

Status of the Polyhedral Mesh Generator using SALOME PLATFORM and TetGen

Sang Yong Lee*, Chan Eok Park, Kim Shin Whan
KEPCO E&C Company, Inc., 150 Deokjin-dong, Yuseong-gu, Daejeon, 305-353
*Corresponding author: sanglee@kepco-enc.com

1. Introduction

Recently developed porous body approach codes such as SPACE[1] and CUPID[2] require a CAD system to estimate the porosity. Since they use the unstructured mesh and they also require reliable mesh generation system. The combination of CAD system and mesh generation system is necessary to cope with a large number of cells and the complex fluid system with structural materials inside. In the past, a CAD system Pro/Engineer[3] and mesh generator Pointwise[4] were evaluated for this application. But, the cost of those commercial CAD and mesh generator is sometimes a great burden.

Therefore, efforts have been made to set up a mesh generation system with open source programs. The evaluation of the TetGen[5] has been made in focusing the application for the polyhedral mesh generation[6].

In this paper, SALOME[7] will be described for the efforts to combine TetGen with it. In section 2, brief introduction will be made on the CAD and mesh generation capability of SALOME and Tetgen. Polyhedral mesh generation process will be described in section 3. Conclusion will be followed.

2. SALOME and TetGen

2.1 SALOME Platform

SALOME is open source software that provides a generic platform for Pre- and Post-Processing for numerical simulation. It is based on an open and flexible architecture made of reusable components. SALOME consists of many modules. It provides a generic user-friendly and efficient user interface, which helps to reduce the costs and delays of carrying out the studies. It reduces training time to the specific time for learning the software solution based on this platform. It provides access to all functionalities via the integrated Python console. It create/modify, import/export (IGES, STEP, BREP), repair/clean CAD models. It can mesh CAD models, edit mesh, check mesh quality, import/export mesh (MED, UNV, DAT, STL). It can handle physical properties and quantities attached to geometrical items. It performs computation using one or more external solvers. It displays computation results.

2.2 TetGen Mesh Generator

One of the conspicuous problems to use TetGen is that it can't handle CAD model. It only accepts

processed geometric systems such as POLY or STL file formats. It is inevitable for TetGen to be integrated with some CAD to be a full pledged polyhedral meshing system. This is the reason why investigation is made, on combining TetGen with SALOME that has CAD capability.

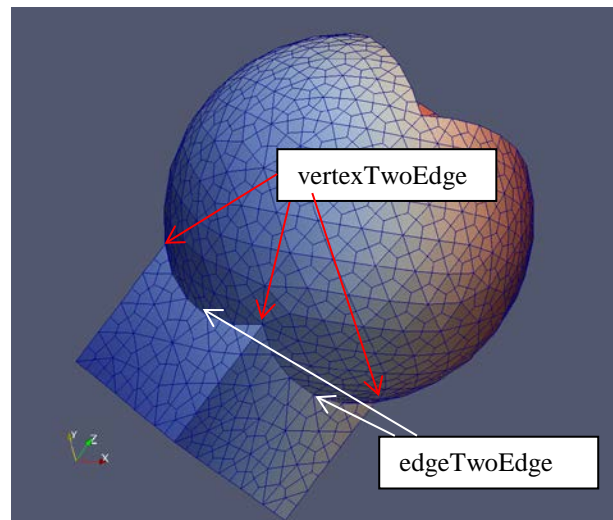


Fig. 1. TetGen generated Polyhedral Mesh (view-1).

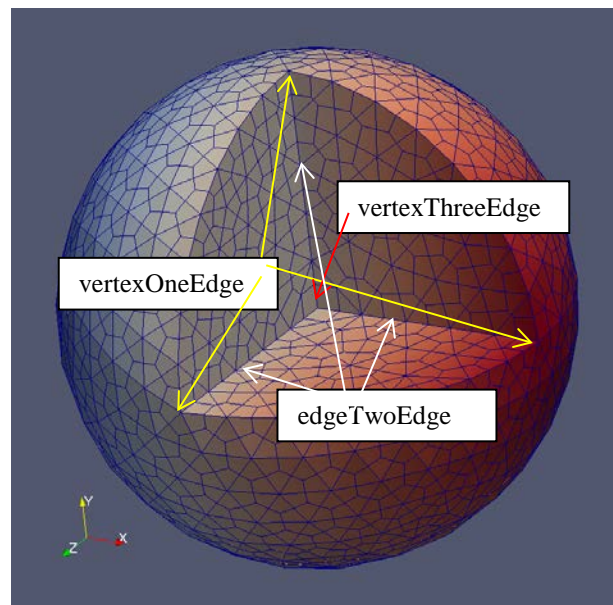


Fig. 2. TetGen generated Polyhedral Mesh (view-2)

3. Polyhedral Mesh Generation

Even though the polyhedral mesh generation using Delaunay-Voronoi dual property is simple in principle,

there are many complicated steps to follow to get final results. To make steps minimized, it is assumed that the faces and cells at the boundaries have the Gabriel property. Voronoi cell generation procedures are presented as the following algorithm.

