Elsevier Editorial System(tm) for Computer Methods in Applied Mechanics and Engineering

Manuscript Draft

Manuscript Number: CMAME-D-11-00269R1

Title: A new approach to solid modeling with trivariate T-splines based on mesh optimization

Article Type: Research Paper

Keywords: Trivariate T-spline; isogeometric analysis; volumetric parameterization; mesh optimization; meccano method

Corresponding Author: Professor R. Montenegro, Ph.D.

Corresponding Author's Institution: University of Las Palmas de Gran Canaria

First Author: J.M. Escobar, Ph.D.

Order of Authors: J.M. Escobar, Ph.D.; J.M. Cascón, Ph.D.; E. Rodríguez, Ph.D.; R. Montenegro, Ph.D.

Abstract: We present a new method to construct a trivariate T-spline representation of complex genus-zero solids for the application of isogeometric analysis. The proposed technique only demands a surface triangulation of the solid as input data. The key of this method lies in obtaining a volumetric parameterization between the solid and the parametric domain, the unitary cube. To do that, an adaptive tetrahedral mesh of the parametric domain is isomorphically transformed onto the solid by applying a mesh untangling and smoothing procedure. The control points of the trivariate T-spline are calculated by imposing the interpolation conditions on points sited both on the inner and on the surface of the solid. The distribution of the interpolating points is adapted to the singularities of the domain in order to preserve the features of the surface triangulation.

Reviewers' comments:

Reviewer #1: This paper presents an approach to construct a trivariate T-spline representation of complex genus-zero solids using a volumetric parameterization mapping between the solid and the parametric unitary cube.

Does the presented mesh optimization method work for objects with other genus number? You may need to construct one simple object with the same topology, not a cube any more. Please comment on the challenges and which part of the algorithm needs to be modified.

As it is commented in the conclusions of the paper, we introduced in [24] and [25] the way to construct volumetric parameterizations of solids of genus greater than zero, where the surface parameterization is explicitly given. The method is based on the construction of a rough approximation of the solid joining cuboids (the meccano). But the construction of a T-spline representation also demands a T-mesh adapted to the discretization of the meccano. Maybe, the easiest way to undertake this task is partitioning the bounding box enclosing the meccano following the standard octree subdivision scheme. This subdivision produces vertices both inside and outside the meccano, but only the inner vertices must be considered as anchors. The external vertices will be used to complete the unclamped knot vectors.

We think that the procedure above described could be accomplished without major problems. Possibly, the most complex question, from a technical point of view, is the automatic generation of a meccano by using a specific CAD system and the corresponding surface parameterization. This last topic is widely discussed in the literature and we could proceed as in PolyCube-Maps [20, 29, 30].

As we all know, conformal mapping is a nice mapping method which preserves angles. What are the differences between the presented method and conformal mapping in terms of angle preservation, efficiency and robustness?

Our technique could be considered as a method that preserves angles. On the one hand, we obtain the surface mapping using Floater parameterization, which is shape preserving. On the other hand, the interior nodes are relocated by considering rectangular tetrahedra as target. In this sense, our procedure tries to preserve angle, at least in a rough way. We also remark that the restriction to map the input surface to cube faces could produce some distortion around the neighborhood of the edges. The conformal mapping between the input surface and a sphere has nice properties [Gu et. Al 2004] but they are not compatible with our scheme in its present form.

Harmonic volumetric functions have also been used to generate a volumetric parameterization [19,22,23*]. For example, Li et al. [19] construct a harmonic volumetric mapping through a meshless procedure by using a boundary method. The algorithm can be applied to any genus data but it is complex and requires placing some source and collocation points on an offset surface. Optimal results of source positions are unknown, and in practice they are chosen in a trial-and-error manner or with the help of human experience. Therefore, the problem is ill-conditioned and regular system solvers often fail.*

Martin et al. [22,23] present a methodology based on discrete harmonic functions to parameterize a solid. They solve several Laplace's equations, first on the surface and then on the complete 3-D domain with FEM, and use a Laplacian smoothing to remove irregularities. During the process, new vertices are inserted in the mesh and retriangulations (in 2-D and 3-D) are applied in order to introduce the new vertex set in the mesh. The user has to make an initial choice of two critical points to establish the surface parameterization and to fix a seed for generating the skeleton. The parameterization has degeneracy along the skeleton. The extension to genus greater than zero [23] requires finding suitable midsurfaces.

Our technique is simple and automatically produces a T-spline adapted to the geometry with a low computational complexity and low user intervention. As in other methods, our parameterization can introduce some distortion, especially along the cube edges.

- Remark: The above paragraphs in italics have been included in the introduction of the paper.

There is a lack of discussion on the properties of the constructed solid T-spline, such as surface accuracy, continuity, quality and linear independence.

The T-spline is enforced to interpolate all the nodes of the tetrahedral mesh and this mesh is as close as we want to the input surface (a triangular mesh in our case). Moreover, the interpolating points are exactly sited on the input surface. These reasons suggest a good accuracy between the T-spline and the input surface. In order to estimate the gap between both surfaces we have analyzed the differences between the volumes enclosed by input surface and the T-spline. The first volume is measured by applying the divergence theorem and the second one is calculated by

$$\iiint_C \det\left(\mathrm{S}_{\xi_1}, \mathrm{S}_{\xi_2}, \mathrm{S}_{\xi_2}\right) \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3$$

with 8 Gaussian quadrature points in each cell. The quadrature points with negative Jacobians have been rejected from the calculations.

As a example, for bunny application we have:

Volume (input surface): 754.9
Volume (tetrahedral mesh): 750.9 (difference of 0.5% in relation to the input).
Volume (isogeometric model): 757.4 (difference of 0.3% in relation to the input).
Only 39 cells out of 9696 have any quadrature point with negative Jacobian.

- Remark: The above paragraphs in italics have been included in section 4.2.

The T-spline $S(\xi)$ defined in (9) is the sum of rational C^2 blending functions, so $S(\xi)$ is also a C^2 function. Nevertheless, as the surface of the solid is the union of six patches obtained by mapping the six faces of the cube, and these faces match with C^0 continuity, we only can assure the C^0 continuity for the surface of the solid. We think the construction of more regular surfaces (C^1 or C^2) would be possible by adding new blending functions and imposing appropriated conditions on the control points in order to assure a smooth junction between patches.

At present, there are no quality metrics for isogeometric analysis analogous to the ones for traditional FEA. We have analyzed the scaled Jacobian in the quadrature points of the cells as a tentative of a quality measure.

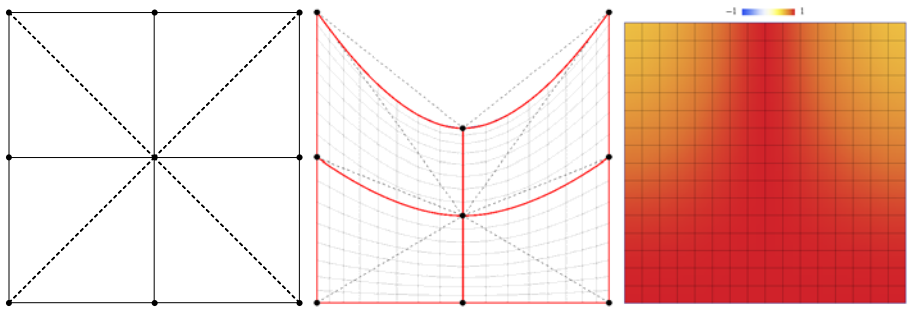
Recently Buffa et al. [5] have analyzed the linear independence of the bi-cubic T-spline blending functions corresponding to some particular T-meshes. In particular, they prove linear independence of hierarchical T-meshes (2D) generated as the refinement of a coarse and uniform T-mesh (this is the in 2D counterpart to our case). This result requires two assumptions. The first one is about the refinement algorithm. It must satisfy that the spaces of blending functions associated to the sequence of T-meshes are nested (the refinement algorithm of [Sederberg et. al 2004] produces T-meshes with this property). The second assumption says that in each step of refinement procedure we only have to add new anchors living on new edges. In 2D, the hierarchical refinement can be decomposed in elementary steps that satisfy the previous assumptions. Unfortunately, this is not possible in 3D and thus we cannot extend the result of Buffa et al. to our T-meshes. However, we think that a similar result can be proved if we consider, as an elementary refinement step, the addition of a cross in a cell, but at this point we do not have a proof.

Moreover, *the linear independence has become evident in all the applications considered until now, as the resolution of eq. (11) is only possible if the blending functions $R_\alpha(\xi)$ are linearly independent.*

- Remark: The above paragraphs in italics have been included in section 3.

In Figure 10, there are some concave elements which yield negative Jacobians and influence the convergence of isogeometric analysis solver. How to solve this problem?

At present we can't give a conclusive response to this question. The strategy taken in the paper consists on refining the degenerated cells in order to reduce the regions in which the Jacobians become negative. Note that we can get a T-spline representation of non-convex domains without inverted Jacobians, as it is shown in Figure (R1).



R1 (a) R1 (b) R1 (c)

Fig. R1 (a): Parametric domain.
Fig. R1 (b): Non convex real domain.
Fig. R1 (c): Representation of the scaled Jacobian in the parametric domain.

The problem arises when a corner of the T-mesh is mapped into a reentering vertex, just like in Figure 10 of the paper, reproduced below.

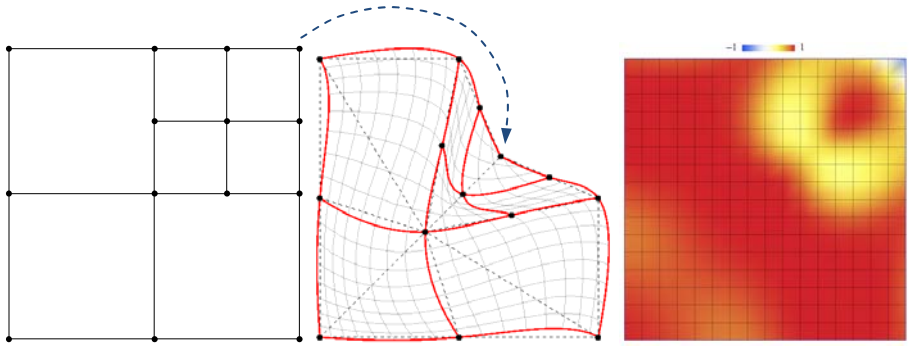


Fig. 10 (b) Fig. 10 (d) Fig. 10 (f)

Fig. 10 (b): Parametric domain (T-Mesh) refined on the corner
Fig. 10 (d): Real domain with a reentering corner.
Fig. 10 (f): Representation of the scaled Jacobian in the parametric domain.

In Figure R2 it is shown the effect of undertaking an extra refinement around the reentering vertex. It can be seen how the region involving negative or close to zero Jacobians is reduced, but it does not disappear.

