The Meccano Method for Isogeometric Solid Modeling

J.M. Escobar[1], J.M. Cascón[2], E. Rodríguez[1] and R. Montenegro[1]

[1] University Institute for Intelligent Systems and Numerical Applications in Engineering, SIANI, University of Las Palmas de Gran Canaria, Spain, {jmescobar,erodriguez,rmontenegro}@siani.es, http://www.dca.iusiani.ulpgc.es/proyecto2008-2011

[2] Department of Economics and History of Economics, Faculty of Economics and Management, University of Salamanca, Spain, casbar@usal.es

Summary. We present a new method to construct a trivariate T-spline representation of complex solids for the application of isogeometric analysis. The proposed technique only demands the surface of the solid as input data. The key of this method lies in obtaining a volumetric parameterization between the solid and a simple parametric domain. To do that, an adaptive tetrahedral mesh of the parametric domain is isomorphically transformed onto the solid by applying the meccano method. The control points of the trivariate T-spline are calculated by imposing the interpolation conditions on points situated 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.

Key words: Trivariate T-spline, isogeometric analysis, volumetric parameterization, mesh optimization and 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 [20, 22, 23, 24, 31].

For example, Li et al. [20] 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. [23, 24] 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 [24] 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 [27]) 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 [28] 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 [28]).

In this paper we present a new method [13] 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 by using the meccano method [7, 8, 25, 26]. 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.

In [25], we introduced the meccano method 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, i.e. the meccano. In this case, the construction of a T-spline representation also demands a T-mesh adapted to the discretization of the meccano. In this paper, we introduce a way to undertake this task by an octree subdivision of a cube enclosing a polycube decomposition of the meccano. 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 [21, 30, 31].

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, 13, 25, 26], 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 of a genus-zero solid consists of 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, 26] 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 [14, 15]. 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 [16, 17]), 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 us consider $\mathcal{C}_K$ is a tetrahedral mesh of $\mathcal{C}$ resulting after applying several local bisections of the Kossaczky algorithm [19] to an initial mesh formed by six tetrahedra (see Fig. 1(a)). Three consecutive global bisections are presented in Figures 1(b), 1(c) and 1(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), 1(b) and 1(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

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

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, it 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

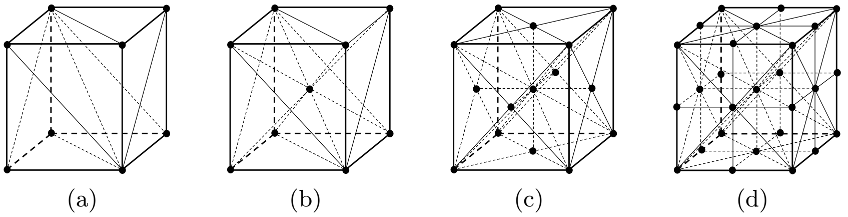


Fig. 1. Refinement of a cube by using Kossaczky's algorithm: (a) cube subdivision into six tetrahedra, (b) bisection of all tetrahedra by inserting a new node in the cube main diagonal, (c) new nodes in diagonals of cube faces and (d) global refinement with new nodes in cube edges.

$\Pi_K(T)$ 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. 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.

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. 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 what is 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 $\mathcal{T}$.

Objective Functions

Several tetrahedron shape measures could be used to construct an objective function. Nevertheless, those obtained by algebraic operations [18] 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 us 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 $A = \mu R W$, 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.

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 $\delta = h(0)$. Thus, the new objective function here proposed is given by

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

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. With this modification, we can untangle the mesh and, at the same time, improve its quality. An implementation of the simultaneous untangling and smoothing procedure for an equilateral reference tetrahedron is freely available in [12].

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 $\mathcal{T}_S$, 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 $\Pi_{\varepsilon_i} : \mathcal{C}_K(\varepsilon_i) \to \mathcal{T}(\varepsilon_i)$ 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. 2 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.