Algorithm –I: Algorithm to create dual mesh.

- 1: Create a dual vertex at a Voronoi point in each primal mesh region.
- 2: Create a dual vertex at a Voronoi point in each primal mesh face on the boundary.
- 3: Create a dual vertex at mid-point of each mesh primal edge on a model edge.
- 4: Create a dual vertex at each primal mesh vertex on a model vertex.
- 5: Create an interior dual face corresponding to each primal mesh edge on the interior of the domain.
- 6: Create one or more interior dual faces corresponding to each primal mesh edge on the boundary.
- 7: Create one or more boundary dual faces corresponding to each boundary primal vertex.
- 8: Create one dual region corresponding to each interior primal vertex.
- 9: Create one or more dual regions corresponding to each boundary vertex.

Following the algorithm-I leads to the mesh presented in figure 1 and 2. There are two issues to be solved for the above mesh. The first one is the boundary faces. Boundary faces of Voronoi cell can be merged if they are on the same geometrical model face. The concavity of cells is the second issue to solve. It can be made convex if it is cut along the concave edges appropriately. Before the cut is made boundary faces are merged into one face on the individual solid model face. Therefore, algorithm-I, is extended as follows;

- 10: Merge boundary faces on the same model face.
- 11: Cut along concave edges.

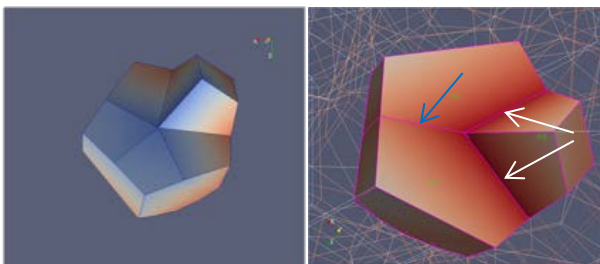


Fig. 3. A Concave Cell and its Convex Cut Cells with Merged Face

Many procedures are developed to perform Step 11. One of concave cells marked by red arrows in figure 1, for example, is shown separately in figure 3 (left). Faces on the boundary are merged first. And cuts are made along the two concave edges marked by white arrows in figure 3 (right). Then another cut is made through the plane made by concave edge cut as indicated by blue arrow in the same figure.

There are several types of concave cells. Cells along a concave geometrical edge is classified as edgeTwoEdge type as indicated in the figure 5. This type of cell is handled by the function edgeTwoEdge() in the program. Other types such as vertexOneEdge, vertexTwoEdge and vertexThreeEdge shown in figure 5, also can be cut by the functions vertexOneEdge(), vertexTwoEdge() and vertexThreeEdge() respectively. Of course, Step 10 is performed before any cuts are made.

Final mesh is presented in figure 4. As expected, most of the surface meshes are merged to be hexagonal unlike those at figure 1. Cells along concave edge and vertex are divided into convex cells.

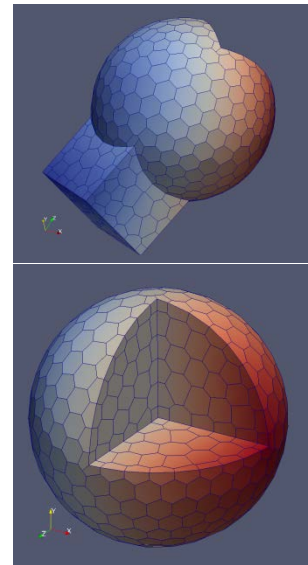


Fig. 4. Completed Polyhedral Mesh.

4. Conclusions

SALOME and TetGen codes are being integrated to construct robust polyhedral mesh generator.

Procedures to merge boundary faces and to cut concave cells are developed to remove concave cells to get final convex polyhedral mesh.

Treating the internal boundary face, i.e. non-manifold face will be the next task in the future investigation.

REFERENCES

- [1] S. J. Ha, C. E. Park, K. D. Kim, and C. H. Ban, "Development of the SPACE Code for Nuclear Power Plants", Nuclear Technology, Vol. 43, No. 1, 2011.
- [2] J. J. Jeong, et. al., "The Cupid code development and assessment strategy", Nuclear Engineering and Technology, vol. 42 no.6 December 2010.
- [3] <http://www.ptc.com/>
- [4] <http://www.pointwise.com/>
- [5] <http://wias-berlin.de/software/tetgen/>
- [6] KNS Spring mtg, 2014, "SALOME PLATFORM and TetGen for Polyhedral Mesh Generation"
- [7] <http://www.salome-platform.org/>