THE MECCANO METHOD FOR SOLID PARAMETRIZATION AND MESHING

R. MONTENEGRO[1], J.M. CASCÓN[2], E. RODRÍGUEZ[1], AND J.M. ESCOBAR[1]

[1]Institute for Intelligent Systems and Numerical Applications in Engineering (SIANI),
University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain.
[2]Department of Economics and Economic History, Faculty of Economics and
Management, University of Salamanca, Spain.

ABSTRACT. We review the novel meccano method for constructing adaptive tetrahedral meshes of 3-D complex solids and their volume parametrization. Specifically, we will consider a solid whose boundary is a surface of genus zero. In this particular case, the automatic procedure is defined by a surface triangulation of the solid, a simple meccano composed by one cube and a tolerance that fixes the desired approximation of the solid surface. The procedure is based on an automatic mapping from the cube faces to the solid surface, a 3-D local refinement algorithm and our simultaneous mesh untangling and smoothing procedure. The meccano method constructs high quality surface and volume adaptive meshes. Several examples show the efficiency of the proposed technique. In addition, we comment an application of the method to construct volume T-meshes for isogeometric analysis.

AMS (MOS) Subject Classification. 65M50, 65N50.

1. INTRODUCTION

We introduced the new meccano method in (Montenegro et al., 2006; Cascón et al., 2007; Montenegro et al., 2009; Cascón et al., 2009) for constructing adaptive tetrahedral meshes of solids. We have given this name to the method because the process starts with a coarse approximation of the solid, i.e. a meccano composed by connected polyhedral pieces. The method builds a 3-D triangulation of the solid as a deformation of an appropriate tetrahedral mesh of the meccano. A particular case is when meccano is a polycube.

The new automatic mesh generation strategy uses no Delaunay triangulation, nor advancing front technique, and it simplifies the geometrical discretization problem for 3-D complex domains, whose surfaces can be mapped to the meccano faces. The main idea of the meccano method is to combine a local refinement/derefinement algorithm for 3-D nested triangulations (Kossaczky, 1994), a parameterization of surface triangulations (Floater, 1997) and a simultaneous untangling and smoothing procedure (Escobar et al., 2003).

Mesh optimization is guided by the minimization of certain overall functions and it is usually done in a local fashion. In general, this procedure involves two steps (Freitag & Plassmann, 2000; Freitag & Knupp, 2002): the first is for mesh untangling and the second one for mesh smoothing. Each step leads to a different objective function. In this paper, we use the improvement proposed by (Escobar et al., 2003), where a simultaneous untangling and smoothing guided by the same objective function is introduced.

Some advantages of the meccano technique are that: surface triangulation is automatically constructed, the final 3-D triangulation is conforming with the object boundary, inner surfaces are automatically preserved (for example, interface between several materials), node

Received April 15, 2010

distribution is adapted in accordance with the object geometry, and parallel computations can easily be developed for meshing the meccano pieces. However, our procedure demands an automatic construction of the meccano and an admissible mapping between the meccano boundary and the object surface must be defined.

In this paper, we consider a complex genus-zero solid, i.e. a solid whose boundary is a surface that is homeomorphic to the surface of a sphere, and we assume that the solid geometry is defined by a triangulation of its surface. In this case, it is sufficient to fix a meccano composed by a single cube and a tolerance that fixes the desired approximation of the solid surface. In order to define an admissible mapping between the cube faces and patches of the initial surface triangulation of the solid, we have introduced a new automatic method to decompose the surface triangulation into six patches that preserves the same topological connections than the cube faces. Then, a discrete mapping from each surface patch to the corresponding cube face is constructed by using the parameterization of surface triangulations proposed by M. Floater in (Floater, 1997; Floater & Hormann, 2005). The shape-preserving parametrizations, which are planar triangulations on the cube faces, are the solutions of linear systems based on convex combinations.

In the near future, more effort should be made in developing an automatic construction of the meccano when the genus of the solid surface is greater than zero. Currently, several authors are working on this aspect in the context of polycube-maps, see for example (Lin et al., 2008; Wang et al., 2008). They are analyzing how to construct a polycube for a generic solid and, simultaneously, how to define a conformal mapping between the polycube boundary and the solid surface. Although harmonic maps have been extensively studied in the literature of surface parameterization, only a few works are related to volume parametrization, for example a procedure is presented in see (Li et al., 2007).

