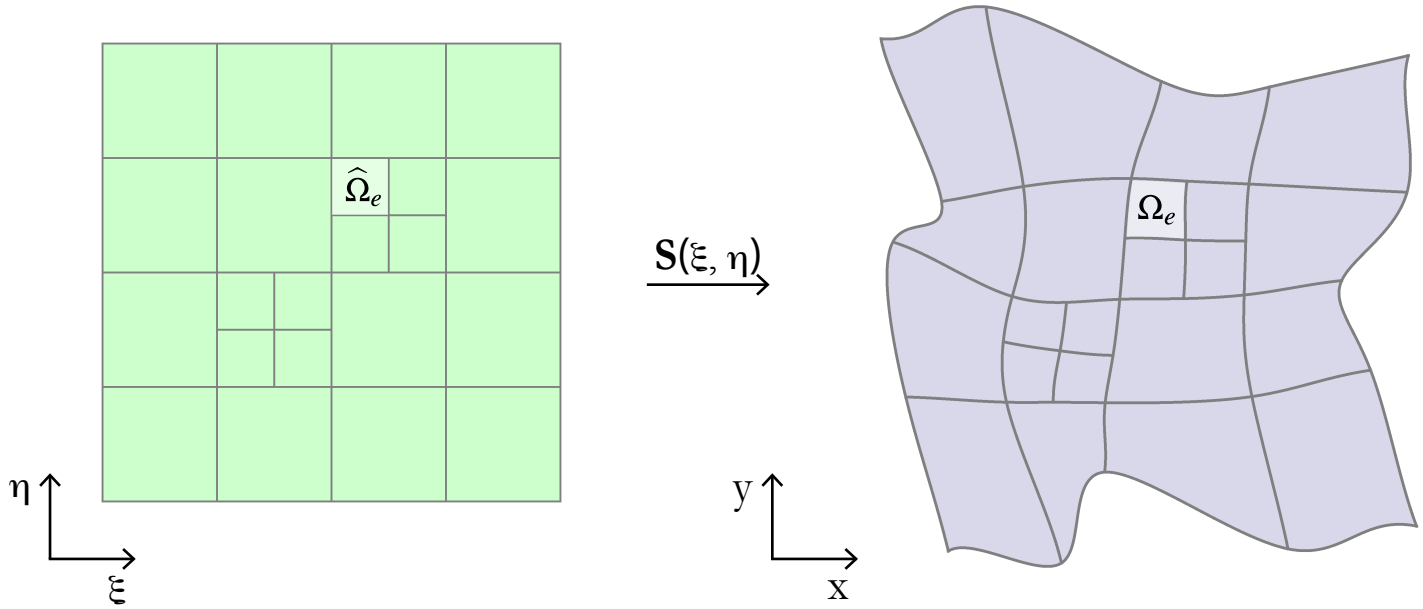
DOCTORAL DISSERTATION

Construction of spline parameterizations for its application in Isogeometric Analysis



José Iván López González

Discretization and Applications Division

Las Palmas de Gran Canaria · January 2017

INSTITUTO UNIVERSITARIO SIANI INGENIERIA COMPUTACIONAL

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Construction of spline parameterizations for its application in Isogeometric Analysis

José Iván López González

Programa de doctorado:
Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería
Instituto Universitario SIANI

Director:
José María Escobar Sánchez

Director:
Rafael Alejandro Montenegro Armas

Las Palmas de Gran Canaria, enero de 2017

Dedicated to Mari,
who has always stood by my side.

Acknowledgements

I sincerely thank to my supervisors Prof. Rafael Montenegro Armas and José María Escobar Sánchez for their support in this long journey. At first, this was a hard task due to the distance between my academic formation and the research field of this work. As José María always said me: "you play away from home". This thesis would not have been possible without their continuous advice, help, guidance and dedication. They really motivated me until the end. I wish also to express special appreciation to my co-worker Marina Brovka. Our close collaboration and mutual support made possible our researches. I will always be grateful for all her help. Many thanks to all the members of the research group: Dr. Albert Oliver Serra, Dr. Eduardo Rodríguez Barrera, Guillermo Socorro, Prof. Gustavo Montero García and Jabel Ramírez Naranjo. I am very pleased for their invaluable help and for all shared moments. Specially, I would like to express my deepest gratitude to Guillermo for two reason. First, because I always found his help when I need it, and second, because his good sense of humour always makes me smile. I also extend my appreciation to all the staff of the university institute SIANI, specially to Juan Ignacio González, for all his help and technical support.

I want to thank to my mother, for understanding me and being present in all the challenges of my life. A special thought for my father, who could not shared this moment with us, but I am sure he would have been proud of his son.

And Mari, thanks for supporting me all the moment, for your faith in me and for being always at my side. Thank you for your very valuable advices, it helped me a lot.

The author, who is a recipient of a fellowship from "Programa de FPU del Ministerio de Educación, Cultura y Deporte", wishes to acknowledge this financial support. This work was also supported by CONACYT-SENER ("Fondo Sectorial CONACYT SENER HIDROCARBUROS," grant contract: 163723).

Construction of spline parameterizations for its application in Isogeometric Analysis

Abstract

This dissertation addresses the issue of construction of spline parameterization of 2D and 3D geometries for their use in Isogeometeric Analysis. We present a strategy, based on the idea of the Meccano method and a new T-mesh optimization procedure, to construct high quality spline parameterizations from the description of the boundary of the geometry. As first step, our method defines a parametric mapping between the input boundary of the object and the boundary of the parametric domain. Then, a T-mesh adapted to the geometric singularities is constructed in order to preserve the features of the object with a desired tolerance. The key of the method lies in defining an isomorphic transformation between the parametric and physical T-meshes finding the optimal position of the interior nodes by applying a new T-mesh untangling and smoothing algorithm. Spline representation of the geometry is calculated by imposing interpolation conditions using the data provided by one-to-one correspondence between the meshes of the parametric and physical domains. To asses the quality of the parametric mapping, we evaluate its mean ratio Jacobian. Thereby, we detect the areas with low quality and perform an adaptive refinement in order to increase the degree of freedom in the region with high distortion. This strategy allows to obtain a parameterization suitable for analysis without negative Jacobian, even for complex geometries. A detailed description of the proposed method is given, and its effectiveness is tested with several geometries. Also we present some examples of the application of Isogeometric Analysis in geometries parameterized with our method.

Contents

List of Figures

1. Introduction

1.1 Isogeometric Analysis concept

Computer-aided design (CAD) is the process of creating a design using computer software. After CAD has emerged (1960-1970) all engineering drawings, done manually using pencil and paper, was replaced by files describing the physical model. That entailed a huge advance in all engineering industry. Nowadays CAD is a widely used technology with numerous applications in automotive, shipbuilding, aerospace industries, industrial and architectural design, computer animation for special effects in cinema. The most commonly used mathematical tool for representing curves and surfaces in CAD are B-splines and NURBS (Non-Uniform Rational B-splines) [1, 2]. They offer a convenient free-form surface modelling and can exactly represent all conic sections. Efficient algorithms to handle spline object are available. Besides, B-splines possess some additional useful properties as higher smoothness and convex hull property.

On the other hand, another important computational technique that has been under intensive development since its emerging (1950-1960) is Finite Element Method (FEM). This numerical method for solving partial differential equations has received a big amount of research effort and became a standard tool in engineering industry. The basic components of the method are the variational formulation of the physical problem and the discrete space used to approximate its solution. The approximation space is defined by its basis functions (Lagrange or Hermite interpolating polynomials). Each basis function is defined locally on its element. To perform the analysis, the computational domain of the problem should be decomposed in non-overlapping simple elements (triangles, quadrilaterals, tetrahedra, hexahedra). This step is referred to as mesh generation process. Taking into account that nowadays the geometry comes from CAD, there is a necessity to generate an analysis-suitable mesh of the object from its CAD representation. It is not a trivial task due to increasing complexity of engineering design. Finite Element Analysis (FEA) and CAD technologies have evolved separately and they use different geometrical tool. That impedes the communication between them and make the process of data transferring very time-consuming. Creation of the analysis-suitable model and mesh generation process consumes the major part of all analysis time. Tight communication between CAD model and FEA geometry is also crucial to make possible design optimization. In addition, finite element mesh does not reproduce the exact model designed by CAD. For some problems geometry imprecision can lead to important error in numerical solution. To approximate a curved boundary with a desired precision an adaptive mesh refinement is required. This is possible only with an automatic interaction between exact CAD geometry and FEM mesh. With all these inconveniences it became evident the necessity to unify the design and the analysis in a unique engineering process. CAD has to provide directly a geometric model suitable for analysis, or it is necessary to have an automatic method to obtain analysis suitable model from its CAD representation. For that, the analysis process should be changed and adapted for its application on CAD geometries. The thought was proposed in 2005 by Tom Hughes and co-workers in [3]. The concept received the name Isogeometric Analysis (IGA). The idea is to use for the analysis the same basis

functions (originally NURBS) that have been used for construction of the CAD model. Isogeometric Analysis can be seen as a generalization of FEM that uses basis functions with higher regularity.

Since its introduction IGA attracted a lot of attention in research community and it has been an object of numerous investigation works in the last years. At present it is considered a promising tool for bridging the gap between CAD and FEA industries. Besides, it can offer some additional beneficial properties and possibilities compared with classical Finite Element Method. Here are some of them:

- Working with the exact geometry leads to superior accuracy in the problems sensitive to geometry approximation (contact problems, boundary layer problems in aerodynamic and hydrodynamic). Even if the exact geometry is not achieved, smooth boundary approximation is necessary for some problems that require curvature continuity of the boundary.
- Better accuracy of numerical solution due to higher regularity of the basis functions [4].
- Higher degree of continuity also allows the use of IGA for solving partial differential equations of order greater than two without the necessity to change variational formulation. For example, Cahn-Hilliard phase-field equation [5, 6, 7, 8].
- Possibility of collocation methods using strong formulation [9, 10, 11, 12].

However, to accomplish the ambitious idea of Isogeometric Analysis and make it work in practice, some issues should be resolved first. This question is revised in the next section.

1.2 State of the art

IGA is relatively new method that needs a theoretical framework to be developed and some open problems to be solved. Clearly, a lot of question should be studied: error estimation theory for IGA [13, 14, 15], efficient quadrature rules [16, 17], efficiency of the direct and iterative solvers for the systems arising in the method [18, 19], efficient implementation of the structures and procedures arising in the method, imposing boundary conditions, the influence of parameterization quality on accuracy of the solution [20]. However, the most urgent and important problems of IGA are the local refinement problem and the parameterization of computational domain.

1.2.1 Domain parameterization

CAD models usually provide only the boundary surface of a solid. This spline representation of the boundary can be used directly in isogeometric shell analysis or isogeometric boundary integral method. But the application of Isogeometric Analysis requires

a full volumetric representation of the geometry. An open problem 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 Cottrell et al. in [21], "the most significant challenge facing Isogeometric Analysis is developing three-dimensional spline parameterizations from surfaces". Parameterization is suitable for analysis if it does not have self-intersection, i.e. it is invertible. Moreover, in order to expect a high accuracy in numerical results it is necessary to obtain a good quality volume parameterization. Orthogonality and uniformity of isoparametric curves are desirable for the tensor-product structured parameterization. It is not trivial task to obtain a good quality smooth global parametric mapping for complex domains and it can be very time-consuming. For the application of IGA it is essential to have an effective method to construct a spline parameterization. Here are some attempts to tackle the problem.

In [22] the parameterization is found by solving a constraint optimization problem for a planar B-spline surface. Constraints are defined by imposing injectivity sufficient conditions in terms of the control points, and the optimization consists in minimizing some energy function in order to reach a good orthogonality and uniformity of the parametric mapping. The idea was extended for 3D in [23]. Another similar technique was proposed by these authors in [24, 25]. They use a harmonic mapping obtained by solving an optimization problem for the control points. Additional term is added to the objective function in order to improve the quality where needed.

The use of harmonic mapping is a common characteristic of several works dealing with 2D and 3D parameterization methods. For example, Li et al. [26] 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.

Martin et al. [27] present a methodology based on discrete harmonic functions to parameterize a solid, where the input data are surface triangulation and a tetrahedral mesh of the solid. They solve several Laplace equations, first on the surface to establish surface parameterization and then on the complete 3D domain using FEM. The two obtained orthogonal harmonic functions are used to construct a structured hexahedral mesh of the solid, which is smoothed to remove irregularities. 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.

Zhang et al. proposed in [28] a procedure to construct T-spline representation of a genus-zero solid from its boundary triangulation. First, a parametric mapping between the triangulation and the boundary of the parametric domain, a unit cube, is established using Floater parameterization method. Then an octree subdivision is carried out for the cube until the error between the T-mesh and the input triangle mesh is less than a threshold. During this process, the boundary nodes are mapped to the input triangle mesh and, then, the interior nodes are relocated via T-mesh optimization, which is maximizing the worst Jacobian of the trilinear map for each

element. In [29] the method was extended for arbitrary genus topology. A smooth harmonic scalar field over the surface triangulation is computed, and saddle points are extracted to determine the topology of the object and to construct a polycube with the same topology that serves as the parametric domain for the trivariate T-spline.

1.2.2 Local refinement

Originally Isogeometric Analysis concept was proposed and tested using NURBS basis functions inasmuch as NURBS was the most common tool used by CAD software at that moment. However, it suffers from an important drawback: their tensor product structure does not allow for local refinement. NURBS surface is defined by a set of control points which lie, topologically, in a rectangular grid. A new control point insertion induces the insertion of an entire row of control points extended through the entire domain. That leads to a large number of superfluous control points. The representation of local features is inefficient and requires the use of several NURBS patches joined together. Frequently this join is discontinuous, i.e. it has gaps and overlaps that make the model not suitable for analysis. T-splines were proposed by Sederberg et al. [30] as an alternative to NURBS that permits local refinement. T-splines can be seen as a generalization of NURBS. T-splines allows the insertion of a partial row of control points that terminates in T-junction, which makes them locally refinable. T-spline plug-in are currently available for two NURBS-based CAD software Maya and Rhino, see `http://www.tsplines.com/products/tsplines-for-rhino.html`. T-splines offer a flexible tool for creating one-patch watertight surfaces with local detailed features and lower number of control points compared with NURBS, see Fig. 1.1. However, it appears that initially introduced T-splines are not suitable for their use in Isogeometric Analysis since they lack some properties essential for analysis and for proper convergence behaviour: linear independence, polynomial reproduction property and nesting behaviour of approximation spaces. Besides, rational T-splines complicate and increase computational cost of the calculus of the derivatives.

Therefore, it is still an open issue in the context of Isogeometric Analysis to find an alternative to NURBS that overcomes the local refinement problem and can be used in analysis. This issue has been the object of numerous research works in recent years.

Analysis-suitable T-splines, proposed by Scott et al. in [31], are a class of T-splines defined over T-meshes that meet certain topological restrictions formulated in terms of T-junction extensions. Blending functions defined over an extended analysis-suitable T-mesh are linearly independent [32] and possess polynomial reproduction property. The refinement algorithm allows to accomplish highly localized refinements and construct nested T-spline spaces, but it presents an elevated implementation complexity and, as far as we know, the generalization of the strategy to 3D cases is still an open question.

Another well known approach for local enrichment of the approximation space is the hierarchical refinement, originally introduced by Forsey and Bartels in [33] and later developed in [34]. Recently, hierarchical refinement technique in the context of Isogeometric Analysis was described in [35, 36, 37]. This approach is based on a simple and

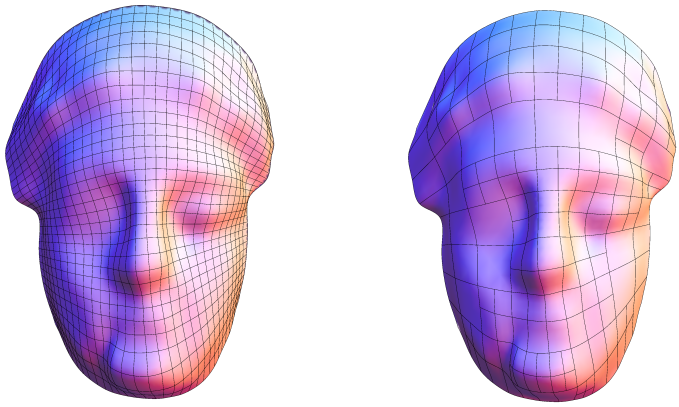


Figure 1.1. NURBS vs T-spline. (a) Igea face modelled as a NURBS with 1225 control points. (b) T-spline version with 288 control points.

natural idea to construct multilevel spaces by replacing coarse level functions with finer basis functions. Starting from an initial uniform mesh, hierarchical refinement scheme leads to sequential construction of nested spline spaces with linearly independent basis functions. Relatively simple implementation and straightforward generalization to 3D make it an attractive option for local refinement. However, a shortcoming of this strategy is the impossibility to define a spline space over a given arbitrary T-mesh, as well as the presence of redundant basis functions and excessive support overlapping. An interesting theoretical approach to the latter problem was given in [38]. The truncation technique is applied to redefine the function supports and reduce their overlapping. However, that elevates considerably the computational cost of the strategy.

Other strategies for performing local refinement of spline spaces are C^1-continuous PHT-splines [39] and Locally Refined B-splines (LR-splines) [40].

It is worth mentioning another group of options for defining multivariate spline functions that do not have tensor product structure. Namely, bivariate simplex splines over triangulation [41]; simplex splines over Delaunay configurations, as a natural generalization of univariate B-splines, was proposed in [42, 43, 44]; box splines [45]; quadratic C^1-continuous splines over Powell-Sabin triangle split [46]. Recent examples of isogeometric analysis using splines over triangulations can be found in [47, 48, 49]. Splines over triangulation could offer a better flexibility and adaptivity to irregular domains compared to more rigid tensor product splines. However, it is not a trivial task to define globally smooth spline spaces over triangulations. Currently available options presents a high complexity, and it seems that tensor product splines are still the most popular and standard option for CAD due to their simplicity.

1.3 Goal and outline of the thesis

This thesis is result of our group research work on Isogeometric Analysis. As was mentioned before, IGA is a relatively new method with many questions to explore. A lot of new interesting things about IGA have been studied and discovered since we have embarked on this journey. But, mainly, our attention was focused on two open problems of IGA: the problem of parameterization of a computational domain from its boundary representation and the construction of spline spaces with nice properties suitable for analysis. A brief summary about construction of appropriate spline spaces developed in our research is included in this work, however this problem was studied in depth and presented in another doctoral dissertation of the group.

This thesis addresses the issue of construction of a spline parameterization suitable for its application in Isogeometric Analysis. The main contribution of the dissertation is a new method to obtain high quality spline parameterization of both, 2D and 3D geometries, for their use in Isogeometeric Analysis. The strategy is based on previous works of the group [50, 51] in which a volumetric parameterization was obtained by deforming a tetrahedral mesh of the solid. Our goal in this thesis is to extend the ideas of the Meccano method [52] to work directly over a T-mesh, without the necessity of an intermediate tetrahedral mesh. The proposed method only demands a boundary representation of the domain as input data, and constructs automatically a spline transformation between the physical and parametric domains. The strategy first establishes a parametric mapping between the input and the boundary of the parametric domain. Then, a parametric T-mesh adapted to the geometric singularities is built in order to preserve the features of the object with a desired tolerance. A physical T-mesh is obtained by mapping the boundary points of the parametric adapted T-mesh to the physical boundary. The inner nodes of the physical T-mesh are relocated by applying a T-mesh untangling and smoothing procedure, which is the key of the method. Spline representation of the geometry is calculated by imposing interpolation conditions using the data provided by one-to-one correspondence between the meshes of the parametric and physical domains.

The manuscript is organized as follows.

- Chapter 2 introduces some basic concepts about modelling with B-splines, as well as the concept of T-spline and T-mesh. Then, a brief summary of a volumetric parameterization method based on the Meccano method is exposed.

- Chapter 3 presents one of the main results of this thesis: a T-spline parameterization method for 2D geometries. The strategy proposed here was published in [53].

- In chapter 4 several strategies, based on different objective functions, for optimization of hexahedral meshes are studied. This research established the basis that allowed us to propose an optimization method for 3D T-meshes. The results of this study was published in [54].

- Chapter 5 presents a brief description of another result of our research: a strategy for constructing polynomial spline spaces over hierarchical T-meshes with quad- and octree subdivision scheme, which is used in our general (2D and 3D) parameterization method. This work was presented in another doctoral dissertation of the group [55] and in [56].

- Chapter 6 is the core of this dissertation. Here we present a generalization of our spline parameterization method to work with 2D and 3D geometries.

- In chapter 7 the proposed strategies for spline parameterization are tested with different geometries. We apply IGA for different type of problems using the parameterized geometries. Another possible application of our optimization method is presented: optimization of unstructured quadrilateral and hexahedral high order meshes.

- Chapter 8 exposes the summary, concluding remarks and future research directions.

1.4 Published works derived from this Ph.D. thesis

The strategies proposed in this dissertation, as well as part of the results, have been previously presented in different journals [53, 56, 54] and conferences.

1.4.1 Contributions in JCR journals

- A new method for T-spline parameterization of complex 2D geometries. M. Brovka, J.I. López, J.M. Escobar, J.M. Cascón, R. Montenegro. Engineering with Computers, 30 (4), 2013, pp. 457–473.

- A simple strategy for defining polynomial spline spaces over hierarchical T-meshes. M. Brovka, J.I.López, J.M. Escobar, R. Montenegro, J.M. Cascón. Computer-Aided Design, 72, 2016, pp. 140–156.

- Strategies for optimization of hexahedral meshes and their comparative study. J.I. López, M. Brovka, J.M. Escobar, R. Montenegro and G. V. Socorro. Engineering with Computers, 33 (1), 2017, pp. 33–-43.

1.4.2 Conference contributions

- T-spline Parameterization of 2D Geometries Based on the Meccano Method with a New T-mesh Optimization Algorithm, Proceedings of the 22nd International Meshing Roundtable, Springer, 2014, pp 57-74, J. I. López, M. Brovka, J. M. Escobar, J. M. Cascón, R. Montenegro.

- Construction of Polynomial Spline Spaces Over Quadtree and Octree T-meshes, 23rd International Meshing Roundtable (IMR23), Procedia Engineering, Volume 82, 2014, pp 21–33, M. Brovka, J.I. López, J.M. Escobar, J.M. Cascón, R. Montenegro.
- An Optimization Based Method for the Construction of 2D Parameterizations for Isogeometric Analysis with T-splines, New Challenges in Grid Generation and Adaptivity for Scientific Computing, SEMA SIMAI Springer Series, Volume 5, 2015, pp 91-112, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón and R. Montenegro.
- Application to Complex Solids of Adaptive Isogeometric Analysis using T-splines, 10th WCCM , July 8-13, 2012, Sao Paulo, Brazil, R. Montenegro, J.M. Cascón, E. Rodríguez, J.M. Escobar, M. Brovka, J.I. López, J. Ramírez.
- The Meccano Method for Finite Element and Isogeometric Analysis, ACM 2013 A Conference Celebrating the 70th Birthday of Thomas J.R. Hughes, February 24-27, 2013, San Diego, CA, USA, R. Montenegro, J.M. Cascón, E. Rodríguez, J.M. Escobar, M. Brovka, J.I. López, J. Ramírez.
- Three-dimensional Simulation of Wind Fields and Air Pollution over Complex Terrain, CEID Annual Seminar 2014, May 27, 2014, Finland, A. Oliver, E. Rodríguez, J. Ramírez, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, F. Díaz, G.V. Socorro, G. Montero and R. Montenegro.
- Modelización de sólidos mediante estructura octree, Congress on Numerical Methods in Engineering 2013, June 25-28, 2013, Bilbao, Spain, J.I. López, M. Brovka, J. Ramírez, R. Montenegro, J.M. Escobar, J.M. Cascón, E. Rodríguez.
- Extensión del Método del Mecano para Análisis Isogeométrico con T-splines, Congress on Numerical Methods in Engineering 2013, June 25-28, 2013, Bilbao, Spain, J.I. López, M. Brovka, J. Ramírez, R. Montenegro, J.M. Escobar, J.M. Cascón, E. Rodríguez.
- Mesh Generation for Isogeometric Analysis with T-spline, IV Thetrahedron 2013, July 1, 2013, Verbania, Italia, J.M. Escobar, R. Montenegro, M. Brovka, J.I. López.
- T-spline Parameterization of 2D Geometries Based on the Meccano Method with a New T-mesh Optimization Algorithm, Proceedings of the 22nd International Meshing Roundtable, 22nd International Meshing Roundtable, October 13-16, 2013, Orlando, USA, J. I. López, M. Brovka, José María Escobar, J. M. Cascón, Rafael Montenegro.
- Advances of the Meccano Method for Isogeometric Analysis of Irregular Planar Domains, Geometric and Physical Modeling, GD/SPM13, the Curtis-a DoubleTree by Hilton Hotel Denver, Colorado, USA, November 11, 2013, J. I. López, M. Brovka, J. M. Escobar, J. M. Cascón, R. Montenegro.

- Adaptive T-spline Refinement for Isogeometric Analysis in Planar Geometries, Isogeometric Analysis: Integrating Design and Analysis (IGA 2014), January 8-10, 2014, Austin, Texas, USA, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, R. Montenegro
- Mesh Generation and Isogeometric Analysis, Department of Computer Science AGA University of Science, April 2, 2014, Krakow, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, A. Oliver, J. Ramírez, G.V. Socorro, E. Rodríguez, G. Montero, R. Montenegro.
- Some advances on T-spline parameterization based on the Meccano method, Proceedings of 11th World Congress on Computational Mechanics (WCCM XI), 5th European Conference on Computational Mechanics (ECCM V) and 6th European Conference on Computational Fluid Dynamics (ECFD VI), July 20–25, 2014, Barcelona, Spain, J. I. López, M. Brovka, José María Escobar, J. M. Cascón, Rafael Montenegro.
- El método del mecano y análisis isogeométrico, Tercer Encuentro Conjunto de la Real Sociedad Matemática Española y la Sociedad Matemática Mexicana, September 1-4, 2014, Zacatecas, México, R. Montenegro, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, A. Oliver, F. Díaz, J. Ramírez, G.V. Socorro, E. Rodríguez, G. Montero.
- Construction of polynomial spline spaces over quadtree and octree T-meshes, Proceedings of the 23rd International Meshing Roundtable, 23rd International Meshing Roundtable, October 12-15, 2014, London, M. Brovka, J.I. López, J.M. Escobar, J.M. Cascón, R. Montenegro.
- 3-D Wind Field Simulation over Complex Terrain, Congreso RSME, February 2-6, 2015, Granada, Spain, A. Oliver, E. Rodríguez, J. Ramírez, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, F. Díaz, G.V. Socorro, R. Montenegro, G. Montero.
- Some advances in open problems of isogeometric analysis, XXIV Congreso de Ecuaciones Diferenciales y Aplicaciones (XXIV CEDYA / XIV CMA), June 8-12, 2015, Cádiz, Spain, M. Brovka, J.I. López, J.M. Escobar, R. Montenegro.
- Analysis of several objective functions for optimization of hexahedral meshes, VII European Congress on Computational Methods in Applied Sciences and Engineenring (ECCOMAS Congress 2016), June 5-10, 2016, Island of Crete, Greece, J.I. López, M. Brovka, J.M. Escobar, R. Montenegro and G. V. Socorro.

2. Preliminaries

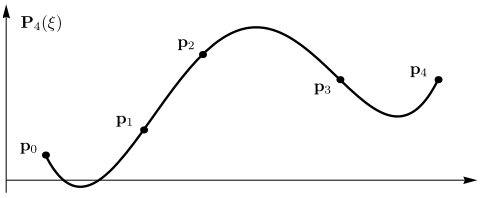
In this chapter some basic concepts are introduced for the correct understanding of this thesis. First, we summarize the general idea of spline and explain the fundamental mathematical details about modeling with B-splines. Then, domain parameterization in IGA is introduced, as well as the concept of T-spline and T-mesh. Finally, a brief summary of a volumetric parameterization method based on the Meccano method is exposed. For a more comprehensive review we refer the reader to [57, 2, 3, 50, 51].

2.1 Introduction to spline

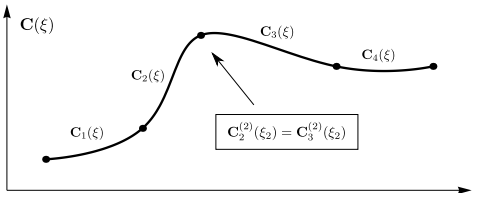
Data fitting is a fundamental concept for geometric design. A common problem in curve modelling is data interpolation: given data points $\mathbf{p}_i$ and their corresponding parameter values ξ_i, find a parametric curve that passes through the points $\mathbf{p}_i$. A well known method to solve this problem is polynomial interpolation, that is, find a n-order polynomial $\mathbf{P}_n$ that satisfy interpolation conditions

$$\mathbf{P}_n(\xi_i) = \mathbf{p}_i, \quad i = 0, \ldots, n.$$

However, a single segment polynomial interpolation have some shortcomings: a large number of data points requires a high degree polynomial which is inefficient and can present oscillations, known as Runge's phenomenon, and numerical unstability (see [2] for an example of ill-conditioned Lagrange polynomial interpolation). A solution to this problem is piecewise polynomial interpolation. The idea is to construct a polynomial segments $\mathbf{C}_i(\xi)$ for each parameter interval $[\xi_{i-1}, \xi_i]$ so that the resulting piecewise curve $\mathbf{C}(\xi)$ passes through the data points $\mathbf{p}_i$ and the polynomial segments join with a certain level of continuity, i.e., $\mathbf{C}_i^{(k)}(\xi_i) = \mathbf{C}_{i+1}^{(k)}(\xi_i)$.



(a) A single 4-th order polynomial fitting data points.



(b) Cubic spline curve fitting the same data points with C^2-continuity between the segments.

Figure 2.1. Data interpolation. (a) Lagrange polynomial interpolation. (b) Piecewise polynomial interpolation.

This piecewise polynomial curve $\mathbf{C}(\xi)$ is called spline, see Fig. 2.1. The term spline comes from the name of flexible rulers that were used for technical drawings and were free to bent to pass through the specified points. Spline curve can be seen as a mathematical generalization of the physical spline.

The set of all piecewise polynomial curves of degree n and some level of continuity over a given sequence of parameter values $\{\xi_i\}$ form a vector space. It is possible to define some elemental piecewise polynomial functions of the same degree and level of continuity that form a basis for this space. Then, any spline curve can be constructed as linear combination of the basis piecewise polynomials. These basis functions are called B-splines (Basis splines).

2.2 B-spline basis

Here is a brief summary of the main concepts about B-splines.

A set of B-spline basis functions $B_{i,p}$ $(i = 1,2,\ldots,n)$ of degree p, inferred from a non-decreasing sequence $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, called knot vector, is defined by the Cox-de Boor recursion formula

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if} \quad \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{othewise.} \end{cases}$$

$$B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi).$$

The values ξ_i are called knots and each interval (ξ_i, ξ_{i+1}) is known as knot span. A knot vector Ξ is called open knot vector if the first and the last knots are repeated $p+1$ times. It is said that a knot have multiplicity m when it is repeated m times, and the B-spline is C^{p-m}-continuous at the knot. In Fig. 2.2 you can see an example of cubic B-splines. Next we enumerate some basic properties of B-spline functions:

- Local support: $B_{i,p}(\xi) = 0$ for ξ outside the interval $[\xi_i, \xi_{i+p+1})$.
- Non-negativity: $B_{i,p}(\xi) \geq 0$.
- Partition of unity: $\sum_{i=1}^{n} B_{i,p}(\xi) = 1$, $\xi \in [\xi_1, \xi_{n+p+1}]$.
- Linear independence: $\sum_{i=1}^{n} c_i B_{i,p}(\xi) \equiv 0 \quad \Rightarrow \quad c_i = 0, i = 1,\ldots,n.$
- Except for the case $p = 0$, $B_{i,p}$ attains exactly one maximum.
- Each function $B_{i,p}$ is a piecewise polynomial of degree p with C^{p-m} continuity at its knots of multiplicity m.
- The derivative of a B-spline function $B_{i,p}$ can be expressed as a linear combination of B-splines of lower degree

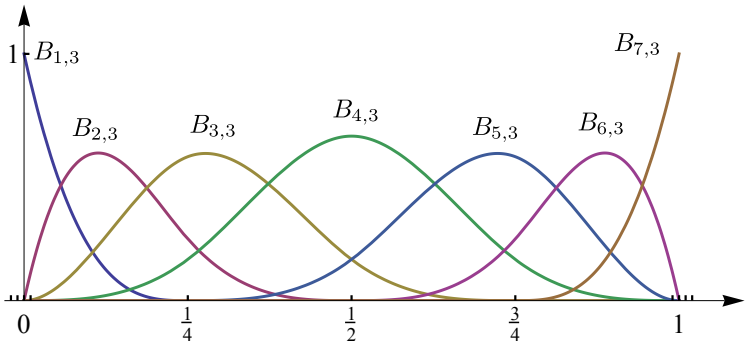


Figure 2.2. A set of cubic B-splines inferred from the open knot vector $\Xi = \{0,0,0,0,1/4,1/2,3/4,1,1,1,1\}$.

2.3 B-spline curves, surfaces and solids

A B-spline curve is defined as a linear combination of B-spline basis functions as

$$\mathbf{S}(\xi) = \sum_{i \in I} \mathbf{P}_i \, B_{i,p}(\xi),$$

where coefficients $\mathbf{P}_i \in \mathbb{R}^s$ are called control points, typically $s = 2$ or 3. $\{B_{i,p}\}_{i=1}^{n}$ are the B-spline basis functions defined on the open knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$. The B-spline curve does not interpolate the control points, except for the first one and the last one, due to the open knot vector. The polygon formed by the control points $\mathbf{P}_i$ is called the control polygon. The continuity of the spline curve is determined by the continuity of its basis functions. See Fig. 2.3 for an example of B-spline curve and its control polygon.

A B-spline curve has the following properties:

- Convex hull property: the curve is contained within the convex hull of its control points. This property is useful for checking if two curves intersect each other.
- Affine invariance: the curve is invariant under affine transformation of its control points. It implies that an affine transformation of the curve can be obtained by applying the transformation directly to the control points.
- Variation diminishing property: no plane has more intersections with the curve than with its control polygon. This property makes the B-spline curve more monotone, compared with oscillating behaviour that can have Lagrange polynomials.
- Local control property: moving a control point $\mathbf{P}_i$ affects only locally the shape of the curve $\mathbf{C}(\xi)$, namely only the interval $[\xi_i, \xi_{i+p+1})$. This property is important for geometric design since it allows to change locally the shape without the necessity to recalculate the entire curve.

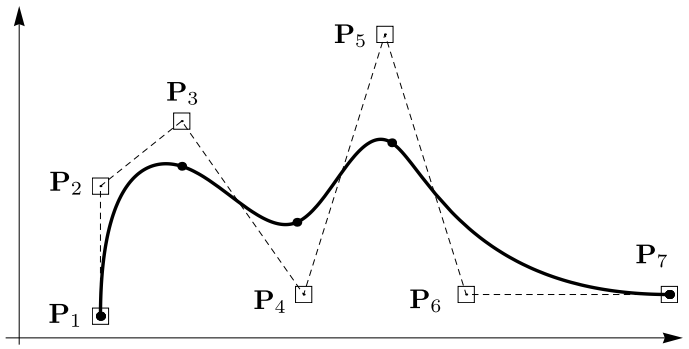


Figure 2.3. Cubic B-spline curve and its control polygon.

Multivariate B-splines are defined as a tensor product of univariate B-spline functions

$$B_{\mathbf{i},p}(\boldsymbol{\xi}) = \prod_{k=1}^{d} B_{i_k,p}(\xi^k),$$

where $\boldsymbol{\xi} = (\xi^1, \ldots, \xi^d)$ and the multi-index $\mathbf{i} = (i_1, \ldots, i_d) \in I$. The multi-index set is defined by $I = \{1,2,\ldots,n_1\} \times \cdots \times \{1,2,\ldots,n_d\}$. For example, let $\{B_{i,p}(\xi)\}_{i=1}^{n_1}$ and $\{B_{j,q}(\eta)\}_{j=1}^{n_2}$ be two sets of B-spline basis functions of degree p and q, defined over the knot vectors $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n_1+p+1}\}$ and $\mathscr{H} = \{\eta_1, \eta_2, \ldots, \eta_{n_2+q+1}\}$, respectively, which form a Cartesian grid in parametric domain. Then, bivariate basis functions are defined as follows

$$B_{(i,j),(p,q)}(\xi,\eta) = B_{i,p}(\xi) B_{j,q}(\eta), \quad (i,j) \in I,$$

been $I = \{1,2,\ldots,n_1\} \times \{1,2,\ldots,n_2\}$. See an example of bivariate B-spline function in Fig. 2.4 and an example of a set of B-spline functions defined over a parametric grid in Fig. 2.5.

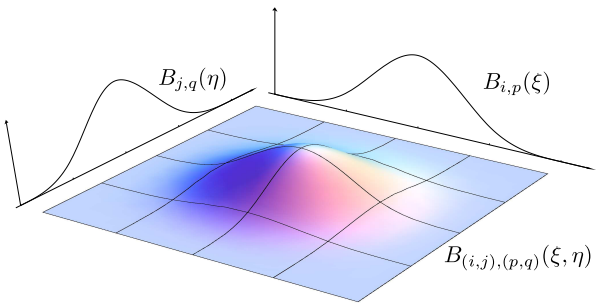
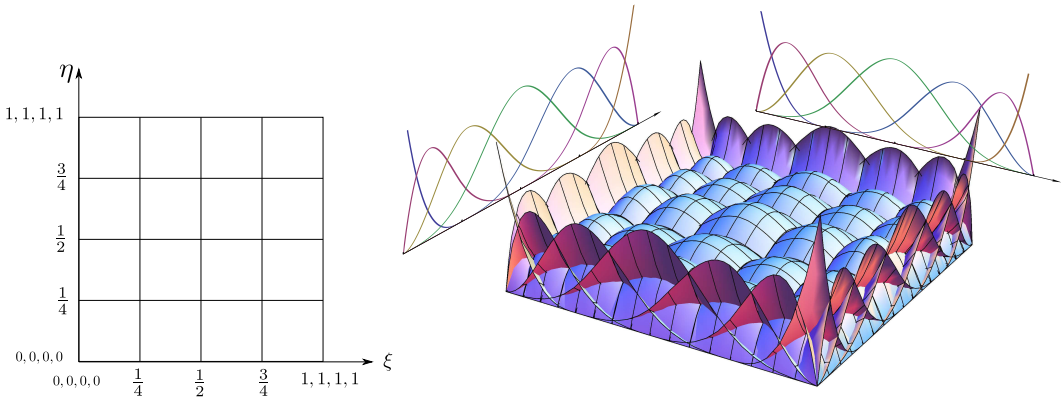


Figure 2.4. Bivariate cubic B-spline function.

A B-spline surface is defined as a linear combination of bivariate B-spline functions



(a) (b)

Figure 2.5. Bivariate cubic B-spline functions defined over a parametric grid formed by the knot vectors $\Xi = \mathscr{H} = \{0,0,0,0,1/4,1/2,3/4,1,1,1,1\}$.

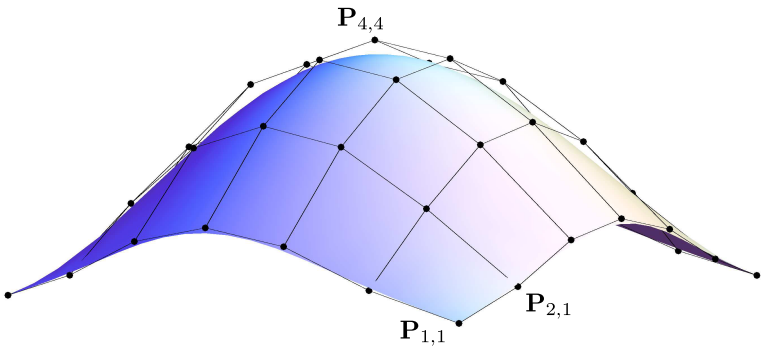


Figure 2.6. B-spline surface and its control net.

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\mathbf{i} \in I} \mathbf{P}_{\mathbf{i}} \, B_{\mathbf{i},p}(\boldsymbol{\xi}),$$

where the control points $\mathbf{P}_{\boldsymbol{i}} \in \mathbb{R}^3$ form a control net. See an example of B-spline surface in Fig. 2.6. B-spline surface conserves some properties of the spline curve: convex hull property, affine invariance and local control property, but no variation diminishing property.

B-spline solids are defined in analogous way, as a linear combination of trivariate B-spline basis functions. Multivariate B-spline basis possess the same properties as the univariate B-splines.

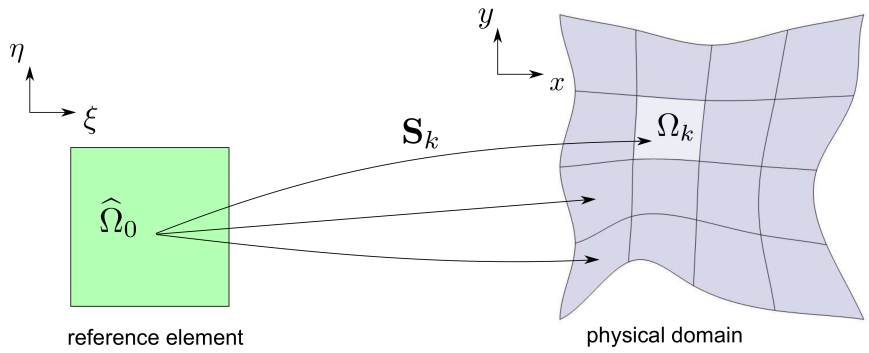


Figure 2.7. Local parametric mapping for each element in Finite Element Method.

2.4 IGA and the parameterization of the domain

The main idea of Isogeometric Analysis is to use for the solution space the same basis functions that are used to model the geometry. This notion is called isoparametric concept, and it is common for IGA and FEA. However, as it was pointed by Cottrell et al. in [21], "The fundamental difference between this new concept of Isogeometric Analysis and the old concept of isoparametric finite element analysis is that, in classical FEA, the basis chosen to approximate the unknown solution fields is then used to approximate known geometry. Isogeometric Analysis turns this idea around and selects a basis capable of exactly representing the known geometry and uses it as a basis for the fields we wish to approximate. In a sense, we are reversing the isoparametric arrow such that it points from the geometry toward the solution space, rather that vice versa".

In IGA, the computational domain is defined by a global transformation. This is one of the major differences between Isogeometric Analysis and the Finite Element Method. In FEM, each element of the physical domain has its own parametric mapping that transforms the reference element into the physical one. This local parameterization S_k depends only on the nodes of the physical element and it is constructed with the basis functions defined on the reference element, typically Lagrange basis. See in Fig. 2.7 a representation of the FEM local transformation of a element. In IGA, a unique global parametric mapping $\mathbf{S} : \widehat{\Omega} \to \Omega$ transforms the entire parametric domain into the physical one and it is defined as

$$\mathbf{S}(\xi,\eta) = \sum_j \mathbf{P}_j \widehat{N}_j(\xi,\eta),$$

where $\{\widehat{N}_j\}$ is the set of spline basis functions defined on the whole parametric domain $\widehat{\Omega}$. The parametric domain is a rectangular grid formed by two global knot vectors Ξ and $\mathscr{H}$, as you can see in Fig. 2.8(a). Each element Ω_k in the physical domain has its own counterpart $\widehat{\Omega}_k$ in the parametric domain. And the parametric mapping $\mathbf{S}$ restricted to Ω_k depends not only on the control points associated with the element but also on the neighbouring elements.

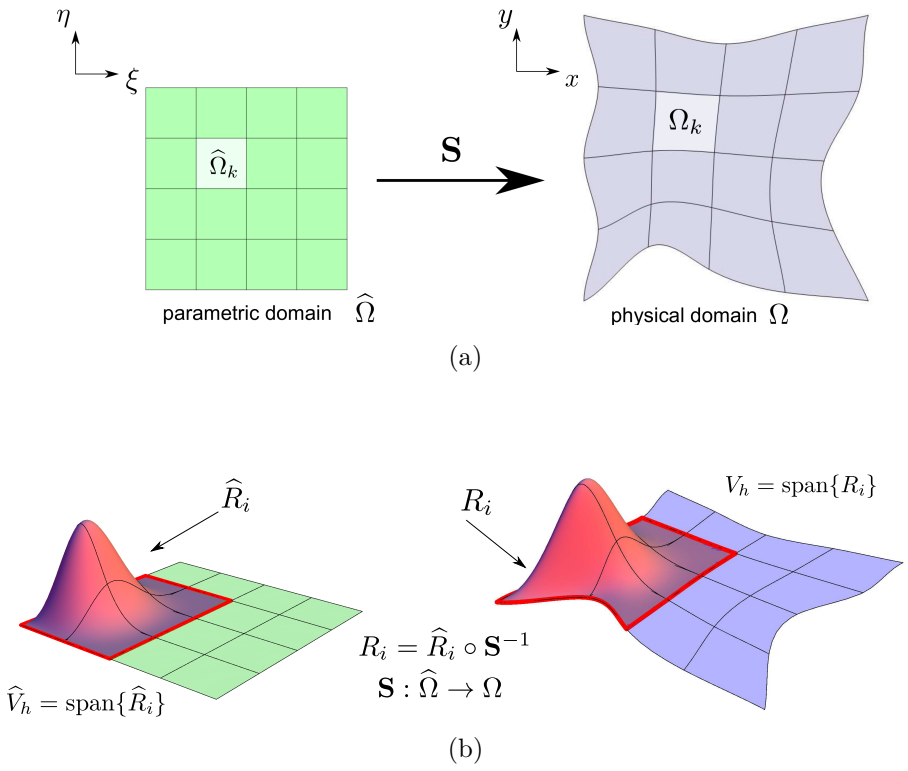


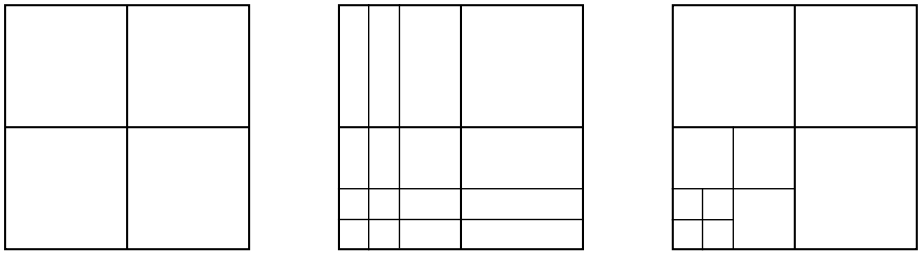
Figure 2.8. (a) Global parametric mapping in Isogeometric Analysis. (b) Basis function in parametric and physical space.

2.5 T-splines and T-meshes

To achieve a desired accuracy of the numerical solution with as few basis functions as possible, it is necessary to identify, using some a posteriori error estimation, the areas that contribute the most to the global error and increase the number of basis functions in this area. Both for analysis and design, it is of great importance to perform efficient local enrichment of approximation space. B-spline suffers from an important drawback: their tensor product structure does not allow for local refinement. B-spline surface is defined over a rectangular grid. A new knot insertion induces the insertion of a knot line extended through the entire domain, see Fig. 2.9(b).

T-splines were proposed by Sederberg et al. [30] as an alternative to NURBS that permits local refinement. T-splines can be seen as a generalization of NURBS. T-splines allows the insertion of a partial line of new knots that does not propagate throughout the domain, which makes them locally refinable, see Fig. 2.9(c).

An axes-aligned grid that allows T-junctions is called T-mesh. The concept T-junction is similar to hanging node in the classical Finite Element Method. The underlying idea of T-splines consists in defining the blending functions by means of a set of local knot vectors instead of a global knot vector, as in the case of B-splines or NURBS. To define

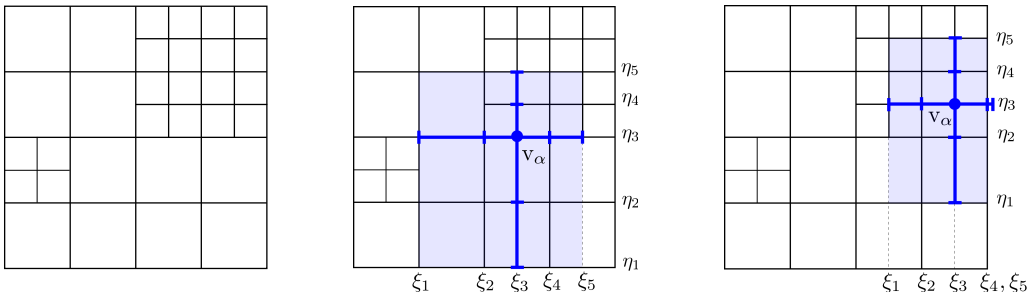


(a) initial mesh (b) global refinement (c) local refinement

Figure 2.9. Refinement of B-splines. (b) A knot insertion induces the insertion of an entire knot line. (c) Local refinement with T-mesh.

a set of basis functions for a given T-mesh, to each vertex (node) v_α of the mesh a blending function B_α is associated. The vertex v_α is called anchor of the function. Next, for simplicity, we are going to explain the idea for the case of cubic T-spline functions over 2D T-meshes. Local knot vectors $\Xi_\alpha = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5)$ and $\mathscr{H}_\alpha = (\eta_1, \eta_2, \eta_3, \eta_4, \eta_5)$ for each bivariate function B_α are inferred by marching through the T-mesh edges in both parametric directions, starting from its anchor $\mathrm{v}_\alpha = (\xi_3, \eta_3)$, until a perpendicular edge is encountered, see Fig. 2.10(b). When, during this marching the boundary is reached, the knots are repeated to create an open knot vector structure along the boundary, see Fig. 2.10(c). For a pair of local knot vectors Ξ_α and $\mathscr{H}_\alpha$ the bicubic spline function B_α is defined as $B_\alpha(\xi, \eta) = B[\Xi_\alpha](\xi) B[\mathscr{H}_\alpha](\eta)$, where $B[\Xi_\alpha](\xi)$ and $B[\mathscr{H}_\alpha](\eta)$ are the univariate B-splines corresponding to the knot vector Ξ_α and $\mathscr{H}_\alpha$, respectively.

T-splines should be normalized in order to form a partition of unity. This leads to



(a) (b) (c)

Figure 2.10. An example of T-mesh and inferring of local knot vectors for bicubic T-spline functions by traversing T-mesh edges.

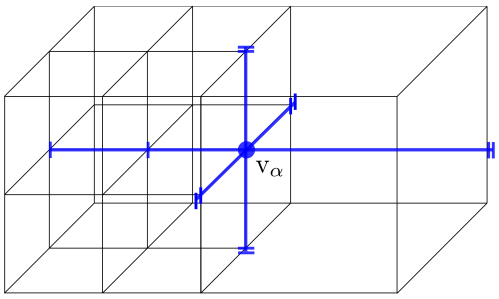


Figure 2.11. 3D T-mesh and inferring local knot vectors for a trivariate T-spline function.

rational blending functions:

$$R_\alpha(\xi,\eta) = \frac{B_\alpha(\xi,\eta)}{\sum_{\beta\in A_T} B_\beta(\xi,\eta)},$$

where A_T is the index set of the basis spanned by the T-mesh T. A T-spline approximation is constructed as a linear combination of all blending functions: $S(\xi,\eta) = \sum_{\alpha\in A_T} P_\alpha R_\alpha(\xi,\eta)$. The rational T-spline functions are capable to reproduce a constant function, but, in general, cannot reproduce a polynomial of a higher order. That is, T-spline blending functions do not span a complete polynomial space.

Generalization of bivariate T-splines to 3D case is straightforward. Trivariate T-spline function are defined using three local knot vectors inferred by marching through the 3D T-mesh until intersect perpendicularly a mesh face, see Fig. 2.11. Then, the function is constructed as a product of three univariate B-spline functions: $B_\alpha(\xi,\eta,\zeta) = B[\Xi_\alpha](\xi)B[\mathscr{H}_\alpha](\eta)B[Z_\alpha](\zeta)$.

2.5.1 Quadtree and Octree structures for T-meshes

In practise, it is difficult to develop an efficient code to manage general T-meshes. A quadtree [58] is a tree data structure in which each tree node can be subdivided into four children. Its three-dimensional analogy is the octree, in which each node is subdivided into eight children. These structures are often used to partition the space by recursively subdividing it into 4 quadrants (or 8 octants), for example in 3D graphics and 3D game engines. Due to its simplicity, quadtree and octree structures are an attractive tool for performing adaptive refinement in IGA and geometric modelling. For example, we can start from an initial subdivision of the parametric space into 4 equal cells (or 8 in 3D case), and each cell is associated to a tree node. Then, we can subdivide a cell in four subcells. This refinement is represented internally as a tree node subdivision. The node that has children is called internal node and the node that does not have children is called leaf node. Each leaf node of the quadtree represents a cell of the T-mesh, see Fig. 2.12. The resulting mesh has a hierarchy, where each cell has certain level k associated to it, depending on the level of refinement that produces

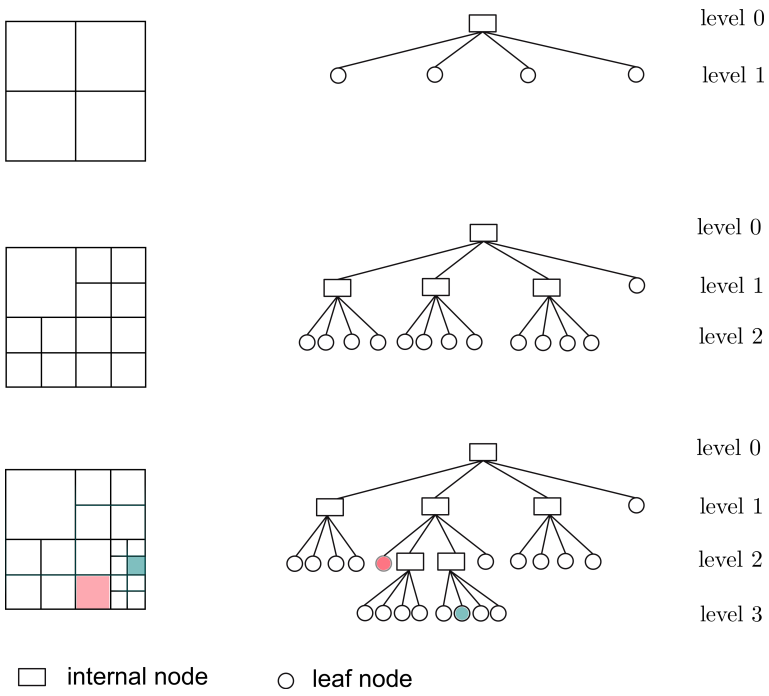


Figure 2.12. Hierarchical T-mesh implementation using quadtree data structure.

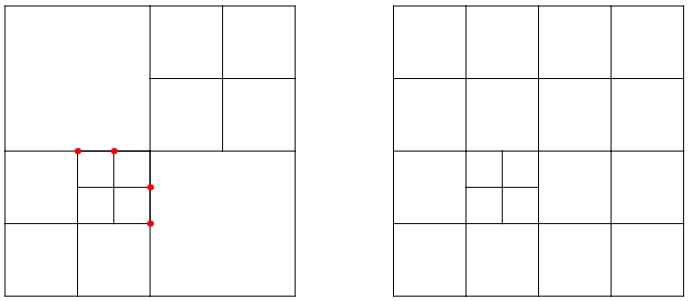


Figure 2.13. Balancing procedure. (a) An unbalanced quadtree mesh with more than hanging node per edge (marked in red). (b) Resulting balanced mesh after a balancing procedure.

the cell. If an internal tree node belongs to level k, then its children belong to level $k+1$. The root node has level $k=0$. If a k-level leaf node is refined then it turns into a k-level internal node. A quadtree mesh $\boldsymbol{T}$ is said to be of level k if its smallest cell is of level k.

To guarantee a good quality of the approximation space constructed over a T-mesh, it is preferable to have a gradual transition from the coarse mesh to the finely refined

zone. That is why it is common to work with balanced quadtree and octree meshes, limiting the number of T-junctions per edge. For example, a mesh with tree structure is said to be 1-balanced if for any k, no cell at level k shares an edge with a cell at level greater than $k+1$. In other words, a 1-balanced quadtree mesh implies that any cell has contact (edge or face) only with cells that differ at most in one level of depth. An example of 1-balanced procedure quadtree is shown in Fig. 2.13. To obtain a balanced quadtree, a standard balancing procedure is applied. Note that refinements performed during the balancing procedure do not propagate, see [59].

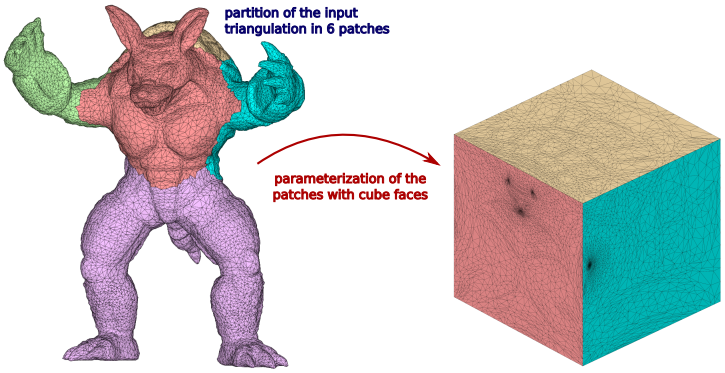
2.6 Volumetric parameterization based on the Meccano method

In previous works of the group [50, 51] a method to obtain a volumetric spline parameterization of a solid was developed. It is based on the idea of Meccano method [52] for automatic tetrahedral mesh generation, which, at the same time, obtains a one-to-one correspondence between the tetrahedral mesh of the parametric domain $\widehat{\Omega} = [0,1]^3$ and the physical domain Ω. This correspondence is obtained by deforming isomorphically the adapted tetrahedral mesh of the parametric domain into the physical mesh of the solid. The deformation is performed by applying a tetrahedral mesh untangling and smoothing procedure, see [60]. The deformation only affects the positions of the nodes, that is, there is not any change in their connectivity: we say that both meshes are isomorphic. Given that any 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 parametric tetrahedral mesh and the physical mesh of the solid, assuming that the barycentric coordinates are the same in both spaces. Thus, we obtain a volumetric parameterization of the solid. Figure 2.14 illustrates the main steps of Meccano method. Input triangulation of a solid is divided into six patches, see Fig. 2.14(a), that are mapped onto the cube faces via Floater parameterization [61]. Then, an adapted tetrahedral mesh of the cube is constructed using Kossaczký refinement until the geometry is approximated with a prescribed tolerance, see Fig. 2.14(b). As the result of this stage, the position of the boundary nodes in the physical domain are known and the position of the inner nodes will be defined by means of tetrahedral mesh optimization procedure. Figure 2.14(c) shows the adapted parametric mesh, the tangled physical mesh obtained after mapping the cube boundary to the Armadillo surface and the final optimized mesh. Optimization procedure pursues that each tetrahedron T of the physical mesh is as similar as possible to its counterpart (target) tetrahedron T_t in the parametric space.

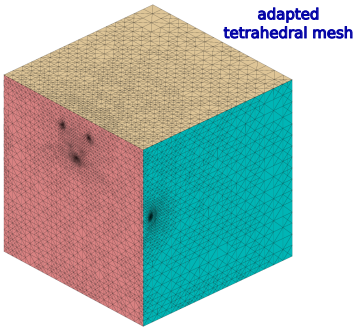
Meccano method was originally designed for automatic tetrahedral mesh generation, however, at the same time it provides directly a volumetric parameterization of the solid, which can be used for construction of a spline parametrization of the object. For that, a parametric T-mesh of $\widehat{\Omega}$, with a similar resolution that the tetrahedral mesh, is constructed. Trivariate spline mapping $\mathbf{S} : \hat{\Omega} \to \Omega$ is constructed by imposing

interpolation conditions, where interpolation points are located in physical space via tetrahedral parameterization provided by Meccano method.

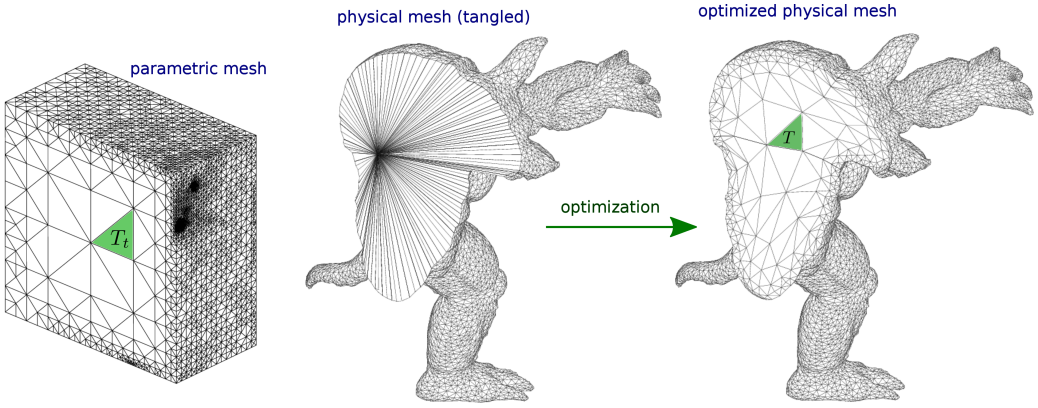
However, the procedure described is able to obtain an acceptable spline parameterization quality only for slightly distorted geometries. Meccano method achieves a good quality tetrahedral mesh, but it may not be sufficient for constructing high quality trivariate spline mapping, since tetrahedral mesh does not take into account the necessities of tensor product structure of spline mapping. High quality spline mapping implies a good orthogonality and uniformity of isoparametric curves, which is not achieved using the tetrahedral volumetric parameterization. This has motivated the necessity to extend the method and develop an optimization procedure directly for T-mesh. That is the main objective of this thesis.



(a)



(b)



(c)

Figure 2.14. Main steps of the Meccano method. (a) Surface paremeterization of the solid. (b) Adapted tetrahedral mesh of the cube (Kossaczký refinement). (c) Optimization of tetrahedral physical mesh.

3. Method for T-spline parameterization of complex 2D geometries

In this chapter we present a strategy, based on the idea of the meccano method and a novel T-mesh optimization procedure, to construct a T-spline parameterization of 2D geometries for the application of Isogeometric Analysis.

The proposed method only demands a boundary representation of the geometry as input data. The algorithm obtains, as a result, high quality parametric transformation between 2D objects and the parametric domain, the unit square. First, we define a parametric mapping between the input boundary of the object and the boundary of the parametric domain. Then, we build a T-mesh adapted to the geometric singularities of the domain in order to preserve the features of the object boundary with a desired tolerance. The key of the method lies in defining an isomorphic transformation between the parametric and physical T-mesh finding the optimal position of the interior nodes by applying a new T-mesh untangling and smoothing procedure. Bivariate T-spline representation is calculated by imposing the interpolation conditions on points sited both on the interior and on the boundary of the geometry. Proposed method also permits the modeling of objects with embedded geometries that can be used to solve problems with domains composed of several materials.

In previous works of the group [50, 51], a volumetric parameterization was obtained by deforming a tetrahedral mesh of the solid. In general, this approach does not provide an optimal T-mesh quality in the sense of its uniformity and orthogonality. Now, we propose a different approach where the optimization is applied directly to the T-mesh. We use an interpolation scheme to fit a T-spline object to the data instead of an approximation, as other authors do [22, 24]. This performs a more accurate adaptation of the T-spline to the input data.

