only specify:

- A mask to determine which tetrahedrons need to be remeshed
- A map of cell size based on the reference of the tetrahedrons



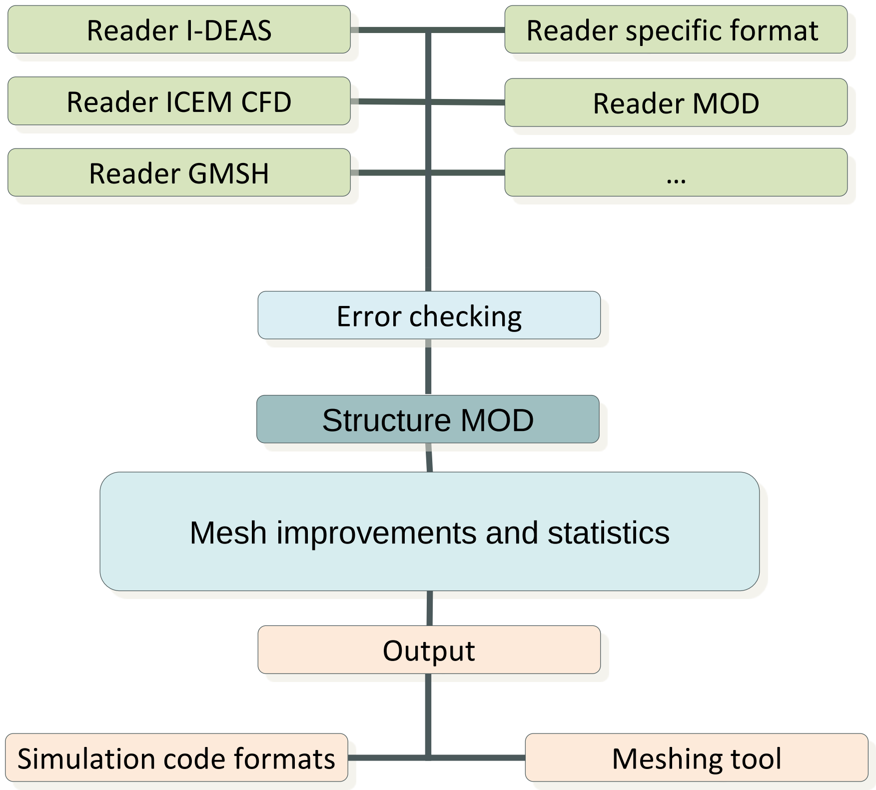
Before remeshing

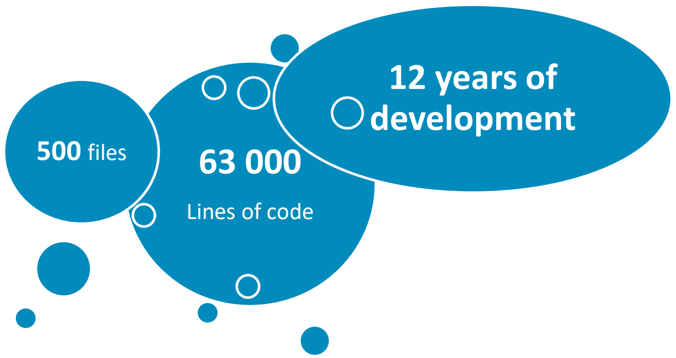


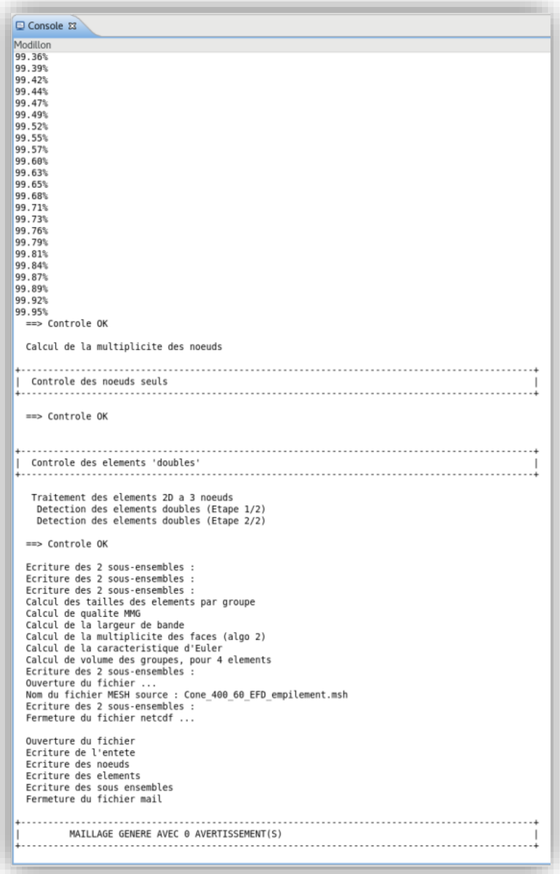
After remeshing

Mesh verification

- **Implementation of a tool to verify and convert meshes (MODILLON)**
  - To guaranty the coherance between the meshing tools and the simulation codes
  - To ease the repeated tasks of engineers on meshes
  - Error checking and modification adapted to the requirements of the EM simulation codes
  - Software architecture : C++, C,