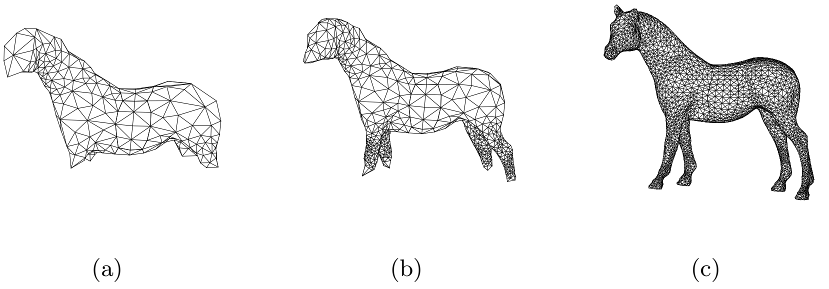


Fig. 2. Gradual approximations: from a coarse mesh to the final accurate mesh.

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 [29]. 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 situated 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. 3 to understand how the knot vectors are deduced from the anchor. The parametric coordinates of the anchor $\mathbf{t}_\alpha$ in Fig. 3(a) are given by $\left(\xi_4^1, \xi_4^2\right)$, 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 $\Xi_1^\alpha = \left(\xi_1^1, \xi_2^1, \xi_4^1, \xi_5^1, \xi_6^1\right)$ and, the knot vector in ξ^2 direction is $\Xi_2^\alpha = \left(\xi_2^2, \xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2\right)$. In the case of Fig. 3(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. 3(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 $\boldsymbol{\xi} = \left(\xi^1, \xi^2, \xi^3\right)$.

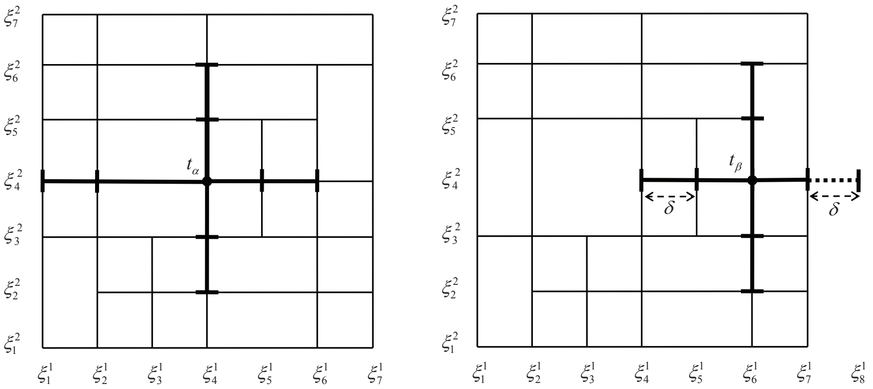
A T-spline is a rational function from the parametric domain to the physical space given by

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

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

$$R_\alpha \left(\boldsymbol{\xi}\right) = \frac{w_\alpha B_\alpha \left(\boldsymbol{\xi}\right)}{\sum\limits_{\beta \in A} w_\beta B_\beta \left(\boldsymbol{\xi}\right)} \tag{10}$$

being w_α its weight and $B_\alpha \left(\boldsymbol{\xi}\right) = N_\alpha^1 \left(\xi^1\right) N_\alpha^2 \left(\xi^2\right) N_\alpha^3 \left(\xi^3\right)$ 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.



(a) (b)

Fig. 3. 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).

The T-spline $\mathbf{S}\left(\boldsymbol{\xi}\right)$ 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 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 appropriate for reducing 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 equations

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

where the images $\mathit{\Pi}\left(\mathbf{t}_\beta\right)$ 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.

4 Results

4.1 Test Example

We have chosen a 2-D domain as first example in order to dicuss how the proposed technique works.

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. [32, 33] 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

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

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 example [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 test model shows a procedure, based on local mesh refinement, to improve the scaled Jacobian values. The goal is to reach values of the scaled Jacobian close to one in most parts of the solid.

The test model (see Fig. 4) 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 degenerate 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. 4(a)) and 14 (Fig. 4(d)) interpolating points. The corresponding T-spline representations are shown in Figures 4(b) and 4(e), respectively. Note that the representation of Fig. 4(b) 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. 4(e)). This example indicates that, although the refinement of the T-mesh around the corners (and edges in 3-D) does not completely solve the problem of degenerate cells, it tends to diminish the region in which the Jacobians become negative. It can be more clearly seen in Figures 4(c) and 4(f), where the scaled Jacobian has been represented by a color map. The dark colors correspond to the regions in which the Jacobian is negative.

