

The meccano method for simultaneous volume parametrization and mesh generation of complex solids

R Montenegro¹, J M Cascón², J M Escobar¹, E Rodríguez¹ and G Montero¹

¹ Institute for Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, Las Palmas de G.C., Spain

² Department of Economics and Economic History, Faculty of Economics and Management, University of Salamanca, Spain

E-mail: {rmontenegro, jmescoar, erodriguez, gmontero}@siani.es, casbar@usal.es

Abstract. The meccano method is a novel and promising mesh generation method for simultaneously creating adaptive tetrahedral meshes and volume parametrizations of a complex solid. We highlight the fact that the method requires minimum user intervention and has a low computational cost. The method builds a 3-D triangulation of the solid as a deformation of an appropriate tetrahedral mesh of the meccano. The new mesh generator combines an automatic parametrization of surface triangulations, a local refinement algorithm for 3-D nested triangulations and a simultaneous untangling and smoothing procedure. At present, the procedure is fully automatic for a genus-zero solid. In this case, the meccano can be a single cube. The efficiency of the proposed technique is shown with several applications.

1. Introduction

The automatic and adaptive mesh generation is a crucial aspect in finite element applications [4, 16, 17, 27, 28, 29, 31]. 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, but the 3-D problem is still open [1]. 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.

In this direction, we have introduced the new meccano technique [2, 3, 24, 25] 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 (a particular case is when meccano is composed by connected cubes, i.e. a polycube.). The idea of the method is to build a volume mesh of the solid as a deformation of an appropriate tetrahedral mesh of the meccano.

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 meccano method combines a local refinement/derefinement algorithm for 3-D nested triangulations [21], a parameterization of surface triangulations [9] and our simultaneous untangling and smoothing procedure [6]. At present, the meccano technique has been implemented by using the local refinement/derefinement of Kossaczky [21], but the idea could be implemented with other types of local refinement algorithms [18]. We note that 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 [26]. Adaptive nested meshes have been constructed with refinement and derefinement algorithms for evolution problems [8]. Mappings between physical and parametric spaces have been analyzed by several authors. Significant advances in surface parametrization have been done in [9, 11, 12, 23, 30, 32], but the volume parametrization is still open. Floater et al [13] 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 [19, 20]. 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 [15, 14]: the first is for mesh untangling and the second one for mesh smoothing. Each step leads to a different objective function. The meccano method uses the improvement proposed by [6, 7], where a simultaneous untangling and smoothing guided by the same objective function is introduced.

Some advantages of the meccano method are that: solid 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 applications of the meccano method to complex genus-zero solid, i.e. a solid whose boundary is a surface that is homeomorphic to the surface of a sphere. In this case, we assume that the solid geometry is defined by a triangulation of its surface and that the user only has 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 an automatic method to decompose the solid 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 [9, 10, 11, 12]. 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 [23, 30, 32]. 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 [22].

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 algorithm of the meccano method

The main steps of the general *meccano tetrahedral mesh generation algorithm* are summarized in this section. A detailed description of the method can be analyzed in [2, 3, 24, 25]. 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

- (i) Construct a meccano approximation of the 3-D solid formed by polyhedral pieces.
- (ii) Define an admissible mapping between the meccano boundary faces and the solid boundary.
- (iii) Build a coarse tetrahedral mesh of the meccano.
- (iv) 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.
- (v) Move the boundary nodes of the meccano to the object surface with the mapping (ii).
- (vi) Relocate the inner nodes of the meccano.
- (vii) 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 third step, 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 of the meccano method from a cube

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.

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++. 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 [9, 11].

We have implemented the meccano method by using the local refinement of ALBERTA. This code is an adaptive multilevel finite element toolbox developed in C, see <http://www.alberta-fem.de>. This software can be used to solve several types of 1-D, 2-D or 3-D problems. ALBERTA uses the Kossaczky refinement algorithm [21] and requires an initial mesh topology [26]. 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 an armadillo and the second to a bone. We have obtained a surface triangulation of these objects from internet.

