Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II
WCECS 2009, October 20-22, 2009, San Francisco, USA

The Automatic Meccano Method to Mesh Complex Solids

R. Montenegro, J.M. Cascón, J.M. Escobar, E. Rodríguez, G. Montero *

Abstract—**In this paper, we present significant advances of the novel meccano technique to construct adaptive tetrahedral meshes of 3-D complex solids. Specifically, we will consider a solid whose boundary is a surface of genus 0, i.e. a surface that is homeomorphic to the surface of a sphere. 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 main idea is based on an automatic mapping from the cube faces to the solid surface, a 3-D local refinement algorithm and a simultaneous mesh untangling and smoothing procedure. Although the initial surface triangulation can be a poor quality mesh, the meccano technique constructs high quality surface and volume adaptive meshes. A crucial consequence of the new mesh generation technique is the resulting discrete parametrization of a complex volume (solid) to a simple cube (meccano). Several examples show the efficiency of the proposed technique. Future possibilities of the meccano method for meshing a complex solid, whose boundary is a surface of genus greater than zero, are commented.**

Keywords: tetrahedral mesh generation, adaptive refinement, nested meshes, mesh untangling and smoothing, surface and volume parametrization

1 Introduction

Many authors have devoted great effort to solving the automatic mesh generation problem in different ways [3, 14, 15, 27], but the 3-D problem is still open [1]. Along the past, the main objective has been to achieve high quality adaptive meshes of complex solids with minimal user intervention and low computational cost. At present, it is well known that most mesh generators are based on Delaunay triangulation and advancing front technique, but problems, related to mesh quality or mesh conformity with the solid boundary, can still appear for complex geometries. In addition, an appropriate definition of element sizes is demanded for obtaining good quality elements and mesh adaption. Particularly, local adaptive refinement strategies have been employed to mainly adapt the mesh to singularities of numerical solution. These adaptive methods usually involve remeshing or nested refinement.

We introduced the new meccano technique in [22, 2, 23] for constructing adaptive tetrahedral meshes of solids. We have given this name to the method because the process starts with the construction of 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 composed by connected cubes, i.e. 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 [19], a parameterization of surface triangulations [7] and a simultaneous untangling and smoothing procedure [4]. At present, the meccano technique has been implemented by using the local refinement/derefinement of Kossaczky [19], but the idea could be implemented with other types of local refinement algorithms [16]. The resulting adaptive tetrahedral meshes with the meccano method have good quality for finite element applications.

Our approach is based on the combination of several former procedures (refinement, mapping, untangling and smoothing) which are not in themselves new, but the overall integration is an original contribution. Many authors have used them in different ways. Triangulations for convex domains can be constructed from a coarse mesh by using refinement/projection [24]. Adaptive nested meshes have been constructed with refinement and derefinement algorithms for evolution problems [6]. Mappings between physical and parametric spaces have been ana-

*This work has been partially supported by the Spanish Government, “Secretaría de Estado de Universidades e Investigación”, “Ministerio de Ciencia e Innovación”, and FEDER, grant contract: CGL2008-06003-C03. Preprint submitted to the ICMSC 2009 (WCECS 2009) June 7, 2009. Institute for Intelligent Systems and Numerical Applications in Engineering (SIANI), University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain, Email: {rmontenegro,jmescobar,erodriguez,gmontero}@siani.es, http://www.dca.iusiani.ulpgc.es/proyecto2008-2011, and Department of Mathematics, Faculty of Sciences, University of Salamanca, 37008 Salamanca, Spain, E-mail: casbar@usal.es

lyzed by several authors. Significant advances in surface parametrization have been done in [7, 9, 10, 26, 21, 28], but the volume parametrization is still open. Floater et al [11] give a simple counterexample to show that convex combination mappings over tetrahedral meshes are not necessarily one-to-one. Large domain deformations can lead to severe mesh distortions, especially in 3-D. Mesh optimization is thus key for keeping mesh shape regularity and for avoiding a costly remeshing [17, 18]. In traditional mesh optimization, mesh moving is guided by the minimization of certain overall functions, but it is usually done in a local fashion. In general, this procedure involves two steps [13, 12]: 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 [4, 5], 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 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 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 introduce 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 [7, 8, 9, 10]. 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 [26, 21, 28]. 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 [20].

