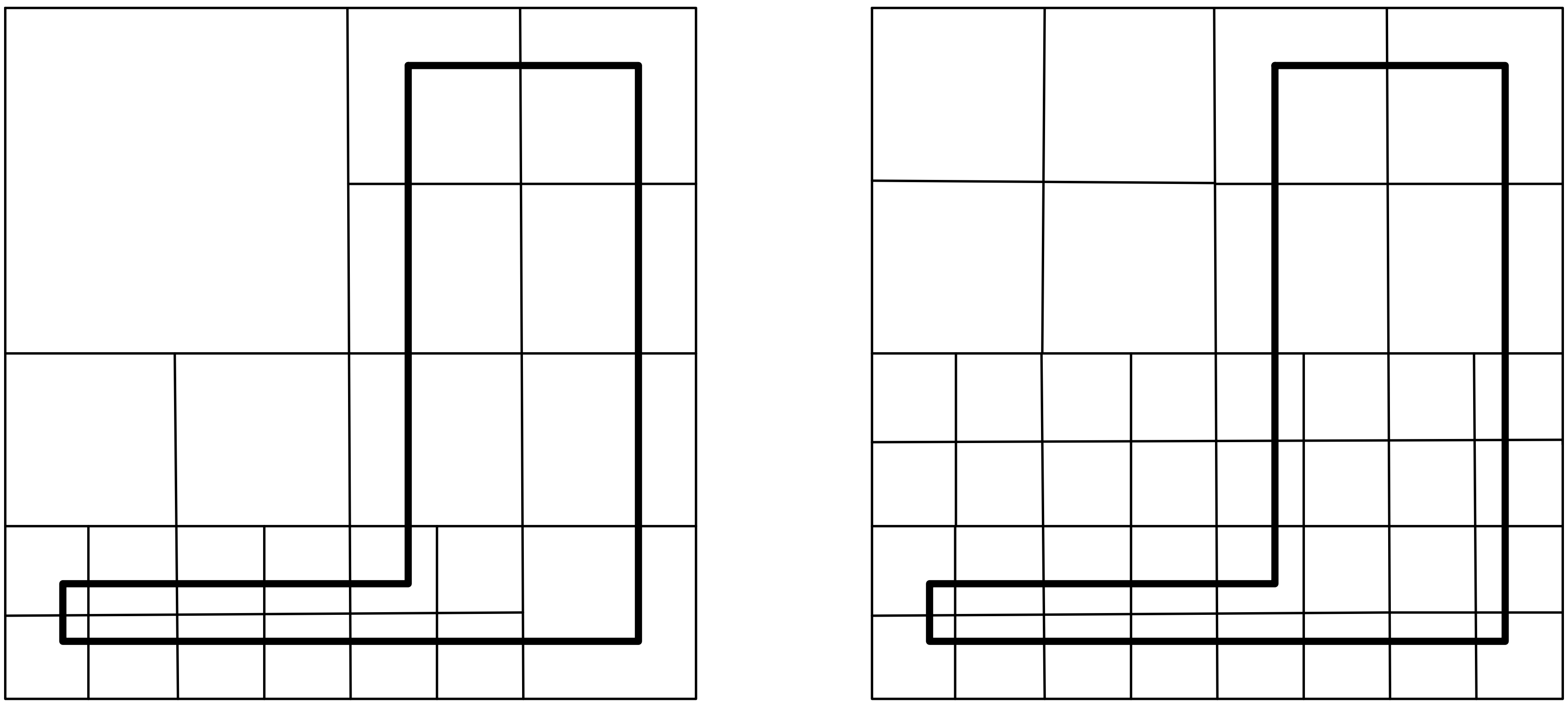
HEXOTIC:
AN AUTOMATED
HEXAHEDRAL MESH GENERATOR

Loïc Maréchal

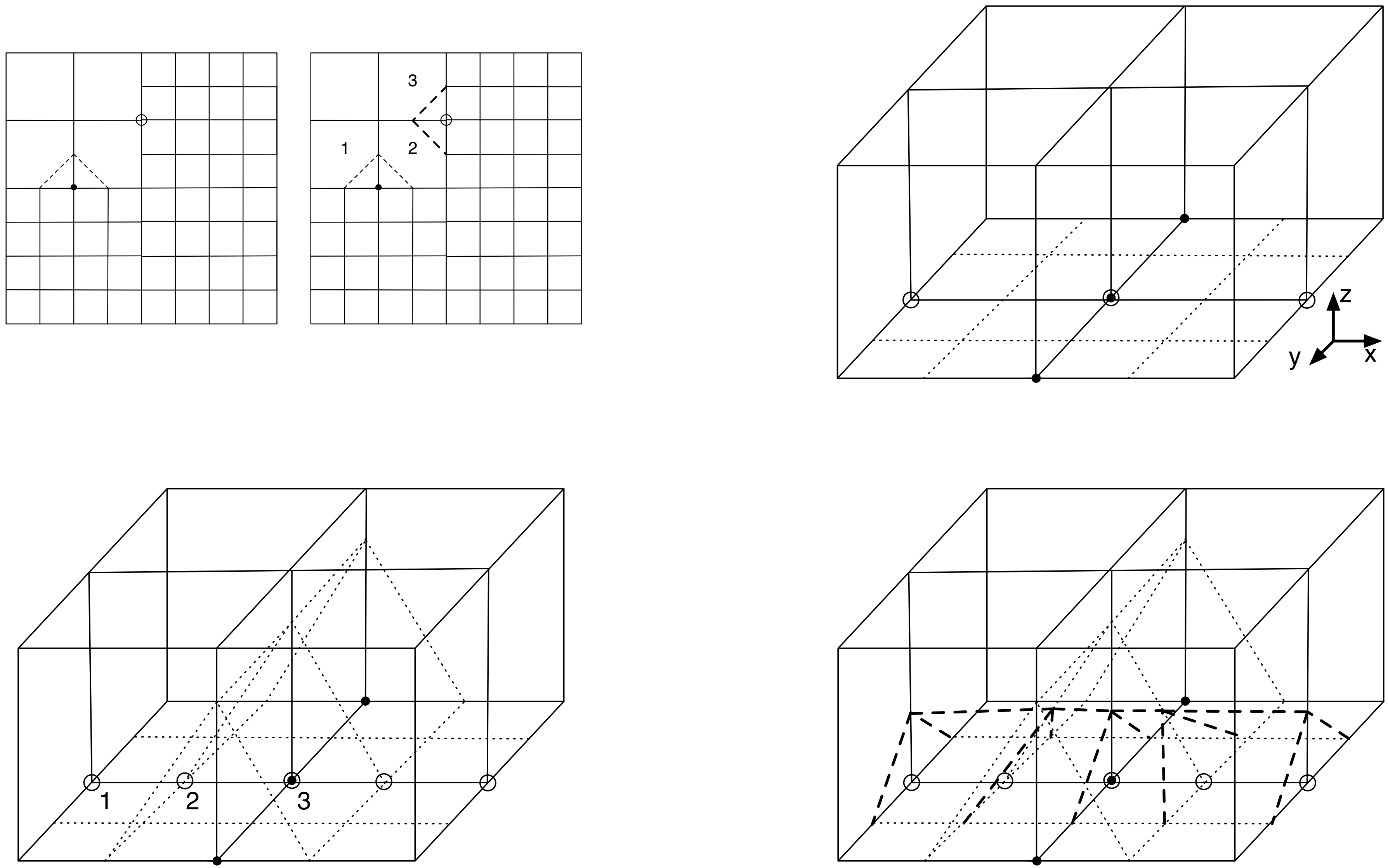
INRIA, nov 2013

1 / Basics of octree meshing:

-Set of subdivision criteria : geometry's thickness, curvature, a posteriori error estimate.
-Balancing rule : a cell should not be more than two time bigger or smaller than it neighbors.
-Pairing rule : if a cell is to be subdivided, then it "brothers" should be subdivided too.



1b / Octree conforming using polyhedral cutting:



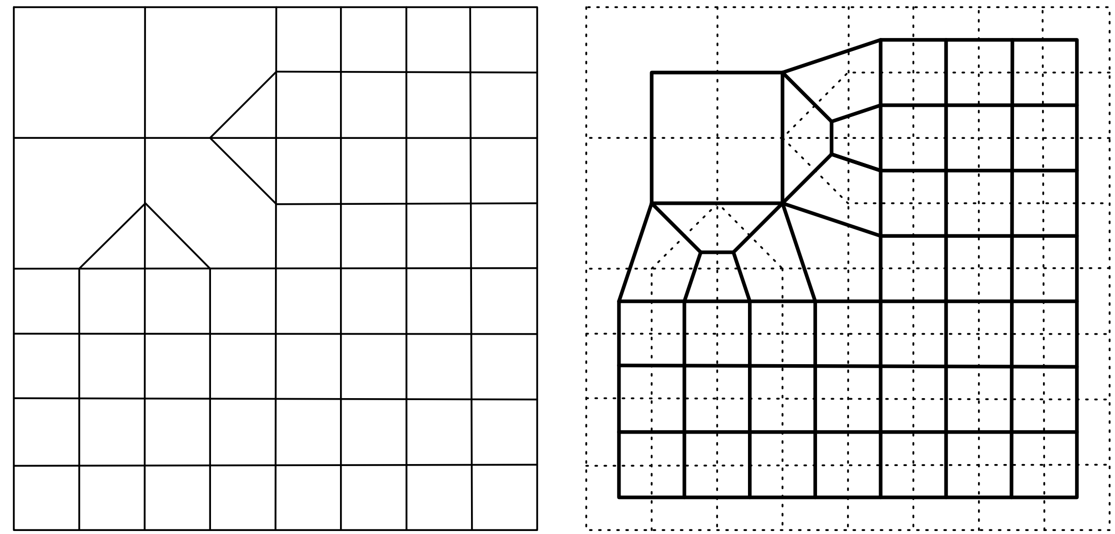
2 / Dual mesh:

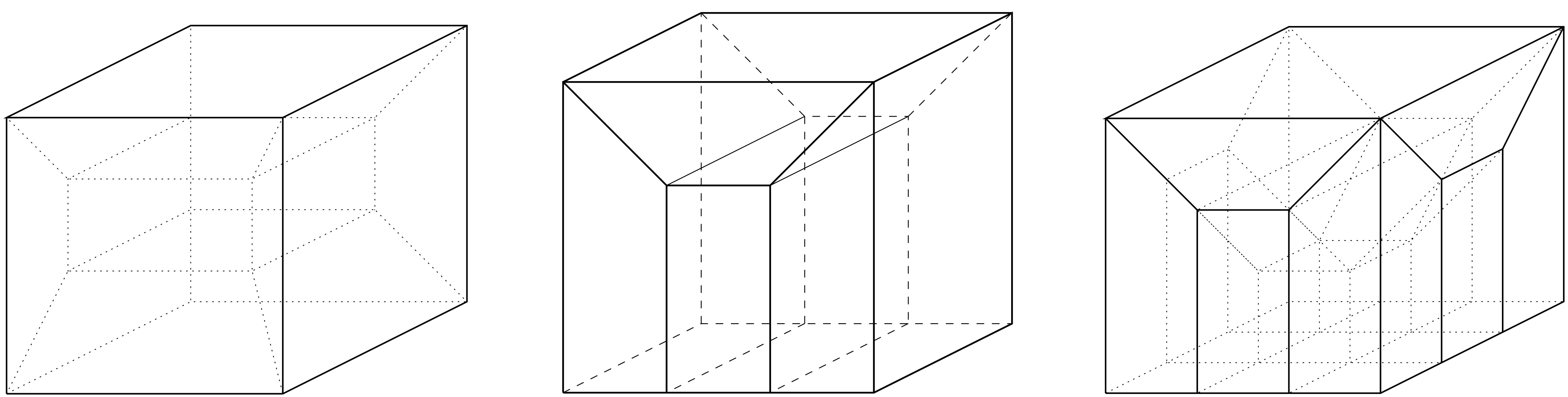
2D :