3.1. Application to an armadillo

The original surface triangulation of the Armadillo has been obtained from the Stanford Computer Graphics Laboratory, <http://graphics.stanford.edu/data/3Dscanrep/>, and is shown in Figure 1. It has 30000 triangles and 15002 nodes. The bounding box of the solid is defined by the points $(x, y, x)_{min} = (-60, -50, -26)$ and $(x, y, z)_{max} = (68, 66, 90)$.

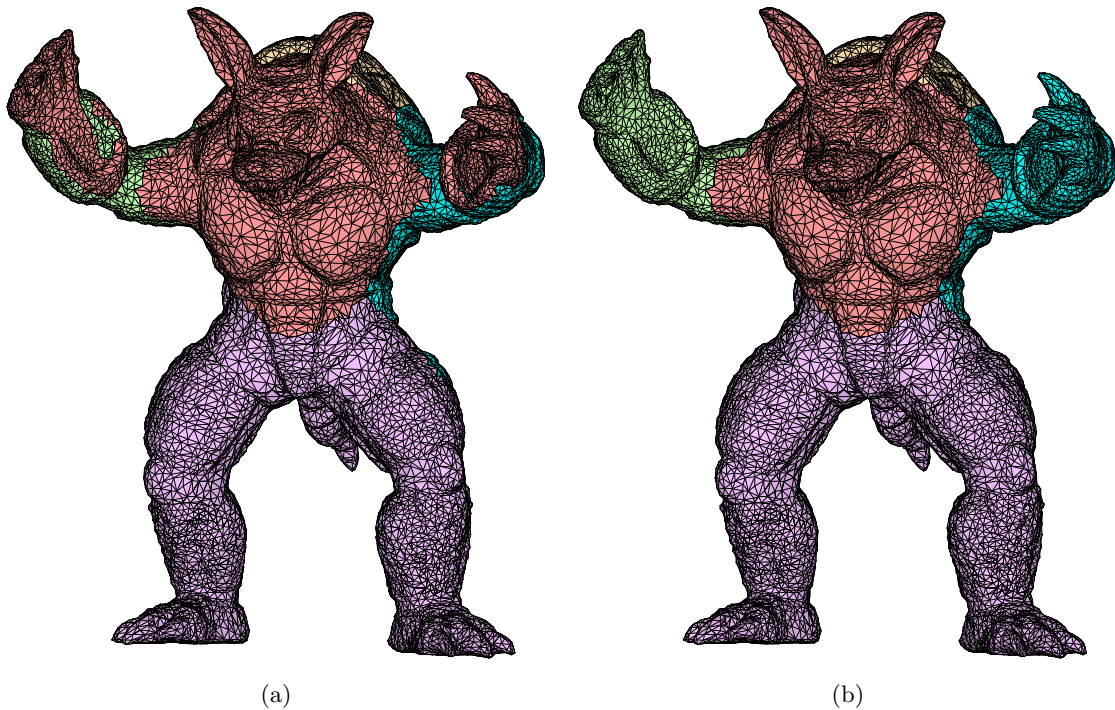


Figure 1. Original surface triangulation of Armadillo, (a) initial subdivision in connected subtriangulations using Voronoi diagram and (b) compatible partition $\{\mathcal{T}_S^i\}_{i=0}^5$ obtained after applying our reduction algorithm, removing dividing edges and smoothing the interfaces between patches.

We consider a unit cube as meccano. Its center is placed inside the solid at the point $(7.5, 17.5, 55.5)$. We obtain an initial subdivision of Armadillo surface in eleven maximal connected subtriangulations using the Voronoi diagram associated with the centers of the cube faces, see Figure 1(a). In order to get a compatible decomposition of the surface triangulation,

we use the iterative procedure proposed in [3] to reduce the current eleven vertices of the graph \mathcal{G}_S to six. Figure 1(b) shows the resulting compatible partition $\{\mathcal{T}_S^i\}_{i=0}^5$.