This chapter is organized as follows. In next section we describe the main steps of the proposed algorithm. In section 3.2 we explain the process of boundary parameterization and the construction of T-mesh adapted to the singularities of the object boundary. Section 3.3 describes the simultaneous T-mesh untangling and smoothing procedure that leads to the construction of a high quality T-mesh of the object. The modeling of the geometry by means of bivariate T-splines is developed in section 3.4. A quality improvement strategy, based on adaptive refinement, is described in the section 3.5. Construction of embedded geometries is described in section 3.6. Finally, in section 3.7 we present the conclusions and set out next challenges.

3.1 General scheme of the method

In this section we summarize the proposed method. Some ideas are taken from previous works of the group on mesh untangling and smoothing and the Meccano method [62, 63, 60, 64, 65, 52], but they have been adapted to the requirements of the present work.

The algorithm includes the following stages:

1. Boundary parameterization and construction of an adapted T-mesh: A bijective correspondence between the input boundary of the object and the boundary of the parametric domain is defined. Then, an adapted T-mesh is generated by

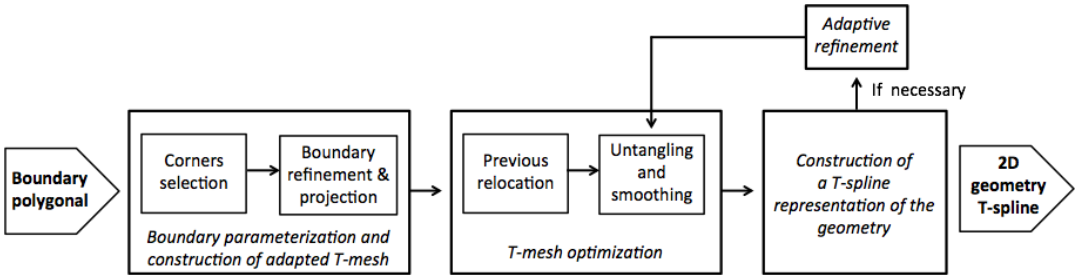


Figure 3.1. General scheme of the T-spline parameterization method.

refining the initial mesh in order to approximate the geometry with a prescribed tolerance. During this process, the boundary nodes of the parametric domain are mapped to the boundary of the object.

2. T-mesh optimization: We relocate the inner nodes of the T-mesh by applying a simultaneous mesh untangling and smoothing procedure. A previous relocation of the inner nodes is accomplished in order to facilitate this task.

3. Construction of a T-spline representation of the geometry: The T-spline parameterization is obtained by imposing interpolation conditions. As interpolation points, we take the vertices of the physical T-mesh obtained after the optimization process and other additional points.

4. Adaptive refinement to improve the parameterization quality: If the quality of the parameterization is not satisfactory, we apply an adaptive refinement in order to increase the degree of freedom in the areas with high distortion. Then, we return to step 2 and repeat the process until reaching a good T-spline parameterization.

The T-mesh used in this strategy has a balanced quadtree structure [58]: all refinements are performed by dividing a cell into 4 equal cells and a procedure of balancing is applied in order to allow just one hanging node per edge. The input boundary representation is given by a polygonal, however it could be given by B-spline curves.

A scheme of the algorithm is shown in Fig. 3.1. In the next sections we describe the details of each stage.

3.2 Boundary parameterization and construction of an adapted T-mesh

In order to define a parametric mapping between the input boundary polygonal of the object and the boundary of the parametric domain, the unit square, we have to select four points of the polygonal that will correspond to the four corners of the square.

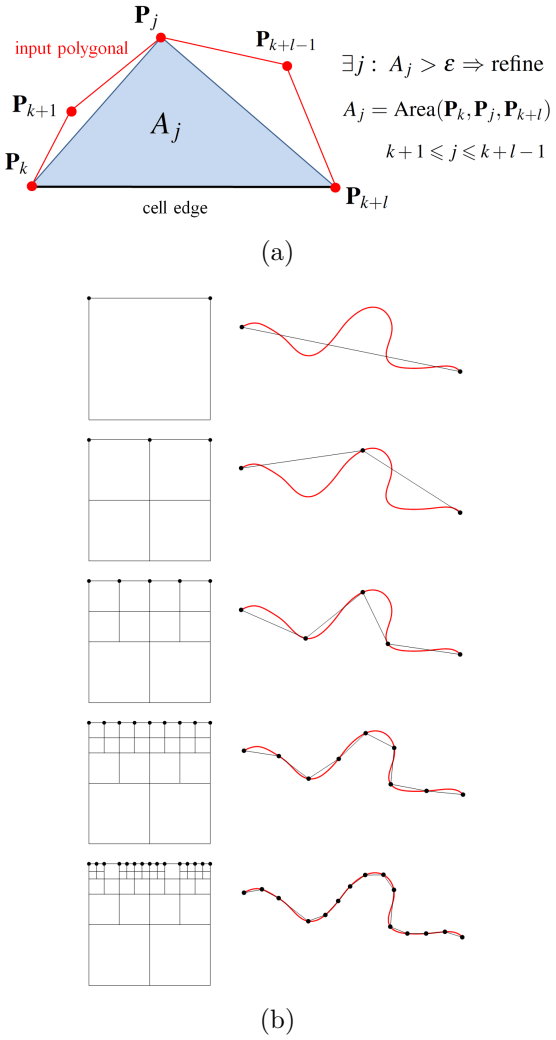


Figure 3.2. (a) Illustration of the refinement criterion used for the construction of an adapted T-mesh in order to approximate the object boundary with a desired tolerance. (b) Example of the boundary approximation corresponding to the upper edge of the unit square. Red line represents the input boundary.

These points divide the input polygonal into four parts that are mapped via chord-length parameterization into its corresponding edge of the square. The selection of the corners must be appropriate. Namely, the corners should be situated in convex areas of the input boundary polygonal, where the inner angle formed by the polygonal in these points is less than 180°. Obviously, the optimal value of this angle is 90°.

Next, we construct an adapted T-mesh that approximates the input boundary with a pre-defined tolerance ε. To do that, an approximation error is calculated for each boundary cell and the cell is refined if this error is greater than ε. A cell refinement produces a new boundary point that is projected over the input boundary polygonal,

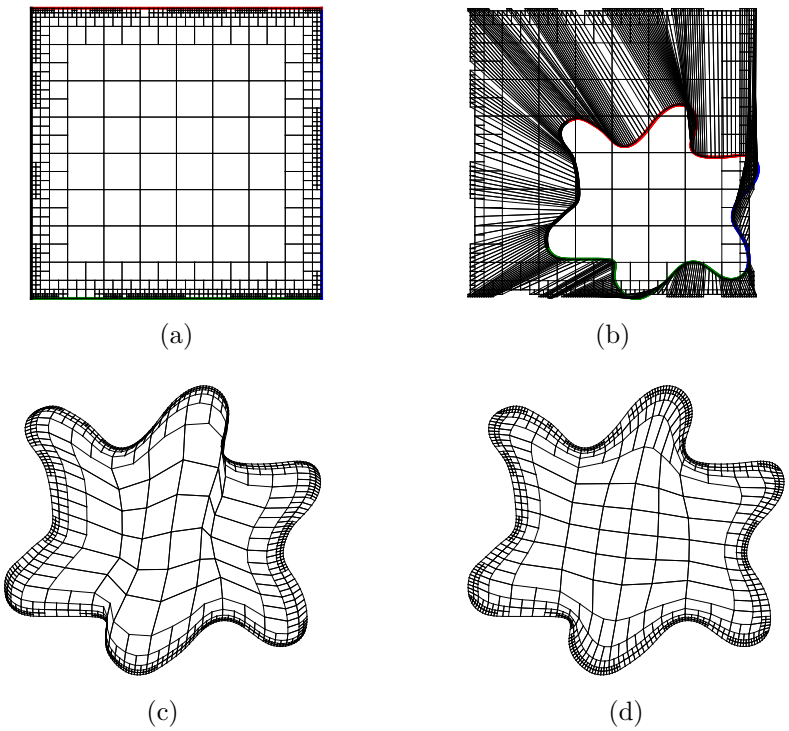


Figure 3.3. Stages of T-mesh construction for the Spot test model geometry. (a) Parametric T-mesh adapted to the boundary of the geometry; (b) tangled physical mesh after boundary mapping (the color bold lines represent the correspondence between parametric and physical boundaries); (c) previous relocation by using Coons patch; (d) optimized physical T-mesh.

obtaining a more accurate approximation of the geometry, as illustrated in Fig. 3.2(b). As approximation error, we evaluate the area of the triangles formed by the cell edge in the physical space and each vertex of the input polygonal approximated by this edge. A cell is refined if there is at least one triangle whose area exceeds $\boldsymbol{\varepsilon}$. The refinement criterion is illustrated in Fig. 3.2(a). Another approximation error criterion can be used, such as the Hausdorff distance between the input boundary polygonal and the cell edge.

Optionally, it is possible to start the boundary refinement and projection procedure with some initial refinement of the parametric domain in order to guarantee a sufficient number of free nodes in the interior of the geometry. This may facilitate the untangling procedure in complex geometries.

As result of this stage, the position of the boundary nodes in the physical domain are known and the position of the inner nodes will be defined by means of the T-mesh optimization procedure developed in next section. Figure 3.3(a) shows an example of the adapted parametric T-mesh constructed in this stage and, Fig. 3.3(b), illustrates the resulting tangled T-mesh of the physical domain that will be optimized.

3.3 T-mesh optimization

The key of the proposed method lies in the optimization procedure that allows to obtain a high quality physical T-mesh, that is used to construct the T-spline representation of the object.

3.3.1 Previous relocation

It is preferable to perform a previous relocation of the inner nodes in order to reduce the computational effort during the optimization process. We have used for this purpose Coons patch [66, 67] to define a surface that interpolates given boundary curves. Let the given boundary curves be called $\mathbf{x}(\xi,0)$, $\mathbf{x}(\xi,1)$, $\mathbf{x}(0,\eta)$, $\mathbf{x}(1,\eta)$. The Coons function is defined as

$$\begin{aligned}\mathbf{x}(\xi,\eta) =& (1-\xi)\mathbf{x}(0,\eta)+\xi\mathbf{x}(1,\eta)\\ &+(1-\eta)\mathbf{x}(\xi,0)+\eta\mathbf{x}(\xi,1)\\ &-\begin{bmatrix} 1-\xi & \xi \end{bmatrix}\begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1)\\ \mathbf{x}(1,0) & \mathbf{x}(1,1)\end{bmatrix}\begin{bmatrix} 1-\eta\\ \eta\end{bmatrix}\end{aligned}$$

This previous relocation procedure facilitates the untangling process, but in general does not obtain a satisfactory mesh quality and can produce self-intersections, as shown in Fig. 3.3(c). Therefore, it is essential to apply an efficient optimization algorithm.

3.3.2 Objective function

The mesh optimization process is carried out by iterative relocation of each inner node of the mesh in such a way that the new position of the node improves the quality of the local submesh corresponding to this node. A local submesh is the set of all the elements connected with the movable or free node. The local objective function for a free node is based on algebraic shape quality metrics proposed by Knupp in [68, 69] for triangular and quadrilateral elements. Shape quality metric for a given triangle is defined in terms of the Jacobian matrix of the affine mapping from ideal triangle to the given one. This shape quality metric represents the deviation of the physical triangle from the ideal one. It attains its maximum value, 1, if the triangle is similar to the ideal one, and it equals 0 if the triangle is degenerated. The distortion metric of an element is defined as the inverse of its quality metric. In order to asses the quality of the local submesh for a given free node of a T-mesh, we have to decompose each neighboring cell into triangles and asses the quality of each triangle. For a T-mesh, this decomposition depends on the type of the free node. There are two types of free node: a regular node and a hanging node. The optimal position of each free node is determined by minimizing a local objective function. We define the objective function as a sum of shape distortion metrics of the triangles of the local submesh. For each triangle of the physical mesh, the corresponding triangle of the parametric mesh is used as its ideal element. Therefore, each cell of the physical mesh tends to have the same

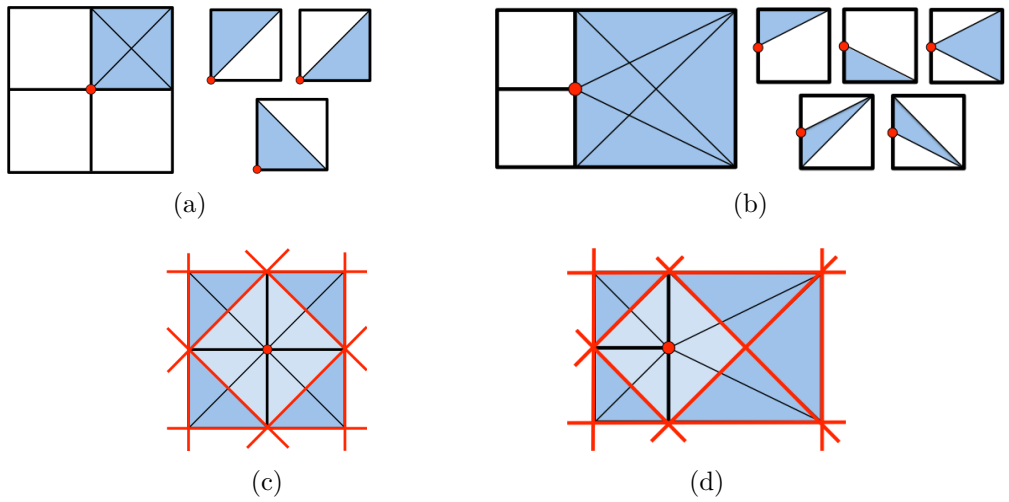


Figure 3.4. Triangular decomposition of the local submesh. (a) Regular node case, where each cell is decomposed in three triangles; (b) hanging node case, where five triangles are formed in the cell where the node generates a T-junction; (c) barriers and feasible region induced by the 12 triangles in the objective function for a regular node; (d) barriers and feasible region induced by the 11 triangles in the objective function for a hanging node.

shape as its counterpart cell of the parametric mesh. Thus, repeating this procedure for all the inner nodes of the mesh, we achieve the physical mesh of the object be as similar to the parametric one as possible.

A regular node is surrounded by four cells with equal or different sizes. In order to perform the mesh improvement, the local submesh is decomposed in twelve triangles, three triangles per cell whose qualities depends on the position of the free node. Figure 3.4(a) illustrates this decomposition and Fig. 3.4(c) shows the feasible region of the objective function.

In a hanging node case, the free node is surrounded by three cells and the local submesh is decomposed in eleven triangles. The cell in which the node forms a T-junction is decomposed in five triangles whose qualities depend on the position of the free node. Each one of the other two cells is decomposed in three triangles, as was described in the case of a regular node. Figure 3.4(b) shows the decomposition of a local submesh for hanging node case and the feasible region of the objective function. Note that, for the ideal case shown in Fig. 3.4(d), the feasible region induced by these eleven triangles is the same as the one obtained after a refinement of the T-junction cell, see Fig. 3.4(c).

In order to define the objective function we introduce the following concepts.

Let T be a triangle whose vertices are given by $\mathbf{x}_k = (x_k, y_k)^T \in \mathbb{R}^2$, $k = 0,1,2$ and T_R be the reference triangle with vertices $\mathbf{u}_0 = (0,0)^T$, $\mathbf{u}_1 = (1,0)^T$ and $\mathbf{u}_2 = (0,1)^T$. 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 $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0)$.

Let consider that T_I is our ideal or target triangle whose vertices are $\mathbf{v}_0$, $\mathbf{v}_1$ and $\mathbf{v}_2$. If we take $\mathbf{v}_0 = (0,0)^T$, the linear map that takes T_R to T_I is $\mathbf{v} = W\mathbf{u}$, where $W = (\mathbf{v}_1, \mathbf{v}_2)$ is its Jacobian matrix. As the parametric and real meshes are topologically identical, each triangle in the physical space has its counterpart in the parametric space.

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. Quality metrics of the triangle T cab be defined in terms of the matrix S. For example, the mean ratio, $q = \frac{2\sigma}{\|S\|^2}$, 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 triangle has quality measure zero. We can derive an optimization function from this quality metric. Thus, let $\mathbf{x} = (x,y)^T$ be the position of the free node, and let S_m be the weighted Jacobian matrix of the m-th triangle connected to this free node. We define the objective function of $\mathbf{x}$, associated to an m-th triangle as

$$\eta_m = \frac{\|S_m\|^2}{2\sigma_m} \tag{3.1}$$

The local objective function used for mesh quality improvement is defined by means of the inverse of mean ratio quality metric of each triangle of the local submesh. The function to be minimized is given by

$$K(\mathbf{x}) = \sum_{m=1}^{M} \frac{\|S_m\|^2}{2\,\sigma_m} \tag{3.2}$$

where M is the number of triangles in the local submesh and S_m is the Jacobian matrix associated to the affine mapping from the ideal triangle to the physical one.

Objective function defined by Eq. 3.2 is appropriate to improve the quality of a valid mesh, but it does not work properly when there are inverted elements [70, 71]. In previous works [60, 72] we have used a modified objective function K^*, where the untangling and smoothing are carried out in the same stage. This modified objective function K^* does not have singularities, it works as the original function K for the valid elements and tends to untangle the inverted and degenerated ones. This objective function is defined as

$$K^*(\mathbf{x}) = \sum_{m=1}^{M} \frac{\|S_m\|^2}{2h(\sigma_m)} \tag{3.3}$$

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

Objective function K^* is smooth in $\mathbb{R}^2$, so the unconstrained optimization problem can be easily solved with any standard method (see for example [73]).

In case of a conformal local submesh, the result obtained by minimizing the objective function K^* is, when possible, an orthogonal submesh, as shown in Fig. 3.5(a). However, not satisfactory result is obtained for a non-conformal submesh. In this case, two special situations can appear: a regular node surrounded by cells of different scales and a hanging node. In these situations, a variation in the position of the free node does not affect in the same way to the quality of the triangles of the local submesh. The objective function tends to form triangles as similar as possible to the reference ones, but the influence of the smaller cells are greater than the bigger ones. For example, in Fig. 3.5(b) it can be seen how the free node is moved toward the small cell and, therefore, the resulting mesh is not orthogonal.

This problem is solved by a modification of the objective function K^*, namely multiplying the terms of the objective function by appropriate weights.

3.3.3 Weighted objective function

The terms of the objective function K^* can be grouped according to the belonging to each cell of the local submesh. Each group is multiplied by an appropriated weight in order to avoid the problems mentioned in section 3.3.2.

For a regular node, the weighted objective function is

$$\begin{aligned} K^*_\tau(\mathbf{x}) = \tau_1 \sum_{m=1}^{3} \frac{\|S_m\|^2}{2h(\sigma_m)} + \tau_2 \sum_{m=4}^{6} \frac{\|S_m\|^2}{2h(\sigma_m)} + \\ + \tau_3 \sum_{m=7}^{9} \frac{\|S_m\|^2}{2h(\sigma_m)} + \tau_4 \sum_{m=10}^{12} \frac{\|S_m\|^2}{2h(\sigma_m)} \end{aligned} \tag{3.4}$$

where each summation is the group associated to each cell and τ_i is the applied weight. This weight is equal to the scale factor of the cell in the parametric space. We assume that the smallest cells in the local submesh have scale factor $\tau = 1$ and the other cells can have scale factor $\tau = 2$ or $\tau = 4$, as illustrated in Fig. 3.6(a). Figure 3.5(c) shows the resulting orthogonal mesh when these weights are applied.

A hanging node is surrounded by three cells as it was mentioned above. In this case, the weighted objective functions is

$$K^*_\tau(\mathbf{x}) = \tau_1 \sum_{m=1}^{3} \frac{\|S_m\|^2}{2h(\sigma_m)} + \tau_2 \sum_{m=4}^{6} \frac{\|S_m\|^2}{2h(\sigma_m)} + \tau_3 \sum_{m=7}^{11} \frac{\|S_m\|^2}{2h(\sigma_m)} \tag{3.5}$$

Hanging node is a more particular case because its local submesh is decomposed in different types of triangles. To guarantee the orthogonality in the local submesh after optimization, we have determined that the weights are $\tau_3 = \frac{8}{5}$ for the cell where the node forms a T-junction and $\tau_1 = \tau_2 = 1$ for the other two cells, as shown in Fig. 3.6(b). The election of these weights is justified in [53].

Two different options, weighted and no weighted objective functions, are available for T-mesh optimization, that should be chosen by the user according to necessities

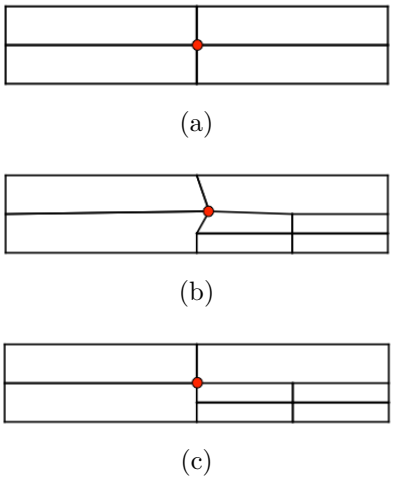


Figure 3.5. Resulting meshes after optimization with different objective functions. (a) Orthogonal mesh using K^*; (b) not satisfactory result for a non-conformal local submesh using K^*; (c) orthogonal mesh using weighted objective function K^*_τ.

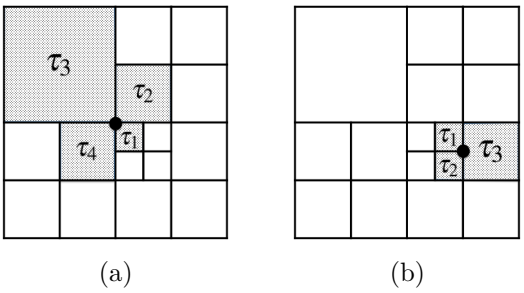
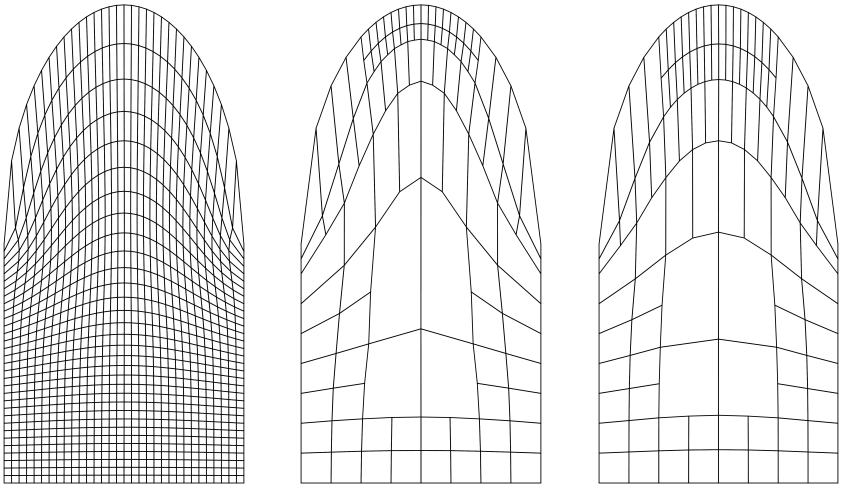


Figure 3.6. Weights applied to the objective function K^*_τ. (a) Regular node where $\tau_1 = 1$, $\tau_2 = \tau_4 = 2$ and $\tau_3 = 4$; (b) hanging node where $\tau_1 = \tau_2 = 1$ and $\tau_3 = \frac{8}{5}$.

of the problem to solve. Optimization process with weighted objective function obtains a T-mesh as orthogonal and similar to a regular mesh as possible. However, no weighted objective function does not produce T-meshes so orthogonal, but it provides more flexibility in order to facilitates the untangling process in very complicated geometries. On the other hand, the quality of the mesh tends to be more uniform when it is optimized with the weighted objective function. Figure 3.7 shows the effects of optimizing a T-mesh without weights (b) and with weights (c) in comparison with an optimized regular mesh (a).



(a) (b) (c)

Figure 3.7. Comparison of optimization procedure with weighted and no weighted objective functions. (a) Regular mesh; (b) T-mesh optimized without weights; (c) the same T-mesh optimized with weights. Note that the positions of the nodes in case (c) are closer to the node positions of the regular mesh than in case (b).

3.4 Construction of a T-spline representation of the geometry

In this section we describe the construction of T-spline representation for 2D geometries. We have to obtain a global one-to-one parametric transformation that maps the parametric domain into the physical domain $\mathbf{S} : \hat{\Omega} = [0,1]^2 \rightarrow \Omega$, see Fig. 3.8.

We build bivariate T-spline surface representation of our physical domain as lineal combination of T-spline blending functions

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha(\boldsymbol{\xi}) \tag{3.6}$$

where $\mathbf{P}_\alpha \in \mathbb{R}^2$ is the control point corresponding to the α-th blending function. The T-spline blending functions that we use in this work are rational B-spline functions defined as

$$R_\alpha(\boldsymbol{\xi}) = \frac{N_\alpha(\boldsymbol{\xi})}{\sum_{\beta \in A} N_\beta(\boldsymbol{\xi})} \tag{3.7}$$

being $N_\alpha(\boldsymbol{\xi}) = N^1_\alpha(\xi) N^2_\alpha(\eta)$ the bivariate B-spline function defined over its local knot vectors $\Xi_\alpha = \{\Xi_\alpha, H_\alpha\}$, and A is the index set of the basis spanned by T-mesh. Control points $\mathbf{P}_\alpha$ are found by imposing interpolation conditions. Assuming that the set of

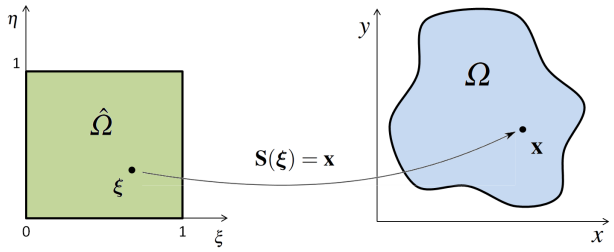


Figure 3.8. One-to-one global mapping between parametric domain $\hat{\Omega}$ and physical domain Ω.

blending functions are linearly independent, we need as many interpolation points as blending functions.

As interpolation points, first of all, we use the anchors of the blending functions. Each anchor coincides with a T-mesh vertex, $\boldsymbol{\xi}_\alpha^v$, and its position in the physical space, $\mathbf{x}_\alpha^v$, was determined by the mesh optimization process.

As we are dealing with an open knot vector structure along the boundary of the mesh, the total number of blending functions is greater than the number of vertices of the T-mesh. Therefore, we have to take additional interpolation points. These additional points are associated to functions whose knot vectors Ξ_α or H_α contains exactly three repeated knots. We assign to each of these functions an interpolation point approximately sited where the function attains its maximum. The positions of these interpolation points in parametric and physical spaces are shown in Fig. 3.9(a). For each additional interpolation point $\boldsymbol{\xi}_\alpha$, its position in the physical space, $\mathbf{x}_\alpha$, should be defined. The interpolation points $\boldsymbol{\xi}_\alpha^e$, situated at the mid point of an edge of the parametric mesh, are mapped to the mid point of the same edge of the physical T-mesh. The positions in the physical space of the four interpolation points $\boldsymbol{\xi}_\alpha^c$ situated in the center of the corner cells of the parametric mesh are determined by optimizing the local submesh formed after a fictitious quadtree partition of these cells.

Finally we solve the linear system of equations

$$\mathbf{x}_\beta = \mathbf{S}\left(\boldsymbol{\xi}_\beta\right) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha\left(\boldsymbol{\xi}_\beta\right), \quad \forall \boldsymbol{\xi}_\beta,\ \beta \in A \tag{3.8}$$

where $\boldsymbol{\xi}_\beta$ are interpolation points in parametric space and $\mathbf{x}_\beta$ are their images in the physical space.

3.5 Adaptive refinement to improve the quality

3.5.1 Mean ratio Jacobian

Our objective is to get high-quality geometry parameterization suitable for isogeometric analysis. The parametric T-spline mapping of Eq. 6.4.1 is suitable for analysis if it

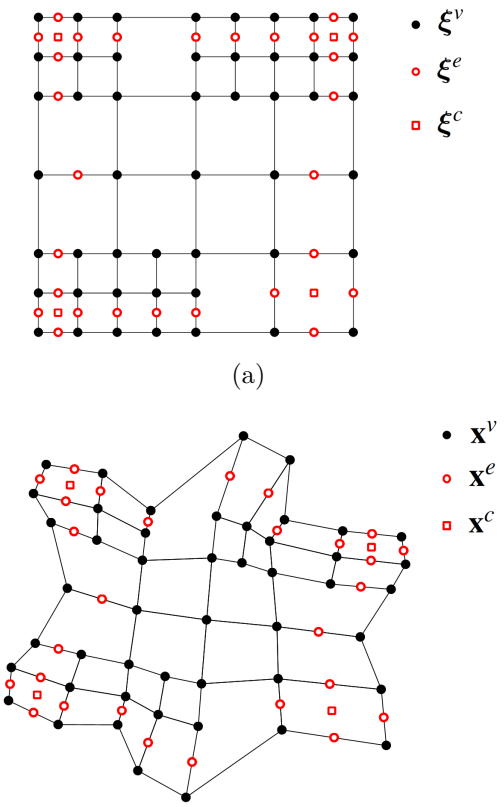
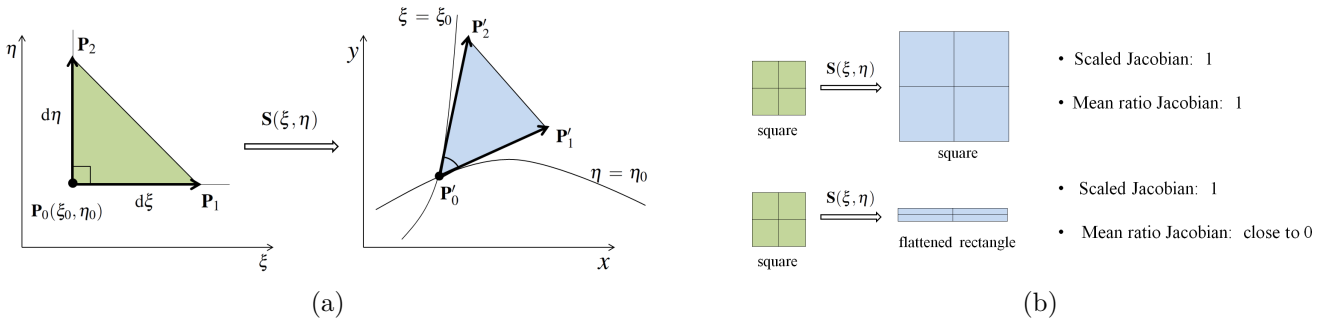


Figure 3.9. Interpolation points in parametric and physical domains. (a) Parametric domain, $\boldsymbol{\xi}^v$ - vertices of the mesh, $\boldsymbol{\xi}^e$ - additional interpolation points situated on the edges of the mesh, $\boldsymbol{\xi}^c$ - additional points situated in the four corner cells; (b) interpolation points in physical domain.

has positive Jacobian in all the domain. High distortion of the geometry can produce a large variation of the Jacobian that can lead to a poor accuracy in the numerical results. Therefore, a good uniformity and orthogonality of the isoparametric curves are desired for the parametric mapping $\mathbf{S}$. A high quality of the optimized T-mesh is a necessary, but not sufficient, condition for a high quality of the T-spline mapping. It can happen that the Jacobian of the spline parameterization takes negative values even if all the cells of the T-mesh are valid. In order to assess the quality of the constructed parametric transformation we analyze the mean ratio Jacobian, given by

$$q_s(\boldsymbol{\xi}) = \frac{2\det(\mathbf{J}_S)}{\|\mathbf{J}_S\|^2}, \tag{3.9}$$

where $\mathbf{J}_S$ is the Jacobian matrix of the mapping $\mathbf{S}$ at the point $\boldsymbol{\xi} = (\xi, \eta)$ and $\|\mathbf{J}_S\|$ is its Frobenius norm.



(a) (b)

Figure 3.10. Mean ratio Jacobian. (a) A quality metric of the parametric mapping $\mathbf{S}$ at any point $\mathbf{P}_0$ in terms of the mean ratio of the triangle $\mathbf{P}'_0\mathbf{P}'_1\mathbf{P}'_2$. (b) Comparison between the mean ratio Jacobian and the scaled Jacobian.

The value of the mean ratio Jacobian at any point $\mathbf{P}_0$ of the parametric domain is a shape quality metric for the infinitesimal triangle formed by two isoparametric curves of the physical domain passing through the point $\mathbf{P}'_0 = \mathbf{S}(\mathbf{P}_0)$, as illustrated in Fig. 3.10(a). In contrast to the scaled Jacobian, that represents a quality of the mapping $\mathbf{S}$ in the sense of the orthogonality of its isoparametric curves, the mean ratio Jacobian represents both: a quality of the mapping in the sense of the orthogonality and uniformity of its isoparametric curves. Figure 3.10(b) shows the comparison between the scaled Jacobian and the mean ratio Jacobian. Scaled Jacobian attains its maximum value 1 at the given point if the mapping conserves orthogonality of the isoparametric curves. Mean ratio Jacobian is equal 1 at the point $\mathbf{P}_0$ if the mapping conserves orthogonality and produces the same length distortion in both parametric directions, i.e., the mapping is conformal at this point.

It is easy to see that $\forall \boldsymbol{\xi}: \quad 0 \leq |q_s(\boldsymbol{\xi})| \leq |q_{sc}(\boldsymbol{\xi})| \leq 1$, where $q_{sc} = \frac{\det(\mathbf{J}_S)}{\|\mathbf{S}_\xi\| \|\mathbf{S}_\eta\|}$ is the scaled Jacobian.

3.5.2 Adaptive refinement

Parameterization of complex geometries entails a severe distortion that can lead to appearance of low quality cells, even cells with negative Jacobian. This can be explained by the lack of degrees of freedom provided by the inner nodes. In order to improve the mesh quality in this case, we propose an adaptive strategy that refines all the cells with low quality. A similar idea was implemented for tetrahedral meshes in [74].

We proceed as follows. For each cell of the mesh, the mean ratio Jacobian of the T-spline parameterization is calculated at Gauss quadrature points. We use $16 = 4 \times 4$ quadrature points per cell. A cell $\hat{\Omega}_e$ is marked to refine if, at least, one of its quadrature points has mean ratio Jacobian less than a certain threshold δ. The refined T-mesh is optimized again and the process is repeated until a satisfactory quality is obtained. Figure 3.11 illustrates the efficacy of the proposed strategy. Additional refinements were applied to Gran Canaria Island domain with $\delta = 0.2$. The initial mesh with 3439 cells

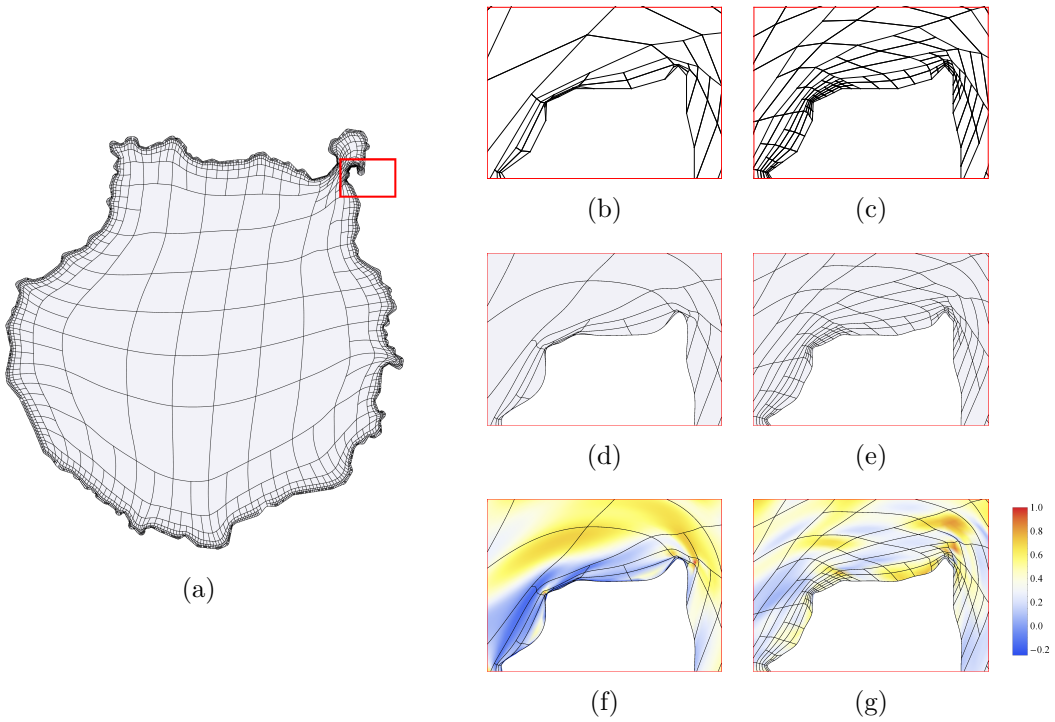


Figure 3.11. Adaptive refinement strategy to improve the parametric transformation quality in Gran Canaria island domain. (a) T-spline representation of the domain; (b) initial physical T-mesh; (c) final physical T-mesh; (d) initial T-spline parametrization with negative Jacobian; (e) resulting T-spline parameterization with no negative Jacobian after applying adaptive refinement; (f) mean ratio Jacobian of the initial parametrization; (g) mean ratio Jacobian of the final parametrization.

produces a T-spline parametric mapping with low quality in some areas and negative Jacobian in the North East part of the island. After adaptive refinement we have a mesh with 3577 cells and positive Jacobian in all the domain. Moreover, the minimum value of the mean ratio Jacobian at the quadrature points is 0.21.

In most cases, adaptive refinements only affect to a localized area of the geometry. A good strategy to optimize the mesh after adaptive refinement is to apply the optimization algorithm only to refined areas. Initially it is marked to optimize a list of nodes composed by the nodes of the new cells created in the refinement and the nodes that conform the local submesh of each of them. After optimization, if the position of the node does not change significantly, the node is extracted of the list. Otherwise, the node remains in the list and its local submesh is added. The process finalizes when the list of nodes is empty. This strategy avoids unnecessary iterations in zones where the optimization process does not produce relevant changes in the mesh. This accelerates the optimization after each adaptive refinement.

3.6 Parameterization of embedded geometries

The proposed method can be easily extended to parameterize embedded geometries. Embedded geometries are composed by the insertion of individuals figures into another one. Each of these figures can be parameterize individually with the method seen until now, obtaining the parametric and physical T-meshes of each one. The key to parameterize the composed geometry is to build a global parametric space based on the individual ones. In our method, the parametric space has a quadtree structure. A cell of a quadtree could be seen as the root of a new quadtree. This allows to insert the quadtree of an inner geometry in a cell or set of cells belonging to an outer quadtree.

After the insertion, we obtain a new full quadtree for the global geometry. This new quadtree is not balanced, so it is necessary to apply the 2:1 balance algorithm after the quadtree merging. The untangling and smoothing procedure is carried out with this new balanced quadtree, starting at the node positions obtained in the individual construction of each figure. In this untangling step, the nodes of the inner boundaries are fixed in the same manner that the outer boundary. Figure 3.12 describes the process to build a T-mesh with embedded geometries. The rest of the process to construct the T-spline representation of the geometry and the adaptive refinement is exactly the same that for a basic geometry.

With this strategy it is possible to insert any number of geometries into another one. Moreover, we can build geometries with holes, simply deleting the cells corresponding to an inner geometry.

3.7 Conclusions and challenges

We have proposed a new technique for obtaining a single T-spline parameterization of 2D geometries for the application of isogeometric analysis. A new T-mesh untangling and smoothing procedure have been applied in order to define an isomorphic transformation between parametric and physical T-meshes. Presented technique is simple and easy to implement. The algorithm have been tested in several 2D geometries as it is shown in Results and applications chapter. For all cases, we have obtained a high quality parametric transformation between the object and the parametric domain. To asses the quality of the parametric mapping, we evaluate its mean ratio Jacobian. Thereby, we detect the areas with low quality and perform an adaptive refinement in order to increase the degree of freedom in the areas with high distortion. This strategy allows to obtain a parameterization suitable for analysis with no negative Jacobian, even for complex geometries. Moreover, we have proposed an extension of the method to parameterize embedded geometries.

In next chapters, we extend the present method to deal with 3D geometries. To do that, the main challenge is to develop an optimization procedure for 3D T-meshes. We first study several options to optimize a hexahedral mesh, and then, we generalize the parameterization method to 3D.

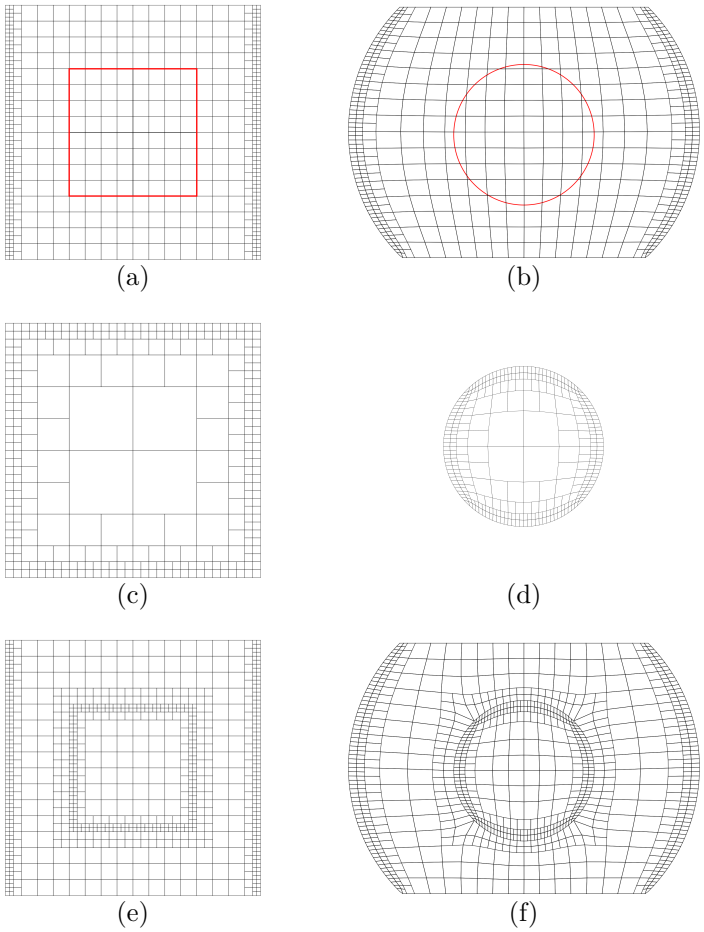


Figure 3.12. T-mesh construction of an embedded geometry. (a) Parametric space of the outer geometry. Red lines define the area where the inner parametric space will be inserted; (b) T-mesh of the outer geometry. The position of the inner boundary is marked in red; (c) parametric space of the inner geometry; (d) T-mesh of the inner geometry; (e) resulting parametric space after inserting the inner one into a region of the outer one, with a subsequent 2:1 balance; (f) T-mesh of the global geometry after optimization.

4. Strategies for optimization of hexahedral meshes

In previous chapter we have proposed a parameterization method for 2D geometries. The main challenge for the extension of this strategy to deal with 3D geometries is to develop an optimization method for 3D T-meshes. The appropriated simplex decomposition in 3D is not evident. In this case, a cell of a balanced T-mesh can be formed by 26 nodes and there are multiple possibilities to decompose a cell into tetrahedra. Before to propose a solution for 3D T-mesh optimization, we first focus to study the optimization problem for hexahedral meshes. This chapter is the result of that research. Some of the conclusions of this study help us to design a solution for the 3D T-mesh optimization problem, as it is shown in the following chapters.

Here we study several strategies, based on different objective functions, for optimization of hexahedral meshes. We consider two approaches to construct objective functions. The first one is based on the decomposition of a hexahedron into tetrahedra. The second one is derived from the Jacobian matrix of the trilinear mapping between the reference and physical hexahedral element. A detailed description of all proposed strategies is given. Some computational experiments have been developed to test and compare the untangling capabilities of the considered objective functions. In the experiments a sample of highly distorted hexahedral elements is optimized with the proposed objective functions and the rate of success of each function is obtained. The results of these experiments are presented and analyzed.

4.1 Introduction

For many simulation processes hexahedral meshes present several numerical advantages over tetrahedral ones. For example, in Finite Element Method simulations, hexahedral meshes are best suited to solve elastic, structural or fluid mechanics problems [75, 76, 77].

The quality of the mesh has high repercussion on the numerical behaviour of FEM. In some cases, mesh generators construct poor quality meshes or, even worse, with inverted elements. Indeed, a good quality mesh could degenerate into a mesh with inverted elements when the simulation requires nodes movement. For this reason, it is of major importance to apply a mesh optimization algorithm to untangle the inverted elements and increase the quality of the mesh.

A widely used technique for mesh optimization consists in an iterative process in which each node is moved to a new position in order to improve the quality of the local mesh. This new position of the node is determined by minimizing certain objective function based on a distortion measure of the elements of the local mesh. In 2D, the optimization of a first order quadrilateral element is carried out by decomposing the element in four triangles [68]. The non-degeneracy of these triangles is a necessary and sufficient condition for the validity of a bilinear element. However, for 3D case, necessary and sufficient conditions for the validity of a trilinear element are not known. Hence, it is not clear how to construct an objective function for hexahedral mesh optimization. In this work we study several options for this issue. On the one hand, we form the objective function through the decomposition of the hexahedral elements

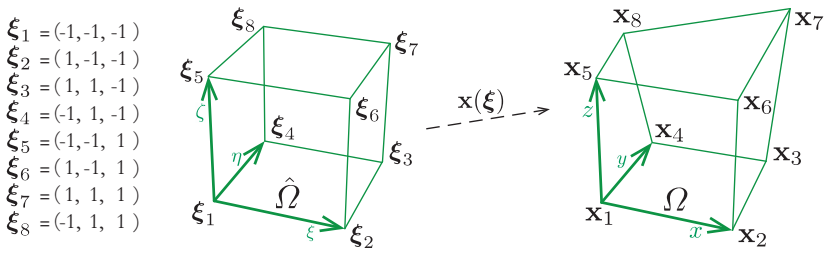


Figure 4.1. Master element $\hat{\Omega}$ and physical element Ω.

into tetrahedra [69, 78, 79, 80]. On the other hand, the objective function is built up from a shape distortion measure derived from the trilinear mapping between the reference and physical hexahedron [81].

Although the analyzed objective functions are able to optimize a large variety of meshes, some of them could not work properly for highly distorted elements. We have designed two computational experiments to compare the untangling capabilities of the proposed objective functions. In the experiments a sample of highly distorted hexahedral elements is optimized with the studied objective functions and the rate of success of each function is obtained.

This chapter is organized as follows. In section 2 some preliminary concepts are given. The construction of objective functions through the decomposition of the element into tetrahedra is explained in section 3. In section 4 we propose an objective function constructed by using a shape distortion measure derived from the trilinear mapping. In section 5 the experiments are explained and the results are discussed in detail. Finally, conclusions are given in section 6.

4.2 Preliminary concepts

In this comparative study we will consider conformal meshes formed by first order hexahedral elements defined by the trilinear transformation

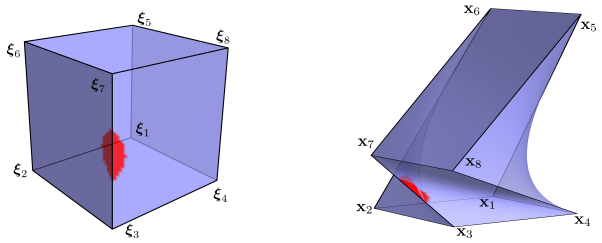
$$\mathbf{x}(\boldsymbol{\xi}) = (x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)) = \sum_{i=1}^{8} \mathbf{x}_i N_i(\boldsymbol{\xi}), \tag{4.1}$$

that maps the master element $\hat{\Omega} = [-1,1]^3$ with vertexes $\boldsymbol{\xi}_i$ into the physical hexahedron Ω with vertexes $\mathbf{x}_i \in \mathbb{R}^3$, see Fig. 4.1, where N_i are the trilinear shape functions

$$N_i(\boldsymbol{\xi}) = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i),$$

that verify $N_i(\boldsymbol{\xi}_j) = \delta_{ij}$.

Here, we consider that a hexahedral element is valid if its Jacobian determinant is strictly positive, i.e., $J(\boldsymbol{\xi}) > 0$, $\boldsymbol{\xi} \in [-1,1]^3$, where $J(\boldsymbol{\xi}) = \det(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}})$. Thus, we assume that during the mesh generation process the positivity of the Jacobian determinant



(a) (b)

Figure 4.2. Example of an invalid element with positive Jacobian in all corner points. (a) Master element, (b) physical element. Red area indicates the region of negative Jacobian.

of the trilinear mapping is pursued. We say that an element is invalid if its Jacobian takes negative values or equals zero at some point of the element. Sometimes we refer to the invalid element also as tangled element or degenerate element.

It is known that, unlike 2D bilinear element, the positivity of the Jacobian at the corner points does not guarantee the validity of the trilinear element. Moreover, Jacobian positivity along the edges is not sufficient for its positivity in the whole element, see [82]. Figure 4.2 shows an example of trilinear element with strictly positive Jacobian at all 8 corners. However, its Jacobian takes negative values in some points of the element. According to the notation of Fig. 4.1, the coordinates of the vertexes of the physical element are $\mathbf{x}_1 = (-1,-1,-1)$, $\mathbf{x}_2 = (1,-1,-1)$, $\mathbf{x}_3 = (1,1,-1)$, $\mathbf{x}_4 = (-1,1,-1)$, $\mathbf{x}_5 = (-1,1,6)$, $\mathbf{x}_6 = (-1,-1,6)$, $\mathbf{x}_7 = (1,-1,1)$ and $\mathbf{x}_8 = (1,1,1)$.

Since necessary and sufficient conditions for the validity of a trilinear element are not known, it is not clear how to construct an objective function for hexahedral mesh optimization. There are some sufficient conditions for validity of a trilinear element, that can be used for mesh optimization. Also, mesh optimization can be based on some necessary conditions for validity of a trilinear element. The objective of the present work is to analyze and compare the options for hexahedral mesh optimization.

4.3 Optimization based on tetrahedral decomposition of a hexahedron

Hexahedral mesh optimization can be converted into a tetrahedral optimization problem by decomposing each hexahedron in a set of tetrahedra. However, there is not a set of necessary and sufficient conditions, based on tetrahedra decomposition, that guarantees the validity of a hexahedron. In this section we first introduce the quality and distortion measures for a tetrahedron. Then, different objective functions for hexahedral mesh optimization are defined by imposing sufficient or necessary conditions for the validity of a hexahedron.

4.3.1 Quality and distortion measures for tetrahedra

Here we summarize the formulation of the algebraic quality and distortion measures proposed by Knupp [68] for a tetrahedron. These measures are defined in terms of the deviation from an ideal tetrahedron that represents the desired shape to achieve.

Let τ be a tetrahedron whose vertexes are given by $\mathbf{x}_k = (x_k, y_k, z_k)^T \in \mathbb{R}^3$, $k = 0, \ldots, 3$ and τ_R be the reference tetrahedron with vertexes $\mathbf{u}_0 = (0,0,0)^T$, $\mathbf{u}_1 = (1,0,0)^T$, $\mathbf{u}_2 = (0,1,0)^T$ and $\mathbf{u}_3 = (0,0,1)^T$. If we choose $\mathbf{x}_0$ as the translation vector, the affine map that takes τ_R to τ 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 τ_I is our ideal or target tetrahedron whose vertexes 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 τ_R to τ_I is $\mathbf{v} = W\mathbf{u}$, where $W = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is its Jacobian matrix.

Affine map that takes τ_I to τ is given by $\mathbf{x} = AW^{-1}\mathbf{v} + \mathbf{x}_0$, and its Jacobian matrix is $S = AW^{-1}$. Quality metrics of the tetrahedron τ can be defined in terms of the matrix S. For example, the mean ratio

$$q = \frac{3\sigma^{2/3}}{\|S\|^2},$$

is an easily computable algebraic quality metric of τ, 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 τ and τ_I are similar tetrahedra. Besides, any flat tetrahedron has quality measure zero.

The distortion measure for a tetrahedron is defined as the inverse of its quality, i.e., $\eta = 1/q$. The distortion η is equal to 1 for the ideal tetrahedron and tends to ∞ when the tetrahedron tends to be degenerated. The objective function to be minimized is defined as $K = \sum_{i=1}^{N} \eta_i^p$, where N is the number of tetrahedra in the local mesh and, usually, $p = 1$ or $p = 2$.

The objective function K becomes discontinuous when the volume of any tetrahedron tends to zero. Due to these singularities, the function K improves the quality of valid elements but it does not work properly when the mesh is tangled ($\sigma \leq 0$). In [60] we proposed a modification of K by replacing σ by the positive and increasing function $h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2})$. Then, the modified distortion becomes

$$\eta^* = \frac{\|S\|^2}{3h(\sigma)^{2/3}}. \tag{4.2}$$

This modification eliminates the barriers associated with their singularities and the new objective function $K^* = \sum_{i=1}^{N} (\eta_i^*)^p$ becomes smooth in $\mathbb{R}^3$. In the feasible region (subset of $\mathbb{R}^3$ where the free node could be placed for the local mesh to be valid) the modified objective function K^* approximates the original function K as $\delta \to 0$ and, then, the minimum of the original and modified objective functions are nearly identical when δ is small. When this region does not exist, the minimum of the modified objective

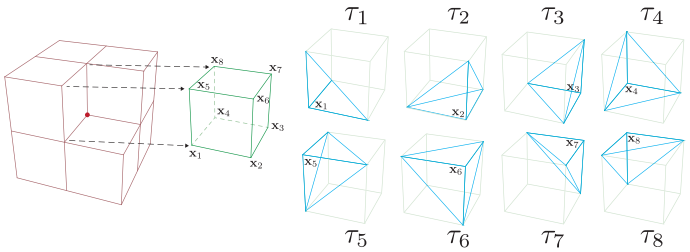


Figure 4.3. On the left, an example of a hexahedral local mesh composed by 8 elements ($M = 8$) where the red point is the free node. On the right, the tetrahedra decomposition proposed by Knupp for a hexahedron.

function is located in such a way that it tends to untangle the local mesh. Thus, the modified objective function allows the simultaneous untangling and smoothing of the mesh. The value of δ is selected in terms of the local mesh under consideration, making it as small as possible and in such a way that the evaluation of the minimum of the modified function does not present any computational problem. For more details see [60, 83]. The unconstrained optimization problem can be easily solved with any standard method, see for example [73].

4.3.2 Objective function for a hexahedral mesh based on necessary conditions

Knupp proposes a distortion measure for a hexahedral element based on its decomposition in eight tetrahedra [69, 84]. The tetrahedron τ_i is the one formed by the i-th vertex of the hexahedron and its edges coincident on this vertex, see Fig. 4.3. Let us consider the Jacobian matrix A_i, the weighted Jacobian matrix $S_i = A_i W_i^{-1}$ and the distortion $\eta(S_i)$ of the tetrahedron τ_i. Then, the distortion of the hexahedron is given by

$$\eta = \frac{1}{8}\sum_{i=1}^{8}\eta(S_i) = \frac{1}{8}\sum_{i=1}^{8}\frac{\|S_i\|^2}{3\,\sigma(S_i)^{2/3}}.$$

In our case the ideal hexahedron is the cube and, therefore, the ideal tetrahedron is the rectangular isosceles. Note that $1 \leq \eta < \infty$ being $\eta = 1$ for a cube.

The modified distortion of a hexahedron is defined by

$$\eta^* = \frac{1}{8}\sum_{i=1}^{8}\eta^*(S_i) = \frac{1}{8}\sum_{i=1}^{8}\frac{\|S_i\|^2}{3\,h(\sigma(S_i))^{2/3}}. \tag{4.3}$$

Let us consider the free node $\mathbf{x}$ of a hexahedral local mesh. Then, following [79], the modified objective function for this local mesh is

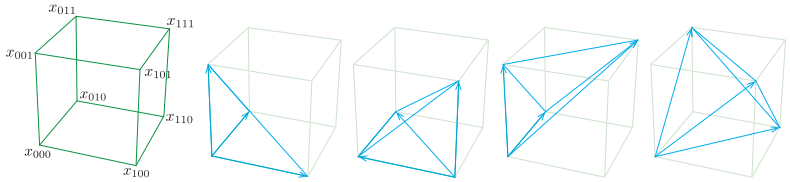


Figure 4.4. Examples of tetrahedra for sufficient conditions. (a) Vertex enumeration, (b) α_{000}, (c) β^1_{000}, (d) γ^1_{000} and (e) κ_{000}.

$$K^*(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}(\eta^*_m)^p(\mathbf{x}), \tag{4.4}$$

where M is the number of hexahedra connected to the free node $\mathbf{x}$ and η^*_m is the modified distortion of the m-th hexahedron, see Fig. 4.3.

As we previously said, a hexahedron is valid if $J(\boldsymbol{\xi}) > 0$, $\boldsymbol{\xi} \in [-1,1]^3$ and, in particular, this must be true for all the vertexes $\boldsymbol{\xi}_i$ of the hexahedron. It is easy to prove that the Jacobian matrix of the trilinear mapping $\mathbf{x}(\boldsymbol{\xi})$ evaluated at the vertexes $\boldsymbol{\xi}_i$ is related to the tetrahedra matrix S_i as follows: $(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}})\big|_{\boldsymbol{\xi}=\boldsymbol{\xi}_i} = \mu S_i$, where μ is a positive scalar. Therefore, $\det(S_i) > 0$ are necessary conditions for the validity of the hexahedron.

4.3.3 Objective function for a hexahedral mesh based on sufficient conditions

Ushakova [80] proposes a set of sufficient conditions that guarantee the non-degeneracy of a hexahedron. These conditions are formulated in terms of the volumes of certain tetrahedra constructed from the vertexes $\mathbf{x}_{i_1 i_2 i_3}$ of the hexahedron, see Fig. 4.4. The tetrahedra are grouped in four groups α, β, γ and κ:

$$\alpha_{i_1 i_2 i_3} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 i_3}, \mathbf{x}_{i_1 i_2 \bar{i}_3}),$$

$$\begin{aligned}
\beta^1_{i_1 i_2 i_3} &= \beta^{1 i_1}_{i_2 i_3} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 \bar{i}_3}),\\
\beta^2_{i_1 i_2 i_3} &= \beta^{2 i_2}_{i_3 i_1} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 \bar{i}_3}),\\
\beta^3_{i_1 i_2 i_3} &= \beta^{3 i_3}_{i_1 i_2} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 \bar{i}_3}, \mathbf{x}_{i_1 i_2 \bar{i}_3}),
\end{aligned}$$

$$\begin{aligned}
\gamma^1_{i_1 i_2 i_3} &= \gamma^{1 i_1}_{i_2 i_3} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 \bar{i}_2 \bar{i}_3}, \mathbf{x}_{i_1 \bar{i}_2 i_3}, \mathbf{x}_{i_1 i_2 \bar{i}_3}),\\
\gamma^2_{i_1 i_2 i_3} &= \gamma^{2 i_2}_{i_3 i_1} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 \bar{i}_2 \bar{i}_3}, \mathbf{x}_{i_1 i_2 \bar{i}_3}),\\
\gamma^3_{i_1 i_2 i_3} &= \gamma^{3 i_3}_{i_1 i_2} = (\mathbf{x}_{i_1 i_2 i_3}, \mathbf{x}_{\bar{i}_1 i_2 i_3}, \mathbf{x}_{i_1 \bar{i}_2 i_3}, \mathbf{x}_{\bar{i}_1 \bar{i}_2 \bar{i}_3}),
\end{aligned}$$

$$\kappa_{i_1} = (\mathbf{x}_{i_1 i_1 i_1}, \mathbf{x}_{i_1 \bar{i}_1 \bar{i}_1}, \mathbf{x}_{\bar{i}_1 i_1 \bar{i}_1}, \mathbf{x}_{\bar{i}_1 \bar{i}_1 i_1}),$$

where $i_1, i_2, i_3 = 0, 1$ and $\bar{0} = 1$, $\bar{1} = 0$.

Tetrahedra α are formed by three edges with the common corner (8 tetrahedra), tetrahedra β by two edges and the diagonal of one of adjacent faces (24 tetrahedra), tetrahedra γ by two edges and the inner diagonal of the hexahedron (24 tetrahedra) and tetrahedra κ by diagonals of faces (2 tetrahedra). See some samples in Fig. 4.4. The total number of tetrahedra is $8+24+24+2=58$. Note that the 8 tetrahedra α are the same 8 tetrahedra proposed by Knupp for optimization of a hexahedron.

Non-degeneracy sufficient conditions are expressed in form of 27 inequalities [80]:

$$\det(A(\alpha_{i_1 i_2 i_3})) \geq 0, \quad i_1, i_2, i_3 = 0, 1; \tag{4.5}$$

$$\sum_{i_k=0}^{1} \det(A(\beta_{i_l i_m}^{k i_k})) \geq 0, \quad (klm) = (123),\ i_l, i_m = 0, 1; \tag{4.6}$$

$$\sum_{i_l, i_m=0}^{1} \det(A(\gamma_{i_l i_m}^{k i_k})) \geq 0, \quad (klm) = (123),\ i_k = 0, 1; \tag{4.7}$$

$$\sum_{i_1=0}^{1} \det(A(\kappa_{i_1})) \geq 0; \tag{4.8}$$

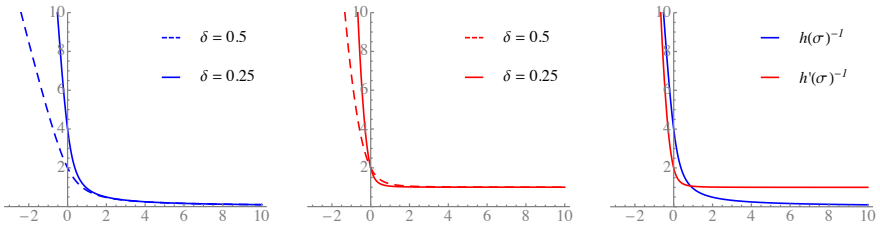
where $A(\tau)$ is the Jacobian matrix of the tetrahedron τ, taking as reference node the first one in the list of vertexes that defines τ.

Among 27 inequalities we have eight inequalities (4.5) for non-degeneracy in the vertexes, 12 inequalities (4.6) for non-degeneracy on edges, 6 inequalities (4.7) for non-degeneracy on faces, and 1 inequality (4.8) for non-degeneracy in the inner part of a hexahedron.

As pointed in [80], these non-degeneracy conditions are difficult to impose in a mesh generation process because they involve sum of volumes. For this reason, Ushakova proposes other sets of conditions, easier to impose, based on the non-degeneracy of some of the previously related tetrahedra. For example, one of the proposed tests checks the validity of α and β tetrahedra

$$\det(A(\alpha_{i_1 i_2 i_3})) > 0, \quad \det(A(\beta_{i_1 i_2 i_3}^{k})) > 0, \quad i_1, i_2, i_3 = 0, 1,\ k = 1, 2, 3. \tag{4.9}$$

Note that only the non-degeneracy of the tetrahedra α are necessary conditions for the validity of the hexahedron. The non-degeneracy of the remainder individual tetrahedra β, γ and κ, are not necessary conditions.



