International Conference on Adaptive Modeling and Simulation
ADMOS 2007
P. Díez, K. Runesson

A NEW TECHNIQUE FOR CONSTRUCTING ADAPTIVE 3-D TRIANGULATIONS

J.M. CASCON*, R. MONTENEGRO†, J.M. ESCOBAR†, E. RODRIGUEZ† AND G. MONTERO†

*Department of Mathematics, Faculty of Sciences, University of Salamanca, Spain

†University Institute for Intelligent Systems and Numerical Applications in Engineering
University of Las Palmas de Gran Canaria, Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria, Spain
e-mail: rafa@dma.ulpgc.es - Web page: http://www.dca.iusiani.ulpgc.es/proyecto0507

Key words: Adaptive Mesh Generation, Local Refinement/Derefinement, Nested Meshes, Mesh Smoothing, Mesh Untangling, Tetrahedral Meshes

Abstract. In this paper we present new ideas and applications of an innovative tetrahedral mesh generator which was introduced in [1, 2]. This automatic mesh generation strategy uses no Delaunay triangulation, nor advancing front technique, and it simplifies the geometrical discretization problem in particular cases. The main idea of the new mesh generator is to combine a local refinement/derefinement algorithm for 3-D nested triangulations [3] and a simultaneous untangling and smoothing procedure [4]. 3-D complex domains, which surfaces can be mapped from a *meccano* face to object boundary, are discretized by the mesh generator. Resulting adaptive meshes have an appropriate quality for finite element applications. Finally, an example is showed.

1 INTRODUCTION TO THE *MECCANO* TECHNIQUE

The idea of the new tetrahedral mesh generator starts with the definition of a *meccano* domain as connected cuboids, such that the boundary of the object is obtained by a one-to-one projection (or mapping) from the boundary faces of the *meccano*. Once the *meccano* decomposition into cubes is done, we build an initial coarse tetrahedral mesh by splitting all cubes into six tetrahedra [3]. We continue with a refinement/derefinement strategy to obtain an adapted mesh which can approximate the boundaries of the domain within a given precision. Then, we construct a mesh (usually a tangled mesh) of the domain by projecting boundary nodes from *meccano* plane faces to the object surface and by relocating the *meccano* inner nodes on a reasonable position. Although this node movement does not solve the tangle mesh problem, it normally makes it decrease. Finally, in order to obtain a resulting valid mesh, a mesh optimization procedure (untangling and smoothing) [4] is applied. This process is crucial in the proposed technique and could be combined with a previous smoothing [5] of the object surface triangulation.

At present, this idea has been implemented in ALBERTA code [6, 7]. This software can be used for solving several types of 1-D, 2-D or 3-D problems with adaptive finite elements. The local refinement and derefinement can be done by evaluating an error indicator for each element of the mesh and it is based on element bisection. To be more specific, the newest vertex bisection method is implemented for 2-D triangulations [8]. Actually, ALBERTA has implemented an efficient data structure and adaption for 3-D domains which can be decomposed into hexahedral elements as regular as possible. Each hexahedron is subdivided into six tetrahedra by constructing a main diagonal and its projections on its faces. The local bisection of the resulting tetrahedra is recursively carried out by using ideas of the longest edge [9] and the newest vertex bisection methods. Details about the local refinement technique implemented in ALBERTA for two and three dimensions can be analyzed in [3, 8]. This strategy works very efficiently for initial meshes with a particular topology and high-quality elements (obtained by subdivision of regular quadrilateral or hexahedral elements). In these cases, the degeneration of the resulting 2-D or 3-D triangulations after successive refinements is avoided. The restriction on the initial element shapes and mesh connectivities makes necessary to develop a particular mesh generator for ALBERTA. Obviously, the new meshing technique could be applied for generating meshes with other types of codes. Besides, these ideas could be combined with other type of local refinement/derefinement algorithms for tetrahedral meshes [10, 11].

2 TEST EXAMPLE

We apply our technique to construct a 3-D triangulation of the Earth including a legend related to the present Conference. To define the topography we have used GTOPO30 (*http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html*). GTOPO30 is a global digital elevation model (DEM) with a horizontal grid spacing of 30 arc seconds (approximately 1 kilometer).