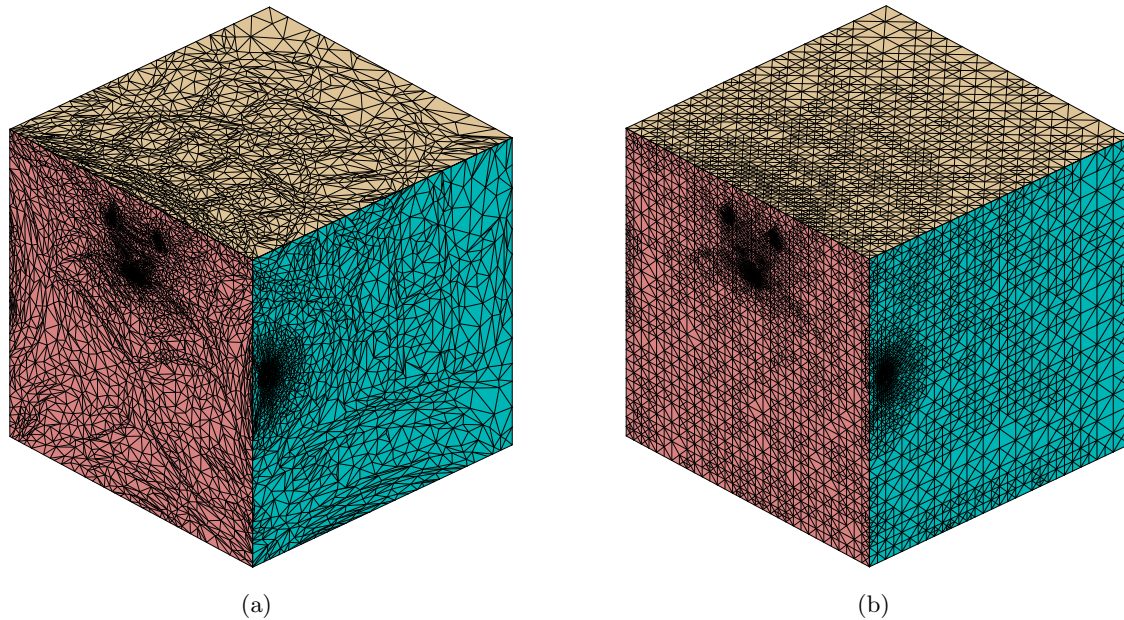


Figure 2. (a) Floater's parametrization of $\{\mathcal{T}_S^i\}_{i=0}^5$ on corresponding cube faces, (b) cube tetrahedral mesh obtained by the meccano method after refinement.

We map each surface patch Σ_S^i to the cube face Σ_C^i by using the Floater parametrization [9]. The definition of the one-to-one mapping between the cube and Armadillo boundaries is straightforward once the parametrization of the Armadillo surface triangulation is built, see Figure 2(a). A detail of the each parametrization of \mathcal{T}_S^i on the corresponding cube face Σ_C^i is shown in Figure 3.

The cube is divided into six tetrahedra. In order to approximate the Armadillo surface with a tolerance $\varepsilon_2 = 0.1$, see [3], we refine the cube mesh by applying 68 Kossaczky recursive bisections, see Figures 2(b) and 5(a). This mesh contains 144964 tetrahedra and 33889 nodes, with 31316 triangles and 15660 nodes on its boundary. The mapping of the cube external nodes to the Armadillo surface produces a 3-D tangled mesh with 10871 inverted elements, see Figure 4(a). The relocation of inner nodes by using volume parametrizations reduces the number of inverted tetrahedra to 186. We use the tetrahedral mesh optimization, presented in [6], such that the mesh is untangled in 6 iterations. The mesh quality is improved to a minimum value of 0.08 and an average $\bar{q}_\kappa = 0.71$ after 6 smoothing iterations. We note that the meccano technique generates a high quality tetrahedral mesh, see Figure 4(b): only 4 tetrahedra have a quality of less than 0.1, 150 less than 0.2 and 1046 less than 0.3. In order to show the efficiency of the mesh optimization technique inside the Armadillo, we display in Figure 5 two cross sections before (b) and after (c) its application. The location of the cube can be observed in Figure 5(b). We also note that, due to the high quality surface triangulation obtained with our method, the mesh improvement is not significant if we previously apply the smoothing surface triangulation algorithm introduced in [7].