4.2 Solid With Surface of Genus Zero

In Fig. 5 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 isoparametric curves are nearly orthogonal in most parts of the

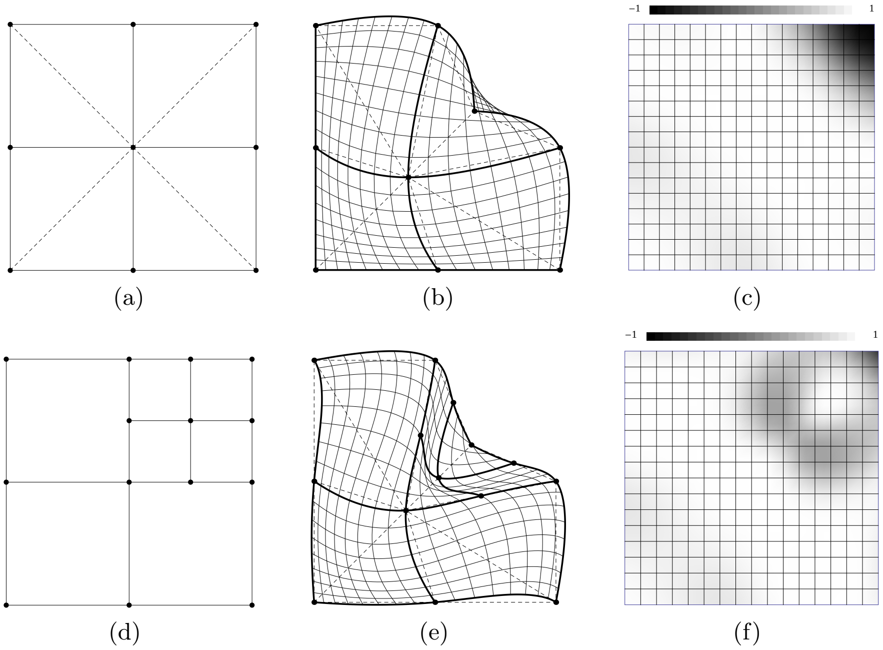
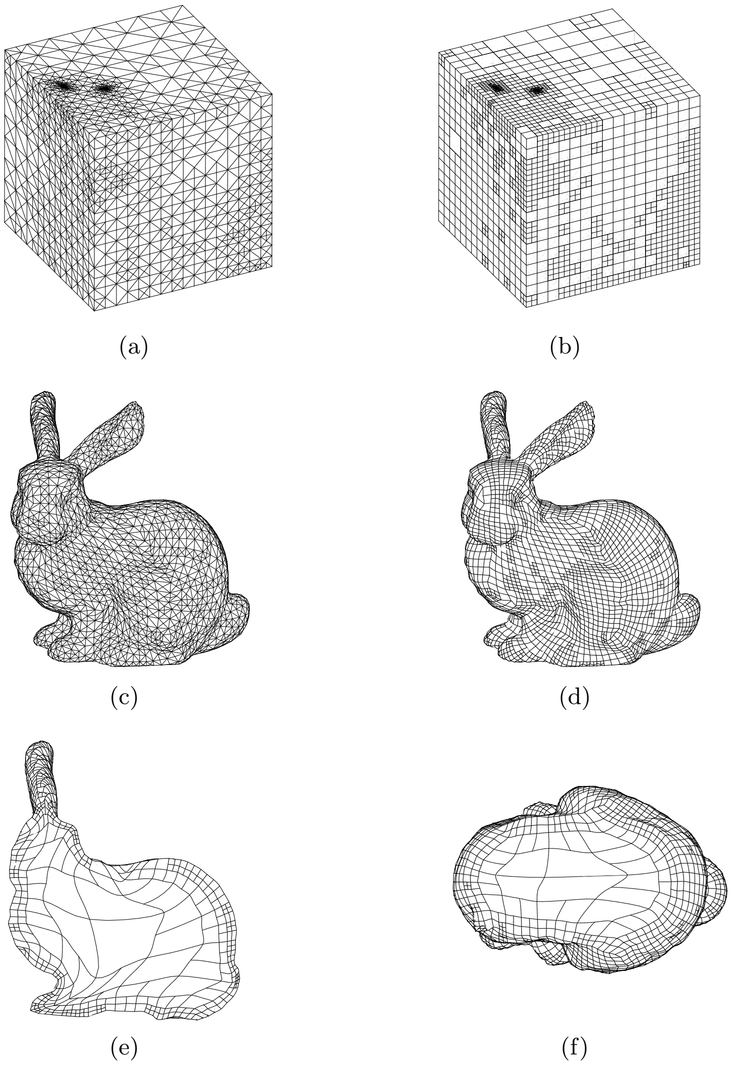