- element <-> node
- edge <-> edge

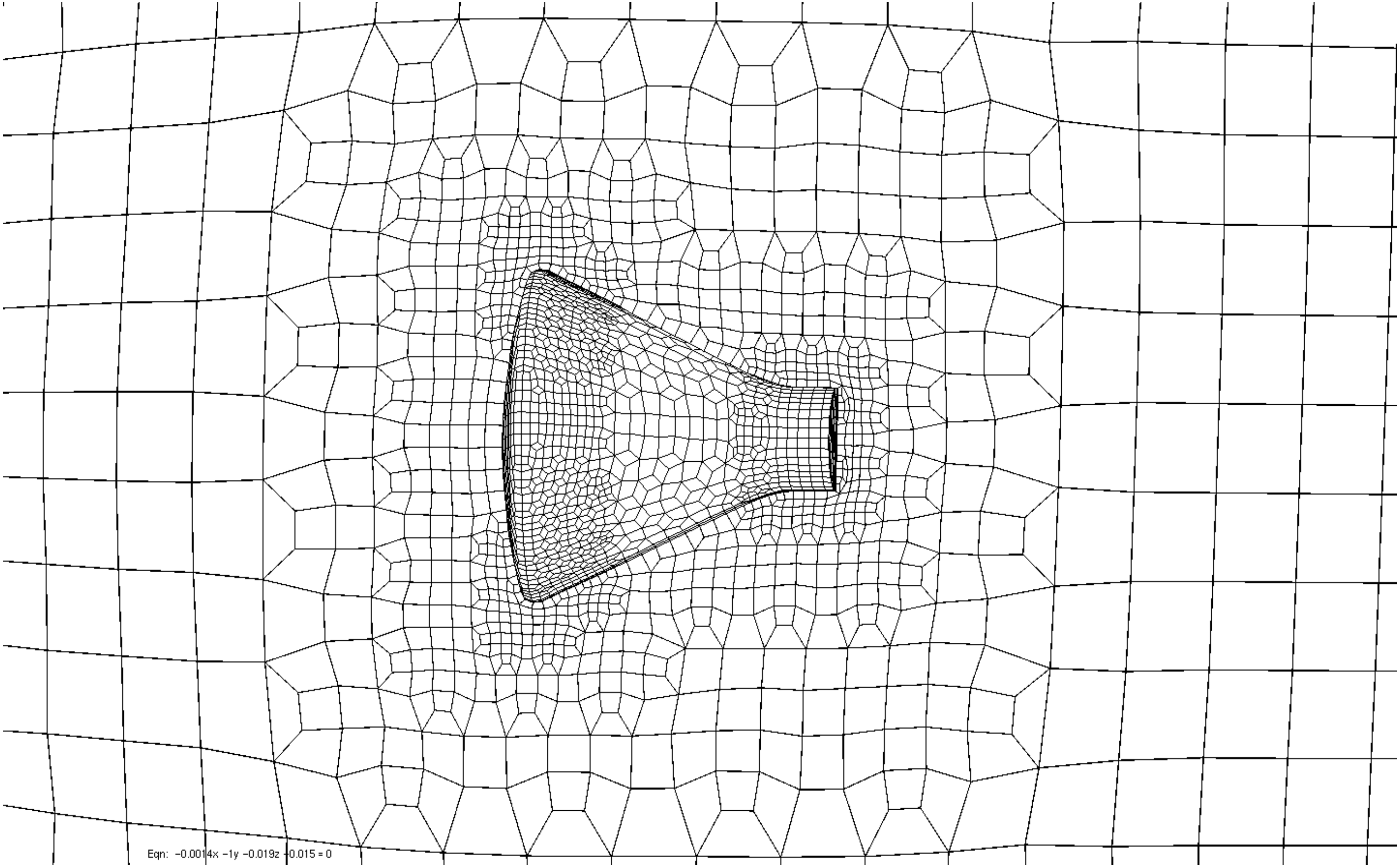
3D :

- element <-> node
- face <-> edge





2b / Dual mesh:

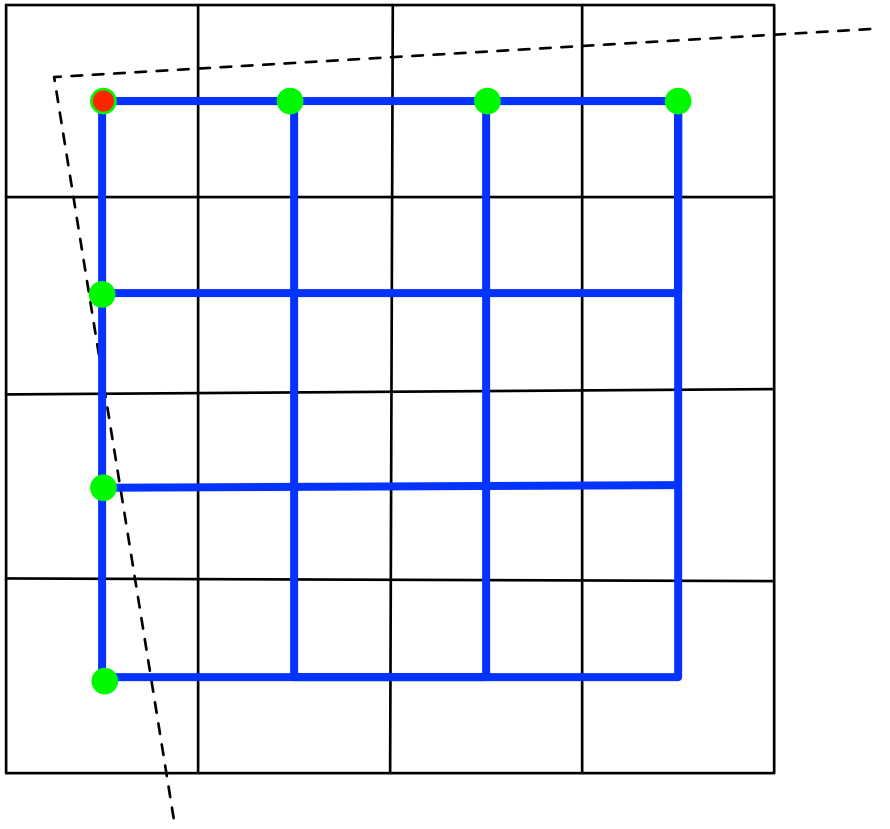


3 / Surface meshing:

Analysis of edges (triangles) intersecting each octant :

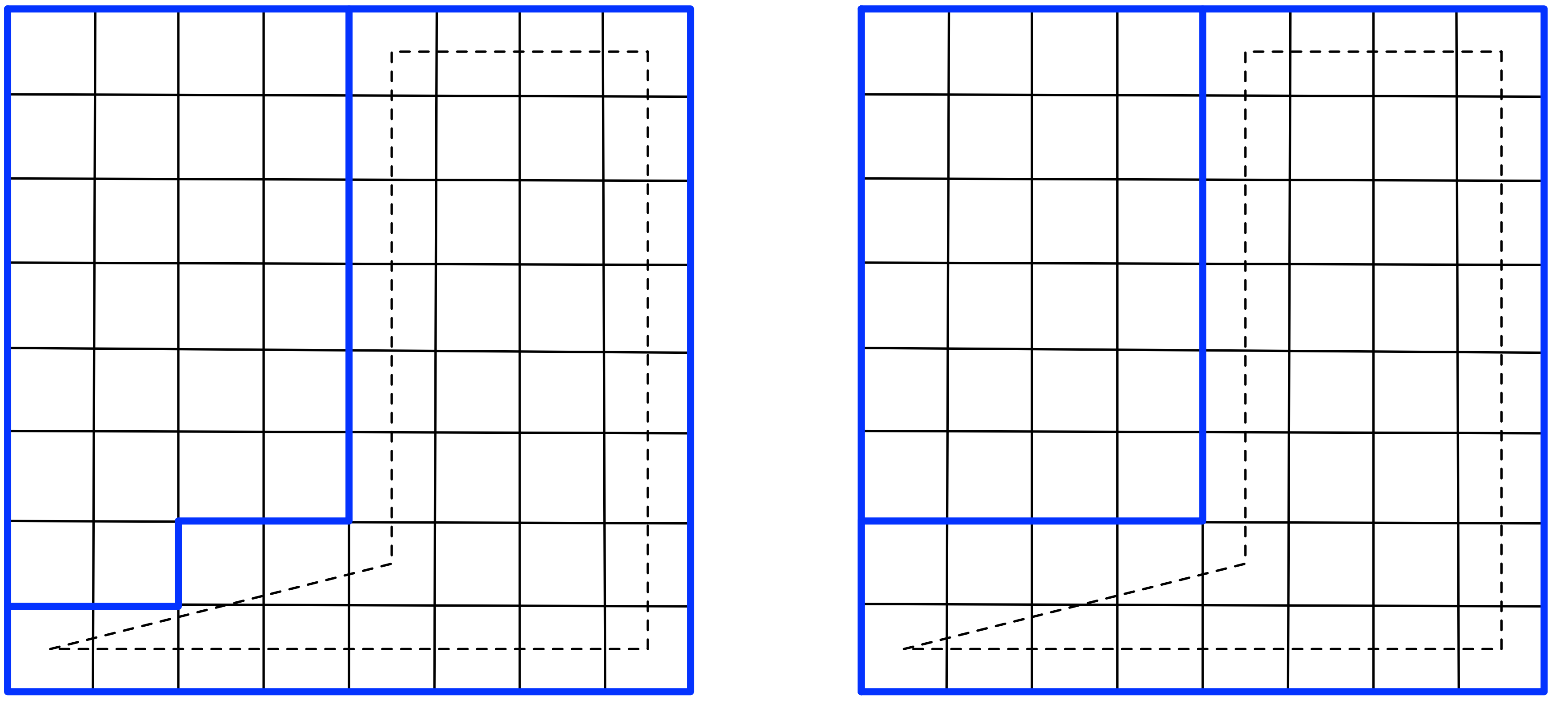
2D :
1 group -> median line.
2 groups -> two intersecting lines making a sharp angle.
3+ too complex.

3D :
1 group -> median plane.
2 groups -> two intersecting planes making a sharp edge.
3 groups -> three intersecting planes making a sharp corner.
4+ too complex.



4 / Subdomain recovering:

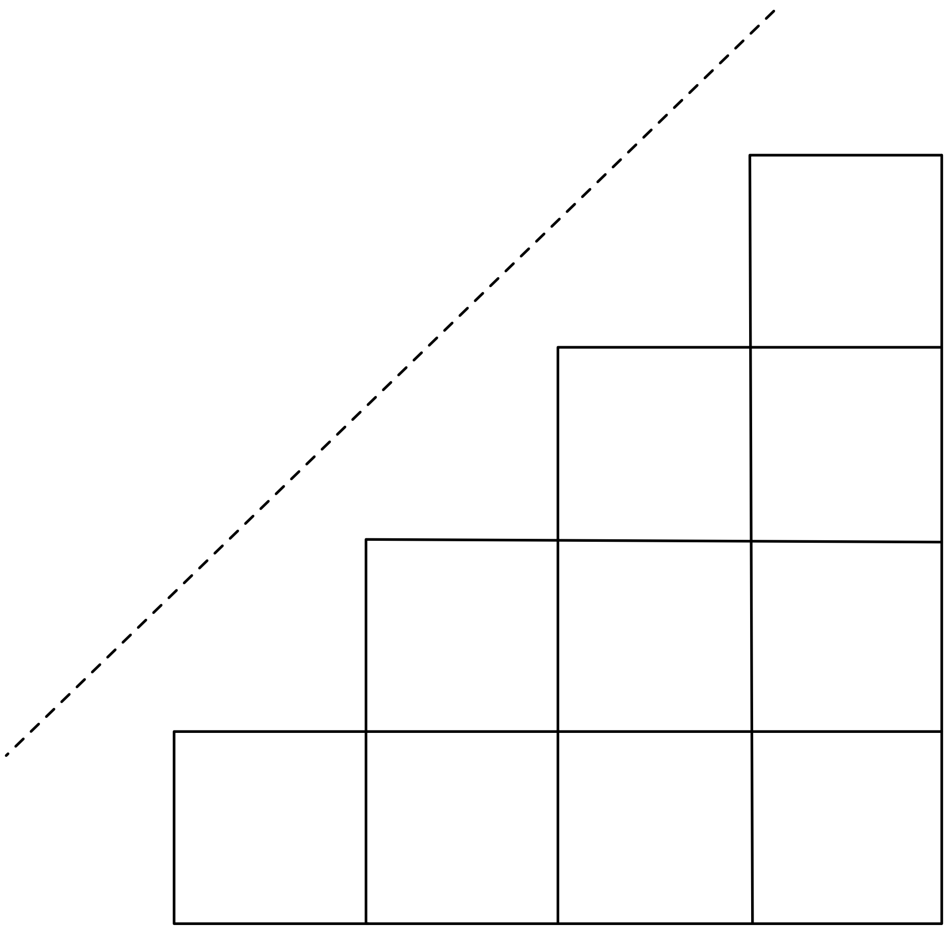
- -Basic inside/outside algorithm.
- -It can handle internal surfaces (non-manifold geometries).
- -Domains must be two-element thick in any direction.



5 / Surface projection:

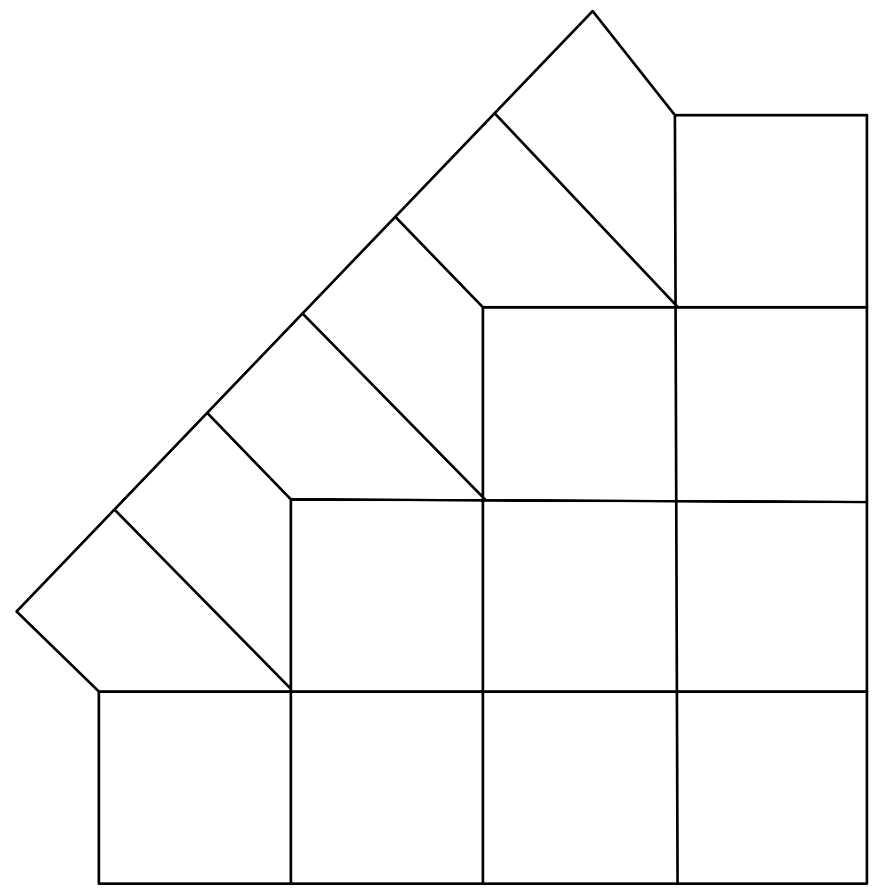
Problem : octree generates a so called "staircase mesh".