In the following Section we present a brief description of the main stages of the method for a generic meccano composed of polyhedral pieces. In Section 3 we introduce applications of the algorithm in the case that the meccano is formed by a simple cube. Finally, conclusions and future research are presented in Section 4.

2. THE MECCANO METHOD

The main steps of the general *meccano tetrahedral mesh generation algorithm* are summarized in this section. The input data are the definition of the solid boundary (for example by a given surface triangulation) and a given tolerance (corresponding to the solid surface approximation). The following algorithm describes the whole mesh generation approach:

1. Construct a meccano approximation of the 3-D solid formed by polyhedral pieces.
2. Define an admissible mapping between the meccano boundary faces and the solid boundary.
3. Build a coarse tetrahedral mesh of the meccano.
4. Generate a local refined tetrahedral mesh of the meccano, such that the mapping of the meccano boundary triangulation approximates the solid boundary for a given precision.
5. Move the boundary nodes of the meccano to the object surface with the mapping defined in 2.
6. Relocate the inner nodes of the meccano.
7. Optimize the tetrahedral mesh with the simultaneous untangling and smoothing procedure.

The first step of the procedure is to construct a meccano approximation by connecting different polyhedral pieces. Once the meccano approximation is fixed, we have to define an *admissible* one-to-one mapping between the boundary faces of the meccano and the boundary of the object. In step 3, the meccano is decomposed into a coarse and valid tetrahedral mesh by an appropriate subdivision of its initial polyhedral pieces. We continue with a local refinement strategy to obtain an adapted mesh which can approximate the boundaries of the domain within a given precision. Then, we construct a mesh of the solid by mapping the boundary nodes from the meccano faces to the true solid surface and by relocating the inner nodes at a reasonable position. After those two steps the resulting mesh is tangled, but it has an admissible topology. Finally, a simultaneous untangling and smoothing procedure is applied and a valid adaptive tetrahedral mesh of the object is obtained.

We note that the general idea of the meccano technique could be understood as the connection of different polyhedral pieces. So, the use of cuboid pieces, or a polycube meccano, are particular cases.