(a) (b) (c)

Figure 4.5. Barrier functions. (a) $h(\sigma)^{-1}$ with different values of δ. (b) $h'(\sigma)^{-1}$ with different values of δ. (c) Comparison of $h(\sigma)^{-1}$ and $h'(\sigma)^{-1}$ with $\delta = 0.25$.

Proposed objective functions

Here we present objective functions based on the two types of conditions given by (4.5)-(4.8) and (4.9).

In (4.5)-(4.8) we differentiate two types of conditions. Conditions (4.5) involve individual tetrahedra and conditions (4.6)-(4.8) concern several tetrahedra simultaneously. Equation (4.5) is the set of necessary conditions mentioned in section 4.3.2, so for them we propose same terms as in (4.2), see first summation in eq. (4.10). For conditions (4.6)-(4.8) we have added additional terms to guarantee the validity of the hexahedron, see second, third and fourth summations in eq. (4.10). These conditions could be imposed by means of the function $h(\sigma)$. For example, the term associated to the condition $\det(A(\beta_{00}^{10})) + \det(A(\beta_{00}^{11})) > 0$ could be $h(\det(A(\beta_{00}^{10})) + \det(A(\beta_{00}^{11})))^{-1}$. Nevertheless, function $h(\sigma)^{-1}$ is not dimensionless and then it is sensitive to scale changes. For this reason, we propose $h'(\sigma)^{-1} = 2 - \frac{\sigma}{h(\sigma)}$ in order to impose conditions (4.6)-(4.8). This function has a similar behavior to $h(\sigma)^{-1}$, see Fig. 4.5, but it is dimensionless and scale invariant when $\delta \to 0$.

Let us consider $S(\tau)$ the weighted Jacobian matrix of a tetrahedron τ, taking as reference tetrahedron its counterpart in the master hexahedron, and $\sigma(\tau) = \det(S(\tau))$. Then, the proposed objective function to impose conditions (4.5)-(4.8) for a hexahedron is

$$
\begin{aligned}
k^* = \sum_{i_1,i_2,i_3} \frac{\|S(\alpha_{i_1 i_2 i_3})\|^2}{3h\left(\sigma(\alpha_{i_1 i_2 i_3})\right)^{2/3}} &+ \sum_{k,i_l,i_m} \frac{1}{h'(\sum_{i_k} \sigma(\beta_{i_l i_m}^{k i_k}))} + \\
&+ \sum_{k,i_k} \frac{1}{h'(\sum_{i_l,i_m} \sigma(\gamma_{i_l i_m}^{k i_k}))} + \\
&+ \frac{1}{h'(\sum_{i_1} \sigma(\kappa_{i_1}))},
\end{aligned}
\tag{4.10}
$$

where $i_1, i_2, i_3 = 0, 1$, $(klm) = (123)$ and $i_l, i_m, i_k = 0, 1$. Note that the first term of (4.10) tends to improve the quality of the tetrahedra α and the rest of the terms just impose

barriers associated to conditions (4.6)-(4.8).

For conditions like (4.9), which only involve individual tetrahedra, the objective function for a hexahedron is

$$k^* = \sum_{i_1,i_2,i_3} \frac{\|S(\alpha_{i_1 i_2 i_3})\|^2}{3h(\sigma(\alpha_{i_1 i_2 i_3}))^{2/3}} + \sum_{k,i_1,i_2,i_3} \frac{\left\|S(\beta^k_{i_1 i_2 i_3})\right\|^2}{3h(\sigma(\beta^k_{i_1 i_2 i_3}))^{2/3}} \tag{4.11}$$

where $i_1, i_2, i_3 = 0, 1$ and $k = 1, 2, 3$.

Finally, either for conditions (4.5)-(4.8) or (4.9), the objective function for a local hexahedral mesh is

$$K^*(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} (k^*_m)^p(\mathbf{x}), \tag{4.12}$$

where M is the number of hexahedra connected to the free node $\mathbf{x}$ and k^*_m is the term associated to the m-th hexahedron of the local mesh.

4.4 Optimization based on the global distortion of the hexahedron

An objective function for a hexahedral mesh can be defined through a global distortion measure for a hexahedron, as proposed in [81].

We can define the modified pointwise distortion as in (4.2), but now considering that $S(\boldsymbol{\xi}) = (\partial \mathbf{x}/\partial \boldsymbol{\xi}) W(\boldsymbol{\xi})^{-1}$ is the weighted Jacobian matrix of the trilinear transformation (6.3.1), where $\partial \mathbf{x}/\partial \boldsymbol{\xi}$ is the Jacobian matrix of the mapping from the master element to the physical one, and $W(\boldsymbol{\xi})$ is Jacobian matrix of the mapping from master element to the ideal (target) one. In our case the ideal hexahedron coincides with the master element $\hat{\Omega} = [-1, 1]^3$, so the matrix $W(\boldsymbol{\xi})$ is the unit matrix and, thus, $S(\boldsymbol{\xi}) = (\partial \mathbf{x}/\partial \boldsymbol{\xi})$.

Then, the pointwise modified distortion for a hexahedron is defined as

$$\eta^*(\boldsymbol{\xi}) = \frac{\|S(\boldsymbol{\xi})\|^2}{3h(\sigma(\boldsymbol{\xi}))^{2/3}}, \quad \boldsymbol{\xi} \in [-1, 1]^3,$$

where $\sigma(\boldsymbol{\xi}) = \det(S(\boldsymbol{\xi}))$.

The pointwise distortion $\eta^*(\boldsymbol{\xi})$ attains values between 1 and ∞. If the physical element Ω is a cube, then $S(\boldsymbol{\xi}) = \mu I$ and, in this case, $\eta^*(\boldsymbol{\xi}) \to 1$ when $\delta \to 0$. On the other hand, if $\sigma(\boldsymbol{\xi}) \leq 0$ for some $\boldsymbol{\xi}$, then $\eta^*(\boldsymbol{\xi}) \to \infty$ when $\delta \to 0$.

The global modified distortion measure of a hexahedron Ω is given by

$$\eta^*_\Omega = \frac{1}{V_{\hat{\Omega}}} \int_{\hat{\Omega}} \eta^*(\boldsymbol{\xi})\, d\hat{\Omega}, \tag{4.13}$$

where $V_{\hat{\Omega}} = 8$ is the volume of $\hat{\Omega}$. In [81], the quality of a hexahedron is defined as $(\eta^*_\Omega)^{-1}$ with $\delta \to 0$.

This distortion measure presents pseudo-barriers, that is, $\eta^*_\Omega \to \infty$ when $\sigma(\boldsymbol{\xi}) \leq 0$ and $\delta \to 0$. Then, we define the objective function for a local mesh as

$$K^*(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \eta^*_{\Omega_m}(\mathbf{x}), \tag{4.14}$$

being M the number of hexahedra connected to the free node $\mathbf{x}$.

The integral is evaluated using a numerical quadrature rule. It is important to highlight that the objective function (6.3.2) does not work properly when the quadrature rule "does not see" the pseudo-singularities of η^*_Ω, that is, there is not any quadrature point inside the region where η^*_Ω takes very high values.

4.4.1 Numerical quadrature of the distortion

We use the simple quadrature rule given by

$$\int_{\hat{\Omega}} \eta^*(\boldsymbol{\xi})\, d\hat{\Omega} \approx \sum_{i,j,k=1}^{2} w_{ijk}\, \eta^*(\boldsymbol{\xi}_{ijk}), \tag{4.15}$$

where $\boldsymbol{\xi}_{ijk}$ are the vertexes of the master element $\hat{\Omega}$ and $w_{ijk} = \frac{1}{8}\text{Vol}(\hat{\Omega})$. It is reasonable to place quadrature points on the vertexes because the singularities are usually close to the boundary of the $\hat{\Omega}$.

In some cases, the optimization of a highly distorted hexahedron by means of the function (6.3.2) may not untangle the element. This is because the numerical evaluation of the distortion is not accurate enough. In order to improve the precision of the numerical integration, we increase the number of quadrature points subdividing the element and performing the integration on each sub-element using the same quadrature rule given by (4.15). This subdivision can be global or local. Global subdivisions can be computationally impractical because the number of terms in the objective function increases exponentially when several subdivisions steps are required. We propose an adaptive quadrature scheme using an octree subdivision.

4.4.2 Adaptive numerical quadrature of the hexahedron distortion

The strategy for adaptive quadrature is based on dividing only sub-elements of $\hat{\Omega}$ where the Jacobian is negative. In this way, quadrature points tend to be concentrated in the neighborhood of the pseudo-singularities of $\eta^*(\boldsymbol{\xi})$.

The algorithm (1) describes the optimization procedure using adaptive quadrature.

The procedure consists in a repeat loop. In a first step (line 2), the hexahedron Ω is optimized. The first time that Ω is optimized we take as quadrature points the vertexes of $\hat{\Omega}$. In the second step of the algorithm (line 3), the control values b_{ijk} of the expansion of the Jacobian J in Bernstein polynomials are computed as explained in Appendix A. If the lower bound of J, $\min(b_{ijk})$, is not positive (line 4), then the validity of the element cannot be determined. In this case, the Jacobian is minimized

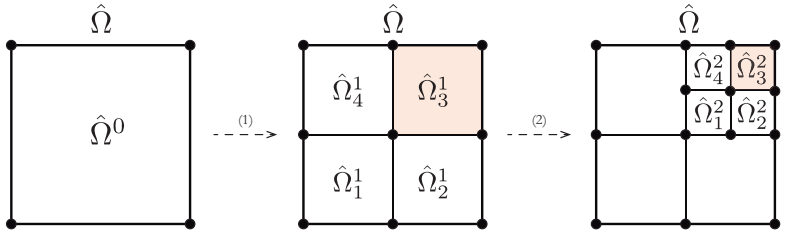


Figure 4.6. Example of two steps of adaptive quadrature in a 2D element. The red shaded $\hat{\Omega}_i^n$ is the sub-element that needs to be refined because it contains $\boldsymbol{\xi}_{min}$. The black points are the quadrature points in each $\hat{\Omega}_i^n$.

Algorithm 1: Optimization procedure using adaptive quadrature.

```
1  repeat
2      Optimize(Ω)
3      b_ijk ⟵ BernsteinCoefficients(J)
4      if min(b_ijk) ≤ 0 then
5          (J_min, ξ_min) ⟵ Minimize(J)
6          if J_min ≤ 0 then
7              Ω̂_i^n ⟵ SearchSubelement(Ω̂, ξ_min)
8              Refine(Ω̂_i^n)
9          end
10     end
11 until min(b_ijk) > 0 || J_min > 0 || n > maxRefinement
```

(line 5), obtaining as result its minimum value J_{min} and $\boldsymbol{\xi}_{min} = \arg\min J(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \hat{\Omega}$. If $J_{min} \leq 0$ (line 6), then, the sub-element $\hat{\Omega}_i^n$ that contains $\boldsymbol{\xi}_{min}$ (line 7) is refined (line 8). The superscript n indicates the level of refinement of this sub-element ($\hat{\Omega} = \hat{\Omega}^0$).

Note that there are three exit criteria for the loop (line 11). The first two criteria check if the element is already valid, either because the Jacobian is positively bounded ($min(b_{ijk}) > 0$) or its minimum is positive ($J_{min} > 0$). The third exit criterion ($n > maxRefinement$) avoids an infinity loop by limiting the maximum number of recursive refinements that can be made. In practice, we have observed that in most cases 5 steps of refinement are enough. An example of two steps of adaptive quadrature in 2D is illustrated in Fig. 4.6.

When we deal with a local mesh instead of a single element, we apply the same adaptive quadrature to each element of the local mesh.

Remark 1 Note that with any quadrature rule, the minimization of the objective function (6.3.2) imposes necessary conditions for the validity of a hexahedron insomuch as it imposes the positivity of the Jacobian at the quadrature points. If the integration is performed without any subdivision, i.e., using the 8 vertexes of the hexahedron as quadrature points, the objective function (6.3.2 is equivalent to the objective function

(4.4), see the last paragraph of the section 4.3.2. So the objective function (4.4) is a particular case of (6.3.2). If the integration is performed with one subdivision, the positivity of the Jacobian is imposed in 27 quadrature points: 8 corners, center of each face, center of each edge and the center of the element.

4.5 Experiments and results

The goal of the proposed experiments is to compare the untangling capability of different objective functions. We have considered three groups of objective functions: TD (Tetrahedron Distortion), SC (Sufficient Conditions) and GD (Global Distortion). The TD group consists of several objective functions based on the distortion of certain tetrahedra, SC is based on non-degeneracy sufficient conditions defined through tetrahedra volumes, and GD are based on a global distortion measure of a hexahedron defined through the trilinear transformation.

We compare the nine objective functions listed below:

1. TD_α: objective function based on the distortion of the tetrahedra α.
2. $TD_{\alpha\beta}$: objective function based on the distortion of the tetrahedra α and β.
3. $TD_{\alpha\gamma}$: objective function based on the distortion of the tetrahedra α and γ.
4. $TD_{\alpha\kappa}$: objective function based on the distortion of the tetrahedra α and κ.
5. $TD_{\alpha\beta\gamma\kappa}$: objective function based on the distortion of the tetrahedra α, β, γ and κ.
6. SC: objective function based on sufficient conditions.
7. GD_0: objective function based on global distortion of the hexahedron, evaluated without any subdivision of $\hat{\Omega}$ (8 quadrature points).
8. GD_1: objective function based on global distortion of the hexahedron, evaluated with one subdivision of $\hat{\Omega}$ (27 quadrature points).
9. GD_n: objective function based on global distortion of the hexahedron, evaluated with adaptive quadrature for $\hat{\Omega}$.

The TD_α objective function is given by eq. (4.4) and the rest of $TD's$ are based on the eq. (4.11) and (4.12) adding the terms associated to the β, γ or κ tetrahedra. The SC is based on eq. (4.10) and (4.12). It is worth mentioning that $TD_{\alpha\beta\gamma\kappa}$ also imposes sufficient conditions and that they are more restrictive than the conditions imposed by SC. Finally, the $GD's$ are based on eq. (6.3.1), (6.3.2) and (4.15). Note that functions TD_α and GD_0 are equivalent, see Remark 1.

We have designed two computational experiments that are explained in the following sections.

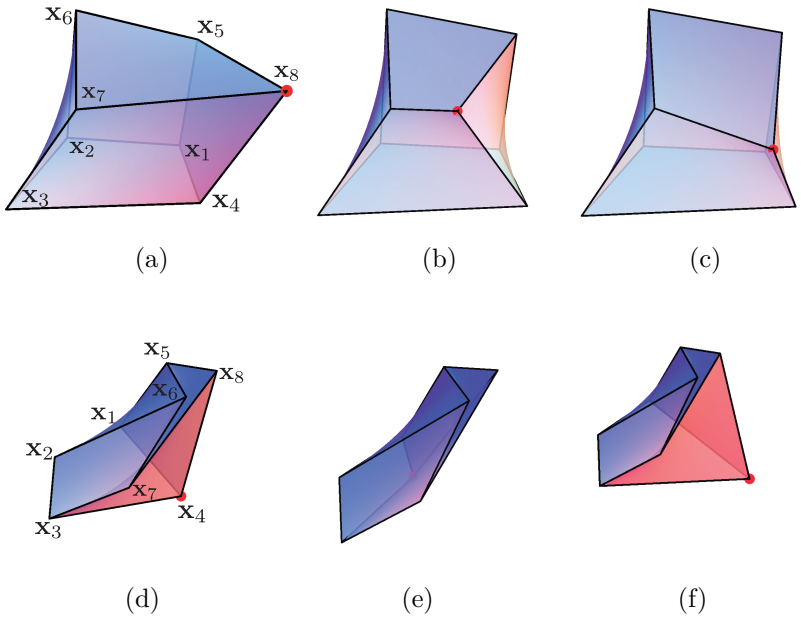


Figure 4.7. Elements for the experiment 1. (a) Initial invalid element 1. (b) Resulting invalid element 1 after relocating the node $\mathbf{x}_8$ using $TD_{\alpha\gamma}$. (c) Resulting valid element 1 after relocating $\mathbf{x}_8$ using GD_n. (d) Initial valid element 2. (e) Resulting invalid element 2 after relocating $\mathbf{x}_4$ using $TD_{\alpha\gamma}$. (e) Resulting valid element 2 after relocating $\mathbf{x}_4$ using GD_n.

4.5.1 Experiment 1

The first experiment consists in optimizing two specific isolated elements taken from [85], one invalid and another valid. Each element is optimized with the proposed objective functions considering $\mathbf{x}_i$ $(i = 1,\ldots,8)$ as free node and fixing the remaining vertexes. Then, we check the validity of the optimized elements.

Table (4.1) shows the results of this experiment. Element 1 is formed by the vertexes $\mathbf{x}_1 = (0,0,0)$, $\mathbf{x}_2 = (1,0,0)$, $\mathbf{x}_3 = (1.5,1.25,0)$, $\mathbf{x}_4 = (0,1,0)$,$\mathbf{x}_5 = (0,0,1)$, $\mathbf{x}_6 = (1,-0.5,1)$, $\mathbf{x}_7 = (1,0.5,0.5)$ and $\mathbf{x}_8 = (-0.5,1,1)$, see Fig. 4.7(a).

This element is initially invalid, so it is not possible to guarantee that it has a feasible region when certain vertexes act as free node. This is due to the fact that the degeneration of the hexahedron can be produced by the position of the other vertexes. For example, this occurs when the vertex $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_4$ or $\mathbf{x}_5$ is taken as free node. If the free node is another vertex, the validity of the optimized hexahedron depends on the chosen objective function. For example, we can see that $TD_{\alpha\gamma}$ does not obtain a valid element when the free node is the vertex $\mathbf{x}_8$, but using GD_n the optimized element is valid, see Fig. 4.7(b) and (c).

The vertexes of the element 2 are $\mathbf{x}_1 = (0,0,0)$, $\mathbf{x}_2 = (1,0,0)$, $\mathbf{x}_3 = (1.5,0.5,0)$, $\mathbf{x}_4 = (0,1,0)$, $\mathbf{x}_5 = (0,0,1)$, $\mathbf{x}_6 = (0.5,0.5,1.25)$, $\mathbf{x}_7 = (1,0.75,0.5)$ and $\mathbf{x}_8 = (0.25,0.5,1.5)$, see

Table 4.1. Experiment 1 results. The symbols ✓ and ✗ show if the optimization produces a valid element or not, taking the vertex $\mathbf{x}_i$ as free node.

Obj. Function	Free node							
	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$
	element 1							
TD_α (GD_0)	✗	✗	✓	✗	✗	✓	✓	✓
$TD_{\alpha\beta}$	✗	✗	✓	✗	✗	✓	✓	✓
$TD_{\alpha\gamma}$	✗	✗	✓	✗	✗	✓	✓	✗
$TD_{\alpha\kappa}$	✗	✗	✓	✗	✗	✓	✓	✓
$TD_{\alpha\beta\gamma\kappa}$	✗	✗	✓	✗	✗	✓	✓	✗
SC	✗	✗	✓	✗	✗	✓	✓	✓
GD_1	✗	✗	✓	✗	✗	✓	✓	✓
GD_n	✗	✗	✓	✗	✗	✓	✓	✓
	element 2							
TD_α (GD_0)	✓	✓	✓	✗	✗	✓	✗	✓
$TD_{\alpha\beta}$	✓	✓	✓	✓	✓	✓	✓	✓
$TD_{\alpha\gamma}$	✓	✗	✓	✗	✓	✓	✓	✓
$TD_{\alpha\kappa}$	✓	✓	✓	✗	✗	✓	✗	✓
$TD_{\alpha\beta\gamma\kappa}$	✓	✓	✓	✗	✓	✓	✓	✓
SC	✓	✓	✓	✗	✗	✓	✗	✓
GD_1	✓	✓	✓	✓	✗	✓	✓	✓
GD_n	✓	✓	✓	✓	✓	✓	✓	✓

Fig. 4.7(d). This is a valid element, so it always exists a feasible region regardless of the vertex taken as free node. Nevertheless, the optimization with some objective functions produces an invalid element, as for example $TD_{\alpha\gamma}$ with the vertexes $\mathbf{x}_4$ as free node, see Fig. 4.7(e). Instead, using for example GD_n, the resulting element is always valid, see Fig. 4.7(f).

4.5.2 Experiment 2

In this experiment we optimize a sufficiently large sample of highly distorted elements, collect statistical data and compare the rate of success of each strategy.

The experiment is carried out on two types of samples. In the first one, we consider isolated hexahedral elements. We generate 10^5 highly distorted hexahedra, all of them valid. The elements are generated by placing the vertexes on random positions, with coordinates between 0 and 1, and collecting only the cases when the hexahedron is valid. Generally, the elements generated by this procedure are deformed enough to test the untangling ability of the studied objective functions.

The second testing sample is composed by local meshes formed by eight hexahedral elements. Due to the low probability to generate a valid local mesh by using random numbers, we have opted for a procedure driven by an objective function. In the appendix we explain in detail the procedure used for the generation of highly distorted local meshes.

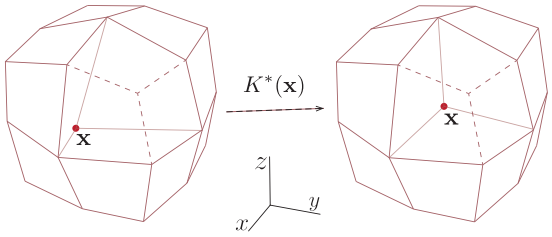


Figure 4.8. Local mesh optimization.

For each objective function we count the number of isolated elements and local meshes that are valid after optimization. We have selected for the experiment only valid elements to be sure that the feasible region exists, i.e., the free node can be placed making the element valid. Otherwise, we could not be able to know whether a strategy fails to produce a valid element or this element just cannot be untangled. Note that it is irrelevant whether we start with a valid element or not, since the final position of the free node after optimization does not depend on its initial position. It is important to mention that some optimization strategies can tangle a valid element. Actually, none of the considered strategies is based on necessary and sufficient conditions for non-degeneracy of a hexahedron. Therefore, any of these strategies could tangle an initially valid element (specially, when the element is highly distorted).

For isolated elements, we take as free node $\mathbf{x}_1$, and for local meshes, the free node is the common vertex of the eight elements, see Fig. 4.8.

Table (4.2) shows the results of the experiment 2. We start with some comments about the results of the first group of objective functions (TD). In this group, the objective function TD_α reports the best results with the 94.36% rate of success for isolated elements and the 94.03% for local meshes. The remaining $TD's$ involve a larger number of tetrahedra and their untangling rates are lower.

When the terms corresponding to β, γ or κ tetrahedra are added, the objective function imposes sufficient but not necessary conditions for the validity of an element. These conditions are excessively restrictive and sometimes difficult to satisfy. Consequently, the optimization of this objective function may not place the free node within its feasible region. The objective function SC also obtains lower rates of success than TD_α. Note that both $TD_{\alpha\beta\gamma\kappa}$ and SC impose sufficient conditions. However, the conditions used for SC strategy are less restrictive than the conditions of $TD_{\alpha\beta\gamma\kappa}$, and thus its success rate is higher than the rate of $TD_{\alpha\beta\gamma\kappa}$.

The best optimization success rates are obtained with the $GD's$ objective functions. In the case of GD_1, the 98.74% of isolated elements and the 98.51% of local meshes are valid after the optimization. With an adaptive distribution of the quadrature points, GD_n, these rates are even better, with only 32 and 66 invalid isolated elements and local meshes. We suspect that the cases where GD_n fails are due to numerical problems in the optimization process. This can happen when the number of terms of the objective function is very large after performing several steps of adaptive subdivisions.

The computational cost of each strategy is mainly the cost of the minimization of the objective function. This minimization is more expensive as more terms (tetrahedra or integration points) are added to the objective function. TD_α and GD_0 are equivalent, see Remark 1. This strategy has only 8 terms by element and it is the most efficient strategy. Objective functions $TD_{\alpha\beta}$, $TD_{\alpha\gamma}$, $TD_{\alpha\kappa}$ and $TD_{\alpha\beta\gamma\kappa}$ have an approximate cost 4, 6, 2 and 10 times higher than TD_α. The relative cost of SC and GD_1 is 4 and 11 times the cost of TD_α. The most expensive strategy is GD_n, but it strongly depends on the mesh where it is applied. When the mesh has relatively well formed elements, the number of adaptive refinements required is lower than when the elements are highly distorted.

In practice it is reasonable to optimize a mesh in two steps. Firstly, the objective function TD_α (GD_0) is applied. Then, the validity of the mesh is analyzed, and, if needed, the mesh is optimized with GD_n.

Table 4.2. Experiment 2 results. IE_s and LM_s represent the number of valid isolated elements and local meshes after optimizing. The sample sizes are 10^5 for IE_s and 3×10^4 for LM_s. The rates of success of each case are $IE_\%$ ($100IE_s/10^5$) and $LM_\%$ ($100LM_s/3 \times 10^4$).

Obj.Function	IE_s	$IE_\%$	LM_s	$LM_\%$
TD_α (GD_0)	94361	94.36%	28210	94.03%
$TD_{\alpha\beta}$	87965	87.96%	17620	58.73%
$TD_{\alpha\gamma}$	66852	66.85%	11669	38.90%
$TD_{\alpha\kappa}$	92155	92.16%	25279	84.26%
$TD_{\alpha\beta\gamma\kappa}$	69650	69.65%	13289	44.30%
SC	91947	91.95%	25853	86.18%
GD_1	98737	98.74%	29553	98.51%
GD_n	99968	99.97%	29944	99.81%

4.6 Conclusions

In this chapter we have analyzed the ability of several objective functions to optimize hexahedral elements. We have studied three groups of objective functions based on the distortion of some tetrahedra formed with the vertexes of the hexahedron (TD), sufficient conditions for the non-degeneracy of a hexahedron (SC) and a global distortion measure for the hexahedron (GD). We have performed two computational experiments in order to test the objective functions with highly distorted elements.

Some important conclusion can be extracted from the results of the experiments. The objective function TD_α (GD_0) obtains good results with a lower computational cost than the rest of strategies. The addition of more tetrahedra to the objective function ($TD_{\alpha\beta}$, $TD_{\alpha\gamma}$, ...) reports worse results and increases the computational cost. Objective function SC does not improve the success rates of TD_α and its computational

cost is higher, too. The best results are obtained with GD_1 and, specially, with GD_n, but with the disadvantage of increasing the computational cost with respect to TD_α (GD_0).

It is worth noting that the strategy TD_α (GD_0) obtains a quite high rate of success (94%) for the sample of highly distorted meshes, which suggests even better results for moderately distorted meshes. So, in practice, it can be considered a reasonable choice since it succeeds in optimizing the great majority of real meshes. However, when dealing with extremely distorted meshes, a practical approach would be to optimize the mesh in two steps. Firstly, the optimization strategy TD_α (GD_0) is applied until reaching a stopping criterion, for example, that the nodes displacement is under a certain threshold. In the second step, the validity of the mesh is analyzed and, if there are some invalid elements, the mesh is optimized again with GD_n.

An adaptive optimization strategy similar to GD_n can be especially useful for high-order hexahedral and tetrahedral meshes.

4.A Appendix A. Bounds for Jacobian determinant

To verify the validity of a physical element we can minimize the Jacobian and check if its minimum value is negative or not. In order to reduce the computational cost, some bounds can be used to identify quickly some valid and invalid elements.

The Jacobian of the trilinear transformation is at most a degree two polynomial in each variable. The expansion of this polynomial in terms of Bernstein basis allows to find tight bounds for this polynomial. Let us consider the Jacobian

$$J(\boldsymbol{\xi}) = J(\xi,\eta,\zeta) = \sum_{i=0}^{l_i}\sum_{j=0}^{l_j}\sum_{k=0}^{l_k} a_{ijk}\xi^i\eta^j\zeta^k.$$

In our case $l_i, l_j, l_k = 2$ and the total degree of the polynomial is 4, i.e., $a_{ijk} = 0$ if $i+j+k > 4$. The expansion of $J(\boldsymbol{\xi})$ in terms of Bernstein basis is

$$J(\boldsymbol{\xi}) = \sum_{i=0}^{l_i}\sum_{j=0}^{l_j}\sum_{k=0}^{l_k} b_{ijk}B_i^{l_i}(\xi)B_j^{l_j}(\eta)B_k^{l_k}(\zeta)$$

where b_{ijk} are the control values and

$$B_i^l(\xi) = \begin{pmatrix} l \\ i \end{pmatrix} \frac{(\xi+1)^i(1-\xi)^{l-i}}{2^l}$$

is the ith Bernstein polynomial of degree l in the interval $[-1,1]$.

An explicit expression for the control values b_{ijk} can be found in [86]. Due to the properties of the Bernstein functions, i.e., positivity and partition of unity, it can be shown that the control values bound the Jacobian

$$\min_{i,j,k} b_{ijk} \leq J(\boldsymbol{\xi}) \leq \max_{i,j,k} b_{ijk},$$

where $0 \leq i \leq l_i$, $0 \leq j \leq l_j$ and $0 \leq k \leq l_k$.

The minimum of b_{ijk} sets a tight lower bound for $J(\boldsymbol{\xi})$, so the condition $\min b_{ijk} > 0$ determines the validity of an element with a low computational cost. Only in a few ambiguous cases, when $\min b_{ijk} \leq 0$ and $\max b_{ijk} > 0$, we have to minimize $J(\boldsymbol{\xi})$ to check if the element is valid. More accurate bounds can be found by recursively splitting the initial interval in smaller ones [87, 88]. This option is specially appropriate for high degree polynomials in which the minimization is computationally impractical.

4.B Appendix B. Random local meshes generation

Isolated hexahedra for the first sample of the experiment 2 is constructed by assigning a random value from the interval $[0,1]$ to the coordinates of the 8 vertexes of the element. Then, the validity of the hexahedron is checked and only valid ones (positive Jacobian) are selected to participate in the experiment. The probability to generate a valid random hexahedron is approximately $1/1000$, i.e., one valid element is generated per 1000 randomly generated elements. Due to the low computational cost to generate a random element, it is feasible to obtain a sufficiently big sample for the experiment.

For the second sample of the experiment 2 we need a set of random local meshes. As we are dealing with local meshes formed by 8 elements, we have to generate randomly 26 of the 27 nodes that compose the local mesh (we can fix the free node). We need to select for the experiment only valid meshes, i.e., each element of the local mesh should be valid. However, during random generation of the nodes coordinates, the probability to generate a valid local mesh is very low. In fact, we could not generate any valid local mesh using this procedure. Therefore another approach was used to generate a sample of highly distorted local meshes. The procedure is as follows. We take a local mesh composed by 8 ideal elements and deform it by moving 26 nodes of the local mesh. The deformation is performed by means of an optimization process where the nodes $\{\mathbf{p}_j\}_{j=1,\dots,26}$ are relocated in order to achieve a predefined distortion at certain points of each element of the local mesh. First, we select a set of points where we intend to impose the distortion. For example, we are going to impose the predefined distortion at the vertexes of each element, so there are 8×8 points. Let us denote these points as $\{\mathbf{x}_{k,i}\}_{k,i=1,\dots,8}$, where the index k is for the elements numeration and the index i is for the nodes numeration of the k-th element. Next, we assign to each of these points a distortion value $\bar{\eta}_{k,i} = 1/\bar{q}_{k,i}$, where the quality $\bar{q}_{k,i}$ is a randomly generated number from $[\varepsilon, 1]$. Then, the goal is to deform the initial mesh so that for each element Ω_k its trilinear mapping $\mathbf{x}^k(\xi)$ has the predefined distortion values at the corners $\{\mathbf{x}_{k,i}\}_{i=1,\dots,8}$ of the element, that is

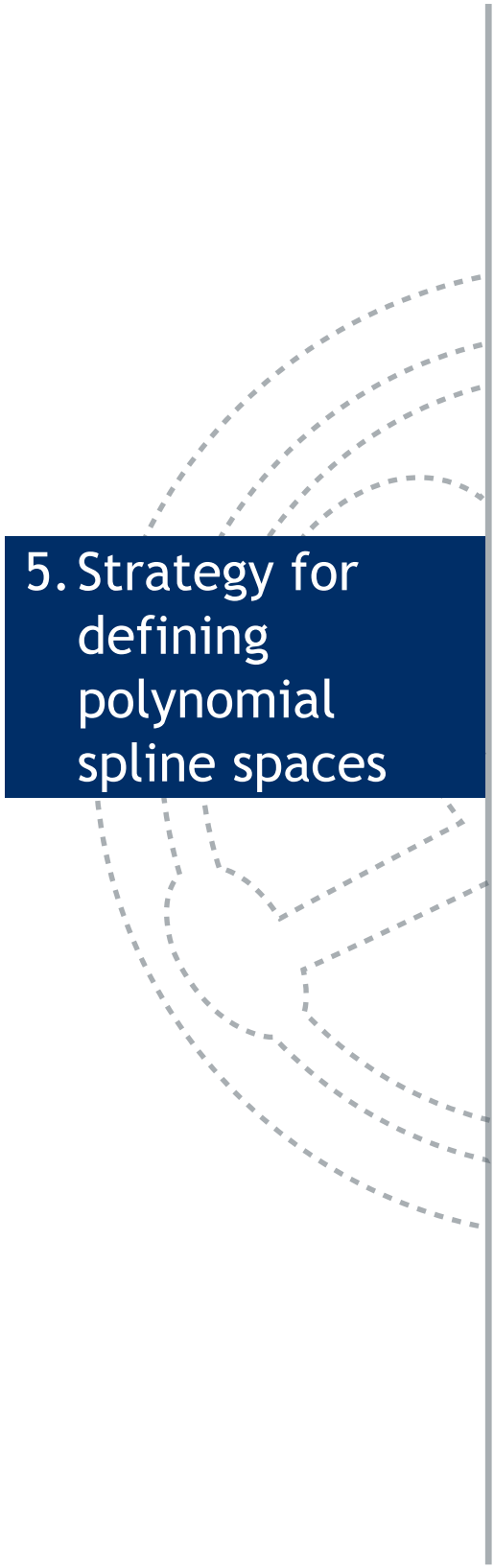
$$\eta^*_{k,i} = \frac{\left\|S_{k,i}\right\|^2}{3h(\sigma(S_{k,i}))^{2/3}} = \bar{\eta}_{k,i},$$

where $S_{k,i}$ is the Jacobian matrix of the mapping $\mathbf{x}^k(\xi)$ evaluated in the i-th node of the element Ω_k. For that, the 26 nodes are relocated by minimizing the following objective

function

$$F(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{26}) = \sum_{k=1}^{8}\sum_{i=1}^{8} \left(\eta^*_{k,i} - \bar{\eta}_{k,i}\right)^2 = \sum_{k=1}^{8}\sum_{i=1}^{8} \left(\frac{\left\|S_{k,i}\right\|^2}{3h(\sigma(S_{k,i}))^{2/3}} - \bar{\eta}_{k,i} \right)^2 .$$

As a result of this minimization we obtain the new position of the nodes $\{\mathbf{p}_j\}_{j=1,\ldots,26}$ that provide a highly distorted local mesh. These local meshes are not necessarily valid, so analogously to the first sample, the validity is checked and only valid local meshes are selected for the experiment. It is worth noting that, in general, it is not possible to construct a hexahedron with predefined quality at certain set of points because the required qualities may not be compatible. However, this procedure provides a good method for generating arbitrary highly distorted local meshes.



5. Strategy for defining polynomial spline spaces

In this chapter we briefly expose another result of our research: a simple strategy for constructing spline spaces over hierarchical T-meshes with quad- and octree subdivision scheme. To use spline functions for numerical analysis and obtain a proper convergence behaviour, these functions must meet some requirements: linear independence, polynomial reproduction property, local supports and possibility to perform local adaptive refinement. This problem was studied in depth in another doctoral dissertation of the group [55]. For a better understanding of the next chapter, we briefly describe our strategy to construct spline spaces with nice properties for analysis. A more detailed explanation can be found in [56]. In next chapter we use these basis functions to parametrize the geometry.

5.1 Steps of the strategy

The strategy we propose has some similarity with T-splines inasmuch as we define the blending functions from local knot vectors that are inferred by traversing the T-mesh edges. Some additional rules and requirements are imposed for the local knot vectors in order to obtain spline spaces with nice properties. In our strategy, the process of spline space construction for a given T-mesh can be divided in the following three steps:

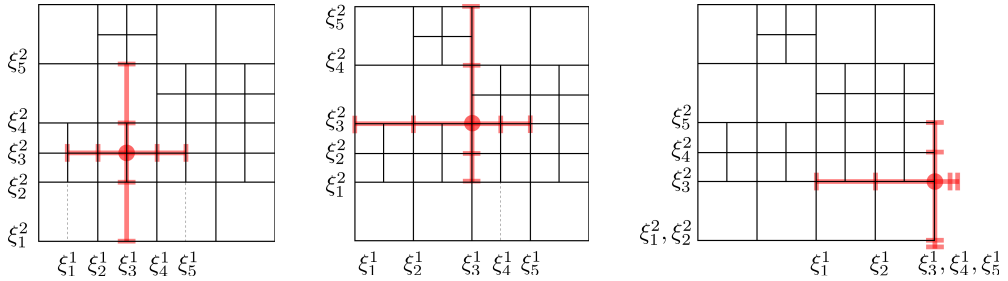
1. Mesh pretreatment (0-balancing)
2. Inferring local knot vectors
3. Modification of local knot vectors

Next, we give a description of each step of the process.

5.1.1 Mesh pretreatment

The strategy we propose is designed exclusively for 0-balanced T-meshes. A mesh with tree structure is said to be 0-balanced if for any k, no cell at level k shares a vertex (0-face) with a cell at level greater than $k+1$. In other words, a 0-balanced quadtree mesh implies that any cell has contact (through vertex, edge or face) only with cells that differ at most in one level of depth. To obtain a 0-balanced quadtree, a standard balancing procedure can be applied. Note that refinements performed during the 0-balancing procedure do not propagate, see [59].

It should be highlighted that 0-balancing the T-mesh is an essential prerequisite for the construction of spline spaces by means of our technique. In general, if the T-mesh is not 0-balanced, our rules for inferring local knot vectors do not lead to polynomial spaces. Also, it is important to emphasize that, for our 2D (3D) T-meshes, a subdivision of any cell is performed by subdividing the cell in 4 (8) equal subcells so that, all cells of the same level have the same size and the edge size of a k-level cell is twice larger than the edge size of a $(k+1)$-level cell.



(a) No T-junction is skipped.

(b) Two T-junctions are skipped in each direction.

(c) Two T-junctions are skipped and the knots are repeated when the boundary is reached.

Figure 5.1. Inferring local knot vectors for a bivariate function by traversing the T-mesh edges.

5.1.2 Inferring local knot vectors

Let us consider a T-mesh T of the squared parametric domain $\Omega = [0,1]^d$, $d = 2$ or 3. We call regular node the node of the mesh that is not a T-junction. We associate a blending function only to regular nodes of the mesh, as it is usual in classical finite element methods when working with hanging nodes. The skeleton of a d-dimensional mesh T is the geometric set of points composed of the union of all $(d-1)$-faces of the mesh and it is denoted by $\mathbf{skt}(T)$. That is, for a 2D space, the mesh skeleton is the union of all the edges of the mesh, and the skeleton of a 3D mesh is the union of all its faces.

To define our cubic tensor product spline blending functions over a given d-dimensional T-mesh, a local knot vector for d parametric directions should be assigned to each function N_α: $\Xi_\alpha^j = \left(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j, \xi_5^j\right)$, $j = 1, ..., d$. Similarly to [30], these knot vectors are inferred by traversing the T-mesh skeleton. For simplicity, let us describe this procedure for a two-dimensional T-mesh. Starting from the central knot (ξ_3^1, ξ_3^2), i.e., the anchor of the function, we walk across the T-mesh until intersecting perpendicularly a mesh edge. According to our strategy, we should skip over the T-junctions where the missing edge is perpendicular to the direction of our marching, see Fig. 5.1. When the boundary of the parametric domain is reached while walking across the mesh, we repeat knots creating an open knot vector structure along the boundary, see Fig. 5.1(c). Note that all interior knots have multiplicity 1. Thus, we obtain for mesh T a set of blending functions $\{N_\alpha\}_{\alpha \in A_T}$, where A_T is the index set. Figure 5.2 illustrates an example of T-mesh in the parameter space and the anchors of all blending functions defined over this mesh. Any interior regular node has exactly one function associated to it and the boundary nodes have more than one function associated due to the open knot vector structure.

The process of inferring local knot vectors can be resumed as follows:

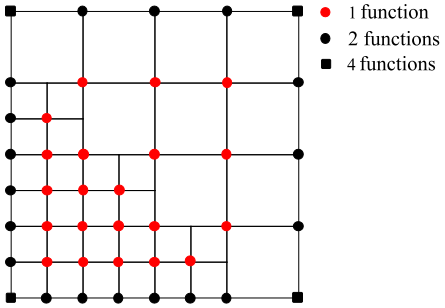


Figure 5.2. T-mesh with its anchors. Red circles represent the interior nodes that have one blending function associated to them, black circles are the boundary nodes that have 2 blending functions and the black squares are the boundary nodes with 4 blending functions due to the open knot vector structure along the boundary.

- Blending functions are associated only to regular nodes of the mesh.
- Local knot vectors are inferred by walking across the mesh until intersecting the mesh skeleton. This intersection should not coincide with a T-junction perpendicular to the marching direction.
- Boundary knots are repeated to create an open knot vector structure along the boundary.

Next, in order to span a spline space with good properties, some function supports should be modified.

5.1.3 Modification of local knot vectors

The key of our strategy lies in some simple rules used for the modification of the function supports that lead to the construction of a polynomial spline space over a given 0-balanced T-mesh. In order to describe the idea, let us introduce some notation. For the local knot vectors $\Xi_\alpha^j = \left(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j, \xi_5^j\right)$, $j = 1, ..., d$ let us denote the length of each knot interval as $\Delta_i^j = \xi_{i+1}^j - \xi_i^j$, $j = 1, ..., d$ and $i = 1, ..., 4$.

The support of a d-variate blending function N_α is a d-dimensional rectangular box: $[\xi_1^1, \xi_5^1] \times \cdots \times [\xi_1^d, \xi_5^d]$. We are going to call frame of a function support the union of all $(d-2)$-faces of this box and we will denote it by $\mathbf{frm}(\mathrm{supp} N_\alpha)$. That is, for the rectangular support of a bivariate function, the frame is the union of the four vertices of this rectangle. For the cuboidal support of a trivariate function, its frame is composed of the union of the twelve edges of this cuboid.

Once the function supports are inferred, we modify them in such a way that, for each blending function N_α, its knot vectors Ξ_α^j $j = 1, ..., d$ verify the following simple conditions:

Condition 1: Local knot vectors of the d-variate function N_α verify[1]

$$\Delta_1^j \geqslant \Delta_2^j = \Delta_3^j \leqslant \Delta_4^j, \quad j = 1, ..., d, \tag{5.1}$$

Condition 2: The frame of the function support should be situated over the mesh skeleton:

$$\mathrm{frm}(\mathrm{supp} N_\alpha) \in \mathrm{skt}(T). \tag{5.2}$$

Thus, the function supports that do not meet Conditions 1 or 2 should be modified. To perform this modification we extend the original support by changing some knot intervals until the resulting support satisfies both conditions. We are going to refer to these support modifications as Extension rule 1 and 2, respectively.

In the next section this procedure is detailed for 2D and 3D cases.

5.2 Support modification

Here, we present simple support Extension rules 1 and 2 to obtain local knot vectors that fulfill Conditions 1 and 2 formulated in the previous section. We proceed as follows. First, if after traversing the T-mesh skeleton the local knot vectors of a function do not satisfy Condition 1, we modify some of their knots in order to meet Condition 1. Then, the fulfillment of Condition 2 is checked and, if it is not satisfied, another appropriate modification of the support is carried out. As a result of these modifications we obtain a new extended support with local knot vectors which verify both conditions. These modifications are easily implemented taking into account the balanced tree structure of the mesh. Let see in detail this procedure.

To clarify the notation, in the rest of the paper we denote the parametric coordinates as (ξ, η, ζ) and it is related to the previous notation as $(\xi^1, \xi^2, \xi^3) = (\xi, \eta, \zeta)$. Consequently, $(\Xi^1, \Xi^2, \Xi^3) = (\Xi, \mathscr{H}, \mathscr{Z})$ and $(\Delta_i^1, \Delta_i^2, \Delta_i^3)$
$= (\Delta_i^\xi, \Delta_i^\eta, \Delta_i^\zeta)$.

5.2.1 Support extension for 2D meshes

The skeleton $\mathrm{skt}(T)$ of a two-dimensional mesh T is the union of all edges of the mesh. For a bivariate function let us denote the vertices of its rectangular support as $V_{1,1} = (\xi_1, \eta_1)$, $V_{5,1} = (\xi_5, \eta_1)$, $V_{5,5} = (\xi_5, \eta_5)$ and $V_{1,5} = (\xi_1, \eta_5)$. Then, the frame of a function support is the union of its four vertices, i.e., $\mathrm{frm}(\mathrm{supp} N_\alpha) = \{V_{n,m},\, n, m \in \{1, 5\}\}$. Figure 5.3 illustrates the introduced notation for a bivariate function support.

Formulation of Condition 1 for the local knot vectors Ξ and $\mathscr{H}$ is simple and does not need any clarification. Condition 2 adapted to 2D case is formulated as follows: The four vertices of a function support should be situated over the mesh edges.

[1] Except the cases involving repeated knots that are explained at the end of Section 5.2.1.

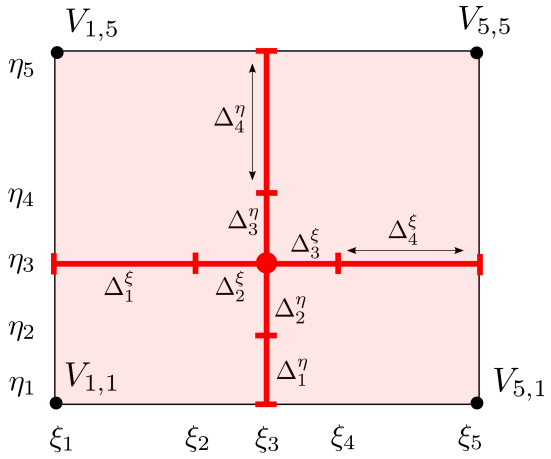
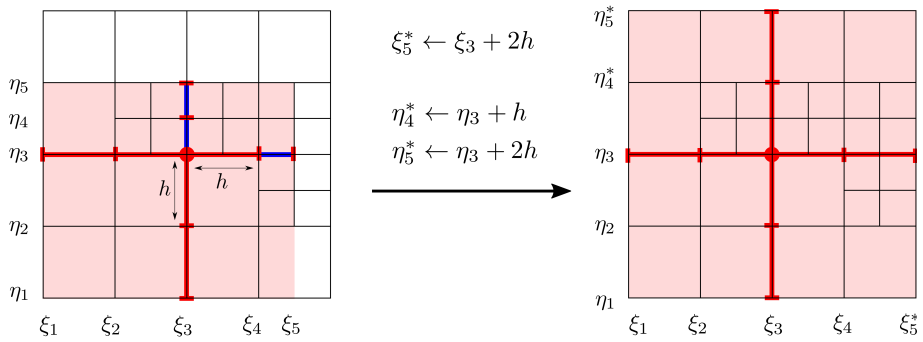


Figure 5.3. Support notation for 2D case.

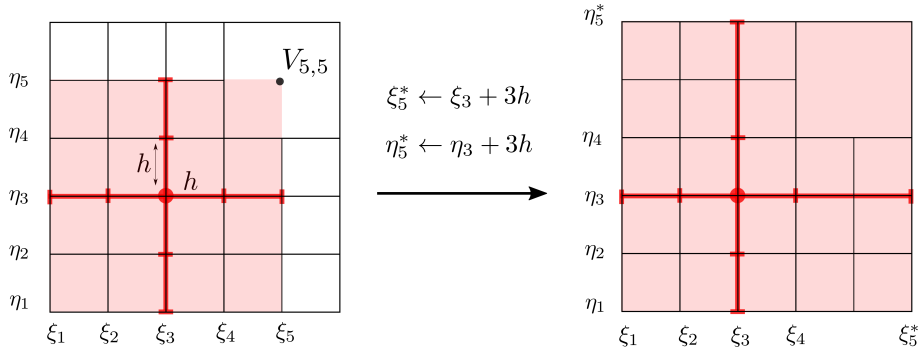
Extension rule 1. If the local knot vector Ξ of a function does not satisfy Condition 1, we modify this vector by skipping over the minimal number of knots until $\Delta_1^\xi \geqslant \Delta_2^\xi = \Delta_3^\xi \leqslant \Delta_4^\xi$ is verified, and analogously, for $\mathscr{H}$. This modification is made independently for each parametric direction applying certain extension rules. Let see an example of support extension for a bivariate function. Leftmost function support shown in Fig. 5.4(a) does not meet Condition 1. For the knot vector Ξ we have $\Delta_3^\xi > \Delta_4^\xi$, so the knot interval Δ_4^ξ should be modified. Let us denote $h = \max(\Delta_2^\xi, \Delta_3^\xi) = \max(\Delta_2^\eta, \Delta_3^\eta)$. Note that both maxima coincide due to the quadtree structure and the fact that the T-junctions are skipped. Then, the fifth knot ξ_5 is redefined as $\xi_5^* \leftarrow \xi_3 + 2h$. For the local knot vector $\mathscr{H}$ we have $\Delta_2^\eta > \Delta_3^\eta$, so the knots η_4 and η_5 should be modified as $\eta_4^* \leftarrow \eta_3 + h$, $\eta_5^* \leftarrow \eta_3 + 2h$.

Extension rule 2. Once Condition 1 is satisfied, in order to fulfill Condition 2, we check whether the vertices of the function support are situated over the mesh edges. If not, we modify the knot vectors by skipping over a knot for both parametric directions and placing this vertex over the mesh edges. Figure 5.4(b) illustrates this procedure. The checking is performed independently for each of the four quadrants of the function support. Note that for our 0-balanced quadtree we should make this checking only for some functions. For example, without loss of generality, the support vertex $V_{5,5} = (\xi_5, \eta_5)$ must be checked only if $\Delta_3^\xi = \Delta_4^\xi = \Delta_3^\eta = \Delta_4^\eta$. An example of a function support violating Condition 2 is illustrated in Fig. 5.4(b). The vertex $V_{5,5} = (\xi_5, \eta_5)$ of this support is not situated over a mesh edge, so the fifth knots for both parametric directions should be redefined as $\xi_5^* \leftarrow \xi_3 + 3h$, $\eta_5^* \leftarrow \eta_3 + 3h$, and thus, the new vertex $V_{5,5}$ is placed over the mesh edges.

The extension of any other function support is completely analogous to these two examples. In all possible cases, the extension of a function support implies to change one or two knot intervals by duplicating its size.



(a) Condition 1 is not satisfied because $\Delta_4^\xi < \Delta_3^\xi$ and $\Delta_3^\eta < \Delta_2^\eta$.



(b) Condition 2 is not satisfied because $V_{5,5} \notin skt(T)$.

Figure 5.4. Extension rules. (a) An example of support modification by Extension rule 1. (b) An example of support modification by Extension rule 2.

Detailed algorithms for the Extension rule 1 and 2 used to modify a bivariate function support according to Conditions 1 and 2 are given in Algorithms 2 and 3.

Note that an exception for Condition 1 is a knot vector that contains a knot interval of length 0 due to the open knot vector structure along the boundary. In this case, a knot vector should fulfill the inequality (5.1) not taking into account the knot intervals of length 0. Consequently, an exception for the application of the extension rules is the case when the boundary of the parametric domain is reached traversing the T-mesh edges.

5.2.2 Support extension for 3D meshes

The skeleton $\mathbf{skt}(T)$ of a three-dimensional mesh T is the union of all faces of the mesh. For a trivariate function let us denote the vertices of its support as $V_{n,m,k} = (\xi_n, \eta_m, \zeta_k)$ where $n,m,k \in \{1,5\}$. And the edge of a support formed by the two vertices $V_{n,m,k}$ and $V_{p,q,r}$ is denoted as $E_{(n,m,k),(p,q,r)}$. Then, the frame $\mathbf{frm}(\mathbf{supp} N_\alpha)$ of a trivariate function support is the union of its twelve edges. Figure 5.5 illustrates the introduced notation for the support of a trivariate blending function.

Algorithm 2: Extension rule 1.

Input: A knot vector $\Xi = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5)$, $\xi_i \in [0,1]$.

Function Modify1(Ξ)
- $\Xi^* \leftarrow \Xi$
- $h = \max(\Delta_2, \Delta_3)$
- if $\Delta_2 < \Delta_3$ *and* $\xi_2 > 0$ then
  - $\xi_2^* \leftarrow \xi_3 - h$
  - $\xi_1^* \leftarrow \xi_2^*$
  - if $\xi_1^* > 0$ then $\xi_1^* \leftarrow \xi_3 - 2h$
- if $\Delta_1 < \Delta_2$ *and* $\xi_1^* > 0$ then
  - $\xi_1^* \leftarrow \xi_3 - 2h$
- if $\Delta_2 > \Delta_3$ *and* $\xi_4 < 1$ then
  - $\xi_4^* \leftarrow \xi_3 + h$
  - $\xi_5^* \leftarrow \xi_4^*$
  - if $\xi_5^* < 1$ then $\xi_5^* \leftarrow \xi_3 + 2h$
- if $\Delta_4 < \Delta_3$ *and* $\xi_5^* < 1$ then
  - $\xi_5^* \leftarrow \xi_3 + 2h$
- return Ξ^*

Output: A corrected knot vector Ξ^* that satisfies Condition 1.

Algorithm 3: Extension rule 2 in 2D.

Input: A 0-balanced mesh T and a pair of local knot vectors $S = \{\Xi, \mathscr{H}\}$.

Function Modify2(T, S)
- $S^* \leftarrow S$
- $h = \max(\Delta_2^{\xi}, \Delta_3^{\xi})$
- for $n \in \{1,5\}$ do
  - for $m \in \{1,5\}$ do
    - if $(\xi_n, \eta_m) \notin skt(T)$ then
      - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
      - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
- return S^*

Output: A modified support $S^* = \{\Xi^*, \mathscr{H}^*\}$ that satisfies Condition 2.

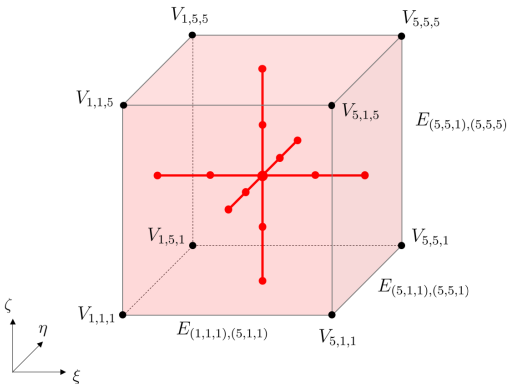


Figure 5.5. Support notation for 3D case.

The formulation of Condition 1 for the local knot vectors of a trivariate function is analogue to the 2D case. Condition 2 adapted to 3D meshes is stated as follows: Edges of the cuboidal function support should be situated over the mesh faces.

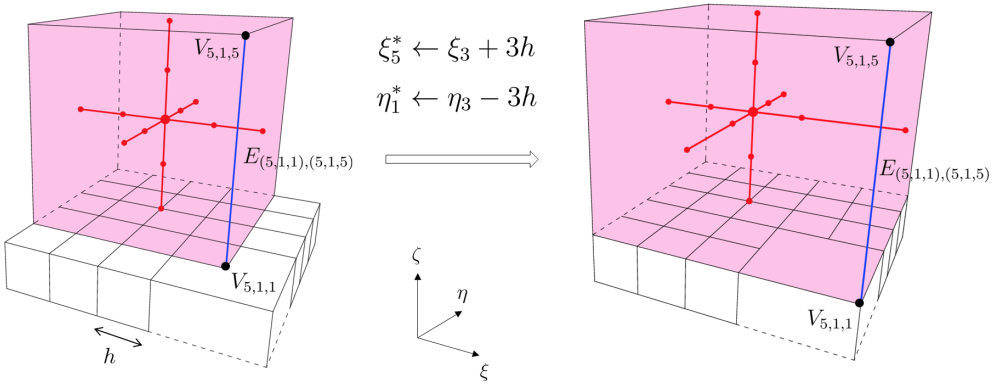
The implementation of the strategy for 3D is similar to the 2D case. To satisfy Condition 1, Extension rule 1 is applied to each of the three local knot vectors of a function analogously to the 2D case using Algorithm 2.

Extension rule 2. In order to fulfill Condition 2, we check whether the edges of a function support are situated over the mesh faces. If not, the two knot vectors perpendicular to this edge should be modified by skipping over a knot for both parametric directions and placing this edge over the mesh faces. This checking is performed independently for each of the eight quadrants of the function support and, in each quadrant, three edges should be checked. Figure 5.6 illustrates the support extension procedure for the quadrant of the vertex $V_{5,1,1}$. Due to the octree structure only two cases can take place: (i) the quadrant contains one edge that is not situated over the mesh faces or (ii) the quadrant contains three edges and a vertex that are not situated over the mesh skeleton. Now we study each case.

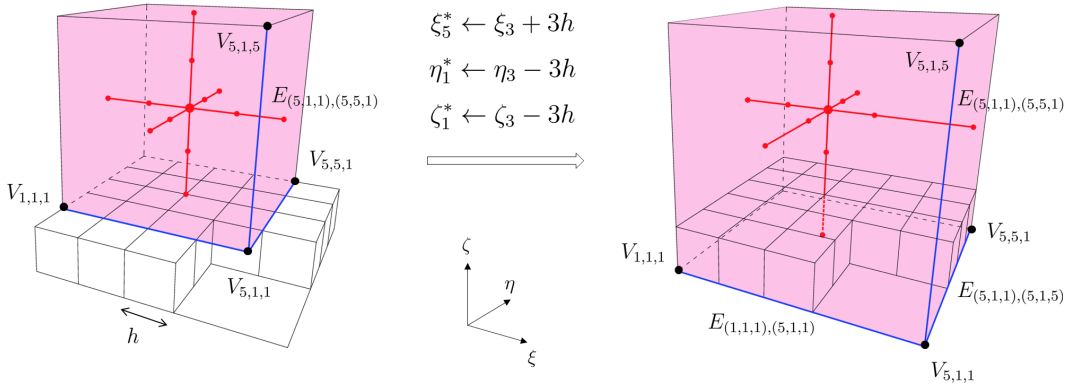
(i) If a quadrant contains one edge that does not fulfill Condition 2, then two knot vectors perpendicular to this edge are modified, see Fig. 5.6(a). For the function support shown in Fig. 5.6(a) left, the edge $E_{(5,1,1),(5,1,5)}$ is not situated over the mesh faces. Therefore, its two knot vectors Ξ and $\mathscr{H}$ perpendicular to this edge are modified in order to place the edge over the mesh faces, namely, knots ξ_5 and η_1 are redefined as $\xi_5^* \leftarrow \xi_3 + 3h$ and $\eta_1^* \leftarrow \eta_3 - 3h$, where $h = \max(\Delta_2^\xi, \Delta_3^\xi) = \max(\Delta_2^\eta, \Delta_3^\eta) = \max(\Delta_2^\zeta, \Delta_3^\zeta)$.

(ii) If a quadrant contains three edges that are not situated over the mesh faces, then the three knot vectors are modified by skipping over a knot for each of the three parametric directions, see Fig. 5.6(b). Vertex $V_{5,1,1}$ and the three edges connected to it are not situated over the mesh skeleton, so all the three knot vectors are modified to place the three edges over the mesh faces: $\xi_5^* \leftarrow \xi_3 + 3h$, $\eta_1^* \leftarrow \eta_3 - 3h$, $\zeta_1^* \leftarrow \zeta_3 - 3h$.

Algorithm 4 explains the Extension rule 2 used to modify a trivariate function support according to Condition 2.



(a) Only one edge violating Condition 2. Node $V_{5,1,1}$ is sited in the center of the face of size $2h$. The support is extended in two directions.



(b) Three edges violating Condition 2. Node $V_{5,1,1}$ is sited in the center of the cell of size $2h$. The support is extended in three directions.

Figure 5.6. Extension rule 2 for support modification of a trivariate function.

5.3 Properties of EP-splines

We name EP-splines to the basis functions obtained with our strategy. EP-splines refers to Extended Polynomial splines. We conjecture that for any strongly balanced T-mesh T of the domain Ω the set of EP-spline functions spans a space $S_T(\Omega) = \text{span}\{N_\alpha : \alpha \in A_T\}$ with the following properties:

1. Functions $\{N_\alpha\}_{\alpha \in A_T}$ are C^2-continuous.
2. Non-negativity: $N_\alpha \geq 0$.
3. Functions $\{N_\alpha\}_{\alpha \in A_T}$ are linearly independent (globally).
4. Non-negative weighted partition of unity: $\sum_{\alpha \in A_T} c_\alpha N_\alpha = 1$, where $c_\alpha \geq 0$.

Algorithm 4: Extension rule 2 in 3D.

Input: A 0-balanced T-mesh T and three local knot vectors $S = \{\Xi, \mathscr{H}, \mathscr{Z}\}$.

Function Modify2 (T, S)
- $S^* \leftarrow S$
- $h = \max(\Delta_2^{\xi}, \Delta_3^{\xi})$
- $mov(i) := i + 4\,\mathrm{sgn}(3-i)$
- for $n \in \{1,5\}$ do
  - for $m \in \{1,5\}$ do
    - for $k \in \{1,5\}$ do
      - if $E_{(n,m,k),(mov(n),m,k)} \notin skt(T)$ then
        - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
        - $\zeta_k^* \leftarrow \zeta_3 + 3h\,\mathrm{sgn}(\zeta_k - \zeta_3)$
      - if $E_{(n,m,k),(n,mov(m),k)} \notin skt(T)$ then
        - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
        - $\zeta_k^* \leftarrow \zeta_3 + 3h\,\mathrm{sgn}(\zeta_k - \zeta_3)$
      - if $E_{(n,m,k),(n,m,mov(k))} \notin skt(T)$ then
        - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
        - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
- return S^*

Output: A modified support $S^* = \{\Xi^*, \mathscr{H}^*, \mathscr{Z}^*\}$ that satisfies Condition 2.

5. Spaces spanned by nested T-meshes are also nested:
   $T_1 \subset T_2 \;\Rightarrow\; S_{T_1} \subset S_{T_2}$.

6. Order 3 polynomial reproduction property: $\mathbb{P}_3(\Omega) \subset S_T(\Omega)$.

A detailed discussion about these properties can be found in [55].

6. Method for spline parameterization of 2D and 3D geometries

In this chapter we present a method to obtain high quality spline parameterization of both, 2D and 3D geometries, for their use in Isogeometeric Analysis. This is a generalization of the strategy exposed in chapter 3 for 2D geometries. Some steps are common in both strategies, so here we focus on explaining the particular differences in detail.

We propose a method to obtain the parameterization of an object Ω (physical domain) by deforming a T-mesh of the parametric domain, the unitary cube $\widehat{\Omega} = [0,1]^d$, $d = 2,3$, until it achieves the shape of the object. 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. Due to any point can be fully determined by the local coordinates relative to the cell in which it is contained, we can define a one-to-one mapping between $\widehat{\Omega}$ and the object. Mesh deformation is accomplished by means of simultaneous T-mesh untangling and smoothing procedure based on a pointwise mean ratio quality measure instead of the usual simplex decomposition. This quality metric gains both good orthogonality and uniformity of the isoparametric curves. We have based on our previous research in hexahedral optimization (chapter 4) to propose this new T-mesh optimization procedure. Spline representation of the object is calculated by imposing interpolation conditions using the data provided by the one-to-one correspondence between the mesh of the parametric domain and the mesh of the physical object. The interpolation is performed using polynomial spline spaces (EP-splines) constructed via a technique proposed in chapter 5. This strategy allows to define easily a cubic spline space with nice properties over a given strongly balanced quadtree/octree T-mesh. Construction of the spline representation can also be performed with other spline basis defined over a T-mesh.

