Visual DoMesh - A meshing software for mixed high-order finite element discretization

Christian Sorger, Stefan Kollmannsberger, and Ernst Rank

Chair for Computation in Engineering, Technische Universität München, Germany
sorger@bv.tum.de

Abstract. This paper presents the automatic mesh generator *Visual DoMesh*. This is a software which is designed to create curved hexahedral and tetrahedral elements for the discretization of mixed structures. To this end, the recursive domain division is utilized for the generation of triangular and quadrilateral meshes of freeform surfaces. Hexahedral elements are formed by a projection approach which extrudes quadrilateral elements generated on reference surfaces with respect to predefined mathematical rules. The generation of tetrahedral elements is performed by the advancing front algorithm as implemented in *Netgen*. Quasi-regional mapping is applied to represent curved high order solid elements.

Key words: Mesh generation, high-order, tetrahedral, hexahedral

1 Introduction

Many complex structures in civil engineering are computed with dimensionally reduced finite element models. A more accurate approach is to discretize structures with three-dimensional volume elements and apply a strictly three-dimensional continuum approach utilizing high-order finite elements [1].
Nevertheless, mesh generation with pure and conforming hexahedral elements is not always possible. As an alternative, the more robust tetrahedral meshing techniques may be applied. However, hexahedral meshes provide more accurate results in finite element computations. Therefore, it is of practical interest to generate hexahedral elements wherever hexahedral meshing can be applied.
To enable a full 3-dimensional discretization of the structure, the underlying geometric model is split into boundary representation volumes to be meshed with tetrahedral elements and dimensionally reduced volumes for hexahedral element generation. Within such a mixed model, the transition of physical quantities from the tetrahedral to the hexahedral elements can be achieved by applying the mortar method for finite elements of high-order [2].
The following sections describe the geometric model and the applied mesh generation techniques.

2 Geometric model

The underlying geometric model covered in this work includes a boundary-representation (BRep). Thus a hierarchical organization of the structural geometry is applied with a strict distinction between geometry and topology.

- **Node**: Represents a geometric point.
- **Edge**: Bounded by two nodes, lying on a geometric curve.
- **Wire**: Sequence of edges.
- **Region**:
  - **Boundary**: Defined by a closed wire as outer boundary.
  - **Holes**: Defined by closed wires as inner boundaries.
  - **Shape**: Defined by a geometric surface.
- **Shell**: Sequence of regions.
- **BRep-Volume**:
  - **Boundary**: Defined by a closed shell as outer boundary.
  - **Holes**: Defined by closed shells as inner boundaries.
  - **Shape**: Defined by the boundary.
- **Extrusion Volume**:
  - **Reference shell**: Describes the surfaces to extrude.
  - **Extrusion path**: Given by a constant vector, the normal direction, a reference edge or an abstract extrusion rule.
  - **Thickness**: Defined by a constant parameter or upper and lower boundary shells.

3 Visual DoMesh

Visual DoMesh is a mesh-generator and pre-processor tailored for the needs of the high-order finite element code *AdhoC*. The program is split into four main modules and provides visualization, meshing and graphical user interaction features. Figure 1 shows the program architecture of *Visual DoMesh*.

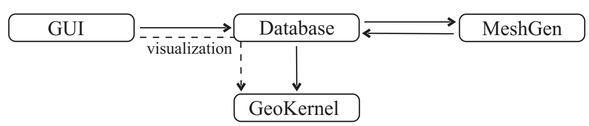


Fig. 1. *Visual DoMesh* program architecture.

3.1 Database

The database is the central module of the application and holds the topology of the imported data. Data import and export is provided by parsers for several geometry and mesh file formats. Optimal data access is guaranteed by utilizing *stl* containers.

3.2 Geometry Kernel

The geometry kernel utilizes *Open CASCADE* and provides all standard CAD shapes which are associated with the topologic entities of the database. Thus all geometric operations required by the mesh-generator *MeshGen* (sec. 3.4) may be performed. Furthermore *Open CASCADE* provides additional parsers for common standardized file-formats (iges, step, brep, csfdb, stl, vrml) and interactive visualization objects for GUI implementations.

3.3 GUI

The graphical user interface (GUI) is implemented with *Qt* and is the control-unit of the application. Besides visualization and user interaction features it is linked to the geometry kernel to enable interactive picking and selecting of the visualized objects.

3.4 MeshGen

MeshGen is the meshing module of the application. The algorithm is briefly sketched in figure 2. This section outlines the processing part of the algorithm.