3. **APPLICATIONS**

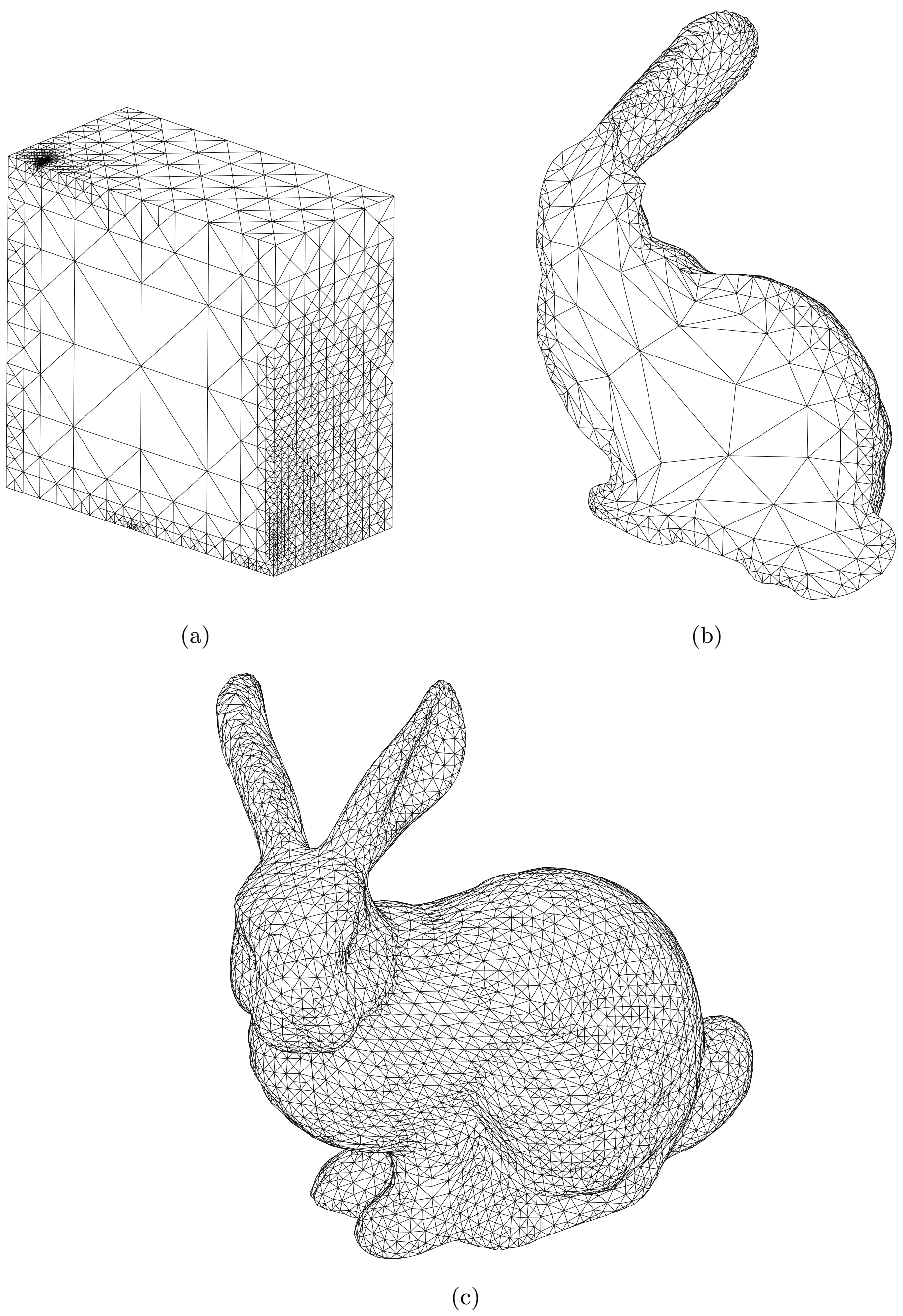
We introduce an automatic parametrization between the surface triangulation of the solid and the cube boundary. To that end, we automatically divide the surface triangulation into six patches, with the same topological connection that cube faces, so that each patch is mapped to a cube face. These parametrizations have been done with GoTools core and modules from SINTEF ICT, available in the website http://www.sintef.no/math_software. This code implements Floater's parametrization in C++ (Floater & Hormann, 2005).

We have implemented the meccano method by using the local refinement of ALBERTA. This code is an adaptive multilevel finite element toolbox (Schmidt & Siebert, 2005) developed in C. This software uses the Kossaczky refinement algorithm (Kossaczky, 1994) and requires an initial mesh topology.

The performance of our novel tetrahedral mesh generator is shown in the following application to the Stanford Bunny. The original surface triangulation of the Bunny has been obtained from the website *http://graphics.stanford.edu/data/3Dscanrep/* , i.e. the Stanford Computer Graphics Laboratory. It has 12654 triangles and 7502 nodes. We consider a unit cube as meccano. The method generates a cube tetrahedral mesh with 54496 tetrahedra and 13015 nodes, see Figure 1(a). This mesh has 11530 triangles and 6329 nodes on its boundary.

The mapping of the cube external nodes to the Bunny surface produces a 3-D tangled mesh with 2384 inverted elements. The relocation of inner nodes by using volume parametrizations reduces the number of inverted tetrahedra to 42. We apply 8 iterations of the tetrahedral mesh optimization and only one inverted tetrahedron can not be untangled. To solve this problem, we allow the movement of the external nodes of this inverted tetrahedron and we apply 8 new optimization iterations. The mesh is then untangled and, finally, we apply 8 smoothing iterations fixing the boundary nodes. The resulting mesh quality is improved to a minimum value of 0.08 and an average $\overline{q}_\kappa = 0.68$, see Figures 1(b) and 1(c). We note that the meccano technique generates a high quality tetrahedra mesh: only 1 tetrahedron has a quality below 0.1, 41 below 0.2 and 391 below 0.3. The CPU time for constructing the final mesh of the Bunny is 40.28 seconds on a Dell precision 690, 2 Dual Core Xeon processor and 8 *Gb* RAM memory.