Example of error checking and mesh improvements (some specific to EM codes)

Domain :

- Unit of the mesh regarding to the unit of the code and tolerance management

Nodes and elements :

- Numbering of nodes and elements (duplication of labels, holes, start from 1)
- Unwanted element removal
- Elements orientation and plane z=0 orientation
- Coincident nodes/elements removal
- Bandwidth node numbering optimization
- Creation of the boundary node list
- Free edge
- Euler constant
- Bad shape elements (MMG Quality or distortion)
- Edge length histogram

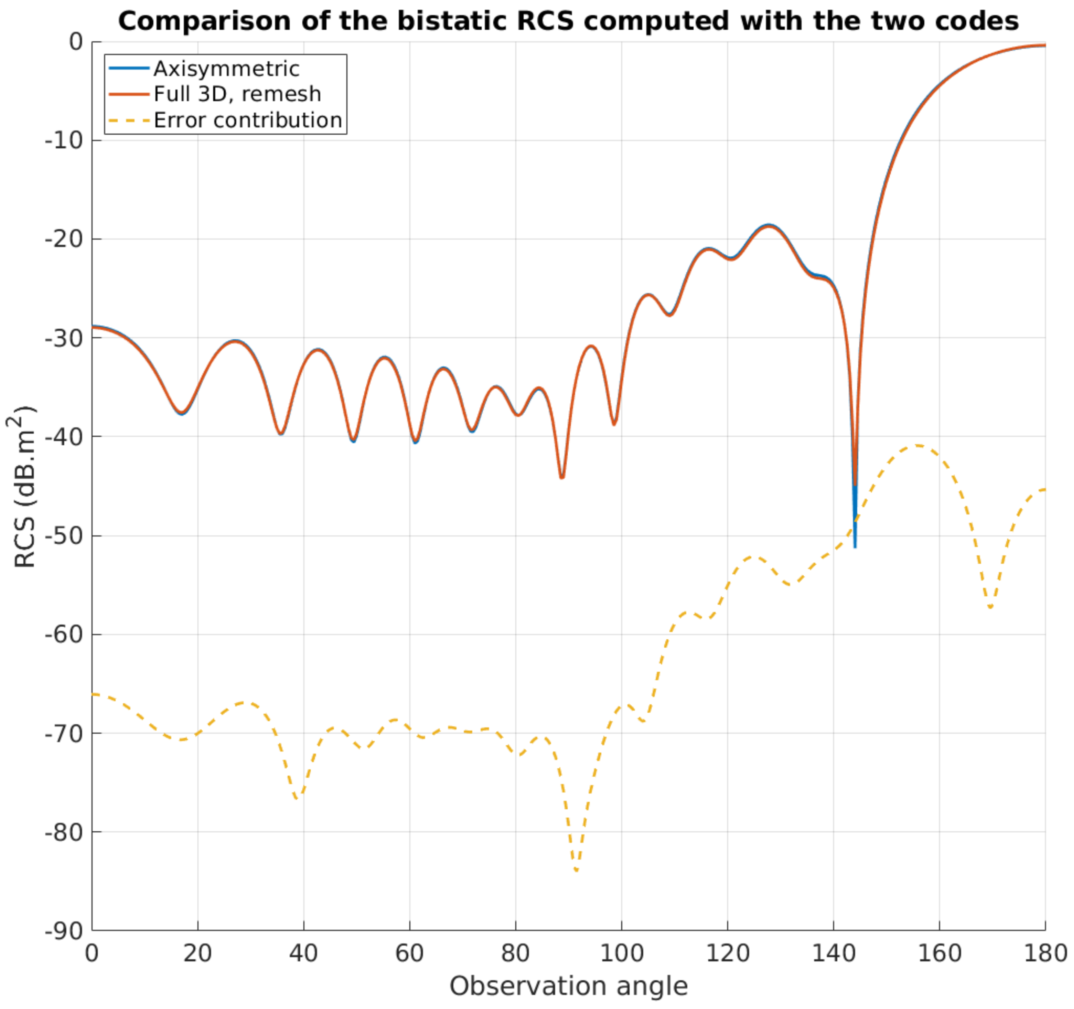
Metrics and error checking
Mesh improvement by MODILLON