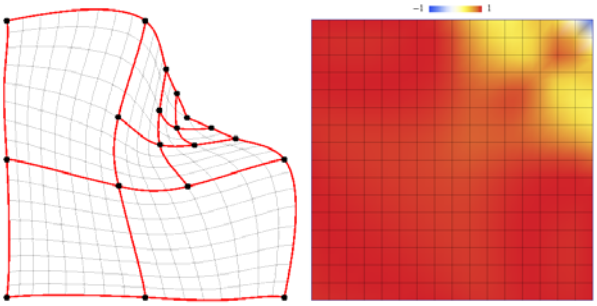


Fig. R2

Another alternative is relaxing the geometric constraints allowing the reentering vertex to move slightly away from its exact position, such as it is shown in Fig. R3.

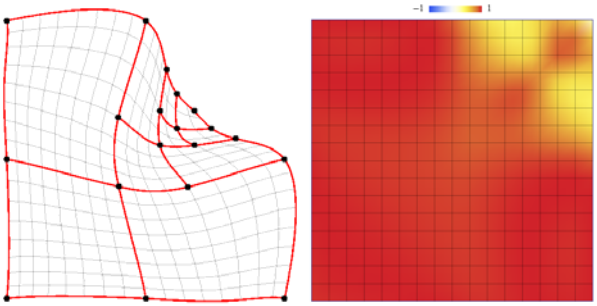


Fig. R3

It is clear that the orthogonality of the cells is a measure of the deformation produced by the mapping between parametric and real spaces. We are studying at present a new parameterization strategy that distributes the inner nodes in such a way that the cells tend to be as orthogonal as possible. Although this new parameterization works properly for inner cells, it can't solve the problem associated to the reentering vertices, as these nodes are restricted to move on the boundary.

In addition, there is no any simulation results using isogeometric analysis. It will be plausible it some simulations can be provided.

We agree with the referee, but we have not implemented any isogeometric code. Our purpose is to apply our isogeometric solid modeling to engineering problems in the next future. However, we have used the tetrahedral meshes generated by the meccano method to solve elliptic and parabolic problems with standard adaptive FEM. You can see the adaptive finite element mesh and the solution for a transient heat transfer problem in the first application of:

http://www.dca.iusiani.ulpgc.es/proyecto2008-2011/html_ingles/Resultados_Proyecto.html

To some extent, the volume evaluation of the bunny isogeometric model can be considered as an elementary application.

Pictures in Figure 7 (Stanford bunny) are missing.

We can see correctly the pictures in the pdf file that was generated by the journal web system (see detail below). We don´t know which is the problem. Please, suggest us how to proceed if the problem persists.

In Fig. 7 it is shown the interpolation points and their associated control points for the Standford bunny of Fig. 11.

(a) (b)

Figure 7: The interpolation points (a), and their corresponding control points for the Standford bunny (b).

There are some grammar and spelling errors, which should be corrected by a careful proofreading.

English has been revised.

Other references:

[Gu et. al. 2004] Gu, X., Wang, Y., Chan, T., Thompson, P. and Yau, S. T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Trans. Med. Imaging 23, 8, 949-958.

[Sederberg et. al. 2004] Sederberg, T., Cardon, D., Finnigan, G., North, N., Zheng, J., Lyche, T. 2004. T-spline simplification and local refinement. ACM Trans. Graph. 23, 3, 276-283.

A new approach to solid modeling with trivariate T-splines based on mesh optimization

J.M. Escobar[a], J.M. Cascón[b], E. Rodríguez[a], R. Montenegro[a]

[a] *University of Las Palmas de Gran Canaria, University Institute for Intelligent Systems and Numerical Applications in Engineering (SIANI), Spain.*
[b] *University of Salamanca, Department of Economics and History of Economics, Faculty of Sciences, Spain.*

Abstract

We present a new method to construct a trivariate T-spline representation of complex genus-zero solids for the application of isogeometric analysis. The proposed technique only demands a surface triangulation of the solid as input data. The key of this method lies in obtaining a volumetric parameterization between the solid and the parametric domain, the unitary cube. To do that, an adaptive tetrahedral mesh of the parametric domain is isomorphically transformed onto the solid by applying a mesh untangling and smoothing procedure. The control points of the trivariate T-spline are calculated by imposing the interpolation conditions on points sited both on the inner and on the surface of the solid. The distribution of the interpolating points is adapted to the singularities of the domain in order to preserve the features of the surface triangulation.

Keywords: Trivariate T-spline, isogeometric analysis, volumetric parameterization, mesh optimization, meccano method.

1. Introduction

CAD models usually define only the boundary of a solid, but the application of isogeometric analysis [2, 3, 10] requires a fully volumetric representation. An open problem in the context of isogeometric analysis is how to generate a trivariate spline representation of a solid starting from the CAD description of its boundary. As it is pointed by Cotrell et al. in [10], "the most significant challenge facing isogeometric analysis is developing three-dimensional spline parameterizations from surfaces".

There are only a few works addressing this problem, and they all have in common the use of harmonic functions to establish the volumetric parameterization [19, 21, 22, 23, 30].

For example, Li et al. [19] construct a harmonic volumetric mapping through a meshless procedure by using a boundary method. The algorithm can be applied to any genus data but it is complex and requires placing some source and collocation points on an offset surface. Optimal results of source positions are unknown, and in practice they are chosen in a trial-and-error manner or with the help of human experience. Therefore, the problem is ill-conditioned and regular system solvers often fail.

Martin et al. [22, 23] present a methodology based on discrete harmonic functions to parameterize a solid. They solve several Laplace's equations, first on the surface and then on the complete 3-D domain with FEM, and use a Laplacian smoothing to remove irregularities. During the process, new vertices are inserted in the mesh and retriangulations (in 2-D and 3-D) are applied in order to introduce the new vertex set in the mesh. The user has to make an initial choice of two critical points to establish the surface parameterization and to fix a seed for generating the skeleton. The parameterization has degeneracy along the skeleton. The extension to genus greater than zero [23] requires finding suitable midsurfaces.

We propose a different approach in which the volumetric parameterization is accomplished by transforming a tetrahedral mesh from the parametric domain to the physical domain. This is a special feature of our procedure; we do not have to give the tetrahedral mesh of the solid as input, as it is a result of the parameterization process. Another characteristic of our work is that we use an interpolation scheme to fit a trivariate B-spline to the data, instead of an approximation, as other authors do. This performs a more accurate adaptation of the T-spline to the input data.

One of the main drawbacks of NURBS (see for example [26]) is that they are defined on a parametric space with a tensor product structure, making the representation of detailed local features inefficient. This problem is solved by the T-splines, a generalization of NURBS conceived by Sederberg [27] that enables the local refinement. The T-splines are a set of functions defined on a T-mesh, a tiling of a rectangular prism in $\mathbb{R}^3$ allowing T-junctions (see [2] and [27]).

In this paper we present a new method for constructing volumetric T-meshes of genus-zero solids whose boundaries are defined by surface triangulations. Our procedure can be summarized in two stages. In the first one,

a volumetric parameterization of the solid is developed. Broadly speaking, we can consider that the construction of a volumetric parameterization is a process in which an adaptive tetrahedral mesh, initially defined in the unitary cube $\mathcal{C} = [0,1]^3$, is deformed until it achieves the shape of the solid (the physical domain). This deformation only affects the positions of the nodes, that is, there is not any change in their connectivities: we say that both meshes are isomorphic. Given that a point is fully determined by the barycentric coordinates relative to the tetrahedron in which it is contained, we can define a one-to-one mapping between $\mathcal{C}$ and the solid assuming that the barycentric coordinates are the same in both spaces.

In the second stage, the modeling of the solid by trivariate T-splines is carried out. The control points of the T-splines are calculated enforcing the T-splines to verify the interpolation conditions. Here is where the volumetric parametrization plays its part, mapping the interpolation points from the parametric domain, the T-mesh, onto the solid. In our case, the T-mesh is an octree partition of $\mathcal{C}$ with a similar resolution than the tetrahedral mesh defined in $\mathcal{C}$.

Our technique is simple and it automatically produces a T-spline adapted to the geometry with a low computational complexity and low user intervention. As in other methods, our parameterization can introduce some distortion, especially along the cube edges.

The paper is organized as follows. In the next section we describe the main steps to parameterize a genus-zero solid onto a cube. Some parts of this section are taken from our previous works on mesh untangling and smoothing and the meccano method [7, 8, 11, 24, 25], but they have been adapted to the requirements of the present work. The representation of the solid by means of trivariate T-splines is developed in section 3. In section 4 we show a test problem and several applications that highlight the ability of our method for modeling complex objects. Finally, in section 5 we present the conclusions and set out some challenges.

2. Volumetric parameterization

2.1. Boundary mapping

The first step to construct a volumetric parameterization consists on establishing a bijective correspondence between the boundary of the cube and the solid. To do that, the given surface triangulation of the solid, $\mathcal{T}_S$, is divided in six patches or *connected subtriangulations*, $\mathcal{T}_S^i$ $(i = 1, 2, \ldots, 6)$,

having the same connectivities as the cube faces. Specifically, if we consider that each subtriangulation corresponds to a vertex of a graph and two vertices of the graph are connected if their corresponding subtriangulations have at least a common edge, then, the graphs corresponding to the solid and the graph of the cube must be isomorphic (see [8, 25] for details).

Once $\mathcal{T}_S$ is decomposed into six patches, we map each $\mathcal{T}_S^i$ to the corresponding cube face by using the parameterization of surface triangulations proposed by M. Floater in [13, 14]. This is a well-known method to transform a surface triangulation onto a plane triangulation defined in a convex domain, that is, the cube faces in our case. Many and more recent alternative solutions have been proposed to solve the surface parameterization (see for example the surveys [15, 16]), but in most of them the plane triangulation is not defined in a convex set, which is a restriction for us. Thus, if τ_F^i is the resulting triangulation on the i-th face of the cube, the parameterization $\Pi_F^i : \tau_F^i \to \mathcal{T}_S^i$ is a piece-wise linear function that maps a point p inside triangle $T \in \tau_F^i$ onto a point q belonging to triangle $\Pi_F^i(T) \in \mathcal{T}_S^i$ with identical barycentric coordinates.

In order to ensure the compatibility of $\{\Pi_F^i\}_{i=1}^6$, the boundary nodes of $\{\tau_F^i\}_{i=1}^6$ must coincide on common cube edges. The six transformations $\{\Pi_F^i\}_{i=1}^6$ define a global parameterization between $\tau_F = \bigcup_{i=1}^6 \tau_F^i$ and $\mathcal{T}_S$ given by

$$\Pi_F : \tau_F \to \mathcal{T}_S \tag{1}$$

The parameterization Π_F is used in the following step of the algorithm to map a new triangulation defined over the boundary of $\mathcal{C}$ onto the boundary of the solid.

