

# PhD thesis proposal: Combine grid placement with other meshing tools

## PhD supervisors:

Nicolas Ray	INRIA	<a href="mailto:Nicolas.Ray@inria.fr">Nicolas.Ray@inria.fr</a>
Dmitry Sokolov	University of Lorraine	<a href="mailto:Dmitry.Sokolov@univ-lorraine.fr">Dmitry.Sokolov@univ-lorraine.fr</a>
Franck Ledoux	CEA	<a href="mailto:Franck.Ledoux@cea.fr">Franck.Ledoux@cea.fr</a>

## I. Proposal's context

Numerical simulations solve physical equations to better understand complex phenomena. It is a key aspect in both industry and science: the idea is to replace costly physical experiments (e.g. wind tunnel in fluid dynamics) with computer simulation. This dramatically reduces the overall cost of product development, because it allows to prune hypotheses before undertaking expensive real actions such as the realization of a mechanical part, on site investments such as drilling of oil wells, and then submitting it to tests. One of the central points in numerical simulations is the ability to represent the functions (temperature, pressure, speed, etc.) on the studied objects. The most versatile way to do so is to discretize the domain of the interest into the cells of a mesh. But mesh generation is a notoriously difficult task. **NASA indicates mesh generation as one of the major challenges for 2030, and estimates that it costs 80% of time and effort in numerical simulation [Slotnick2014].** The main difficulty is due to the need for constructing supports that match both the geometry and the physics of the system to be modeled. More specifically, hexahedral meshes are known to be very hard to produce, if only because of the “rigid” nature of hexahedra limiting the potential combination with their neighbors, but they exhibit unique properties that are mandatory for certain physics simulations (deformation mechanics, fluid dynamics, etc.).

Today, a majority of pure or highly dominant hexahedral meshes is still hand-made, because no method has proven yet to be versatile and largely applicable enough with an acceptable accuracy, thus inducing large costs and time overheads. Figure 1 shows an example of a hand-made mesh; it took more than one month of engineer time to build it. Our ambition is to reduce this time.

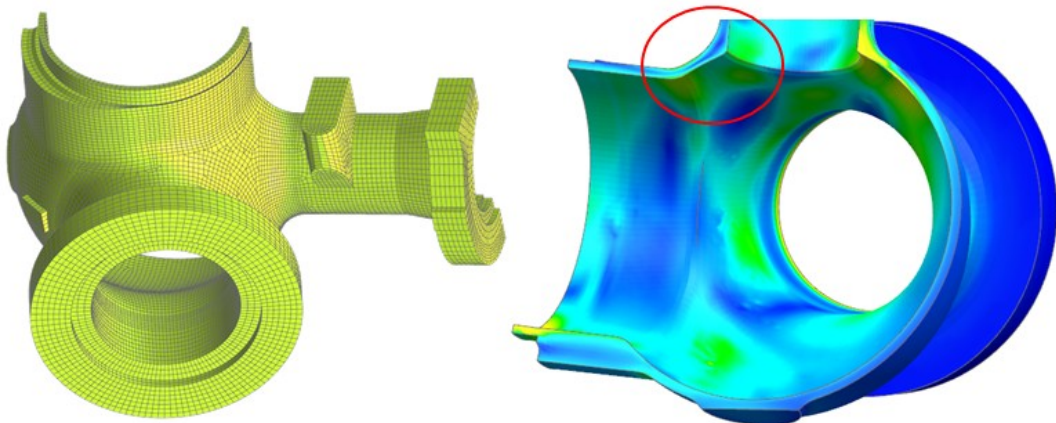


Figure 1: Examples of nonlinear thermo-mechanical structure computation (SIEMENS): Current generation time for this mesh is 30 days (230 000 elements). A typical high gradient area is highlighted in red.

A new and original approach to generate hexahedral meshes was proposed recently by the team [Sokolov2016, Ray2018]. It comes from the observation that good quality hexahedral meshes look like a deformed grid almost everywhere. The idea is to define a deformation of the object such that if the final hexahedral mesh (the result) undergoes this deformation, it will match a unit, axis aligned grid. The direct application of this idea computes this deformation, applies its inverse to the unit grid, and obtains a hexahedral mesh. In practice, we introduce more degrees of freedom by considering global parameterizations instead of deformation. In this case, the deformation functions have some

discontinuities that make it possible to represent a much larger family of hexahedral meshes: the deformed grid can be cut and glued to itself in a non-trivial way (refer to Figure 2).