Globally smooth 3D parametrization entails some important limitations due to the tensor-product nature of spline functions. First, for complex domains it is necessary to subdivide the object and perform several-patches parameterization with loss of regularity in some zone of the merging. Second, for the given boundary surface of a solid it is not trivial to find optimal partition of this surface into six patches corresponding to six faces of the parametric cube. Besides, to guarantee regular parameterization (positive Jacobian), the boundary patches must merge in a convex region, which can be impossible for some geometries.

Here we are not going to tackle these problems inherent to tensor-product spline parameterization. Thus, in this study, we are dealing with genus-zero geometries given by six boundary patches that merge forming convex regions in the unions.

This chapter is organized as follows. In next section we describe the main steps of the proposed algorithm. In section 6.2 we explain the process of the construction of T-mesh adapted to the singularities of the object boundary. Section 6.3 describes the T-mesh optimization procedure which is the key of the method. The modelling of the geometry by means of cubic splines is developed in section 6.4.

6.1 General scheme of the method

The main steps are very similar to the method exposed in chapter 3. Here we summarize them:

1. Construction of an adapted T-mesh: An adapted T-mesh is generated by refining the initial mesh until the input boundary is approximated with a prescribed tolerance. During this process, the boundary nodes of the parametric domain are mapped to the boundary of the object.

2. T-Mesh optimization: The inner nodes of the physical T-mesh are relocated by applying a simultaneous mesh untangling and smoothing procedure.

3. Construction of a spline representation of the geometry: Spline representation of the object is calculated by interpolating the data provided by the transformation between the parametric and physical meshes of the object.

4. Adaptive refinement to improve the quality (optional step): If the quality of the parameterization is not satisfactory, we apply an adaptive refinement of the mesh in order to increase the degree of freedom in the areas with high distortion. Then, we return to step 2 and repeat the process until reaching a good spline parameterization.

Due to its simplicity, quadtree and octree meshes are an attractive tool for performing adaptive refinement in IGA and geometric modelling. For spline representation of the object we use EP-spline spaces. This strategy allows to define easily a cubic spline space with nice properties over a given strongly balanced quadtree/octree T-mesh. For these reasons, here we work with 0-balanced quadtree/octree T-meshes.

As was said before, the proposed parameterization algorithm is suitable for 2D and 3D geometries. It is worth noting that our 3D parameterization method is a direct generalization of the 2D case. To facilitate the illustration of the whole process, we are going to give a description of the main steps of the method both for 2D and 3D domains.

6.2 Construction of an adapted T-mesh

In this section we describe the process to construct an adapted T-mesh of the object to represent its boundary features with a desired tolerance.

6.2.1 2D case

For planar domains we assume that the input boundary can be given as four spline curves or polygonals. Also, we assume that the four curves (polygonals) meet in a convex zone (the inner angle is less than 180°). If the input boundary is given by spline

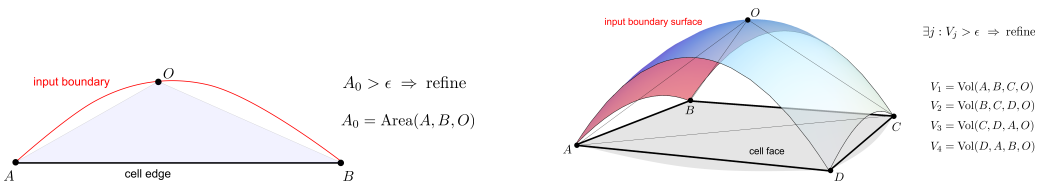


Figure 6.1. Error criterion for construction of an adapted T-mesh. (a) Error criterion for the boundary approximation in 2D. (b) Error criterion for the boundary surface approximation in 3D.

parametric curves, we take as its parametric space the sides of the square to map them to the input boundary. If the input boundary are polygonals, each edge of the unit square are mapped into its corresponding polygonal via chord-length parameterization.

Let Γ^i $(i = 1,2,\ldots,4)$ be the four boundary curves of the object Ω, and $\widehat{\Gamma}^i$ be the four sides of the parametric square $\widehat{\Omega} = [0,1]^2$. Given the four parametrizations $\Pi_b^i : \widehat{\Gamma}^i \to \Gamma^i$, we can define a global parametric mapping between the square boundary $\partial\widehat{\Omega} = \widehat{\Gamma} = \bigcup_{i=1}^4 \widehat{\Gamma}^i$ and the boundary of the object $\partial\Omega = \Gamma = \bigcup_{i=1}^4 \Gamma^i$:

$$\Pi_b : \widehat{\Gamma} \to \Gamma$$

Next, we construct an adapted quadtree T-mesh to approximate the input boundary with a predefined tolerance ε. Let consider $\widehat{\Omega}_k$ is a T-mesh of $\widehat{\Omega}$ resulting after applying several quadtree refinements according some approximation error criterion. Then after mapping the $\widehat{\Gamma}_k = \partial\widehat{\Omega}_k$ using Π_b into the physical space, we obtain a new boundary Γ_k which is an approximation for the input boundary Γ, i.e.:

$$\widehat{\Gamma}_k \to \Pi_b(\widehat{\Gamma}_k) =: \Gamma_k \approx \Gamma$$

In order to improve this approximation we must refine the cells of $\widehat{\Omega}_k$ that contacts with the boundary, in such a way that the distance between Γ_k and Γ decreases until reaching a prescribed tolerance ε. In this work, the following simple approximation criterion is used. We evaluate the area of the triangle formed by the edge AB of the physical cell and the image of the edge center O over the input boundary, see Fig. 6.1(a). A cell is refined if the error is greater than ε. Each refinement produces a new boundary point that is projected over the input boundary.

6.2.2 3D case

For 3D geometries we assume that the input boundary is given by six spline surfaces, see Fig. 6.2(a), or six surface triangulations, merging in a convex region. If the input boundary is given by spline parametric surfaces, we take as parametric space the faces of the cube. If the input surfaces are triangulations, first each triangulation is

parameterized by means of the method proposed by M. Floater in [61]. This method establishes a one-to-one mapping between the surface triangulation and a planar triangulation of the parametric domain, which is used to map a face of the unit cube to the input boundary.

Let Γ^i $(i = 1,2,\ldots,6)$ be the six boundary surfaces of the object Ω, and $\widehat{\Gamma}^i$ be the surfaces of the parametric cube $\widehat{\Omega} = [0,1]^3$. Given the parametrizations $\Pi_b^i : \widehat{\Gamma}^i \to \Gamma^i$, we can define a global parametric mapping between the cube boundary $\partial\widehat{\Omega} = \widehat{\Gamma} = \bigcup_{i=1}^6 \widehat{\Gamma}^i$ and the boundary of the object $\partial\Omega = \Gamma = \bigcup_{i=1}^6 \Gamma^i$:

$$\Pi_b : \widehat{\Gamma} \to \Gamma$$

An adapted parametric octree T-mesh $\widehat{\Omega}_k$ is constructed analogously to 2D case, that is, $\widehat{\Omega}$ is refined according some approximation error criterion. As approximation error, we evaluate the volume of the four tetrahedra *ABCO*, *BCDO*, *CDAO*, *DABO* constructed from the boundary cell face *ABCD*, where *O* is the image of the face center, see Fig. 6.1(b). The cell is refined if there is at least one volume greater that ε. A cell refinement introduces at most five new boundary points that are projected over the input surface, obtaining a more accurate approximation of the geometry. Then, after mapping the $\widehat{\Gamma}_k = \partial\widehat{\Omega}_k$ into the physical space, we obtain a new boundary Γ_k which is an approximation for the input boundary Γ, i.e.:

$$\widehat{\Gamma}_k \to \Pi_b(\widehat{\Gamma}_k) =: \Gamma_k \approx \Gamma$$

6.3 T-mesh optimization

As result of the previous stage, the boundary $\widehat{\Gamma}_k$ of the adapted T-mesh $\widehat{\Omega}_k$ is mapped to the boundary of the object, generating the surface Γ_k, which is the final approximation of Γ. The position of the inner nodes are determined by means of the T-mesh optimization procedure developed in this section. Once the inner nodes are relocated, the resulting physical T-mesh Ω_k is an approximation of the original geometry. The corresponding piece-wise volumetric parameterization is denoted as

$$\Pi : \widehat{\Omega}_K \to \Omega_K,$$

where any point p of a cell of $\widehat{\Omega}_K$ is mapped into a point q of the transformed cell of Ω_K by using an appropriate local mapping related to the cell.

The key of the proposed method lies in the optimization procedure that allows to obtain a high quality physical T-mesh, which is used to construct the spline representation of the object. To reduce the computational effort during the optimization process it is preferable to perform a previous relocation of the inner nodes. For this purpose Laplacian smoothing or Coons patch [66, 67] can be used. This previous relocation facilitates the untangling process, but in general does not obtain a satisfactory

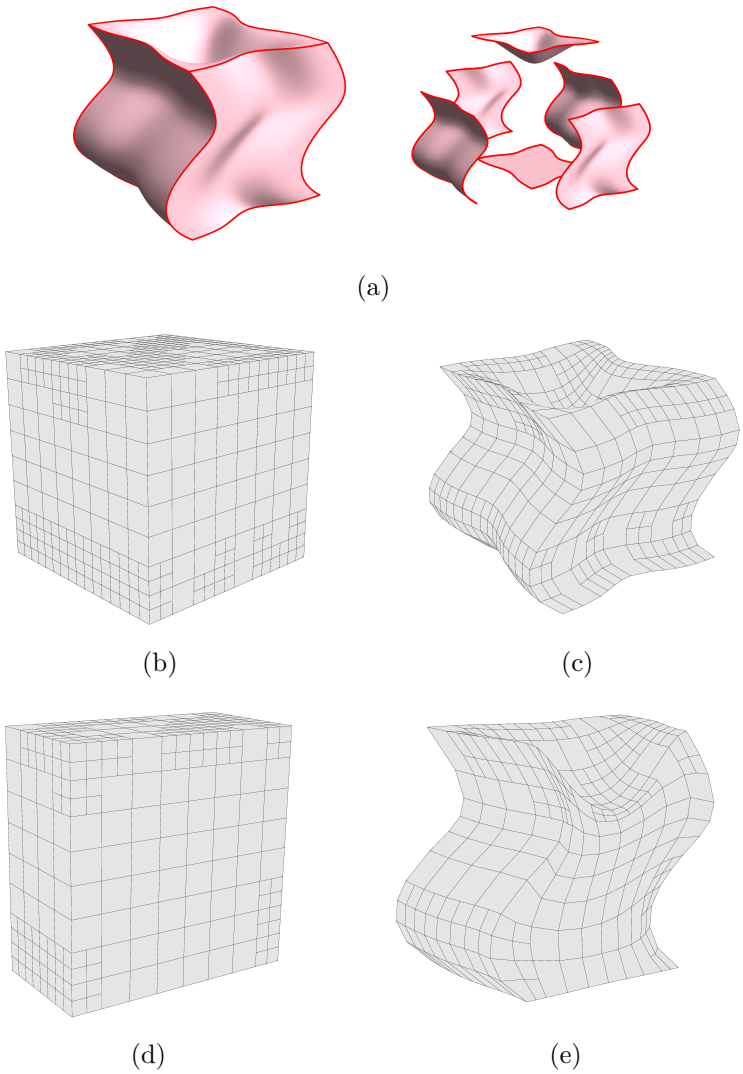


Figure 6.2. T-mesh construction for Deformed cube geometry. (a) Parametric T-mesh adapted to the boundary of the geometry. (b) Optimized physical T-mesh. (c) A section of the parametric T-mesh. (d) A section of the physical T-mesh.

mesh quality and can produce self-intersections. Therefore, it is essential to apply an effective optimization algorithm.

The mesh optimization procedure consists in an iterative process where each node is moved to a new position in order to improve the quality of the local submesh. This new position of the node is determined by minimizing an objective function based on the mean ratio shape quality measure. This shape quality measure was originally introduced for a triangle, and it is defined in terms of the Jacobian matrix of the affine mapping from ideal (target) triangle to the given one. It represents the deviation of the shape of the physical triangle from the ideal one. The distortion of an element is defined

as the inverse of its quality. To asses the distortion of a quadrilateral (hexahedral) cell we use a pointwise distortion measure defined in terms of the Jacobian matrix of the appropriate mapping from the ideal element to the physical one. Then, we can define a global distortion measure of the entire physical element by means of L^1 norm of the pointwise distortion.

The objective function for a given free node is constructed as a sum of shape distortion metrics of all elements of its local submesh. For each cell of the physical T-mesh, the corresponding parametric cell is used as its ideal element. Repeating the optimization procedure for all inner nodes we achieve to minimize the deformation of the physical mesh with respect to the parametric one.

6.3.1 Jacobian-based shape quality metric for arbitrary elements

To extend the concept of shape quality metric for non-simplicial elements (or for higher-order simplicial ones) with non-constant Jacobian matrix, we introduce the following concepts.

Let Ω_e be any type physical cell of the local mesh of the free node $\mathbf{x}$. First, we define a pointwise distortion measure for the given cell using some appropriate mapping $\mathbf{G} : \widehat{\Omega}_e \to \Omega_e$ from the ideal element to the physical one. Here, we will use quadratic mapping.

The distortion of a element Ω_e at any point $\boldsymbol{\xi}$ of the ideal element is defined as

$$\eta^*(\boldsymbol{\xi}) = \frac{\|\mathbf{J}_G(\boldsymbol{\xi})\|^2}{d\, h(\gamma(\boldsymbol{\xi}))^{2/d}}, \quad \boldsymbol{\xi} \in \widehat{\Omega}_e,$$

where $d = 2,3$, $\mathbf{J}_G(\boldsymbol{\xi}) = (\partial \mathbf{G}/\partial \boldsymbol{\xi})$ is the Jacobian matrix of the mapping $\mathbf{G}$ at the point $\boldsymbol{\xi}$, $\|\mathbf{J}_G(\boldsymbol{\xi})\|$ is its Frobenius norm and $\gamma(\boldsymbol{\xi}) = \det(\mathbf{J}_G)$. Function h is defined as $h(\gamma) = \frac{1}{2}(\gamma + \sqrt{\gamma^2 + 4\delta^2})$, see [60].

The global distortion measure for the physical cell Ω_e can be defined through L^1 norm of the pointwise distortion, namely

$$\eta^*_{\Omega_e} = \frac{1}{V_{\widehat{\Omega}_e}} \int_{\widehat{\Omega}_e} \eta^*(\boldsymbol{\xi})\, d\widehat{\Omega}_e,$$

where $V_{\widehat{\Omega}_e}$ is the volume of $\widehat{\Omega}_e$. Other norms, for example L^2 can be used.

The global distortion $\eta^*_{\Omega_e}$ takes values between 1 and ∞. If the physical element Ω_e coincides with the ideal one, then $\eta^*(\boldsymbol{\xi}) \to 1$ when $\delta \to 0$. On the other hand, this global distortion measure presents pseudo-barriers, that is, $\eta^*_{\Omega} \to \infty$ when $\sigma(\boldsymbol{\xi}) \leq 0$ and $\delta \to 0$.

Global quality of the element is defined as the inverse of its global distortion, i.e.

$$q^*_{\Omega_e} = \frac{1}{\eta^*_{\Omega_e}},$$

and it takes values from $[0,1]$.

We define the objective function for a free node $\mathbf{x}$ using the introduced global distortion measure for the cells

$$K^*(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \eta^*_{\Omega_i}(\mathbf{x}),$$

being N the number of cells connected to the free node $\mathbf{x}$ and $\eta^*_{\Omega_i}(\mathbf{x})$ the global distortion for the i-th cell.

The integral is evaluated by a numerical quadrature. Let $\{\boldsymbol{\xi}_j\}_{j=1,M}$ be a set of quadrature points over the ideal element $\widehat{\Omega}_e$. Then the objective function for the free node $\mathbf{x}$ becomes

$$K^*(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \left(\sum_{j=1}^{M} w_j \eta^*_{\Omega_i}(\boldsymbol{\xi}_j) \right),$$

where w_j are the corresponding weights for the quadrature points $\boldsymbol{\xi}_j$. Thus, in practice, the objective function is based on the pointwise distortion measure for each cell, evaluated at certain set of sample points within the ideal cell. Analogously to triangular mesh, the unconstrained optimization problem can be easily solved with any standard method, see for example [73].

In order to not increase the computational cost of the minimization it is preferable to use few quadrature points. Also, it is preferable to evaluate the quality at the corner vertices of the cell, because usually the major distortion occurs at these points. Therefore we use Gauss-Lobatto quadrature rule that includes boundary quadrature points. More specifically, we use three quadrature points $\{-1,0,1\}$ of the reference interval $[-1,1]$.

Parametric mapping for 2D T-mesh elements

As was said before, for each cell of the physical T-mesh its counterpart in parametric domain is used as its ideal element. In our particular case of quadtree subdivisions all parametric cells are squares, so we take the master element $\widehat{\Omega}_e = [-1,1]^2$ as the ideal element for all cells. Physical cells of our T-mesh, in general, are not straight-sided quadrilaterals since they can contain hanging nodes, which are not aligned with the nodes of its edge. For our balanced quadtree T-meshes an edge can contain at most one hanging node. To take into account all mesh nodes for the distortion measure of a given cell, we can consider curvilinear second order element constructed via Lagrangian interpolation. Let $\boldsymbol{\xi}_i \in \mathbb{R}^2$, $i = 1,..9$, be the nodes of the master element $\widehat{\Omega}_e$, and N_i are the nodal biquadratic Lagrangian shape functions defined on $\widehat{\Omega}_e$. Then, parametric mapping

$$\mathbf{G}(\boldsymbol{\xi}) = (x(\xi,\eta), y(\xi,\eta)) = \sum_{i=1}^{9} \mathbf{x}_i N_i(\boldsymbol{\xi}), \tag{6.1}$$

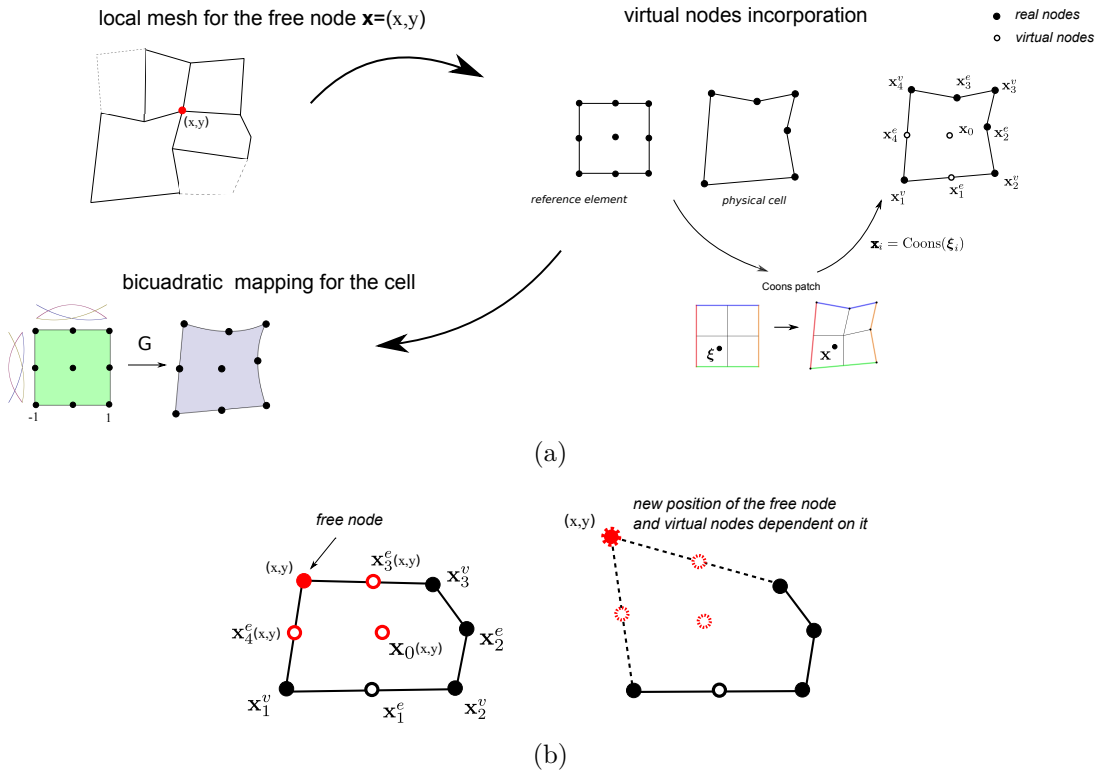


Figure 6.3. Virtual nodes assignment for 2D case. (a) Defining a local quadratic mapping for each cell. (b) Virtual nodes connected to the free node update their position when the free node is moved during optimization process.

maps $\widehat{\Omega}_e$ into the physical cell Ω_e with the nodes $\mathbf{x}_i$, $i = 1,9$. So, to define this quadratic transformation, 9 cell nodes are needed. In our case some of these 9 nodes may not exist in the mesh, so we have to complement a given cell with some virtual nodes. The position of these virtual nodes depends on the real (existing) nodes of the cell, see Fig. 6.3(a). To assign reasonable positions to the virtual nodes we appeal to Coons patch interpolation of the boundary edges of the cell. Suppose we define parameterizations of the four edges as $\mathbf{x}(0,\eta)$, $\mathbf{x}(1,\eta)$, $\mathbf{x}(\xi,0)$ and $\mathbf{x}(\xi,1)$. Bilinear Coons patch is defined as

$$\begin{aligned}\mathbf{x}(\xi,\eta) =& (1-\xi)\mathbf{x}(0,\eta)+\xi\mathbf{x}(1,\eta)\\ &+(1-\eta)\mathbf{x}(\xi,0)+\eta\mathbf{x}(\xi,1)\\ &-\begin{bmatrix} 1-\xi & \xi \end{bmatrix}\begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1)\\ \mathbf{x}(1,0) & \mathbf{x}(1,1)\end{bmatrix}\begin{bmatrix} 1-\eta\\ \eta\end{bmatrix}\end{aligned}$$

Doing so, we obtain the following positions for the virtual nodes of the cell. The missing edge nodes are placed in the midpoint of their corresponding edges, i.e., fo-

llowing the notation of the Fig. 6.3(a), the virtual node of the cell edge $(\mathbf{x}_k^v, \mathbf{x}_{k+1}^v)$ is defined as

$$\mathbf{x}_k^e = \frac{1}{2}(\mathbf{x}_k^v + \mathbf{x}_{k+1}^v), \quad k \in \{1,2,3,4\}. \tag{6.2}$$

Then, to place the interior center node of the cell we apply $\mathbf{x}(1/2,1/2)$ and obtain the following expression for center node in terms of the boundary nodes of the cell

$$\mathbf{x}_0 = \frac{1}{2}(\mathbf{x}_1^e + \mathbf{x}_2^e + \mathbf{x}_3^e + \mathbf{x}_4^e) - \frac{1}{4}(\mathbf{x}_1^v + \mathbf{x}_2^v + \mathbf{x}_3^v + \mathbf{x}_3^v). \tag{6.3}$$

Note that virtual nodes do not introduce additional degrees of freedom for the mesh, since their position depends on the existing mesh nodes, in particular on the coordinates (x,y) of the free node. Thus, the virtual nodes connected to the free node update their position when the free node is moved during optimization process, see Fig. 6.3(b).

Parametric mapping for 3D T-mesh elements

Generalization of the ideas for 3D T-mesh optimization is straightforward. To obtain a global shape distortion measure for each cell we define a triquadratic mapping from the master element $\widehat{\Omega}_e = [-1,1]^3$ to the physical one, which requires 3^3 mesh nodes. A cell is complemented with virtual nodes calculated in terms of the existing nodes of the cell. First, virtual nodes are assigned to each face of the cell in order to define biquadratic mapping for the faces. For that we apply exactly the same procedure developed for 2D case using Coons interpolation of the four edges of the face. Some faces may contain all the necessary 9 nodes, and therefore there is no need to add virtual nodes for them. If the face misses some nodes, then its virtual mid-side nodes and the face center node is assigned using the formula (6.2) and (6.3). When all the six faces have their biquadtratic parametric representation, we define 3D Coons patch interpolating the six faces in order to define the position of the central virtual node of the cell, see Fig. 6.4. To place the interior center node of the cell we apply $\mathbf{x}(1/2,1/2,1/2)$ and obtain the following expression:

$$\mathbf{x}_0 = \frac{1}{8}(\mathbf{x}_1^v + \mathbf{x}_2^v + \cdots + \mathbf{x}_8^v) - \frac{1}{4}(\mathbf{x}_1^e + \mathbf{x}_2^e + \cdots + \mathbf{x}_{12}^e) + \frac{1}{2}(\mathbf{x}_1^f + \mathbf{x}_2^f + \cdots + \mathbf{x}_6^f),$$

where $\mathbf{x}_i^v$, $i = 1,8$ are the corner-vertex nodes, $\mathbf{x}_i^e$, $i = 1,12$ are the mid-side edge nodes and $\mathbf{x}_i^f$, $i = 1,6$ are the face center nodes.

Then, a triquadratic mapping from the master element $\widehat{\Omega}_e$ into the physical one Ω_e given by its nodes $\mathbf{x}_i$, $i = 1,27$, is defined as

$$\mathbf{G}(\boldsymbol{\xi}) = (x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)) = \sum_{i=1}^{27} \mathbf{x}_i N_i(\boldsymbol{\xi}),$$

where N_i are the triquadratic Lagrangian shape functions defined on the master element.

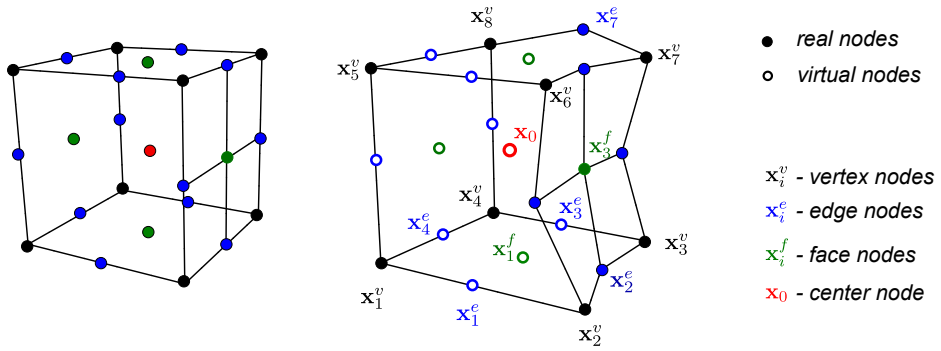


Figure 6.4. Virtual nodes assignment for 3D case.

Remark 2 Note that the distortion measure for a triangle defined in terms of its constant Jacobian matrix is a particular case of global distortion measure based on the L^1 norm of the pointwise distortion defined for a given arbitrary element. Also, the pointwise shape distortion measure of any mapping $\mathbf{G}$ is related to the shape distortion measure for a triangle as follows: the value of the distortion measure at any point $\mathbf{P}_0$ of the master element is the distortion measure of the infinitesimal triangle in physical domain formed by two isoparametric curves passing through the point $\mathbf{P}_0' = \mathbf{G}(\mathbf{P}_0)$.

Remark 3 Another way to evaluate the distortion of a non-simplicial element is to decompose it into several overlapping simplicial elements and define the distortion as a sum of the distortions of each triangle of this decomposition. We have used this approach in chapter 3 for the T-spline parameterization of 2D geometries. However, this approach has some shortcomings due to the large variety of possible simplicial decompositions, especially in 3D case, and can not be considered as a generalization of the shape distortion measure originally introduced for a linear triangular (tetrahedral) element.

6.3.2 Weighted objective function

The objective function given in section 6.3.1 is a sum of the terms corresponding to global distortion measure of each cell. A conformal uniform mesh is a particular case of our quadtree T-mesh. In this case each free node is surrounded by cells of the same scale, in the sense that their counterparts in parametric domain have the same size. For this situation it is reasonable to compose the objective function by the sum of global distortion measures. However, for a general quadtree T-mesh, a free node can be surrounded by cells of different scales. In computational experiments we have observed that for these case a weighted objective function produces better results. Namely, we can define the objective function as follows

$$K^*(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} w_i\, \eta^*_{\Omega_i}(\mathbf{x}),$$

where the weights w_i have some relation with the scale of the corresponding cell, that is, the cells of the same scale will have the same weight. For our balanced quadtree/octree meshes a free node can be surrounded by cells of at most two different levels. So we need two different weights for our terms. Putting the smallest weight to 1 for cell of the level k, we need one value for the weight of the cells of level $k-1$. Assigning different values to this weight we can obtained slightly different physical T-meshes. In practice we have observed that in several cases, parametrization results are better using weight $w_i = 2$ for the biggest cells of local mesh. Weighted objective function tends to obtain better orthogonality of the parametric curves in some cases.

6.4 Construction of a spline representation of the geometry

In this section we construct a spline representation of the geometry. For simplicity, the process is described for 2D being the generalization to 3D obvious.

We have to obtain a global one-to-one spline transformation that maps the parametric domain into the physical one $\mathbf{S} : \widehat{\Omega} = [0,1]^2 \to \Omega$. For this purpose we use EP-spline spaces constructed via the strategy proposed in chapter 5. According to this strategy, each regular node of the T-mesh has one blending function assigned. Similarly to T-splines [30], local knot vectors for each function are inferred by walking across the mesh and traversing its edges. Then, in order to span spaces with nice properties, some of the inferred function supports have to be modified applying certain modification rules.

6.4.1 Interpolation

Spline representation of the physical domain is build as lineal combination of EP-spline blending functions defined over the adapted parametric T-mesh $\boldsymbol{T}$

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A_T} \mathbf{P}_\alpha \widehat{N}_\alpha(\boldsymbol{\xi})$$

where $\mathbf{P}_\alpha \in \mathbb{R}^2$ is the control point corresponding to the α-th blending function.

Control points $\mathbf{P}_\alpha$ are found by imposing interpolation conditions. A standard choice for an interpolation with B-splines is Greville abscissae, which is the averages of the knots. These points correspond approximately to the points where the B-spline functions attain their maximum. Generalization of this concept for T-mesh leads to so-called Greville collocation points. Dealing with T-mesh, we do not have global knot vector, so a set of Greville points is obtained as the knot average of each local knot vector, that is, for the local knot vectors $\Xi_\alpha = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$ and $\mathscr{H}_\alpha = \{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5\}$ of the function $\widehat{N}_\alpha$, its Greville point is defined as $\boldsymbol{\xi}_\alpha = (\xi_\alpha, \eta_\alpha)$, where

$$\xi_\alpha = \frac{\xi_2 + \xi_3 + \xi_4}{3} \qquad \text{and} \qquad \eta_\alpha = \frac{\eta_2 + \eta_3 + \eta_4}{3}.$$

In the case of our EP-spline functions, Greville points coincide with their anchors and thus with the vertices of the T-mesh, except for the functions that contain three repeated boundary knots. For each vertex $\boldsymbol{\xi}_\alpha$ in the parametric space its image $\mathbf{x}_\alpha$ in the physical space was determined by the mesh optimization process.

Finally we solve the linear system of equations

$$\mathbf{x}_\beta = \mathbf{S}\left(\boldsymbol{\xi}_\beta\right) = \sum_{\alpha \in A_T} \mathbf{P}_\alpha N_\alpha\left(\boldsymbol{\xi}_\beta\right), \quad \forall \boldsymbol{\xi}_\beta,\ \beta \in A_T$$

where $\boldsymbol{\xi}_\beta$ are interpolation points in parametric space and $\mathbf{x}_\beta$ are their images in the physical space.

We remark that construction of the spline representation can also be performed with other spline basis defined over a T-mesh.

6.4.2 Quality assessment and its improvement

To assess the pointwise quality of the obtained parameterization we appeal to the same idea used in the optimization procedure. Namely, our objective function was based on the pointwise quality measure of the local quadratic mapping for each cell. Now we can analyze the pointwise quality measure of the constructed spline transformation. That is, we calculate at each point of the domain the mean ratio Jacobian given by

$$q_s(\boldsymbol{\xi}) = \frac{d\,\sigma(\boldsymbol{\xi})^{2/d}}{\|\mathbf{J}_S\|^2},$$

where $\mathbf{J}_s$ is the Jacobian matrix of the mapping $\mathbf{S}$ at the point $\boldsymbol{\xi}$ of the parametric domain and $\sigma(\boldsymbol{\xi}) = \det(\mathbf{J}_s)$. In contrast to the scaled Jacobian, that represents a quality of the mapping $\mathbf{S}$ in the sense of the orthogonality of its isoparametric curves, the mean ratio Jacobian represents both: a quality of the mapping in the sense of the orthogonality and uniformity. Mean ratio Jacobian is equal 1 at the point $\mathbf{P}_0$ if the mapping conserves orthogonality and produces the same length distortion in both parametric directions, i.e. the mapping is conformal at this point.

Parameterization of complex geometries entails a severe distortion that can lead to low quality mesh cells, or even with negative Jacobian. In some cases, the optimization procedure cannot obtain a good-quality mesh due to the lack of degrees of freedom provided by the inner nodes; in other words, the refinement level of the mesh is not sufficient to parameterize the given geometry. In these cases, in order to improve the mesh quality and the resulting spline parameterization, we have to increase the number of degrees of freedom in that areas. We adopt the following adaptive strategy: we refine the cells where the spline parameterization has low quality and optimize again the refined mesh. More precisely, for each cell of the mesh, the mean ratio Jacobian of the spline parameterization is calculated at some sample points (normally quadrature points). A cell $\widehat{\Omega}_e$ is marked to refine if, at least, one of its sample points has mean ratio Jacobian less than a certain threshold δ. The refined T-mesh is optimized again, and the process is repeated until a satisfactory quality is obtained.

7. Results and applications

Here we test the proposed strategies for spline parameterization using different geometries. For all of them, we take as input data compatible patches that describe the geometry boundary. To check the spline parameterization validity, we evaluate its mean ratio Jacobian q_S at 4 Gauss quadrature points per parametric direction in each cell. We consider that a parameterization is valid if for all cells $q_s(\boldsymbol{\xi}_i) > 0$, being $\boldsymbol{\xi}_i$ the quadrature points at the cell. When it is necessary, we perform adaptive refinements in order to improve the final parameterization, as was explained in the description of the strategy.

For all showed geometries, we obtain a valid spline parameterization suitable for using in Isogeometric Analysis. We present statistics data about the behavior of the mean ratio Jacobian for each parameterized geometry. At each cell, we evaluate the average, minimum and maximum value of the mean ratio Jacobian. Different type of problems are resolved with IGA using the parameterized geometries. We also expose another application of the T-mesh optimization algorithm described in chapter 6: optimization of high order quadrilateral and hexahedral meshes for FEM.

The parameterization methods and Isogeometric Analysis codes were implemented in Wolfram Language. Moreover, we have developed a 3D optimization C++ library that can be used for 0-balanced T-mesh optimization and for lineal or high order hexahedral mesh optimization. We plan to release all this software.

This chapter is organized as follows. In the first section, some examples illustrate the effectiveness of the 2D T-spline parameterization method introduced initially in chapter 3. Several examples of EP-spline parameterization of both 2D and 3D geometries are showed in the second section. In these cases, the geometries were parameterized using the general strategy defined in chapter 6. Then, in section 3, we resolve different type of problems with Isogemetric Analysis using some of the parameterized geometries. Finally, in the last section, examples of unstructured quadrilateral and hexahedral second order meshes optimization are presented.

7.1 T-spline parameterization of 2D geometries

In these examples, the input consist of 4 compatible polylines that are mapped to each edge of the parametric space by means of chord-length parameterization. Coloured lines showed at figures represent correspondence between input polygonal and parametric boundary edges.

7.1.1 Spot geometry

Figures 7.1(a) and (b) show the parametric T-mesh and physical T-mesh after optimization. Figure 7.2(b) shows the resulting T-spline representation and the color map of the mean ratio Jacobian, represented in physical domain. The method obtains a mesh with 844 cells, and 1456 control points were necessary to generate the T-spline representation. Jacobian of the T-spline parameterization is positive at all quadrature points of each cell, and in this case, no adaptive refinement was applied. The mini-

mum value of mean ratio Jacobian at the quadrature points is 0.31. Figure 7.2(a) shows statistics data of the mean ratio Jacobian. The mean ratio Jacobian is sorted by the increasing order of the average value on the cell. The red and blue lines correspond to the maximum and minimum values in each cell, respectively. The highest oscillation of the Jacobian in a cell was 0.46.

7.1.2 Gran Canaria island geometry

Figures 7.3(a) and (b) show the parametric T-mesh and physical T-mesh after optimization. We have constructed the adapted T-mesh starting from an initial uniform 8×8 mesh. Adaptive refinements were applied in order to improve the mesh quality in some areas with high distortion. In particular, three iterations were necessary to obtain a parametric mapping with mean ratio Jacobian greater than 0.2 at quadrature points. The resulting mesh has 3577 cells and for the T-spline parameterization 6054 control points were necessary. Figures 7.4(a) and (b) show statistics of the mean ratio Jacobian and the T-spline representation of the Gran Canaria Island geometry.

7.1.3 Flower geometry

Another example is shown in Fig. 7.5 and 7.6. In this case, one adaptive refinement was necessary to reach a mean ratio Jacobian greater than 0.15 at quadrature points. The generated mesh has 2323 cells and the T-spline representation has 3935 control points.

7.1.4 Hole geometry

Figures 7.7(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. The resulting mesh has 2287 cells. Figures 7.8(a) and (b) show mean ratio Jacobian statistics, and the T-spline representation and the color map of the mean ratio Jacobian. For the T-spline representation 3363 control points were necessary. We have constructed the adapted T-mesh starting from an initial uniform 8×8 mesh. In this case, no adaptive refinement was applied. Minimal value of mean ratio Jacobian at quadtrature points is 0.138.

7.1.5 Puzzle geometry

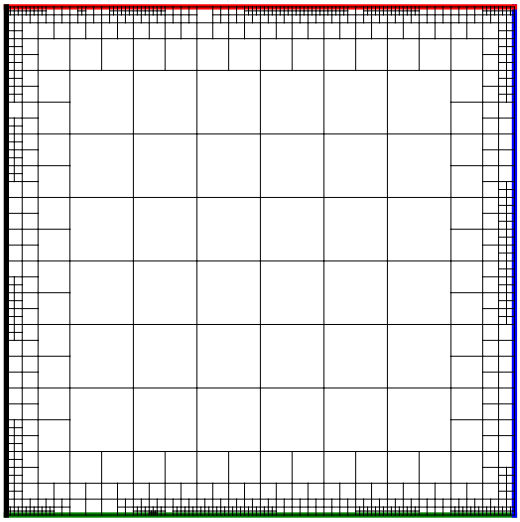
Figures 7.9(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed the adapted T-mesh starting from an initial uniform 32×32 mesh in order to increase the degree of freedom in the inner of the geometry. Only one adaptive refinement was necessary to improve the mesh quality and assure positive Jacobian in all quadrature points. The resulting mesh has 3748 cells and the T-spline has 5946 control points. Figures 7.10(a) and (b) show mean ratio Jacobian statistics, and the T-spline representation and the color map of the mean ratio Jacobian. The minimal value of mean ratio Jacobian at quadrature points is 0.19.

7.1.6 Embedded Test geometry

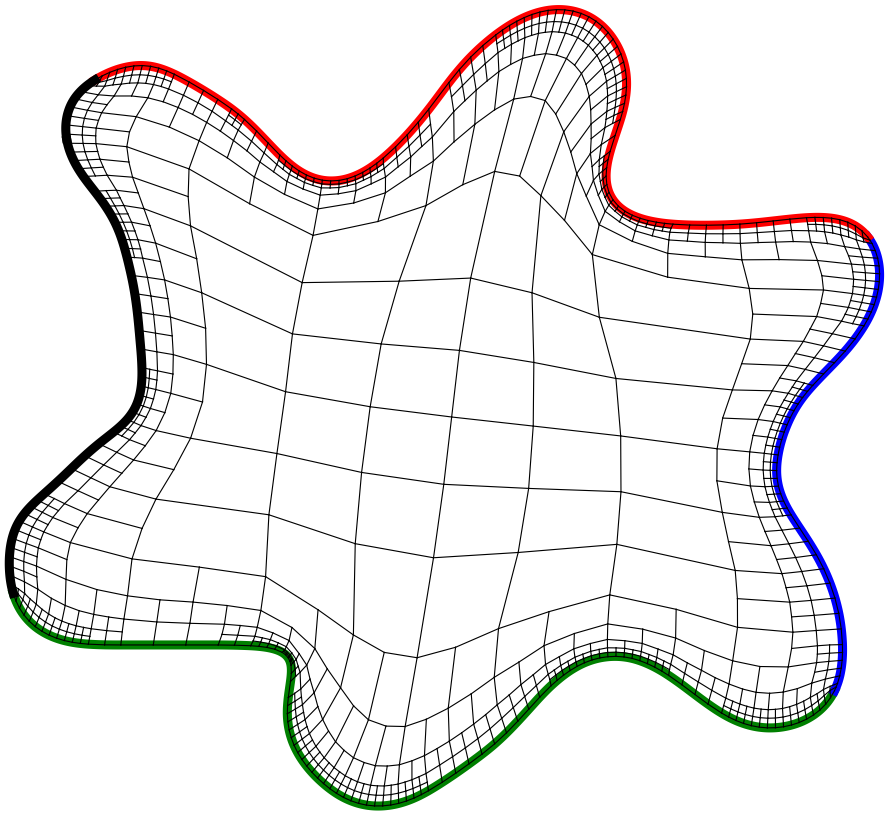
Figures 7.11 and 7.12 show the meshes and the parameterization of the Embedded Test geometry exposed in 3.6. Each figure is parameterized individually, obtaining their parametric and physical T-meshes. Then, the parametric mesh of the inner geometry is inserted in a set of cells belonging to the outer parametric quadtree. Once we have a global parametric space and its corresponding image, we optimize the physical T-mesh starting at the node positions obtained in the individual construction of each figure. In this untangling step, the nodes of the inner boundaries are fixed in the same manner that the outer boundary. The resulting T-mesh has 916 cells and the T-spline representation has 1317 control points. In this case no adaptive refinement was necessary and minimum value of the mean ratio Jacobian at quadrature points is 0.26.

7.1.7 Embedded Complex geometry

Here we generate a more complex example of embedded geometries. In this case, the Spot geometry is inserted into the Gran Canaria island geometry. Figures 7.13(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. The resulting T-mesh has 6466 cells and the T-spline parameterization has 9439 control points. No adaptive refinements were necessary to obtain a valid parameterization and minimum value of the mean ratio Jacobian at quadrature points is 0.04. Figures 7.14(a) and (b) show statistics of the mean ratio Jacobian and the T-spline representation.

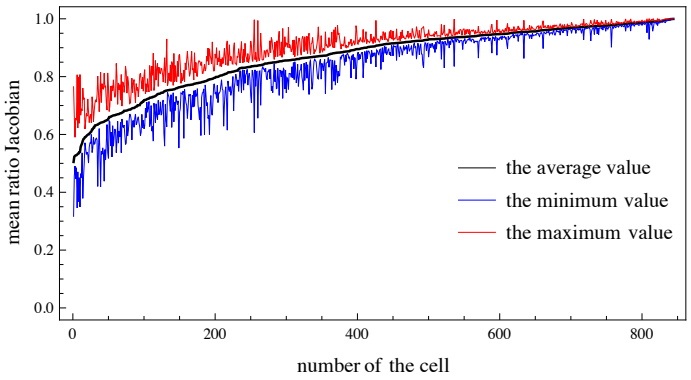


(a)

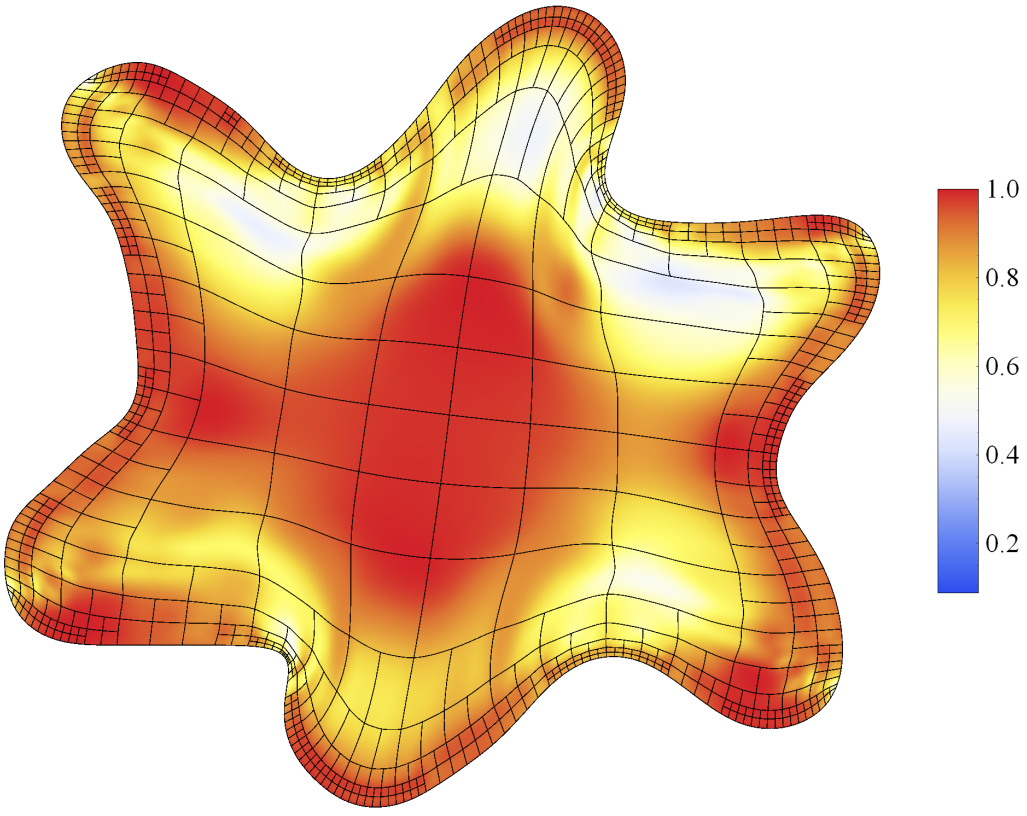


(b)

Figure 7.1. Spot geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

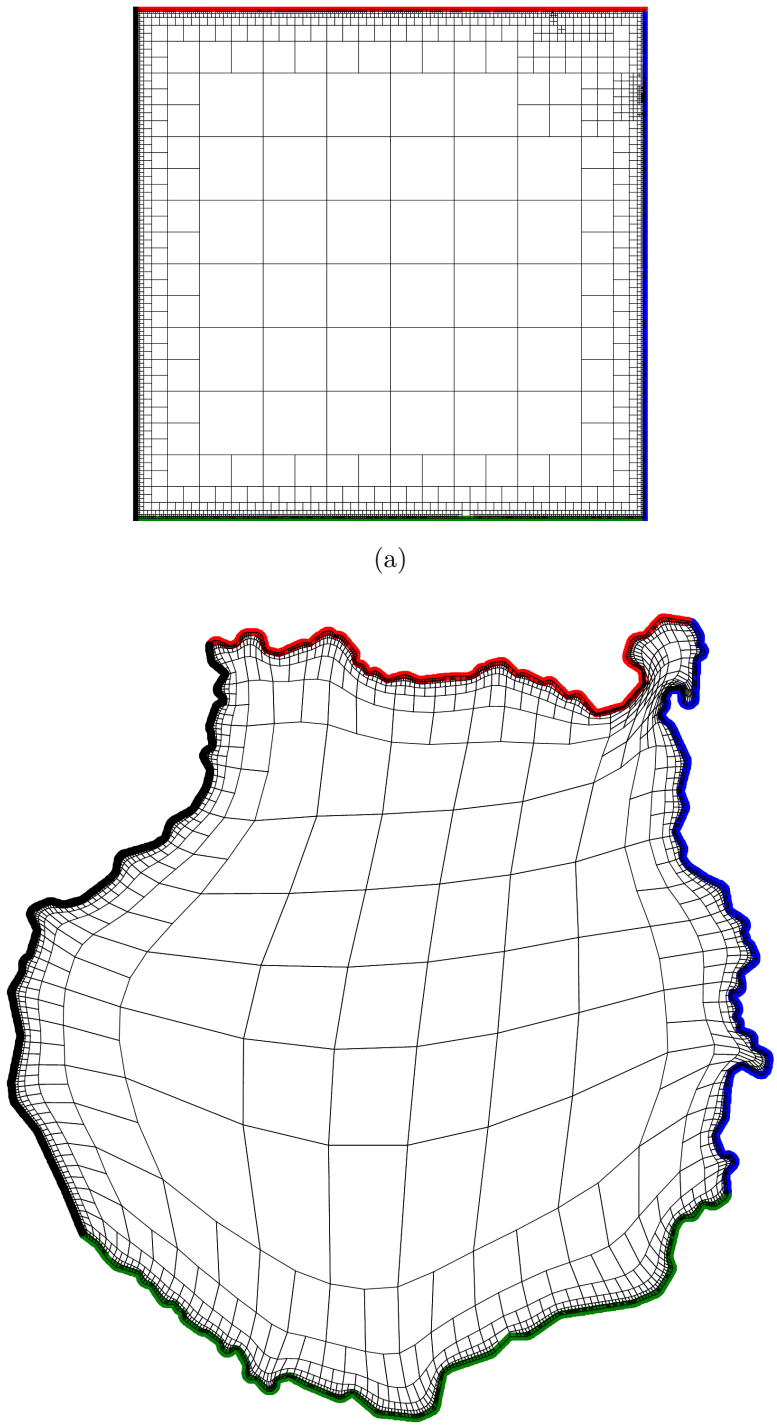


(a)



(b)

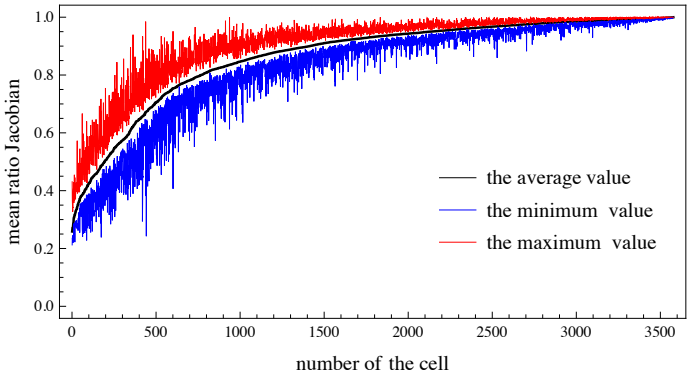
Figure 7.2. Spot geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.



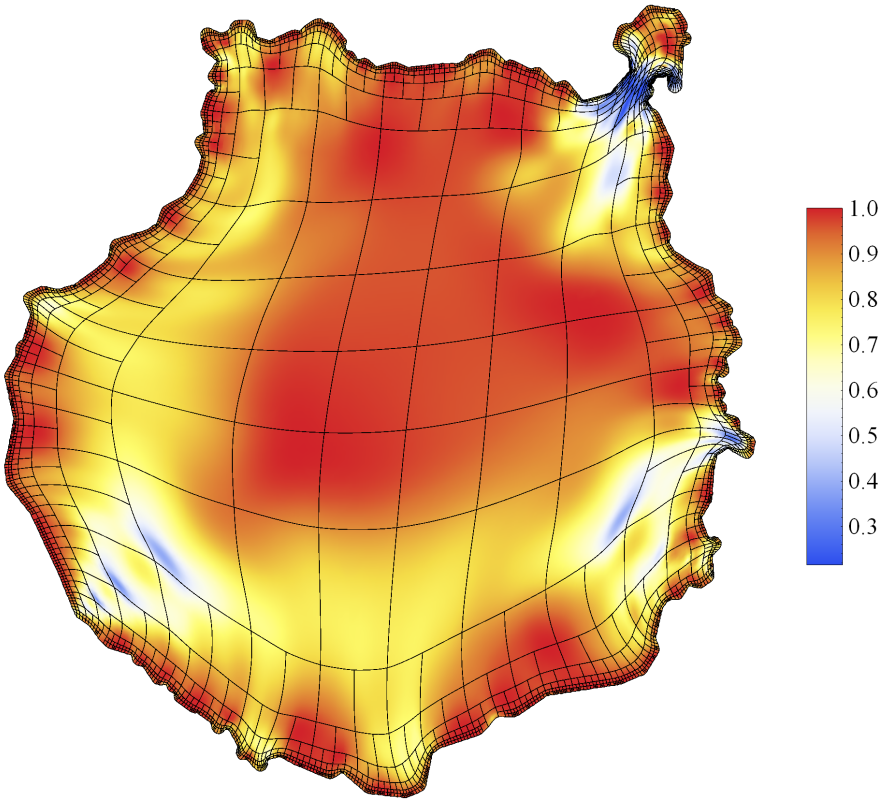
(a)

(b)

Figure 7.3. Gran Canaria Island geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

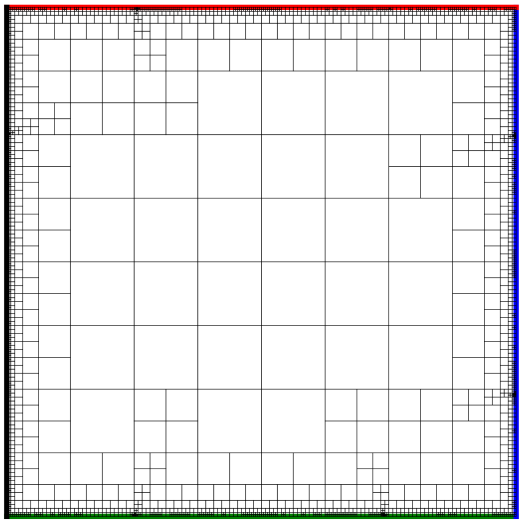


(a)

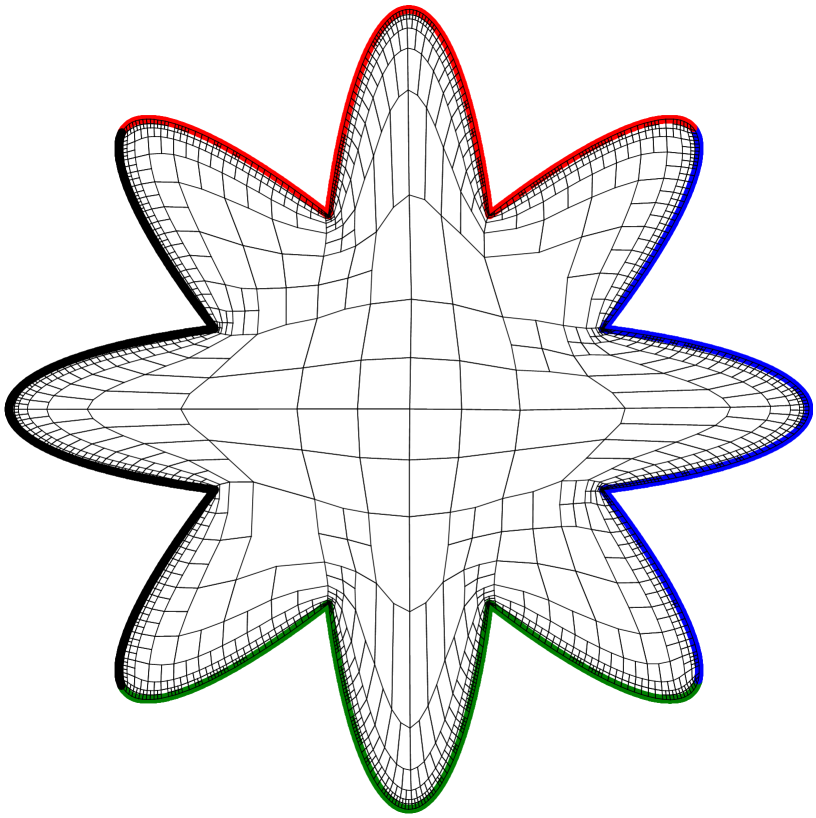


(b)

Figure 7.4. Gran Canaria Island geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.

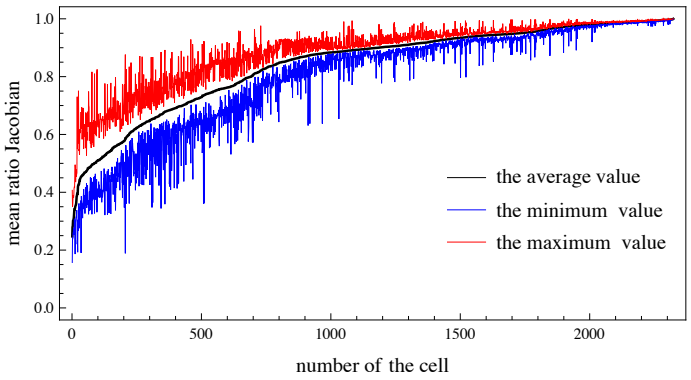


(a)

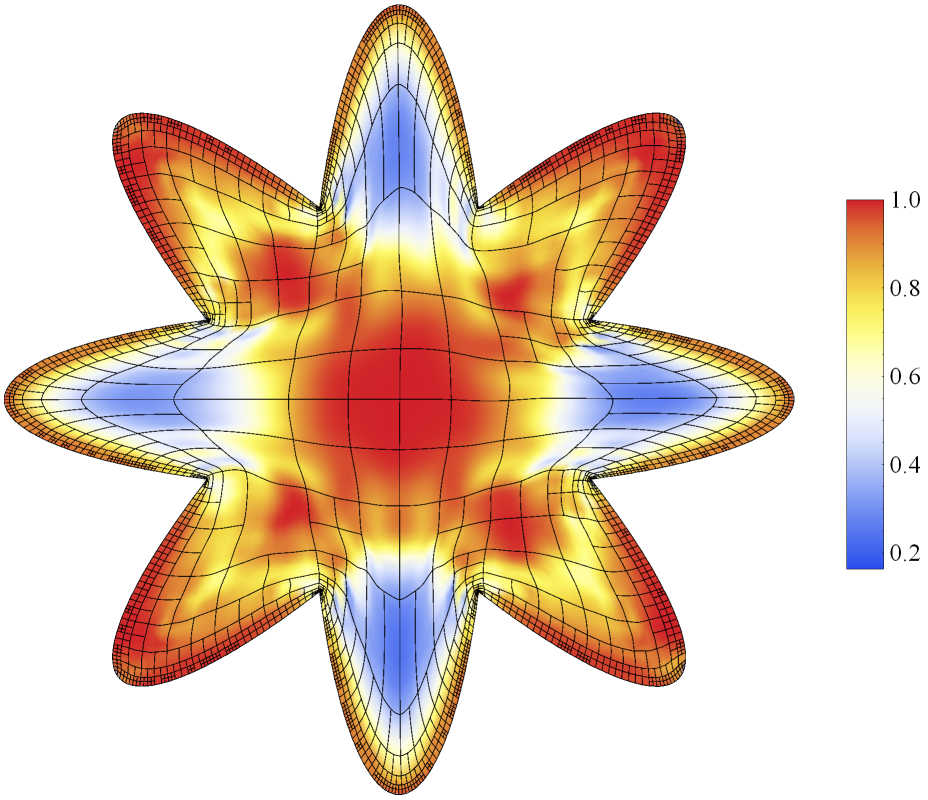


(b)

Figure 7.5. Flower geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

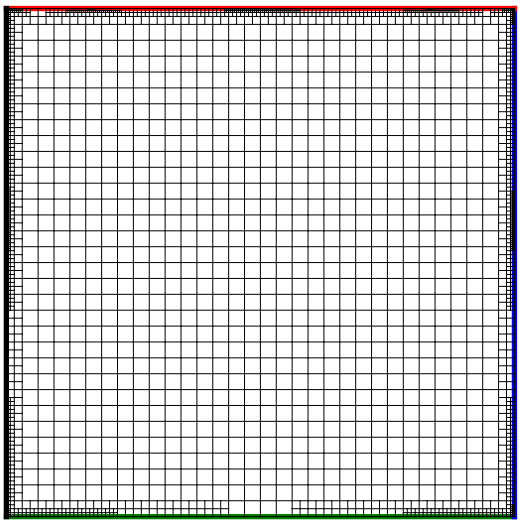


(a)

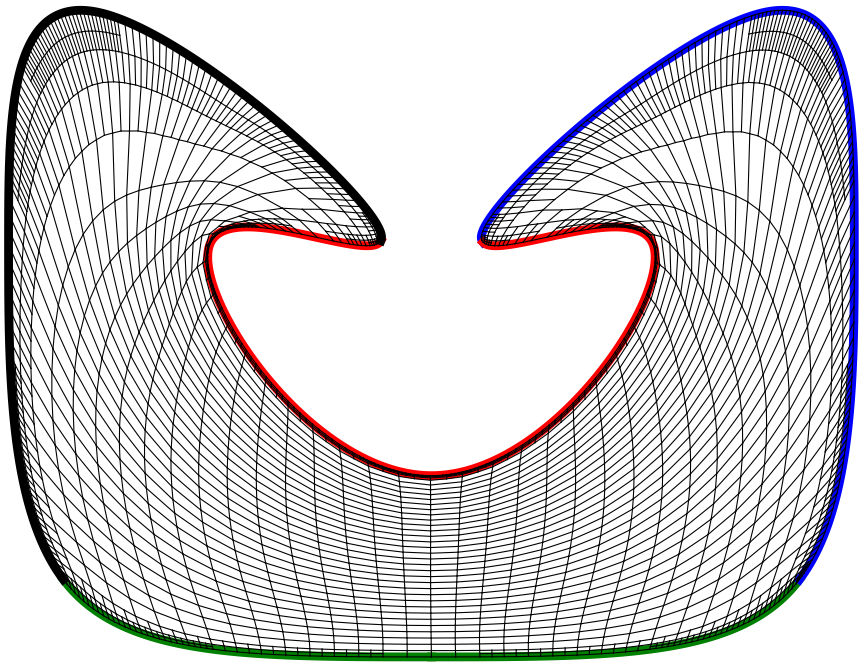


(b)

Figure 7.6. Flower geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.