-Hexes may have two or three faces to be projected on the same plane -> degenerated.
-Surface quad may have two edges to be projected on the same sharp line.

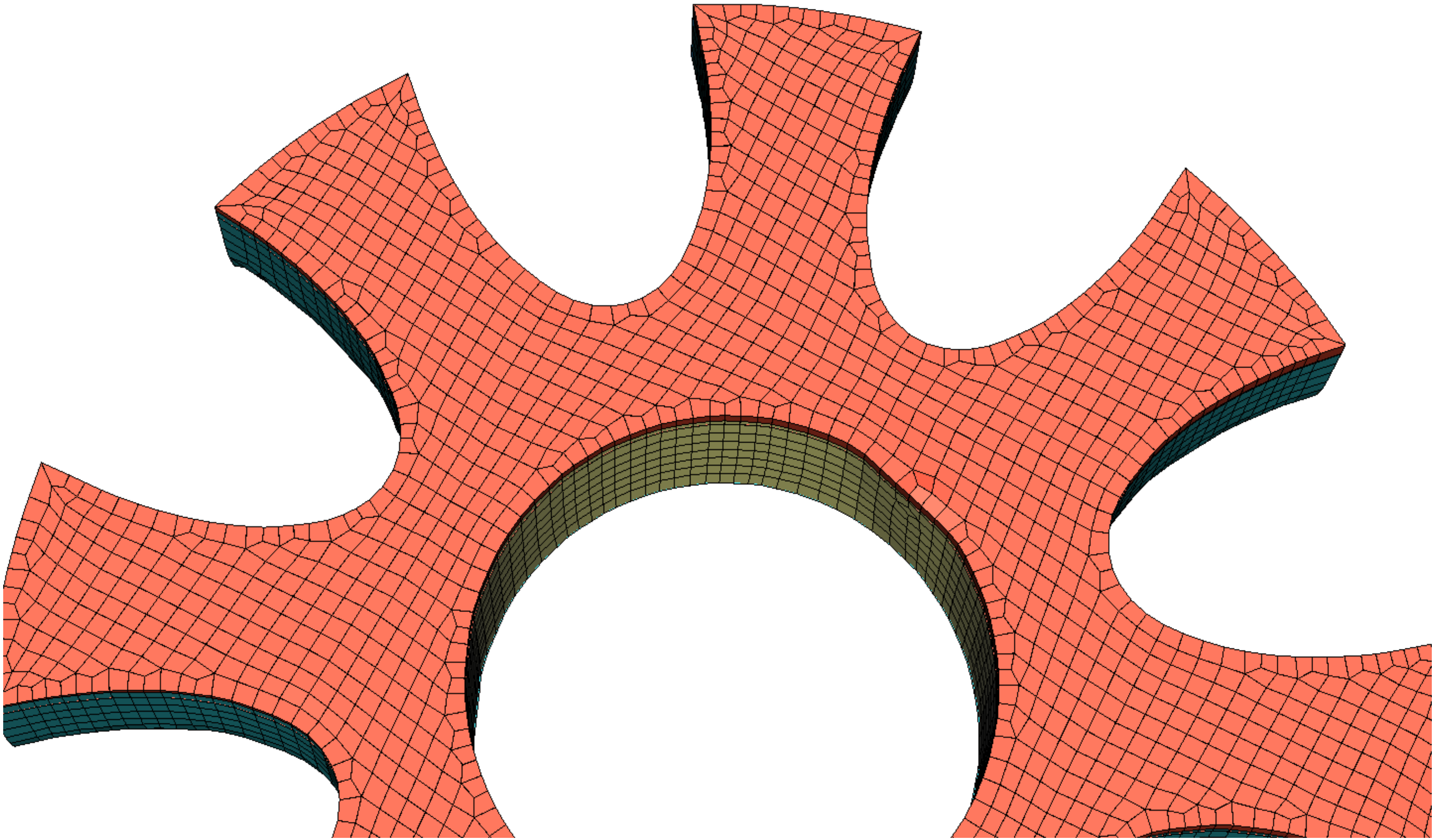


Solution : buffer-layer insertion

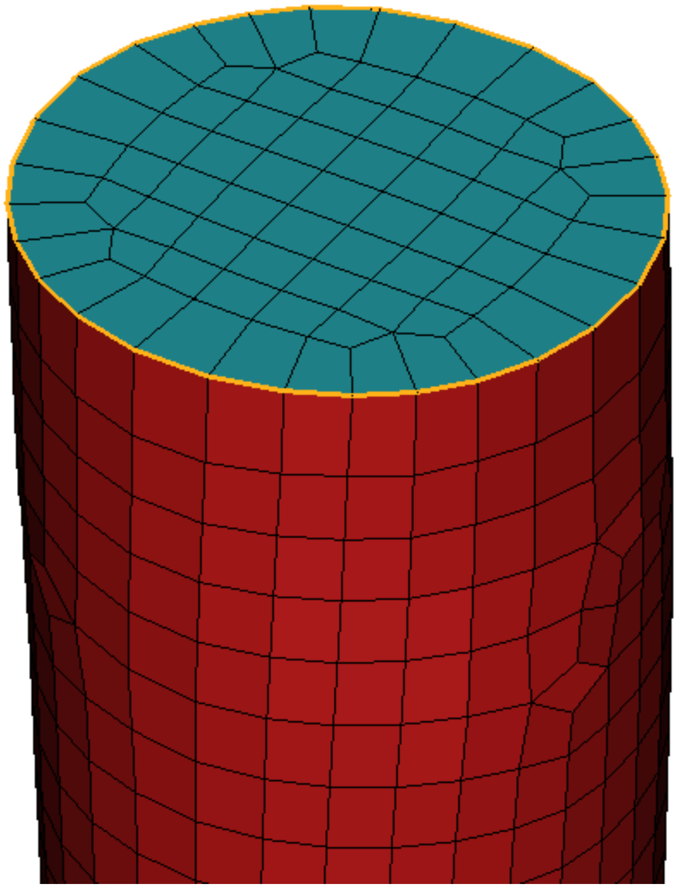
-A first layer of hexes is inserted around the staircase mesh so that new boundary elements have only one face to be projected on the real geometry.
-Likewise, a second layer of element is inserted around sharp edges.

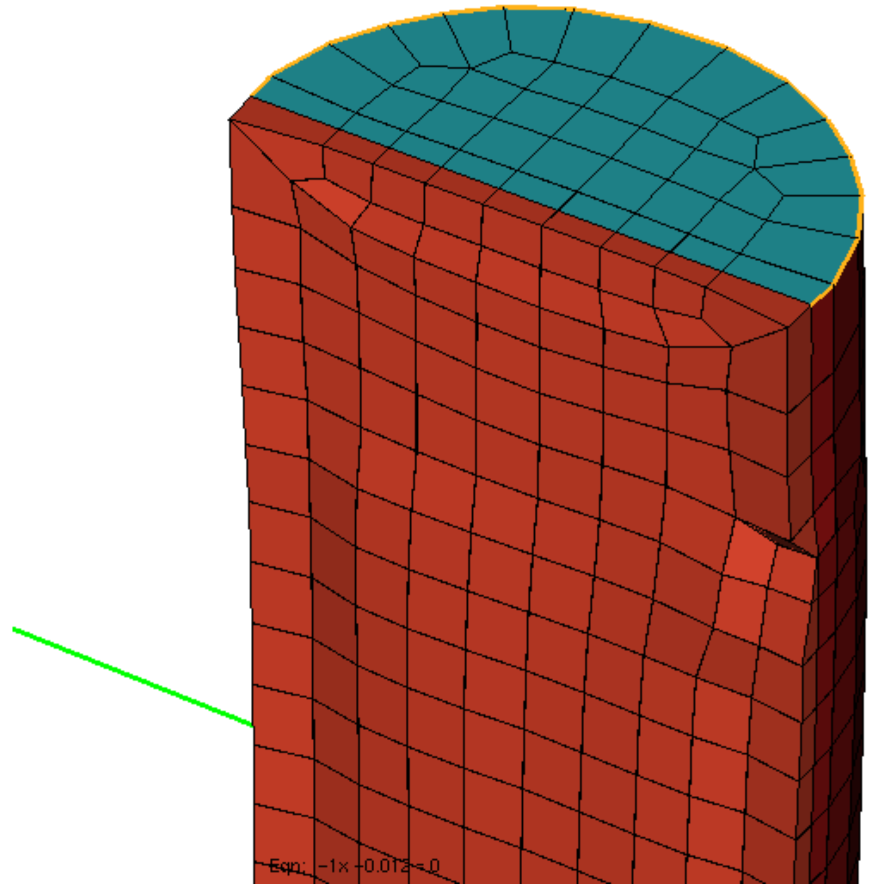


5b / Boundary after projection:



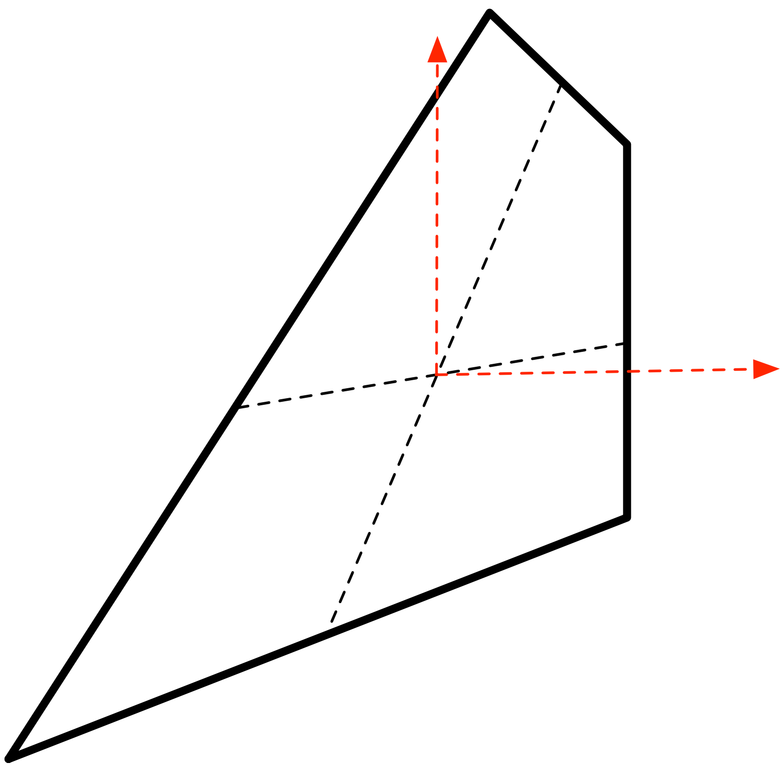
5c / Second layer:

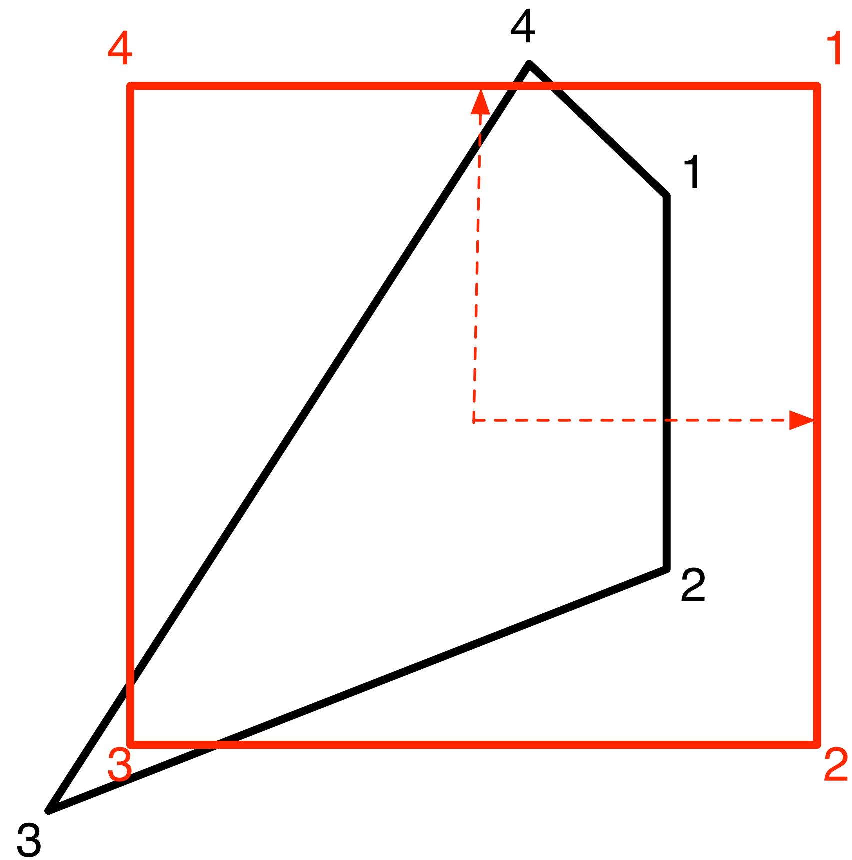




6 / Quality optimization via node smoothing:

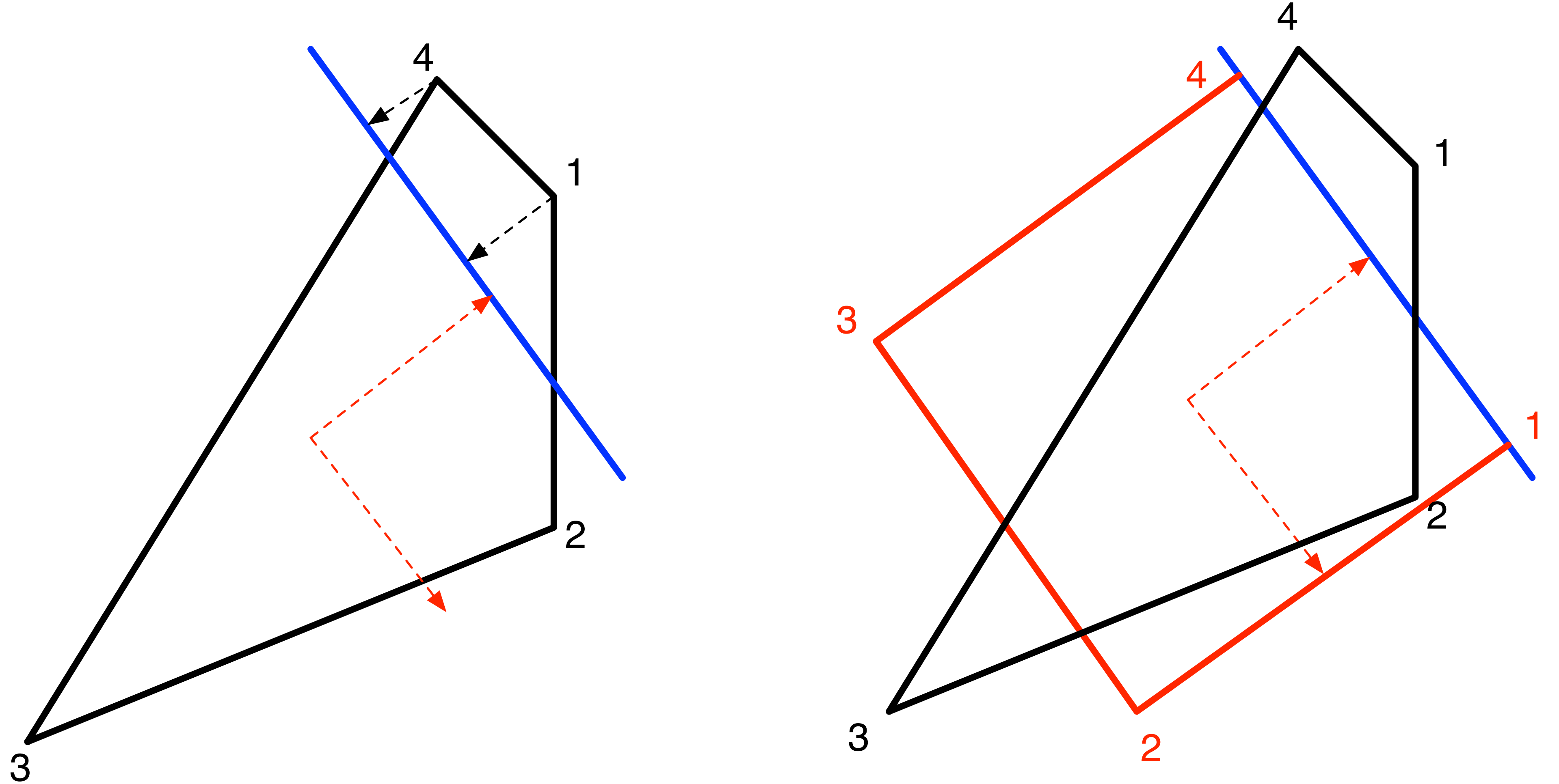
- Each solver has its own quality criterion.
- Only one common ground : each hex must have a positive volume.
- So we can only try to get as close as possible to the perfect cube.
- The optimizing process finds the closest perfect cube from each hex and adds the contributions to a new set of nodes' coordinates which eventually will result in a better mesh.
- This step has been multi-threaded with the help of the LPlib3 and achieves a 7.5 speed-up on an 8-core server.
- A version has also been ported in OpenCL with the GMlib2 and is 24 times faster than the CPU serial version on a Quadro 6000 GPU.



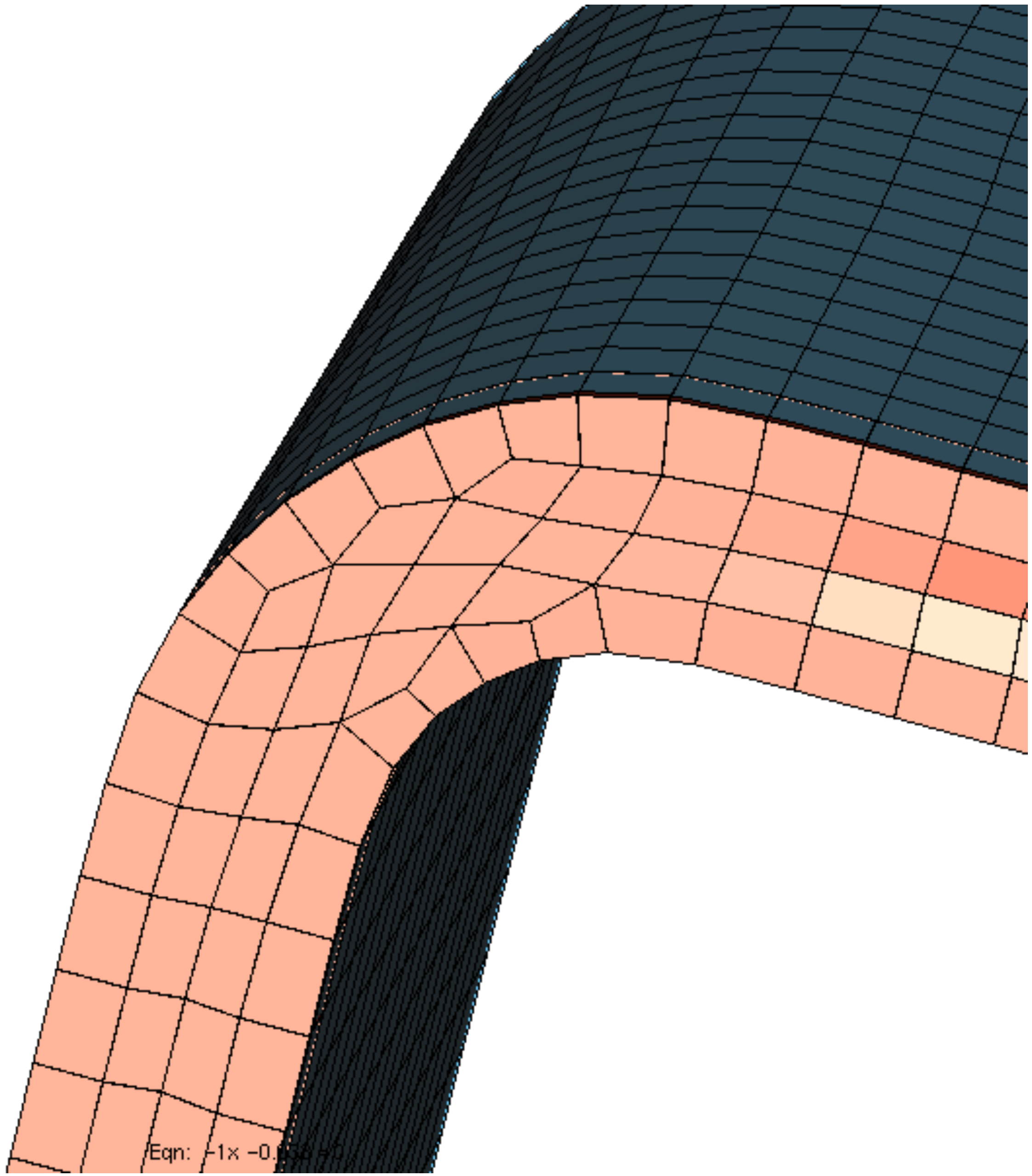


7 / Boundary constrained optimization scheme:

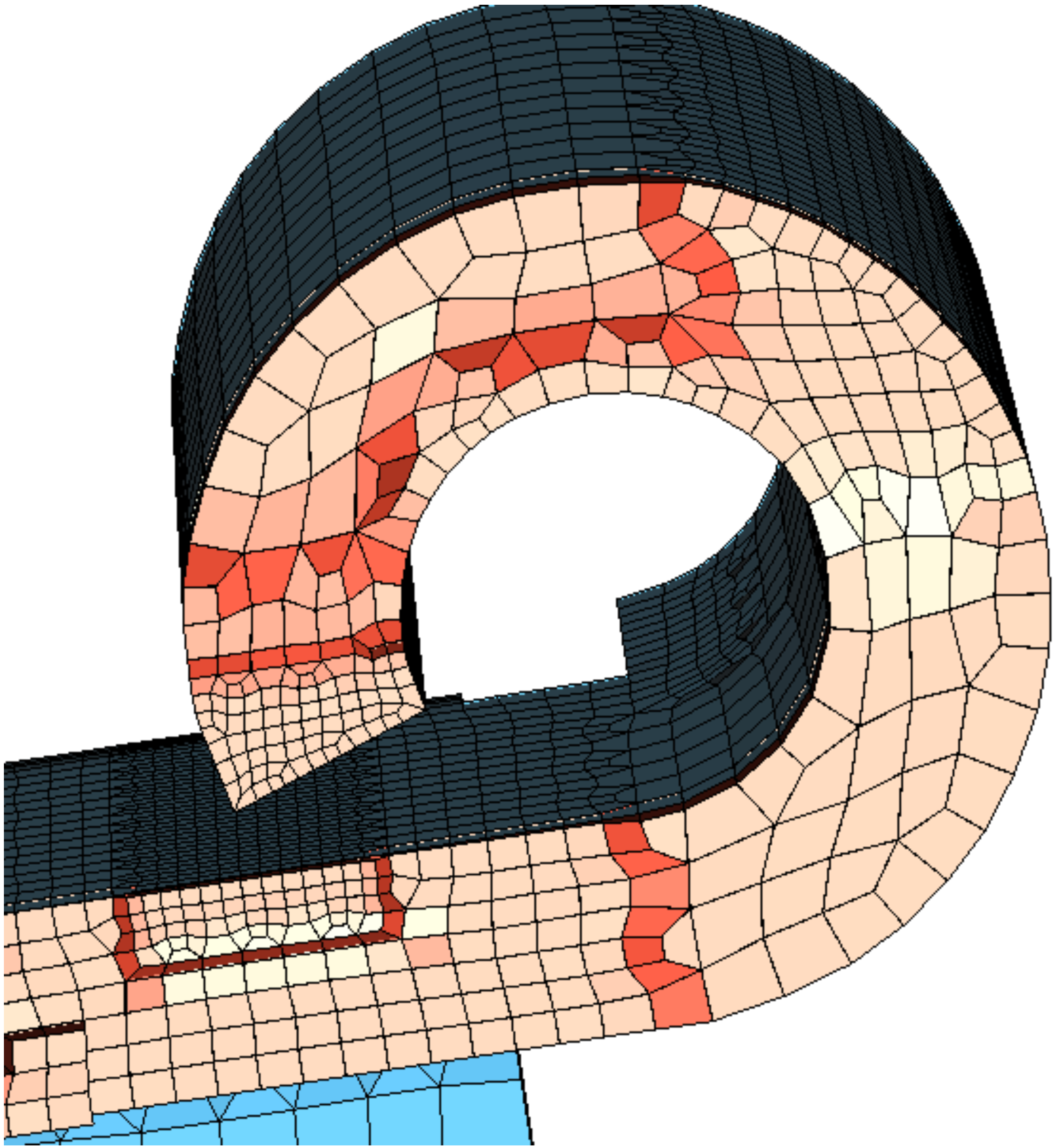
- Boundary elements could be treated the same way as others and surface node could be projected on the geometry afterward -> interlocking : smoothing may move nodes in one direction and projection may move them back !
- Geometry should be part of the smoothing scheme : the perfect cube is rotated and pushed away so that its surface face matches the geometry it should represent.