The CPU time for constructing the final mesh of the Armadillo is approximately 112.78 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: 5.45 seconds for the subdivision

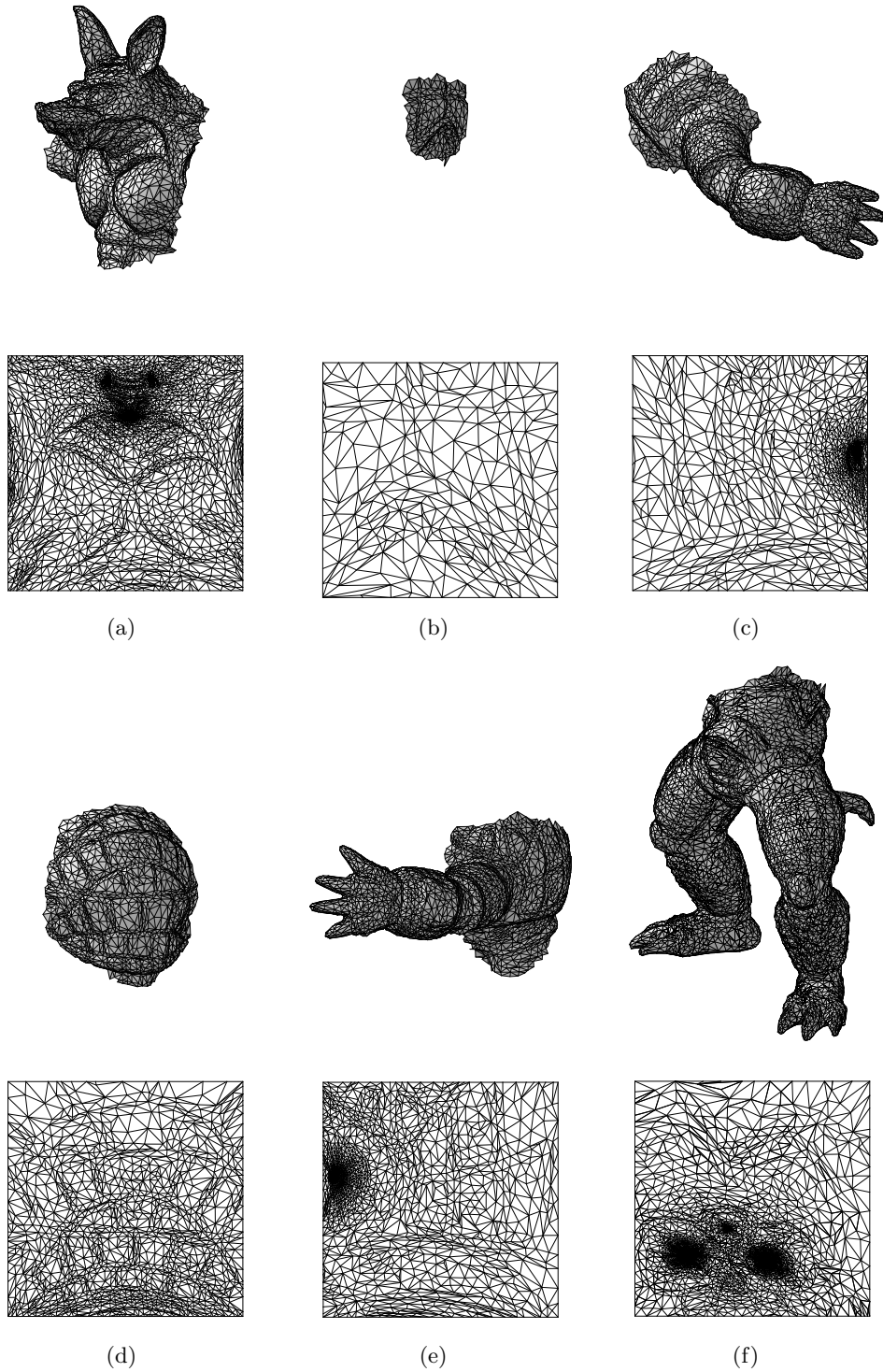


Figure 3. Detail of Floater's parametrization of Armadillo surface subtriangulations $\{\mathcal{T}_S^i\}_{i=0}^5$.

of the initial surface triangulation into six patches, 1.01 seconds for the Floater parametrization, 33.11 seconds for the Kossaczky recursive bisections, 2.25 seconds for the external node mapping and inner node relocation, and 80.96 seconds for the mesh optimization.