Elements Groups :

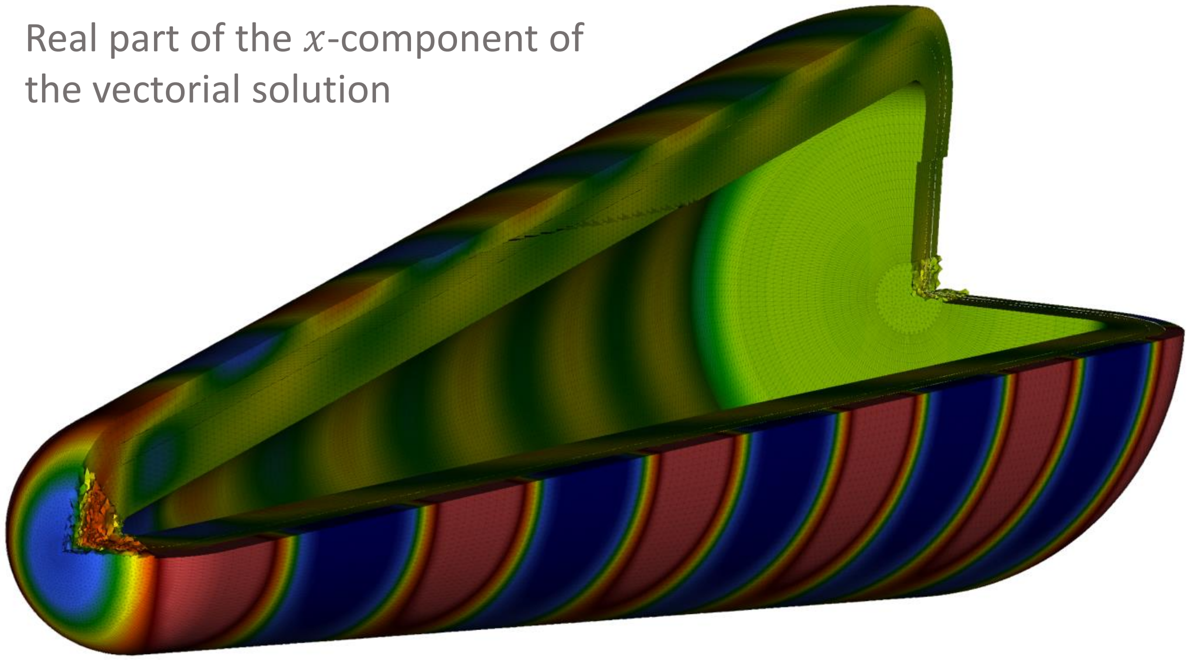
- Material numbering check
- Lengths, surfaces and volumes determination by groups