The volume parametrization presented in this paper has applications in other fields different from tetrahedral mesh generation. For example, it can be used to construct a volume T-mesh for isogeometric analysis (Bazilevs et al., 2010). The key lies in using the mapping, provided by the volume parametrization, to transform a T-mesh defined on the parametric domain (a unit cube in our case) into the physical domain. The T-mesh of the parametric domain is the parametric space in which the set of T-splines are defined.



(a)

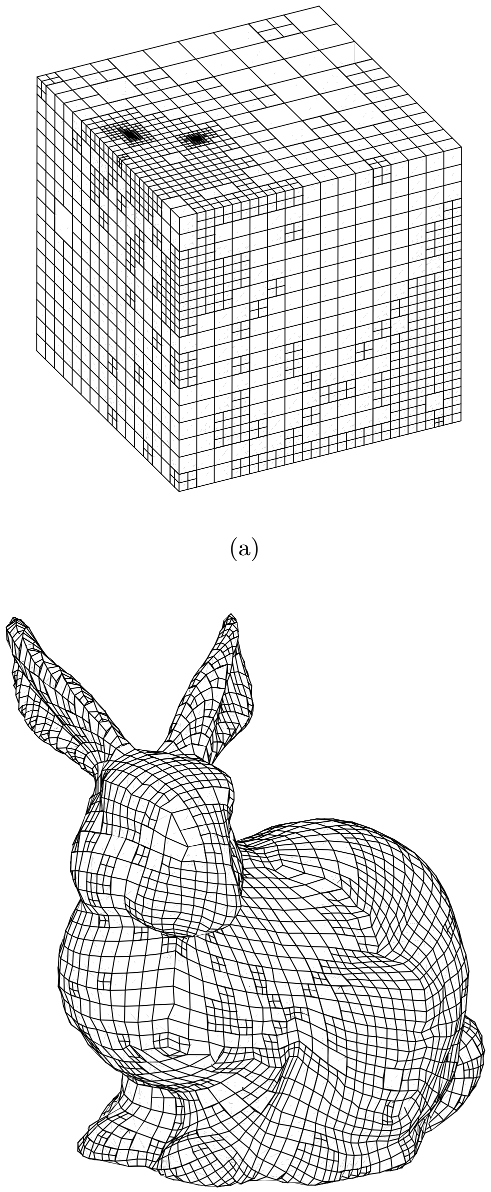
(b)

(c)

Figure 1. Cross sections of the cube (a) and the Bunny tetrahedral mesh (b). Resulting tetrahedral mesh of the Bunny obtained by the meccano method (c).

The technique to construct a T-mesh starts by dividing the parametric cube in lower cubes by using an octree in such a way that each leaf of the octree is divided in eight children (eight cubes). The division continues until each terminal cube of the octree does not contain a node of the Kossaczky's mesh in its inner. The octree defines a T-mesh in the parametric space that it is used to determine the *local knot vector* and the *anchors* of the T-splines following the description of (Bazilevs et al., 2010).

Figure 2(a) shows the parametric volume T-mesh and Figure 2(b) shows the corresponding T-mesh of the Standford bunny.



(a)

(b)

Figure 2. Parametric volume T-mesh (a) and corresponding physical T-mesh (b)

4. CONCLUSIONS

The meccano technique is an efficient adaptive tetrahedral mesh generator for solids whose boundary is a surface of genus zero. We remark that the method requires minimum user intervention and has a low computational cost. The procedure is fully automatic and it is only defined by a surface triangulation of the solid, a cube and a tolerance that fixes the desired approximation of the solid surface. A crucial consequence of the new mesh generation technique is the resulting discrete parametrization of a complex volume (solid) to a simple cube (meccano). On this direction, we have introduced an application of the meccano method in isogeometric analysis.

The main ideas presented in this paper can be applied for constructing tetrahedral or hexahedral meshes of complex solids. In future works, the meccano technique can be extended for meshing a complex solid whose boundary is a surface of genus greater than zero. In this case, the meccano can be a polycube or constructed by polyhedral pieces.