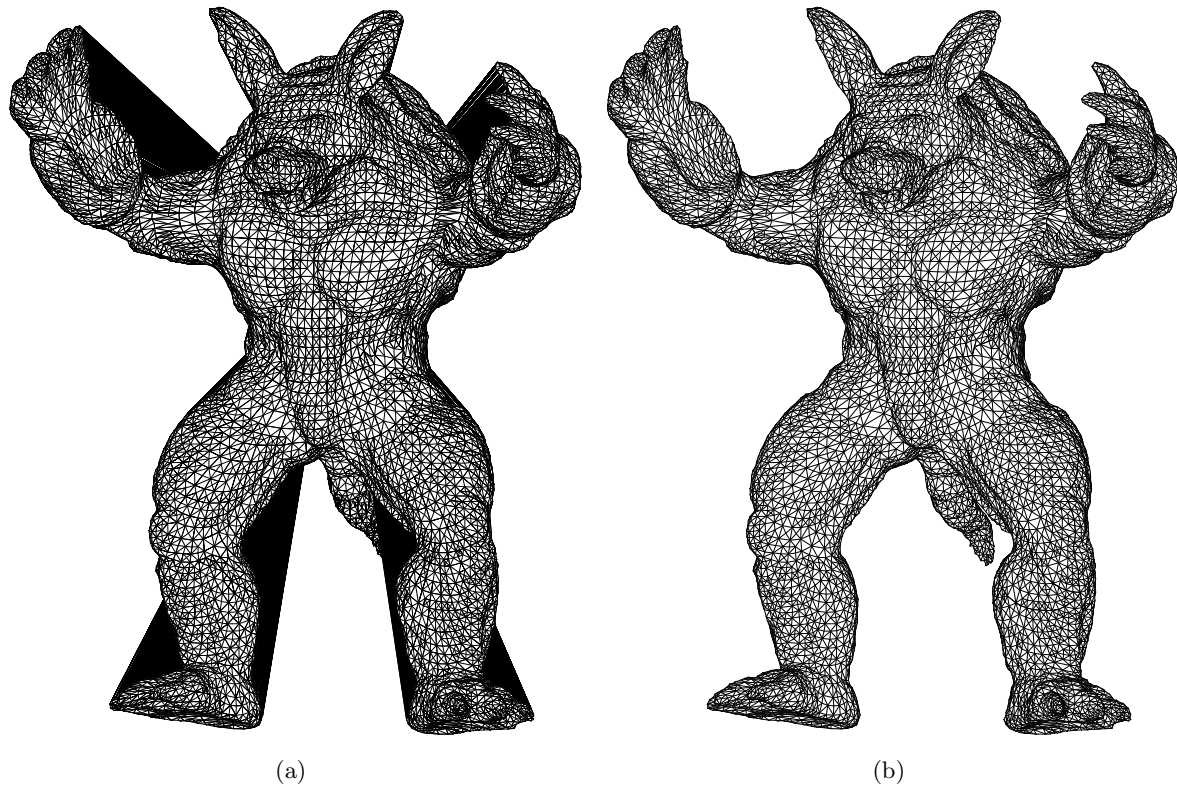


Figure 4. (a) Tangled mesh after the mapping of the external nodes of the cube to the Armadillo surface and (b) resulting valid mesh after inner node relocation and mesh optimization.