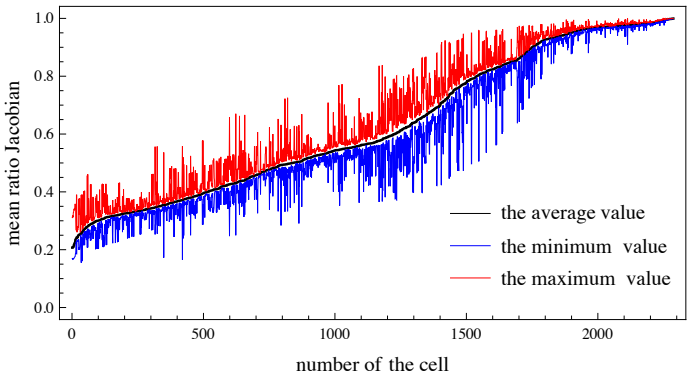


(a)

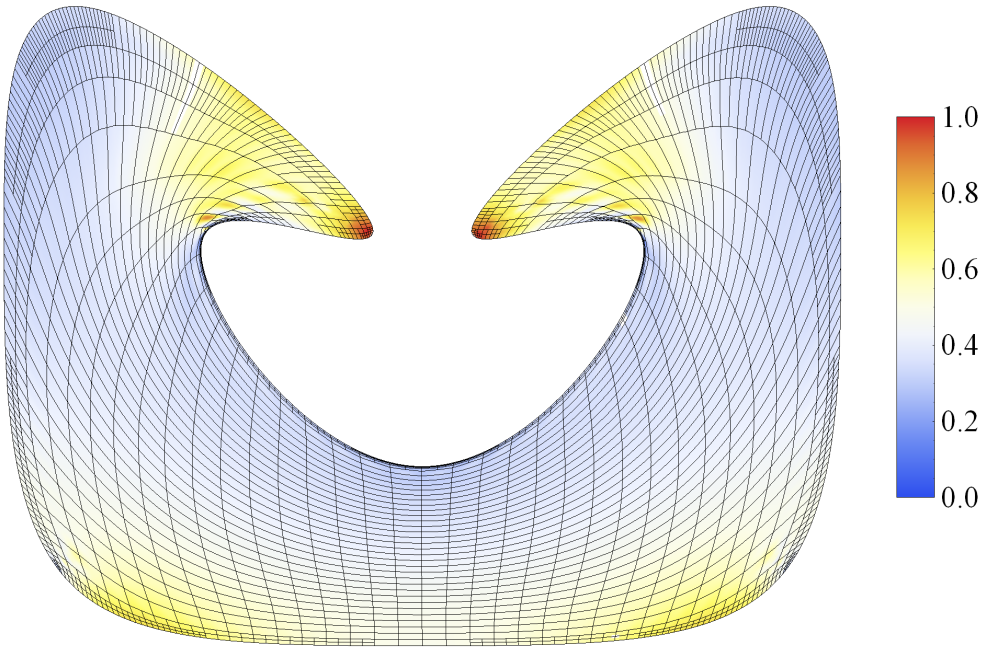


(b)

Figure 7.7. Hole geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

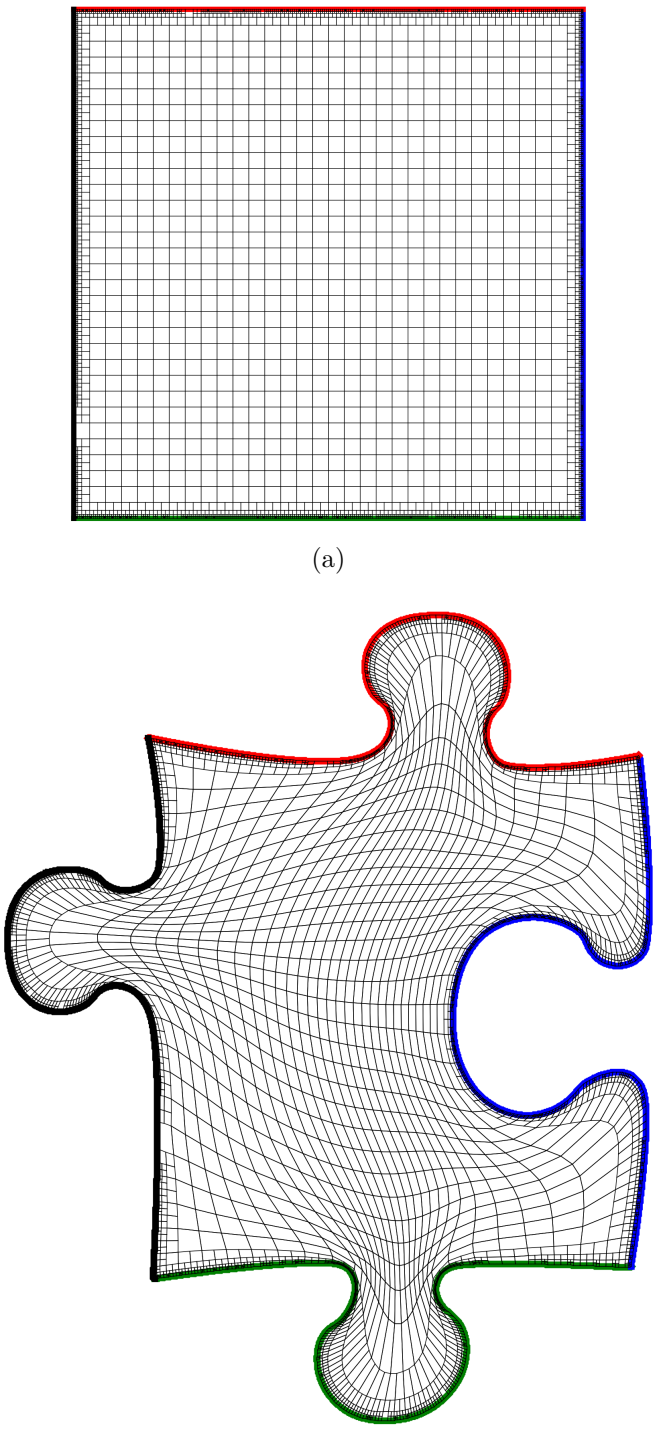


(a)



(b)

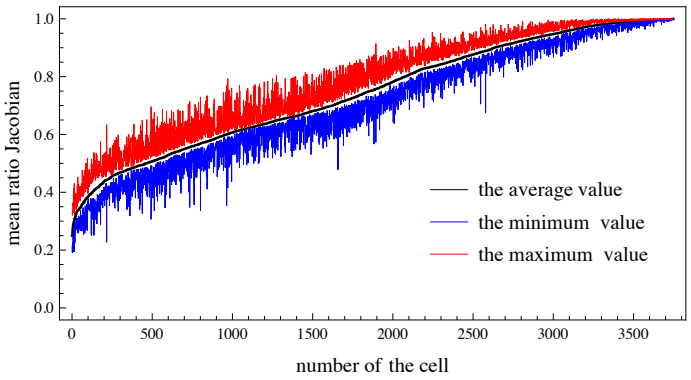
Figure 7.8. Hole geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.



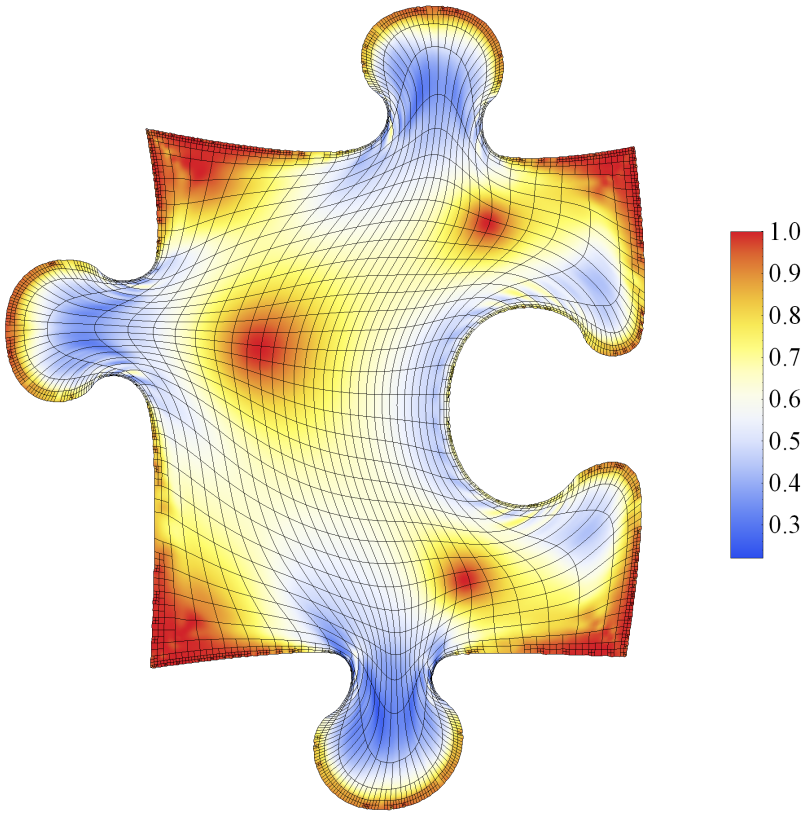
(a)

(b)

Figure 7.9. Puzzle geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

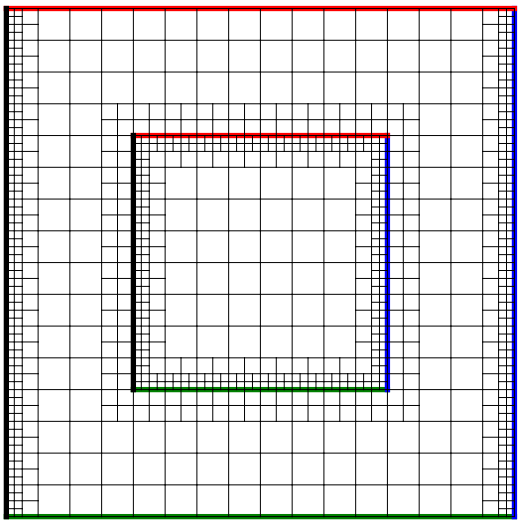


(a)

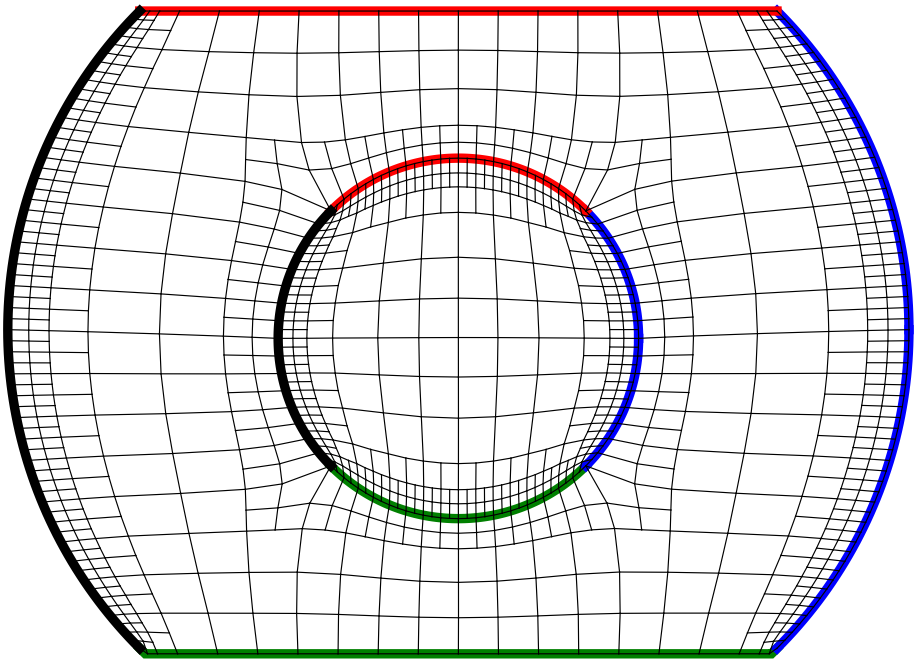


(b)

Figure 7.10. Puzzle geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.

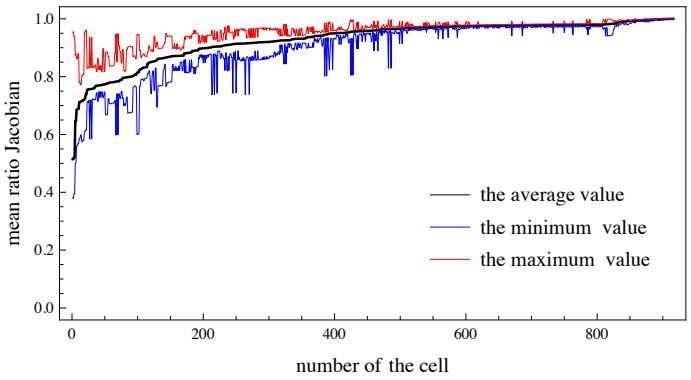


(a)

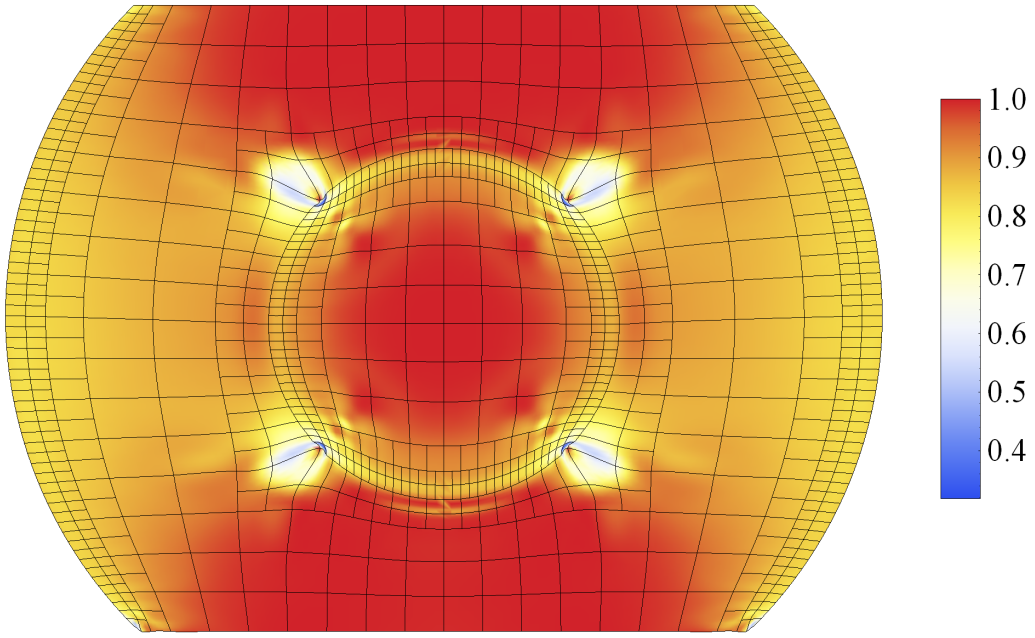


(b)

Figure 7.11. Embedded Test geometry mesh. (a) Global parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometries.

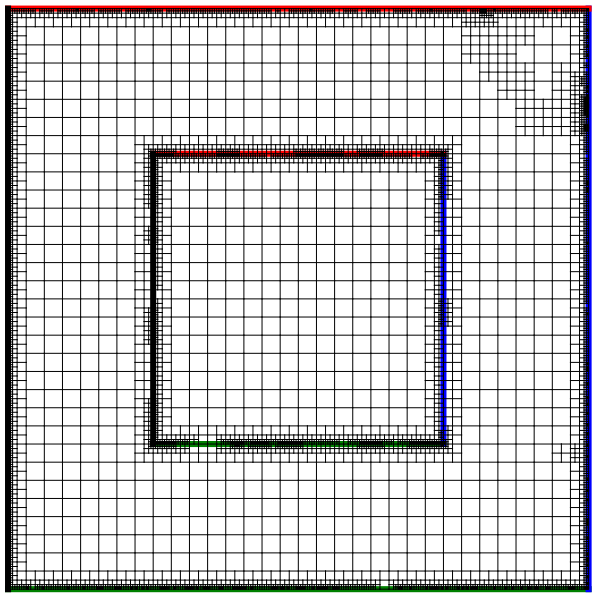


(a)

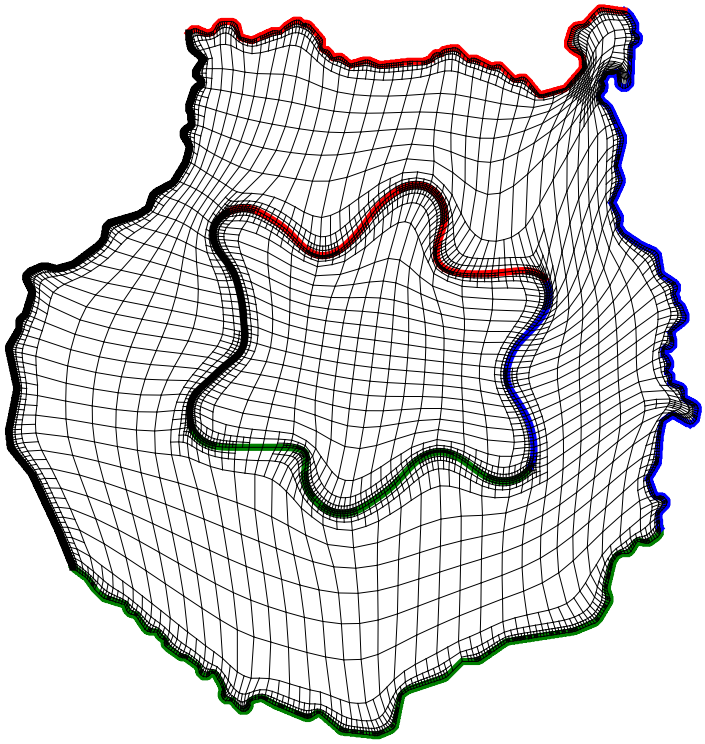


(b)

Figure 7.12. Embedded Test geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.

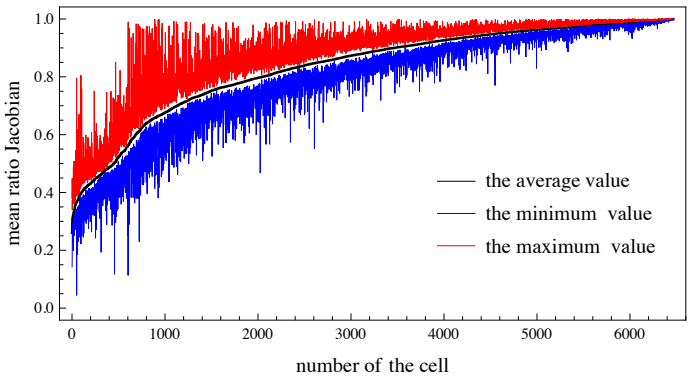


(a)

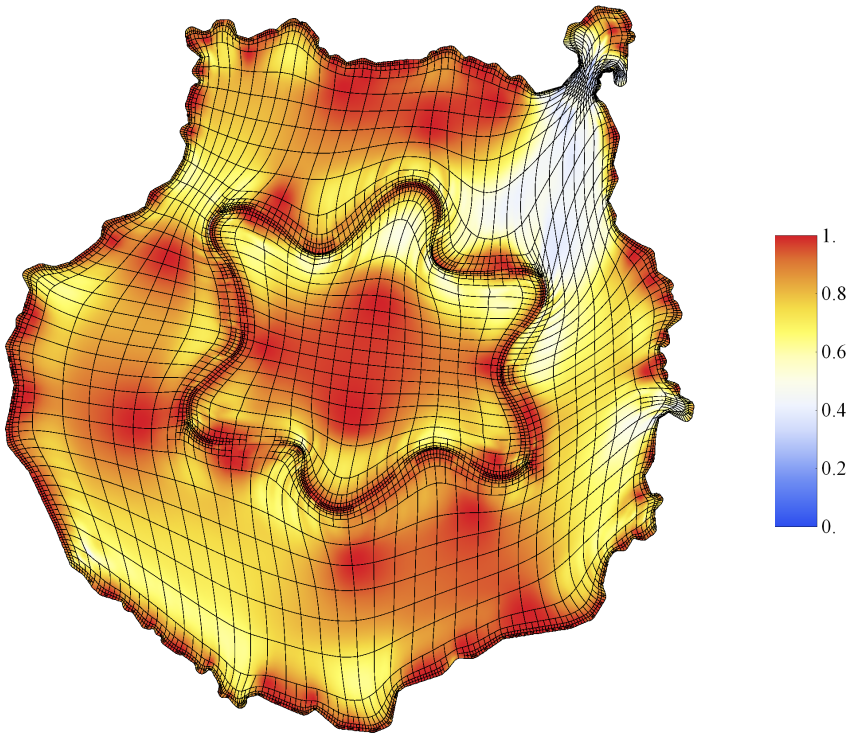


(b)

Figure 7.13. Embedded Complex geometry mesh. (a) Global parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometries.



(a)



(b)

Figure 7.14. Embedded Complex geometry T-spline parameterization. (a) Mean ratio Jacobian statistics. (b) T-spline representation and color map of the mean ratio Jacobian.

7.2 EP-spline parameterization of 2D and 3D geometries

In this section we test the generic parameterization strategy proposed in the chapter 6. We build EP-spline parameterizations for both, 2D and 3D geometries. As in previous section, the input consists of compatible patches that are mapped to the boundary of the parametric space. Some of the resulting geometry parameterizations are used in the next section for solving problems with Iseogemetric Analysis.

7.2.1 Seal geometry

Figures 7.15(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization (see input polylines marked in different colors). We have constructed an adapted T-mesh starting from an initial uniform 16×16 mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 2$ for the biggest cells of the local mesh. The resulting mesh has 1351 cells and the EP-spline representation was built with 1762 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.05. Figures 7.16(a) and (b) show statistics of the mean ratio Jacobian and the EP-spline representation.

7.2.2 Rabbit geometry

Figures 7.17(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform 16×16 mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 2$ for the biggest cells of the local mesh. The resulting mesh has 1942 cells and the EP-spline representation has 2514 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.23. Figures 7.18(a) and (b) show statistics of the mean ratio Jacobian and the EP-spline representation.

7.2.3 Cat geometry

Figures 7.19(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform 16×16 mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 2$ for the biggest cells of the local mesh. The resulting mesh has 1330 cells and the EP-spline representation has 1727 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.37. In next section, Helmholtz equation is solved over this geometry using Isogeometric Analysis. Figures 7.20(a) and (b) show statistics of the mean ratio Jacobian and the EP-spline representation.

7.2.4 USA geometry

Here a more complex example is presented. The geometry represents the contour of the USA map. A major number of refinements are necessary to adapt the T-mesh

to the geometry features. Figures 7.21(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform 16×16 mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 2$ for the biggest cells of the local mesh. In this case, three adaptive refinements were required to obtain a valid EP-spline parameterization. The resulting mesh has 3292 cells and 4059 control points were necesary to build the EP-spline parameterization. The minimal value of mean ratio Jacobian at quadrature points is 0.09. Figures 7.22(a) and (b) show statistics of the mean ratio Jacobian and the EP-spline representation.

7.2.5 Deformed cube geometry

Here we present the application of the strategy to a CAD model designed in Rhinoceros software. The model consist in a deformation of the faces of a cube. As input, we take the spline representation of each face. Figures 7.23(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform $8 \times 8 \times 8$ mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 8$ for the biggest cells of the local mesh. We use $w = 8$ to try to preserve the volume relations of the parametric elements. This imposes more orthogonality, so it is a good strategy for cube type geometries. The resulting mesh has 1121 cells and 2360 control points were necessary for the EP-spline representation. Figure 7.23(c) shows statistics of the mean ratio Jacobian and the EP-spline representation at a section and at whole physical domain is presented in Fig. 7.23(d) and (e). The minimal value of mean ratio Jacobian at quadrature points is 0.35. In next section, a Poisson problem is solved over this geometry using Isogeometric Analysis.

7.2.6 Analytic geometry

This geometry is defined by an analytic deformation of one face of the cube. Figures 7.24(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform $8 \times 8 \times 8$ mesh. The T-mesh optimization was carried out without weight for big cells, that is, $w = 1$ for all cells. The resulting mesh has 1940 cells and the EP-spline representation has 3145 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.33. Figures 7.24(c) show statistics of the mean ratio Jacobian and Fig. 7.24(d) and (e) illustrate the EP-spline representation at a section and at whole physical domain.

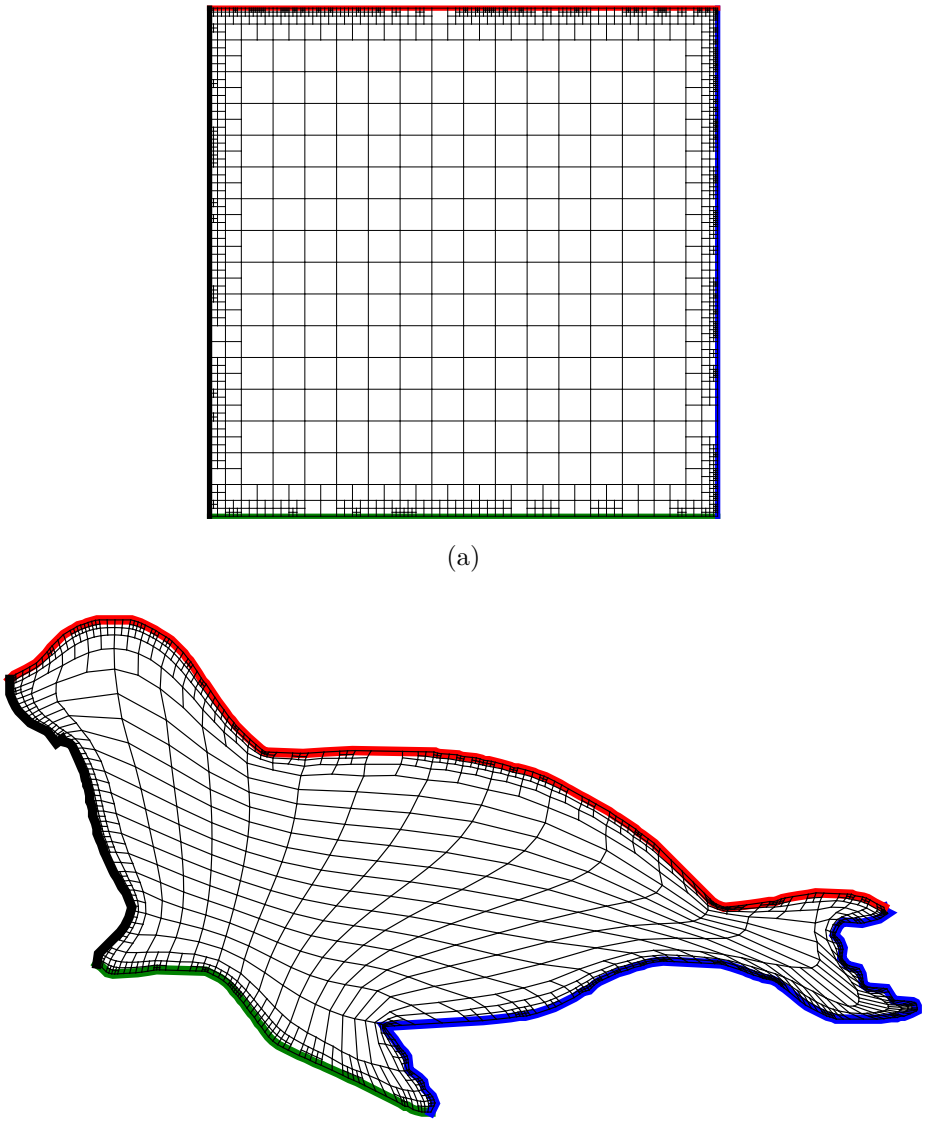
7.2.7 Sample orography

An important application of the proposed method is the parameterization of orography type geometries. We have used this type of geometries to solve problems of wind field adjustment where we need to mesh the air over a certain complex terrain, see [89, 90].

In this example we parameterize a simple orography described by a triangulation of the surface. Figures 7.25(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. For a correct visualization of the orography mesh we show the terrain surface at top and the air at bottom. We have constructed an adapted T-mesh starting from an initial uniform $8 \times 8 \times 8$ mesh. The T-mesh optimization was carried out without weight for big cells, that is, $w = 1$ for all cells. The resulting mesh has 2003 cells and the EP-spline representation has 3432 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.73. Figure 7.25(c) shows statistics of the mean ratio Jacobian. Figure 7.25(d) and (e) show the EP-spline representation and the color map of the mean ratio Jacobian at a section of the physical domain and at whole domain respectively.

7.2.8 La Palma orography

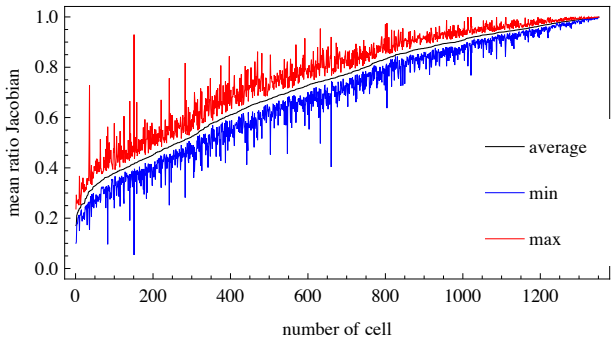
This is a more complex orography that corresponds to an extraction of La Palma island. The input is taken from an triangulation description of the boundary. Figures 7.26(a) and (b) show the parametric T-mesh and the physical T-mesh after optimization. We have constructed an adapted T-mesh starting from an initial uniform $16 \times 16 \times 16$ mesh. The T-mesh optimization was carried out with the weighted objective function, setting weight $w = 2$ for the biggest cells of the local mesh. The resulting mesh has 16619 cells and the EP-spline representation has 22191 control points. The minimal value of mean ratio Jacobian at quadrature points is 0.35. Figures 7.27(a) and (b) show mean ratio Jacobian statistics and the EP-spline representation at a section of the physical domain. Figures 7.28(a) and (b) illustrate the EP-spline surface of the orography and the EP-spline representation at the whole domain.



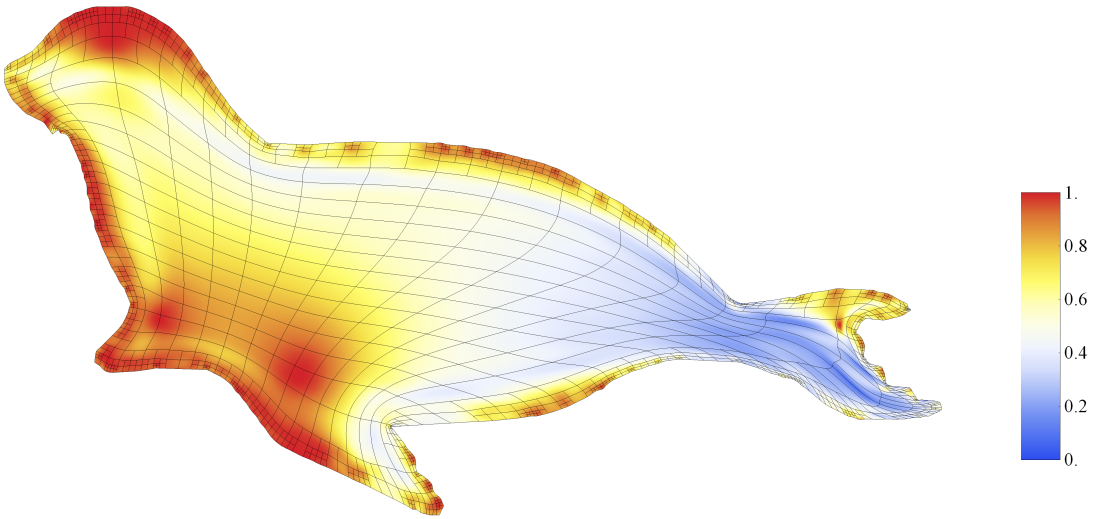
(a)

(b)

Figure 7.15. Seal geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

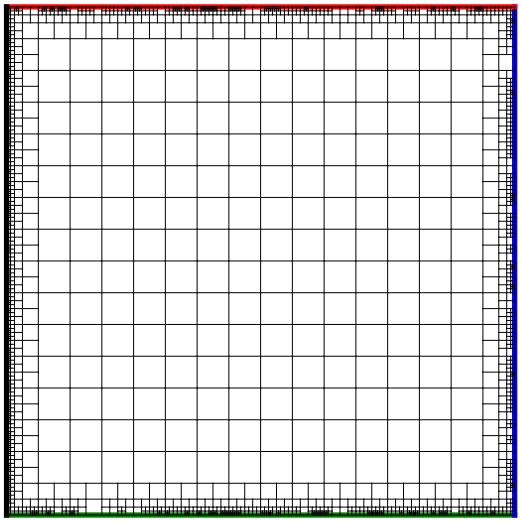


(a)

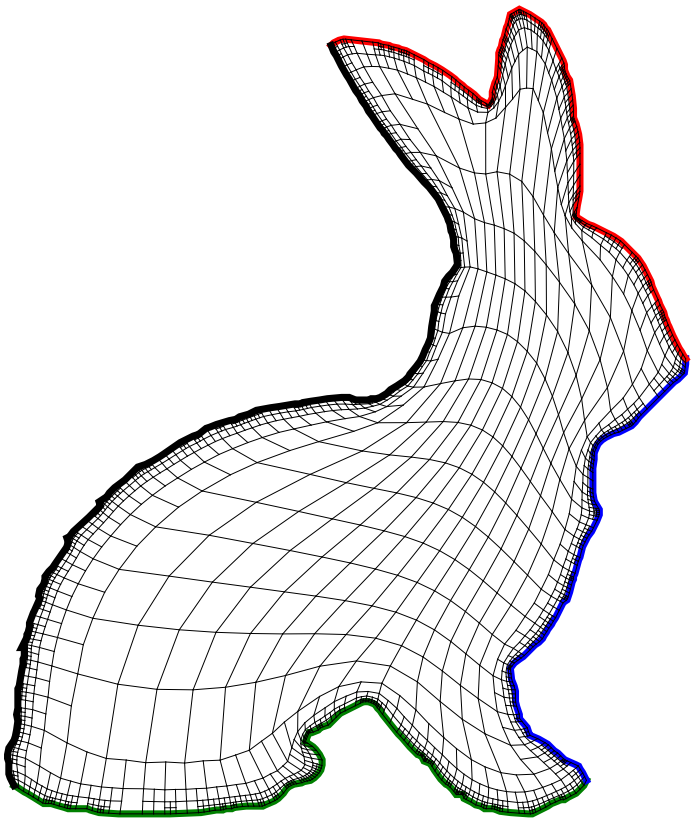


(b)

Figure 7.16. Seal geometry EP-spline parameterization. (a) Mean ratio Jacobian statistics. (b) EP-spline representation and color map of the mean ratio Jacobian.

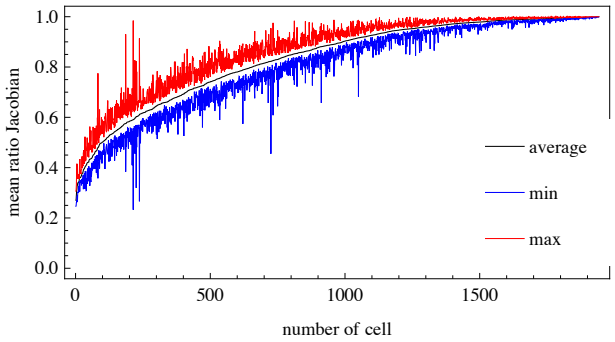


(a)

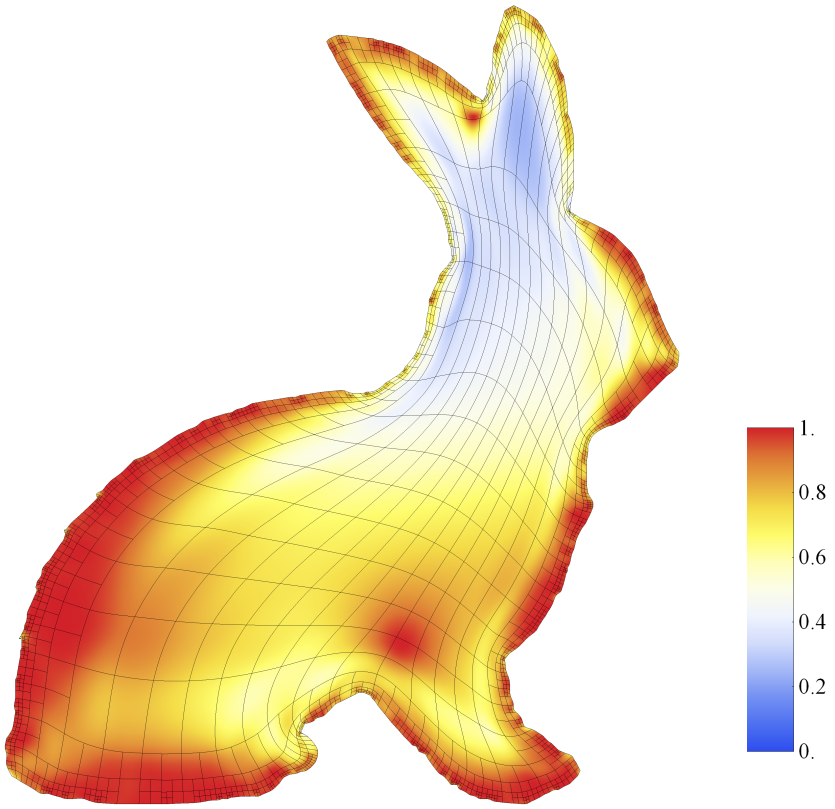


(b)

Figure 7.17. Rabbit geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

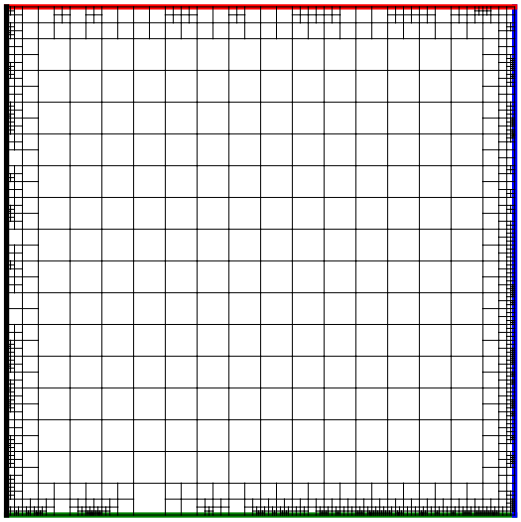


(a)

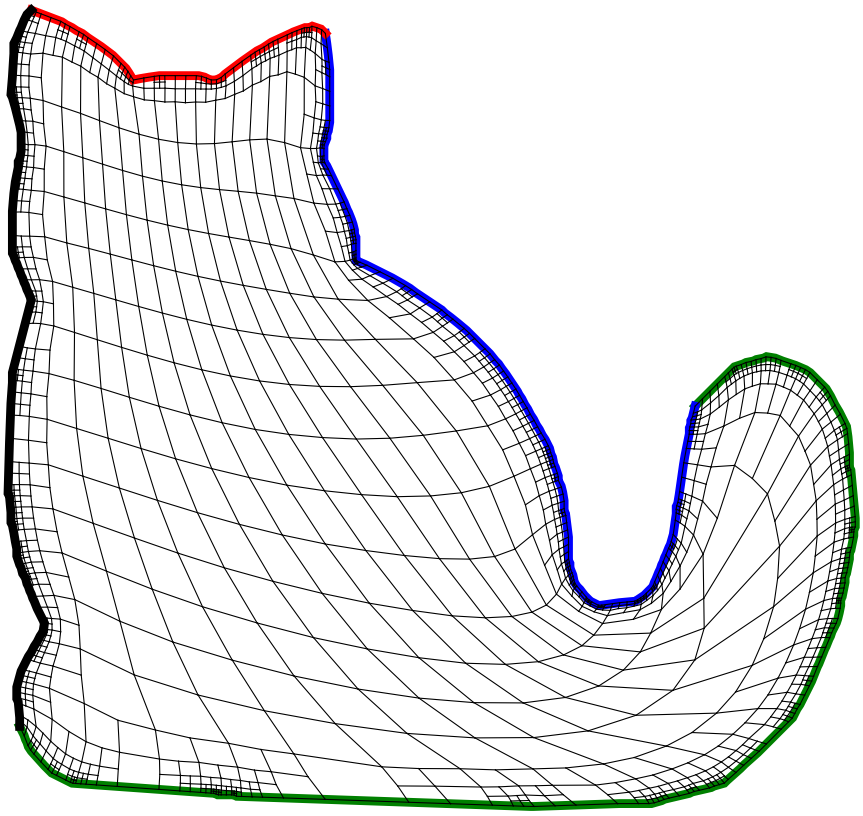


(b)

Figure 7.18. Rabbit geometry EP-spline parameterization. (a) Mean ratio Jacobian statistics. (b) EP-spline representation and color map of the mean ratio Jacobian.

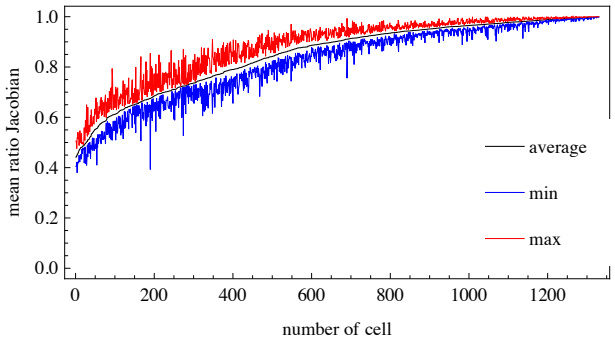


(a)

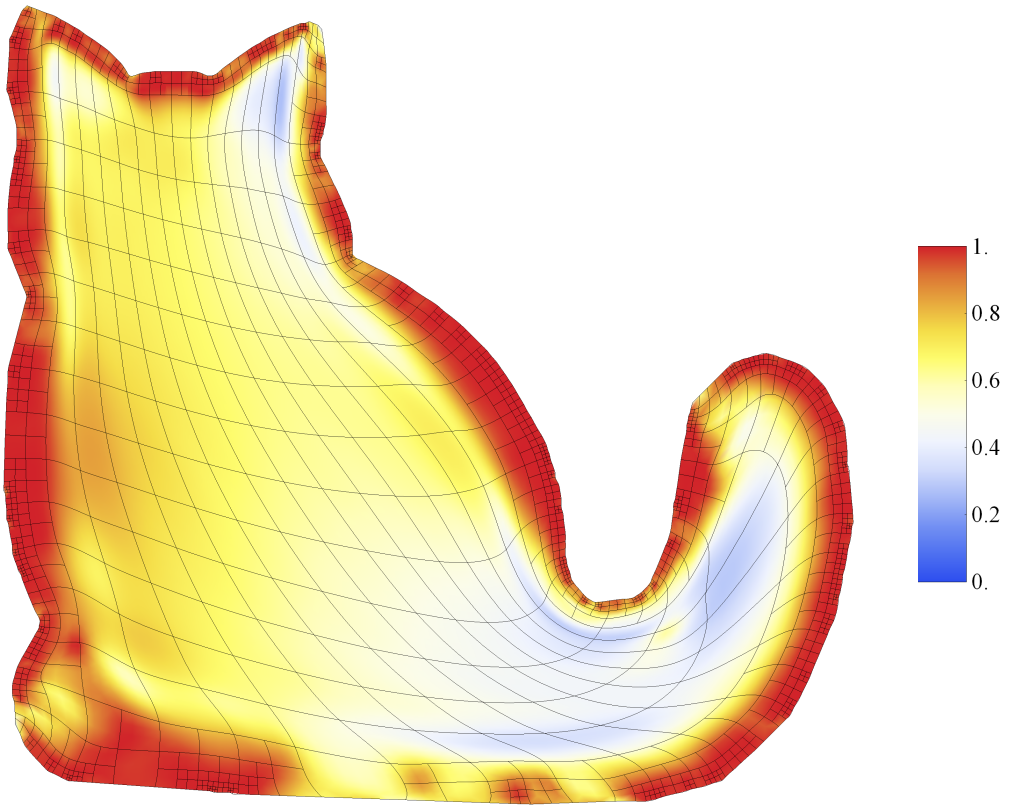


(b)

Figure 7.19. Cat geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

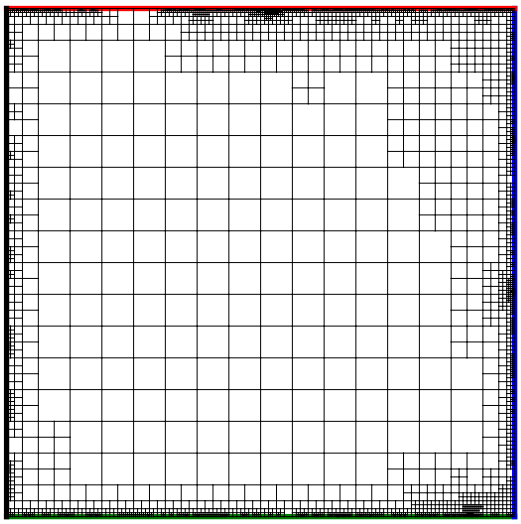


(a)

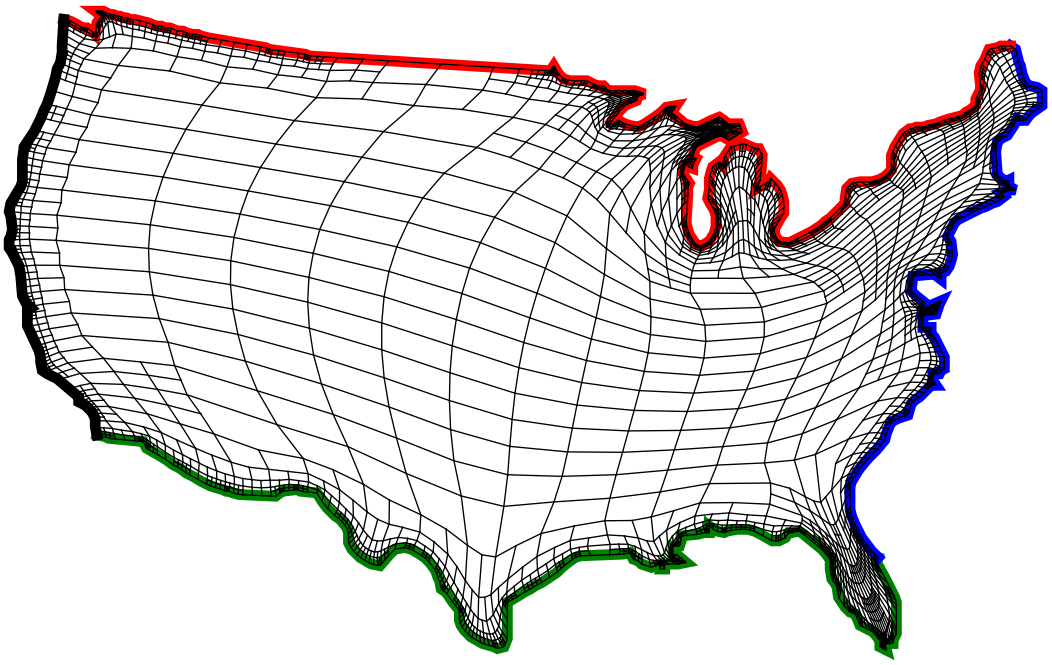


(b)

Figure 7.20. Cat geometry EP-spline parameterization. (a) Mean ratio Jacobian statistics. (b) EP-spline representation and color map of the mean ratio Jacobian.

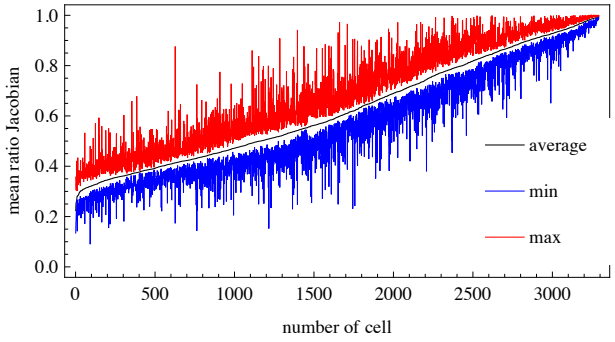


(a)

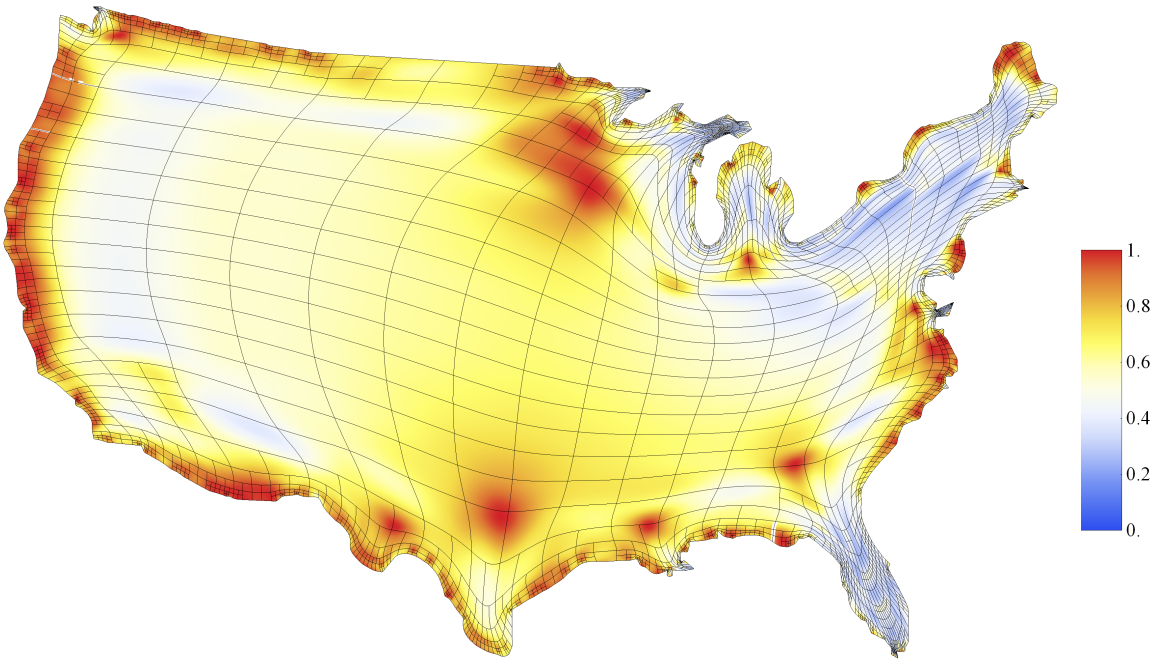


(b)

Figure 7.21. USA geometry mesh. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

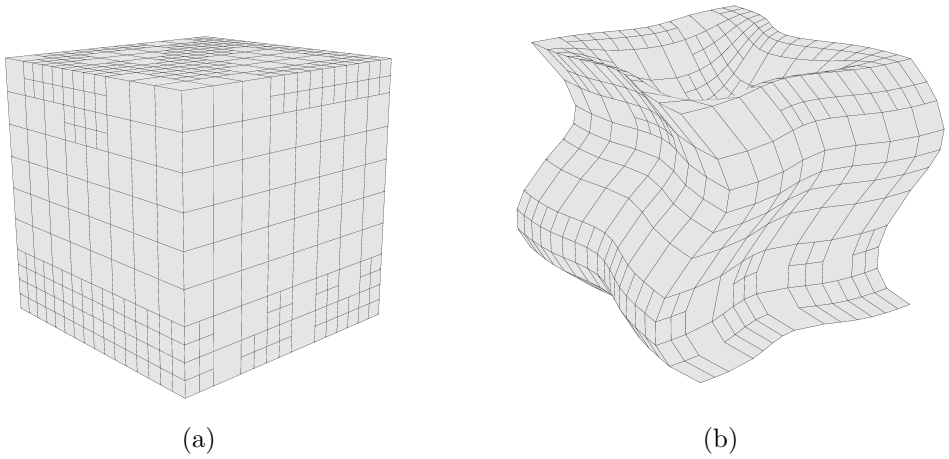


(a)

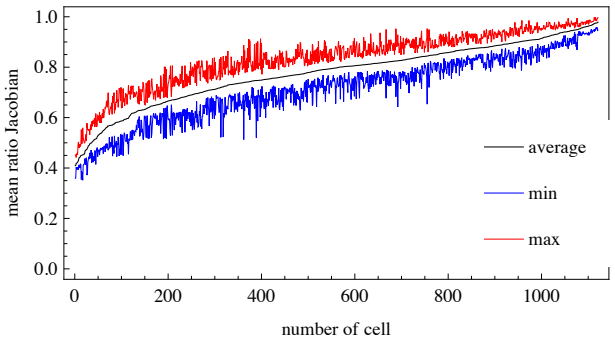


(b)

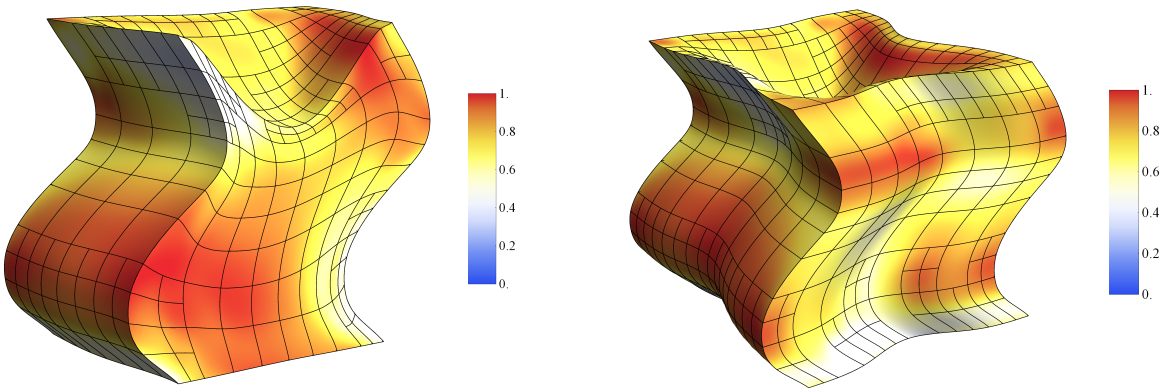
Figure 7.22. USA geometry EP-spline parameterization. (a) Mean ratio Jacobian statistics. (b) EP-spline representation and color map of the mean ratio Jacobian.



(a) (b)



(c)



(d) (e)

Figure 7.23. Deformed Cube geometry. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry. (c) Mean ratio Jacobian statistics. (d) EP-spline representation and color map of the mean ratio Jacobian at section $\xi = 1/2$. (e) Mean ratio Jacobian in the whole geometry.

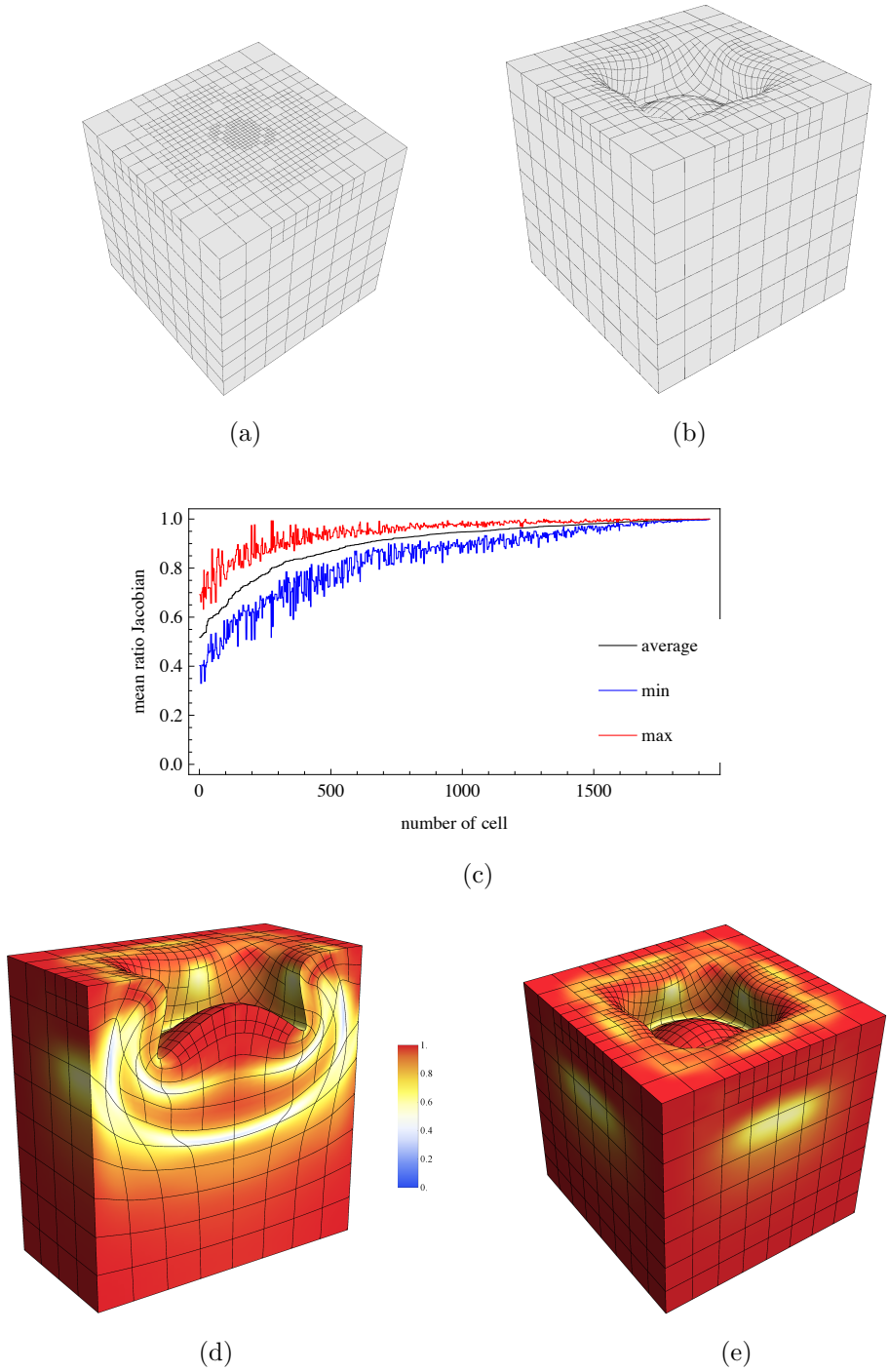
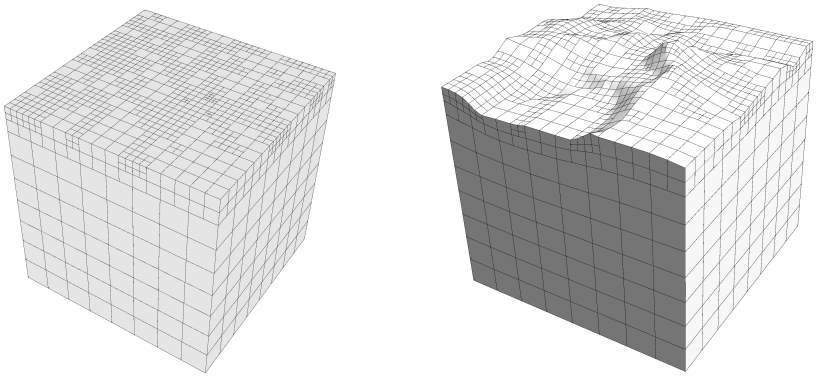
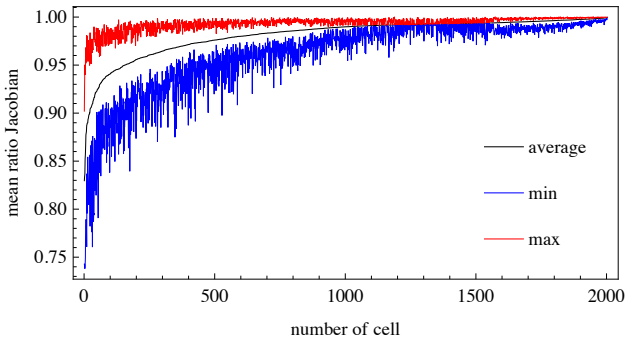


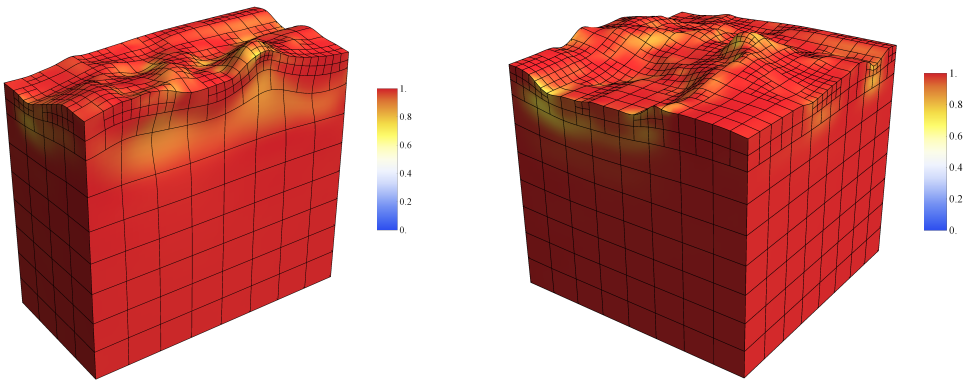
Figure 7.24. Analytic geometry. (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry. (c) Mean ratio Jacobian statistics. (d) EP-spline representation and color map of the mean ratio Jacobian at section $\xi = 1/2$. (e) Mean ratio Jacobian in the whole geometry.



(a) (b)

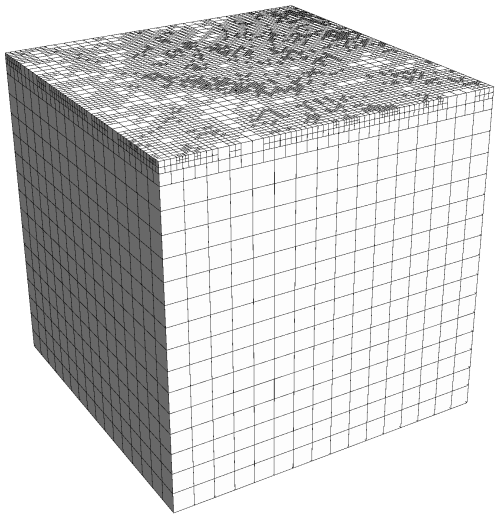


(c)

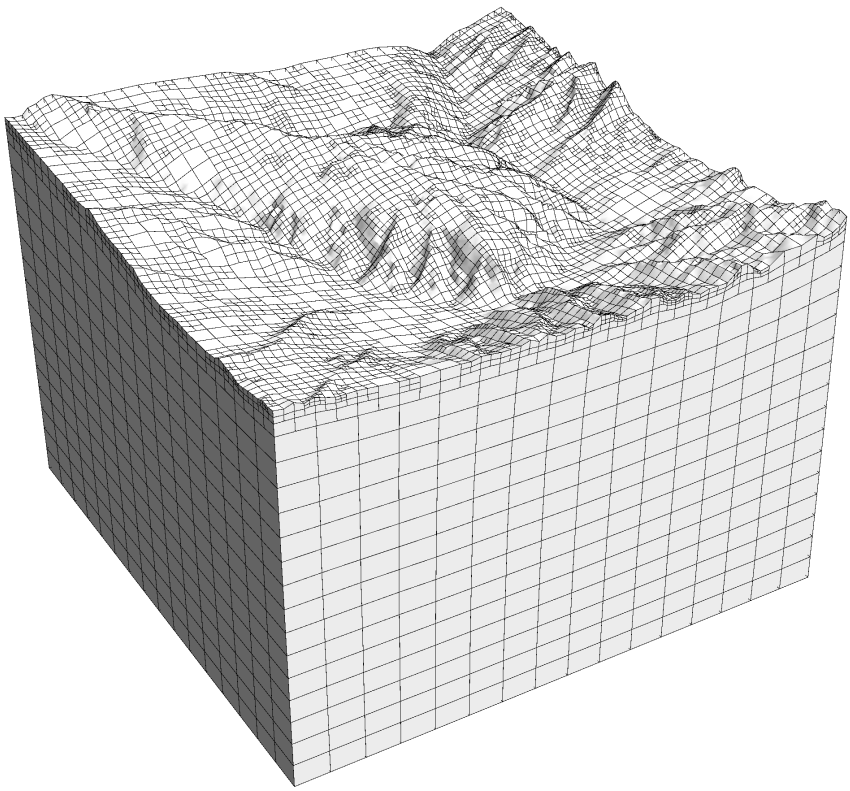


(d) (e)

Figure 7.25. Sample orography (terrain surface at top and the air at bottom). (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry. (c) Mean ratio Jacobian statistics. (d) EP-spline representation and color map of the mean ratio Jacobian at section $\xi = 1/2$. (e) Mean ratio Jacobian in the whole geometry.