2.2. Generation of an adapted tetrahedral mesh of the cube

Let consider $\mathcal{C}_K$ is a tetrahedral mesh of $\mathcal{C}$ resulting after applying several local bisections of the Kossaczky algorithm [18] to an initial mesh formed by six tetrahedra (see Fig. 1(a)). Three consecutive global bisections are presented in figures 1 (b), (c) and (d). The mesh of Fig. 1(d) contains 8 cubes similar to the one shown in Fig. 1(a). Therefore, the successive refinement of this mesh produces similar tetrahedra to those of figures 1 (a), (b) and (c).

If $\tau_K = \partial\mathcal{C}_K$ is the new triangulation defined on the boundary of $\mathcal{C}$, then we define a new parameterization

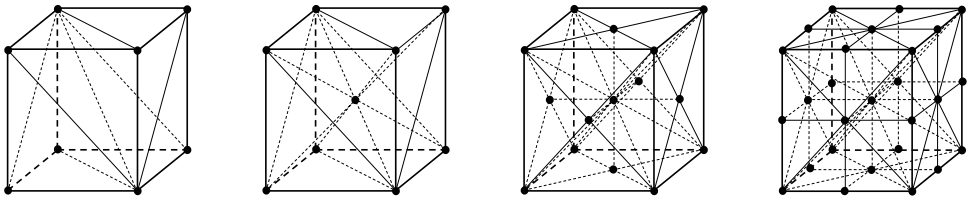


Figure 1: Refinement of a cube by using Kossaczky's algorithm: (a) cube subdivision into six tetrahedra, (b) bisection of all tetrahedra by inserting a new node in the cube main diagonal, (c) new nodes in diagonals of cube faces and (d) global refinement with new nodes in cube edges.

$$\Pi_K : \tau_K \rightarrow \mathcal{T}_S^* \tag{2}$$

where $\mathcal{T}_S^*$ is the surface triangulation obtained after Π_F-mapping the nodes of τ_K. The points of τ_K are mapped to $\mathcal{T}_S^*$ by preserving their barycentric coordinates. Note that $\mathcal{T}_S^*$ is an approximation of $\mathcal{T}_S$. In order to improve this approximation we must refine the tetrahedra of $\mathcal{C}_K$ in contact with the surface of the cube in such a way that the *distance* between $\mathcal{T}_S^*$ and $\mathcal{T}_S$ decreases until reaching a prescribed tolerance ε. The concept of *distance* between two triangulations can be defined and implemented in several ways. In our case is as follows:

Let T be a triangle of τ_K, where a, b and c are their vertices and let $p_k \in \{p_i\}_{i=1}^{N_q}$ be a Gauss quadrature point of T, then, the distance, $d\,(T)$, between $\Pi_K(T)$ and the underlaying triangulation $\mathcal{T}_S$ is defined as the maximum of the volumes of the tetrahedra formed by $\Pi_F(a)$, $\Pi_F(b)$, $\Pi_F(c)$ and $\Pi_F(p_k)$. If we considerer the distance between $\mathcal{T}_S^*$ and $\mathcal{T}_S$ as the maximum of all $d\,(T)$, the local refinement stops when $d\,(T) < \epsilon$ for all $T \in \tau_K$. A more accurate approach based on Hausdorff distance can be found in [4].

Once the adapted tetrahedral mesh $\mathcal{C}_K$ has been constructed by using the proposed method, the nodes of τ_K are mapped to the surface of the solid giving the triangulation $\mathcal{T}_S^*$, which is the final approximation of $\mathcal{T}_S$. Note that inner nodes of $\mathcal{C}_K$ stay in their initial positions, so the current tetrahedral mesh of the solid will most likely be tangled (see for example Fig. 13(a) of subsection 4.2). The following step plays a crucial roll in our procedure. We have to relocate the inner nodes in suitable positions such

that this tetrahedral mesh gets untangled and the distortion introduced by the associated parameterization is as small as possible (Fig. 13(b)).

2.3. Relocation of inner nodes

Usual techniques to improve the quality of a *valid* mesh, that is, one that does not have inverted elements, are based upon local smoothing. In short, these techniques consist of finding the new positions that the mesh nodes must hold, in such a way that they optimize an objective function. Such a function is based on a certain measurement of the quality of the *local submesh* $N(q)$, formed by the set of tetrahedra connected to the *free node* q. As it is a local optimization process, we can not guarantee that the final mesh is globally optimum. Nevertheless, after repeating this process several times for all the nodes of the current mesh, quite satisfactory results can be achieved. Usually, objective functions are appropriate to improve the quality of a valid mesh, but they do not work properly when there are inverted elements. This is because they present singularities (barriers) when any tetrahedron of $N(q)$ changes the sign of its Jacobian.

Most of the stated below is taken from [11], where we developed a procedure for untangling and smoothing meshes simultaneously. For that purpose, we use a suitable modification of the objective function such that it is regular all over $\mathbb{R}^3$. When a feasible region (subset of $\mathbb{R}^3$ where q could be placed, being $N(q)$ a valid submesh) exists, the minima of both the original and the modified objective functions are very close and, when this region does not exist, the minimum of the modified objective function is located in such a way that it tends to untangle $N(q)$. The latter occurs, for example, when the fixed boundary of $N(q)$ is tangled. With this approach, we can use any standard and efficient unconstrained optimization method to find the minimum of the modified objective function, see for example [1].

If we name $\mathcal{T}$ to the tetrahedral mesh of the solid once the inner nodes have been relocated, the corresponding volumetric parameterization is

$$\Pi : \mathcal{C}_K \to \mathcal{T} \tag{3}$$

A point p included in a tetrahedron of $\mathcal{C}_K$ is mapped, preserving barycentric coordinates, into a point q belonging to the transformed tetrahedron of $\mathcal{T}$.

2.3.1. Objective functions

Several tetrahedron shape measures could be used to construct an objective function. Nevertheless, those obtained by algebraic operations [17] are specially indicated for our purpose because they can be computed very efficiently and they allow us to choose the shape of the tetrahedra to optimize. Our objective is to relocate the nodes of $\mathcal{T}$ in positions where not only the mesh gets untangled, but also the distortion introduced by the parameterization is minimized.

Let T be a tetrahedral element of $\mathcal{T}$ whose vertices are given by $\mathbf{x}_k = (x_k, y_k, z_k)^T \in \mathbb{R}^3$, $k = 0, 1, 2, 3$ and T_R be the reference tetrahedron with vertices $\mathbf{u}_0 = (0, 0, 0)^T$, $\mathbf{u}_1 = (1, 0, 0)^T$, $\mathbf{u}_2 = (0, 1, 0)^T$ and $\mathbf{u}_3 = (0, 0, 1)^T$. If we choose $\mathbf{x}_0$ as the translation vector, the affine map that takes T_R to T is $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, where A is the Jacobian matrix of the affine map referenced to node $\mathbf{x}_0$, and expressed as $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \mathbf{x}_3 - \mathbf{x}_0)$.

Let consider that T_I is our ideal or target tetrahedron whose vertices are $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. If we take $\mathbf{v}_0 = (0, 0, 0)^T$ the linear map that takes T_R to T_I is $\mathbf{v} = W\mathbf{u}$, where $W = (\mathbf{v}_1 - \mathbf{v}_0, \mathbf{v}_2 - \mathbf{v}_0, \mathbf{v}_3 - \mathbf{v}_0)$ is its Jacobian matrix. As the parametric and real meshes are topologically identical, each tetrahedron of $\mathcal{T}$ has its counterpart in $\mathcal{C}_K$. Thus, in order to reduce the distortion in the volumetric parameterization we will fix the target tetrahedra of $N(q)$ as their counterparts of the local mesh in the parametric space.

The affine map that takes T_I to T is given by $\mathbf{x} = AW^{-1}\mathbf{v} + \mathbf{x}_0$, and its Jacobian matrix is $S = AW^{-1}$. Note that this weighted matrix S depends on the node chosen as reference, so this node must be the same for T and T_I. We can use matrix norms, determinant or trace of S to construct algebraic quality metrics of T. For example, the *mean ratio*, $Q = \frac{3\sigma^{\frac{2}{3}}}{|S|^2}$, is an easily computable algebraic quality metric of T, where $\sigma = \det(S)$ and $|S|$ is the Frobenius norm of S. The maximum value of Q is the unity, and it is reached when $A = \mu RW$, where μ is a scalar and R is a rotation matrix. In other words, Q is maximum if and only if T and T_I are similar. Besides, any flat tetrahedron has quality measure zero. We can derive an optimization function from this quality metric. Thus, let $\mathbf{x} = (x, y, z)^T$ be the position of the free node, and let S_m be the weighted Jacobian matrix of the m-th tetrahedron of $N(q)$. We define the objective function of $\mathbf{x}$, associated to an m-th tetrahedron as

$$\eta_m = \frac{|S_m|^2}{3\sigma_m^{\frac{2}{3}}} \tag{4}$$

Then, the corresponding objective function for $N(q)$ is constructed by using the p-norm of $(\eta_1, \eta_2, \ldots, \eta_M)$ as

$$|K_\eta|_p(\mathbf{x}) = \left[\sum_{m=1}^{M} \eta_m^p(\mathbf{x})\right]^{\frac{1}{p}} \tag{5}$$

where M is the number of tetrahedra in $N(q)$.

Although this optimization function is smooth in those points where $N(q)$ is a valid submesh, it becomes discontinuous when the volume of any tetrahedron of $N(q)$ goes to zero. It is due to the fact that η_m approaches infinity when σ_m tends to zero and its numerator is bounded below. In fact, it is possible to prove that $|S_m|$ reaches its minimum, with strictly positive value, when q is placed in the geometric center of the fixed face of the m-th tetrahedron. The positions where q must be located to get $N(q)$ to be valid, i.e., the feasible region, is the interior of the polyhedral set P defined as $P = \bigcap_{m=1}^{M} H_m$,where H_m are the half-spaces defined by $\sigma_m(\mathbf{x}) \geqslant 0$. This set can occasionally be empty, for example, when the fixed boundary of $N(q)$ is tangled. In this situation, function $|K_\eta|_p$ stops being useful as an optimization function. Moreover, when the feasible region exists, that is $int\ P \neq \emptyset$, the objective function tends to infinity as q approaches the boundary of P. Due to these singularities, it is formed a barrier which avoids reaching the appropriate minimum when using gradient-based algorithms, and when these start from a free node outside the feasible region. In other words, with these algorithms we can not optimize a tangled mesh $N(q)$ with the above objective function.