8 / Strong limitation : he lack of topological operators on hex meshes:

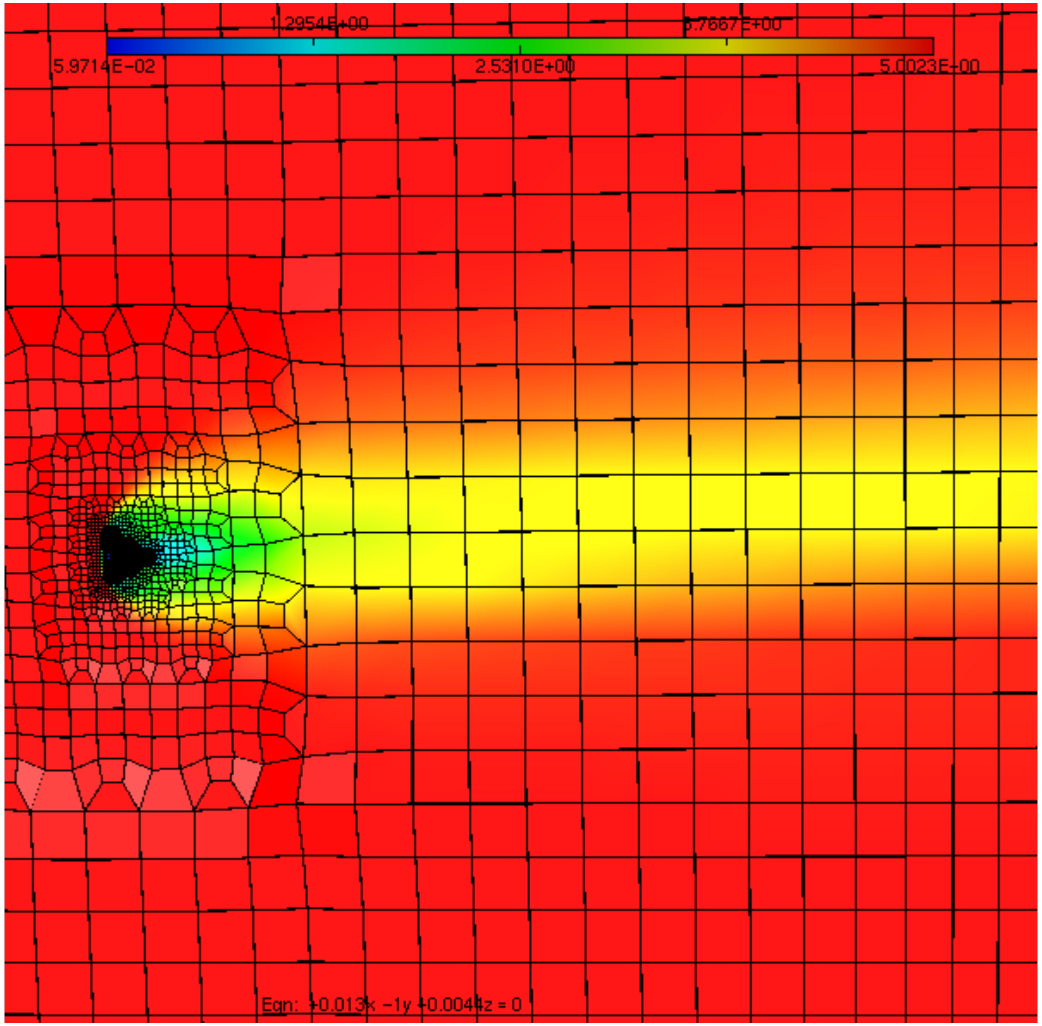


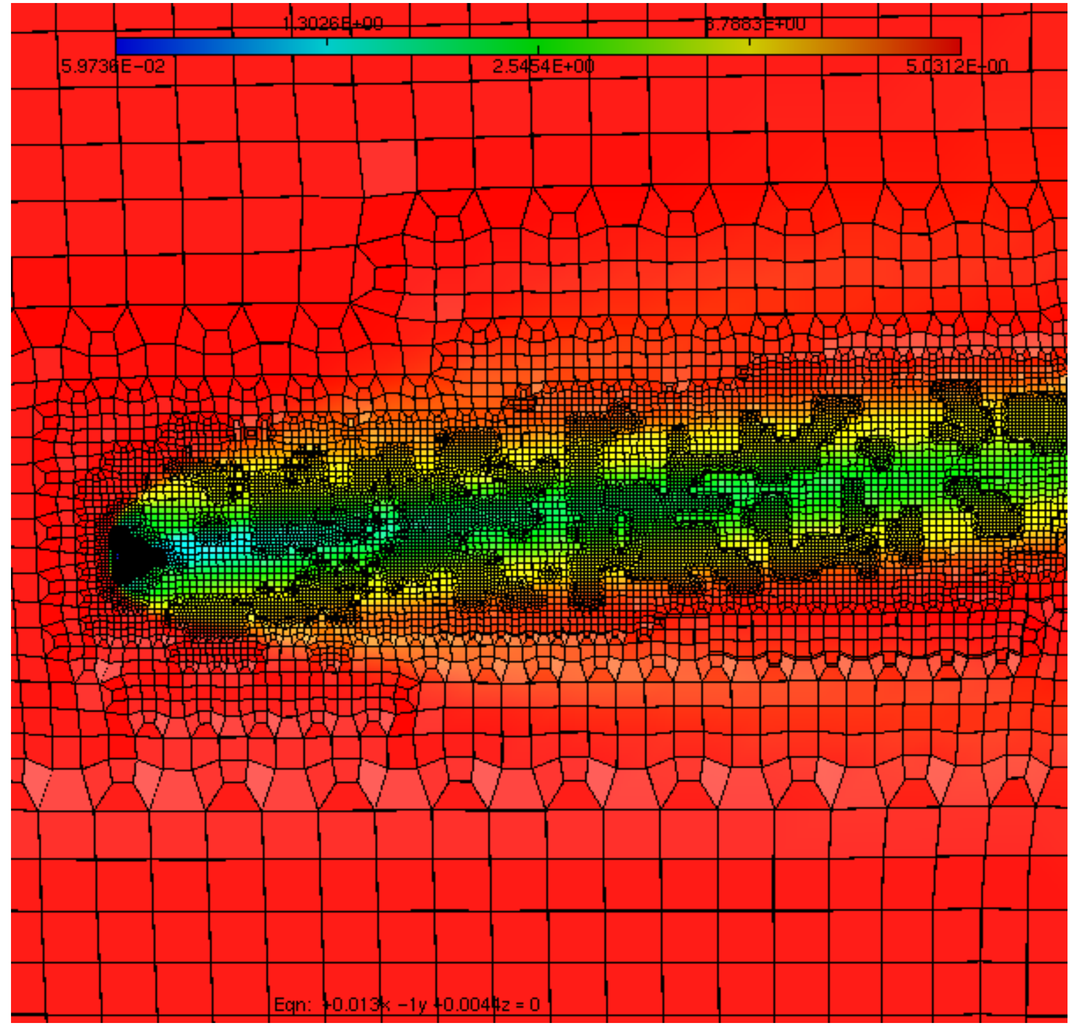
8b / Strong limitation : the lack of topological operators on hex meshes:



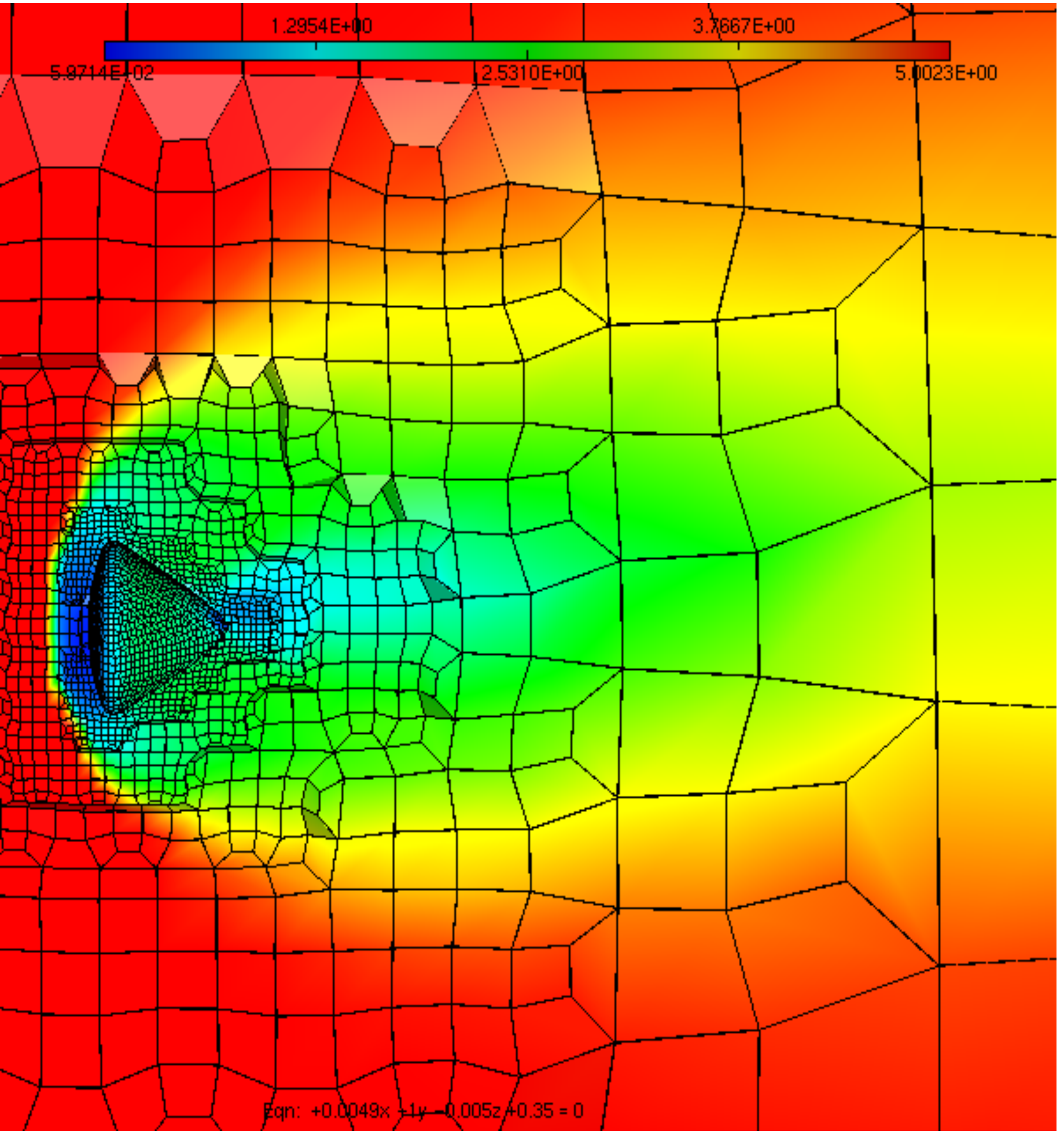
9 / Mesh adaptation:

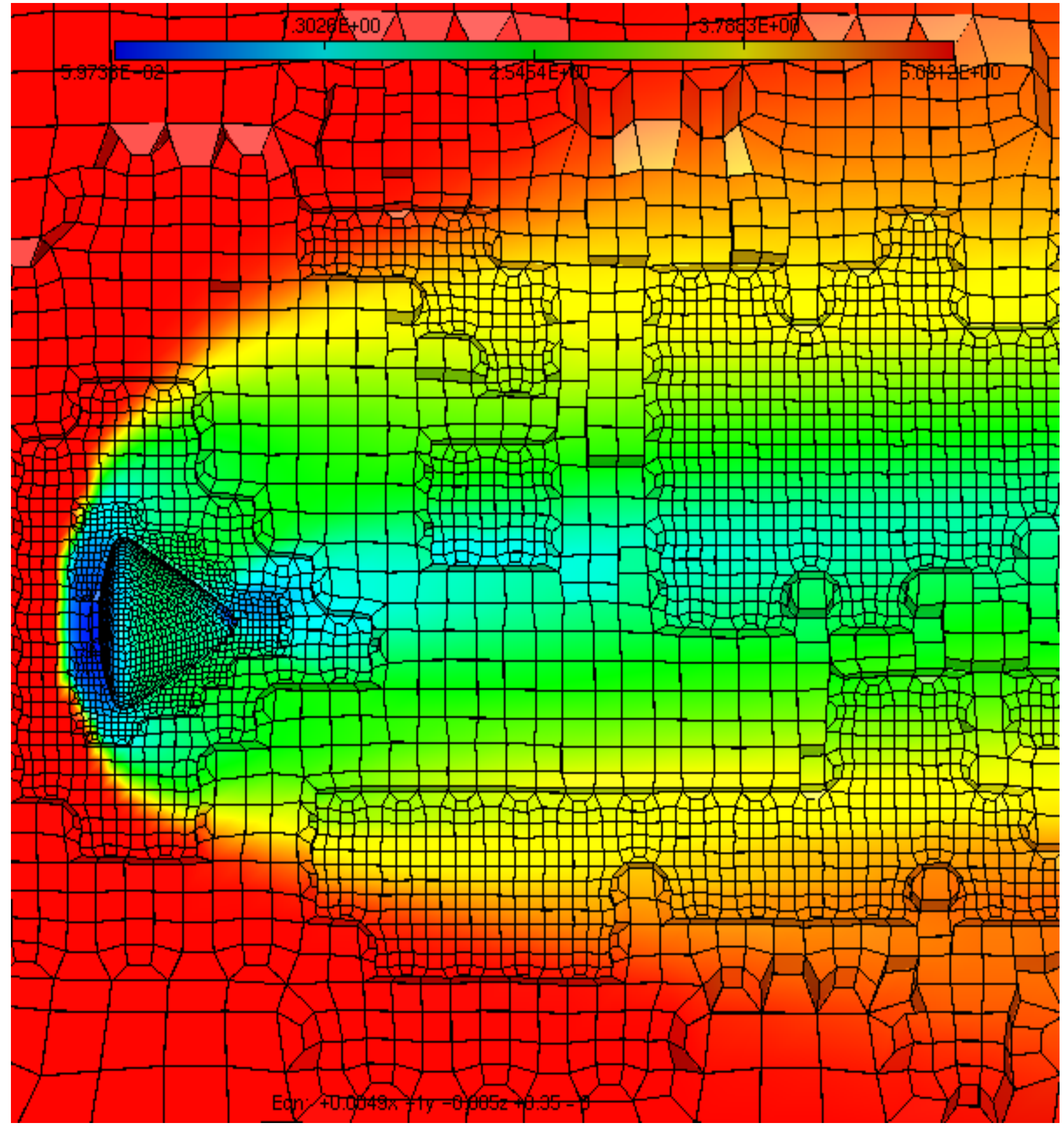
- -Only isotropic adaptation because of the octree method : no anisotropy like in tet meshes.
- -Twofold size ratio between neighboring elements : no smooth size transitions.