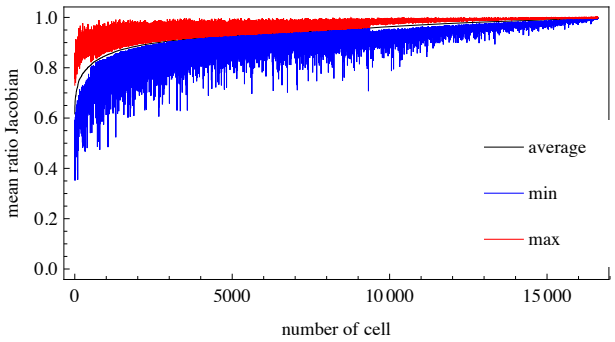


(a)

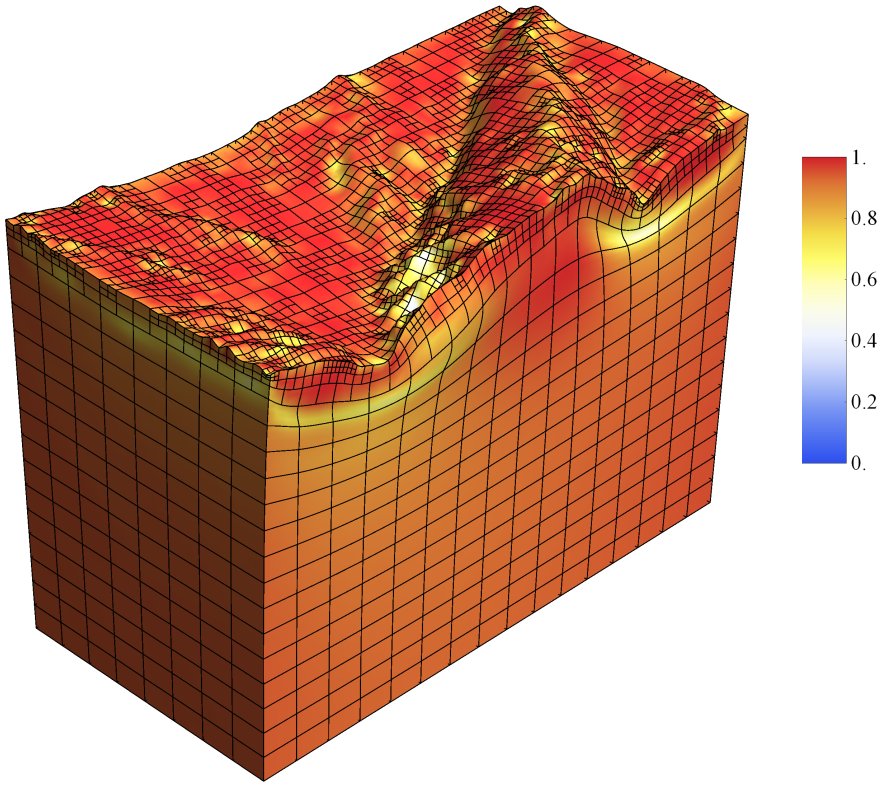


(b)

Figure 7.26. La Palma orography mesh (terrain surface at top and the air at bottom). (a) Parametric T-mesh. (b) Optimized physical T-mesh adapted to the boundary of the geometry.

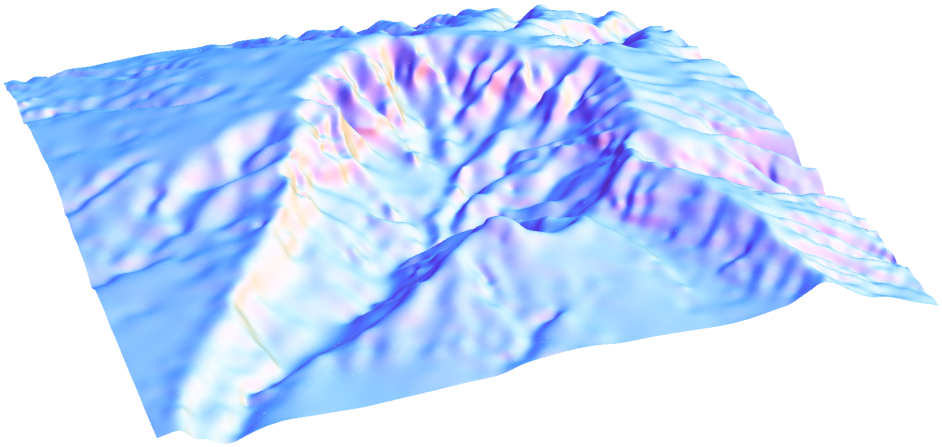


(a)

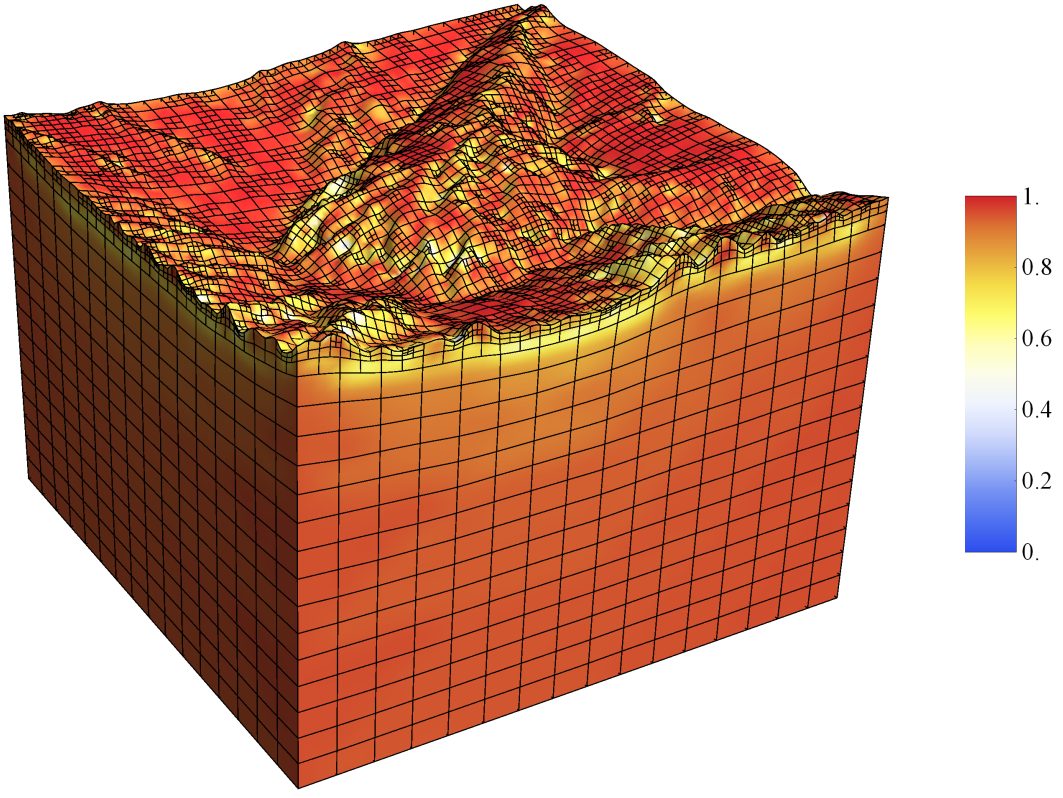


(b)

Figure 7.27. La Palma orography EP-spline parameterization (terrain surface at top and the air at bottom). (a) Mean ratio Jacobian statistics. (b) EP-spline representation and color map of the mean ratio Jacobian at section $\xi = 1/2$.

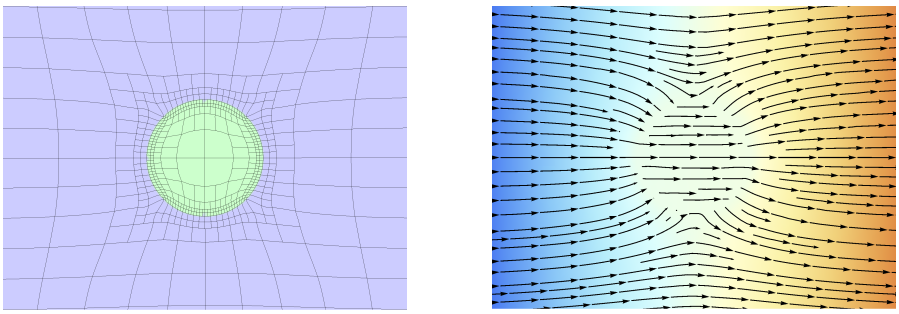


(a)



(b)

Figure 7.28. La Palma orography EP-spline parameterization. (a) EP-spline surface of the orography. (b) EP-spline representation and color map of the mean ratio Jacobian in the whole geometry (terrain surface at top and the air at bottom).



(a) (b)

Figure 7.29. IGA application for dielectric cylinder in a uniform electric field. (a) Zoom of the T-spline centred in the immersed cylinder. (b) The stream lines of the electric field around of the cylinder.

7.3 Isogeometric analysis applications

7.3.1 Dielectric cylinder in a uniform electric field

In this section we present an example of the resolution of the Poisson equation for a domain composed of two materials using isogeometric analysis with T-splines. Let us consider the next problem

$$\begin{aligned} -\nabla(k(\mathbf{x})\nabla u) &= f \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \partial\Omega. \end{aligned}$$

This problem is an infinity dielectric cylinder of radio b immersed in uniform electric field $E_0\mathrm{i}$. Actually, it is a bi-dimensional problem as there is no dependence with z coordinate. The domain Ω is a rectangle with a embedded circle of radius b in its center where the dielectric constants are

$$k(\mathbf{x}) = \begin{cases} \varepsilon_0\varepsilon_r & \text{if } \rho < b \\ \varepsilon_0 & \text{if } \rho \geq b \end{cases}$$

The analytic solution is given in cylindrical coordinates by

$$u_{\rho<b} = \frac{-2E_0\rho\cos\varphi}{(\varepsilon_r+1)}$$

$$u_{\rho\geq b} = E_0\cos\varphi\left(-\rho + \frac{b^2(\varepsilon_r-1)}{\rho(\varepsilon_r+1)}\right)$$

By calculating the electric field $\mathrm{E} = -\nabla u$ it can be seen that the electric field tends to initial electric field $E_0\mathrm{i}$ as ρ tends to infinity and it is constant inside the cylinder.

In order to obtain a numerical solution as accurate as possible, we impose the boundary Dirichlet conditions taking into account the exact solution.

The T-spline geometry of the embedded domains is shown in Fig. 7.29 (a) and the result of the numerical simulation with $b = 0.5$ and $\varepsilon_r = 20$ is presented in Fig. 7.29 (b). Note that the stream lines of the electric field are constant in the cylinder. Comparing with the analytic solution, we have measured a maximum error in the potential of 0.88%.

7.3.2 Helmholtz equation with variable frequency

Here we present an example of solving a Helmholtz equation in the Cat domain using isogeometric analysis with EP-splines

We have the following problem with Dirichlet boundary conditions

$$\begin{aligned} -\Delta u - \frac{1}{(\alpha + r)^4} u &= f \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \partial\Omega, \end{aligned}$$

where $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$. The exact solution of the problem is given by

$$u(r) = \sin\left(\frac{1}{\alpha + r}\right). \tag{7.1}$$

The function (7.1) is highly oscillatory near the point (x_0, y_0), and the number of oscillations is determined by the parameter α. We used $\alpha = 1/(4\pi)$. Besides, it has discontinuous gradient at the point (x_0, y_0).

We denote by $\widehat{\Omega} = [0,1]^2$ the parametric domain and by Ω the physical (computational) domain for our problem. Let T be a T-mesh of $\widehat{\Omega}$, and $\widehat{V}_T = \operatorname{span}\{\widehat{N}_i\}_{i \in I}$ is the finite dimensional space spanned by the spline blending functions defined over the parametric T-mesh T. We construct the parametric mapping $\mathbf{S} : \widehat{\Omega} \to \Omega$ as $\mathbf{S} = \sum_{i \in I} \mathbf{P}_i \widehat{N}_i$. Then, the discrete approximation space V_T in the physical domain is defined as follows:

$$V_T = \operatorname{span}\left\{ N_i \,:\, N_i = \widehat{N}_i \circ \mathbf{S}^{-1}, \text{for all } \widehat{N}_i,\ i \in I \right\}.$$

Test function space is denoted by $V_{0,T} = \operatorname{span}\{N_i\}_{i \in I_0}$. $V_{g_h,T}(\Omega)$ is the subspace of functions of V_T that are equal to g_h at the boundary, where g_h is an interpolant of g.

Discrete variational formulation consists in finding $u_h \in V_{g_h,T}$ such that

$$a(u_h, N_j) = F(N_j) \quad \forall N_j \in V_{0,T},$$

where

$$a(u,v) = \int_\Omega \left(\nabla u \cdot \nabla v + k(r)^2 u\, v \right) \mathrm{d}\Omega \quad \text{and} \quad F(v) = \int_\Omega f\, v \, \mathrm{d}\Omega,$$

being $k(r) = \dfrac{1}{(\alpha + r)^2}$.

Adaptive refinement is performed using the residual-based error estimator given by

$$\eta(\Omega_e)^2 = h^2 \| \left(f + \Delta u_h + k(r)^2 u_h \right) \|^2_{L^2(\Omega_e)},$$

where h is the diameter of the cell Ω_e. Marking strategy used for adaptive process: a cell Ω_e is marked to be refined if $\eta(\Omega_e) > \gamma \max_i \{\eta(\Omega_i)\}$, being $\gamma \in [0,1]$. Normally we use $\gamma = 0.5$.

The numerical solution of the problem and the mesh corresponding to the final refinement iteration are shown in Fig. 7.30(a)-(d). As expected, during adaptive refinement the error estimator has marked the singularity zone. The evolution of the exact error in L^2-norm and H^1-seminorm and its comparison with the uniform refinement are shown in Fig. 7.30(e). As can be observed, adaptive refinement gains the optimal rates of convergence.

7.3.3 3D Poisson problem

Now we consider a 3D Poisson problem with Dirichlet boundary condition on Deformed cube model:

$$\begin{aligned} -\triangle u &= f \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \partial\Omega. \end{aligned}$$

The problem is set up so that the exact solution is a function with steep wave front given by

$$u(r) = \arctan(\alpha(r - r_0)),$$

where $r = \sqrt{(x-x_c)^2 + (y-y_c)^2 + (z-z_c)^2}$, the parameter α determines the steepness of the wave front and r_0 is its location. In this example $\alpha = 200$ and $r_0 = 0.6$. The center of the wave front (x_c, y_c, z_c) is situated outside our computational domain, so the function is smooth in Ω. Adaptive refinement with EP-splines is performed using residual-based error estimator. The numerical solution of the problem and the mesh corresponding to the final refinement iteration is shown in Fig. 7.31. As expected, the error estimator has marked for refinement the zone of the wave front. The evolution of the exact error in L^2-norm and H^1-seminorm and its comparison with uniform refinement are shown in Fig. 7.31(c).

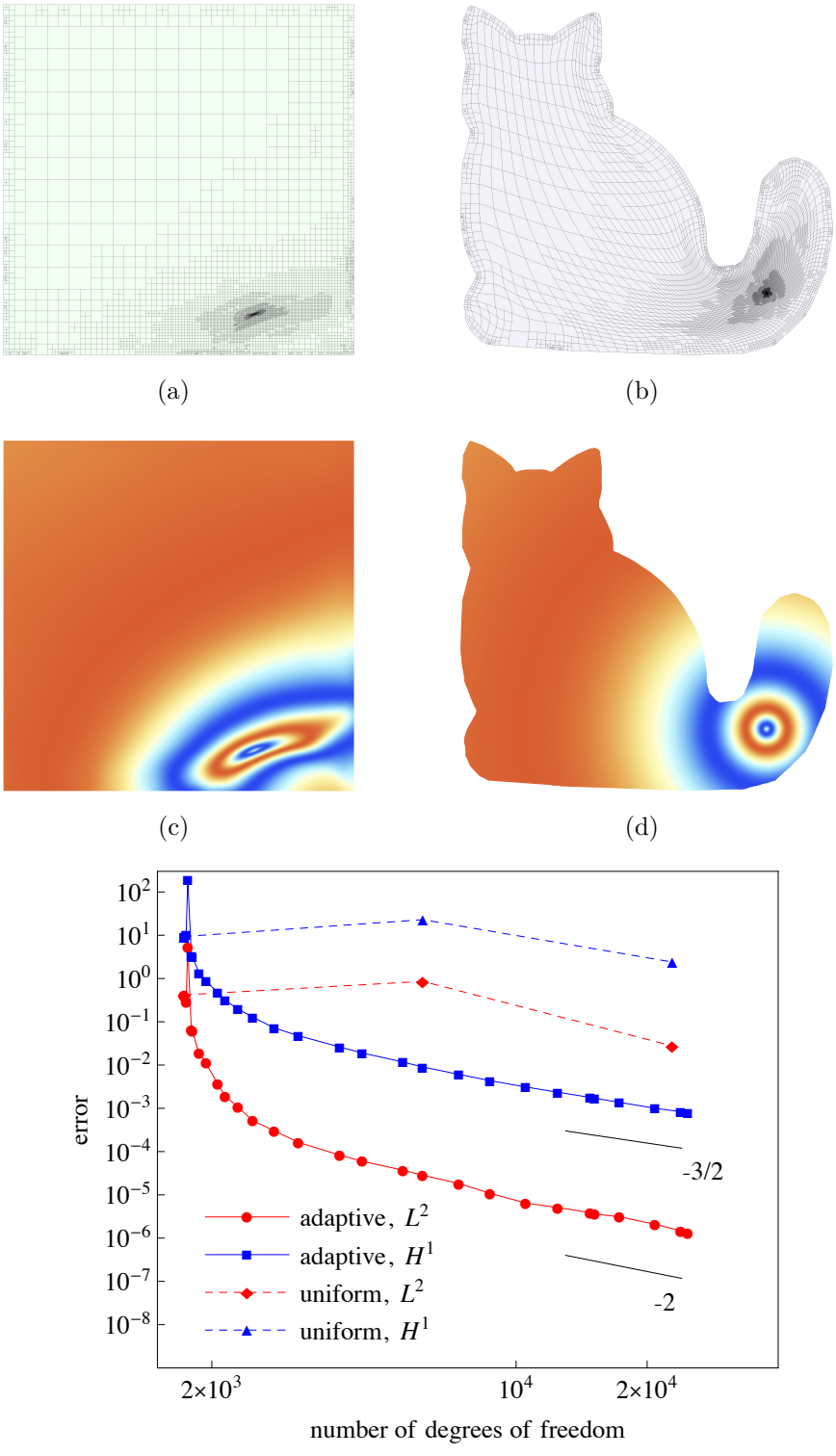


Figure 7.30. IGA application for Helmholtz equation in the Cat geometry. (a) Final refinement in the parametric domain. (b) Final refinement in the physical domain. (c) Numerical solution in the parametric domain. (d) Numerical solution in the physical domain. (e) Error convergence for the uniform and adaptive refinements.

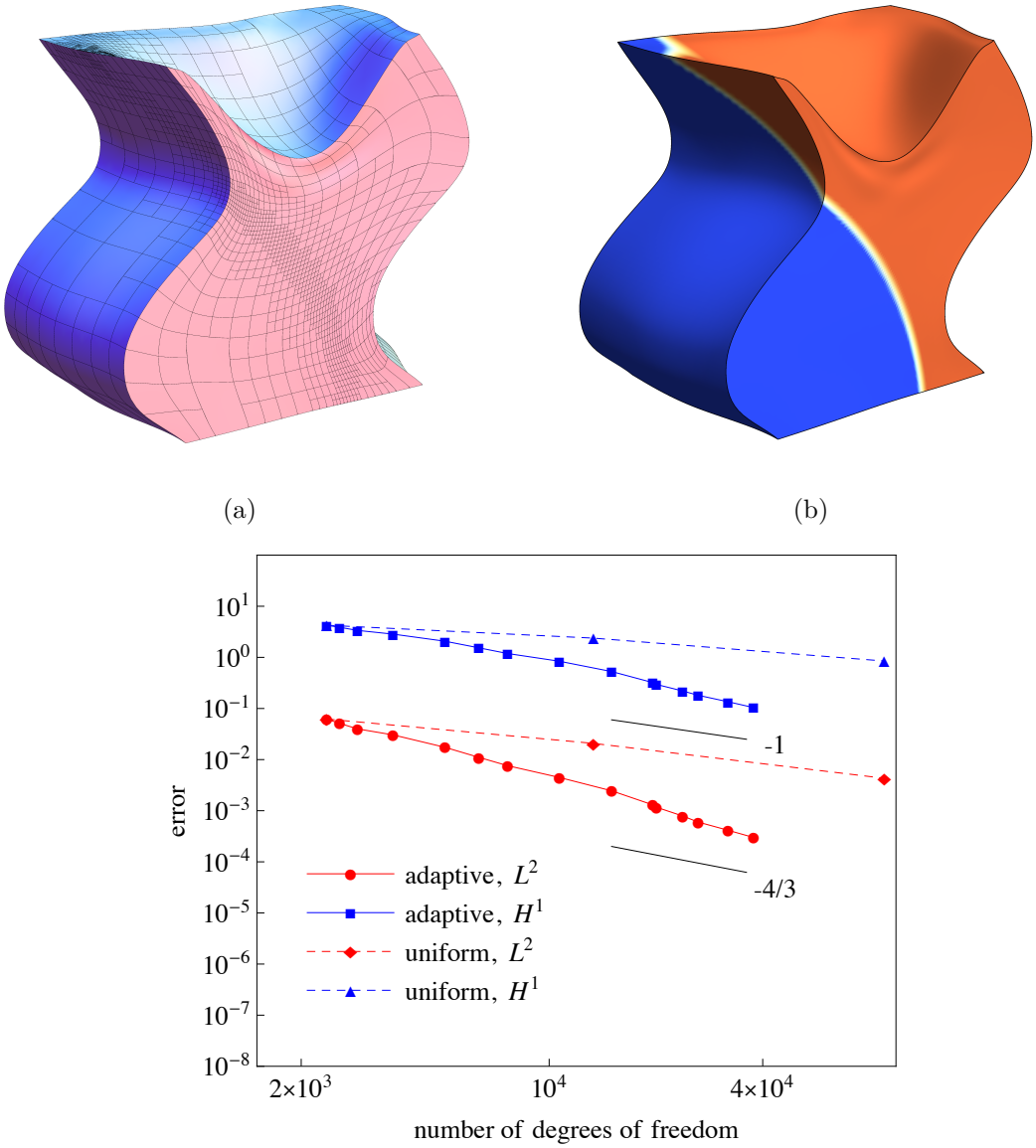


Figure 7.31. IGA application for 3D Poisson problem in Deformed Cube geometry. (a) Final refinement in the physical domain. (b) Numerical solution in the physical domain. (c) Error convergence for the uniform and adaptive refinements.

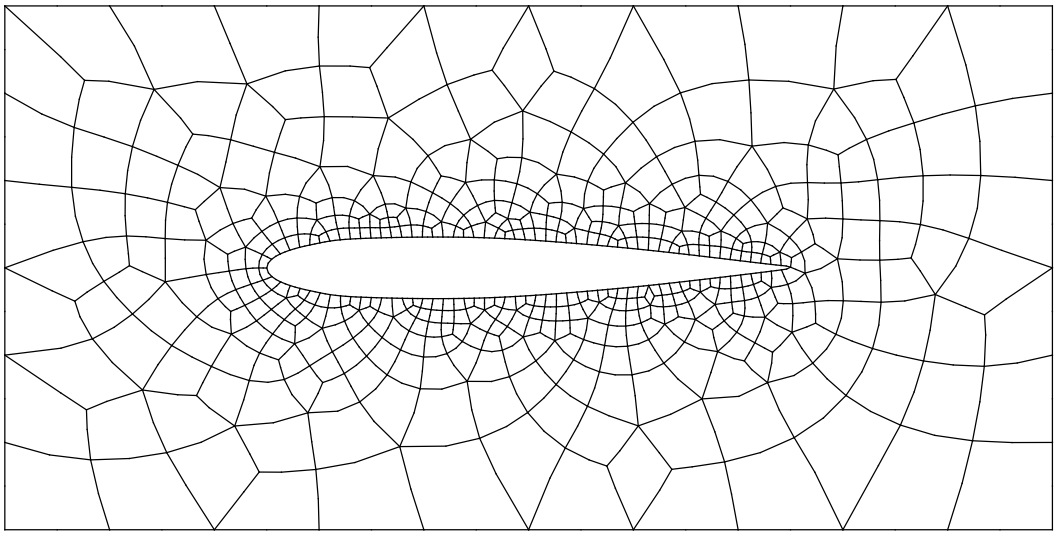
7.4 Unstructured quadrilateral and hexahedral mesh optimization

It is worth noting that T-mesh optimization algorithm described in chapter 6 also can be used for FEM quadrilateral and hexahedral mesh optimization, in particular, for second order meshes. The difference with respect to a T-mesh optimization is that we don't need to complement some elements with virtual nodes since all 9 (27 for 3D) nodes are given for the second order elements, and all of them should be relocated in order to improve the quality of the quadratic mapping of each cell.

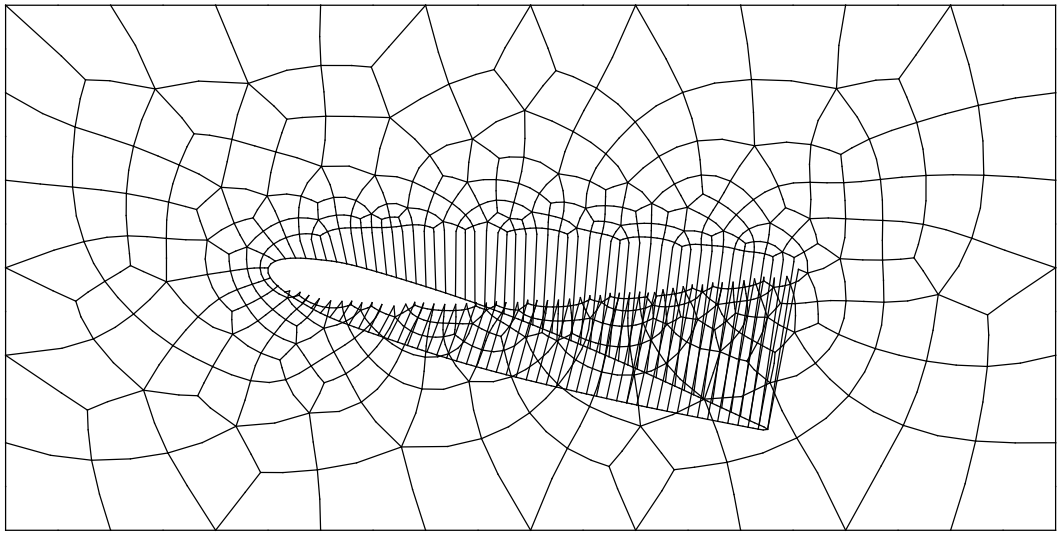
Frequently, for example in shape optimization problems, it is necessary to change the boundary of the computational domain and solve the problem repeatedly over the modified domain. Remeshing the geometry at each step can be very costly and can obstruct the automation of the process. Besides, a new mesh generation can introduce a different number of degrees of freedom for the problem. So, it is desirable to be able to adapt the initial mesh to the new boundary without changing its topology.

Here, we show the efficacy of the optimization algorithm in the following scenario. Figure 7.32(a) shows an unstructured second order quadrilateral mesh generated by Gmsh grid generator [91]. The computational domain represents a simplified model of the airfoil where its surroundings fluid is meshed for applying Finite Element Method. The obtained mesh has 470 elements. Then we change the boundary of the domain by rotating the airfoil $\pi/10$ degrees. New positions of the boundary nodes produce a tangled mesh, see Fig. 7.32(b). Our optimization algorithm is applied to relocate the interior nodes. The resulting optimized mesh adapted to the new boundary is shown in Fig. 7.33(b). Note that this mesh has the same topology as the initial one. Thus, the problem can be solved again over the new domain without remeshing the geometry. Figure 7.33(a) shows the comparison of the mesh elements quality after optimization and after Laplacian relocation, which leaves 213 invalid elements. We compute the element quality as the quadrature of the pointwise quality measure of the local quadratic mapping for each cell.

Similar example was performed for a hexahedral mesh of a wing. 2D quadrilateral mesh was extruded with 25 layers forming a hexahedral mesh with 11750 elements, see Fig. 7.34(a). Then, an end of the airfoil is rotated $\pi/20$ degrees. The nodes over the airfoil surface in the interior layers are progressively rotated, obtaining a 3D twisted wing. These rotations produce a tangled quadrilateral mesh of the boundary surface and a tangled hexahedral elements in the interior, see Fig. 7.34(b). First, the boundary quadrilateral elements are optimized, and then, the hexahedral mesh is optimized, see Fig. 7.35(b). Figure 7.35(a) compares elements quality after optimization and after Laplacian relocation, which produced 5136 invalid elements.

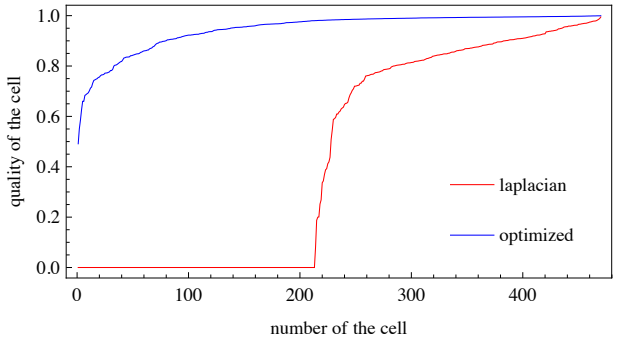


(a)

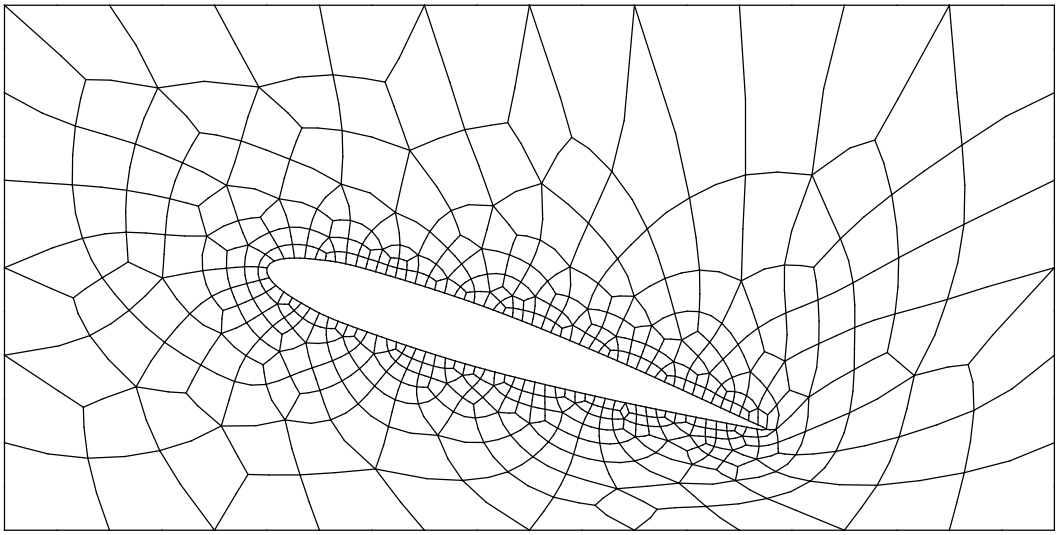


(b)

Figure 7.32. Unstructured second order quadrilateral mesh optimization. (a) Initial mesh. (b) Tangled mesh after rotating $\pi/10$ degrees the contour of the geometry.

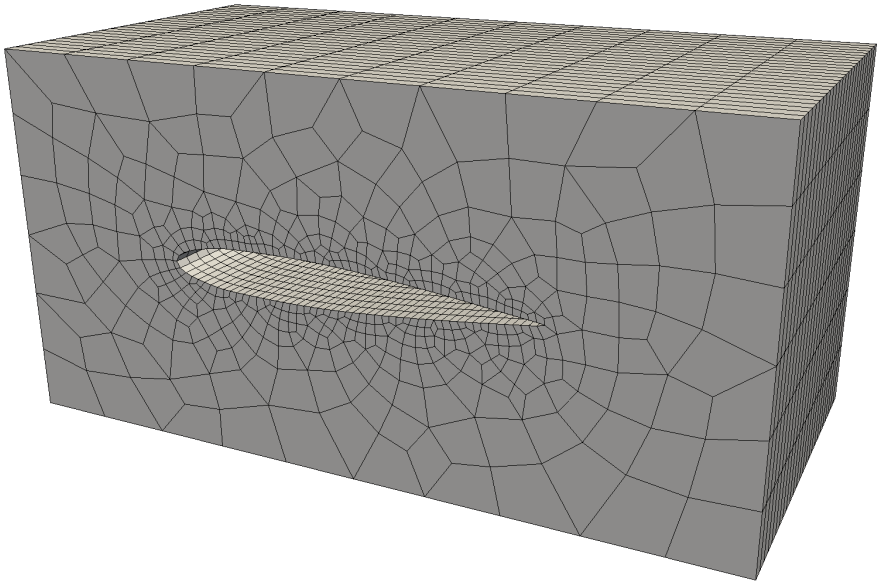


(a)

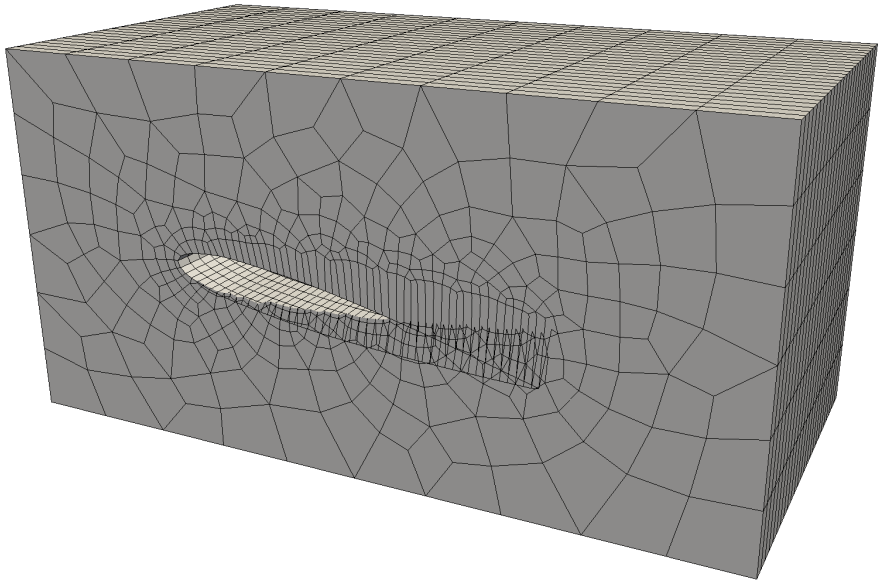


(b)

Figure 7.33. Unstructured second order quadrilateral mesh optimization. (a) Sorted qualities comparison after Laplacian relocation and optimization. (b) Resulting mesh after applying optimization procedure.

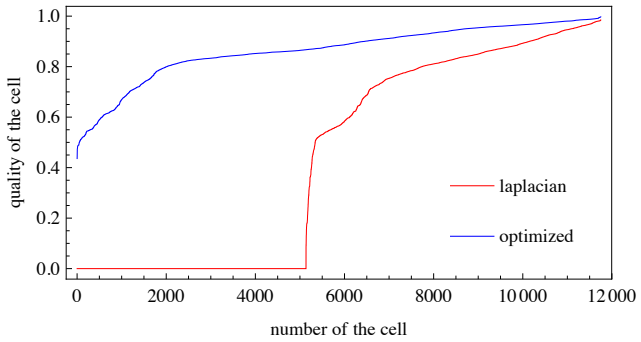


(a)

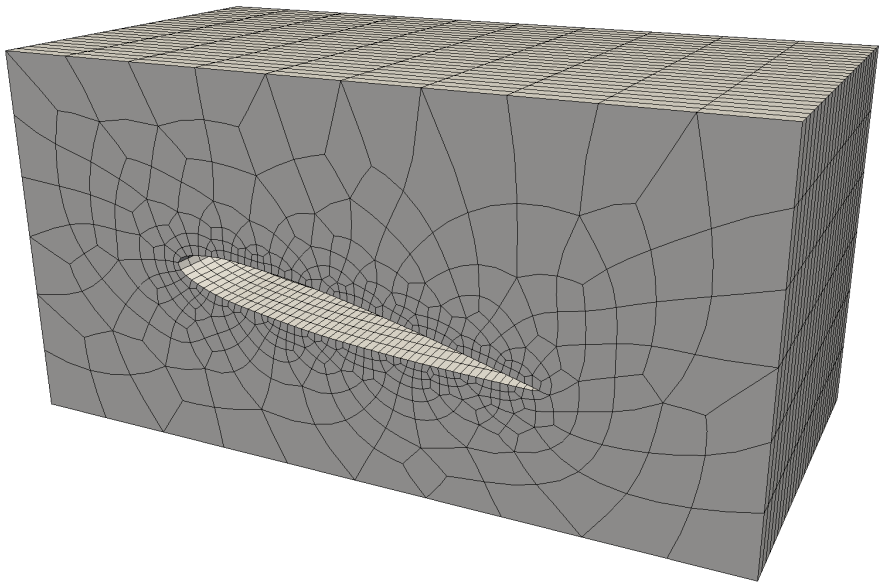


(b)

Figure 7.34. Unstructured second order hexahedral mesh optimization. (a) Initial mesh. (b) Tangled mesh after twisting the contour of the wing.



(a)



(b)

Figure 7.35. Unstructured second order hexahedral mesh optimization. (a) Sorted qualities comparison after Laplacian relocation and optimization. (b) Resulting mesh after applying optimization procedure.

8. Conclusions and future works

8.1 Summary and conclusions

In this thesis we have proposed a strategy for obtaining high quality spline parameterization of 2D and 3D geometries for their use in Isogeometric Analysis. The method only demands a boundary representation of the geometry as input data and constructs automatically a spline transformation between the physical and the parametric domains. The key of the method lies in defining an isomorphic transformation between the parametric and physical T-mesh finding the optimal position of the interior nodes by applying a T-mesh untangling and smoothing procedure. Initially, the method was proposed for T-spline parameterization of 2D geometries, where the T-mesh optimization is carried out by a decomposition of the T-mesh elements into triangles. Then, we extend the method to both 2D and 3D geometries. In this case, we propose another approach for T-mesh optimization based on a pointwise distortion measure instead of the usual simplex decomposition.

Spline representation of the geometry is calculated by imposing interpolation conditions using the data provided by one-to-one correspondence between the meshes of the parametric and physical domains. To asses the quality of the parametric mapping, we evaluate its mean ratio Jacobian. Thereby, we detect the areas with low quality and perform an adaptive refinement in order to increase the degree of freedom in the areas with high distortion. This strategy allows to obtain a parameterization suitable for analysis with no negative Jacobian, even for complex geometries.

For application of IGA it is essential to have an effective method to obtain volumetric parameterization of the geometry from the representation of its boundary. This is still an open problem in the Isogeometric Analysis context. Our goal was to develop an alternative method for constructing spline parameterizations based on the ideas of the Meccano method.

This thesis includes the following main parts:

- description of a method to construct a T-spline parameterization of 2D geometries for the application of isogeometric analysis;
- study of several strategies, based on different objective functions, for optimization of hexahedral meshes;
- description of the extended method to obtain high quality spline parameterization of both, 2D and 3D geometries, where the T-mesh optimization is based on a punctual distortion measure of the elements instead of a simplex decomposition;
- testing of the parameterization methods with several geometries, and different problems are solved using IGA with some of the parameterized geometries.

The effectiveness of the method was tested with different geometries. For all of them, the strategy obtained a valid parameterization suitable for Isogeometric Analysis. For some geometries, adaptive refinements were necessary to guarantee positive

Jacobian at all quadrature points. Different type of problems was resolved with Isogemetric Analysis using some of the parameterized geometries. Moreover, another application of the optimization algorithm is exposed: second order FEM quadrilateral and hexahedral mesh optimization.

Besides, a summary of the strategy to construct polynomial spline spaces have been exposed in this thesis. The strategy allows to define easily cubic spline spaces with nice properties for analysis over a given strongly balanced quadtree/octree T-mesh. Theses basis were used in our general method for the parameterization of 2D and 3D geometries.

8.2 Future works

Regarding the proposed parameterization method, the following future works should be mentioned:

- All geometries presented in this work have been parameterized with the unit square/cube. However, to facilitate the parameterization of very complex geometries it is essential to work with more complex parametric domains that fit better the input geometry. This would allow to reduce the high distortion that appears when a geometry very different from a square/cube is parameterized. For that, it is necessary to elaborate a procedure to construct spline spaces over this type of parametric domains, trying to conserve the regularity of the basis functions.

- A similar problem appears with geometries with genus bigger than zero. It is necessary to develop a strategy to define spline parameterizations for this type of geometries.

- Globally smooth 3D parametrization entails some important limitations due to the tensor-product nature of spline functions. First, for complex domains it is necessary to subdivide the object and perform several-patches parameterization with loss of regularity in some zone of the merging. Second, for the given boundary surface of a solid it is not trivial to find optimal partition of this surface into six patches corresponding to six faces of the parametric cube. Besides, to guarantee regular parameterization (positive Jacobian), the boundary patches must merge in a convex region, which can be impossible for some geometries. It is essential to design an effective strategy to overcome all these issues.

In a more general context of the Isogeometric Analysis paradigm, we consider interesting to explore the following questions:

- IGA exhibits some advantages and a superior accuracy per degree of freedom over the Finite Element Method for some problems. However, as for any other technique, there are cases where, probably, it is not the best choice. It would

be desirable to study and classify the problems where Isogeometric Analysis can offer a considerable advantages and the problems where it is not so significant, specially taking into account the efficiency of both methods.

- Also, we would like to explore the possibility to construct spline spaces over triangulations of computational domain. Complex domain parameterization using structured hexahedral meshes has considerable limitations. Splines over triangulation can offer more flexibility compared with rigid nature of tensor product splines. However, it is not a trivial task. Currently available options presents a high complexity. There is a necessity of a simple and easy-to-use scheme for constructing globally smooth spline functions over triangulations, able to compete with tensor product splines.

Bibliography

[1] L. Piegl, W. Tiller, The NURBS book, Springer, New York, 1997.

[2] G. Farin, Curves and Surfaces for CAGD: a Practical Guide, 5th Edition, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[3] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Meth. Appl. Mech. Eng. 194 (2005) 4135–4195.

[4] J. Cottrell, A. Reali, Y. Bazilevs, T. Hughes, Isogeometric analysis of structural vibrations, Comput. Meth. Appl. Mech. Eng. 195, Issues 41 43 (2006) 5257–5296.

[5] H. Gómez, V. Calo, Y. Bazilevs, T. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, Comput. Meth. Appl. Mech. Eng. 197, Issues 49 50 (2008) 4333–4352.

[6] H. Gómez, T. J. Hughes, X. Nogueira, V. M. Calo, Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations, Comput. Meth. Appl. Mech. Eng. 199, Issues 25 28 (2010) 1828–1840.

[7] J. Liu, L. Dede, J. A. Evans, M. J. Borden, T. Hughes, Isogeometric analysis of the advective Cahn–Hilliard equation: Spinodal decomposition under shear flow, Journal of Computational Physics 242 (2013) 321–350.

[8] M. Kästner, P. Metsch, R. de Borst, Isogeometric analysis of the Cahn–Hilliard equation – a convergence study, Journal of Computational Physics 305 (2016) 360–371.

[9] F. Auricchio, L. B. D. Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences Vol. 20, No. 11 (2010) 2075–2107.

[10] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, T. J. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Meth. Appl. Mech. Eng. 267 (2013) 170–232.

[11] H. Casquero, L. Liu, Y. Zhang, A. Reali, H. Gómez, Isogeometric collocation using analysis-suitable T-splines of arbitrary degree, Comput. Meth. Appl. Mech. Eng. 301 (2016) 164–186.

[12] C. Anitescu, Y. Jia, Y. J. Zhang, T. Rabczuk, An isogeometric collocation method using superconvergent points, Comput. Meth. Appl. Mech. Eng. 284 (2015) 1073–1097.

[13] Y. Bazilevs, L. Beirão Da Veiga, J. Cottrell, T. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, Mathematical Models and Methods in Applied Sciences 16 (7) (2006) 1031–1090.

[14] L. Beirão da Veiga, A. Buffa, R. J., G. Sangalli, Some estimates for h-p-k-refinement in Isogeometric Analysis, Numer. Math. 118 (2) (2011) 271–305.

[15] A. Tagliabue, L. Dedé, A. Quarteroni, Isogeometric Analysis and error estimates for high order partial differential equations in fluid dynamics, Computers and Fluids 102 (2014) 277–303.

[16] T. J. R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 199, 5-8 (2010) 301–313.

[17] F. Auricchio, F. Calabró, T. J. R. Hughes, A. Reali, G. Sangalli, A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 249-252 (2012) 15–27.

[18] N. O. Collier, L. Dalcín, D. Pardo, V. M. Calo, The cost of continuity: performance of iterative solvers on isogeometric finite elements, SIAM Journal on Scientific Computing 35(2) (2012) A767–A784.

[19] N. O. Collier, D. Pardo, L. Dalcín, M. Paszynski, V. M. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers, Comput. Meth. Appl. Mech. Eng. 213-216 (2013) 353–361.

[20] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, T. J. R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Meth. Appl. Mech. Eng. 199 (2010) 357–373.

[21] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric analysis: Toward integration of CAD and FEA, 1st Edition, Wiley Publishing, 2009.

[22] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parametrization of computational domain in isogeometric analysis: Methods and comparison, Comput. Meth. Appl. Mech. Eng. 200 (2011) 2021–2031.

[23] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis, Comput. Aided Des. 45 (4) (2013) 812–821.

[24] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Variational harmonic method for parameterization of computational domain in 2D isogeometric analysis, in: 12th International Conference on Computer-Aided Design and Computer Graphics, IEEE, Jinan, 2011, pp. 223–228.

[25] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method, Journal of Computational Physics 252 (2013) 275–289.

[26] X. Li, X. Guo, H. Wang, Y. He, X. Gu, H. Qin, Harmonic volumetric mapping for solid modeling applications, in: Proc. of ACM Solid and Physical Modeling Symposium, Association for Computing Machinery, Inc., 2007, pp. 109–120.

[27] T. Martin, E. Cohen, R. Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions, Comput. Aid. Geom. Design 26 (2009) 648–664.

[28] Y. Zhang, W. Wang, T. J. R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Comput. Meth. Appl. Mech. Eng. 249-252 (2012) 185–197.

[29] W. Wang, Y. Zhang, L. Liu, T. J. R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, Computer-Aided Design 45 (2013) 351–360.

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

[31] M. A. Scott, X. Li, T. W. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T-splines, Comput. Meth. Appl. Mech. Eng. 213-216 (2012) 206–222.

[32] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, M. A. Scott, On linear independence of T-spline blending functions, Comput. Aid. Geom. Design 29 (2012) 63–76.

[33] D. R. Forsey, R. H. Bartels, Hierarchical B-spline refinement, Computer Graphics 22(4) (1988) 205–212.

[34] R. Kraft, Surface Fitting and Multiresolution Methods, Vol. 2, Vanderbilt University Press, 1997, Ch. Adaptive and linearly independent multilevel B-splines, pp. 209–218.

[35] A. V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 200 (2011) 3554–3567.

[36] D. Schillinger, L. Debé, M. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and T-spline CAD surfaces, Comput. Meth. Appl. Mech. Eng. 249–252 (2012) 116–150.

[37] P. B. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical B-spline finite element method, Comput. Meth. Appl. Mech. Eng. 253 (2013) 584–598.

[38] C. Giannelli, B. Jüttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (2012) 485–498.

[39] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical T-meshes, Graphical Models 70 (2008) 76–86.

[40] T. Dokken, T. Lyche, K. F. Pettersen, Polynomial splines over locally refined box-partitions, Comput. Aid. Geom. Design 30(3) (2013) 331–356.

[41] C. de Boor, Splines as linear combinations of B-splines. A Survey (1976).

[42] M. Neamtu, Mathematical Methods for Curves and Surfaces, Vanderbilt University, Nashville, TN, USA, 2001, Ch. What is the Natural Generalization of Univariate Splines to Higher Dimensions?, pp. 355–392.

[43] M. Neamtu, Delaunay Configurations and Multivariate Splines: A Generalization of a Result of B. N. Delaunay, Transactions of the American Mathematical Society 359, No. 7 (2007) 2993–3004.

[44] M. Neamtu, Bivariate simplex B-Splines: A New Paradigm, in: In SCCG 01: Proceedings of the 17th Spring conference on Computer graphics, 2001, pp. 71–78.

[45] C. de Boor, K. Höllig, S. Riemenschneider, Box Splines, Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[46] M. J. D. Powell, M. A. Sabin, Piecewise quadratic approximations on triangles, ACM Trans. Math. Softw. 3 (4) (1977) 316–325.

[47] H. Speleers, P. Dierckx, S. Vandewalle, Numerical solution of partial differential equations with powell-sabin splines, J. Comput. Appl. Math. 189 (1-2) (2006) 643–659.

[48] H. Speleers, C. Manni, F. Pelosi, From NURBS to NURPS geometries, Computer Methods in Applied Mechanics and Engineering 255 (2013) 238–254.

[49] N. Jaxon, X. Qian, Isogeometric analysis on triangulations., Computer-Aided Design 46 (2014) 45–57.

[50] J. M. Escobar, J. M. Cascón, E. Rodríguez, R. Montenegro, A new approach to solid modeling with trivariate T-splines based on mesh optimization, Comput. Meth. Appl. Mech. Eng. 200 (2011) 3210–3222.

[51] J. M. Escobar, R. Montenegro, E. Rodríguez, J. M. Cascón, The meccano method for isogeometric solid modeling and applications, Engineering with Computers 30 (3) (2014) 331–343.

[52] R. Montenegro, J. M. Cascón, E. Rodríguez, J. M. Escobar, G. Montero, The meccano method for automatic three-dimensional triangulation and volume parametrization of complex solids, in: Developments and Applications in Engineering Computational Technology, Saxe-Coburg Publications, Stirling, 2010, pp. 19–48.

[53] M. Brovka, J. I. López, J. M. Escobar, J. M. Cascón, R. Montenegro, A new method for T-spline parameterization of complex 2D geometries, Engineering with Computers 30 (4) (2014) 457–473.

[54] J. I. López, M. Brovka, J. M. Escobar, R. Montenegro, G. V. Socorro, Strategies for optimization of hexahedral meshes and their comparative study, Engineering with Computers 33 (1) (2017) 33–43.

[55] M. Brovka, Construction of polynomial spline spaces over t-meshes for its application in isogeometric analysis, Ph.D. thesis, University of Las Palmas de Gran Canaria (2016).

[56] M. Brovka, J. I. López, J. M. Escobar, R. Montenegro, J. M. Cascón, A simple strategy for defining polynomial spline spaces over hierarchical t- meshes, Computer-Aided Design 72 (2016) 140–156.

[57] C. De Boor, A practical guide to splines, Applied mathematical sciences, Springer-Verlag, New York, 2001.

[58] H. Samet, Foundations of Multidimensional and Metric Data Structures, Morgan Kaufmann Publishers, Burlington, Massachusetts, 2006.

[59] D. Moore, The cost of balancing generalized quadtrees, in: Proceedings of the Third ACM Symposium on Solid Modeling and Applications, SMA '95, ACM, New York, NY, USA, 1995, pp. 305–312. doi:10.1145/218013.218078.

[60] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J. M. González-Yuste, Simultaneous untangling and smoothing of tetrahedral meshes, Comput. Meth. Appl. Mech. Eng. 192 (2003) 2775–2787.

[61] M. S. Floater, Parametrization and smooth approximation of surface triangulations, Comput. Aid. Geom. Design 14 (1997) 231–250.

[62] J. M. Cascón, R. Montenegro, J. M. Escobar, E. Rodríguez, G. Montero, A new meccano technique for adaptive 3-D triangulation, in: Proc. of the 16th International Meshing Roundtable, Springer, Berlin, 2007, pp. 103–120.

[63] J. M. Cascón, R. Montenegro, J. M. Escobar, E. Rodríguez, G. Montero, The meccano method for automatic tetrahedral mesh generation of complex genus-zero solids, in: Proc. of the 18th International Meshing Roundtable, Springer, Berlin, 2009, pp. 463–480.

[64] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J. M. González-Yuste, SUS Code: Simultaneous mesh untangling and smoothing code, http://www.dca.iusiani.ulpgc.es/proyecto2012-2014/html/Software.html (2010).

[65] R. Montenegro, J. M. Cascón, J. M. Escobar, E. Rodríguez, G. Montero, An automatic strategy for adaptive tetrahedral mesh generation, Appl. Num. Math. 59 (2009) 2203–2217.

[66] S. A. Coons, Surfaces for computer aided design, Springfield, 1964.

[67] G. Farin, D. Hansford, Discrete Coons patches, Comput. Aid. Geom. Design 16 (1999) 691–700.

[68] P. M. Knupp, Algebraic mesh quality metrics, SIAM J. Sci. Comput. 23 (2001) 193–218.

[69] P. M. Knupp, A method for hexahedral mesh shape optimization, Int. J. Num. Meth. Eng. 58 (2) (2003) 319–332.

[70] L. A. Freitag, P. M. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, Int. J. Num. Meth. Eng. 53 (2002) 1377–1391.

[71] L. A. Freitag, P. Plassmann, Local optimization-based simplicial mesh untangling and improvement, Int. J. Num. Meth. Eng. 49 (2000) 109–125.

[72] J. M. Escobar, R. Montenegro, E. Rodríguez, G. Montero, Simultaneous aligning and smoothing of surface triangulations, Engineering with Computers 27 (2011) 17–29.

[73] J. Dennis, R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Englewood Cliffs, New Jersey, 1983.
URL `http://books.google.es/books?id=RtxcWd0eBD0C`

[74] J. M. Escobar, R. Montenegro, G. Montero, E. Rodríguez, J. M. González-Yuste, Smoothing and local refinement techniques for improving tetrahedral mesh quality, Computers & Structures 83 (2005) 2423–2430.

[75] S. Benzley, E. Steven, E. Perry, K. Merkley, B. Clark, G. Sjaardema, A comparison of all-hexahedral and all-tetrahedral finite element meshes for elastic and elasto-plastic analysis, in: Proceedings 4th International Meshing Roundtable, Sandia National Laboratories, 1995, pp. 179–191.

[76] A. O. Cifuentes, A. Kalbar, A performance study of tetrahedral and hexahedral elements in 3D finite element structural analysis, Finite Elements in Analysis and Design 12 (3-4) (1992) 313–318.

[77] V. Weingarten, The controversy over hex or tet meshing, Machine design 66 (8) (1994) 74–76.

[78] E. Ruiz-Gironés, X. Roca, J. Sarrate, R. Montenegro, J. M. Escobar, Simultaneous untangling and smoothing of quadrilateral and hexahedral meshes using an object-oriented framework, Advances in Engineering Software 80 (2015) 12–24.

[79] T. J. Wilson, J. Sarrate, X. Roca, R. Montenegro, J. M. Escobar, Untangling and smoothing of quadrilateral and hexahedral meshes, in: Proc. of The Eighth International Conference on Engineering Computational Technology, 2012.

[80] O. V. Ushakova, Nondegeneracy tests for hexahedral cells, Computer Methods in Applied Mechanics and Engineering 200 (17-20) (2011) 1649–1658.

[81] A. Gargallo-Peiró, Generation of curved meshes for high-order unstructured methods, Ph.D. thesis, Universitat Politècnica de Catalunya (2014).

[82] P. M. Knupp, On the ivertability of the isoparametric map, Comput. Methods Appl. Mech. Eng. 78 (3) (1990) 313–329.

[83] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J. M. González-Yuste, SUS Code: Simultaneous mesh untangling and smoothing code, Technote, http://www.dca.iusiani.ulpgc.es/SUScode/TechNote.pdf (2003).

[84] P. M. Knupp, Algebraic mesh quality metrics for unstructured initial meshes, Finite Elements in Analysis and Design 39 (2003) 217–241.

[85] P. Knabner, S. Korotov, G. Summ, Conditions for the invertibility of the isoparametric mapping for hexahedral finite elements, Finite Elements in Analysis and Design 40 (2003) 159–172.

[86] J. Garloff, C. Jansson, A. P. Smith, Lower bound functions for polynomials, Computational and Applied Mathematics 157 (1) (2003) 207–225.

[87] A. Johnen, J. F. Remacle, C. Geuzaine, Geometrical validity of high-order triangular finite elements, Engineering with Computers 30 (3) (2014) 375–382.

[88] C. Geuzaine, A. Johnen, L. Lambrechts, J. F. Remacle, T. Toulorge, The generation of valid curvilinear meshes, in: N. Kroll, C. Hirsch, F. Bassi, C. Johnston, K. Hillewaert (Eds.), IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach, Vol. 128, Springer International Publishing, 2015, pp. 15–39.

[89] G. Montero, R. Montenegro, J. Escobar, A 3-d diagnostic model for wind field adjustment, Journal of Wind Engineering and Industrial Aerodynamics 74 76 (1998) 249 – 261.

[90] L. Ferragut, R. Montenegro, G. Montero, E. Rodriguez, M. Asensio, J. Escobar, Comparison between 2.5-d and 3-d realistic models for wind field adjustment, Journal of Wind Engineering and Industrial Aerodynamics 98 (10 11) (2010) 548 – 558.

[91] C. Geuzaine, J. F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

Summary of the dissertation in Spanish

Construcción de parametrizaciones spline para su aplicación en Análisis Isogeométrico

Construcción de parametrizaciones spline para su aplicación en Análisis Isogeométrico

Resumen

En esta tesis se aborda la construcción de parametrizaciones spline de geometrías 2D y 3D para su aplicación en Análisis Isogeométrico. Se presenta una estrategia, basada en las ideas del método del Mecano y en un nuevo procedimiento de optimización de T-mesh, que permite obtener parametrizaciones spline de alta calidad a partir de la descripción de la frontera de la geometría. Como primer paso, nuestro método define un mapeo paramétrico entre la frontera del objeto y el contorno del dominio paramétrico. Luego se construye una T-mesh adaptada para aproximar las singularidades de la geometría con una precisión dada. La clave del método está en definir una transformación isomórfica entre la T-mesh paramétrica y la T-mesh física, encontrando la posición óptima de los nodos interiores de la malla aplicando un nuevo algoritmo de desenredo y suavizado simultáneo. La representación spline de la geometría se calcula imponiendo condiciones de interpolación usando la correspondencia uno a uno entre las mallas paramétricas y físicas. Para medir la calidad de la parametrización se evalúa su mean ratio Jacobian. De este modo, se detectan las áreas de baja calidad y se realiza un refinamiento adaptativo para aumentar el número de grados de libertad en las regiones con alta distorsión. Esta estrategia permite obtener una parametrización válida para el análisis sin Jacobiano negativo, incluso en geometrías complicadas. El método propuesto se describe de forma detallada y su efectividad se prueba con distintas geometrías. Además se presenta algunos ejemplos de resolución con Análisis Isogeométrico utilizando las geometrías parametrizadas con nuestro método.

Índice general

Índice de figuras

1. Introducción

1.1 Concepto de Análisis Isogeométrico

En la década de los 60 (1960-1970) comienza a tener auge el diseño asistido por ordenador (CAD). Esta tecnología supuso un avance importante para la ingeniería, ya que facilitaba enormemente el proceso de diseño industrial que hasta ese momento era una tarea manual. Actualmente las herramientas CAD están fuertemente integradas en sectores como la automoción, la construcción naval, la industria aeroespacial, el diseño industrial y arquitectónico, así como en el cine para el diseño de efectos especiales. El software CAD utiliza herramientas matemáticas para la representación de curvas y superficies, a partir de las cuales se compone el modelo geométrico. Las funciones B-splines y las NURBS (Non-Uniform Rational B-splines) [1, 2] son dos de las tecnologías más utilizadas en los programas CAD. Ambas tecnologías permiten modelar superficies y son capaces de representar secciones cónicas de forma exacta. No obstante, las funciones B-spline ofrecen ciertas ventajas, como una mayor suavidad y la propiedad de envolvente convexa, especialmente importante en CAD.

Otra herramienta muy importante en el mundo de la ingeniería es el Método de los Elementos Finitos (FEM). Desde su emergencia (1950-1960), este método numérico para la resolución de ecuaciones en derivadas parciales, ha sido objeto de un gran número de investigaciones científicas y se ha convertido en un estándar para la ingeniería industrial. Las dos partes fundamentales del método son la formulación variacional del problema físico y la discretización del espacio utilizado para aproximar la solución. El dominio computacional del problema se descompone en elementos simples (triángulos, cuadriláteros, tetraedros, hexaedros) sin intersecciones. A esta parte se le conoce como proceso de generación de malla. FEM típicamente utiliza como funciones de base para el espacio aproximador funciones polinómicas interpoladoras de Lagrange o Hermite, las cuales se definen localmente en cada uno de los elementos en los que se descompone el domino físico.

Habitualmente, la geometría sobre la que se resuelve un problema con FEM proviene de un modelo CAD, a partir del cual se genera una malla con las características adecuadas para el análisis. El proceso de generación de la malla no es una tarea trivial, y suele ser la parte que mayor tiempo requiere de todo el proceso de análisis. Ambas tecnologías, CAD y FEM, han evolucionado de forma separada y utilizan herramientas geométricas diferentes. Esto hace que la comunicación y transferencia de información entre ambas tecnologías requiera un gran esfuerzo y coste. Esta brecha puede suponer un inconveniente importante, especialmente en cierto tipo de problemas como la optimización de diseños. Otra situación donde la interacción automática entre el modelo CAD y la malla FEM cobra importancia es cuando se necesita la aproximación de una frontera curva, para lo cual es necesario realizar refinamientos adaptativos en la malla. Además, generalmente la geometría FEM no es capaz de reproducir de forma exacta el modelo diseñado en CAD, lo que puede hacer que en determinados problemas se produzcan errores importantes en la solución numérica. Parece evidente que con todos estos inconvenientes es necesario unificar en un único proceso tanto el diseño como el análisis. Para ello, existen dos opciones: o bien el software CAD es capaz de generar

un modelo geométrico que pueda ser utilizado directamente en el análisis, o bien existe un método automático para obtener un modelo adecuado para el análisis a partir de su representación CAD. En 2005, Tom Hughes y sus colaboradores [3] propusieron el Análisis Isogeométrico (IGA), en el cual se adapta el proceso de análisis para que se pueda aplicar directamente en geometrías CAD. La idea consiste en utilizar para el análisis las mismas funciones de base que se utilizan para la generación del modelo CAD (originalmente NURBS). El Análisis Isogeométrico puede ser considerado como una generalización de FEM que utiliza funciones de base de mayor regularidad.