2.3.2. Modified objective functions

We proposed in [11] a modification in the previous objective function (5), so that the barrier associated with its singularities will be eliminated and the new function will be smooth all over $\mathbb{R}^3$. An essential requirement is that the minima of the original and modified functions are nearly identical when $int\ P \neq \emptyset$. Our modification consists of substituting σ in (5) by the positive and increasing function

$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \tag{6}$$

being the parameter $\delta = h(0)$. We represent in Fig. 2 the function $h(\sigma)$. Thus, the new objective function here proposed is given by

$$\left|K_{\eta}^{*}\right|_{p}(\mathbf{x}) = \left[\sum_{m=1}^{M}\left(\eta_{m}^{*}\right)^{p}(\mathbf{x})\right]^{\frac{1}{p}} \tag{7}$$

where

$$\eta_{m}^{*} = \frac{|S_m|^2}{3h^{\frac{2}{3}}(\sigma_m)} \tag{8}$$

is the modified objective function for the m-th tetrahedron.

The behavior of $h(\sigma)$ in function of δ parameter is such that, $\lim_{\delta\to 0} h(\sigma) = \sigma$, $\forall\sigma \geq 0$ and $\lim_{\delta\to 0} h(\sigma) = 0$, $\forall\sigma \leq 0$. Thus, if $int\ P \neq \emptyset$, then $\forall\mathbf{x} \in int\ P$ we have $\sigma_m(\mathbf{x}) > 0$, for $m = 1, 2, \ldots, M$ and, as smaller values of δ are chosen, $h(\sigma_m)$ behaves very much as σ_m, so that the original objective function and its corresponding modified version are very close in the feasible region. Particularly, as $\delta \to 0$, function $\left|K_{\eta}^{*}\right|_{p}$ converges pointwise to $|K_{\eta}|_{p}$. Besides, by considering that $\forall\sigma > 0$, $\lim_{\delta\to 0} h'(\sigma) = 1$ and $\lim_{\delta\to 0} h^{(n)}(\sigma) = 0$, for $n \geq 2$, it is easy to prove that the derivatives of this objective function verify the same property of convergence. As a result of these considerations, it may be concluded that the positions of q that minimize original and modified objective functions are nearly identical when δ is *small*. Actually, the value of δ is selected in terms of point q under consideration, making it as small as possible and in such a way that the evaluation of the minimum of modified functions does not present any computational problem.

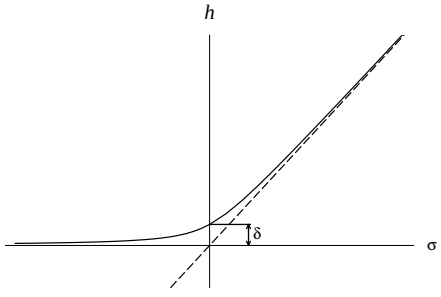


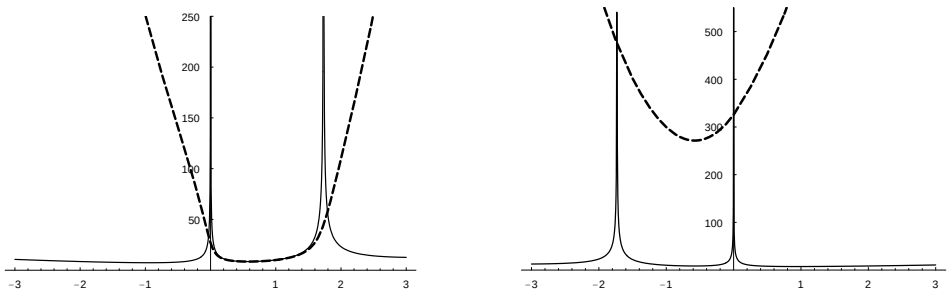
Figure 2: Representation of function $h(\sigma)$.

Suppose that $int\ P = \emptyset$, then the original objective function, $|K_\eta|_p$, is not suitable for our purpose because it is not correctly defined. Nevertheless, the modified function is well defined and tends to solve the tangle. We can reason it from a qualitative point of view by considering that the dominant terms in $\left|K_\eta^*\right|_p$ are those associated to the tetrahedra with more negative values of σ and, therefore, the minimization of these terms imply the increase of these values. It must be remarked that $h(\sigma)$ is an increasing function and $\left|K_\eta^*\right|_p$ tends to ∞ when the volume of any tetrahedron of $N(q)$ tends to $-\infty$, since $\lim_{\sigma \to -\infty} h(\sigma) = 0$.

In conclusion, by using the modified objective function, we can untangle the mesh and, at the same time, improve its quality. Obviously, the modification here proposed can be easily applied to other objective functions. An implementation of the simultaneous untangling and smoothing procedure for an equilateral reference tetrahedron is freely available in [12].

For a better understanding of the behavior of the objective function and its modification, we propose the following 2-D test example. The objective functions of this example have been derived from the mean ratio quality metric for triangles. Let us consider a simple 2-D mesh formed by three triangles, qBC, qCA and qAB, where we have fixed $A(0,-1)$, $B(\sqrt{3},0)$, $C(0,1)$ and $q(x,y)$ is the free node. In this case, the feasible region is the interior of the equilateral triangle ABC. In Fig. 3(a) we show $|K_\eta|_2$ (solid line) and $\left|K_\eta^*\right|_2$ (dashed line) for as a function of x for a fixed value $y = 0$ (the y-coordinate of the optimal solution). The chosen parameter δ is 0.1. We can see that the original objective function presents several local minima and discontinuities, opposite to the modified one. Besides, the original function reaches its absolute minimum outside the feasible region. Vertical asymptotes in the original objective function correspond to positions of the free node for which $\sigma = 0$ for any tetrahedra of the local mesh. As it might be expected, the optimal solution for the modified function results in $q(\sqrt{3}/3, 0)$. Both functions are nearly identical in the proximity of this point, see Fig. 3(a).

Let us now consider the tangled mesh obtained by changing the position of point $B(\sqrt{3},0)$ to $B'(-\sqrt{3},0)$. Here, the mesh is constituted by the triangles $qB'C$, qCA and qAB', where $qB'C$ and qAB' are inverted. The feasible region does not exist in this new situation. The graphics of functions $|K_\eta|_2$ and $\left|K_\eta^*\right|_2$ are represented in Fig. 3(b). Although the mesh cannot be untangled, we get $q(-\sqrt{3}/3, 0)$ as the optimal position of the free node by using our modified objective function. For this position the three triangles are "equally



(a) (b)

Figure 3: (a) Transversal cut of $|K_\eta|_2$ (solid line) and $\left|K_\eta^*\right|_2$ (dashed line) for the 2-D test example; (b) the same objective functions for the tangled mesh.

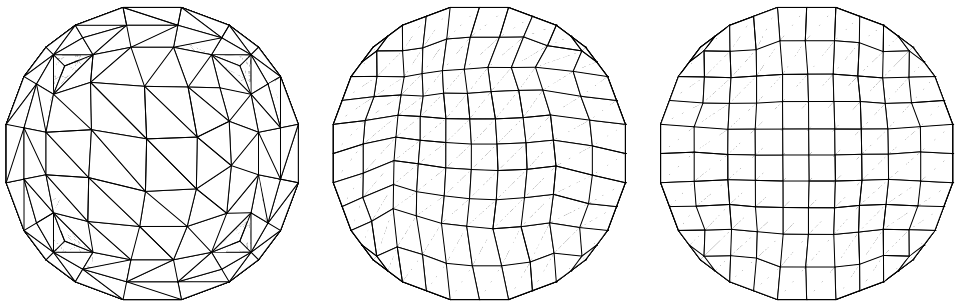
inverted", that is, they have the same negative values of σ.

The next example highlights the difference between relocating the nodes of $\mathcal{T}$ by choosing T_I equal to the equilateral tetrahedron (this option would be the best if we are only interested in an isotropic FEM mesh) and, as we point in section 2.3.1, equal to the counterpart tetrahedron in $\mathcal{C}_K$. The test example is a rough sphere obtained by deforming a uniform Delaunay triangulation of a cube. The nodes of the cube surface have been projected onto the sphere and the inner nodes have been relocated according to the above mentioned criteria. Figure 4 shows the central section of the sphere after mapping a uniform grid defined on the cube. The relocation of the inner nodes has been carried out by taking T_I as an equilateral tetrahedron (b), and taking T_I as the counterpart in the cube (c). It is clear that the second option constructs a better parameterization.

2.3.3. Rearrangement of the inner nodes

The computational effort to optimize a mesh depends on the initial position of the nodes. An arrangement of the nodes close to their optimal positions significantly reduces the number of iterations (and the CPU time) required by the untangling and smoothing algorithm. Therefore, an interesting idea is to construct a rough approximation of the solid and to use the corresponding parametrization to relocate interior nodes of more accurate subsequent approximations.

Taking into account that the grade of refinement attained by the tetrahedral mesh depends on the maximum allowed distance, ε, between $\mathcal{T}_S^*$ and

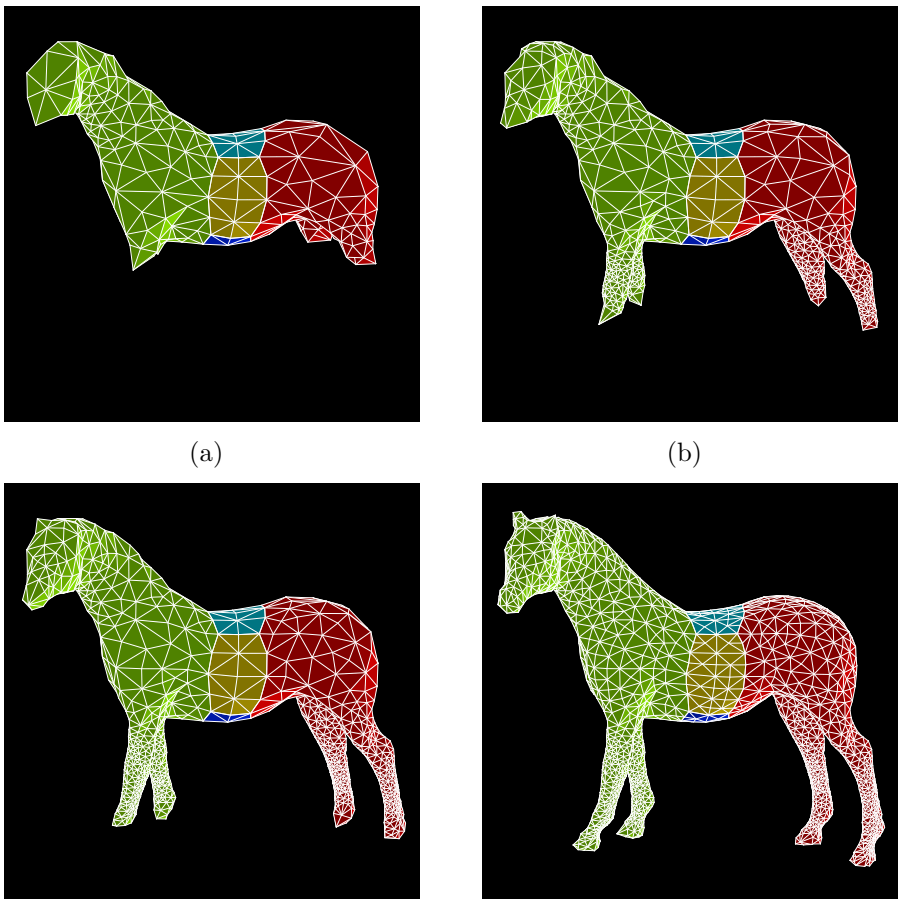


(a) (b) (c)

Figure 4: Rough tetrahedral mesh of a sphere (a). Central cross section of the sphere after mapping a uniform grid with a parameterization obtained by (b) choosing the target tetrahedron equilateral, and (c) choosing the target tetrahedron equal to the counterpart in the cube.