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. A detailed description of this technique can be analyzed in [22, 2, 23]. 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.

Meccano tetrahedral mesh generation algorithm

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 Application of the Meccano Method to Complex Genus-Zero Solids

In this section, we present the application of the meccano algorithm in the case of the solid surface being genus-zero and the meccano being formed by a single cube. We assume as datum a triangulation of the solid surface as data.

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 parametrization modules from SINTEF ICT, available in the website http://www.sintef.no/math_software. This code implements Floater's parametrization in C++. Specifically, in the following application we have used the mean value method for the parametrization of the inner nodes of the patch triangulation, and the boundary nodes are fixed with chord length parametrization [7, 9].

We have implemented the meccano method by using the local refinement of ALBERTA. This code is an adaptive multilevel finite element toolbox [25] developed in C. This software can be used to solve several types of 1-D, 2-D or 3-D problems. ALBERTA uses the Kossaczky refinement algorithm [19] and requires an initial mesh topology [24]. The recursive refinement algorithm could not terminate for general meshes. The meccano technique constructs meshes that verify the imposed restrictions of ALBERTA in relation to topology and structure. The minimum quality of refined meshes is function of the initial mesh quality.

The performance of our novel tetrahedral mesh generator is shown in the following applications. The first corresponds to a Bust, the second to the Stanford Bunny and the third to a Bone. We have obtained a surface triangulation of these objects from internet.

Example 1: Bust

The original surface triangulation of the Bust has been obtained from the website *http://shapes.aimatshape.net*, i.e. AIM@SHAPE Shape Repository. It has 64000 triangles and 32002 nodes. The bounding box of the solid is defined by the points $(x, y, z)_{min} = (-120, -30.5, -44)$ and $(x, y, z)_{max} = (106, 50, 46)$.

We consider a cube, with an edge length equal to 20, as meccano. Its center is placed inside the solid at the point $(5, -3, 4)$. We obtain an initial subdivision of Bust surface in seven maximal connected subtriangulations by using the Voronoi diagram associated to the centers of the cube faces. In order to get a compatible decomposition of the surface triangulation, we apply an iterative procedure to reduce the current seven patches to six.

We map each surface patch Σ_S^i to the cube face Σ_C^i by using the Floater parametrization [7]. The definition of the one-to-one mapping between the cube and Bust boundaries is straightforward once the global parametrization of the Bust surface triangulation is built.

Fixing a tolerance $\varepsilon_2 = 0.1$, the meccano method generates a tetrahedral mesh of the cube with 147352 tetrahedra and 34524 nodes, see a cross section of the cube mesh in Figure 1(a). This mesh has 32254 triangles and 16129 nodes on its boundary and it has been reached after 42 Kossaczky refinements from the initial subdivision of the cube into six tetrahedra. The mapping of the cube external nodes to the Bust surface produces a 3-D tangled mesh with 8947 inverted elements, see Figure 1(b). The location of the cube is shown in this Figure. The relocation of inner nodes by using volume parametrizations reduces the number of inverted tetrahedra to 285. We apply our mesh optimization procedure [4] and the mesh is untangled in 2 iterations. The mesh quality is improved to a minimum value of 0.07 and an average $\bar{q}_\kappa = 0.73$ after 10 smoothing iterations. We note that the meccano technique generates a high quality tetrahedra mesh (see Figures 1(c) and 1(d)): only 1 tetrahedron has a quality lower than 0.1, 13 lower than 0.2 and 405 lower than 0.3.