Simulation results on the cone-sphere use case

We compare the solution computed by the axisymmetric code (reference) and the 3D finite element code



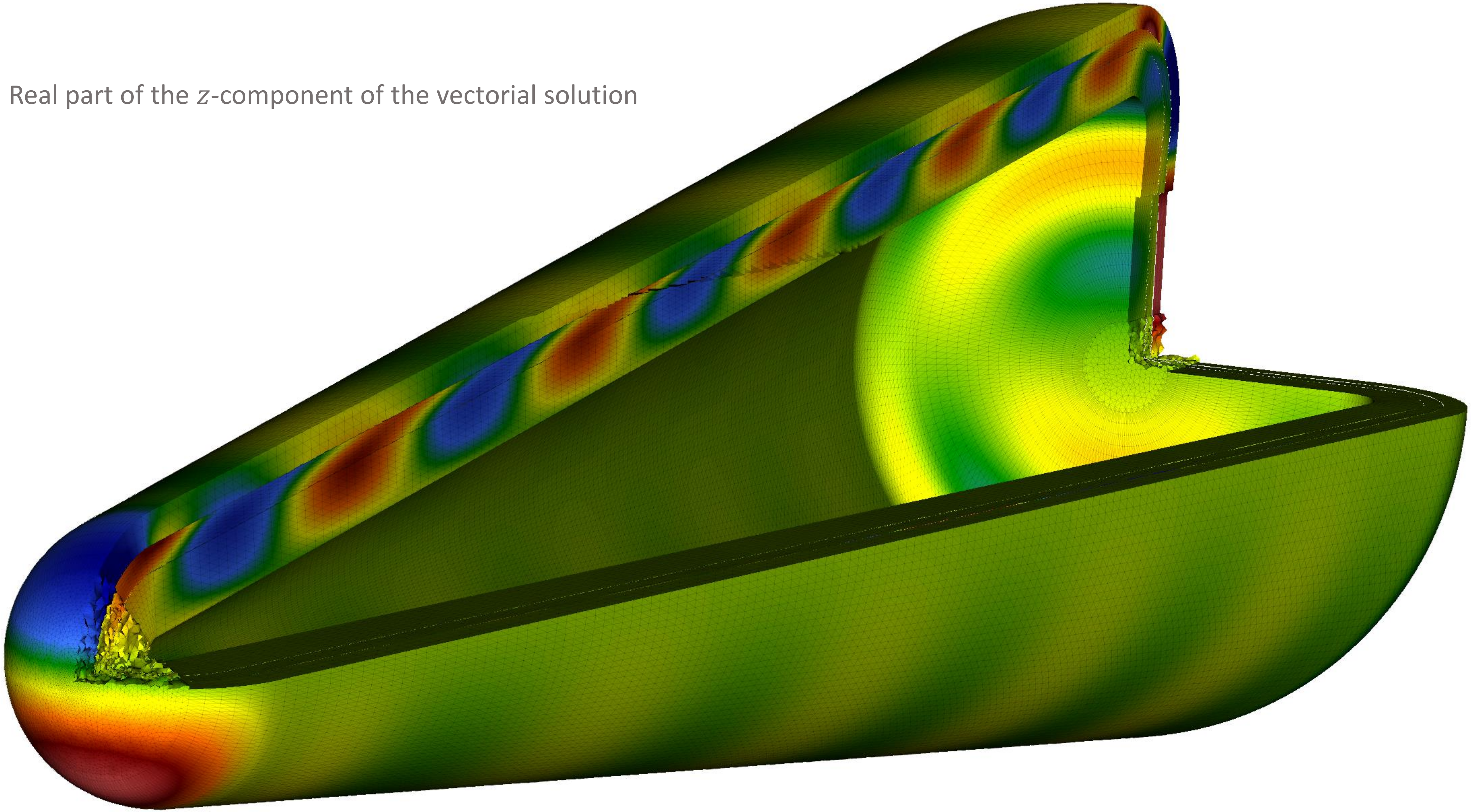
Real part of the x-component of the vectorial solution



