Towards polyhedral meshing algorithms in Salome: a study of available algorithms and development of a new one

Organization: CEA Saclay LGLS (Laboratoire de Génie Logiciel pour la Simulation)

Subject: Meshing, Numerical simulations, CFD

Context: At present computer aided numerical simulations are often used for design, analysis, and optimization for large variety of engineering and physical systems. It would be fair to say that everyday thousands of numerical simulations are performed around the globe. To perform these numerical simulations one of the primary step is meshing – process of converting a numerical domain (CAD) into a two-dimensional or a three-dimensional discretized domain (mesh). Certainly, obtaining a high-quality mesh then becomes a critical factor for many numerical simulations. Meshes with triangles and/or tetrahedral elements are easy, fast, and robust to generate. While many numerical methods rely on a triangular or tetrahedral mesh, some numerical methods give better and faster results with a quadrangular and/or hexahedral mesh. Unfortunately these meshes are much tedious to generate, often requiring long human hours to decompose the shape into hexahedral domains. An alternative approach – polyhedral meshing, which is gaining popularity nowadays, relies on use of polyhedral elements. They provide a good trade-off between good geometry approximation, good numerical properties, and ease of generation. Moreover, such meshes have also proven to work better for certain simulations, e.g., flow through a water jacket of an engine, besides other advantages [3]. Some numerical simulation tools developed in CEA/DES and EDF R&D are now able to deal with polyhedral elements as meshes.

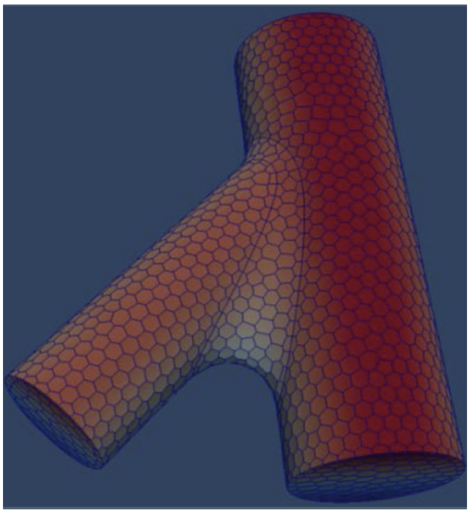

Figure 1: polyhedral mesh of a pipe junction [2].

Generally, polyhedral meshes can be classified into two categories. In the first, one fills the domain with elements of mostly hexagonal faces like honeycombs or soccer balloons, as in figure 1. One method to get this kind of polyhedral meshes is to use the dual mesh of a tetrahedral mesh.

The second one aims to mix the conformal and non-conformal mesh elements, and is used for meshing geometries with different mesh sizes in different zones, cf. Figure 2. Such meshes also arise when adaptive mesh refinement is followed, to enhance the mesh quality pertaining to certain physics.

The objective of the proposed internship is to provide these two kinds of polyhedral mesh generation algorithms in SALOME numerical platform [4]. It will be divided into three parts.

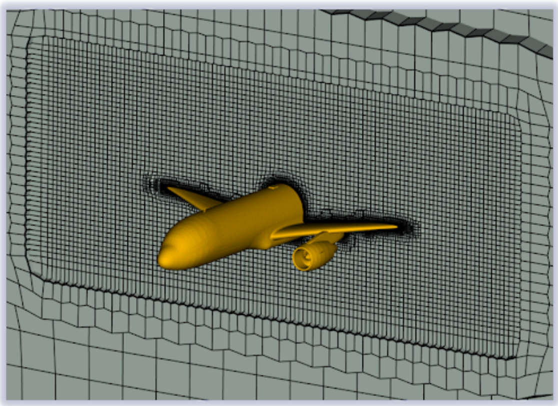

Figure 2: polyhedral mesh of CFD domain around an airplane, created by cfMesh [1].

First, to identify and analyze available open-source polyhedral meshing algorithms. Second, depending on the analysis, make the algorithms more robust or develop a new one. Third, integrate the polyhedral meshing algorithms into SALOME platform as mesh plugins.

Objectives:

- Reviewing the bibliography and former works on polyhedral meshing;
- Making the algorithms more robust or developing a new one;
- Integrating the algorithms in SALOME;
- Writing of a detailed report.

Desired start: March/April 2022

Period: 6 months

Grade: Internship Bac +5

Salary: According to CEA agreements

Requested knowledge:

- Skills in scientific computing and meshing.
- A good level in C++ programming.
- Ability to work with Linux environment.
- Very good level in English.

Locality: CEA Paris-Saclay, at (LGLS Lab)

To apply: Send your CV and grades of your M1 and/or M2 to

- Christophe BOURCIER : christophe.bourcier@cea.fr
- Mohd Afeef BADRI : mohd-afeef.badri@cea.fr

Deadline: 31/03/2022

References:

[1] *cfMesh: A Library for Automatic Mesh Generation.* `https://cfmesh.com/cfmesh/`. Accessed: 2021.

[2] Sang Yong Lee. "Polyhedral mesh generation and a treatise on concave geometrical edges". In: *Procedia Engineering* 124 (2015), pp. 174–186.

[3] Milovan Peric and Stephen Ferguson. "The advantage of polyhedral meshes". In: *Dynamics* 24.45 (2005), p. 504.

[4] *SALOME, the open-source numerical platform for numerical simulation.* `https://www.salome-platform.org`. Accessed: 2021.