ACKNOWLEDGEMENT

This work has been supported by the Spanish Government, "Secretaría de Estado de Universidades e Investigación", "Ministerio de Ciencia e Innovación", and FEDER, grant contracts: CGL2008-06003-C03 and UNLP08-3E-010. The authors are also grateful to the "Junta de Castilla y León", grant number SA124A08, and the "Instituto Tecnológico de Canarias" for their support. See results on http://www.dca.iusiani.ulpgc.es/proyecto2008-2011.

REFERENCES

[1] Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J.A., Hughes, T.J.R., Lipton, S., Scott, M.A., & T.W. Sederberg (2010). Isogeometric analysis using T-splines. Comput. Meth. Appl. Mech. Eng., v. 199, pp. 229–263.

[2] Cascón, J.M., Montenegro, R., Escobar, J.M., Rodríguez, E., & Montero, G. (2007). A New Meccano Technique for Adaptive 3-D Triangulations. Proc. 16th Int. Meshing Roundtable, pp. 103–120. New York: Springer-Verlag.

[3] Cascón, J.M., Montenegro, R., Escobar, J.M., Rodríguez, E., & Montero, G. (2009). The Meccano Method for Automatic Tetrahedral Mesh Generation of Complex Genus-Zero Solids. Proc. 18th Int. Meshing Roundtable, pp. 463–480. New York: Springer-Verlag.

[4] Escobar, J.M., Rodríguez, E., Montenegro, R., Montero, G., & González-Yuste, J.M. (2003). Simultaneous Untangling and Smoothing of Tetrahedral Meshes. Comput. Meth. Appl. Mech. Eng., v. 192, pp.2775–2787.

[5] Floater, M.S. (1997). Parametrization and Smooth Approximation of Surface Triangulations. Comput. Aid. Geom. Design, v. 14, pp. 231–250.

[6] Floater, M.S., & Hormann, K. (2005). Surface Parameterization: A Tutorial and Survey. Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization, pp. 157–186. Berlin: Springer.

[7] Freitag, L.A., & Knupp, P.M. (2002). Tetrahedral Mesh Improvement Via Optimization of the Element Condition Number. Int. J. Num. Meth. Eng., v. 53, pp. 1377–1391.

[8] Freitag, L.A., & Plassmann, P. (2000). Local Optimization-Based Simplicial Mesh Untangling and Improvement. Int. J. Num. Meth. Eng., v. 49, pp. 109–125.

[9] Kossaczky, I. (1994). A Recursive Approach to Local Mesh Refinement in Two and Three Dimensions. J. Comput. Appl. Math., v. 55, pp. 275–288.

[10] Li, X., Guo, X., Wang, H., He, Y., Gu, X., & Qin, H. (2007). Harmonic Volumetric Mapping for Solid Modeling Applications. Proc. of ACM Solid and Physical Modeling Symposium, pp. 109–120. Association for Computing Machinery, Inc.

[11] Lin, J., Jin, X., Fan, Z., & Wang, C.C.L. (2008). Automatic PolyCube-Maps. Lecture Notes in Computer Science, v. 4975, pp. 3–16.

[12] Montenegro, R., Cascón, J.M., Escobar, J.M., Rodríguez, E., & Montero, G. (2006). Implementation in ALBERTA of an Automatic Tetrahedral Mesh Generator. Proc. 15th Int. Meshing Roundtable, pp. 325–338. New York: Springer-Verlag.

[13] Montenegro, R., Cascón, J.M., Escobar, J.M., Rodríguez, E., & Montero, G. (2009). An Automatic Strategy for Adaptive Tetrahedral Mesh Generation. Appl. Num. Math., v. 59, pp. 2203–2217.

[14] Schmidt, A., & Siebert, K.G. (2005). Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA. Lecture Notes in Computer Science and Engineering, v. 42. Berlin: Springer.

[15] Wang, H., He, Y., Li, X., Gu, X., & Qin, H. (2008). Polycube Splines. Comput. Aid. Geom. Design, v. 40, pp. 721–733.