In this case, the *meccano* is composed by one cube and we use a radial projection to generate the topography of the surface of the Earth from GTOPO30. The external faces of the *meccano* are projected on the surface of the Earth, see Figure 1. In this example, we have constructed an automatic adaptive discretization to well approach the coastlines and the boundaries of the legend.

The cube is first split into six tetrahedra forming a coarse initial mesh. Then, we apply 21 recursive bisections on all tetrahedra which have a face placed on the *meccano* boundary and 6 additional bisections on all tetrahedra with a face on the legend window. This fine mesh contains 956901 nodes and 4026768 tetrahedra. In order to obtain a final mesh with a good representation of coastlines and the boundaries of the letters, we have defined the following derefinement condition: a node P could be eliminated if node P and all nodes connected with P are of the same type (sea, land or letter). Perhaps node P can not be finally removed due to conformity reasons. On the other hand, nodes belonging to the coarse initial mesh are not removed from the sequence of nested meshes. The resulting mesh has 124949 nodes and 534246 tetrahedra; see Figure 1.

Figure 1: Resulting adaptive tetrahedral mesh of the Earth after applying the refinement/derefinement procedure on the *meccano* (one cube) and radial node projection from *meccano* faces to Earth surface

The relocation of inner nodes is enough to obtain a untangled mesh, however its quality is very poor. After 19 iterations, the optimization procedure improves the value of the minimum quality to 0.264 and its average to $\overline{q}_\kappa = 0.77$.

The CPU time for constructing the adaptive *meccano* mesh is 77.5 seconds, for the projection/relocation process is 2.1 seconds and for the smoothing procedure is 1129.7 seconds. This experiment has been done on a Dell Precision 960, with two Intel Xeon double kernel processor, 3.2 GHz, 64 bits and 8 Gb RAM, on a Red Hat Enterprise Linux WS v.4 system and using the compiler gcc v.3.4.6. Note that the mesh optimization dominates the mesh generation procedure.

Acknowledgments

This work has been supported by the Spanish Government and FEDER, grant contracts: CGL2004-06171-C03-03/CLI and CGL2004-06171-C03-02/CLI. Besides, we thanks to the authors of ALBERTA [7] for the code availability in internet [6] and for their suggestions.

REFERENCES

[1] R. Montenegro, J.M. Cascón, J.M. Escobar, E. Rodríguez and G. Montero. Implementation in ALBERTA of an automatic tetrahedral mesh generator. *Proc. 15th Int. Meshing Roundtable*, 325–338, Springer, Berlin, (2006).

[2] J.M. Cascón, R. Montenegro, J.M. Escobar, E. Rodríguez and G. Montero. A new *meccano* technique for adaptive 3-D triangulations. To appear in *Proc. 16th Int. Meshing Roundtable*, Springer, Berlin, (2007).

[3] I. Kossaczky. A recursive approach to local mesh refinement in two and three dimensions. *J. Comput. Appl. Math.*, **55**, 275–288, (1994).

[4] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero and J.M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Comput. Meth. Appl. Mech. Eng.*, **192**, 2775–2787, (2003).

[5] J.M. Escobar, G. Montero, R. Montenegro and E. Rodríguez. An algebraic method for smoothing surface triangulations on a local parametric space. *Int. J. Num. Meth. Eng.* **66**, 740–760, (2006).

[6] ALBERTA - An Adaptive Hierarchical Finite Element Toolbox, *http://www.alberta-fem.de/.*

[7] A. Schmidt and K.G. Siebert. Design of adaptive finite element software: the finite element toolbox ALBERTA. *Lecture Notes in Computer Science and Engineering*, **42**, Springer, Berlin, (2005).

[8] W.F. Mitchell. A comparison of adaptive refinement techniques for elliptic problems. *ACM Trans. Math. Soft.*, **15**, 326–347, (1989).

[9] M.C. Rivara. A grid generator based on 4-triangles conforming. Mesh-refinement algorithms. *Int. J. Num. Meth. Eng.*, **24**, 1343–1354, (1987).

[10] J.M. González-Yuste, R. Montenegro, J.M. Escobar, G. Montero and E. Rodríguez. Local refinement of 3-D triangulations using object-oriented methods. *Adv. Eng. Soft.*, **35**, 693–702, (2004).

[11] R. Löhner and J.D. Baum. Adaptive h-refinement on 3-D unstructured grids for transient problems. *Int. J. Num. Meth. Fluids*, **14**, 1407–1419, (1992).