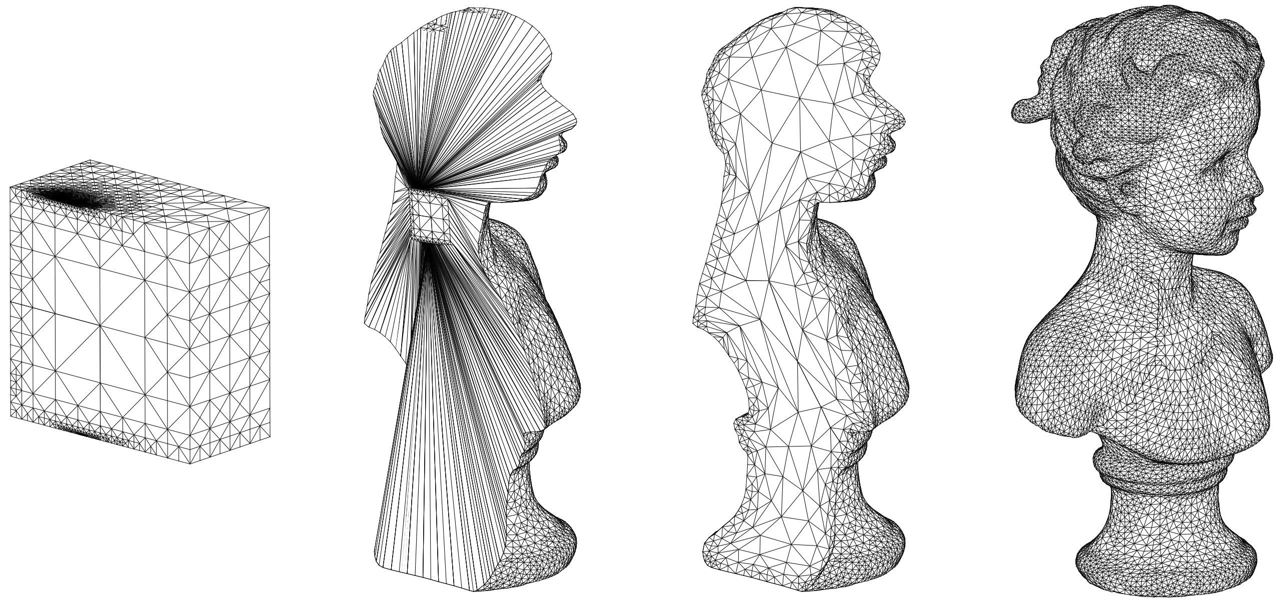
The CPU time for constructing the final mesh of the Bust is 93.27 seconds on a Dell precision 690, 2 Dual Core Xeon processor and 8 *Gb* RAM memory. More precisely, the CPU time of each step of the meccano algorithm is: 1.83 seconds for the subdivision of the initial surface triangulation into six patches, 3.03 seconds for the Floater parametrization, 44.50 seconds for the Kossaczky recursive bisections, 2.31 seconds for the external node mapping and inner node relocation, and 41.60 seconds for the mesh optimization.

Example 2: Bunny

The original surface triangulation of the Stanford 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. The bounding box of the solid is defined by the points $(x, y, z)_{min} = (-10, 3.5, -6)$ and $(x, y, z)_{max} = (6, 2, 6)$.