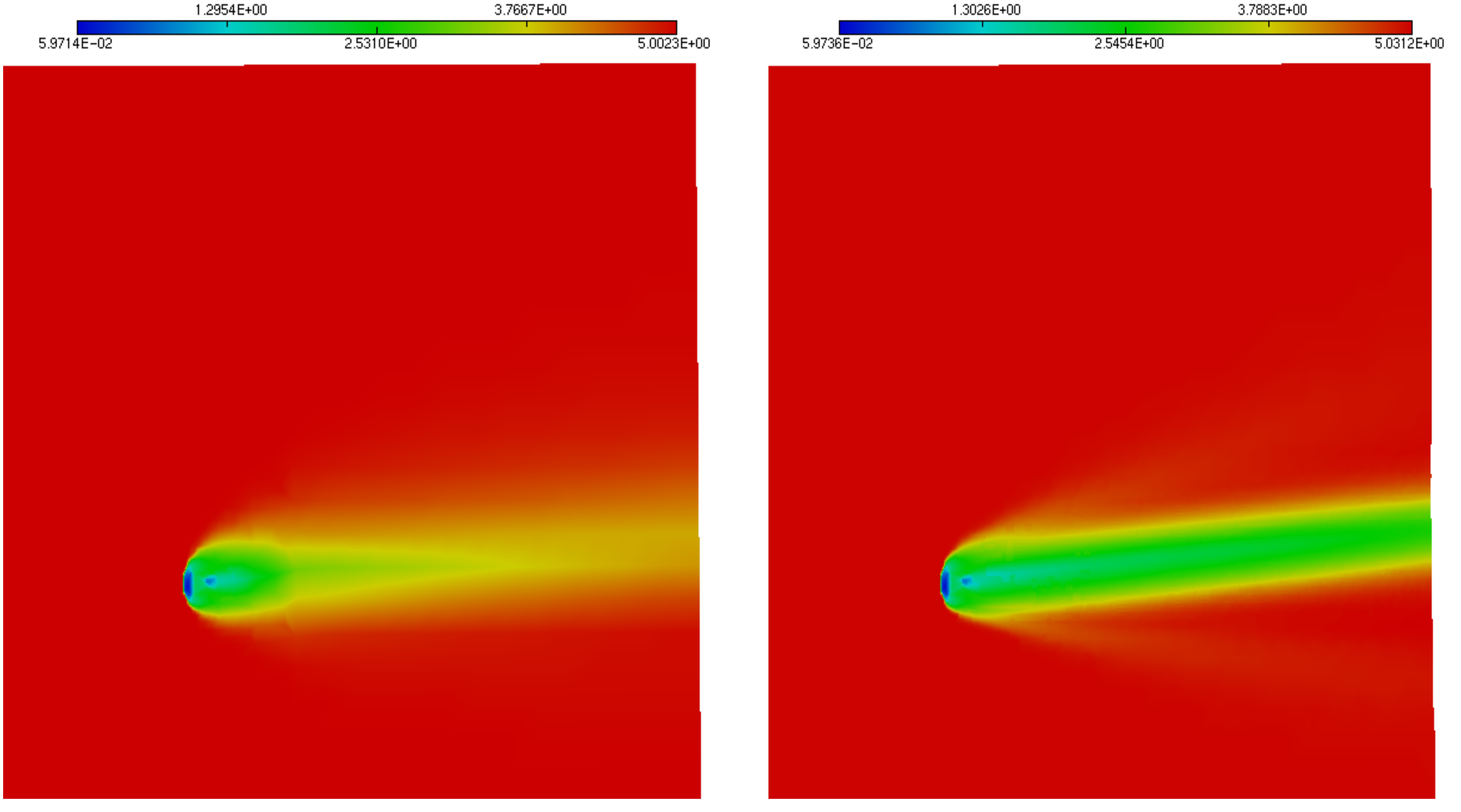


9b / Mesh adaptation: the apollo capsule

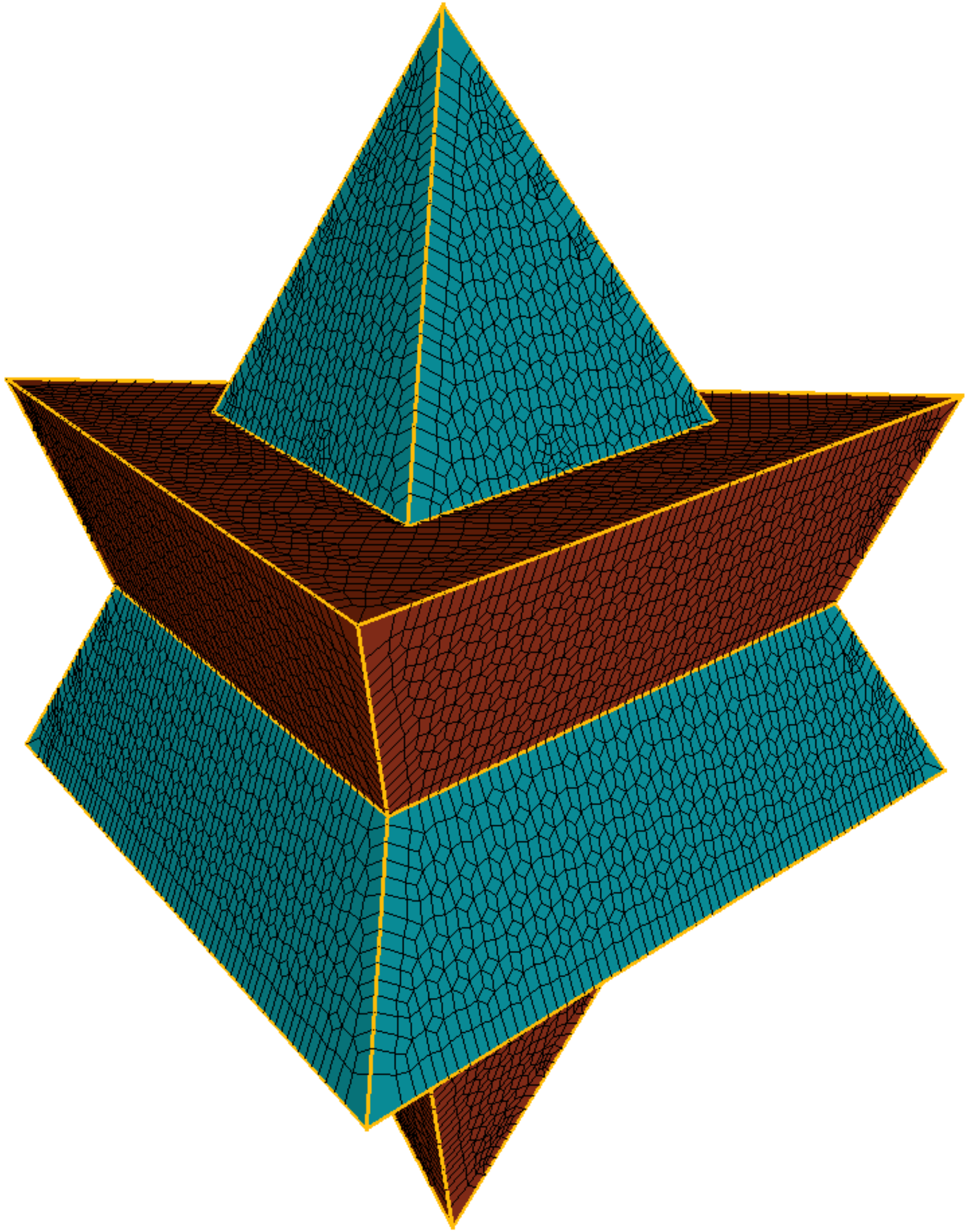




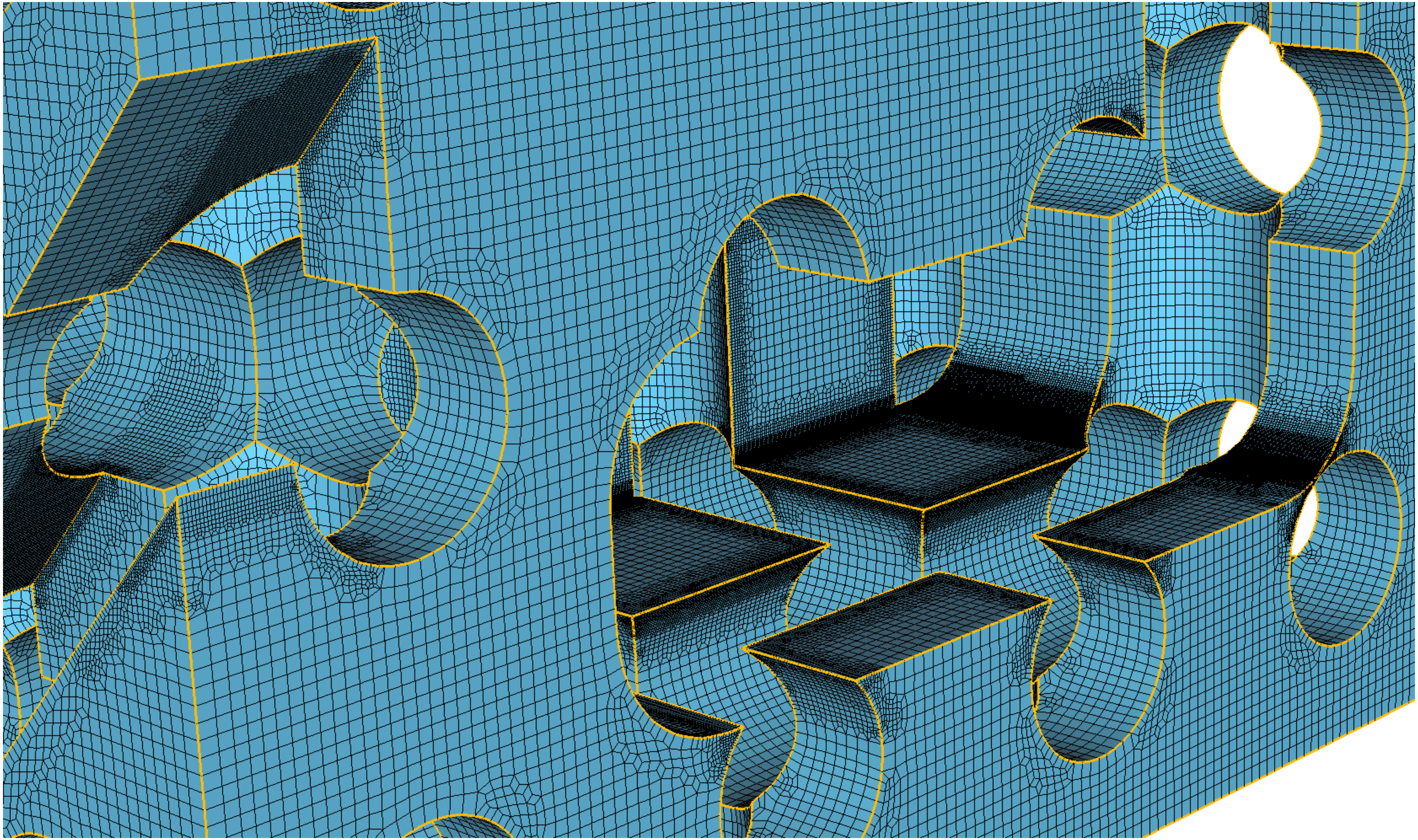
9c / Mesh adaptation: the apollo capsule



