A set of mixed-elements patterns for domain boundary approximation in hexahedral meshes

Article *in* Studies in health technology and informatics · February 2013

DOI: 10.3233/978-1-61499-209-7-268 · Source: PubMed

1 author:

Claudio Lobos

Universidad Técnica Federico Santa María

A Set of Mixed–Elements Patterns for Domain Boundary Approximation in Hexahedral Meshes

Claudio LOBOS [a,1]

[a] *Universidad Técnica Federico Santa María, Departamento de Informática, Chile*

Abstract. Hexahedral meshes are largely used by the Finite Element Method in a high variety of simulation problems. One of the most common problems of these type of meshes is to achieve an adequate approximation of curved domains; a feature typically found in the shape of organs. This work introduces a set of mixed-elements patterns, which are employed at the surface of target domain, and allow to conserve hexahedra elsewhere. These patterns are meant to be combined with any meshing technique producing a regular or non-regular hexahedral mesh.

Keywords. Finite Element Method, Mixed–Elements, Boundary Approximation, Hex–Dominant Meshing

Introduction

Hexahedral meshes are largely used by the Finite Element Method (FEM) in a high variety of simulation problems. Hexahedral meshes might be tangled or present poor quality when representing complex geometry domains. This usually occurs in regions where just a few elements are used to represent: (1) several geometry features or (2) concave regions.

Sometimes it is not mandatory to count with just hexahedra in the mesh. This requirement could be omitted in favor of increasing mesh quality or achieving a better representation of the domain being meshed. For these reasons, this work introduces a set of patterns to replace some hexahedra with other type of elements (wedges, pyramids and/or tetrahedra). This is consistently done following just a few face configurations in terms of inside and outside nodes. Even though the patterns allow to achieve acceptable surface representation of curved domains and ensure proper connectivity among elements, they may still produce poor quality elements. For this reason it is always recommended to use mesh repairing methods [1,2,3] before employing the produced mesh in a simulation context.

[1]Universidad Técnica Federico Santa María, Departamento de Informática, Av. España 1680, Valparaíso, Chile; E-mail: clobos@inf.utfsm.cl.

Methods and Materials

The algorithms that produce an hexahedral mesh take as input the surface of the domain to be meshed. Let *S* be this surface. There are three categories of elements: (1) completely outside *S*, (2) completely inside *S* and (3) **boundary** to *S*, meaning that some nodes of the element are inside and others outside *S*. If the outside nodes of intersecting elements are projected into *S*, two problems can arise: (1) element edge crossing (which is a tangled mesh) and (2) element quality issues due to node proximity. For these reasons, some hexahedra techniques [4,5] propose to remove intersecting elements and then fill the space with an appropriate hexahedra configuration. In the other hand, the patterns to be presented in this section correspond to a different solution, in which intersecting elements are not removed, but replaced when necessary by other type of elements. After all boundary elements are analyzed, remaining outside nodes are projected onto *S*.

The quadrilateral face of an hexahedron intersecting *S* can be split into two triangles using these surface patterns. Therefore, the face subdivision rules of Figure 1 must be respected at all times in order to ensure topological consistency between neighbor elements. Note that these subdivision rules consider all possible configurations of inside and outside nodes in a quadrilateral face and they have also been used in [6,7]. Following these basic rules, Figure 2 summarizes the list of 20 surface patterns, that will be employed to represent the surface of the target domain.

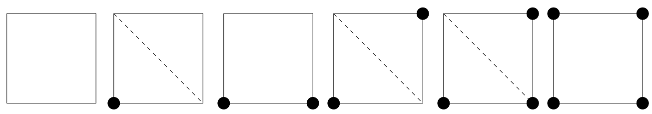

Figure 1. Face consistency patterns. Dashed lines are diagonals to be inserted and circles are inside nodes.

There are several considerations to be made here. The list of patterns is a lookup table which is capable of managing all possible configurations through rotation and mirroring operations. Patterns 4E is the only one that needs a mirroring operation.

Another important consideration is that the presented patterns do not insert additional nodes to the mesh, except for pattern 5C. Due to the configuration of inside nodes that consider this pattern it was impossible to split the initial haxehedron in mixed-elements, and at the same time, be consistent with the face subdivision rules of Figure 1. For this reason, the only solution was to insert the middle point of the hexahedon, build six pyramids considering each quadrilateral face and the mid-point and finally, insert the diagonals following the face subdivision rules.