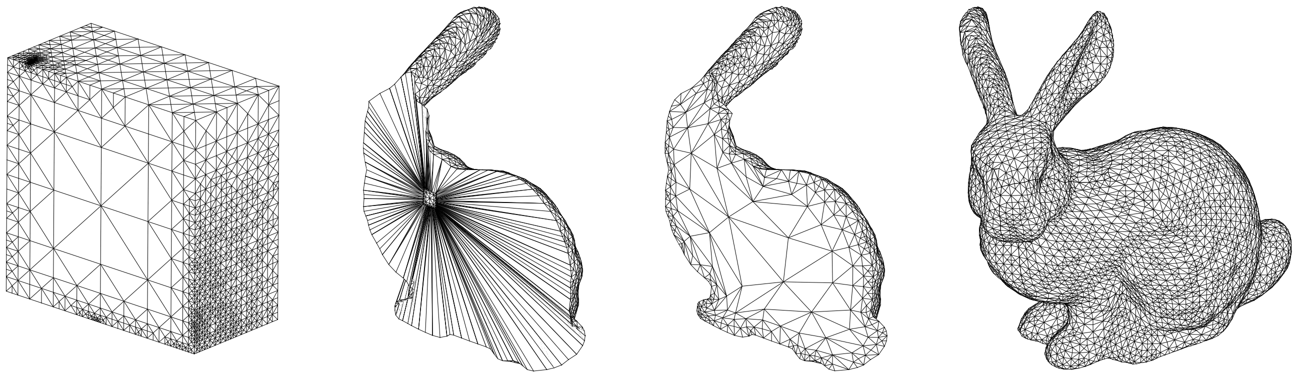
We consider a unit cube as meccano. Its center is placed inside the solid at the point $(-4.5, 10.5, 0.5)$. We obtain an initial subdivision of the Bunny surface in eight maximal connected subtriangulations using Voronoi diagram.

ISBN:978-988-18210-2-7



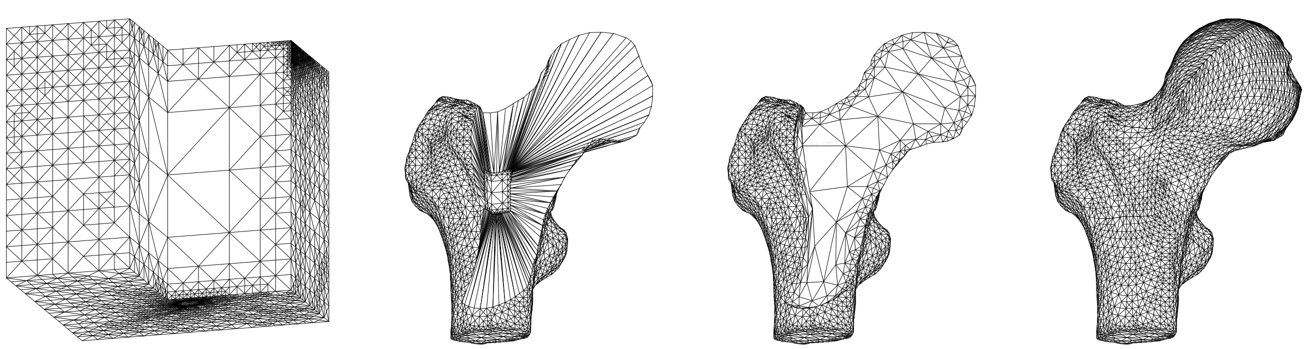
(a) (b) (c) (d)

Figure 1: Cross sections of the cube (a) and the Bust tetrahedral mesh before (b) and after (c) the application of the mesh optimization procedure. (d) Resulting tetrahedral mesh of the Bust obtained by the meccano method.



(a) (b) (c) (d)

Figure 2: Cross sections of the cube (a) and the Bunny tetrahedral mesh before (b) and after (c) the application of the mesh optimization procedure. (d) Resulting tetrahedral mesh of the Bunny obtained by the meccano method.



(a) (b) (c) (d)

Figure 3: Cross sections of the cube (a) and the Bone tetrahedral mesh before (b) and after (c) the application of the mesh optimization procedure. (d) Resulting tetrahedral mesh of the Bone obtained by the meccano method.

We reduce the surface partition to six patches and we construct the Floater parametrization from each surface patch Σ_S^i to the corresponding cube face Σ_C^i. Fixing a tolerance $\varepsilon_2 = 0.0005$, the meccano method generates a cube tetrahedral mesh with 54496 tetrahedra and 13015 nodes, see Figure 2(a). This mesh has 11530 triangles and 6329 nodes on its boundary and has been reached after 44 Kossaczky refinements from the initial subdivision of the cube into six tetrahedra.