10a / Academic examples

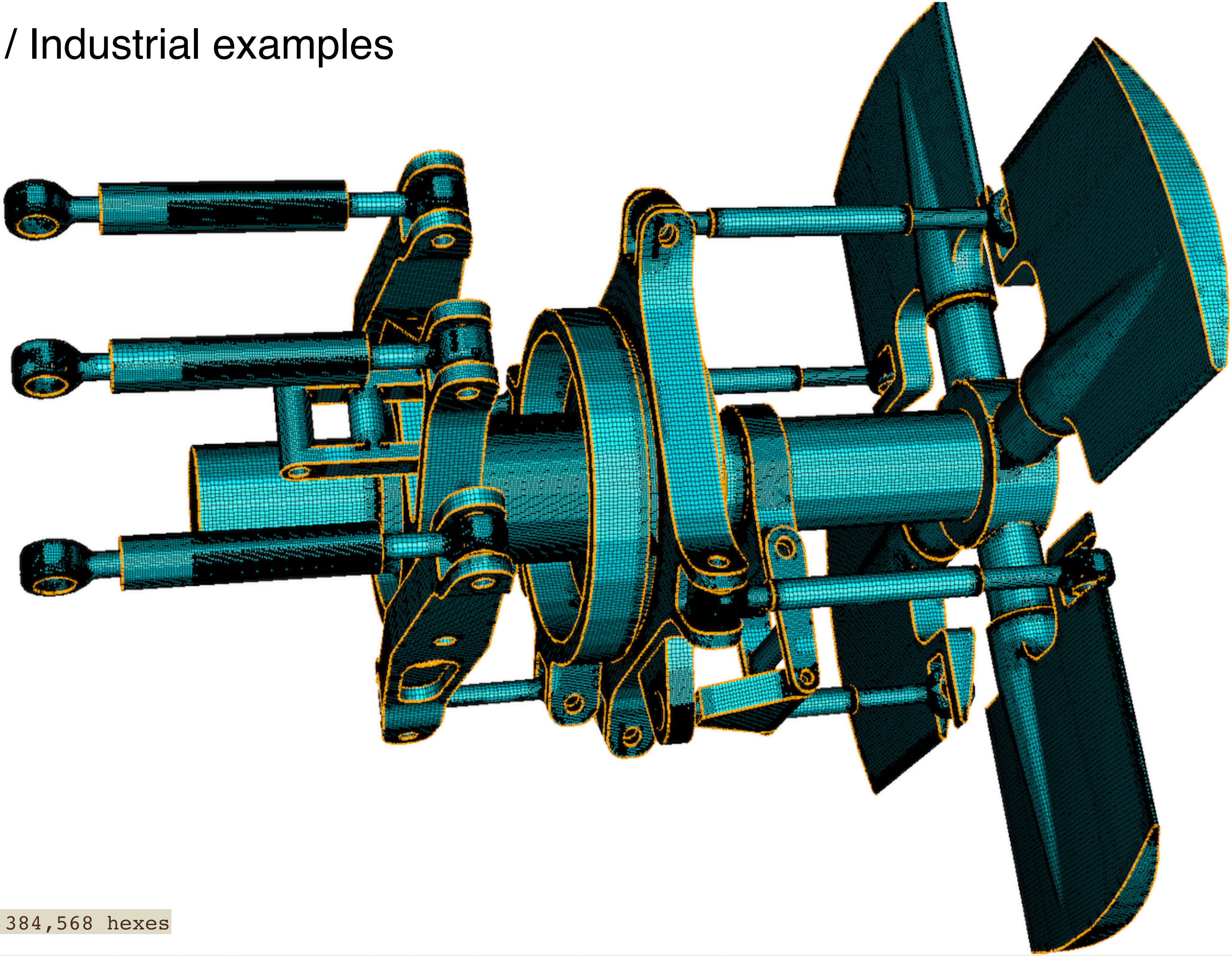


10b / Academic examples



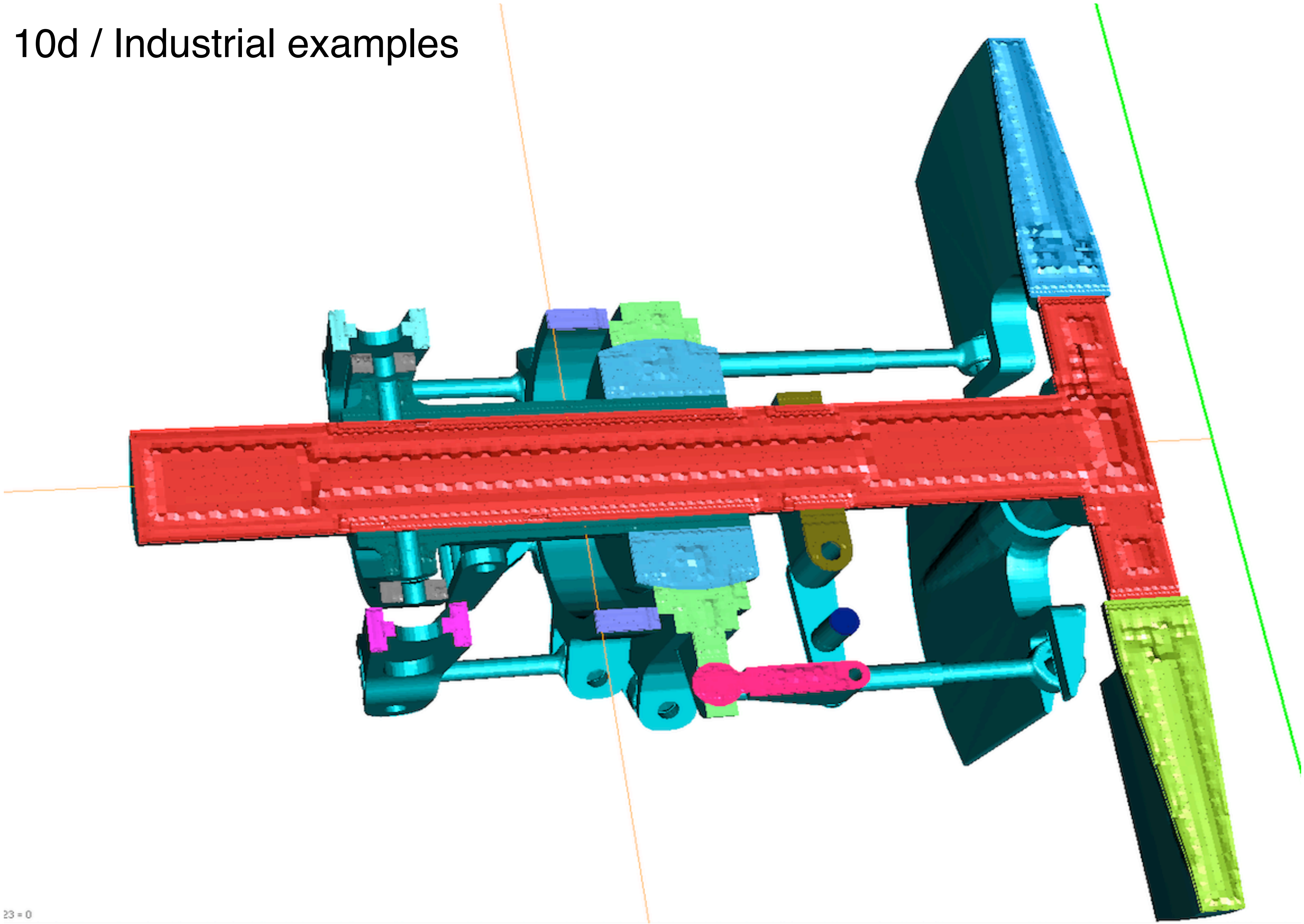
3,556,915 hexes

10c / Industrial examples

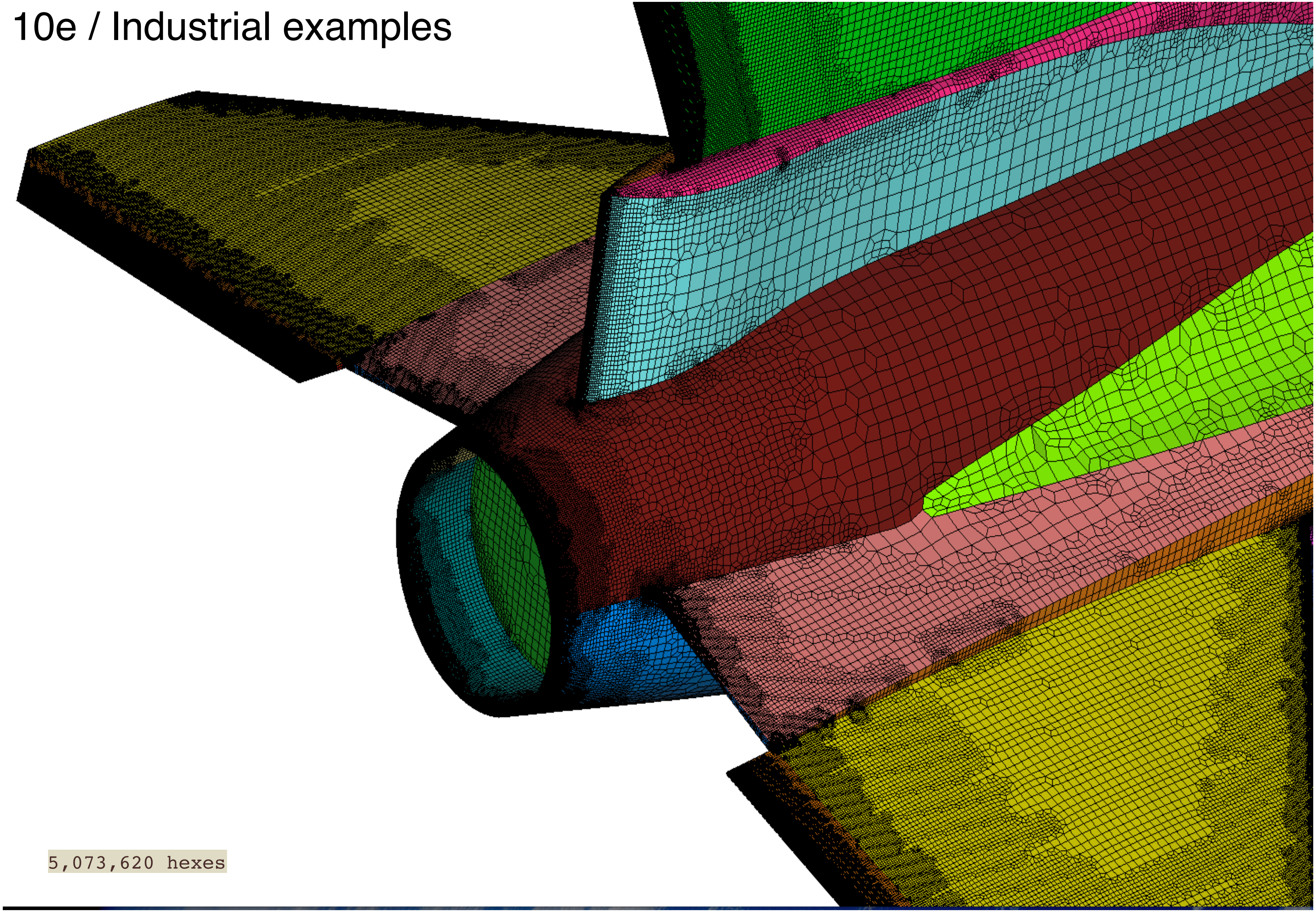


5,384,568 hexes

10d / Industrial examples



10e / Industrial examples

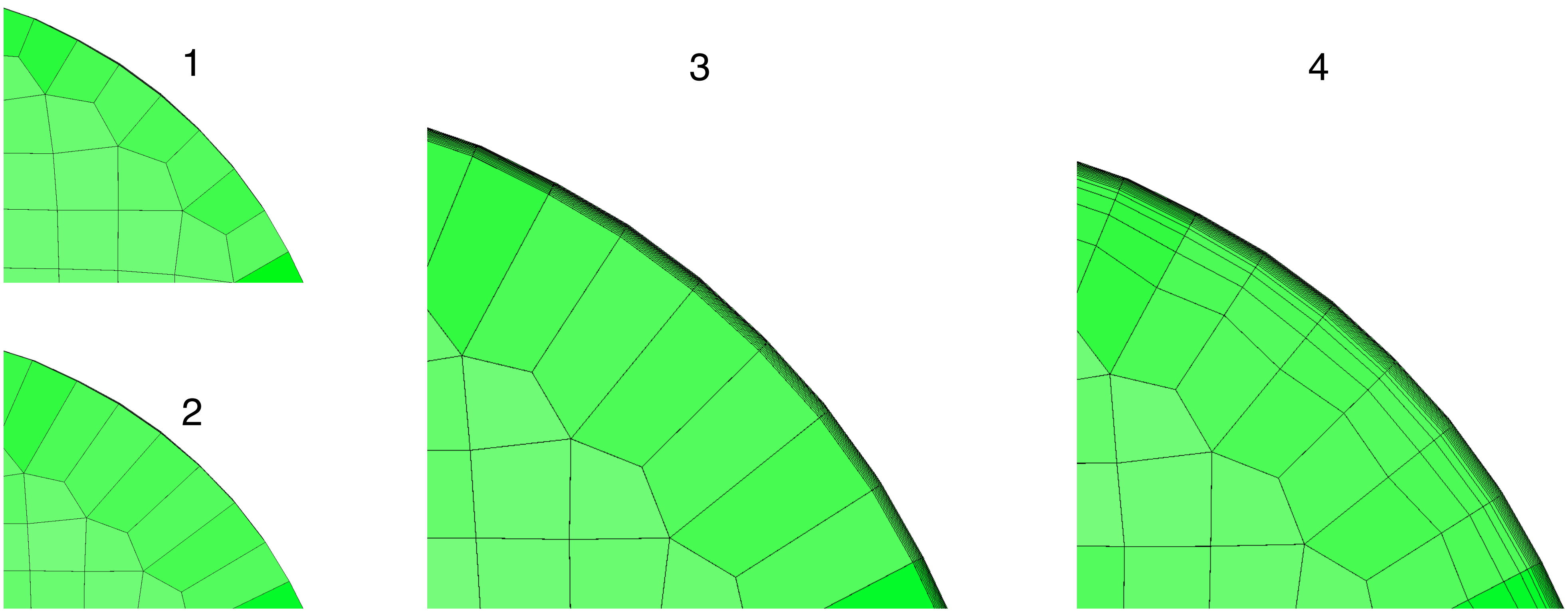


11 / Boundary layers

Boundary layers can be inserted for CFD simulation purposes.

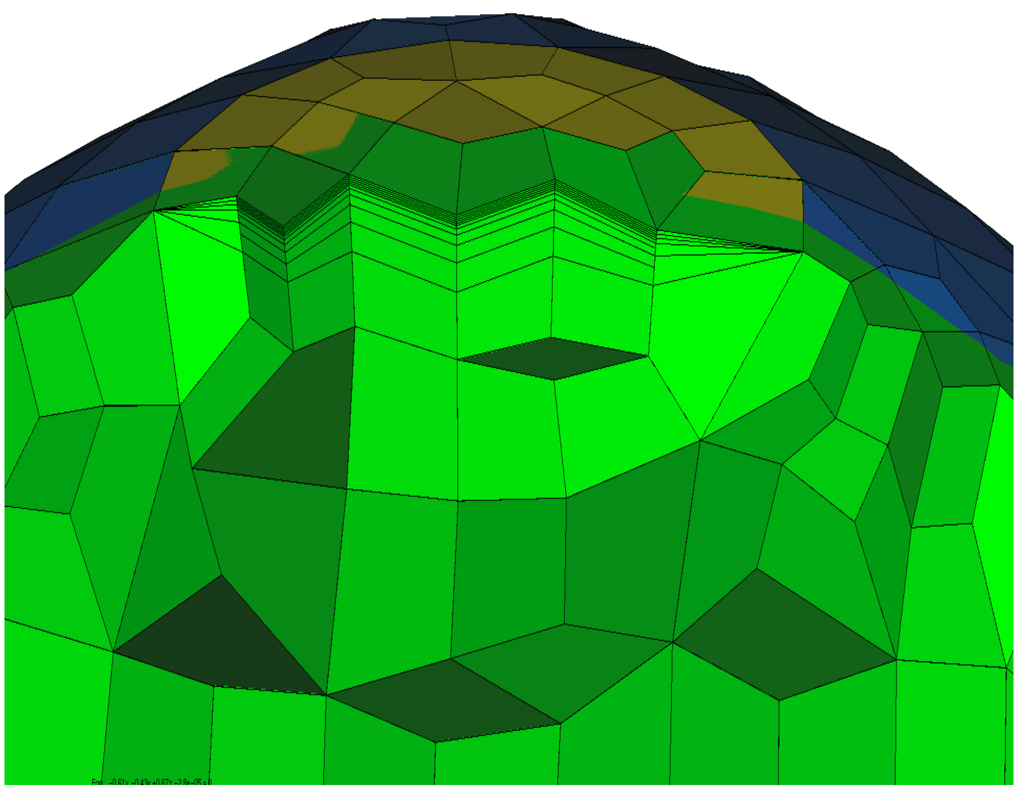
The user has control over the number, size, growth factor as well as the „stem surface references“, that is, the triangles references from which the layers will be extruded.