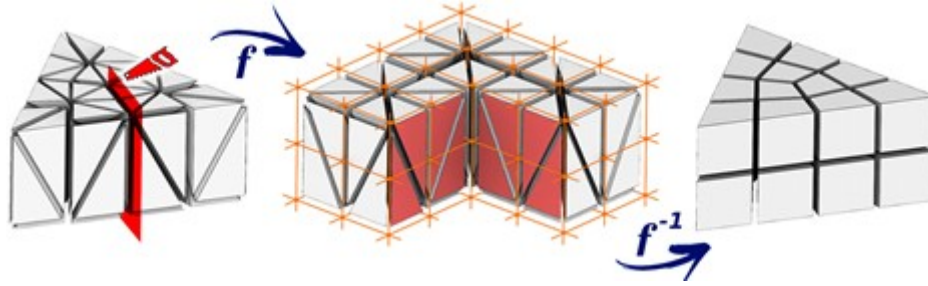


Figure 2: Global parameterization example: the input is cut open along the red plane (left), then it is deformed (middle), and the pre-image of the orange grid produces a hexahedral mesh (right).

Global parameterizations are constructed in two steps: the *Frame Field (FF)* step defines the orientation of the grid at each point of the domain, and the *Cube Covering (CC)* step generates a global parameterization that will align the final mesh with the orientation (FF) defined in the first step. When both steps succeed, the result is a very regular full hexahedral mesh. Unfortunately, the FF step may generate an orientation field with a topological structure (its singularity graph) that is not valid for the second step, and the CC step often generates invalid mappings with (locally) negative Jacobians.

Placing a (deformed) grid inside a volume gives a very regular hexahedral mesh. However, it is very difficult to fill the whole volume with such a grid: the boundary alignment constraints may be locally (fillet, chamfer) or globally (thread on screws) incompatible. In these cases, a global parameterization algorithm simply fails to mesh the model. To deal with this issue, we will integrate the grid placement algorithms in a more complete meshing process. To do so, we will analyze the conflicting constraints and solve them by pre-processing and post-processing. We will also try to better exploit the combinatorial structure of the grid, despite possible geometric inversions, as it is locally done in [Lyon2016].

Improving only the global parameterization will not be sufficient for two reasons: local constraints often require a complex combinatorial configuration (not a deformed grid) to be respected, and the positivity of the parameterization Jacobian is not guaranteed. We will identify and manage complex local configurations using our previous results [Staten2009, Kowalski2012a], and explore means to exploit parts of the global parameterization without explicitly requiring needing positive Jacobian. We have tried this latter approach in [Kowalski2012b, Kowalski2014], and we plan to produce block structured hexahedral meshes by tracing hexahedron layers from the global parameterization.

We can actually generate hexahedral dominant meshes by a post-processing step that remeshes the part of the model that is not successfully filled by hexahedra (Figure 3). It makes the process robust, but it would be much better to detect conflicting constraints earlier, before it deteriorates the global parameterization. Indeed, the hexahedral meshes have a “very rigid” combinatorial structure that makes it possible for a (local) constraint to have an impact on the whole mesh. This new way to think about the problem leads to interesting theoretical questions: what are the conditions on the boundary conditions for a valid global parameterization to exist.

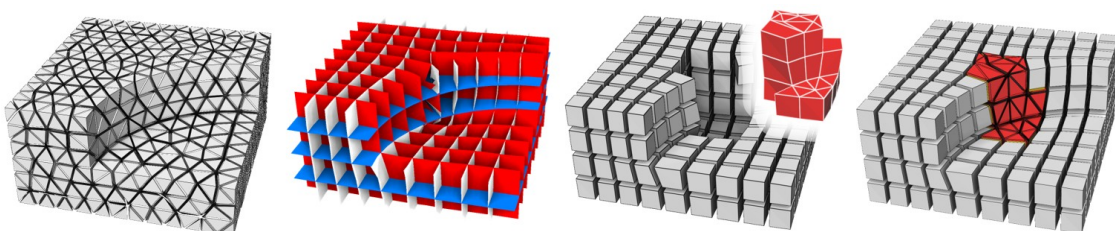


Figure 3: We compute a GP from the input geometry, we mesh failure regions, and merge them into a hexahedral dominant mesh.