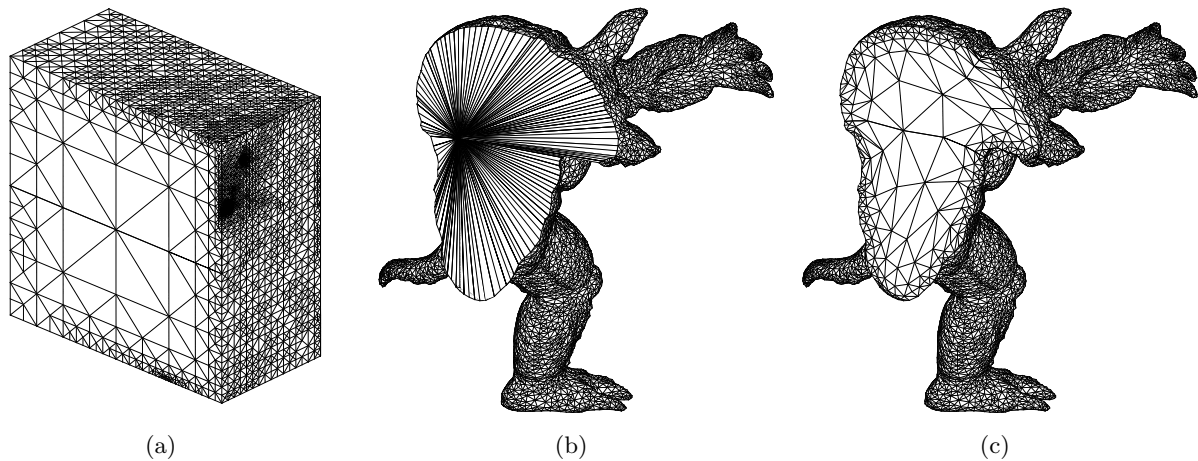


Figure 5. Cross sections of the cube (a), and the Armadillo tetrahedral meshes before (b) and after (c) the application of the mesh optimization procedure.

3.2. Application to a 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 6. 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