$\mathcal{T}_S$, we will write $\mathcal{C}_K(\varepsilon)$, $\mathcal{T}(\varepsilon)$, $\tau_K(\varepsilon)$ and $\mathcal{T}_S^*(\varepsilon)$ to express this dependence.

Let suppose that $\Pi_{\varepsilon_i} : \mathcal{C}_K(\varepsilon_i) \rightarrow \mathcal{T}(\varepsilon_i)$ is the volumetric parameterization for a given tolerance ε_i. We want to find the approximate location of the nodes of a more accurate mesh $\mathcal{T}(\varepsilon_{i+1})$, assuming that $\varepsilon_i > \varepsilon_{i+1}$. Firstly, the mesh $\mathcal{C}_K(\varepsilon_i)$ is locally refined until the distance between $\mathcal{T}_S^*(\varepsilon_i)$ and $\mathcal{T}_S$ is below ε_{i+1}. In that moment we have the new mesh of the cube $\mathcal{C}_K(\varepsilon_{i+1})$. Afterward, their inner nodes are mapped by using the previous parameterization, that is, we construct the new tetrahedral mesh $\mathcal{T}^*(\varepsilon_{i+1})$ after Π_{ε_i}-mapping the nodes of $\mathcal{C}_K(\varepsilon_{i+1})$. Note that $\mathcal{T}^*(\varepsilon_{i+1})$ has the same topology as $\mathcal{T}(\varepsilon_{i+1})$, but their nodes are not located at optimal positions. Although $\mathcal{T}^*(\varepsilon_{i+1})$ could be tangled, their interior nodes are close to their final positions. Therefore, the computational effort to optimize the mesh is drastically reduced. The last step of this iteration consists on relocating the inner nodes of $\mathcal{T}^*(\varepsilon_{i+1})$ in their optimal position following the mesh smoothing and untangling procedure above described. This sequence is repeated several times until we achieve the desired tolerance. In Fig. 5 it is shown a sequence of gradual approximations to the mesh of a horse. The initial surface triangulation $\mathcal{T}_S$ has been obtained from the Large Geometric Model Archives at Georgia Institute of Technology.



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

Figure 5: Gradual approximations: from a initial coarse mesh to the final accurate mesh.

3. Representation of the solid by T-splines

3.1. Construction of an adapted volumetric T-mesh

We will start this section with a short introduction on T-splines. A detailed report about T-splines and their relationship with isogeometric analysis can be found in [2].

The T-mesh is the control grid of the T-splines. In 3-D it is a division of a rectangular prism forming a grid in which the T-juntions are allowed. In 2-D T-junctions are inner vertices of the grid connecting 3 edges. T-junctions in 3-D are inner vertices shared by one edge in some direction and

two edges in other directions at the same time [28]. T-splines are rational spline functions defined by local knot vectors, which are inferred from certain points of the T-mesh known as *anchors* [2]. The anchors of the odd-degree T-splines are sited on the vertices of the T-mesh and the anchors of the even-degree T-splines are located in the center of each prism. We will focus on odd-degree T-splines and, in particular, on cubic T-splines because they are the ones implemented in the present work. Cubic T-splines have 5 knots in each parametric direction. Let us consider the 2-D example of Fig. 6 to understand how the knot vectors are deduced from the anchor. The parametric coordinates of the anchor $\mathbf{t}_\alpha$ in Fig. 6(a) are given by (ξ_4^1, ξ_4^2), then, by examining the intersections of horizontal and vertical lines (red lines in the figure) with the edges of the T-mesh, we deduce that the the knot vector in ξ^1 direction is $\Xi_1^\alpha = (\xi_1^1, \xi_2^1, \xi_4^1, \xi_5^1, \xi_6^1)$ and, the knot vector in ξ^2 direction is $\Xi_2^\alpha = (\xi_2^2, \xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2)$. In the case of Fig. 6(b) only one edge is found when marching horizontally from $\mathbf{t}_\beta$ to the right. In such situations we have two possibilities: repeat knots in order to form a clamped local knot vector or, as we have implemented in our work, add phantom knots and form an unclamped one. These phantom knots are placed following the pattern shown in Fig. 6(b). The construction of knot vector in 3-D is analogous but we must examine the intersections with T-mesh faces encountered when marching in each space direction. The points of the parametric domain are written as $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)$.

A T-spline is a rational function from the parametric domain to the physical space given by

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha(\boldsymbol{\xi}) \tag{9}$$

where $\mathbf{P}_\alpha$ is the control point corresponding to the α-th blending function

$$R_\alpha(\boldsymbol{\xi}) = \frac{w_\alpha B_\alpha(\boldsymbol{\xi})}{\sum_{\beta \in A} w_\beta B_\beta(\boldsymbol{\xi})} \tag{10}$$

being w_α its weight and $B_\alpha(\boldsymbol{\xi}) = N_\alpha^1(\xi^1) N_\alpha^2(\xi^2) N_\alpha^3(\xi^3)$ the product of univariate B-splines. In these expressions $A \subset \mathbb{Z}^3$ represents the index set containing every α such that $\mathbf{t}_\alpha$ is an anchor.

The T-spline $\mathbf{S}(\boldsymbol{\xi})$ is the sum of rational C^2 blending functions, so it is also a C^2 function. Nevertheless, as the surface of the solid is the union of six patches obtained by mapping the six faces of the cube, and these faces match

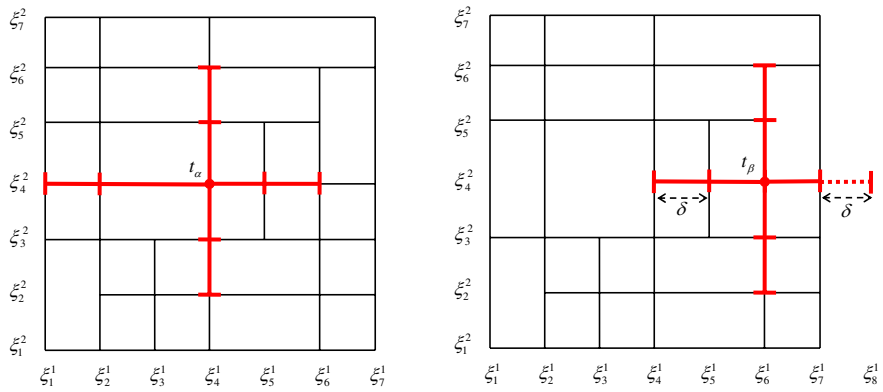


Figure 6: Construction of knot vector in a two-dimensional T-mesh. All the knots associated to the anchor $\mathbf{t}_\alpha$ lie inside the T-mesh (a). The phantom knot ξ_8^1 has been added to construct an unclamped local knot vector (b).

with C^0 continuity, we only can assure the C^0 continuity for the surface of the solid.

Our objective is to get a representation of the solid suitable for isogeometric analysis by means of trivariate T-splines. This representation, $\mathcal{V}$, must preserve the features and details of the input data, the triangulation $\mathcal{T}_S$. To do that, we construct an adapted T-mesh by partitioning the parametric domain $\mathcal{C}$ in cells by using an octree subdivision. The unitary cube $\mathcal{C}$ is divided in 8 identical cells and, each cell is, in turn, divided in other 8 cells and so on, until all the cells of the octree do not contain any node of $\mathcal{C}_K$ in their inner. This last is possible due to the particular characteristics of the Kossaczky subdivision scheme, in which the edges of $\mathcal{C}_K$ are the result of successive division of the edges of $\mathcal{C}$ by two. The octree partition defines a T-mesh, $\mathcal{C}_T$, that is used to determine the local knot vector and the anchors of the T-splines. Note that all the nodes of $\mathcal{C}_K$ are vertices of $\mathcal{C}_T$, so it is to be hoped that the surface of $\mathcal{V}$ achieves the same resolution than the input triangulation $\mathcal{T}_S$. Another consequence of the proposed octree subdivision is that the cell faces of $\mathcal{C}_T$ contain no more than one inner T-junction.

3.2. Interpolation

Basically there are two ways of fitting splines to a set points: interpolation and approximation. We have adopted the first one because it is more

appropriated to reproduce all features of the input triangulation. Assuming that the set of blending functions are linearly independent, we need as many interpolation points as blending functions

Recently Buffa et al. [5] have analyzed the linear independence of the bicubic T-spline blending functions corresponding to some particular T-meshes. They prove linear independence of hierarchical 2-D T-meshes generated as the refinement of a coarse and uniform T-mesh (this is the 2-D counterpart to our case). However, the extension of these results to 3-D is not straightforward.

We have chosen the images of the anchors as interpolation points, and all the weights have been taken equal to 1. Thus, the control points, $\mathbf{P}_\alpha$, are obtained by solving the linear system of equation

$$\Pi\left(\mathbf{t}_\beta\right) = \mathbf{S}\left(\mathbf{t}_\beta\right) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha\left(\mathbf{t}_\beta\right), \ \forall \mathbf{t}_\beta, \ \beta \in A \tag{11}$$

where the images $\Pi\left(\mathbf{t}_\beta\right)$ have been calculated through the volumetric parameterization (3).

The linear independence has become evident in all the applications considered until now, as the resolution of (11) is only possible if the blending functions are linearly independent.

In Fig. 7 the interpolation points and their associated control points for the Stanford bunny of Fig. 11 are shown.

(a) (b)

Figure 7: The interpolation points (a), and their corresponding control points for the Stanford bunny (b).

4. Results

4.1. Test examples

We have chosen a sphere as first example in order to clarify how the proposed technique works. The figures 8 (b) and (d) show the T-mesh and its transformation by (9) to the physical space. Comparing figures 8 (a) and 8 (b) it is noticed that all the nodes of the triangulation $\mathcal{C}_K$ are also nodes of the T-mesh, as it was said in 3.1.

In Fig. 8(e) we have highlighted four critical (red) points in the corners included in nearly degenerated cells. The existence of degenerated cells is inherent in this type of parameterization, because a rectangular cell of $\mathcal{C}_T$, placed in an edge or in a corner of the cube, is transformed into another cell with 6 or 7 vertices in the surface of the sphere. So, we could have problems if the external faces of these cells are mapped to non-convex surfaces. In such a case, the Jacobian of the transformation will be negative. We could mitigate the volume which is affected by these degenerated cells by refining the T-mesh around the edges and corners of the cube. This question will be analyzed in the following test example of Fig. 10.