Desde su aparición, IGA ha atraído la atención de la comunidad científica y ha sido objeto de numerosas publicaciones en los últimos años. Actualmente es considerada una herramienta prometedora para lograr superar la brecha entre las industrias del CAD y FEM. Además, ofrece una serie de ventajas frente al Método de Elementos Finitos clásico, como por ejemplo:

- Una mayor precisión en problemas sensibles a la aproximación geométrica, ya que permite trabajar con la geometría exacta (problemas de contacto, boundary layer problems en aerodinámica y dinámica de fluidos). Además, para ciertos problemas, aunque no se consiga una geometría exacta, es mejor disponer de una aproximación suave del contorno, principalmente si se requiere continuidad en la curvatura del contorno.
- Mejor precisión de la solución numérica debido a la mayor regularidad de las funciones de base [4].
- Debido al mayor grado de continuidad, se puede utilizar IGA para resolver ecuaciones en derivadas parciales de orden mayor a dos sin necesidad de cambiar la formulación variacional. Por ejemplo, la ecuación Cahn-Hilliard phase-field [5, 6, 7, 8].
- Posibilidad de métodos de colocación usando formulación fuerte [9, 10, 11, 12].

Aunque el uso del Análisis Isogeométrico ofrece importantes ventajas, existen algunos problemas que deben ser resueltos antes de poder llevarlo a la práctica. En la siguiente sección se describen estos problemas.

1.2 Estado del arte

IGA es un método relativamente nuevo que necesita del desarrollo de un marco teórico y la superación de ciertos problemas. Aún existen muchos frentes que se deben estudiar: una teoría de estimación de error [13, 14, 15], reglas de cuadratura eficientes [16, 17], mejorar la eficiencia de los solvers directos e iterativos para los sistemas que surgen del método, la imposición de condiciones de contorno o la influencia de la calidad de la parametrización en la precisión numérica de la solución [18]. Sin embargo, de todos los problemas abiertos en IGA, los más importantes para su uso son el problema de la parametrización del dominio computacional y el refinamiento local.

1.2.1 Parameterización del dominio

Los modelos CAD proporcionan la descripción de la superficie de un sólido. Esta representación spline del contorno se puede utilizar directamente en isogeometric shell analysis o en isogeometric boundary integral method. En cambio, para la aplicación del Análsis Isogeométrico se requiere de una descripción volumétrica del dominio. Éste sigue siendo un problema abierto en el Análisis Isogeométrico: cómo generar una representación spline de un sólido partiendo de la descripción CAD de su superficie. Como señalan Cottrell y otros en [19], "the most significant challenge facing Isogeometric Analysis is developing three-dimensional spline parameterizations from surfaces". Una parametrización spline se dice que es válida para el análisis si no tiene intersecciones, es decir, la transformación es invertible. Además, para conseguir una buena precisión en los resultados numéricos también es necesario que la parametrización volumétrica tenga una buena calidad. Para ello es deseable conseguir ortogonalidad y uniformidad de las curvas isoparamétricas. Para dominios complejos no es trivial obtener una parametrización de buena calidad para la aplicación de IGA. Es esencial disponer de un método eficaz para su construcción. A continuación se resume los principales trabajos que han sido realizados para tratar este problema.

En [20], la parametrización se determina resolviendo un problema de optimización con restricciones para encontrar las posiciones de los puntos de control de una superficie plana B -spline. La optimización consiste en la minimización de un funcional de energía que impone condiciones de ortogonalidad y uniformidad, donde además, se imponen restricciones para que la parametrización sea inyectiva. Esta idea se ha extendido a 3D en [21]. Otra técnica similar fue propuesta por estos autores en [22, 23]. Ellos utilizan un mapeo armónico obtenido a través de la resolución de un problema de optimización para los puntos de control. El uso de un mapeo armónico es una característica común de varios trabajos sobre métodos de parametrización en 2D y 3D. Por ejemplo, Li et al. [24] construyen un mapeo volumétrico utilizando el método de elementos de contorno. El algoritmo, que es complejo, puede ser aplicado a cualquier género de datos, pero requiere un proceso de prueba y error que debe ser guiado por la experiencia humana.

Martin et al. [25] presentan una metodología basada en funciones armónicas discretas para parametrizar un sólido, en donde el dato de entrada es una malla de tetraedros del sólido. El método implica la resolución de varias ecuaciones de de tipo Laplace mediante FEM, primero sobre la superficie y después sobre el dominio completo 3D. Se calculan dos funciones ortogonales armónicas para construir una malla de hexaedros del sólido que después se suaviza para eliminar irregularidades. El usuario debe elegir la posición de varios puntos críticos para establecer la superficie de parametrización y para fijar una semilla para generar el esqueleto.

Zhang et al. proponen en [26] un procedimiento para la construcción de sólidos de género cero a partir de la triangulación definida por la frontera de una malla de tetraedros. En primer lugar, se establece un mapeo entre la triangulación y la frontera del dominio paramétrico, el cubo unitario, utilizando para ello la parametrización de Floater. A continuación, se lleva a cabo una división octree del cubo que continúa hasta que el error entre la triangulación de entrada y la T-mesh de la superficie está por

debajo de una tolerancia. Finalmente, los nodos del interior son recolocados mediante un proceso de optimización que maximiza el peor Jacobiano. El método fue extendido a sólidos de genero arbitrario en [27] .

1.2.2 Refinamiento local en IGA

Originalmente el concepto de Análisis Isogeométrico fue propuesto y probado usando funciones de base NURBS ya que estas eran la herramienta más común utilizada por el software de CAD en ese momento. Sin embargo, estas funciones tienen un importante inconveniente: su estructura de producto tensorial no permite el refinamiento local. Una superficie NURBS se define por un conjunto de puntos de control que se encuentran, topologicamente, en una malla rectangular. La inserción de un nuevo punto de control provoca la inserción de toda una fila de puntos de control extendidos por todo el dominio. Esto conduce a un gran número de puntos de control superfluos. La estructura de producto tensorial las hace ineficientes para la representación de características locales requiriendo el uso de varios parches NURBS unidos entre sí. Con frecuencia, esta unión es discontinua, esto es, hay huecos y solapamientos entre diferentes parches que hacen que el modelo geométrico no sea adecuado para el análisis. Las T-splines fueron propuestas por Sederberg et al. [28] como una alternativa a las NURBS que permite el refinamiento local. Las T-splines pueden ser consideradas como una generalización de NURBS donde las filas o las columnas asociadas a los puntos de control pueden quedar truncadas, produciendo lo que se conoce como una T-junction. Esta característica permite el refinamiento local.

Los Programas de CAD basados en NURBS como Maya y Rhino incorporan aplicaciones (plug-in) de T-splines. Las T-splines ofrecen una herramienta flexible para crear superficies sin huecos ni pliegues capaces de representar características locales detalladas con un menor número de puntos de control de los necesarios con NURBS, ver Fig. 1.1. Sin embargo, las T-splines inicialmente introducidas para CAD no son adecuadas para su uso en IGA, ya que carecen de algunas propiedades esenciales para el análisis y la convergencia de IGA, como son la independencia lineal, el encaje de espacios y la consistencia de orden tres sobre todo el dominio (capacidad de representar polinomios cúbicos en todo el dominio). Además, el hecho de ser funciones racionales complican los cálculos y aumentan el coste computacional.

Por lo tanto, sigue siendo un problema abierto en el contexto del Análisis Isogeométrico encontrar una alternativa a las NURBS que permita refinamiento local y pueda ser utilizado para el análisis. Para que las blending functions de las splines puedan ser utilizadas en el análisis numérico y que la convergencia sea adecuada, estas funciones deben cumplir algunos requisitos: independencia lineal, consistencia de orden tres, encaje de espacios de aproximación y la posibilidad de realizar refinamiento local. Este tema ha sido objeto de numerosos trabajos de investigación en los últimos años.

Las T-splines aptas para el análisis (analysis-suitable) propuestas por Scott et al. en [29], son una clase de T-splines definidas sobre T-meshes que cumplen con ciertas restricciones topológicas formuladas en términos de las extensiones de las T-junctions. Las blending functions definidas sobre una analysis-suitable T-mesh son linealmente inde-

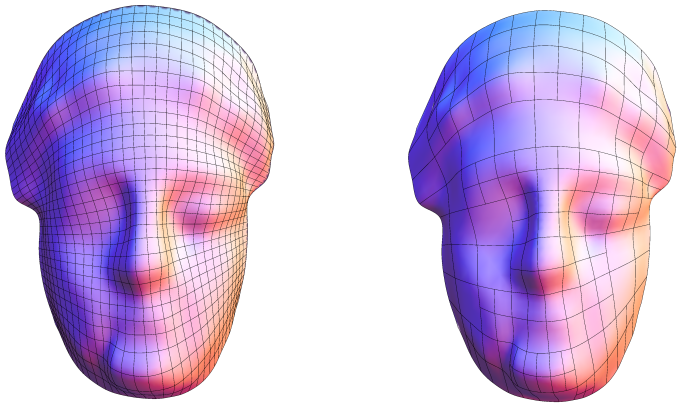


Figura 1.1. Comparación de NURBS y T-spline. (a) Cara de Igea modelada con NURBS con 1225 puntos de control. (b) Versión T-spline con 288 puntos de control.

pendientes [30] y son consistentes. El algoritmo de refinamiento permite lograr mejoras muy localizadas y construir espacios de aproximación anidados, sin embargo presenta una elevada complejidad de implementación y, por lo que sabemos, su generalización a 3D es todavía una cuestión abierta.

Otro enfoque para el problema de enriquecimiento local del espacio de aproximación es el refinamiento jerárquico. Introducido originalmente por Forsey y Bartels en [31] y más tarde desarrollado en [32]. La técnica de refinamiento jerárquico, en el contexto del Análisis Isogeométrico, se describe en [33, 34, 35]. Este enfoque se basa en una idea simple y natural para construir espacios multinivel mediante la sustitución de las funciones de niveles más groseros con funciones de niveles más finos. A partir de una malla inicial uniforme, el esquema de refinamiento jerárquico conduce a la construcción secuencial de espacios anidados de splines con funciones de base linealmente independientes. La fácil generalización a 3D hace de esta técnica una opción atractiva para el refinamiento local. Sin embargo, un inconveniente de esta estrategia es la imposibilidad de definir espacios de splines sobre una T-mesh arbitraria, así como la presencia de funciones base redundantes y un excesivo solapamiento de los soportes de estas funciones.

Otras estrategias para realizar un refinamiento local son las C^1-continuas PHT-splines [36] y las Locally refined B-splines (LR-splines) [37].

Vale la pena mencionar otro grupo de splines que no tienen la estructura de producto tensorial. Consiste en splines definidas sobre triangulaciones [38], y en particular, splines definidas sobre una triangulación de Delaunay [39, 40, 41], box splines [42] y C^1 splines definidas sobre un refinamiento de Powell-Sabin del triángulo [43]. Ejemplos recientes de la aplicación del IGA utilizando splines sobre las triangulaciones se pueden encontrar en [44, 45, 46]. Las splines sobre triangulaciones ofrecen mayor flexibilidad y capacidad de adaptación a dominios irregulares que las splines basadas en producto tensorial. No

obstante, quedan muchas cuestiones por resolver en este campo, como por ejemplo su extensión a 3D.

1.3 Objetivo y estructura de la tesis

Esta tesis es uno de los resultados del trabajo de nuestro grupo en el campo del Análisis Isogeométrico. Como se ha indicado anteriormente, IGA es un método relativamente nuevo con muchos puntos aún por explorar. Desde que nos embarcamos en este proyecto hemos estudiado y descubierto muchas cuestiones interesantes acerca de IGA. Pero de todas ellas, prestamos especial atención a dos problemas abiertos en IGA: la parametrización de un dominio computacional a partir de su descripción superficial y la construcción de espacios spline con las propiedades adecuadas para el análisis. En esta tesis se incluye un resumen de nuestro trabajo para construir espacios spline, aunque este problema fue estudiado en profundidad y presentado en otra tesis doctoral de nuestro grupo.

En esta tesis se aborda el problema de la construcción de una parametrización spline válida para ser utilizada en el Análisis Isogeométrico. La principal contribución de esta tesis es un nuevo método que permite obtener una parametrización spline de alta calidad tanto para geometrías 2D como 3D. La estrategia está basada en algunos trabajos previos del grupo [47, 48] en los cuales se propone un método para obtener una parametrización volumétrica mediante la deformación de una malla de tetraedros de un sólido. Nuestro objetivo en esta tesis es extender las ideas del método del Meccano [49] para poder trabajar directamente sobre una T-mesh, sin la necesidad de una malla de tetraedros intermedia. La estrategia que se propone sólo requiere como dato de entrada una descripción de la frontera de la geometría, y construye automáticamente una transformación spline entre el dominio paramétrico y físico. Para ello primero se establece un mapeo paramétrico entre el input y la frontera del dominio paramétrico. Se construye una T-mesh paramétrica adaptada a las singularidades de la geometría para aproximar las características de la frontera del objeto con una precisión establecida. Luego se construye la T-mesh física mapeando al dominio físico los puntos de la frontera de la malla paramétrica. Los nodos del interior de la T-mesh física se colocan aplicando un procedimiento de desenredo y suavizado de T-mesh, el cual es la clave de nuestro método. Por último, se obtiene la representación spline de la geometría imponiendo condiciones de interpolación, utilizando para ello la información proporcionada por la correspondencia uno a uno establecida entre la malla paramétrica y física.

La tesis se organiza de la siguiente forma.

- En el capítulo 2 se presenta uno de los principales resultados de esta tesis: un método de parametrización T-spline para geometrías 2D. La estrategia propuesta fue publicada in [50].
- El capítulo 3 resume otro resultado de nuestra investigación: una estrategia para la construcción de espacios spline polinómicos sobre una T-mesh jerárquica con estructura quadtree y octree. Estas bases spline se utilizan posteriormente en

nuestro método general de parametrización 2D y 3D. Este trabajo fue presentado en otra tesis doctoral del grupo [51] y fue publicado en [52].

- En el capítulo 4 se generaliza nuestro método de parametrización spline para geometrías tanto 2D como 3D.
- En el capítulo 5 se prueba las estrategias de parametrización propuestas utilizando diferentes geometrías. Se resuelve diferentes problemas con IGA empleando algunas de las parametrizaciones obtenidas. Además se muestra un ejemplo de otra posible aplicación de nuestro trabajo: la optimización de mallas de cuadriláteros y hexaedros de alto orden.
- El capítulo 6 presenta las conclusiones y se exponen algunas líneas futuras de investigación.

2. Método para la parametrización T-spline de geometrías 2D

En este capítulo se presenta una estrategia para la construcción de parametrizaciones T-spline de geometrías 2D para su aplicación en el Análisis Isogeométrico. Nuestra estrategia está basada en la idea del método del Mecano y un nuevo procedimiento de optimización de T-mesh.

El método propuesto sólo demanda como dato de entrada una representación de la frontera de la geometría. El algoritmo obtiene como resultado una transformación paramétrica de alta calidad entre el objeto 2D y el dominio paramétrico, el cuadrado unitario. Primero se define un mapeo paramétrico entre el input y la frontera del dominio paramétrico. Luego se construye una T-mesh adaptada a las singularidades geométricas del dominio para preservar sus características con una tolerancia deseada. La clave del método está en definir una transformación isomórfica entre la T-mesh paramétrica y física encontrando la posición óptima de los nodos del interior mediante la aplicación de un nuevo método de desenredo y suavizado de T-mesh. La representación T-spline bivariada se calcula imponiendo condiciones de interpolación en los puntos situados tanto en la frontera como en el interior de la geometría. El método que se propone permite además modelar objetos con geometrías anidadas, la cuales se pueden utilizar para resolver problemas con dominios compuestos de varios materiales.

En trabajos previos del grupo [47, 48] se propuso un método para obtener una parametrización volumétrica mediante la deformación de una malla de tetraedros del sólido. En general, esta aproximación no proporciona una T-mesh de calidad óptima en el sentido de su ortogonalidad y uniformidad. En este trabajo proponemos otra estrategia, donde la optimización se realiza directamente sobre la T-mesh.

Este capítulo está organizado de la siguiente forma. En la siguiente sección se describe las etapas principales del algoritmo de parametrización propuesto. En la sección 2.2 se explica el proceso de parametrización del contorno y la construcción de la T-mesh adaptada a las singularidades de la geometría. La sección 2.3 describe el procedimiento de desenredo y suavizado simultáneo de T-mesh. El modelado de la geometría mediante T-splines bivariadas se desarrolla en la sección 2.4. En la sección 2.5 se describe una estrategia para la mejora de la calidad de la parametrización spline basada en refinamientos adaptativos. La construcción de geometrías anidadas se desarrolla en la sección 2.6. Y finalmente, en la sección 2.7 se presentan algunas conclusiones y se establece los siguientes retos.

2.1 Esquema general del método

En esta sección se resume el método de parametrización spline propuesto. Algunas ideas están basadas en trabajos previos del grupo en el área del desenredo y suavizado y en el método del Mecano [53, 54, 55, 56, 57, 49], adaptándolas a los requerimientos del trabajo presente.

El algoritmo propuesto incluye las siguientes etapas:

1. Parametrización del contorno y construcción de una T-mesh adaptada: se define una correspondencia biyectiva entre input que describe la geometría y la frontera

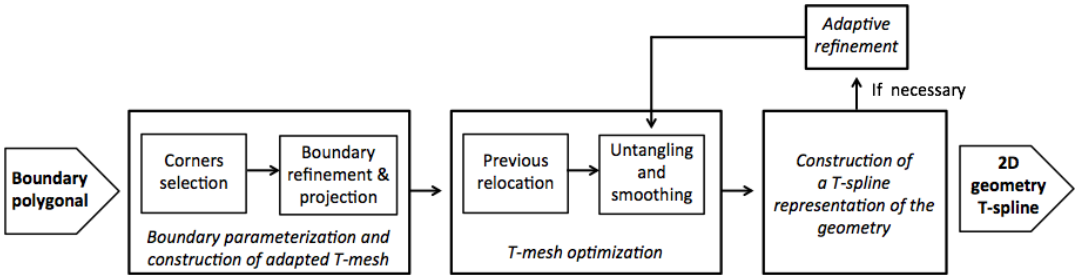


Figura 2.1. Esquema general del método de parametrización T-spline.

del dominio paramétrico. Luego, se genera una T-mesh adaptada refinando para aproximar la geometría con una tolerancia dada. Durante este proceso, los nodos del contorno del dominio paramétrico se mapean al contorno de la geometría.

2. Optimización de la T-mesh: se colocan los nodos del interior de la T-mesh mediante un procedimiento de desenredo y suavizado simultáneo. Para facilitar el desenredo, previamente se realiza una recolocación inicial.

3. Construcción de una representación T-spline de la geometría: la parametrización T-spline se obtiene imponiendo condiciones de interpolación. Como puntos de interpolación se toman los nodos de la T-mesh física obtenidos tras el proceso de optimización de T-mesh. Además, es necesario tomar ciertos puntos de control adicionales.

4. Refinamiento adaptativo para mejorar la calidad de la parametrización: si la calidad de la parametrización no es suficiente, se realizan refinamientos adaptativos para aumentar el número de grados de libertad en las áreas de alta distorsión. Luego se repite el proceso a partir del paso 2 hasta alcanzar una buena parametrización T-spline.

La T-mesh que se emplea en nuestra estrategia tiene una estructura de quadtree balanceado [58]: todos los refinamientos se realizan dividiendo una celda en 4 celdas iguales y se aplica un algoritmo de balanceo de modo que sólo se permita un hanging node por arista. El input para representar la geometría viene dado por una poligonal, aunque también podría venir dado por curvas B-spline.

En la figura 2.1 se muestra un esquema del método. En las siguientes secciones se describe cada una de sus etapas.

2.2 Parametrización del contorno y construcción de una T-mesh adaptada

Para definir el mapeo paramétrico entre la poligonal de entrada que describe la geometría y la frontera del dominio paramétrico, el cuadrado unitario, primero se se-

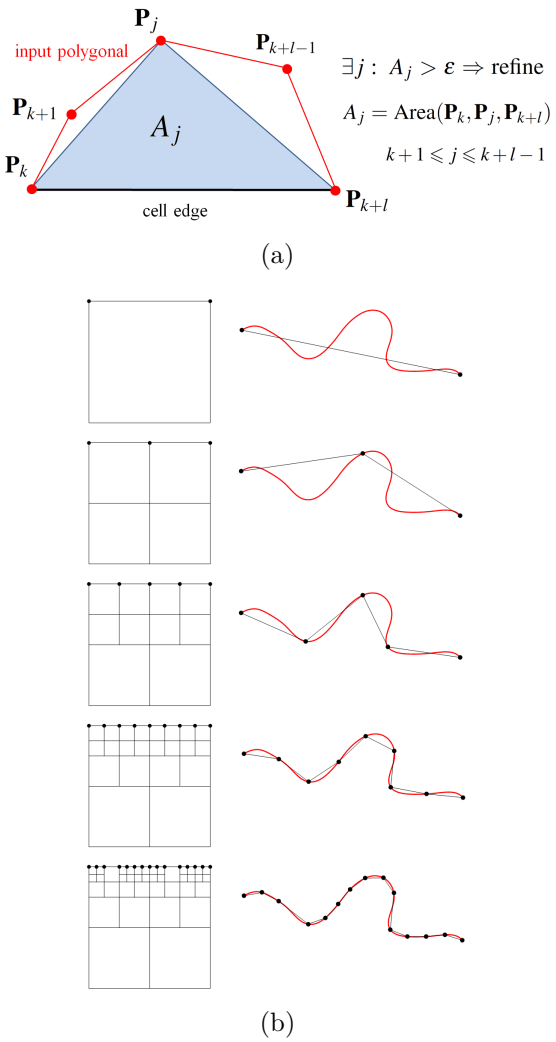


Figura 2.2. (a) Ilustración del criterio de refinamiento para aproximar la frontera de la geometría con una tolerancia dada. (b) Ejemplo de aproximación del contorno correspondiente a la arista superior del cuadrado unitario. La línea roja representa el input.

lecciona cuatro puntos de la poligonal que se harán corresponder con las cuatro esquinas del espacio paramétrico. Estos puntos dividen la poligonal en cuatro partes que luego se mapean sobre cada lado del cuadrado vía parametrización chord-length. Se debe realizar una elección apropiada de los cuatro puntos de la poligonal. Estos puntos deben estar situados en zonas donde la poligonal forme una región convexa, es decir, el ángulo interior formado por la poligonal en esos puntos debe ser inferior a 180°. Obviamente, el valor óptimo para este ángulo es 90°.

Luego se construye una T-mesh adaptada que aproxima la polygonal con una tolerancia ε establecida. Para ello se calcula un error de aproximación para cada celda del

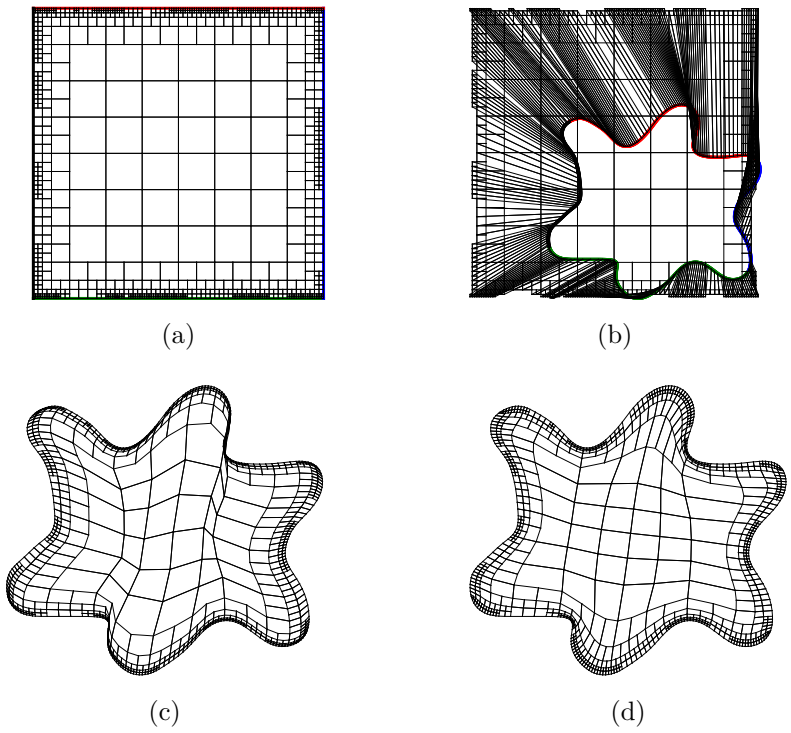


Figura 2.3. Etapas de construcción de una T-mesh para la geometría Spot. (a) T-mesh paramétrica adaptada a la frontera de la geometría. (b) Malla física enredada tras el mapeo del contorno (las líneas en color representan la correspondencia entre las fronteras paramétrica y física). (c) Recolocación previa mediante Coons patch. (d) T-mesh física optimizada.

contorno de la T-mesh, de modo que la celda se refina si su error es mayor que $\boldsymbol{\varepsilon}$. El refinamiento de una celda genera un nuevo punto de frontera que se proyecta sobre la poligonal input, obteniendo así una aproximación más ajustada de la geometría como se muestra en la Fig. 2.2(b). Como criterio de error se evalúa el área de los triángulos formados por la arista de la celda en el espacio físico y cada vértice de la porción de poligonal aproximada por esta arista. Una celda se refina si hay al menos un triángulo cuya área es mayor que $\boldsymbol{\varepsilon}$. El criterio de refinamiento se ilustra en Fig. 2.2(a).

También es posible comenzar el proceso partiendo de una malla paramétrica con algunos refinamientos globales para asegurar un número suficiente de nodos libres en el interior de la geometría. Esto puede facilitar el proceso de desenredo y suavizado en geometrías complejas.

Como resultado de esta etapa se conoce la posición de los nodos de la frontera de la T-mesh. Los nodos del interior se situarán en sus posiciones óptimas mediante un procedimiento de optimización de T-mesh que se desarrolla en la siguiente sección. La figura 2.3(a) muestra un ejemplo de T-mesh paramétrica construida en esta etapa y la Fig. 2.3(b) muestra la T-mesh enredada en el espacio físico, la cual debe ser optimizada.

2.3 Optimización T-mesh

La clave de nuestro método está en el procedimiento de optimización que permite obtener una T-mesh física de alta calidad, la cual se utiliza para la construcción de la representación T-spline del objeto.

2.3.1 Recolocación previa

Es preferible realizar una recolocación de los nodos del interior para reducir el esfuerzo computacional durante el proceso de optimización. Para ello empleamos Coons patch [59, 60] para definir una superficie que interpola las curvas que definen la geometría. Llamemos $\mathbf{x}(\xi,0)$, $\mathbf{x}(\xi,1)$, $\mathbf{x}(0,\eta)$, $\mathbf{x}(1,\eta)$ a las curvas que definen el contorno. La interpolación Coons se define como

$$\begin{aligned}\mathbf{x}(\xi,\eta) =&(1-\xi)\mathbf{x}(0,\eta)+\xi\mathbf{x}(1,\eta)\\ &+(1-\eta)\mathbf{x}(\xi,0)+\eta\mathbf{x}(\xi,1)\\ &-\begin{bmatrix} 1-\xi & \xi \end{bmatrix}\begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1)\\ \mathbf{x}(1,0) & \mathbf{x}(1,1)\end{bmatrix}\begin{bmatrix} 1-\eta \\ \eta \end{bmatrix}\end{aligned}$$

Este proceso de recolocación facilita el desenredo, pero en general no es capaz de obtener una malla de buena calidad y puede producir enredos, como se puede ver en Fig. 2.3(c). Por lo tanto, es imprescindible aplicar un algoritmo eficaz de optimización.

2.3.2 Función objetivo

El proceso de optimización se realiza mediante una recolocación iterativa de cada nodo del interior de la malla (nodo libre), de tal forma que la nueva posición del nodo mejore la calidad de su malla local. La malla local de un nodo es el conjunto de todos los elementos conectados a ese nodo. La función objetivo para un nodo libre está basada en las medidas algebraicas de calidad propuestas por Knupp en [61, 62] para triángulos y cuadriláteros. La medida de calidad de forma para un triángulo se define en términos de la matriz Jacobiana del mapeo afín desde un triángulo ideal al triángulo dado (triángulo físico). Esta medida de calidad representa la desviación del triángulo físico respecto al ideal. Alcanza su valor máximo, 1, cuando el triángulo físico es similar al idea, y es igual a 0 si el triángulo está degenerado. La medida de distorsión de un elemento se define como la inversa de su calidad. Para medir la calidad de una malla local en una T-mesh, se descompone cada celda de la malla local en un conjunto de triángulos y se mide la calidad de cada uno de esos triángulos. La forma en la que se descompone la celda en triángulos depende del tipo de nodo libre. En concreto existen dos opciones: cuando el nodo libre es un nodo regular y cuando es un hanging node. La posición óptima del nodo libre se determina minimizando su función objetivo local. La función objetivo la definimos como la suma de las distorsiones de los triángulos formados en las celdas de la malla local. Para cada uno de estos triángulos

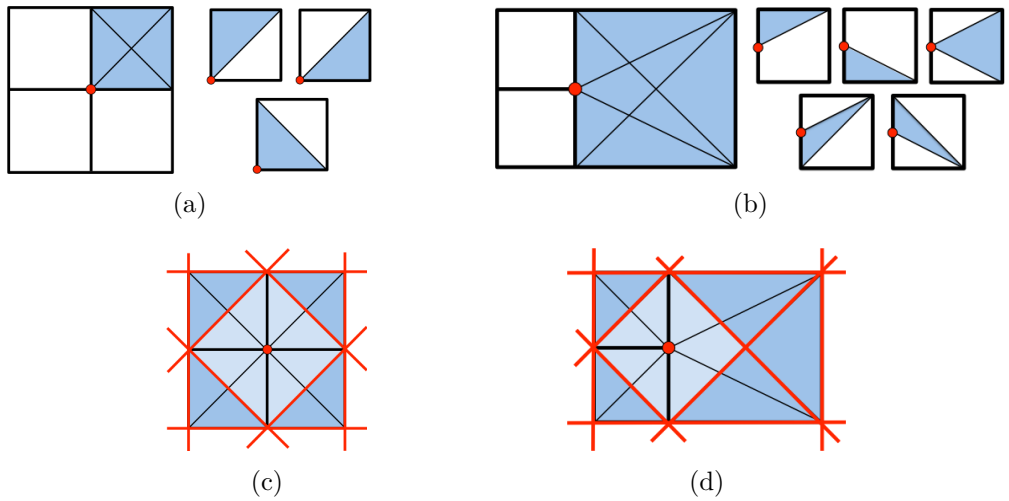


Figura 2.4. Descomposición en triángulos de la malla local. (a) Caso de nodo regular, donde cada celda se divide en tres triángulos. (b) Caso de hanging node, donde se forman 5 triángulos en la celda donde el nodo genera una T-junction. (c) Barrera y región factible inducidas por los 12 triángulos en la función objetivo para un nodo regular. (d) Barrera y región factible inducidas por los 11 triángulos en la función objetivo para un hanging node.

en la malla física, se toma como triángulo ideal su triángulo correspondiente en la malla paramétrica. De este modo, cada celda física tiene a tener la misma forma que su celda equivalente en el espacio paramétrico. Repitiendo este proceso para todos los nodos del interior de la malla, conseguimos que la malla física se parezca lo máximo posible a la malla paramétrica.

Un nodo regular puede estar rodeado por celdas de diferente tamaño. Para llevar a cabo la optimización, la malla local del nodo regular se descompone en 12 triángulos, tres triángulos por cada celda cuyas calidades dependen de la posición del nodo libre. La figura 2.4(a) ilustra esta descomposición en triángulos y la Fig. 2.4(c) muestra la región factible de la función objetivo.

En el caso de un hanging node, el nodo libre está rodeado por tres celdas y la malla local se descompone en un total de 11 triángulos. La celda donde el nodo libre forma T-junction se divide en 5 triángulos cuyas calidades dependen de la posición del nodo libre. Las otras dos celdas se dividen en 3 triángulos, exactamente igual que en el caso de un nodo regular. La figura 2.4(b) muestra la descomposición en triángulos de la malla local de un hanging node y la región factible de la función objetivo. Hay que destacar que para el caso ideal mostrado en la Fig. 2.4(d), la región factible inducida por estos 11 triángulos es la misma que se obtiene tras un refinamiento de la celda donde se forma la T-junction, ver Fig. 2.4(c).

A continuación se introduce una serie de conceptos para definir la función objetivo.

Sea T un triángulo cuyos vértices vienen dados por $\mathbf{x}_k = (x_k, y_k)^T \in \mathbb{R}^2$, $k = 0, 1, 2$

y sea T_R el triángulo de referencia cuyos vértices son $\mathbf{u}_0 = (0,0)^T$, $\mathbf{u}_1 = (1,0)^T$ y $\mathbf{u}_2 = (0,1)^T$. Si elegimos $\mathbf{x}_0$ como vector de traslación, el mapeo afín que va de T_R a T es $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, donde A es la matriz jacobiana del mapeo afín referente al nodo $\mathbf{x}_0$, y se expresa como $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0)$.

Consideremos que T_I es nuestro triángulo ideal y cuyos vértices son $\mathbf{v}_0$, $\mathbf{v}_1$ y $\mathbf{v}_2$. Si tomamos $\mathbf{v}_0 = (0,0)^T$, el mapeo lineal que va de T_R a T_I es $\mathbf{v} = W\mathbf{u}$, donde $W = (\mathbf{v}_1, \mathbf{v}_2)$ es su matriz Jacobiana. Como las mallas paramétricas y físicas son topológicamente idénticas, cada triángulo en el espacio físico tiene su equivalente en el espacio paramétrico.

El mapeo afín que va de T_I a T viene dado por $\mathbf{x} = AW^{-1}\mathbf{v} + \mathbf{x}_0$, y su matriz Jacobiana es $S = AW^{-1}$. Notar que esta matriz S depende del nodo elegido como nodo de referencia, por tanto, siempre se debe tomar el mismo para T y T_I. Para un triángulo T se puede definir medidas de calidad en términos de la matriz S. Por ejemplo, el mean ratio, $q = \frac{2\sigma}{\|S\|^2}$, es una medida algebraica de calidad de T fácil de calcular, donde $\sigma = \det(S)$ y $\|S\|$ es la norma de Frobenius de S. El valor máximo de q es 1 y se alcanza cuando $A = \mu RW$, donde μ es un escalar y R es una matriz de rotación. En otras palabras, q alcanza su máximo si T y T_I son similares. Además, cualquier triángulo aplastado tiene calidad 0. A partir de esta medida de calidad podemos definir nuestra función objetivo. Sea $\mathbf{x} = (x,y)^T$ la posición del nodo libre, y sea S_m la matriz Jacobiana del triángulo m-ésimo conectado al nodo libre. Definimos la función objetivo de $\mathbf{x}$, asociada al triángulo m como

$$\eta_m = \frac{\|S_m\|^2}{2\sigma_m} \tag{2.1}$$

La función objetivo utilizada para mejorar la calidad de la malla se define en términos de la inversa del mean ratio de cada triángulo de la malla local. La función a minimizar viene dada por

$$K(\mathbf{x}) = \sum_{m=1}^{M} \frac{\|S_m\|^2}{2\,\sigma_m} \tag{2.2}$$

donde M es el número de triángulos en la malla local y S_m es la matriz Jacobiana asociada al mapeo afín desde el triángulo ideal al físico.

La función objetivo definida por la Eq. 2.2 es apropiada para mejorar la calidad de una malla válida, pero no funciona correctamente cuando hay elementos invertidos [63, 64]. En trabajos previos del grupo [55, 65] se propuso una función objetivo modificada K^*, donde el desenredo y suavizado se lleva a cabo en la misma etapa. Esta función objetivo modificada K^* no tiene singularidades, se comporta de forma similar a la función original K para los elementos válidos y tiende a desenredar los elementos invertidos y degenerados. Esta función objetivo se define como

$$K^*(\mathbf{x}) = \sum_{m=1}^{M} \frac{\|S_m\|^2}{2h(\sigma_m)} \tag{2.3}$$

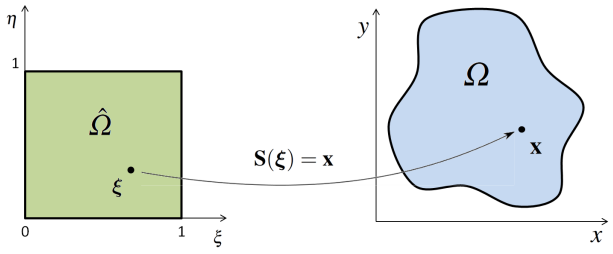


Figura 2.5. Mapeo global uno a uno entre el dominio paramétrico $\hat{\Omega}$ y el físico Ω.

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

La función K^* es suave en todo $\mathbb{R}^2$, por lo que el problema de optimización sin restricciones se puede resolver fácilmente mediante cualquier método estándar (ver por ejemplo [66]).

2.4 Construcción de una representación T-spline de la geometría

En esta sección se describe la construcción de la representación T-spline 2D de la geometría. Para ello tenemos que obtener una transformación paramétrica global que mapee el dominio paramétrico en el físico $\mathbf{S} : \hat{\Omega} = [0,1]^2 \rightarrow \Omega$, ver Fig. 2.5.

Construimos la representación de una superficie T-spline bivariada de nuestro dominio físico mediante combinación lineal de funciones T-spline

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha(\boldsymbol{\xi}) \tag{2.4}$$

donde $\mathbf{P}_\alpha \in \mathbb{R}^2$ es el punto de control correspondiente a la función de base α-th. En este caso las funciones T-spline que utilizamos son B-spline racionales definidas como

$$R_\alpha(\boldsymbol{\xi}) = \frac{N_\alpha(\boldsymbol{\xi})}{\sum_{\beta \in A} N_\beta(\boldsymbol{\xi})} \tag{2.5}$$

siendo $N_\alpha(\boldsymbol{\xi}) = N^1_\alpha(\xi) N^2_\alpha(\eta)$ la B-spline bivariada definida sobre su knot vector local $\Xi_\alpha = \{\Xi_\alpha, H_\alpha\}$, y A es el índice del conjunto de bases definidas sobre la T-mesh. Los puntos de control $\mathbf{P}_\alpha$ se determinan imponiendo condiciones de interpolación. Para ello necesitamos tantos puntos de interpolación como funciones base.

Como puntos de interpolación utilizamos los anclas de las funciones T-spline. Cada ancla coincide con un vértice de la T-mesh, $\boldsymbol{\xi}^v_\alpha$, y su posición en el espacio físico, $\mathbf{x}^v_\alpha$, ha sido determinado mediante el proceso de optimización de la malla.

En nuestro caso estamos trabajando con una estructura de open knot vector a lo largo del contorno de la malla, por lo que el número total de funciones definidas en la

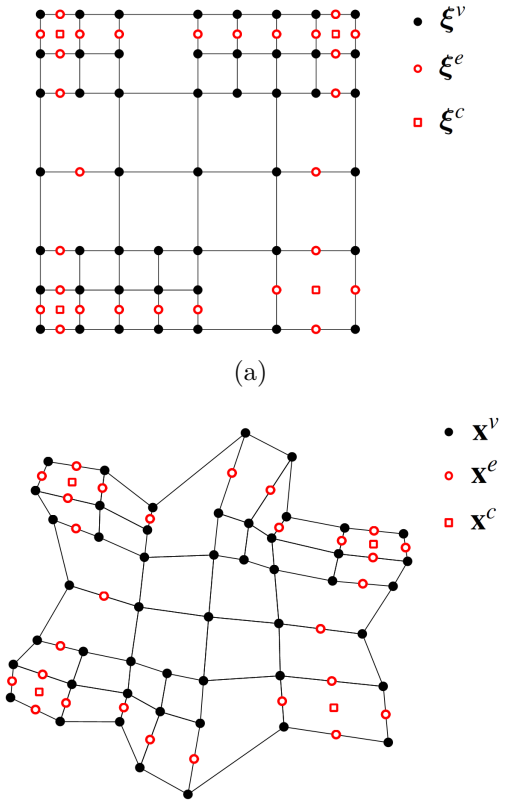
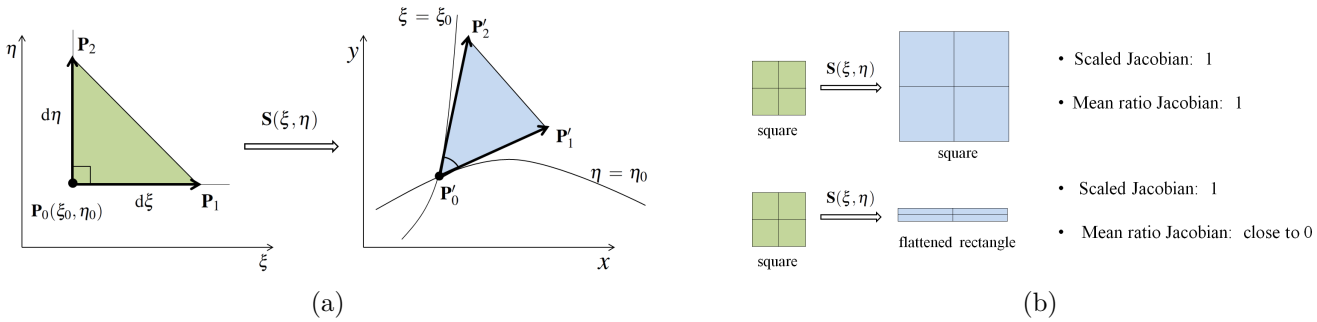


Figura 2.6. Puntos de interpolación en los dominios paramétrico y físico (a) Dominio paramétrico, $\boldsymbol{\xi}^v$ - vétices de la mala, $\boldsymbol{\xi}^e$ - puntos de interpolación adicionales situados en las aristas de la malla, $\boldsymbol{\xi}^c$ - puntos adicionales situados en las cuatro celdas de las esquinas. (b) Puntos de interpolación en el dominio físico.

malla es mayor al número de nodos en la T-mesh. Por tanto tenemos que definir algunos puntos de interpolación adicionales. Estos puntos adicionales se asocian a las funciones cuyos knot vectors Ξ_α o H_α contiene exactamente tres knots repetidos. Para cada una de esa funciones asignamos un punto de interpolación situado aproximadamente donde la función alcanza su máximo. La posición de estos puntos de interpolación en el espacio paramétrico y físico se muestra en la Fig. 2.6(a). Para cada punto adicional $\boldsymbol{\xi}_\alpha$, debemos definir su posición en el espacio físico, $\mathbf{x}_\alpha$. Los puntos de interpolación $\boldsymbol{\xi}^e_\alpha$, situados en el punto medio de una arista del espacio paramétrico se mapean al punto medio de la misma arista en el espacio físico. La posición en el espació físico de los cuatro puntos $\boldsymbol{\xi}^c_\alpha$ situados en los centros de las celdas de las esquinas del espacio paramétrico se determina optimizando la malla local que se forma tras una partición quadtree ficticia de esas celdas.

Finalmente se resuelve el sistema lineal de ecuaciones



(a) (b)

Figura 2.7. Mean ratio Jacobian. (a) Medida de calidad del mapeo paramétrico $\mathbf{S}$ en un punto $\mathbf{P}_0$ en términos del mean ratio del triángulo $\mathbf{P}'_0\mathbf{P}'_1\mathbf{P}'_2$. (b) Comparación entre el mean ratio Jacobian y el scaled Jacobian.

$$\mathbf{x}_\beta = \mathbf{S}\left(\boldsymbol{\xi}_\beta\right) = \sum_{\alpha \in A} \mathbf{P}_\alpha R_\alpha\left(\boldsymbol{\xi}_\beta\right), \qquad \forall \boldsymbol{\xi}_\beta,\ \beta \in A \tag{2.6}$$

donde $\boldsymbol{\xi}_\beta$ son los puntos de interpolación en el espacio paramétrico y $\mathbf{x}_\beta$ son sus imágenes en el espacio físico.

2.5 Refinamiento adaptativo para mejorar la calidad

2.5.1 Mean ratio Jacobian

Nuestro objetivo es conseguir una parametrización geométrica de alta calidad válida para su aplicación en Análisis Isogeométrico. La parametrización T-spline de la Eq. 2.4 es válida para el análisis si tiene Jacobiano positivo en todo el dominio. Una alta distorsión de la geometría puede producir un gran variación del Jacobiano, lo cual puede provocar poca precisión en los resultados numéricos. Por tanto, es deseable conseguir una buena uniformidad y ortogonalidad de las curvas isoparamétricas de la transformación $\mathbf{S}$. Obtener una T-mesh de alta calidad es una condición necesaria, pero no suficiente, para una buena calidad de la transformación T-spline. Puede ocurrir que el Jacobiano de la parametrización spline tome valores negativos incluso siendo válidas todas las celdas de la T-mesh. Para medir la calidad de la parametrización T-spline analizamos el mean ratio Jacobian dado por

$$q_s(\boldsymbol{\xi}) = \frac{2 \det\left(\mathbf{J}_S\right)}{\|\mathbf{J}_S\|^2}, \tag{2.7}$$

donde $\mathbf{J}_S$ es la matriz Jacobiana del mapeo $\mathbf{S}$ en el punto $\boldsymbol{\xi} = (\xi, \eta)$ y $\|\mathbf{J}_S\|$ es su norma de Frobenius.

El valor del mean ratio Jacobian en un punto $\mathbf{P}_0$ del dominio paramétrico es una medida de calidad para el triángulo infinitesimal formado por las dos curvas isoparamétricas del dominio físico que pasan por el punto $\mathbf{P}'_0 = \mathbf{S}(\mathbf{P}_0)$, como se ilustra en Fig.

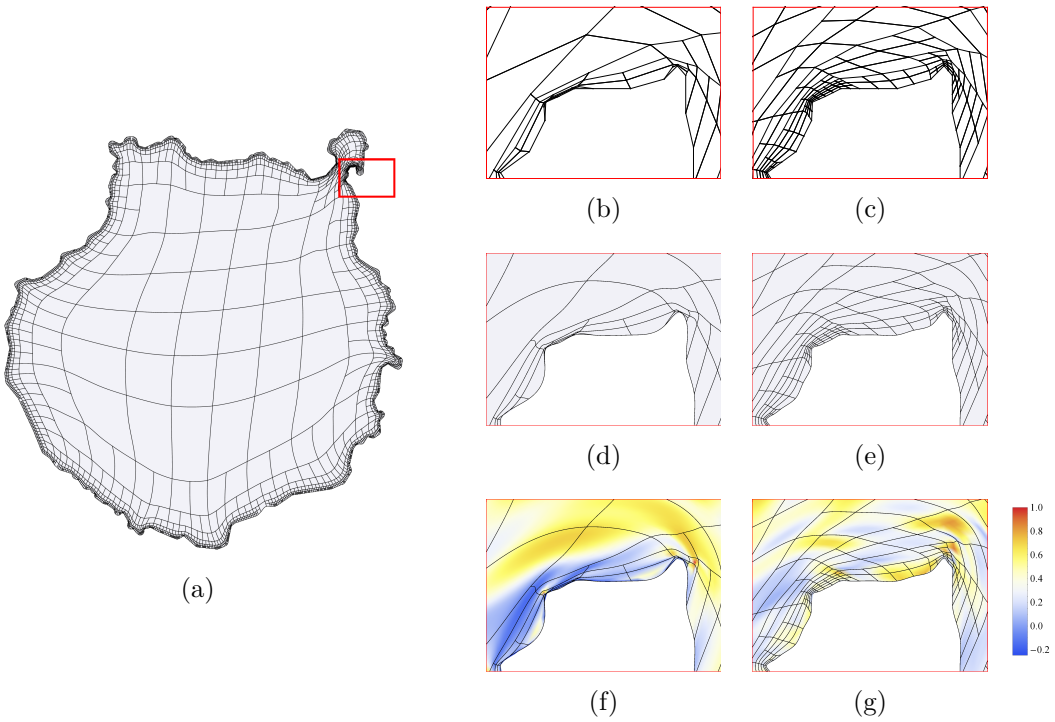


Figura 2.8. Estrategia de refinamiento adaptativo para mejorar la calidad de la transformación en el dominio isla Gran Canaria. (a) Representación T-spline del dominio. (b) T-mesh física inicial. (c) T-mesh física final. (d) Parametrización T-spline inicial con Jacobiano negativo. (e) Parametrización T-spline resultante sin Jacobiano negativo tras aplicar refinamiento adaptativo. (f) Mean ratio Jacobian de la parametrización inicial. (g) Mean ratio Jacobian de la parametrización final.

2.7(a). Al contrario del scaled Jacobian, el cual representa la calidad del mapeo $\mathbf{S}$ en el sentido de la ortogonalidad de sus curvas isoparamétricas, el mean ratio Jacobian representa la calidad del mapeo en el sentido tanto de la ortogonalidad como uniformidad de sus curvas isoparamétricas. La figura 2.7(b) muestra una comparación entre el scaled Jacobian y el mean ratio Jacobian. El scaled Jacobian alcanza su máximo valor 1 en un punto si el mapeo conserva ortogonalidad de las curvas isoparamétricas. En cambio el mean ratio Jacobian es igual a 1 en el punto $\mathbf{P}_0$ si el mapeo conserva ortogonalidad y produce la misma distorsión en ambas direcciones paramétricas, esto es, el mapeo es conforme en ese punto.

Es fácil ver que $\forall \boldsymbol{\xi}: \quad 0 \leq |q_s(\boldsymbol{\xi})| \leq |q_{sc}(\boldsymbol{\xi})| \leq 1$, donde $q_{sc} = \frac{\det(\mathbf{J}_S)}{\|\mathbf{S}_\xi\| \|\mathbf{S}_\eta\|}$ es el scaled Jacobian.

2.5.2 Refinamiento adaptativo

La parametrización de geometrías complejas conlleva una alta distorsión, lo cual puede hacer que aparezca celdas de baja calidad, o incluso celdas con Jacobiano negativo. Esto puede ocurrir por la ausencia de grados de libertad en determinadas zonas. En estos casos, para mejorar la calidad de la malla se lleva a cabo una estrategia adaptativa que refina las celdas de baja calidad. Una idea similar fue desarrollada en [67].

Para cada celda de la malla se calcula el mean ratio Jacobian de la transformación T-spline en puntos de cuadratura de Gauss. En este caso usamos $16 = 4 \times 4$ puntos de cuadratura por celda. Una celda se marca para ser refinada si en al menos un punto de cuadratura el mean ratio Jacobian es inferior a cierto umbral δ. La malla refinada se vuelve a optimizar y el proceso se repite hasta que se alcanza una calidad satisfactoria. La figura 2.8 ilustra la eficacia de la estrategia, donde se aplican refinamientos adicionales en la geometría isla Gran Canaria con $\delta = 0.2$. La malla inicial con 3439 celdas produce una parametrización T-spline con baja calidad en algunas áreas y con Jacobiano negativo en el la parte noreste de la isla. Tras el refinamiento adaptativo se obtiene una malla con 3577 celdas y Jacobiano positivo en todo el dominio. Además, el valor mínimo del mean ratio Jacobian en los puntos de cuadratura es 0.21.

2.6 Parametrización de geometrías anidadas

El método propuesto se puede extender fácilmente para la parametrización de geometrías anidadas. Las geometrías anidadas se forman mediante la inserción de geometrías individuales en otras. Cada una de esas geometrías se puede parametrizar de forma individual con el método visto hasta ahora, obteniendo la malla paramétrica y física de cada una. La clave para parametrizar geometrías compuestas es construir un espacio paramétrico global basado en los espacios paramétricos individuales. En nuestro método, el espacio paramétrico tiene estructura quadtree. Dada su estructura de árbol, la celda de un quadtree puede ser la raiz de un nuevo quadtree construido a partir de esa celda. Esto permite la inserción del quadtree de una geometría interna en una celda o en un conjunto de celdas pertenecientes al quadtree de la geometría externa.

Tras la inserción obtenemos un nuevo quadtree global. Este quadtree puede no estar balanceado, por lo que es necesario aplicar un algorimto de balanceo tras el proceso de inserción. Luego se realiza la etapa de desenredo y suavizado, comenzado con las posiciones de los nodos obtenidas en la construcción individual de las T-meshes de cada geometría. En esta etapa, los nodos del contorno de la geometría del interior se fijan y no se optimizan, igual que la frontera de la geometría externa. La figura 2.9 describe el proceso de construcción de una T-mesh para geometrías anidadas. El resto del proceso para construir la representación T-spline y el refinamiento adaptativo se realiza exactamente de la misma forma que para una geometría individual.

Con esta estrategia es posible insertar cualquier número de geometrías dentro de otra. Además, este método también nos permite construir geometrías con agujeros,

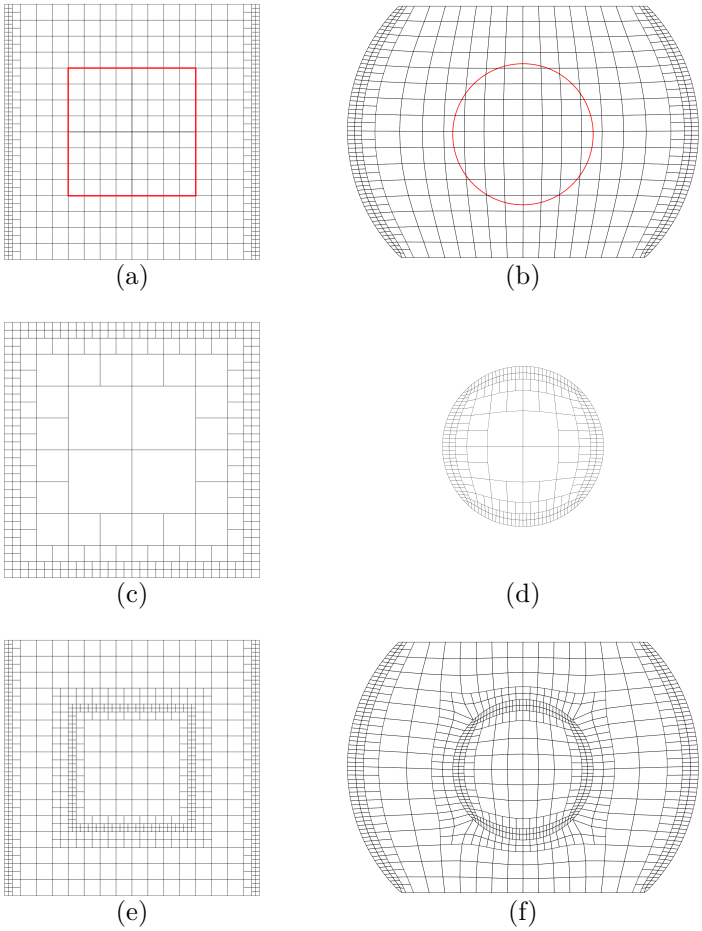


Figura 2.9. Construcción de T-mesh para una geometría anidada. (a) Espacio paramétrico de la geometría exterior. Las líneas rojas definen el área donde será insertado el espacio paramétrico del la geometría interior. (b) T-mesh de la geometría exterior. La posición de la frontera de la geometría interior está marcada en rojo. (c) Espacio paramétrico de la geometría interior. (d) T-mesh de la geometría interior. (e) Espacio paramétrico resultante tras insertar el interior en la región del exterior y aplicar un balanceo 2:1. (f) T-mesh de la geometría anidada tras la optimización.

simplemente eliminando las celdas correspondientes a la geometría interior.

2.7 Conclusiones y siguientes pasos

Se ha propuesto una nueva técnica para obtener una parameterización T-spline de geometrías 2D para su uso en el Análisis Isogeométrico. Se emplea un nuevo procedimiento de desenredo y suavizado de T-mesh con la finalidad de definir una transformación isomórfica entre la T-mesh paramétrica y física. La estrategia que proponemos es sencilla y fácil de implementar. El algoritmo se ha probado en varios tipos de geo-

metrías 2D como se puede ver en el capítulo Resultados y Aplicaciones. En todos los casos hemos obtenido una transformación T-spline de alta calidad. Para medir la calidad de la transformación evaluamos su mean ratio Jacobian. Además, se detectan las zonas con baja calidad y se realiza refinamientos adaptativos para aumentar los grados de libertad en las zonas con alta distorsión. Esta estrategia permite obtener una parametrización válida para el análisis sin Jacobiano negativo, incluso en geometrías complejas. Además, se propone una extensión del método para parametrizar geometrías anidadas.

En los siguientes capítulos se propone una generalización del método para parametrizar geometrías 2D y 3D. Para ello, el principal reto es desarrollar un procedimiento de optimización para T-meshes en 3D.

3. Estrategia para la construcción de espacios spline polinómicos

En este capítulo se describe de forma resumida otro resultado de nuestro grupo: una estrategia para construir espacios spline polinómicos sobre T-meshes jerárquicas con estructura quadtree/octree. Para obtener una convergencia apropiada en el análisis, las funciones spline utilizadas deben tener ciertas características: independencia lineal, propiedad de poder reproducir polinomios, soporte local y posibilidad de realizar refinamiento adaptativo. Este problema se estudió en profundidad en otra tesis doctoral del grupo [51]. Aquí se describe brevemente la estrategia desarrollada para construir espacios spline con propiedades adecuadas para el análisis. Para una descripción más detallada ver [52]. Estas funciones base serán utilizadas en la siguiente sección para parametrizar la geometría.

3.1 Principales pasos de la estrategia

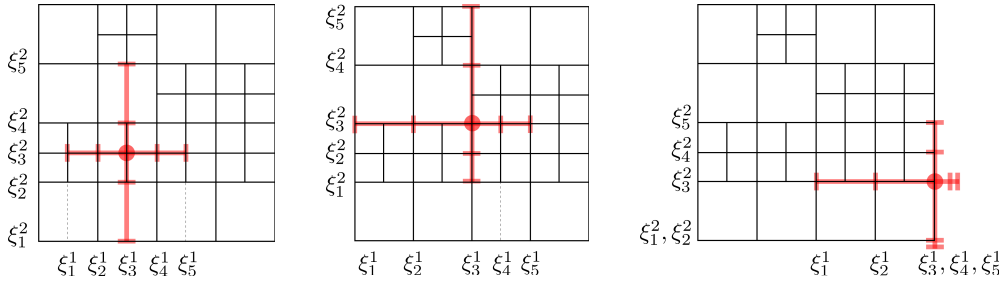
La estrategia que proponemos tiene algunas similitudes con T-splines ya que las funciones de base se definen a partir de su vector de knots local, el cual se infiere atravesando las aristas de la T-mesh. En nuestra estrategia, además imponemos algunas reglas adicionales sobre los vectores de knot locales con la finalidad de obtener espacios spline con las propiedades deseadas. En nuestra estrategia, el proceso de construcción del espacio spline para una T-mesh dada se puede dividir en las siguientes tres etapas:

1. Pretratamiento de la malla (balanceo)
2. Inferencia de vectores de knot locales
3. Modificación de los vectores de knots

3.2 Pretratamiento de la malla

La estrategia que proponemos está diseñada exclusivamente para T-meshes 0 balanceadas. Una malla con estructura tipo tree está 0 balanceada si para cualquier k, ninguna celda de nivel k comparte un vértice (0-cara) con una celda de nivel mayor a $k+1$. En otras palabras, un malla quadtree 0-balanceada implica que todas las celdas tienen contacto (vértice, arista o cara) sólo con celdas que difieren como mucho en un nivel de profundidad. Para obtener un quadtree 0 balanceado se puede aplicar cualquier algoritmo estándar de balanceo. Es importante resaltar que los refinamientos llevados a cabo durante el 0 balanceo no se propagan, ver [68].

Es necesario destacar que el 0 balanceo de la T-mesh es un pre-requisito esencial para la construcción de espacios spline utilizando nuestra técnica. En general, si la T-mesh no está 0 balanceada, nuestras reglas para inferir los vectores de knots no conducen a espacios polinómicos. Además, es importante resaltar que que para nuestras 2D (3D) T-meshes, la subdivisión de cualquier celda se realiza mediante la subdivisión de la celda en 4 (8) subceldas iguales, de modo que todas las celdas del mismo nivel tienen en mismo tamaño y el tamaño de arista de una celda de nivel k is dos veces más grande que el tamaño de arista de una celda de nivel $k+1$.



(a) No T-junction is skipped.

(b) Two T-junctions are skipped in each direction.

(c) Two T-junctions are skipped and the knots are repeated when the boundary is reached.

Figura 3.1. Inferencia de vectores de knots locales para una función bivariada, atravesando las aristas de la T-mesh.

3.3 Inferencia de vectores de knots locales

Consideremos una T-mesh T del espacio paramétrico cuadrado $\Omega = [0,1]^d$, $d = 2$ o 3. Llamamos nodo regular al nodo que no forma una unión en T. Vamos a asociar funciones de base sólo a los nodos regulares de la malla, de forma análoga a como se hace en el método de los elementos finitos cuando se trabaja con hanging nodes. El esqueleto de una malla T de dimensión d consiste en el conjunto geométrico de puntos compuesto por la unión de todas las $(d-1)$ caras de la malla y se denota como $\mathbf{skt}(T)$. En el caso 2D, el esqueleto de la malla sería la unión de todas las aristas de la malla, y en el caso 3D es la unión de todas las caras.

Para definir nuestras funciones spline de base sobre una T-mesh de dimensión d, un vector de knots local para cada dirección paramétrica d debe ser asignado a cada función N_α: $\Xi_\alpha^j = \left(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j, \xi_5^j\right)$, $j = 1, ..., d$. De forma similar a [28], estos vectores de knots se infieren atravesando el esqueleto de la T-mesh. Por simplicidad, vamos a describir este proceso para una malla bidimensional. Comenzando desde el knot central (ξ_3^1, ξ_3^2), es decir, el ancla de la función, caminamos a través de la T-mesh hasta intersectar perpendicularmente una arista de la malla. Según nuestra estrategia, debemos saltar las uniones en T donde la arista ausente es perpendicular a la dirección de nuestra marcha, ver figura 3.1. Cuando se alcanza la frontera del dominio paramétrico, repetimos los knots creando así una estructura de vectores de knots abiertos a lo largo de la frontera, ver figura 3.1(c). Hay que tener en cuenta que todos los knots interiores tienen multiplicidad 1. De este modo obtenemos un conjunto de funciones de base $\{N_\alpha\}_{\alpha \in A_T}$, siendo A_T el conjunto de índices. La figura 3.2 muestra un ejemplo de T-mesh en el espacio paramétrico y los anclas de todas la funciones de base definidas sobre esta malla. Todos los nodos regulares del interior tienen exactamente una función asociada, pero a los nodos del contorno se les asocia más de una función debido a la estructura de vector de knots abierto.

El proceso de inferencia de los vectores de knots locales se puede resumir de la siguiente forma:

- Las funciones de base se asocian sólo a los nodos regulares de la malla.
- Los vectores de knots locales se infieren caminando a través de la malla hasta intersectar perpendicularmente el esqueleto de la malla. Esta intersección no debe coincidir con una unión en T perpendicular a la dirección de marcha.
- Los knots se repiten en el contorno para crear una estructura de vectores de knots abiertos a lo largo de la frontera.

Luego se modifica el soporte de algunas funciones para generar un espacio spline con buenas propiedades.