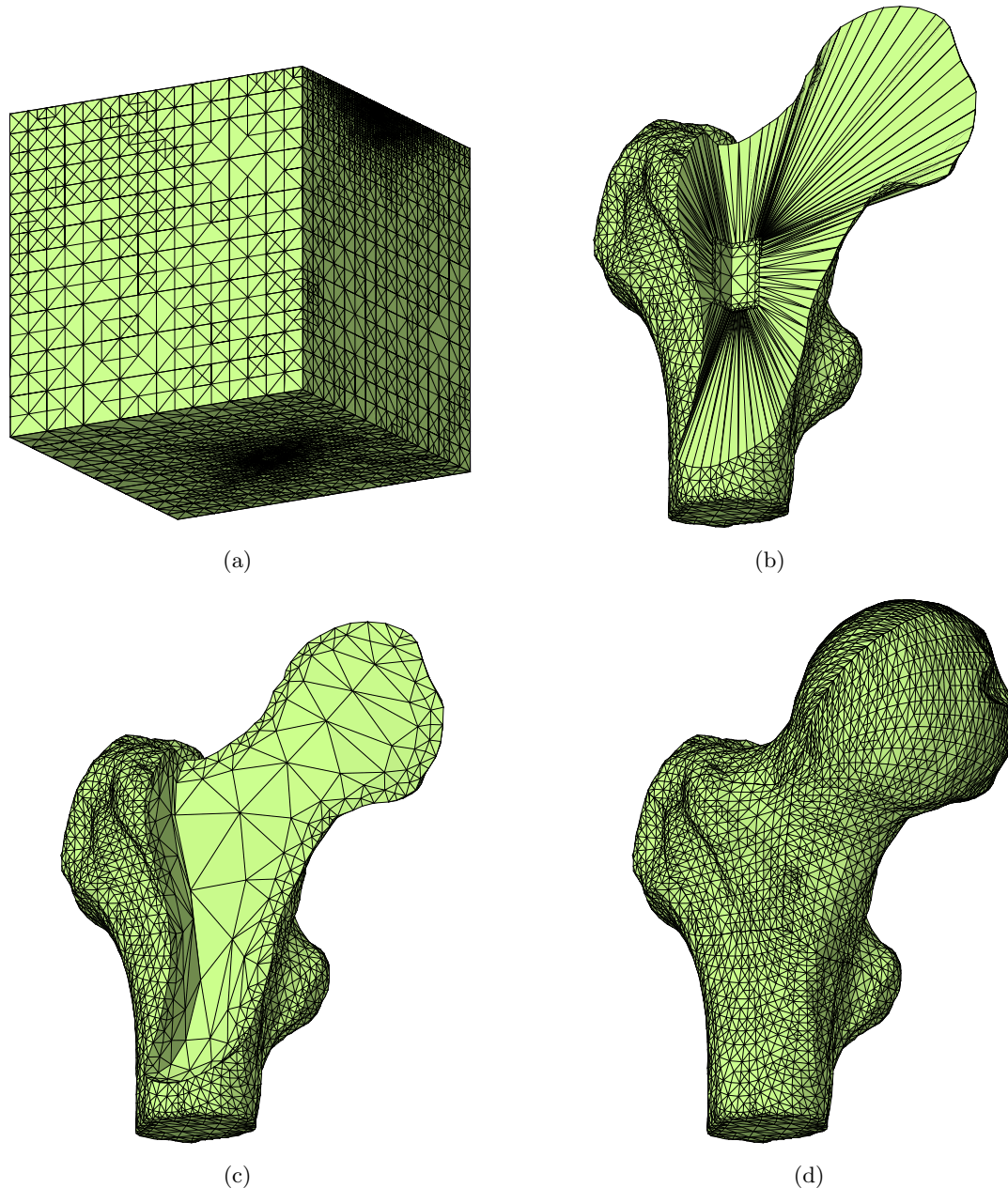


Figure 6. The meccano (a) and a cross sections of the Bone tetrahedral mesh before (b) and after (c) the application of the mesh optimization procedure. (d) Resulting tetrahedral mesh of the Bone obtained by the meccano method.

high quality tetrahedra mesh: the minimum mesh quality is 0.15 and the average quality is 0.64.

We note that the meccano method uses to construct more than 1000 tetrahedra per second on a Dell precision 690, 2 Dual Core Xeon processor and 8 Gb RAM memory.

Finally, we summarize a comparison between our method and standard tetrahedral mesh generation techniques [27, 28, 29]. On the one hand, one of the most important contributions of the meccano method is the resulting volume parametrization of the solid. It can have interesting applications in solid modeling and numerical simulation. For example, the application of isogeometric analysis [1, 5] can be easier. On the other hand, our volume meshes can be utilized in adaptive finite element applications by using the Kossaczky's algorithm. The local refinement steps are very fast because the sequence of solid meshes is defined from the coarse mesh of the meccano, i.e., the dividing edge for tetrahedron bisection is known straightforward. In addition, we note that the minimum mesh quality is bounded during all the mesh adaptation process, because similar elements appear after three consecutive bisections.

For a given solid surface triangulation, we have checked that a constrained Delaunay tetrahedralization [29] can be faster than our method, but the resulting meshes have lower quality. If the admissible minimum quality is increased, many points can be added and the number of tetrahedra increases to reach the objective. If only a conforming Delaunay tetrahedralization is desired, the number of tetrahedra can be much higher than in our method. A great advantage of our method is that the adaptive node distribution on the boundary and the inner of the solid is more structured, because the positions are fixed with a sequence of nested meshes. In the case of advancing front [27], the resulting mesh depends on the quality of the given surface triangulation. We have seen that the optimization of this triangulation, changing the mesh topology, can be too costly.

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 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).

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 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.

Acknowledgments

This work was partially supported by the "Secretaría de Estado de Investigación" of Spanish Government, "Ministerio de Ciencia e Innovación", and FEDER, grant contracts: CGL2008-06003-C03 and UNLP08-3E-010.

References

- [1] Bazilevs Y, Calo V M, Cottrell J A, Evans J, Hughes T J R, Lipton S, Scott M A and Sederberg T W 2008 Isogeometric analysis: Toward unification of computer aided design and finite element analysis *Trends in Engineering Computational Technology* (Stirling: Saxe-Coburg Publications) chapter 1 pp 1–16