The RCS is very well computed by the 3D code and the accuracy (directly linked to the quality of the mesh) is very good

Simulation results on the cone-sphere use case

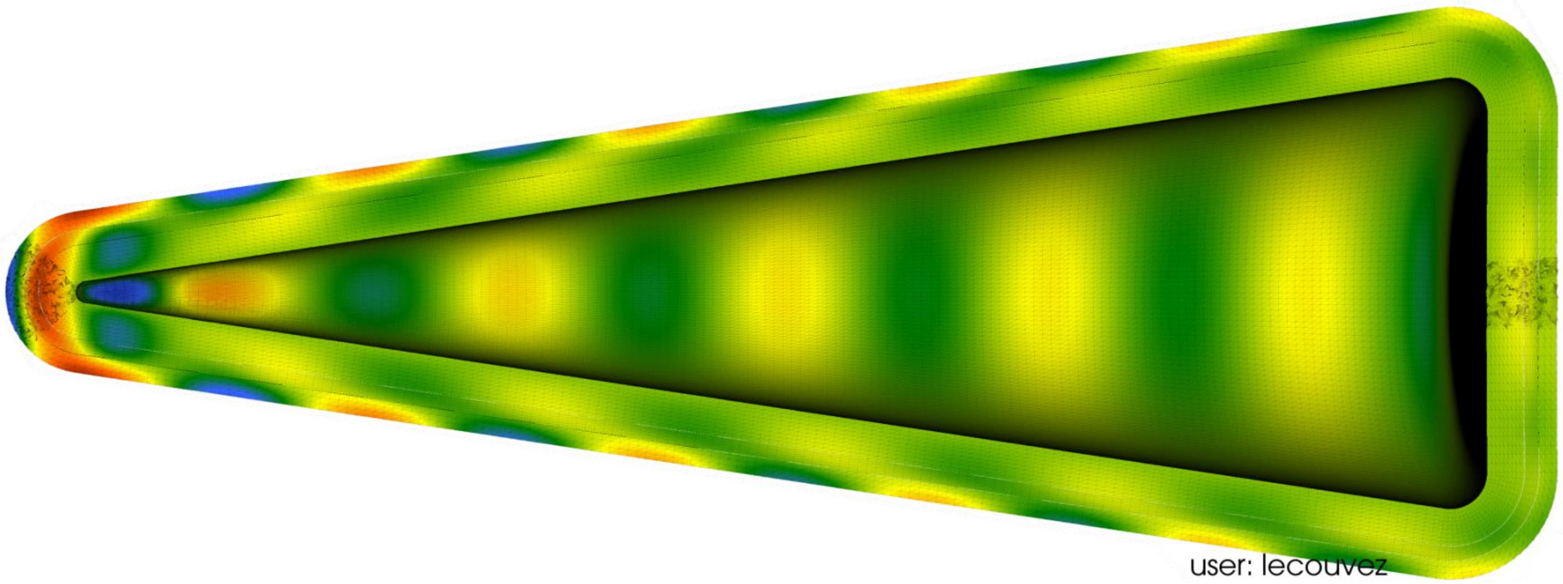
Real part of the z-component of the vectorial solution



Conclusion

We have presented a toolchain developed at CEA to easily generate 3D meshes for axisymmetric geometries

- The meshes generated verify the cell size criterion required by the physics
- We use anisotropic meshes to reduce the size of the mesh without degrading the accuracy of the solution
- Better mesh quality is obtained using the MMG automatic remeshing capabilities
- Assisted 1-D shape modeling and automated meshing tools help to reduce the cycle time in conception of 3D stealth objects



Thank you for your attention.