On the other hand, if the type of parameterization is polar-like, the singularity appears in the origin of the coordinate system, as it is pointed out by [23]. They propose a method blending both types of parameterization where the user has control over the placement of these critical points.

At present, there are no quality metrics for isogeometric analysis analogous to the ones for traditional FEA to help us characterize the impact of the mesh on analysis, as it is indicated in [9]. Xu et al. [31, 32] give sufficient conditions for getting both an injective parameterization for planar splines without self-intersections and an isoparametric net of good uniformity and orthogonality, but there are not similar studies for T-splines.

One of the factors to take into account is the variation of the Jacobian in the elements. Usually, a large variation leads to poor accuracy in the numerical approximation, so we can explore the suitability of a T-spline for isogeometric simulations by analyzing the scaled Jacobian in the quadrature points of the cells. The scaled Jacobian, given by

$$J_s\left(\xi^1,\xi^2,\xi^3\right)=\frac{\det\left(\mathbf{S}_{\xi^1},\mathbf{S}_{\xi^2},\mathbf{S}_{\xi^3}\right)}{\|\mathbf{S}_{\xi^1}\|\,\|\mathbf{S}_{\xi^2}\|\,\|\mathbf{S}_{\xi^3}\|} \tag{12}$$

where $\mathbf{S}_{\xi^i}$ is the derivative of the trivariate T-spline (9) with respect to ξ^i, has been evaluated in the eight Gaussian quadrature points (see for exam-

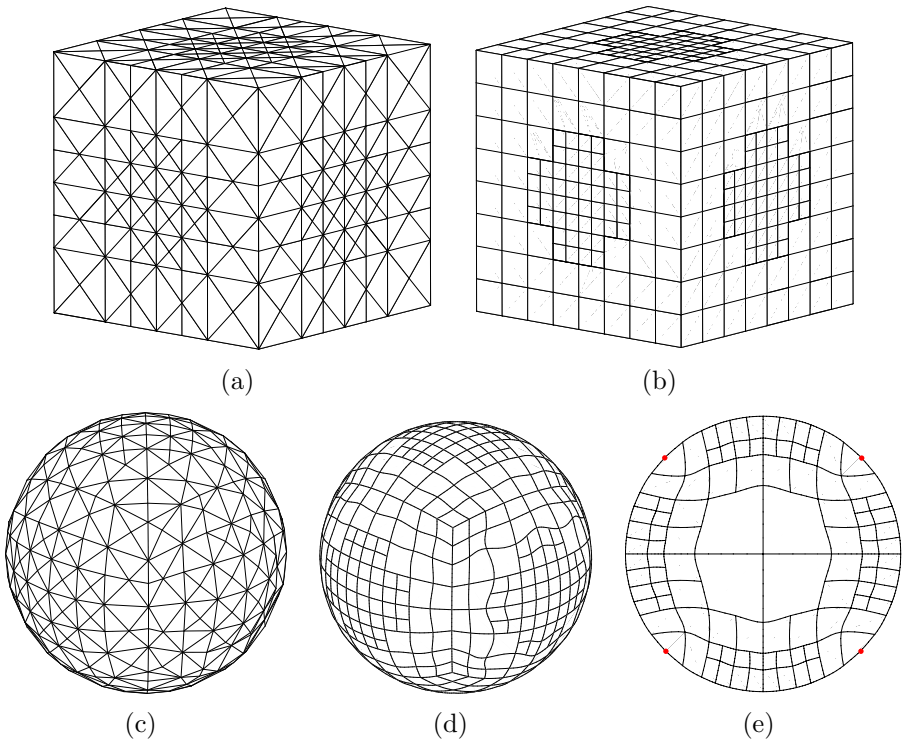


Figure 8: Parameterization and construction of a sphere with T-splines. Parametric domain $\mathcal{C}_K$ (a), T-mesh $\mathcal{C}_T$ (b), tetrahedral mesh of the sphere $\mathcal{T}$ (c), T-spline representation of the sphere $\mathcal{V}$ (d), and its cross section (e).

ple [6]) of each cell in the real domain. For doing that, we set the eight quadrature points in the hexahedra of the parametric domain and calculate their transformation to the real domain by applying (9). We can get an idea about whether the distortion introduced by the spline is or is not too large by plotting the average, minimum and maximum of the scaled Jacobian in the quadrature points. The following graphic shows the average, minimum and maximum of the scaled Jacobian for each cell sorted in increasing order of its average value.

The next test model (see Fig. 10) is a T-spline representation of a deformed unitary square in which the corner $(1, 1)$ has been displaced toward position $\left(\frac{3}{4} - \frac{1}{10}, \frac{3}{4} - \frac{1}{10}\right)$, producing a degenerated cell. This displacement makes the new optimal position for the central node to become $(0.38, 0.38)$. The same model is approximated by two T-meshes with 9 (Fig. 10(a)) and

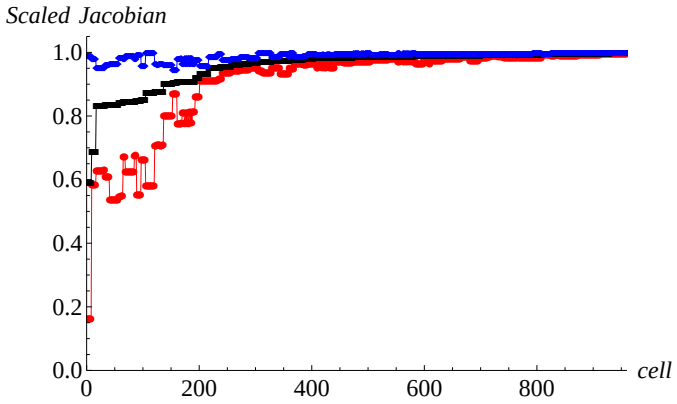


Figure 9: Scaled Jacobian for the sphere test of Fig. 8, evaluated in the quadrature points and sorted by the increasing order of the average value (black line). The red and blue lines correspond to the minimum and maximum values in each cell, respectively.

14 (Fig. 10(b)) interpolating points. The corresponding T-spline representations are shown in figures 10 (c) and 10 (d), respectively. Note that the representation of Fig. 10(c) has a wide folded region around the corner in which the Jacobian is negative. However, this region has been remarkably reduced in the refined version (Fig. 10(d)). This example points up that, although the refinement of the T-mesh around the corners (and edges in 3-D) does not completely solve the problem of degenerated cells, it tends to diminish the region in which the Jacobians become negative. It can be more clearly seen in figures 10 (e) and (f), where the scaled Jacobian has been represented by a color map. The blue hues correspond to the regions in which the Jacobian is negative.

4.2. Applications to complex solids

In Fig. 11 we have shown a tetrahedral and T-spline representation of the Stanford bunny. Note how similar discretization of the respective parametric domains give rise to similar grade of detail in the physical domains. It can be seen how the isoparemetric curves are nearly orthogonal in most parts of the solid, which entails low distortion and scaled Jacobians near to unit. Nevertheless, in some regions of the surface, and specially those close to the seams, the distortion becomes high. We have computed 39 cells out of 9696 in which at least one of the eight Gaussian quadrature points has negative Jacobian.

As we mentioned in the previous section, the T-spline is enforced to interpolate all the nodes of the tetrahedral mesh $\mathcal{T}$ and this mesh is as close as we want to the input surface $\mathcal{T}_S$. Moreover, the interpolating points are exactly sited on the input surface. These reasons suggest a good accuracy between the surface of the T-spline and $\mathcal{T}_S$. In order to estimate the gap between both surfaces we have analyzed the differences between the volumes enclosed by $\mathcal{T}_S$ and the T-spline, $\mathcal{V}$, for the present application. The first volume is measured by applying the divergence theorem and the second one is calculated integrating $\det(\mathbf{S}_{\xi^1}, \mathbf{S}_{\xi^2}, \mathbf{S}_{\xi^3})$ in the unitary cube $\mathcal{C}$ with 8 Gaussian quadrature points in each cell. The quadrature points with negative Jacobians have been rejected from the calculations. The results for the bunny application are: the volume enclosed by $\mathcal{T}_S$ is 754.9, the volume of $\mathcal{T}$ is 750.9 (a difference of 0.5% in relation to $\mathcal{T}_S$) and the volume of $\mathcal{V}$ is 757.4 (a difference of 0.3% in relation to $\mathcal{T}_S$).

Guided by the results of the second test example of Fig. 10, we are interested in knowing the effect of refining the cells with worse quality. To do that, we develop an iterative procedure in which the scaled Jacobian is evaluated in the center of each cell and, if it is negative, we store the point in a list of vertices to be included in the T-mesh of the subsequent iteration. If the impact of such refinement is similar to the one of the test example, it should be hoped a reduction of the region in which the Jacobian had negative values. In fact, the number of cells with negative scaled Jacobian evaluated in their centers have been: 5 in the first iteration, 4 in the second, 2 in the third and 0 in the fourth. Moreover, only 6% of the cells have a scaled Jacobian less than 0.5.

The sequence of images (figures 12, 13 and 14) summarizes the main stages to get a T-spline representation of a solid given by a triangulation. The Fig. 12(a) shows the input triangulation emphasizing the six patches in which it has been divided. Each patch is mapped to the corresponding face of the cube by using Floater parameterization, Fig. 12(b). Afterward, we construct an adapted tetrahedral mesh of the cube by using Kossaczky refinement (Fig. 12(c)) and map its boundary to the true surface (Fig. 13(a)). Then, we relocate the inner nodes in their optimal positions by means of our untangling and smoothing procedure (Fig. 13(b)). Fig. 13(c) represents the resulting tetrahedral mesh. Finally, we generate an adapted T-mesh of the cube (Fig. 14(a)) and map the anchors to the positions ruled by the volumetric parameterization in order to obtain the T-spline representation of the solid (Fig. 14(b)).