- [2] Cascón J M, Montenegro R, Escobar J M, Rodríguez E and Montero G 2007 A new meccano technique for adaptive 3-D triangulations *Proc. 16th Int. Mesh. Round. (Seattle)* (NY: Springer-Verlag) pp 103–20
- [3] Cascón J M, Montenegro R, Escobar J M, Rodríguez E and Montero G 2009 The meccano method for automatic tetrahedral mesh generation of complex genus-zero solids *Proc. 18th Int. Meshing Roundtable (Salt Lake City)* (New York: Springer-Verlag) pp 463–80
- [4] Carey G F 1997 *Computational Grids: Generation, Adaptation, and Solution Strategies* (Washington: Taylor & Francis)
- [5] Cottrell J A, Hughes T J R, Bazilevs Y 2009 *Isogeometric Analysis: Toward Integration of CAD and FEA* (Chichester: John Wiley & Sons)
- [6] Escobar J M, Rodríguez E, Montenegro R, Montero G and González-Yuste J M 2003 Simultaneous untangling and smoothing of tetrahedral meshes *Comput. Meth. Appl. Mech. Eng.* **192** 2775–87
- [7] Escobar J M, Montero G, Montenegro R and Rodríguez E 2006 An algebraic method for smoothing surface triangulations on a local parametric space *Int. J. Num. Meth. Eng.* **66** 740–60
- [8] Ferragut L, Montenegro R and Plaza A 1994 Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems *Comm. Num. Meth. Eng.* **10** 403–12
- [9] Floater M S 1997 Parametrization and smooth approximation of surface triangulations *Comput. Aid. Geom. Design* **14** 231–50
- [10] Floater M S 2002 One-to-one piece linear mappings over triangulations *Math. Comput.* **72** 685–96
- [11] Floater M S 2003 Mean value coordinates *Comput. Aid. Geom. Design* **20** 19–27
- [12] Floater M S and Hormann K 2005 Surface parameterization: A tutorial and survey *Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization* (Berlin: Springer) pp 157–86
- [13] Floater M S and Pham-Trong V 2006 Convex combination maps over triangulations, tilings, and tetrahedral meshes *Advances in Computational Mathematics* **25** 347–56
- [14] Freitag L A and Knupp P M 2002 Tetrahedral mesh improvement via optimization of the element condition number *Int. J. Num. Meth. Eng.* **53** 1377–91
- [15] Freitag L A and Plassmann P 2000 Local optimization-based simplicial mesh untangling and improvement *Int. J. Num. Meth. Eng.* **49** 109–25
- [16] Frey P J and George P L 2000 *Mesh Generation* (Oxford: Hermes Sci. Publishing)
- [17] George P L and Borouchaki H 1998 *Delaunay Triangulation and Meshing: Application to Finite Elements* (Paris: Editions Hermes)
- [18] González-Yuste J M, Montenegro R, Escobar J M, Montero G and Rodríguez E 2004 Local refinement of 3-D triangulations using object-oriented methods *Adv. Eng. Soft.* **35** 693–702
- [19] Knupp P M 2000 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.* **48** 1165–85
- [20] Knupp P M 2001 Algebraic mesh quality metrics *SIAM J. Sci. Comput.* **23** 193–218
- [21] Kossaczky I 1994 A recursive approach to local mesh refinement in two and three dimensions *J. Comput. Appl. Math.* **55** 275–88
- [22] Li X, Guo X, Wang H, He Y, Gu X and Qin H 2007 Harmonic volumetric mapping for solid modeling applications *Proc. of ACM Solid and Physical Modeling Symposium* (Association for Computing Machinery, Inc.) pp 109–20
- [23] Lin J, Jin X, Fan Z and Wang C C L 2008 Automatic polycube-maps *Lecture Notes in Computer Science* **4975** 3–16
- [24] Montenegro R, Cascón J M, Escobar J M, Rodríguez E and Montero G 2006 Implementation in ALBERTA of an automatic tetrahedral mesh generator *Proc. 15th Int. Meshing Roundtable (Birmingham)* (New York: Springer-Verlag) pp 325–38
- [25] Montenegro R, Cascón J M, Escobar J M, Rodríguez E and Montero G 2009 An automatic strategy for adaptive tetrahedral mesh generation *Appl. Num. Math.* **59** 2203–17
- [26] Schmidt A and Siebert K G 2005 *Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA* (Lecture Notes in Computer Science and Engineering vol 42) (Berlin: Springer)
- [27] Schöberl J 1997 NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules *Comput. Visual. Sci.* **1** 41–52
- [28] Si H 2008 Adaptive tetrahedral mesh generation by constrained Delaunay refinement *Int. J. Num. Meth. Eng.* **75** 856–880
- [29] Si H 2009 *TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator* <http://tetgen.berlios.de>, version 1.4.3
- [30] Tarini M, Hormann K, Cignoni P and Montani C 2004 Polycube-maps *ACM Trans. Graph.* **23** 853–60
- [31] Thompson J F, Soni B and Weatherill N 1999 *Handbook of Grid Generation* (London: CRC Press)
- [32] Wang H, He Y, Li X, Gu X and Qin H 2008 Polycube splines *Comput. Aid. Geom. Design* **40** 721–33