In order to measure the quality of the resulting meshes two metrics are employed in the Results section of this work. The first is the normalized Jacobian Ratio (JR), which is employed for pyramids, wedges and hexahera. This value is obtained as follows: compute the Jacobian for each node of the element. The JR of the element will be the quotient between the minimum and maximum Jacobian value found within this element. An element is said to present questionable quality when the JR is in the range (0, 0.2) according to some authors [4,5]. The JR is not a good quality metric for a tetrahedron as all of it nodes present the same Jacobian value. For this reason an Aspect Ratio (AR) quality measure [8,9], that also considers the signed volume of the tetrahedron, is used instead.

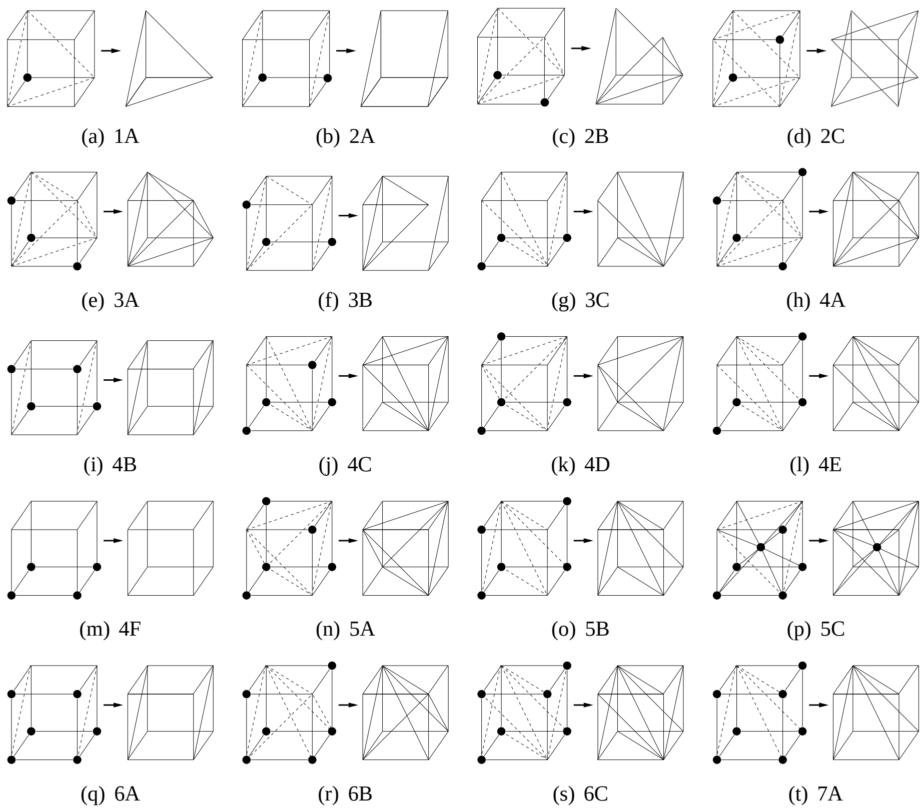

Figure 2. All surface patterns. Circles represent inside nodes.

Results

In order to test the surface patterns introduced in this work, the following algorithm was employed; where the input was a surface triangle mesh *S*: (1) produce an initial grid, (2) push out boundary elements presenting little level of intersection with *S*, (3) apply the surface patterns introduced in this work, (4) project to *S* outside nodes of boundary elements and (5) improve the quality of boundary elements.

After step 1 there will be inside and boundary elements. Step 2 refers to project to *S* **inside** nodes of boundary elements that are close to *S*. By this, boundary elements close to *S* will also be removed in order to avoid "little" size elements at the final mesh. This step also ensures that remaining boundary elements will not be flattened by the following steps of this algorithm. Step 3 will replace all remaining boundary elements with mixed–elements by using the surface patterns introduced in this work. Step 4 will project all the nodes of new boundary elements to *S* and finally, step 5 will improve the quality following the Smart Laplacian filter introduced in [1].

The example of Figure 3 shows a resulting mesh for the lower extremety of a femur. It can be seen the regular alignment of intern elements and the boundary approximation achieved by the use of surface patterns. With respect to quality, all surface elements present acceptable quality ($JR > 0.23$ and $AR > 0.36$) and intern elements are regular hexahedra.

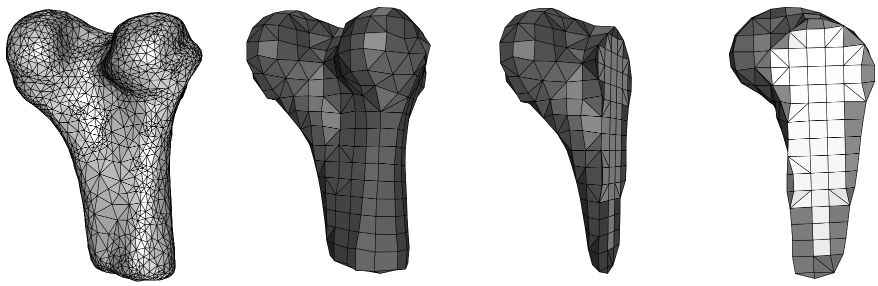

Figure 3. Femur example: (a) input mesh (b) resulting volume mesh, (c) vertical cut and (d) the same cut from another angle.