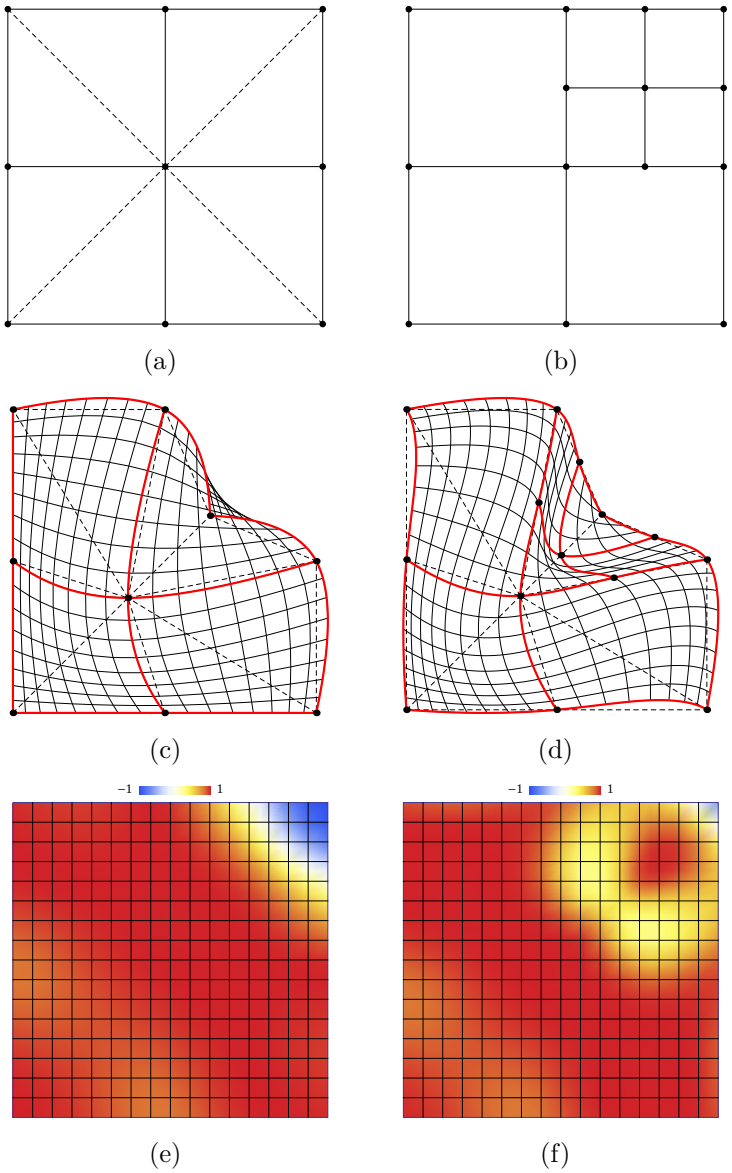


Figure 10: Initial T-mesh in the parametric domain (continuous line) and the underlying triangular mesh (dashed line) (a). T-mesh refined on the corner (b). T-spline of a deformed square with a reentering corner (c). Refined version of the T-spline (d). Scaled Jacobian representation in the parametric domain for the initial (e) and refined meshes (f).

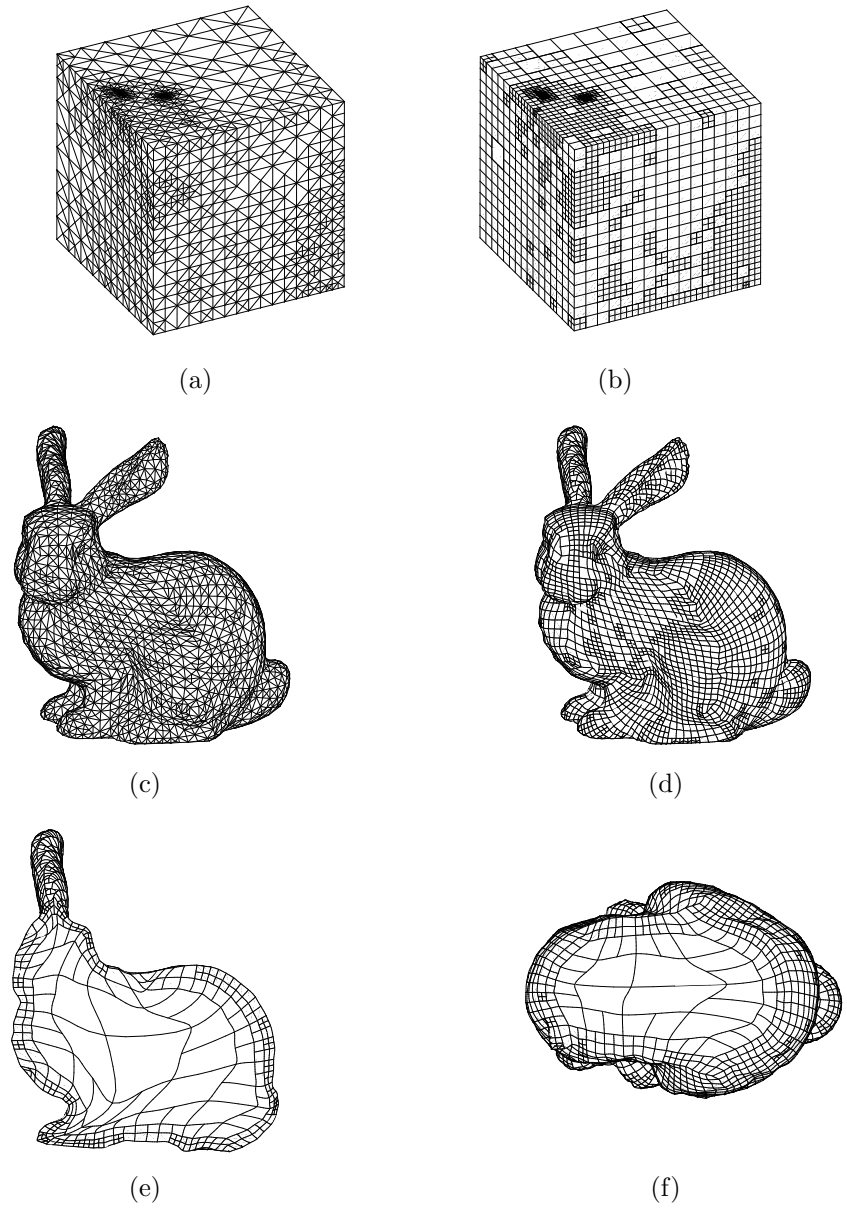
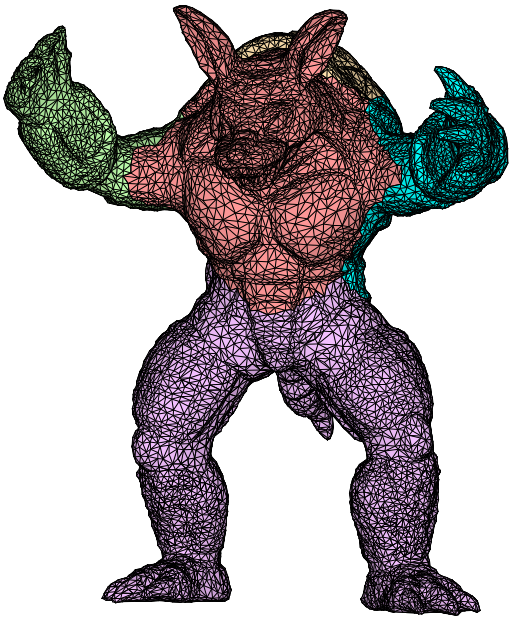
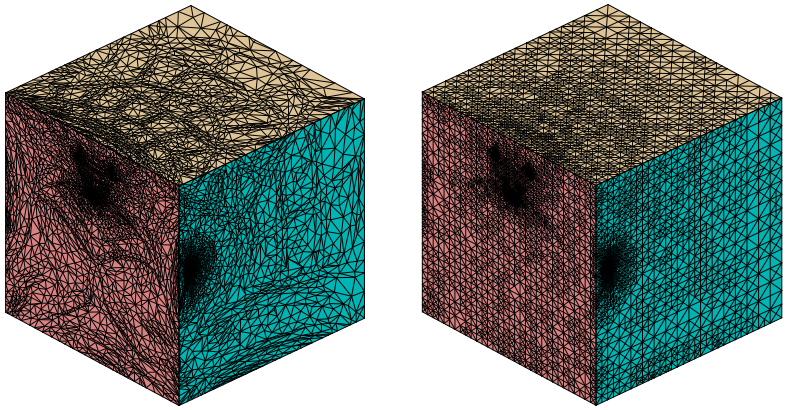


Figure 11: Tetrahedral mesh of the parametric domain $\mathcal{C}_K$ (a), T-mesh $\mathcal{C}_T$ (b), tetrahedral mesh $\mathcal{T}$ (c) and T-spline representation $\mathcal{V}$ of the Stanford bunny. Two transversal sections of $\mathcal{V}$ (d) and (e).

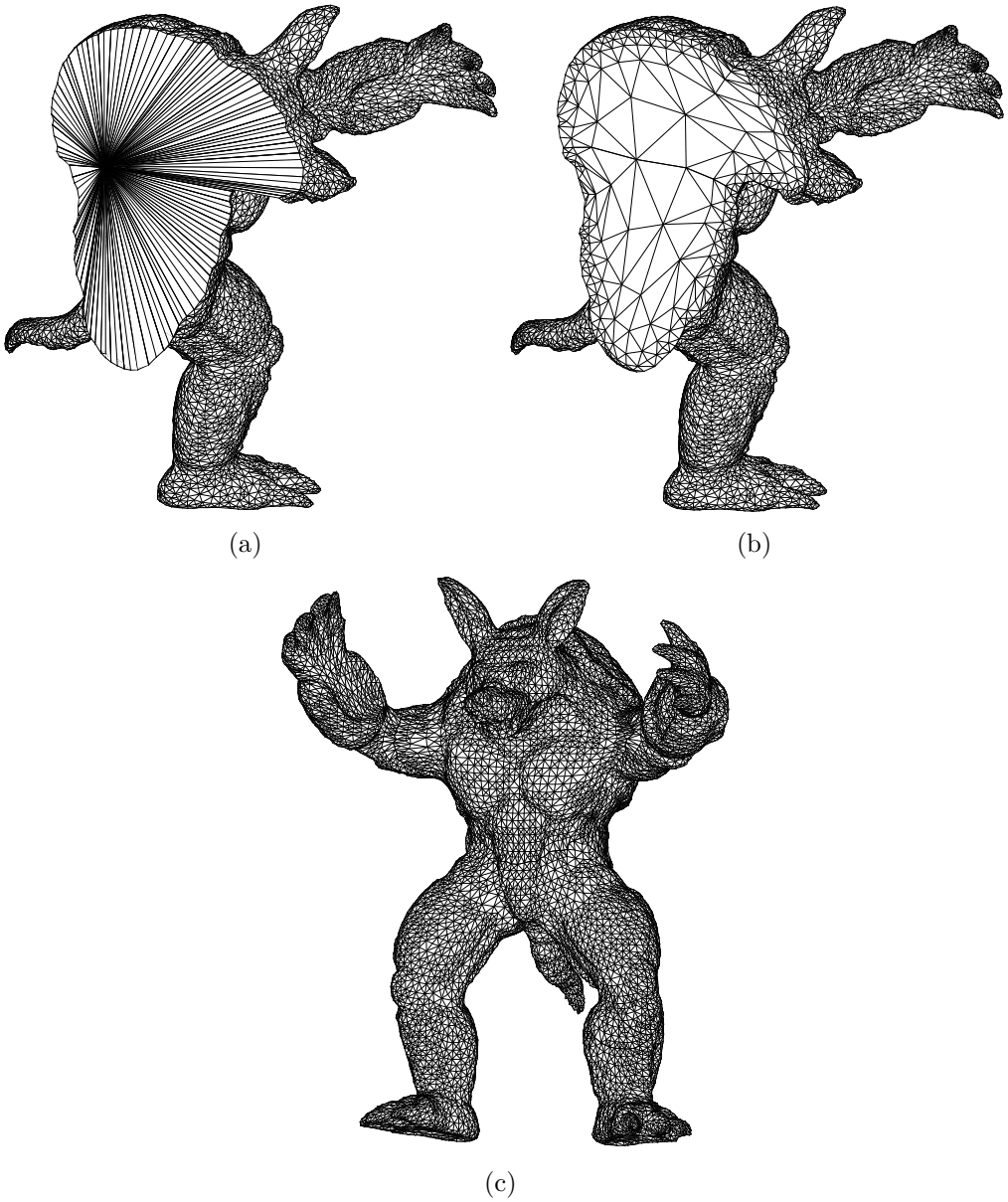


(a)