The mapping of the cube external nodes to the Bunny surface produces a 3-D tangled mesh with 2384 inverted elements, see Figure 2(b). 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 2(c) and 2(d). 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. More precisely, the CPU time of each step of the meccano algorithm is: 0.24 seconds for the subdivision of the initial surface triangulation into six patches, 0.37 seconds for the Floater parametrization, 8.62 seconds for the Kossaczky recursive bisections, 0.70 seconds for the external node mapping and inner node relocation, and 30.35 seconds for the mesh optimization.

Example 3: Bone

The original surface triangulation of the Bone has been obtained from *http://www-c.inria.fr/gamma/download/-affichage.php?dir=ANATOMY&name=ballJoint*, and it can be found in the CYBERWARE Catalogue. This surface mesh contains 274120 triangles and 137062 nodes.

Steps of the meccano technique are shown in Figure 3. The resulting mesh has 47824 tetrahedra and 11525 nodes. This mesh has 11530 triangles and 5767 nodes on its boundary and it has been reached after 23 Kossaczky refinements from the initial subdivision of the cube into six tetrahedra. A tangled tetrahedra mesh with 1307 inverted elements appears after the mapping of the cube external nodes to the bone surface. The node relocation process reduces the number of inverted tetrahedra to 16. Finally, our mesh optimization algorithm produces a high quality tetrahedra mesh: the minimum mesh quality is 0.15 and the average quality is 0.64.

4 Conclusions and Future Research

The meccano technique is a very efficient adaptive tetrahedral mesh generator for solids whose boundary is a surface of genus 0. 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.

We have introduced an automatic partition of the given solid surface triangulation for fixing an admissible mapping between the cube faces and the solid surface patches, such that each cube face is the parametric space of its corresponding patch.

The mesh generation technique is based on sub-processes (subdivision, mapping, optimization) which are not in themselves new, but the overall integration using a simple shape as starting point is an original contribution of the method and has some obvious performance advantages. Another interesting property of the new mesh generation strategy is that it automatically achieves a good mesh adaption to the geometrical characteristics of the domain. In addition, the quality of the resulting meshes is high.

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 with compatible connections. At present, the user has to define the meccano associated to the solid, but we are implementing a special CAD package for more general input solid.

References

[1] Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J., Hughes, T.J.R., Lipton, S., Scott, M.A., Sederberg, T.W., "Isogeometric Analysis: Toward Unification of Computer Aided Design and Finite Element Analysis," *Trends in Engineering Computational Technology*, pp. 1-16. Saxe-Coburg Publications, Stirling, 2008.

[2] Cascón, J.M., Montenegro, R., Escobar, J.M., Rodríguez, E., Montero, G., "A New *Meccano* Technique for Adaptive 3-D Triangulations," *Proc 16th Int Meshing Roundtable*, Seattle, USA, pp. 103-120, 10/07

[3] Carey, G.F., *Computational Grids: Generation, Adaptation, and Solution Strategies*, Taylor & Francis, Washington, 1997.

[4] Escobar, J.M., Rodríguez, E., Montenegro, R., Montero, G., González-Yuste, J.M., "Simultaneous Un-

tangling and Smoothing of Tetrahedral Meshes," *Comput Meth Appl Mech Eng*, V192, pp. 2775-2787, 2003.

[5] Escobar, J.M., Montero, G., Montenegro, R., Rodríguez, E., "An Algebraic Method for Smoothing Surface Triangulations on a Local Parametric Space," *Int J Num Meth Eng*, V66, pp. 740-760, 2006.

[6] Ferragut, L., Montenegro, R., Plaza, A., "Efficient Refinement/Derefinement Algorithm of Nested Meshes to Solve Evolution Problems," *Comm Num Meth Eng*, V10, pp. 403-412, 1994.

[7] Floater, M.S., "Parametrization and Smooth Approximation of Surface Triangulations," *Comput Aid Geom Design*, V14, pp. 231-250, 1997.

[8] Floater, M.S., "One-to-one Piece Linear Mappings over Triangulations," *Mathematics of Computation*, V72, pp. 85-696, 2002.