## II. Organisation of the project

The project will be split into two main parts:

- Months 1-18: pre-processing the model. Here, we will address the pre-processing of the model to help the GP. We will focus on conflicting configurations of the feature edges. The starting point will be to observe failure cases, define the regions that will be impacted, and remove them from the domain to be remeshed (the post-processing will deal with it). Then, we will better manage these regions by replacing them by local hexahedral mesh patterns, and we will try to get an exhaustive classification of conflicting cases.

We will manage this problem at two levels: by proposing new solutions to local conflicting configurations, and by determining mesh configurations along feature edges that respect global constraints [Kowalski2012a]. The first aspect builds on top of the results of our results on polycubes, with more degrees of freedom to solve them. Indeed, inner singularities can prevent local conflicting cases to occur, and local meshing patterns can generate more types of constraints on the surrounding frame field.

- Months 18-36: Extract hexahedral mesh from integers variables only. The basic idea of using global parameterization is to define the hexahedral mesh as the pre-image of the unit grid by a parameterization function. This process assumes that the parameterization function is locally one to one, with consistent orientation enforced by the positive Jacobian condition. In practice, this is very difficult to enforce, but robust extraction [Lyon2016] accepts some local constraint violation. It is very interesting to consider that this algorithm works well thanks to the variables that are constrained to be integers; this is the same set of variables that defines the combinatorial structure of the hexahedral mesh. We would like to improve the robustness of the process by completely ignoring the geometry of parameterization. In theory, we could extract the hexahedral mesh combinatorial using only variables that are constrained to be integers. The geometry can always be optimized directly on the final mesh, that is likely to be a better discretization for this purpose.

If we consider the case of polycubes, the combinatorial structure of the hexahedral mesh is quite easy to extract; the length of the edges of the polycube are directly given by the difference of its vertices coordinates, that are constrained to be integers. It determines directly the combinatorial structure of the final mesh and let us experiment to directly optimize the geometry on the hexahedral mesh. From this point, we can start introducing a singularity graph and extend the combinatorial reconstruction with it. We will experiment it by introducing singularity graphs of increasing complexity until we can generalize the approach to any singularity graph. The risk is that the problem may be too difficult to tackle, but we will at least have a solution for simple cases, plus the geometric optimization of the hexahedral mesh geometry that could help to improve our other results.

### III. References related to the project

- [**Kowalski2012a**] N. Kowalski, F. Ledoux, M.L. Staten, S.J. Owen, Fun sheet matching: towards automatic block decomposition for hexahedral meshes, *Engineering with Computers*, 2012.
- [**Kowalski2012b**] N. Kowalski, F. Ledoux, P. Frey, A PDE based approach to multi-domain partitioning and quadrilateral meshing, *International Meshing Roundtable 2012*.
- [**Kowalski2014**] N. Kowalski, F. Ledoux, P. Frey, Block-structured Hexahedral Meshes for CAD Models Using 3D Frame Fields, *International Meshing Roundtable 2014*.
- [**Lyon2016**] M. Lyon, D. Bommès, L. Kobbelt, HexEx: Robust Hexahedral Mesh Extraction, *ACM Transactions on Graphics*, 2016
- [**Ray2018**] N. Ray, D. Sokolov, M. Reberol, F. Ledoux, B. Lévy, Hex-dominant meshing: Mind the gap!, *Computer-Aided Design*, Volume 102, 2018.
- [**Slotnick2014**] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, G. William, L. Elizabeth and D. Mavriplis, *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*, NASA/CR-2014-218178, NF1676L-18332.
- [**Sokolov2016**] D. Sokolov, N. Ray, L. Untereiner and B. Lévy. 2016. Hexahedral-Dominant Meshing. *ACM Transactions on Graphics* 35.
- [**Staten2009**] M.L. Staten, J.F. Shepherd, F. Ledoux, K. Shimada, Hexahedral Mesh Matching: Converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces, *International journal for numerical methods in engineering*, 2009.