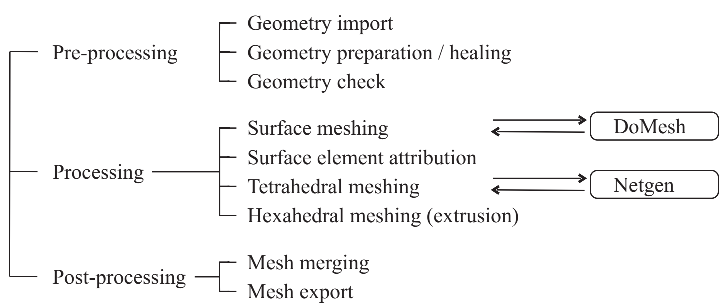


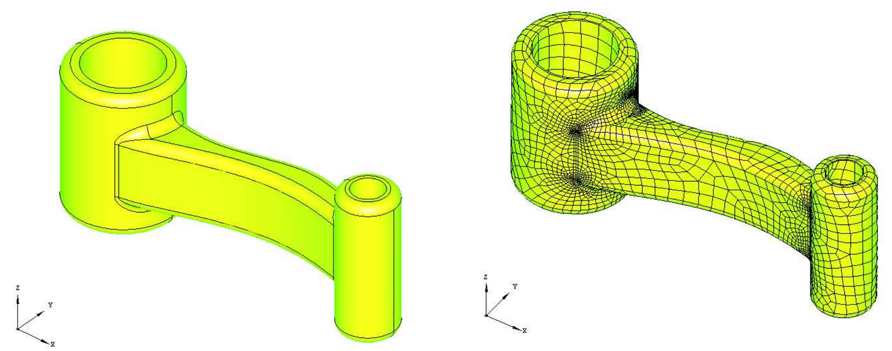
Fig. 2. MeshGen algorithm.

Surface meshing

Surface mesh generation is performed by interfacing the domain subdivision [3] mesh generator *DoMesh*. An extension to 3-dimensional freeform surfaces is achieved by coupling *DoMesh* to the geometry kernel and introduce metric mapping techniques.

Surface high-order element attribution

To benefit from the advantages of the high-order finite element method an accurate representation of the geometry is required. Curved structures, therefore, are meshed with curved elements which in turn are described by interpolating BSplines. Thus geometric attribution is possible by computing a grid of control points assigned to the curved boundaries of the elements. Figure 3 displays a curved surface mesh as it is computed in *Visual DoMesh*.



(a) Surface model. (b) Quad mesh.

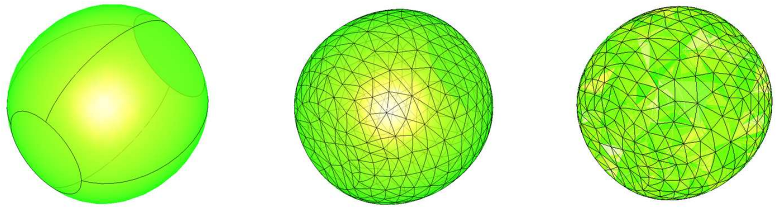
Fig. 3. Geometric model and curved surface mesh.

Tetrahedral meshing

The meshing of BRep volumes to tetrahedral elements introduces a pipeline to the advancing front tetrahedral mesh generator *Netgen* [4]. In a first step the surfaces of the BRep volume (Figure 4(a)) are meshed into triangular elements (Figure 4(b)). This triangular surface mesh is then transferred to *Netgen* through the supported interface and the resulting tetrahedral mesh is then reimported to *MeshGen*(Figure 4(c)).

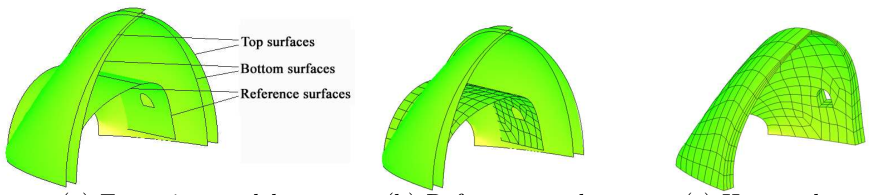
Hexahedral meshing

The generation of hexahedral meshes is performed by extrusion techniques applied on a reference shell [5]. The extrusion path, the thickness and the number of element layers are imported from the geometric model (sec. 2). Figure 5 shows the meshing pipeline for a reference mesh extruded in normal direction and intersected with a top and a bottom shell (three element layers).



(a) BRep model. (b) Surface mesh. (c) Tet mesh.

Fig. 4. Tetrahedral meshing pipeline.



(a) Extrusion model. (b) Reference mesh. (c) Hex mesh.

Fig. 5. Hexahedral meshing pipeline.

4 Conclusion

A software for the generation of mixed high-order finite element meshes is presented. This application provides a number of meshing techniques based on a flexible and powerful program foundation. Future work will focus on extensions of the current meshing algorithms and introduction of new meshing techniques. In addition an integration of *Cubit* is planned.

References

1. B.A. Szabó, A. Düster, and E. Rank. The p-version of the Finite Element Method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, chapter 5, pages 119–139. John Wiley & Sons, 2004.
2. Z. Wassouf. *The mortar method for the finite element method of high order*. PhD thesis, Technische Universität München, 2010.
3. R.E. Bank. PLTMG, A software package for solving elliptic partial differential equations. 7, 1990.
4. J. Schöberl. Netgen - an advancing front 2d/3d-mesh generator based on abstract rules. 1:41–52, 1997.
5. C. Sorger, A. Duester, and E. Rank. Generation of curved high-order hexahedral finite element meshes for thin-walled structures. In *Proceedings of the 11th ISGG Conference*, Montreal, Canada, 2009.