[9] Floater, M.S., "Mean Value Coordinates," *Comput Aid Geom Design*, V20, pp. 19-27, 2003.

[10] Floater, M.S., Hormann, K., "Surface Parameterization: A Tutorial and Survey," *Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization*, pp. 157-186. Springer, Berlin, 2005.

[11] Floater, M.S., Pham-Trong, V., "Convex Combination Maps over Triangulations, Tilings, and Tetrahedral Meshes," *Advances in Computational Mathematics*, V25, pp. 347-356, 2006.

[12] Freitag, L.A., Knupp, P.M., "Tetrahedral Mesh Improvement Via Optimization of the Element Condition Number," *Int J Num Meth Eng*, V53 , pp.1377-1391, 2002.

[13] Freitag, L.A., Plassmann, P., "Local Optimization-Based Simplicial Mesh Untangling and Improvement," *Int J Num Meth Eng*, V49, pp. 109-125, 2000.

[14] Frey, P.J., George, P.L., *Mesh Generation*, Hermes Sci Publishing, Oxford, 2000.

[15] George, P.L., Borouchaki, H., *Delaunay Triangulation and Meshing: Application to Finite Elements*. Editions Hermes, Paris, 1998.

[16] González-Yuste, J.M., Montenegro, R., Escobar, J.M., Montero, G., Rodríguez, E., "Local Refinement of 3-D Triangulations Using Object-Oriented Methods," *Adv Eng Soft*, V35, pp. 693-702, 2004.

[17] Knupp, P.M., "Achieving Finite Element Mesh Quality Via Optimization of the Jacobian Matrix Norm and Associated Quantities. Part II-A Frame Work for Volume Mesh Optimization and the Condition Number of the Jacobian Matrix," *Int J Num Meth Eng*, V48, pp. 1165-1185, 2000.

[18] Knupp, P.M., "Algebraic Mesh Quality Metrics," *SIAM J Sci Comput*, V23, pp. 193-218, 2001.

[19] Kossaczky, I., "A Recursive Approach to Local Mesh Refinement in Two and Three Dimensions," *J Comput Appl Math*, V55, pp. 275-288, 1994.

[20] Li, X., Guo, X., Wang, H., He, Y., Gu, X., Qin, H., "Harmonic Volumetric Mapping for Solid Modeling Applications," *Proc. of ACM Solid and Physical Modeling Symposium*, pp. 109-120. Association for Computing Machinery, Inc., 2007.

[21] Lin, J., Jin, X., Fan, Z., Wang, C.C.L., "Automatic PolyCube-Maps," *Lecture Notes in Computer Science*, V4975, pp. 3-16, 2008.

[22] Montenegro, R., Cascón, J.M., Escobar, J.M., Rodríguez, E., Montero, G., "Implementation in ALBERTA of an Automatic Tetrahedral Mesh Generator," *Proc 15th Int Meshing Roundtable*, Birmingham, USA, pp. 325-338, 9/06

[23] Montenegro, R., Cascón, J.M., Escobar, J.M., Rodríguez, E., Montero, G., "An Automatic Strategy for Adaptive Tetrahedral Mesh Generation," *Appl Num Math*, V59, pp. 2203-2217, 2009.

[24] Schmidt, A., Siebert, K.G., *Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA*. Lecture Notes in Computer Science and Engineering, V42, Springer, Berlin, 2005.

[25] Schmidt, A., Siebert, K.G. "ALBERTA - An Adaptive Hierarchical Finite Element Toolbox," http://-www.alberta-fem.de/

[26] Tarini, M., Hormann, K., Cignoni, P., Montani, C., "Polycube-Maps," *ACM Trans Graph*, V23, pp. 853-860, 2004.

[27] Thompson, J.F., Soni, B., Weatherill N., *Handbook of Grid Generation*, CRC Press, London, 1999.

[28] Wang, H., He, Y., Li, X., Gu, X., Qin, H., "Polycube Splines," *Comput Aid Geom Design*, V40, pp. 721-733, 2008.