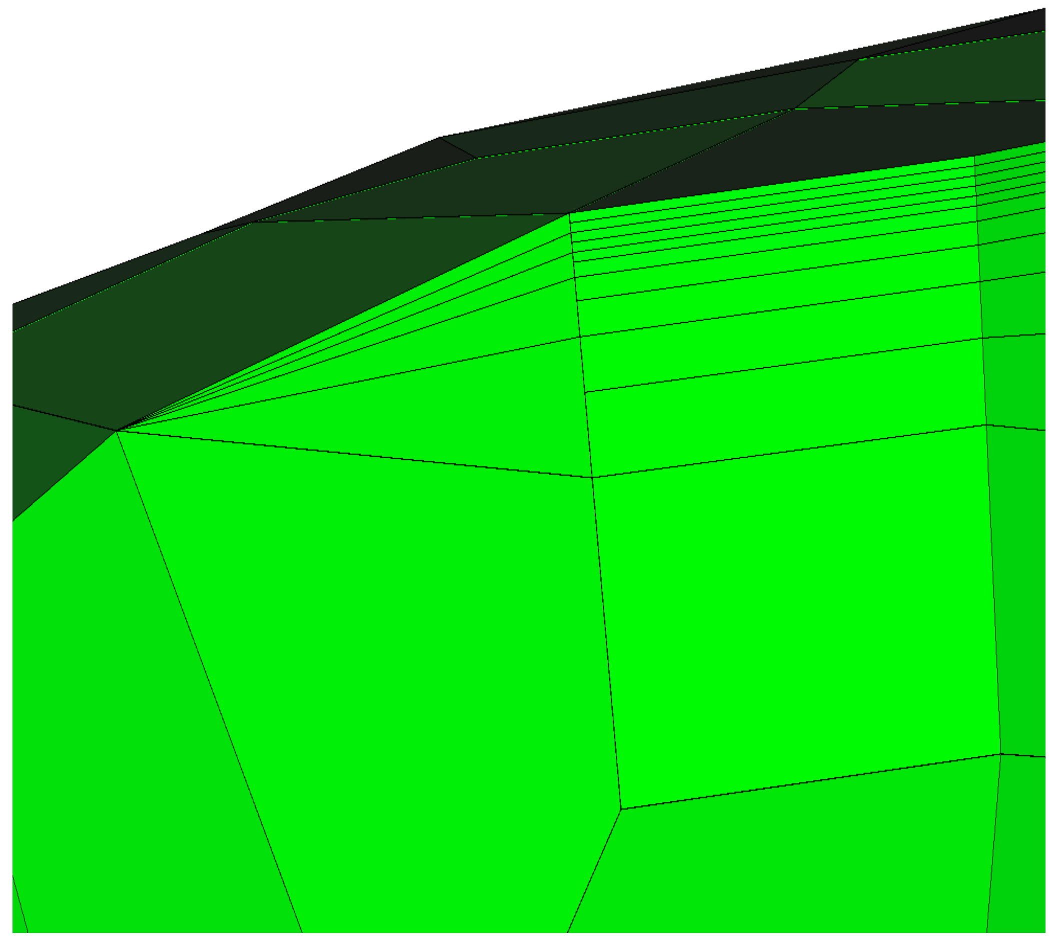
Additional blending layers may be inserted between the first set of physical layers and the free hex mesh in order to mitigate the size transition so that no element is more than two times bigger than its neighbors.



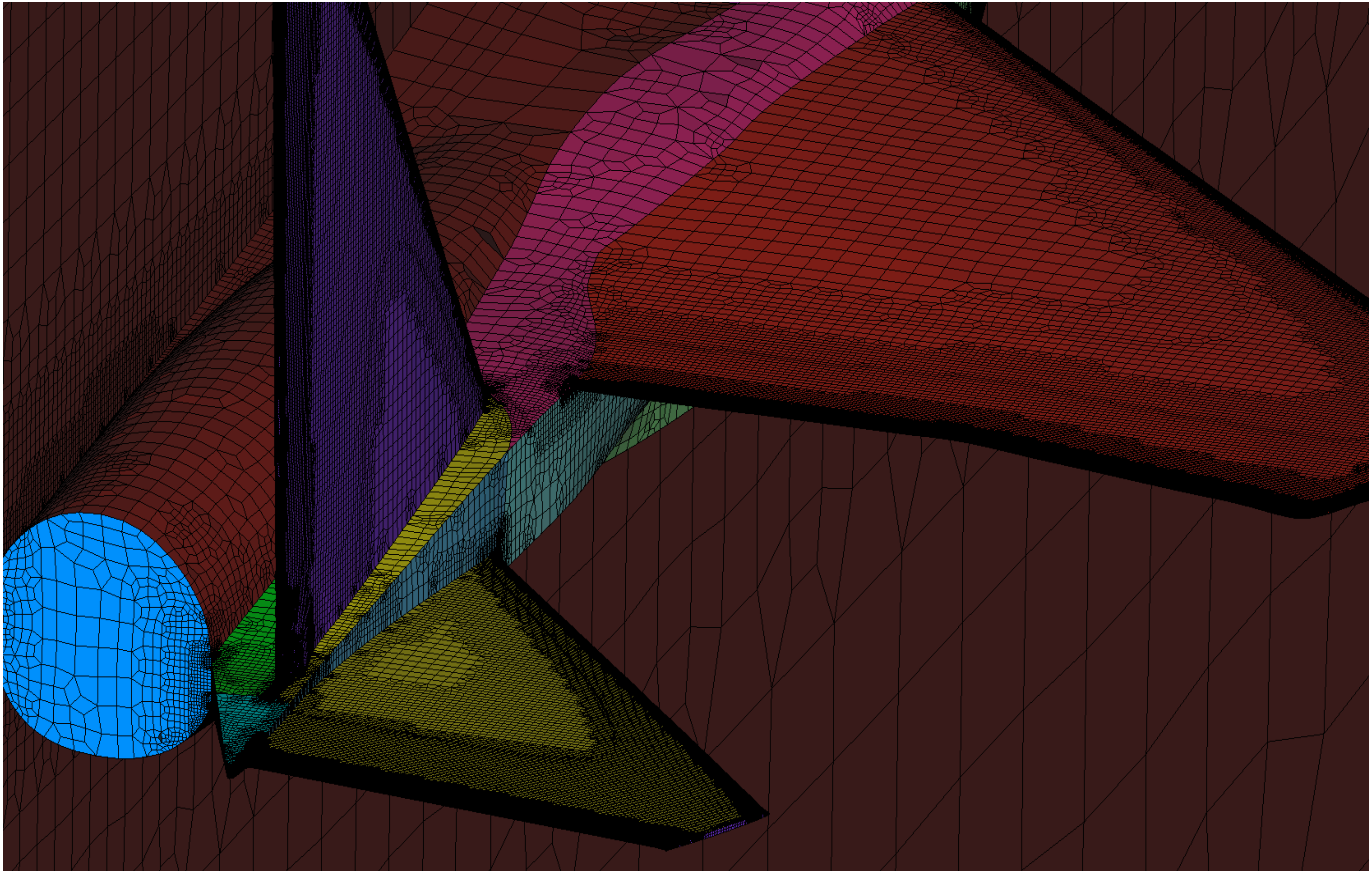
11b / Boundary layers

One set of layers may be specified for each surface references.
A pair of layered elements end up as a wedge like degenerate hex.

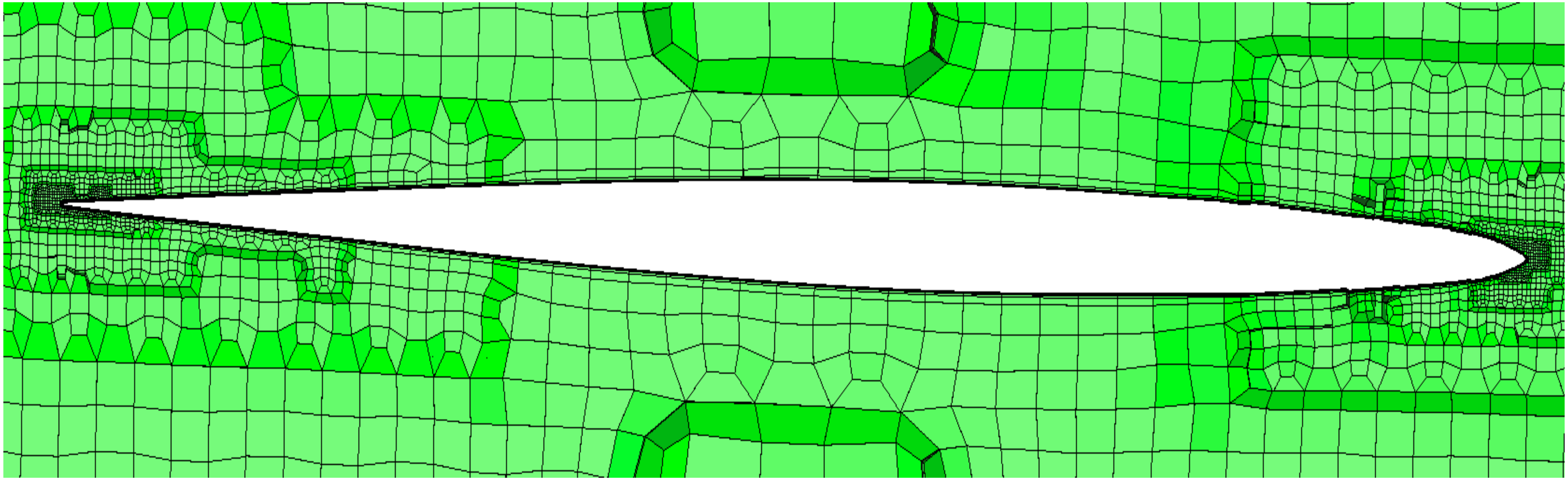


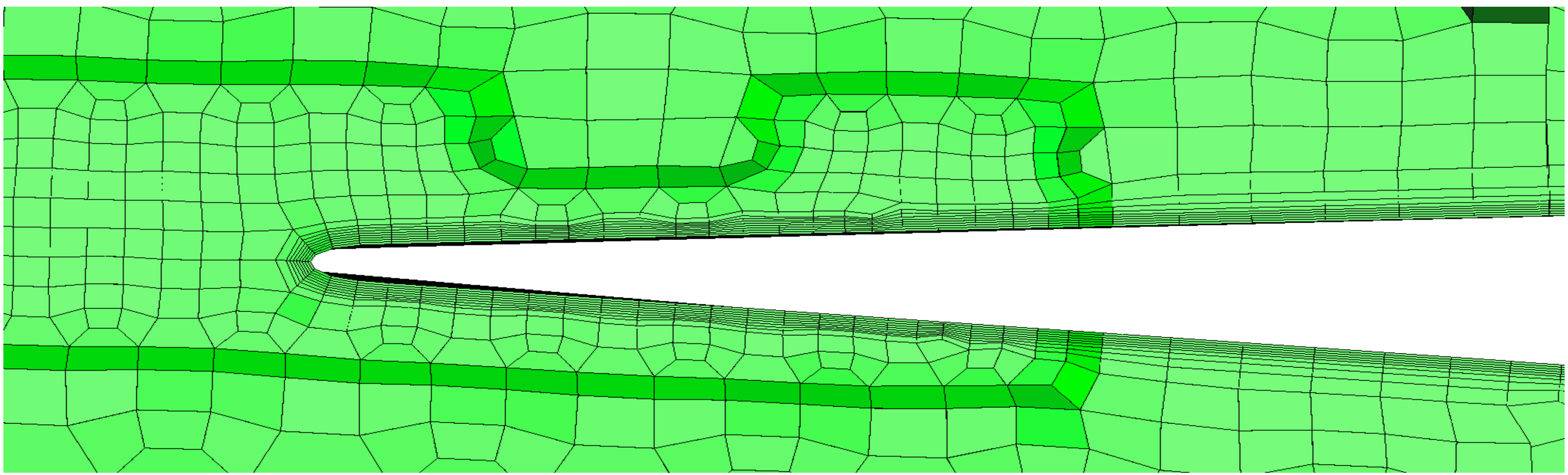


11c / Boundary layers: F15: 4,258,610 hexes



11d / Boundary layers





10 / Strength and weaknesses of the method:

+ Robust : it always produces a result (but it may not be the meshed you dreamed of...)
+ Fast : 6.000.000 elements / minute on this laptop.
+ Simple : command-line program requiring few arguments (min,max sizes, sharp angle threshold)
+ 100 % hexahedral and conformal meshes (no pyramids, prisms or hanging nodes).

-Generates too many elements when it comes to thin geometries (blades, wings, etc...)
-Isotropic meshing only.
-Angles sharper than 60° are smoothed out: otherwise, hexes gets too distorted.
-Unstructured meshes (although larger parts of meshes are grid-like).

Works under way:
+ Lowering the number of elements at the expense of the grid structure.
+ Symmetrical meshes.
+ Boundary layers in-print on the surface.
+ Q2 elements.

Conclusion : there is still a long way to go to reach the "holly grail" of hex meshing !