Fig. 4. Initial T-mesh in the parametric domain (continuous line) and the underlying triangular mesh (dashed line) (a). T-spline of a deformed square with a reentrant corner (b). Scaled Jacobian representation in the parametric domain (c). Corresponding representations for a refined version (d), (e) and (f).

solid, which entails low distortion and values of scaled Jacobian close to one. Nevertheless, the distortion becomes high in some regions of the surface. We have computed 39 cells out of 9696 in which at least one of the eight Gaussian quadrature points has a negative Jacobian.

As we have mentioned, 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 situated 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, $\mathcal{V}$. The first volume is measured by applying the divergence theorem and the second one is calculated integrating $\det\left(\mathbf{S}_{\xi^1}, \mathbf{S}_{\xi^2}, \mathbf{S}_{\xi^3}\right)$ 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 $\mathcal{T}_S$).



(a) (b)

(c) (d)

(e) (f)

Fig. 5. 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 $\mathcal{V}$ (d) of the Stanford bunny. Two transversal sections of $\mathcal{V}$ (e) and (f).

Guided by the results of the test example of Fig. 4, we are interested in knowing the effect of refining the cells with worst 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.

We remark that in this application 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.

4.3 Solid With Surface of Genus Greater Than Zero

We now consider the extension of the proposed isogeometric modeling to solids with surface of genus greater than zero. In this case, the volumetric parameterization of the solid is constructed by using the meccano method [7, 8, 25, 26]. 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. To obtain a T-mesh adapted to the discretization of the meccano, we apply an octree subdivision of a cube enclosing the initial polycube decomposition of the meccano. 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. As an example, we present the modeling of a solid with a genus-one surface that is explicitly given. The main stages of the process are shown in Fig. 6. In this case, the meccano is formed by four cuboids. We remark that we have also obtained positive Jacobians in all the centers of the cells of the T-mesh.