(b) (c)

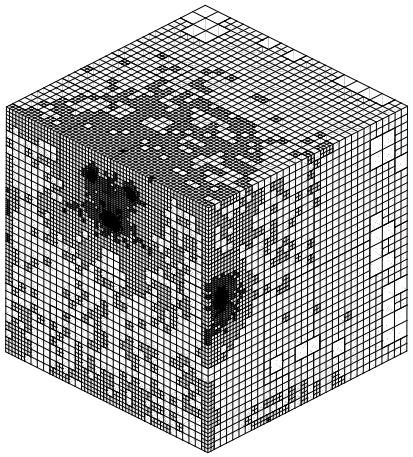
Figure 12: Stages of the volumetric parameterization (I). Input triangulation $\mathcal{T}_S$ divided into six patches (a). Floater parameterization τ_F of the armadillo triangulation onto the cube faces (b). Adapted tetrahedral mesh of the cube by using Kossaczky refinement $\mathcal{C}_K$ (c).



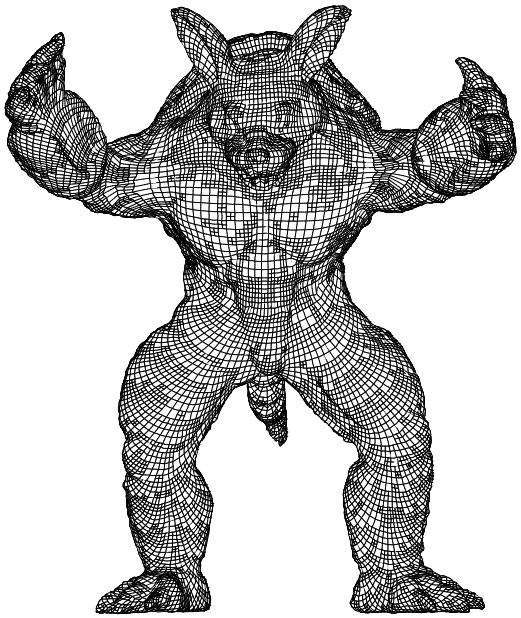
(a) (b)

(c)

Figure 13: Stages of the volumetric parameterization (II). Tangled tetrahedral mesh obtained after mapping the cube boundary to the armadillo surface (a). Cross section of the optimized tetrahedral mesh (b). Final tetrahedral mesh $\mathcal{T}$ (c).



(a)



(b)

Figure 14: Stages of the volumetric parameterization (III). Adapted T-mesh in the parametric domain $\mathcal{C}_T$ (a) and final T-spline representation of the armadillo $\mathcal{V}$ (b).

In this case, the number of cells with negative Jacobian evaluated in their centers has been 1 in the first iteration and 0 in the second. In this last iteration, the number of cells with scaled Jacobian less than 0.5 is 1.5%.

We remark that in these applications we have obtained positive Jacobians in all the centers of the cells of $\mathcal{C}_T$. Therefore, the most distorted cells are susceptible of being integrated with at least one Gaussian quadrature point. Obviously, a better numerical approximation is possible in most of the cells.

5. Conclusions and challenges

Focused on the application of isogeometric analysis, this work is a new approach to the automatic generation of trivariate T-splines representation of solids. Our procedure has been developed for genus-zero solids, but we believe that it could be generalized to arbitrary objects. The key lies on having a volumetric parameterization of the solid. In [24, 25] we pointed the way to construct volumetric parameterizations of solids delimited by surfaces of genus greater than zero. The method is based on the composition of a *meccano*, joining cuboid pieces in order to get a rough approximation of the solid. Afterward, we use a parameterization to map the boundary of the solid to the *meccano* faces. In the *meccano* method such surface parameterization must be explicitly given, but we think that this handicap could be overcome by applying a technique similar to PolyCube-Maps [20, 29, 30]. The rest of the procedure would follow the same steps than the ones described in this paper.

Furthermore, the input data in the present paper is a triangulation defining the boundary of the solid, but this boundary is generally described by CAD. Such information could be used to map the points lying on the surface of the parametric mesh to the surface of the solid, making unnecessary the stage of surface parameterization.

In general, the distortion introduced by the proposed volumetric parameterization is low, but the existence of critical points where the Jacobian of the T-spline may become negative constitutes an inconvenient for isogeometric simulations. Just as we have shown in section 4, the selective refinement of the most degenerated cells palliates the problem, but it cannot be understood like a conclusive solution. As far as we know, this is a problem common to other types of parameterizations that must be satisfactory solved.

Acknowledgements

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, for their support.

References

[1] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, "Nonlinear Programing: Theory and Algorithms", John Wiley and Sons Inc., New York, 1993.

[2] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott and T.W. Sederberg, "Isogeometric analysis using T-splines", Comput. Meth. Appl. Mech. Eng., 199, 229-263, 2010.

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

[4] H. Borouchaki and P.J. Frey, "Simplification of surface mesh using Hausdorff envelope", Comput. Meth. Appl. Mech. Eng., 194, 4864-4884, 2005.

[5] A. Buffa, D. Cho and G. Sangalli, "Linear independence of the T-spline blending functions associated with some particular T-meshes", Comput. Methods Appl. Mech. Engrg., 199, 14371445, 2010.

[6] G.F. Carey and J.T. Oden, "Finite Elements, A Second Course", Prentice-Hall, New Jersey, 1982.

[7] J.M. Cascón, R. Montenegro, J.M. Escobar, E. Rodríguez and G. Montero, "A new *meccano* technique for adaptive 3-D triangulation" in: "Proceedings of 16th International Meshing Roundtable", Springer, Berlin, 103-120, 2007.

[8] J.M. Cascón, R. Montenegro, J.M. Escobar, E. Rodríguez and G. Montero, "The meccano method for automatic tetrahedral mesh generation

of complex genus-zero solids" in: "Proceedings of 18th International Meshing Roundtable", Springer-Verlag, Berlin, 463-80, 2009.

[9] E. Cohen, T. Martin, R.M. Kirby, T. Lyche and R.F. Riesenfeld, "Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis", Comput. Meth. Appl. Mech. Eng., 199, 334-356, 2010.

[10] J.A. Cottrell, T.J.R. Hughes and Y. Bazilevs, "Isogeometric Analysis: Towad Integration of CAD and FEA", John Wiley & Sons, Chichester, 2009.

[11] 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.

[12] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero and J.M. González-Yuste, "SUS Code − Simultaneous mesh untangling and smoothing code", www.dca.iusiani.ulpgc.es/proyecto2008−2011, 2010.

[13] M.S. Floater, "Parametrization and smooth approximation of surface triangulations", Comput. Aid. Geom. Design, 14, 231-250, 1997.

[14] M.S. Floater, "Mean Value Coordinates", Comput. Aid. Geom. Design, 20, 19-27, 2003.

[15] M.S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey, Advances in Multiresolution for Geometric Modelling", Mathematics and Visualization. Springer, Berlin, 157-186, 2005.

[16] K. Hormann, Y. Lévy and A. Sheffer, "Mesh Parameterization: Theory and practice", SSIGGRAPH07: ACMSIGGRAPH2007 courses. ACM Press, NewYork, NY, USA, 2007.

[17] P.M. Knupp, "Algebraic Mesh Quality Metrics", SIAM J. Sci. Comput., 23, 193-218, 2001.

[18] I. Kossaczky, "A Recursive Approach to Local Mesh Refinement in Two and Three Dimensions", J. Comput. Appl. Math., 55, 275-288, 1994.

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

[20] J. Lin, X. Jin, Z. Fan and C.C.L. Wang, "Automatic PolyCube-Maps", Lecture Notes in Computer Science, 4975, 3-16, Springer, Berlin, 2008.

[21] B. Li, X. Li, K. Wang, "Generalized PolyCube trivariate splines ", SMI 2010 - International Conference on Shape Modeling and Applications, 261-265, 2010.

[22] T. Martin, E. Cohen and R.M. Kirby, "Volumetric parameterization and trivariate B-spline fitting using harmonic functions", Computer Aided Geometric Design, 26, 648-664, 2009.

[23] T. Martin and E. Cohen, "Volumetric parameterization of complex objects by respecting multiple materials", Computers and Graphics 34, 187-197, 2010.

[24] R. Montenegro, J.M. Cascón, J.M. Escobar, E. Rodríguez and G. Montero, "An Automatic Strategy for Adaptive Tetrahedral Mesh Generation", Applied Numerical Mathematics, 59, 2203-2217, 2009.

[25] R. Montenegro, J.M. Cascón, E. Rodríguez, J.M. Escobar and G. Montero, "The meccano method for automatic 3-D triangulation and volume parametrization of complex solids", Computational Science, Engineering and Technology Series, 26, 19-48, 2010.

[26] L. Piegl and W. Tiller, "The NURBS Book", Springer-Verlag, New York, 1997.

[27] T.W. Sederberg, J. Zheng, A. Bakenov and A. Nasri, "T-splines and T-NURCCSs", ACM Trans. Graph. 22 (3) 477-484 2003.

[28] W. Song and X. Yang, "Free-form deformation with weighted T-spline", The Visual Computer, 21, 139-155, 2005.

[29] M. Tarini, K. Hormann, P. Cignoni and C. Montani, "Polycube-maps", ACM Trans. Graph., 23, 853-860, 2004.

[30] H. Wang, Y. He, X. Li, X. Gu and H. Qin, Polycube Splines, Computer Aided Geometric Design, 40, 721-733, 2008.

[31] G. Xu, B. Mourrain, R. Duvigneau and A. Galligo, "Optimal analysis-aware parameterization of computational domain in isogeometric analysis", Lecture Notes in Computer Science, 6130, 236-254, 2010.

[32] G. Xu, B. Mourrain, R. Duvigneau and A. Galligo, "Parametrization of computational domain in isogeometric analysis: methods and comparison", Inria-00530758, 1-29, 2010.