Conclusions and Discussion

This work proposes a set of mixed–element patterns that can be used to represent curved domains. Moreover, only one of the 20 patterns increases the number of nodes in the final mesh. This is an important achievement because the time needed to compute a simulation can drastically increase as the number of nodes in the mesh increases.

It is important to note that these surface patterns must not be seen as meshing technique. They are rather a complement for hexahedral meshing techniques, like an Octree [10,4], where boundary hexahedra can be replaced with mixed–elements in order to achieve a better representation of the domain or increase element quality at the boundary.

In difference with other mixed–element meshing techniques [11,12], this work only focuses on achieving the representation of the boundary, leaving the internal structure of the mesh without any changes. Unfortunately, the comparison with other mixed–element approaches follows out of the scope of this preliminary work.

One important issue can arise when combining a grid mesh and the patterns. A boundary element could have an edge with both of its nodes labeled as inside the domain. However, the edge could intersect two or more times, faces of the input domain. In this case, the patterns will fail in representing the domain. For this reason, a meshing technique considering this type of cases should be used instead. Once this type of problems are overcomed, the surface patterns can be employed.

Another important issue is that these pattern do not allow to represent domain sharp features as presented in [13]. This issue could be solved by first, detecting the features to represent, then apply the surface patterns and finally, all the outside elements intersecting a feature should be considered to remain in the mesh whenever they do not cause a quality problems.

The presented work succeeded in adding a tool for hexahedral meshes to represent curved domains. In this work, a basic grid mesh was combined with the surface patterns in order to better represent the input domain. Even though some quality issues might occur with more complex domains, these preliminary results encourage further studies with the use of mixed–elements to represent curved domains and, as a future work, the representation of sharp features.

Acknowledgments

This work was partially financed by projects: EcosConicyt C11–E01, Fondecyt de Iniciación 11121601 and DGIP 24.12.33.

References

[1] Freitag L., Plassmann P. Local optimization-based simplicial mesh untangling and improvement. *International Journal for Numerical Methods in Engineering* **49(12)** (2000), 109–125.

[2] Knupp P. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii a framework for volume mesh optimization and the condition number of the jacobian matrix. *International Journal for Numerical Methods in Engineering* **48** (2000), 1165–1185.

[3] Bucki M., Lobos C., Payan Y., Hitschfeld N. Jacobian-based Repair Method for Finite Element Meshes after Registration. *Engineering with Computers* **27(3)** (2011), 285–297.

[4] Ito Y., Shih A., Soni B. Octreebased reasonablequality hexahedral mesh generation using a new set of refinement patterns. *International Journal for Numerical Methods in Engineering* **77(13)** (2009), 1809–1833.

[5] Shepherd J., Johnson C. Hexahedral mesh generation for biomedical models in scirun. *Engineering with Computers* **25** (2009), 97–114.

[6] Schneiders R. Refining quadrilateral and hexahedral element meshes. *In: Proceedings of the Fifth International Conference on Numerical Grid Generation in Computational Field Simulations* (1996), 679–688.

[7] Zhang Y., Bajaj C. Adaptive and quality quadrilateral / hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering* **195(912)** (2006), 942–960.

[8] Ferrant M., Warfield S., Nabavi A., Jolesz F., Kikinis R. Registration of 3d intraoperative mr images of the brain using a finite element biomechanical model. *In: Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention* (2000), 19–28.

[9] Lobos C., Bucki M., Hitschfeld N., Payan Y. Mixed-element mesh for an intra-operative modeling of the brain tumor extraction. *In: Proceedings, 16 th International Meshing Roundtable* (2007), 387–404.

[10] Shephard M., Georges M. Automatic threedimensional mesh generation by the finite Octree technique. *International Journal for Numerical Methods in Engineering* **32** (1991), 709–749.

[11] Meshkat S., Talmor D. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *International Journal for Numerical Methods in Engineering* **49** (2000), 17-30.

[12] Meyers R.J., Tautges T.J., Tuchinsky P.M. The “Hex–tet” hex–dominant meshing algorithm as implemented in CUBIT. *In: Proceedings of 7th international meshing roundtable* (1998), 151–158

[13] Kobbelt L., Botsch M., Schwanecke U., Seidel H. Featuresensitive surface extraction from volume data. *In: ACM SIGGRAPH* (2001), 57–66.