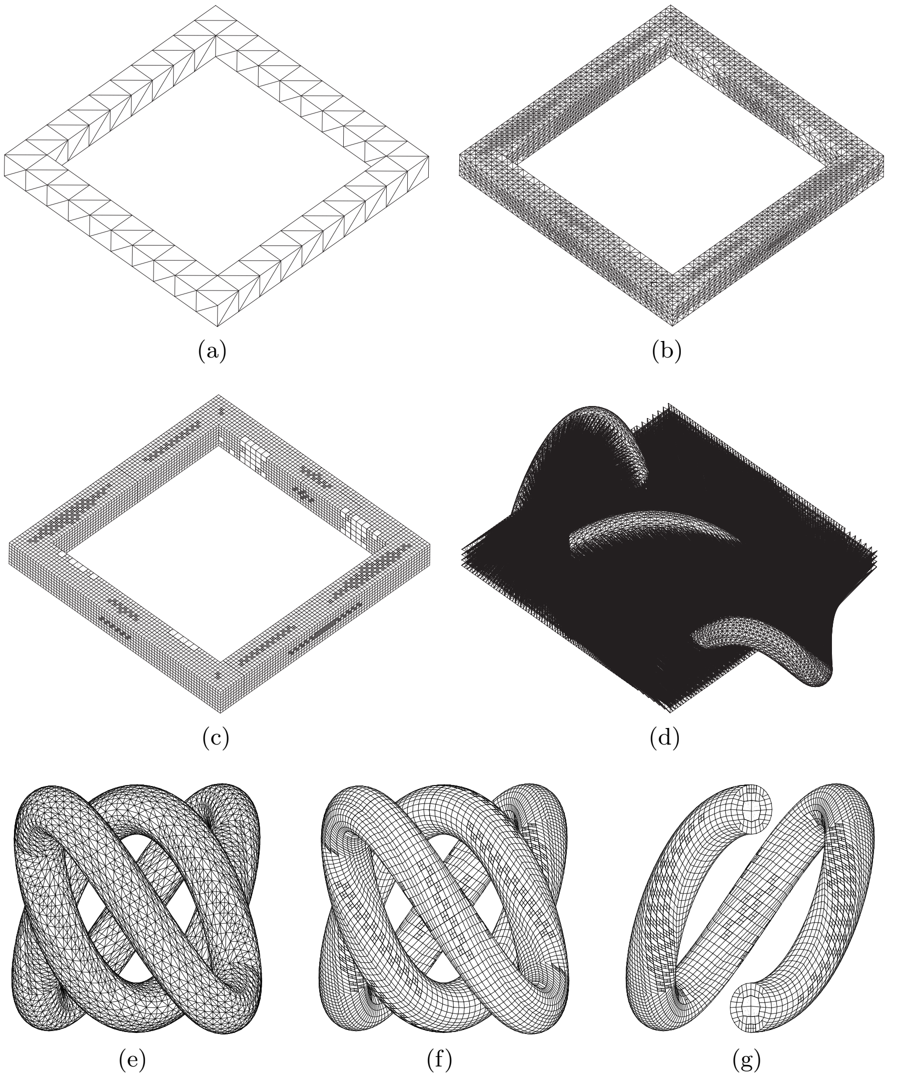
5 Conclusions and Challenges

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 presented with detail for genus-zero solids and has been introduced for genus greater than zero. The key lies on having a volumetric parameterization of the solid by using the meccano method [25, 26]. In this paper we have considered a genus-one surface parameterization explicitly given, but we think that this handicap could be overcome by applying a technique similar to PolyCube-Maps [21, 30, 31].

Furthermore, the input data of the solid 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 the meccano method.

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 inconvenience for isogeometric simulations. Just as we have shown in Section 4, the selective refinement of the most degenerate cells palliates the problem.


(a) (b)

(c) (d)

(e) (f) (g)

Fig. 6. Main stages of the isogeometric modeling of a solid with a surface of genus one: (a) Coarse tetrahedral mesh of the meccano, (b) refined tetrahedral mesh of the meccano, (c) T-mesh of the meccano, (d) tangled tetrahedral mesh after the mapping on solid surface, (e) resulting tetrahedral mesh after inner node relocation and mesh optimization, (f) T-spline representation of the solid and (g) two transversal sections.

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. Bazaraa MS, Sherali HD, Shetty CM (1993) Nonlinear programing: Theory and algorithms. John Wiley and Sons Inc, New York
2. Bazilevs Y, Calo VM, Cottrell JA, Evans J, Hughes TJR, Lipton S, Scott MA, Sederberg TW (2010) Isogeometric analysis using T-splines. Comput Meth Appl Mech Eng 199:229–263
3. Bazilevs Y, Calo VM, Cottrell JA, Evans J, Hughes TJR, Lipton S, Scott MA, Sederberg TW (2008) Isogeometric analysis: Toward unification of computer aided design and finite element analysis. Trends in Engineering Computational Technology 1–16. Saxe-Coburg Publications, Stirling
4. Borouchaki H, Frey PJ (2005) Simplification of surface mesh using Hausdorff envelope. Comput Meth Appl Mech Eng 194:4864–4884
5. Buffa A, Cho D, Sangalli G (2010) Linear independence of the T-spline blending functions associated with some particular T-meshes. Comput Meth Appl Mech Eng 199:1437-1445
6. Carey GF, Oden JT (1982) Finite elements, a second course. Prentice-Hall, New Jersey
7. Cascón JM, Montenegro R, Escobar JM, Rodríguez E, Montero G (2007) A new *meccano* technique for adaptive 3-D triangulations. Proc 16th Int Meshing Roundtable 103–120. Springer, Berlin
8. Cascón JM, Montenegro R, Escobar JM, Rodríguez E, Montero G (2009) The meccano method for automatic tetrahedral mesh generation of complex genus-zero solids. Proc 18th Int Meshing Roundtable 463–80. Springer, Berlin
9. Cohen E, Martin T, Kirby RM, Lyche T, Riesenfeld RF (2010) Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. Comput Meth Appl Mech Eng 199:334–356
10. Cottrell JA, Hughes TJR, Bazilevs Y (2009) Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons, Chichester
11. Escobar JM, Rodríguez E, Montenegro R, Montero G, González-Yuste JM (2003) Simultaneous untangling and smoothing of tetrahedral meshes. Comput Meth Appl Mech Eng 192:2775–2787
12. Escobar JM, Rodríguez E, Montenegro R, Montero G, González-Yuste JM (2010) SUS Code – Simultaneous mesh untangling and smoothing code, `http://www.dca.iusiani.ulpgc.es/proyecto2008-2011`
13. Escobar JM, Cascón JM, Rodríguez E, Montenegro R (2011) A new approach to solid modeling with trivariate T-splines based on mesh optimization. Comput Meth Appl Mech Eng, in press, doi:`10.1016/j.cma.2011.07.004`
14. Floater MS (1997) Parametrization and smooth approximation of surface triangulations. Comput Aid Geom Design 14:231–250

15. Floater MS (2003) Mean Value Coordinates. Comput Aid Geom Design 20:19–27
16. Floater MS, Hormann K (2005) Surface parameterization: A tutorial and survey. Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization 157–186. Springer, Berlin
17. Hormann K, Lévy B, Sheffer A (2007) Mesh parameterization: Theory and practice. SIGGRAPH '07: ACM SIGGRAPH 2007 courses. ACM Press, New York
18. Knupp PM (2001) Algebraic mesh quality metrics. SIAM J Sci Comput 23:193–218
19. Kossaczky I (1994) A recursive approach to local mesh refinement in two and three dimensions. J Comput Appl Math 55:275–288
20. Li X, Guo X, Wang H, He Y, Gu X, Qin H (2007) Harmonic Volumetric Mapping for Solid Modeling Applications. Proc. of ACM Solid and Physical Modeling Symposium 109–120. Association for Computing Machinery, Inc.
21. Lin J, Jin X, Fan Z, Wang CCL (2008) Automatic PolyCube-Maps. Lecture Notes in Computer Science 4975:3–16
22. Li B, Li X, Wang K (2010) Generalized PolyCube trivariate splines. SMI 2010 - Int Conf Shape Modeling and Applications 261–265
23. Martin T, Cohen E, Kirby RM (2009) Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput Aid Geom Design 26:648–664
24. Martin T, Cohen E (2010) Volumetric parameterization of complex objects by respecting multiple materials. Computers and Graphics 34:187–197
25. Montenegro R, Cascón JM, Escobar JM, Rodríguez E, Montero G (2009) An automatic strategy for adaptive tetrahedral mesh generation. Appl Num Math 59:2203-2217
26. Montenegro R, Cascón JM, Rodríguez E, Escobar JM, Montero G (2010) The meccano method for automatic 3-D triangulation and volume parametrization of complex solids. Computational Science, Engineering and Technology Series 26:19–48
27. Piegl L, Tiller W (1997) The NURBS book. Springer-Verlag, New York
28. Sederberg TW, Zheng J, Bakenov A, Nasri A (2003) T-splines and T-NURCCSs. ACM Trans. Graph. 22:477–484
29. Song W, Yang X (2005) Free-form deformation with weighted T-spline. The Visual Computer 21:139–155
30. Tarini M, Hormann K, Cignoni P, Montani C (2004) Polycube-maps. ACM Trans Graph 23:853–860
31. Wang H, He Y, Li X, Gu X, Qin H (2008) Polycube splines. Comput Aid Geom Design 40:721–733
32. Xu G, Mourrain B, Duvigneau R, Galligo A (2010) Optimal analysis-aware parameterization of computational domain in isogeometric analysis. Lecture Notes in Computer Science 6130:236–254
33. Xu G, Mourrain B, Duvigneau R, Galligo A (2010) Parametrization of computational domain in isogeometric analysis: Methods and comparison. INRIA-00530758, 1–29