3.3.1 Modificación de los vectores de knots locales

La clave de nuestra estrategia recae en un conjunto de sencillas reglas para la modificación de los soportes de las funciones que conlleva a la construcción de espacios spline polinómicos sobre una T-mesh 0 balanceada dada. Para describir la idea, primero vamos a introducir alguna notación. Para los vectores de knots locales $\Xi_\alpha^j = \left(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j, \xi_5^j\right)$, $j = 1,...,d$ denominamos la longitud de cada intervalo de knots como $\Delta_i^j = \xi_{i+1}^j - \xi_i^j$, $j = 1,...,d$ y $i = 1,...,4$.

El soporte de un función de base d-variada N_α is un rectángulo d-dimensional box: $[\xi_1^1, \xi_5^1] \times \cdots \times [\xi_1^d, \xi_5^d]$. Llamaremos frame del soporte de una función a la unión de todas las $(d-2)$-caras de su box y lo denotaremos como $\mathrm{frm}(\mathrm{supp}\, N_\alpha)$. Es decir, para el soporte rectangular de una función bivariada, el frame es la unión de los cuatro vértices de su rectángulo. Para el soporte cúbico de una función trivariada, su frame está compuesto por la unión de las doce aristas de ese cubo.

Una vez que se ha inferido el soporte de la función, éste se modifica de modo que para cada función de base N_α sus nuevos vectores de knots Ξ_α^j $j = 1,...,d$ verifican las siguientes condiciones:

Condición 1: Vector de knots local de la función d-variada N_α verifica[1]

$$\Delta_1^j \geqslant \Delta_2^j = \Delta_3^j \leqslant \Delta_4^j, \quad j = 1,...,d, \tag{3.1}$$

Condición 2: El frame del soporte de la función debe estar situado sobre el esqueleto de la malla:

$$\mathrm{frm}(\mathrm{supp}\, N_\alpha) \in \mathrm{skt}(T). \tag{3.2}$$

Por lo tanto, los soportes de función que no cumplan las Condiciones 1 y 2 deben ser modificados. Para ello, el soporte original se extiende modificando algunos intervalos de

[1] Excepto los casos donde se repiten knots, los cuales se explican al final de la Sección 3.4.1.

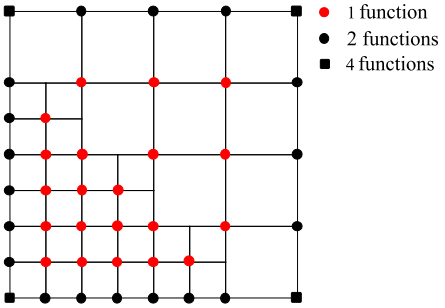


Figura 3.2. Ejemplo de T-mesh con sus anclas. Los círculos rojos representan los nodos del interior que tienen una función de base asociada, los círculos negros son los nodos del contorno que tienen dos funciones de base y los cuadrados negros son los nodos del contorno que tienen 4 funciones debido la estructura de vectores de knots abiertos a lo largo de la frontera.

knots hasta que el soporte resultante satisface ambas condiciones. Nos vamos a referir a estas modificaciones de soporte como Reglas de extensión 1 y 2 respectivamente.

En la siguiente sección se explica de forma detallada este proceso para los casos 2D y 3D.

3.4 Reglas de modificación de soportes

Aquí mostramos las reglas de extensión de soportes para obtener vectores de knots locales que cumplan las Condiciones 1 y 2 formuladas en la sección anterior. El proceso es el siguiente. Primero, si tras atravesar el esqueleto de la T-mesh, el vector de knots local de una función no cumple la Condición 1, algunos de sus knots se modifican para que la cumplan. Luego, se comprueba que cumpla la Condición 2, y si no es así, se realiza otra modificación de los knots. Como resultado de estas modificaciones, se obtiene un nuevo soporte extendido con vectores de knots locales que verifican ambas condiciones. Estas modificaciones se pueden implementar fácilmente debido a la estructura de árbol balanceado de nuestra malla. Veamos este proceso en detalle.

Para aclarar la notación, en adelante denotaremos las coordenadas paramétricas como (ξ,η,ζ), las cuales corresponden a $(\xi^1,\xi^2,\xi^3)=(\xi,\eta,\zeta)$ en la notación previa. Consecuentemente, $(\Xi^1,\Xi^2,\Xi^3)=(\Xi,\mathscr{H},\mathscr{Z})$ y $(\Delta_i^1,\Delta_i^2,\Delta_i^3)$ $=(\Delta_i^\xi,\Delta_i^\eta,\Delta_i^\zeta)$.

3.4.1 Extensión de soportes para mallas 2D

El esqueleto $\mathbf{skt}(T)$ de una malla T de dimensión 2 es la unión de todas las aristas de la malla. Para una función bivariada llamaremos a los vértices de su soporte rectangular como $V_{1,1}=(\xi_1,\eta_1)$, $V_{5,1}=(\xi_5,\eta_1)$, $V_{5,5}=(\xi_5,\eta_5)$ y $V_{1,5}=(\xi_1,\eta_5)$. Luego, el frame

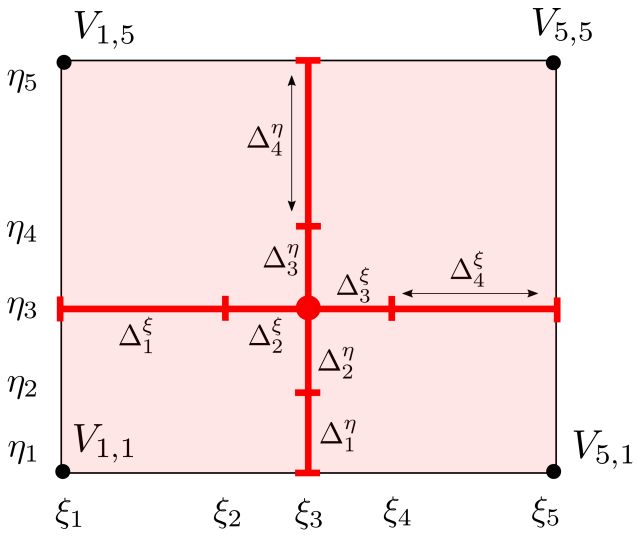


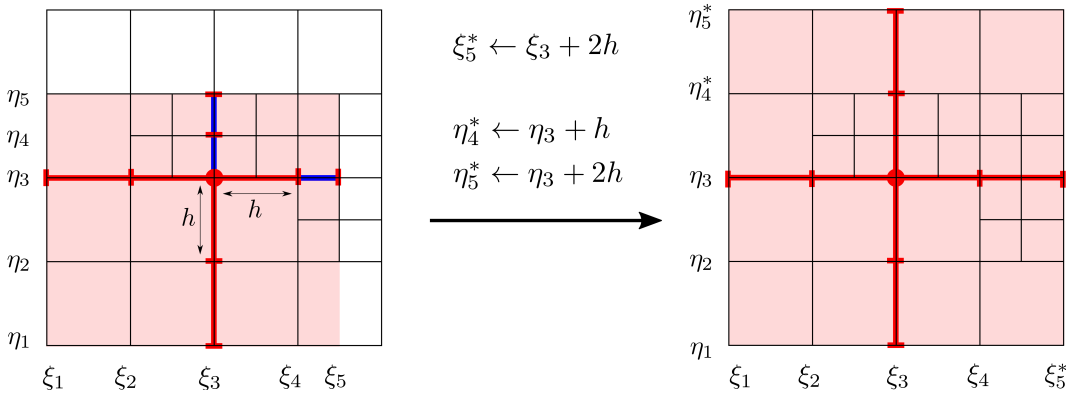
Figura 3.3. Notación de los soportes para el caso 2D.

del soporte de una función la unión de sus cuatro vértices, esto es, $\mathrm{frm}(\mathrm{supp}\, N_\alpha) = \{V_{n,m},\, n,m \in \{1,5\}\}$. La figura 3.3 muestra esta notación para el soporte de una función bivariada.

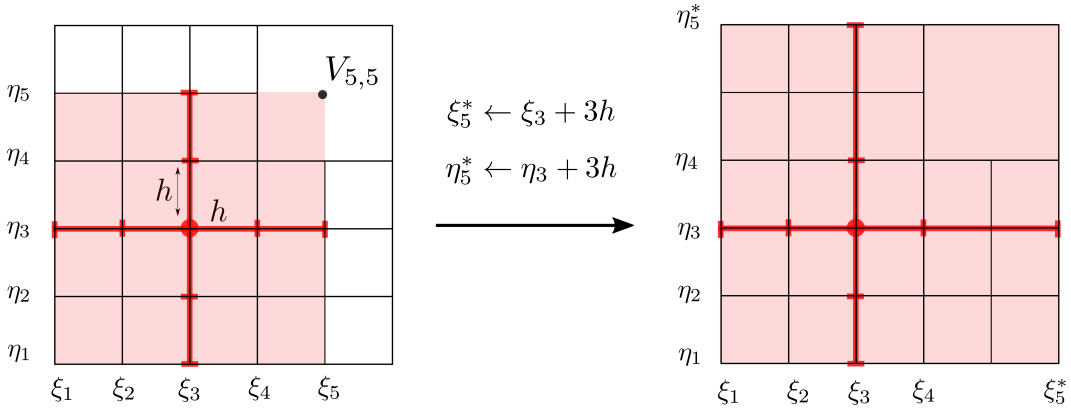
La formulación de la Condición 1 para los vectores de knots locales Ξ y $\mathscr{H}$ es sencillo y no requiere aclaración. En cuanto a la Condición 2 para el caso 2D se formula de la siguiente forma: los cuatro vértices del soporte de la función deben estar situados sobre las aristas de la malla.

Regla de extensión 1. Si el vector de knots local Ξ de una función no satisface la Condición 1, modificamos este vector saltando el número mínimo de knots hasta que se verifica $\Delta_1^\xi \geqslant \Delta_2^\xi = \Delta_3^\xi \leqslant \Delta_4^\xi$, y análogamente, para $\mathscr{H}$. Esta modificación se hace de forma independiente para cada dirección paramétrica aplicando determinadas reglas de extensión. Veamos un ejemplo de extensión de soporte para una función bivariada. El soporte de función de más a la izquierda mostrado en la figura 3.4(a) no satisface la Condición 1. Para el vector de knots Ξ tenemos $\Delta_3^\xi > \Delta_4^\xi$, por lo que el intervalo de knots Δ_4^ξ debe ser modificado. Llamemos $h = \max(\Delta_2^\xi, \Delta_3^\xi) = \max(\Delta_2^\eta, \Delta_3^\eta)$. Destacar que ambos máximos coinciden debido a la estructura quadtree y a que se salta las uniones en T. Luego, el quinto knot ξ_5 se redefine como $\xi_5^* \leftarrow \xi_3 + 2h$. Para el vector de knots $\mathscr{H}$ tenemos $\Delta_2^\eta > \Delta_3^\eta$, por lo que los knots η_4 y η_5 se deben modificar como $\eta_4^* \leftarrow \eta_3 + h$, $\eta_5^* \leftarrow \eta_3 + 2h$.

Regla de extensión 2. Una vez se cumple la Condición 1, para la Condición 2 comprobamos si los vértices del soporte de la función están situados sobre las aristas de la malla. Si no es así, modificamos los vectores de knots saltando un knot en ambas direcciones paramétricas situando así el vértice sobre las aristas de la malla. La figura 3.4(b) muestra este proceso. La comprobación se realiza de forma independiente para cada uno de los cuatro cuadrantes del soporte de la función. Hay que destacar que para



(a) No se cumple la Condición 1 porque $\Delta_4^\xi < \Delta_3^\xi$ y $\Delta_3^\eta < \Delta_2^\eta$.



(b) No se cumple la Condición 2 porque $V_{5,5} \notin skt(T)$.

Figura 3.4. Reglas de extensión. (a) Ejemplo de modificación de soporte según la Regla de extensión 1. (b) Un ejemplo de modificación de soporte según la Regla de extensión 2.

nuestro quadtree 0-balanceado sólo necesitamos hacer esta comprobación para algunas funciones. Por ejemplo, sin perder generalidad, el vértice del soporte $V_{5,5} = (\xi_5, \eta_5)$ se debe comprobar sólo si $\Delta_3^\xi = \Delta_4^\xi = \Delta_3^\eta = \Delta_4^\eta$. En la figura 3.4(b) se muestra un ejemplo de soporte de función que no cumple la condición 2. El vértice $V_{5,5} = (\xi_5, \eta_5)$ de este soporte no está situado sobre las aristas de la malla, por lo que el quinto knot en ambas direcciones paramétricas debe ser redefinido como $\xi_5^* \leftarrow \xi_3 + 3h, \quad \eta_5^* \leftarrow \eta_3 + 3h$, y así, el nuevo vértice $V_{5,5}$ se sitúa sobre las aristas de la malla.

Para otros soportes de funciones, la extensión es completamente análoga a estos dos ejemplos. En todos los casos posibles, la extensión del soporte de una función implica cambiar uno o dos intervalos de knots duplicando su tamaño.

Los Algoritmos 1 y 2 muestran de forma detallada las Reglas de extensión para modificar el soporte de una función bivariada acorde a las Condiciones 1 y 2.

Hay que destacar que una excepción para la Condición 1 es un vector de knots

que contiene un intervalo de knots de longitud 0 debido a la estructura de vector de knots abierto a lo largo de la frontera. In este caso, un vector de knots debe cumplir la desigualdad (3.1) sin tener en cuenta los intervalos de knots de longitud 0. Consecuentemente, una excepción para la aplicación de las reglas de extensión es el caso cuando se alcanza la frontera del dominio paramétrico atravesando las aristas de la T-mesh.

Algorithm 1: Regla de extensión 1.

Input: A knot vector $\Xi = (\xi_1, \xi_2, \xi_3, \xi_4, \xi_5)$, $\xi_i \in [0,1]$.

Function Modify1(Ξ)
- $\Xi^* \leftarrow \Xi$
- $h = \max(\Delta_2, \Delta_3)$
- if $\Delta_2 < \Delta_3$ *and* $\xi_2 > 0$ then
  - $\xi_2^* \leftarrow \xi_3 - h$
  - $\xi_1^* \leftarrow \xi_2^*$
  - if $\xi_1^* > 0$ then $\xi_1^* \leftarrow \xi_3 - 2h$
- if $\Delta_1 < \Delta_2$ *and* $\xi_1^* > 0$ then
  - $\xi_1^* \leftarrow \xi_3 - 2h$
- if $\Delta_2 > \Delta_3$ *and* $\xi_4 < 1$ then
  - $\xi_4^* \leftarrow \xi_3 + h$
  - $\xi_5^* \leftarrow \xi_4^*$
  - if $\xi_5^* < 1$ then $\xi_5^* \leftarrow \xi_3 + 2h$
- if $\Delta_4 < \Delta_3$ *and* $\xi_5^* < 1$ then
  - $\xi_5^* \leftarrow \xi_3 + 2h$
- return Ξ^*

Output: A corrected knot vector Ξ^* that satisfies Condition 1.

Algorithm 2: Regla de extensión 2 en 2D.

Input: A 0-balanced mesh T and a pair of local knot vectors $S = \{\Xi, \mathscr{H}\}$.

Function Modify2(T, S)
- $S^* \leftarrow S$
- $h = \max(\Delta_2^{\xi}, \Delta_3^{\xi})$
- for $n \in \{1,5\}$ do
  - for $m \in \{1,5\}$ do
    - if $(\xi_n, \eta_m) \notin skt(T)$ then
      - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
      - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
- return S^*

Output: A modified support $S^* = \{\Xi^*, \mathscr{H}^*\}$ that satisfies Condition 2.

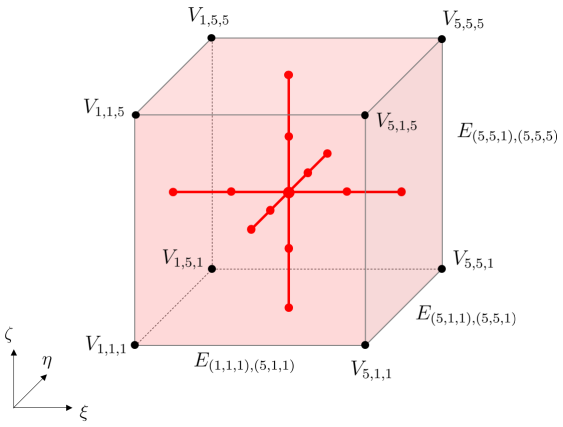


Figura 3.5. Notación del soporte para el caso 3D.

3.4.2 Extensión de soportes para mallas 3D

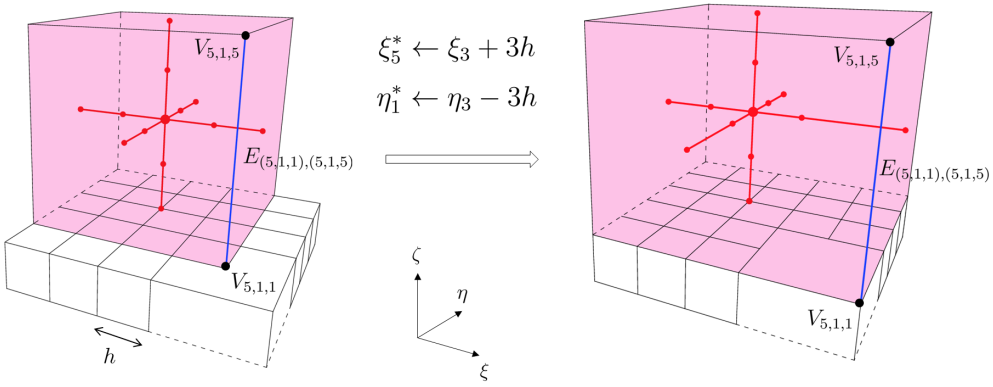
El esqueleto $\mathbf{skt}(T)$ de una malla tridimensional T es la unión de todas las caras de la malla. Para una función trivariada, llamamos a los vértices de su soporte como $V_{n,m,k} = (\xi_n, \eta_m, \zeta_k)$ donde $n,m,k \in \{1,5\}$. Y la arista de un soporte formada por los vértices $V_{n,m,k}$ y $V_{p,q,r}$ se denota como $E_{(n,m,k),(p,q,r)}$. Luego, el frame $\mathbf{frm}(\mathbf{supp}\, N_\alpha)$ del soporte de una función trivariada is la unión de sus 12 aristas. La figura 3.5 muestra la notación introducida para el soporte de una función de base trivariada.

La formulación de la Condición 1 para los vectores de knots locales de un función trivariada es análogo al caso 2D. La Condición 2 adaptada a mallas 3D es la siguiente: las aristas del soporte cuboidal de la función deben estar situadas sobre las caras de la malla.

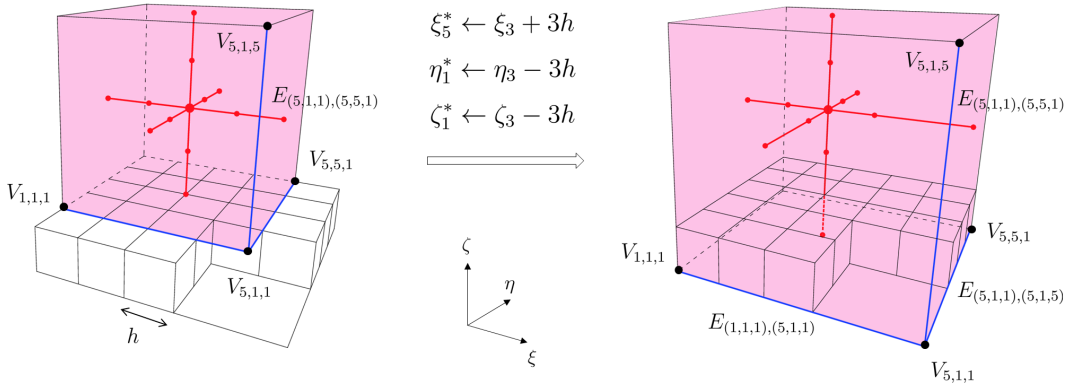
La implementación de la estrategia para 3D es similar al caso 2D. Para satisfacer la Condición 1, se aplica la Regla de extensión 1 a cada uno de los tres vectores de knots de una función, análogamente al caso 2D usando el Algoritmo 1.

Regla de extensión 2. Para satisfacer la Condición 2, se comprueba si las aristas del soporte de la función están situadas sobre las caras de la malla. Si no es así, se debe modificar los dos vectores de knots perpendiculares a esta arista, saltando un knot en ambas direcciones paramétricas y situando así esta arista sobre las caras de la malla. Esta comprobación se realiza de forma independiente para cada uno de los 8 cuadrantes del soporte de la función y, en cada cuadrante, se debe comprobar tres aristas. La figura 3.6 ilustra el procedimiento de extensión de soporte para el cuadrante del vértice $V_{5,1,1}$. Debido a la estructura octree, sólo pueden ocurrir dos casos: (i) el cuadrante contiene una arista que no está situada sobre las caras de la malla o (ii) el cuadrante contiene tres aristas y un vértice que no están situados sobre el esqueleto de la malla. A continuación estudiamos cada caso.

(i) Si un cuadrante contiene una arista que con cumple la Condición 2, entonces se modifican los dos vectores de knots perpendiculares a esta arista, ver figura 3.6(a). Para el soporte de función mostrado en la figura 3.6(a) izquierda, la arista $E_{(5,1,1),(5,1,5)}$



(a) Sólo una arista violando la Condición 2. El nodo $V_{5,1,1}$ se sitúa en el centro de la cara de tamaño $2h$. El soporte se extiende enn dos direcciones.



(b) Tres aristas violando la Condición 2. El nodo $V_{5,1,1}$ se sitúa en el centro de la celda de tamaño $2h$. El soporte se extiende en tres direcciones.

Figura 3.6. Regla de extensión 2 para la modificación del soporte de una función trivariada.

no está situada sobre las caras de la malla. Por tanto, se modifica sus dos vectores de knots Ξ y $\mathscr{H}$ perpendiculares a esta arista de modo que la arista se sitúe sobre las caras de la malla, es decir, los knots ξ_5 y η_1 se redefinen como $\xi_5^* \leftarrow \xi_3 + 3h$ y $\eta_1^* \leftarrow \eta_3 - 3h$, donde $h = \max(\Delta_2^\xi, \Delta_3^\xi) = \max(\Delta_2^\eta, \Delta_3^\eta) = \max(\Delta_2^\zeta, \Delta_3^\zeta)$.

(ii) Si un cuadrante contiene tres aristas que no están situadas sobre las caras de la malla, entonces se modifican los tres vectores de knots saltando un knot en cada dirección paramétrica, ver figura 3.6(b). El vértice $V_{5,1,1}$ y las tres aristas conectadas a él no están situadas sobre el esqueleto de la malla, por lo que se modifican los tres vectores de knots para situar las tres aristas sobre las caras de la malla: $\xi_5^* \leftarrow \xi_3 + 3h$, $\eta_1^* \leftarrow \eta_3 - 3h$, $\zeta_1^* \leftarrow \zeta_3 - 3h$.

El Algoritmo 3 explica la Regla de extensión 2 utilizada para modificar el soporte de una función trivariada acorde a la Condición 2.

Algorithm 3: Regla de extensión 2 en 3D.

Input: A 0-balanced T-mesh T and three local knot vectors $S = \{\Xi, \mathscr{H}, \mathscr{Z}\}$.

Function Modify2 (T, S)
- $S^* \leftarrow S$
- $h = \max(\Delta_2^{\xi}, \Delta_3^{\xi})$
- $mov(i) := i + 4\,\mathrm{sgn}(3-i)$
- for $n \in \{1,5\}$ do
  - for $m \in \{1,5\}$ do
    - for $k \in \{1,5\}$ do
      - if $E_{(n,m,k),(mov(n),m,k)} \notin skt(T)$ then
        - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
        - $\zeta_k^* \leftarrow \zeta_3 + 3h\,\mathrm{sgn}(\zeta_k - \zeta_3)$
      - if $E_{(n,m,k),(n,mov(m),k)} \notin skt(T)$ then
        - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
        - $\zeta_k^* \leftarrow \zeta_3 + 3h\,\mathrm{sgn}(\zeta_k - \zeta_3)$
      - if $E_{(n,m,k),(n,m,mov(k))} \notin skt(T)$ then
        - $\xi_n^* \leftarrow \xi_3 + 3h\,\mathrm{sgn}(\xi_n - \xi_3)$
        - $\eta_m^* \leftarrow \eta_3 + 3h\,\mathrm{sgn}(\eta_m - \eta_3)$
- return S^*

Output: A modified support $S^* = \{\Xi^*, \mathscr{H}^*, \mathscr{Z}^*\}$ that satisfies Condition 2.

3.5 Propiedades de las EP-splines

A las funciones base obtenidas con nuestra estrategia las llamamos EP-splines (Extended Polynomial splines). A continuación se resumen las propiedades y algunas características de los espacios spline construidos con nuestro método.

1. Las funciones $\{N_\alpha\}_{\alpha \in A_T}$ son C^2-continuas.
2. No-negatividad: $N_\alpha \geq 0$.
3. Las funciones $\{N_\alpha\}_{\alpha \in A_T}$ son linealmente independientes.
4. Partición de unidad no-negativa: $\sum_{\alpha \in A_T} c_\alpha N_\alpha = 1$, donde $c_\alpha \geq 0$.
5. Encaje de espacios: $T_1 \subset T_2 \Rightarrow S_{T_1} \subset S_{T_2}$.
6. Capacidad de reproducir polinomios: $\mathbb{P}_3(\Omega) \subset S_T(\Omega)$.

Una discusión detallada de estas propiedades se puede encontrar en [51].

4. Método para la parametrización spline de geometrías 2D y 3D

En este capítulo se presenta un método para obtener una parametrización spline, tanto de geometrías 2D como 3D, para su uso en Análisis Isogeométrico. Este método es una generalización del propuesto en el capítulo 2 para la parametrización T-spline de geometrías 2D. Las etapas del método son similares en ambos casos, por tanto aquí nos centraremos principalmente en describir las nuevas aportaciones.

Se propone un método para obtener la parametrización de un objeto Ω (dominio físico) mediante la deformación de una T-mesh del espacio paramétrico, el cubo unitario $\widehat{\Omega} = [0,1]^d$, $d = 2,3$, hasta que se alcanza la forma del objeto. Esta deformación sólo afecta a la posición de los nodos, es decir, no se producen cambios en sus conectividades: decimos que ambas mallas son isomórficas. La deformación de la malla se realiza mediante un procedimiento de desenredo y suavizado simultáneo basado en una medida puntual de calidad. Esta medida de calidad persigue tanto ortogonalidad como uniformidad de las curvas isoparamétricas. La representación spline del objeto se calcula imponiendo condiciones de interpolación usando la información proporcionada por la correspondencia uno a uno ente la malla paramétrica y la malla física del objeto. Para esta interpolación se utiliza espacios spline polinómicos (EP-splines) construidos con la técnica propuesta en el capítulo 3.

Existen ciertas limitaciones que dificultan la generación de una parametrización 3D global debido a la naturaleza de producto tensorial de las funciones spline. Para dominios complejos es necesario dividir el objeto en parches spline lo que lleva a una pérdida de regularidad en zonas donde se unen dos parches. Dada la superficie de un sólido, no es trivial encontrar una partición óptima en seis parches correspondientes a las seis caras del cubo paramétrico. Además, para garantizar una parametrización regular (con Jacobiano positivo), los parches de la frontera deben unirse en regiones convexas, lo cual puede ser imposible para algunas geometrías.

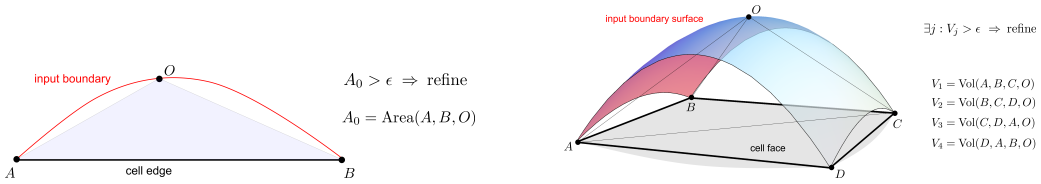
En este trabajo no vamos a tratar todos los problemas inherentes a la parametrización spline. Por tanto, en nuestro estudio nos centraremos en geometrías de género cero dadas por seis parches de la frontera que se unen en regiones convexas.

Este capítulo está organizado de la siguiente forma. En la siguiente sección se describe los pasos principales del método. En la sección 4.2 se explica el proceso de construcción de una T-mesh adaptada a las singularidades de la geometría. Y en la sección 4.3 describe el procedimiento de optimización de T-mesh, que es la clave del método.

4.1 Esquema general del método

Los pasos principales del método son muy similares a los expuestos en el capítulo 2. Aquí se resumen brevemente:

1. Construcción de una T-mesh adaptada: se genera una T-mesh refinando hasta aproximar el contorno de la geometría con una tolerancia dada. Durante este proceso, los nodos del contorno del dominio paramétrico se mapean a la superficie del objeto.



(a) (b)

Figura 4.1. Criterio de error para la aproximación de una T-mesh adaptada. (a) Criterio de error para la aproximación del contorno en 2D. (b) Criterio de error para la aproximación de la superficie del contorno en 3D.

2. Optimización T-mesh: se recolocan los nodos del interior de la T-mesh física aplicando un procedimiento de desenredo y suavizado simultáneo.

3. Construcción de una representación spline de la geometría: se calcula una representación spline del objeto interpolando los datos proporcionados por la transformación entre la malla paramétrica y física del objeto.

4. Refinamiento adaptativo para mejorar la calidad: si la calidad de la transformación spline no es suficiente, se aplica refinamientos adaptativos para aumentar el número de grados de libertad en las zonas de alta distorsión.

Dada su simplicidad, las mallas quadtree/octree son una herramienta atractiva para realizar refinamientos adaptativos tanto en IGA como en modelación geométrica. Para la representación spline del objeto vamos a utilizar espacios EP-spline. Esta estrategia permite definir espacios spline cúbicos con propiedades adecuadas para el análisis sobre una T-mesh quadtree/octree fuertemente balanceada. Por ello, las mallas con las que trabajaremos serán T-meshes quadtree/octree 0-balanceadas.

Como se indicó previamente, el método de parametrización propuesto es válido para geometrías 2D y 3D. Hay que destacar que el método de parametrización 3D es una generalización directa del caso 2D. Para facilitar la explicación del proceso completo, vamos a dar una descripción de las principales etapas del método tanto en 2D como en 3D.

4.2 Construcción de una T-mesh adaptada

En esta sección se describe el proceso para construir una T-mesh adaptada al objecto que represente las características del contorno con una precisión deseada.

4.2.1 Caso 2D

Para dominios planos asumimos que el input viene dado por cuatro curvas spline o cuatro poligonal. Además, se requiere que la unión de las cuatro curvas (poligonal) entre

sí forme regiones convexas, esto es, el ángulo interior formado en la unión es menor a 180°. Si el input viene dado por curvas spline, tomamos cada lado del cuadrado paramétrico como el espacio paramétrico de cada curva, para así poder mapear cada punto del contorno paramétrico. Si el input son poligonal, cada arista del cuadrado paramétrico se mapea a la correspondiente poligonal vía parametrización chord-length.

Sean Γ^i $(i = 1,2,\ldots,4)$ las cuatro que describen la frontera del objeto Ω, y $\widehat{\Gamma}^i$ son los cuatro lados del cuadrado paramétrico $\widehat{\Omega} = [0,1]^2$. Dadas las cuatro parametrizaciones $\Pi_b^i : \widehat{\Gamma}^i \to \Gamma^i$, podemos definir un mapeo paramétrico global entre la frontera del cuadrado $\partial\widehat{\Omega} = \widehat{\Gamma} = \bigcup_{i=1}^4 \widehat{\Gamma}^i$ y la frontera del objeto $\partial\Omega = \Gamma = \bigcup_{i=1}^4 \Gamma^i$:

$$\Pi_b : \widehat{\Gamma} \to \Gamma$$

Luego se construye una T-mesh con estructura quadtree para aproximar el contorno del objeto con una tolerancia predefinida ε. Consideremos que $\widehat{\Omega}_k$ es la T-mesh de $\widehat{\Omega}$ resultante de aplicar varios refinamientos quadtree según determinado criterio de aproximación. Tras mapear $\widehat{\Gamma}_k = \partial\widehat{\Omega}_k$ al espacio físico usando Π_b, obtenemos un nuevo contorno Γ_k que es una aproximación del dato de entrada Γ, es decir:

$$\widehat{\Gamma}_k \to \Pi_b(\widehat{\Gamma}_k) =: \Gamma_k \approx \Gamma$$

Para mejorar la aproximación debemos refinar las celdas de $\widehat{\Omega}_k$ que tienen contacto con la frontera del espacio paramétrico, de forma que la distancia entre Γ_k y Γ disminuye hasta alcanzar la tolerancia prescrita ε. Para medir el error de la aproximación se evalúa el área del triángulo formado por la arista AB de la celda física y la imagen del centro de la arista O mapeada sobre la frontera dato, ver Fig. 4.1(a). La celda se refina si el error es mayor que ε. Cada refinamiento produce un nuevo punto en la frontera del dominio paramétrico el cual se proyecta sobre la frontera real, obteniendo así una aproximación más precisa de la geometría.

4.2.2 Caso 3D

Para las geometrías 3D se asumen que el input viene dado por seis superficies spline, ver Fig. 4.2(a), o por seis triangulaciones, las cuales se unen en regiones convexas. Si las superficies vienen dadas por splines, se toma como espacio paramétrico de cada una las caras del cubo paramétrico. Si el input viene dado por triangulaciones, entonces cada triangulación se parametriza utilizando el método propuesto por M. Floater en [69]. Este método establece un mapeo uno a uno entre una triangulación en el espacio y una triangulación plana en el domino paramétrico, la cual se puede usar para mapear las caras del cubo a la triangulación dato.

Sea Γ^i $(i = 1,2,\ldots,6)$ las seis superficies que describen al objeto Ω, y sea $\widehat{\Gamma}^i$ las caras del cubo paramétrico $\widehat{\Omega} = [0,1]^3$. Dadas la parmetrizaciones $\Pi_b^i : \widehat{\Gamma}^i \to \Gamma^i$, podemos definir un mapeo paramétrico global entre la frontera del cubo $\partial\widehat{\Omega} = \widehat{\Gamma} = \bigcup_{i=1}^6 \widehat{\Gamma}^i$ y el contorno del objeto $\partial\Omega = \Gamma = \bigcup_{i=1}^6 \Gamma^i$:

$$\Pi_b : \widehat{\Gamma} \to \Gamma$$

Se construye una T-mesh octree paramétrica $\widehat{\Omega}_k$ de forma análoga al caso 2D, esto es, $\widehat{\Omega}$ se refina acorde a un criterio de error. Como criterio de error, se evalúa el volumen de los cuatro tetraedros *ABCO*, *BCDO*, *CDAO*, *DABO* construidos en las caras de las celdas que definen el contorno *ABCD*, donde *O* es la imagen del centro de la cara, ver Fig. 4.1(b). La celda se refina si el volumen de alguno de los tetraedros es mayor que ε. El refinamiento de una celda introduce 5 nuevos puntos en la frontera los cuales se proyectan a la superficie del objeto, obteniendo así una aproximación más precisa de la geometría. Luego, tras mapear $\widehat{\Gamma}_k = \partial\widehat{\Omega}_k$ al espacio físico, se obtiene una nueva frontera Γ_k que es una aproximación de la frontera dato Γ, esto es:

$$\widehat{\Gamma}_k \to \Pi_b(\widehat{\Gamma}_k) =: \Gamma_k \approx \Gamma$$

4.3 Optimización de T-mesh

Como resultado de la etapa anterior, el contorno $\widehat{\Gamma}_k$ de la T-mesh paramétrica $\widehat{\Omega}_k$ se mapea al contorno del objeto, generando una superficie Γ_k, la cual es la aproximación final de Γ. La posición de los nodos del interior se determinan mediante un proceso de optimización de T-mesh. Una vez recolocados los nodos interiores, la T-mesh física resultante Ω_k es una aproximación de la geometría original. La parametrización volumétrica a trozos obtenida se denota como

$$\Pi : \widehat{\Omega}_K \to \Omega_K,$$

donde todo punto p de una celda de $\widehat{\Omega}_K$ se pude mapear a un punto q de la celda física Ω_K usando un mapeo apropiado de la celda.

La clave del método está en el proceso de optimización que permite obtener una T-mesh física de alta calidad, la cual se utiliza para construir la representación spline del objeto. Para reducir el esfuerzo computacional durante la optimización, es preferible realizar una recolocación previa de los nodos interiores. Para ello podemos utilizar un suavizado Laplaciano o Coons patch [59, 60]. Esta recolocación previa facilita el proceso de desenredo, pero en general no es capaz de obtener una malla sin enredos. Por tanto, es esencial aplicar un algoritmo efectivo de optimización.

El proceso de optimización de la malla consiste en un proceso iterativo donde cada nodo se mueve a una nueva posición de forma que mejora la calidad de su malla local. Esta nueva posición se determina minimizando una función objetivo basada en la medida de calidad mean ratio. Esta medida de calidad fue introducida originalmente para un triángulo, y se define en términos de la matriz Jacobiana del mapeo afín desde un triángulo ideal al triángulo físico. A su vez, la distorsión de un elemento se define como la inversa de su calidad. En este trabajo, para medir la distorsión de un elemento cuadrilátero (hexaedro) se utiliza una medida de distorsión puntual definida

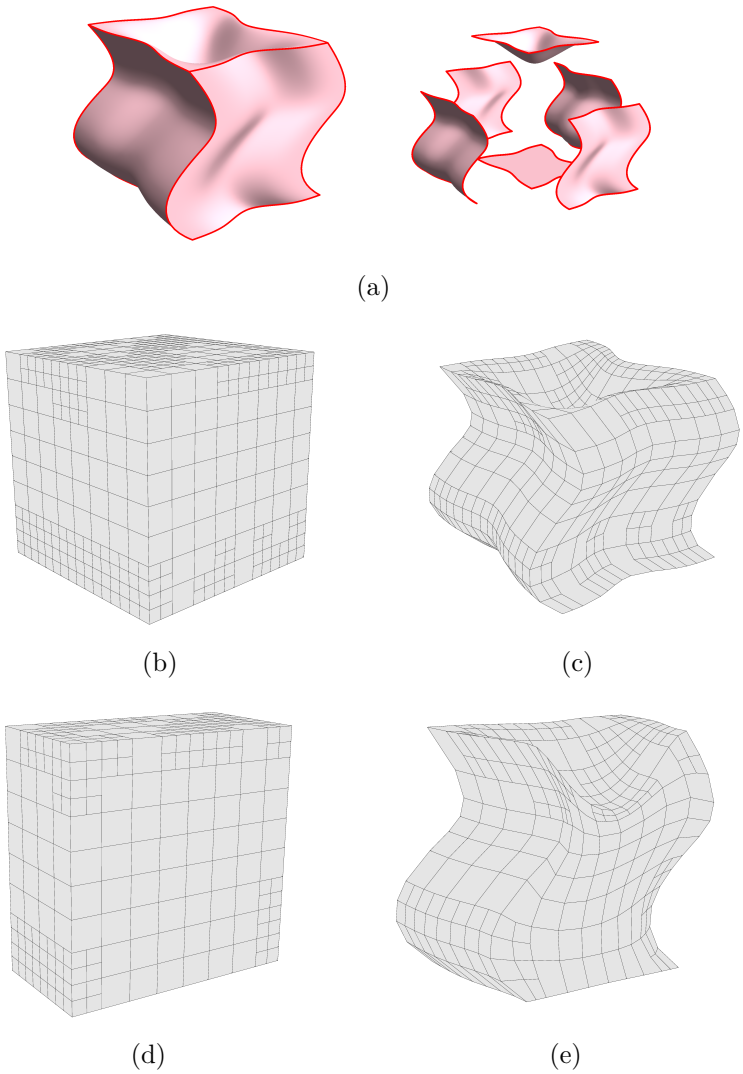


Figura 4.2. Construcción de T-mesh para la geometría Deformed cube. (a) T-mesh paramétrica adaptada al contorno de la geometría. (b) T-mesh física optimizada. (c) Sección de la T-mesh paramétrica. (d) Sección de la T-mesh física.

en términos de la matriz Jacobiana de un mapeo apropiado desde el elemento ideal al físico. Con eso, podemos definir una medida de distorsión en todo el elemento por medio de la norma L^1 de esta distorsión puntual.

La función objetivo para un nodo libre se construye como la suma de las medidas de distorsión de todos los elementos de su malla local. Para cada celda de la T-mesh física se toma como idea su celda correspondiente en la T-mesh paramétrica. Repitiendo el proceso de optimización para todos los nodos interiores de la malla se consigue minimizar la deformación de la malla física respecto a la paramétrica.

4.3.1 Medida de calidad para un elemento arbitrario

Para extender el concepto de medida de calidad para elementos no símplices (o para elementos de alto orden) con una matriz Jacobiana no constante, se introduce los siguientes conceptos.

Sea Ω_e una celda de cualquier tipo en la malla local del nodo libre $\mathbf{x}$. Primero se define una medida de distorsión puntual para la celda dada usando un mapeo apropiado $\mathbf{G} : \widehat{\Omega}_e \to \Omega_e$ desde el elemento ideal al elemento físico. Para ello se utilizará un mapeo cuadrático del elemento.

La distorsión de un elemento Ω_e en un punto $\boldsymbol{\xi}$ del elemento ideal se define como

$$\eta^*(\boldsymbol{\xi}) = \frac{\|\mathbf{J}_G(\boldsymbol{\xi})\|^2}{d\, h(\gamma(\boldsymbol{\xi}))^{2/d}}, \quad \boldsymbol{\xi} \in \widehat{\Omega}_e,$$

donde $d = 2,3$, $\mathbf{J}_G(\boldsymbol{\xi}) = (\partial \mathbf{G}/\partial \boldsymbol{\xi})$ es la matriz Jacobiana del mapeo $\mathbf{G}$ en el punto $\boldsymbol{\xi}$, $\|\mathbf{J}_G(\boldsymbol{\xi})\|$ es su norma de Frobenius y $\gamma(\boldsymbol{\xi}) = \det(\mathbf{J}_G)$. La función h se define como $h(\gamma) = \frac{1}{2}(\gamma + \sqrt{\gamma^2 + 4\delta^2})$, ver [55].

La medida de distorsión global para el elemento físico Ω_e se puede definir a través de la norma L^1 de la distorsión puntual, esto es

$$\eta^*_{\Omega_e} = \frac{1}{V_{\widehat{\Omega}_e}} \int_{\widehat{\Omega}_e} \eta^*(\boldsymbol{\xi})\, d\widehat{\Omega}_e,$$

donde $V_{\widehat{\Omega}_e}$ es el volumen de $\widehat{\Omega}_e$. Otras normas, como por ejemplo L^2 también podrían se usadas.

La distorsión global $\eta^*_{\Omega_e}$ toma valores entre 1 e ∞. Si el elemento físico Ω_e coincide con el ideal, entonces $\eta^*(\boldsymbol{\xi}) \to 1$ cuando $\delta \to 0$. Además, esta medida global de distorsión presenta pseudo-barreras, esto es, $\eta^*_{\Omega} \to \infty$ cuando $\sigma(\boldsymbol{\xi}) \leq 0$ y $\delta \to 0$.

La calidad global de un elemento se define como la inversa de su distorsión global, esto es

$$q^*_{\Omega_e} = \frac{1}{\eta^*_{\Omega_e}},$$

cuyo valor está comprendido en $[0,1]$.

La función objetivo para el nodo libre $\mathbf{x}$ se define usando la medida global de distorsión:

$$K^*(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \eta^*_{\Omega_i}(\mathbf{x}),$$

siendo N el número de elementos conectados al nodo libre $\mathbf{x}$ y $\eta^*_{\Omega_i}(\mathbf{x})$ la medida global de distorsión del i-ésimo elemento.

La integral se evalúa mediante una cuadratura numérica. Sea $\{\boldsymbol{\xi}_j\}_{j=1,M}$ un conjunto de puntos de cuadratura en el elemento ideal $\widehat{\Omega}_e$. La función objetivo para el nodo libre $\mathbf{x}$ sería

$$K^*(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} w_j \eta^*_{\Omega_i}(\boldsymbol{\xi}_j)\right),$$

donde w_j son los correspondientes pesos para los puntos de cuadratura $\boldsymbol{\xi}_j$. Por tanto, en la práctica, la función objetivo está basada en la medida de distorsión puntual del elemento, evaluada en un conjunto de puntos en el elemento ideal. De forma análoga a la malla de triángulos, el problema de optimización sin restricciones se puede resolver con algún método estándar, ver por ejemplo [66].

Para evitar un coste computacional excesivamente elevado en la minimización, es preferible utilizar pocos puntos de cuadratura. Además, también es preferible evaluar la calidad en los vértices del elemento, ya que normalmente la mayor distorsión ocurre en estos puntos. Por ello utilizamos la cuadratura de Gauss-Lobatto, la cual incluye puntos de cuadratura en la frontera del elemento. Concretamente, se utilizan tres puntos de cuadratura $\{-1,0,1\}$ para el intervalo de referencia $[-1,1]$.

Mapeo paramétrico para elementos de una T-mesh 2D

Como se indicó anteriormente, para cada celda de la T-mesh física se toma como elemento ideal su celda correspondiente en la T-mesh paramétrica. En nuestro caso particular en el que la T-mesh se construye mediante un refinamiento tipo quadtree, todas las celdas paramétricas son cuadrados, por lo que tomamos el elemento $\widehat{\Omega}_e = [-1,1]^2$ como elemento ideal para todas las celdas. Las celdas físicas en la T-mesh, en general no son cuadriláteros con lados rectos ya que pueden contener hanging nodes, los cuales no tienen por qué estar alineados con los nodos de la arista. En nuestra T-mesh balanceada, una arista contiene como máximo un hanging node. Para tener en cuenta todos los nodos de la celda en su medida de distorsión, podemos considerar elementos curvilíneos de segundo orden construidos vía interpolación de Lagrange. Sea $\boldsymbol{\xi}_i \in \mathbb{R}^2$, $i = 1,..9$, los nodos del elemento ideal $\widehat{\Omega}_e$, y sea N_i las funciones de forma bicuadráticas de Lagrange definidas en $\widehat{\Omega}_e$. El mapeo paramétrico

$$\mathbf{G}(\boldsymbol{\xi}) = (x(\xi,\eta), y(\xi,\eta)) = \sum_{i=1}^{9} \mathbf{x}_i N_i(\boldsymbol{\xi}), \tag{4.1}$$

transforma $\widehat{\Omega}_e$ en la celda física Ω_e cuyos nodos son $\mathbf{x}_i$, $i = 1,9$. Por tanto, para definir esta transformación cuadrática son necesarios 9 nodos. En nuestro caso algunos de estos nodos pueden no existir en la malla, por lo que tenemos que completar la celda dada con algunos nodos virtuales. La posición de estos nodos virtuales depende de la posición de los nodos reales (existentes) en la celda, ver Fig. 4.3(a). Para asignar posiciones razonables a los nodos virtuales utilizamos una interpolación con Coons patch de las aristas de la celda. Supongamos que definimos la parametrización de las cuatro aristas curvas de la celda como $\mathbf{x}(0,\eta)$, $\mathbf{x}(1,\eta)$, $\mathbf{x}(\xi,0)$ y $\mathbf{x}(\xi,1)$. La transformación bilineal de Coons se define como

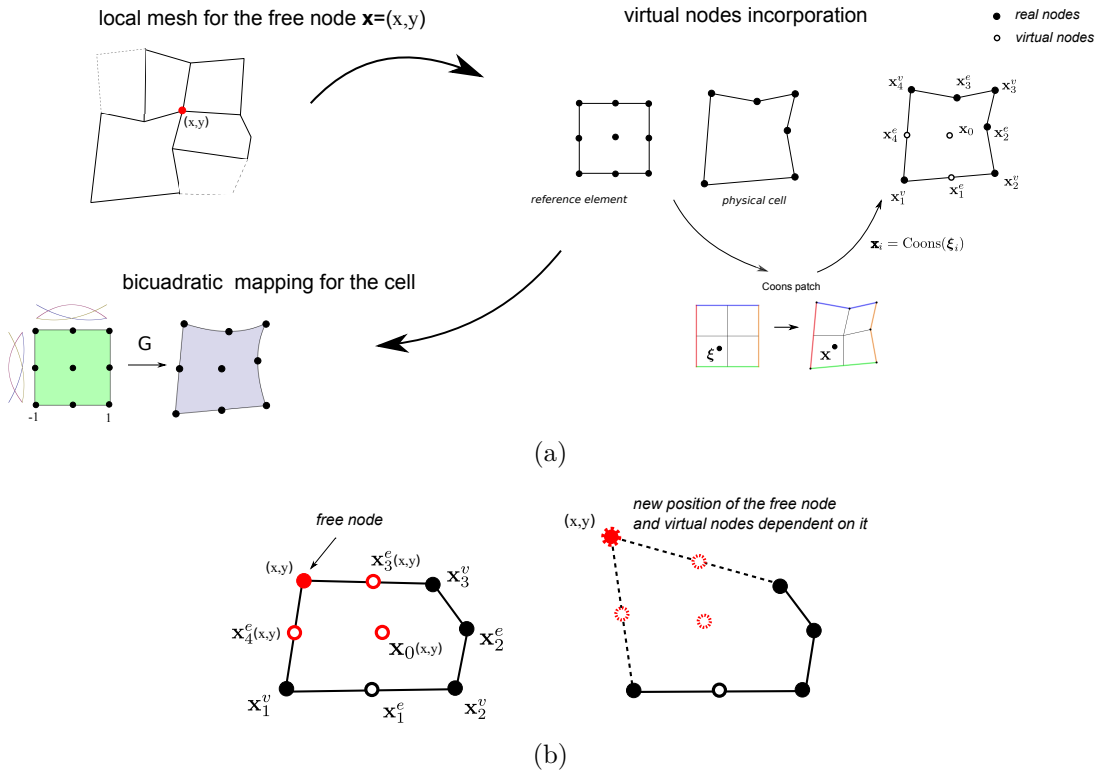


Figura 4.3. Asignación de nodos virtuales para el caso 2D. (a) Definición de un mapeo cuadrático local para cada celda. (b) Los nodos virtuales conectados al nodo libre actualizan su posición cuando el nodo libre se mueve en el proceso de optimización.

$$\begin{aligned}\mathbf{x}(\xi,\eta) =& (1-\xi)\mathbf{x}(0,\eta)+\xi\mathbf{x}(1,\eta)\\ &+(1-\eta)\mathbf{x}(\xi,0)+\eta\mathbf{x}(\xi,1)\\ &-\begin{bmatrix} 1-\xi & \xi \end{bmatrix}\begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1)\\ \mathbf{x}(1,0) & \mathbf{x}(1,1)\end{bmatrix}\begin{bmatrix} 1-\eta\\ \eta\end{bmatrix}\end{aligned}$$

Haciendo esto, obtenemos las siguientes posiciones para los nodos virtuales. Los nodos virtuales de las aristas se sitúan en el punto medio de la arista correspondiente, esto es, siguiendo la notación de la Fig. 4.3(a), el nodo virtual de la arista $(\mathbf{x}_k^v, \mathbf{x}_{k+1}^v)$ se define como

$$\mathbf{x}_k^e = \frac{1}{2}(\mathbf{x}_k^v + \mathbf{x}_{k+1}^v), \quad k \in \{1,2,3,4\}. \tag{4.2}$$

Luego, para situar el nodo interior de la celda se aplica $\mathbf{x}(1/2,1/2)$, obteniendo la siente expresión en términos de los nodos de la frontera de la celda

$$\mathbf{x}_0 = \frac{1}{2}(\mathbf{x}_1^e + \mathbf{x}_2^e + \mathbf{x}_3^e + \mathbf{x}_4^e) - \frac{1}{4}(\mathbf{x}_1^v + \mathbf{x}_2^v + \mathbf{x}_3^v + \mathbf{x}_3^v). \tag{4.3}$$

Destacar que los nodos virtuales no introducen grados de libertad adicionales en la malla, ya que sus posiciones dependen de los nodos ya existentes. Así, los nodos virtuales conectados a un nodo libre actualizan su posición cuando el nodo libre se mueve durante el proceso de optimización, ver Fig. 4.3(b).

Mapeo paramétrico para elementos de una T-mesh 3D

Se puede generalizar de forma directa a 3D las ideas presentadas para el caso 2D. Para obtener una medida de distorsión global de una celda de la T-mesh 3D se define un mapeo tricuadrático desde el elemento ideal $\widehat{\Omega}_e = [-1,1]^3$ al físico, para lo que se requiere 3^3 nodos. La celda se complemente con nodos virtuales calculados en términos de los nodos existentes en la celda. Primero se asignan los nodos virtuales a cada cara de la celda para definir un mapeo bicuadrático de las caras. Para ello se aplica exactamente el mismo procedimiento desarrollado para el caso 2D utilizando interpolación Coons de las cuatro aristas de la cara. En este caso, algunas caras de la celda puede contener los 9 nodos necesarios, y por tanto en ese caso no será necesario asignar nodos virtuales. Si la cara no tiene alguno de los 9 nodos, los nodos virtuales de aristas y del centro de cara se asignan utilizando las fórmulas (4.2) y (4.3). Una vez tenemos la representación bicuadrática de las seis caras de la celda, se define una interpolación Coons 3D de las caras para situar el nodo central de la celda, ver Fig. 4.4. Par situar este nodo central se aplica $\mathbf{x}(1/2,1/2,1/2)$ obteniendo la siguiente expresión:

$$\mathbf{x}_0 = \frac{1}{8}(\mathbf{x}_1^v + \mathbf{x}_2^v + \cdots + \mathbf{x}_8^v) - \frac{1}{4}(\mathbf{x}_1^e + \mathbf{x}_2^e + \cdots + \mathbf{x}_{12}^e) + \frac{1}{2}(\mathbf{x}_1^f + \mathbf{x}_2^f + \cdots + \mathbf{x}_6^f),$$

donde $\mathbf{x}_i^v, i = 1,8$ son los nodos de las esquinas, $\mathbf{x}_i^e, i = 1,12$ son los nodos de centro de arista y $\mathbf{x}_i^f, i = 1,6$ son los nodos de centro de cara.

Luego se define una transformación tricuadrática desde el elemento ideal $\widehat{\Omega}_e$ al elemento físico Ω_e cuyos nodos son $\mathbf{x}_i$, $i = 1,27$:

$$\mathbf{G}(\boldsymbol{\xi}) = (x(\xi,\eta,\zeta), y(\xi,\eta,\zeta), z(\xi,\eta,\zeta)) = \sum_{i=1}^{27} \mathbf{x}_i N_i(\boldsymbol{\xi}),$$

donde N_i son las funciones de forma de Lagrange tricuadráticas definidas en el elemento ideal.

Remark 1 Destacar que la medida de distorsión de un triángulo definida en términos de su matriz Jacobiana constante es un caso particular de la medida global de distorsión basada en la norma L^1 de la distorsión puntual definida para un elemento arbitrario. Además, la medida de distorsión puntual para un mapeo cualquiera $\mathbf{G}$ está relacionad con la medida de distorsión de un triángulo de la siguiente forma: el valor de la distorsión en un punto $\mathbf{P}_0$ del elemento ideal es la distorsión de un triángulo infinitesimal en el dominio físico formado por las dos curvas isoparamétricas que pasan por el punto $\mathbf{P}_0' = \mathbf{G}(\mathbf{P}_0)$.

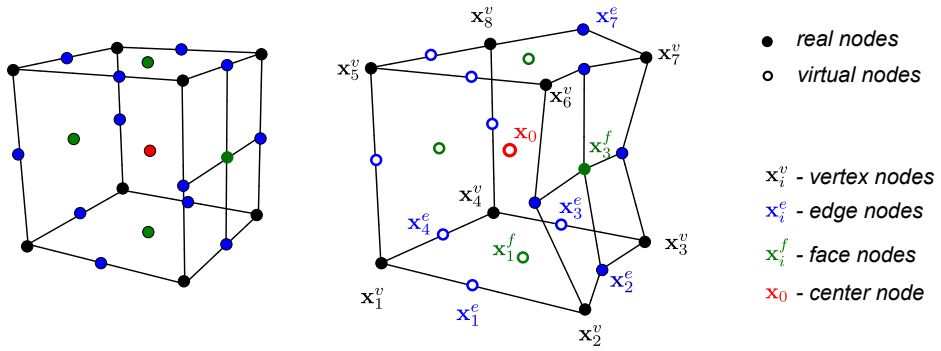


Figura 4.4. Asignación de nodos virtuales para el caso 3D.

Remark 2 Otra forma de evaluar la distorsión de un elemento no símplice es descomponer el elemento en varios símplices y definir la distorsión como la suma de las distorsiones de estos elementos símplices. Esta estrategia se utilizó en el capítulo 2 para la parametrización T-spline de geometrías 2D. Sin embargo, esta aproximación tiene algunos inconvenientes dado la gran variedad de posibles descomposiciones que se pueden dar en 3D, y por tanto no se puede considerar como una generalización de la medida de distorsión introducida originalmente para triángulos.

4.3.2 Función objetivo con pesos

La función objetivo definida en la sección 4.3.1 es una suma de términos correspondientes a la distorsión global de cada celda. Una malla regular (sin hanging nodes) es un caso particular de nuestra T-mesh. En estas mallas, un nodo siempre está rodeado de celdas de la misma escala, en el sentido de que en el espacio paramétrico las celdas correspondientes tienen el mismo tamaño. En esta situación es razonable componer la función objetivo como la suma de las distorsiones globales. En cambio, en una T-mesh general, un nodo libre puede estar rodeado de celdas de diferente tamaño. En nuestros experimentos computacionales hemos observado que para estos una función objetivo con peso produce mejores resultados. Es decir, la función objetivo se define como

$$K^*(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} w_i\, \eta^*_{\Omega_i}(\mathbf{x}),$$

donde los pesos w_i tienen relación con la escala de la celda correspondiente, esto es, las celdas de igual escala tendrán el mismo peso. En nuestra mallas quadtree/octree balanceadas, un nodo libre puede estar rodeado sólo de celdas de dos niveles diferentes. Estableciendo a 1 el peso las celdas del nivel k, necesitamos otro peso para las celdas de nivel $k-1$. Asignar diferentes valores para este peso puede llevar a obtener resultados ligeramente diferentes. En la práctica hemos observado que en muchos casos, el resultado de la parametrización es mejor cuando se utiliza peso $w_i = 2$ para las celdas

grandes de la malla local. La función objetivo con pesos tiende a obtener una mayor ortogonalidad de las curvas paramétricas.

5. Resultados y aplicaciones

En esta sección se pone a prueba las estrategias de parameterización spline propuestas, utilizando para ello distintas geometrías. Para todas ellas, el dato de entrada consiste en un conjunto de parches compatibles (curvas spline, poligonales, superficies spline o triangulaciones) que describen la geometría. Para comprobar la validez de la parametrización obtenida se evalúa su mean ratio Jacobian en 4 puntos de cuadratura por dirección paramétrica en cada celda. Cuando sea necesario, se realiza refinamientos adaptativos para mejorar la calidad de la parametrización final, tal como se ha explicado en la descripción de la estrategia. Para todas las geometrías que se muestra, se ha obtenido una parametrización spline válida para Análisis Isogeométrico. Además se muestra un ejemplo de aplicación de Análisis Isogeométrico utilizando una geometría parametrizada con nuestro método.

5.1 Ejemplo de parameterización T-spline de geometría 2D

En este ejemplo se parametriza una geometría 2D utilizando el método descrito en el capítulo 2. Como dato de entrada se tiene 4 poligonales que describen la geometría. Cada poligonal se mapea a una arista del cuadrado paramétrico vía chord-length. Las figuras 5.1(a) y (b) muestran la T-mesh paramétrica y la T-mesh física tras la optimización. Las poligonales dato están marcadas en color. La figura 5.1(d) muestra la representación T-spline resultante y el mapa de color del mean ratio Jacobian. El método obtiene una malla con 844 celdas. Para construir la representación T-spline fueron necesarios 1456 puntos de control. El Jacobiano de la parametrización T-spline es positivo en todos los puntos de cuadratura, y en este caso no fue necesario aplicar refinamientos adaptativos. El valor mínimo del mean ratio Jacobian en los puntos de cuadratura es 0.31. La figura 5.1(c) muestra algunas estadísticas sobre el comportamiento del mean ratio Jacobian. En esta figura se muestra el mean ratio Jacobian ordenado de forma creciente por la media en cada celda. Las líneas rojas y azules corresponden al valor máximo y mínimo en cada celda.

5.2 Ejemplo de parametrización EP-spline de geometría 2D

Aquí se presenta la parametrización de una geometría compleja utilizado la estrategia propuesta en el capítulo 4. El dato de entrada consiste en 4 poligonales que representan el contorno del mapa de USA. Para esta geometría se requiere un mayor número de refinamientos en la T-mesh para poder aproximar las características del contorno. Las figuras 5.2(a) y (b) muestran la T-mesh paramétrica y física tras la optimización. Para construir la T-mesh se parte de una malla uniforme de 16×16 para garantizar suficientes grados de libertad en el interior. La optimización se realiza con la función objetivo con pesos, estableciendo peso $w = 2$ para las celdas grandes de la

malla local. En este caso fueron necesarios tres refinamientos adaptativos para conseguir una parametrización EP-spline válida. La malla resultante tiene 3292 celdas y fueron necesarios 4059 puntos de control para construir la parametrización EP-spline. El valor mínimo del mean ratio Jacobian en los puntos de cuadratura es 0.09.

5.3 Ejemplo de parametrización EP-spline de geometría 3D

Este ejemplo muestra la aplicación de la estrategia en un modelo CAD diseñado con el software Rhinoceros. El modelo consiste en una deformación de las caras de un cubo. Como dato de entrada tomamos la representación spline de cada cara. Las figuras 5.3(a) and (b) muestran la T-mesh paramétrica y la T-mesh física tras la optimización. Para construir la T-mesh adaptada se comienza con una malla regular de $8\times 8\times 8$ elementos. La optimización se lleva a cabo con la función objetivo con pesos, en este caso con peso $w=8$ para los elementos grandes de la malla local. Utilizamos $w=8$ para tratar de preservar la relación de volúmenes de los elementos paramétricos. Esto impone una mayor ortogonalidad, por lo que es una elección recomendable para geometrías tipo cubo. La malla resultante tiene 1121 celdas y para la representación EP-spline fueron necesarios 2360 puntos de control. El valor mínimo del mean ratio Jacobian en los puntos de cuadratura es 0.35.

5.4 Aplicación de Análisis Isogeométrico

Se resuelve un problema 3D de Poisson con condiciones de contorno tipo Dirichlet en la geometría parametrizada en la sección anterior:

$$\begin{aligned} -\triangle u &= f \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \partial\Omega. \end{aligned}$$

El problema se establece de modo que la solución exacta viene dada por una función con un fuerte frente de onda dado por

$$u(r) = \arctan(\alpha(r-r_0)),$$

donde $r=\sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}$, el parámetro α determina el frente de onda y r_0 es su localización. En este ejemplo se utiliza $\alpha=200$ y $r_0=0.6$. El centro del frente de onda (x_c,y_c,z_c) se sitúa fuera del dominio computacional, de modo que la función es suave en Ω. Se realizan refinamientos adaptativos con EP-splines utilizando un estimador de error basado en el residuo. En la figura 5.4 se muestra la solución numérica correspondiente al refinamiento de la iteración final. Como se esperaba, el estimador de error ha marcado para refinar la zona del frente de onda. La evolución del error exacto en norma L^2 y seminorma H^1, así como su comparación con un refinamiento uniforme, se puede ver en la figura 5.4(c).

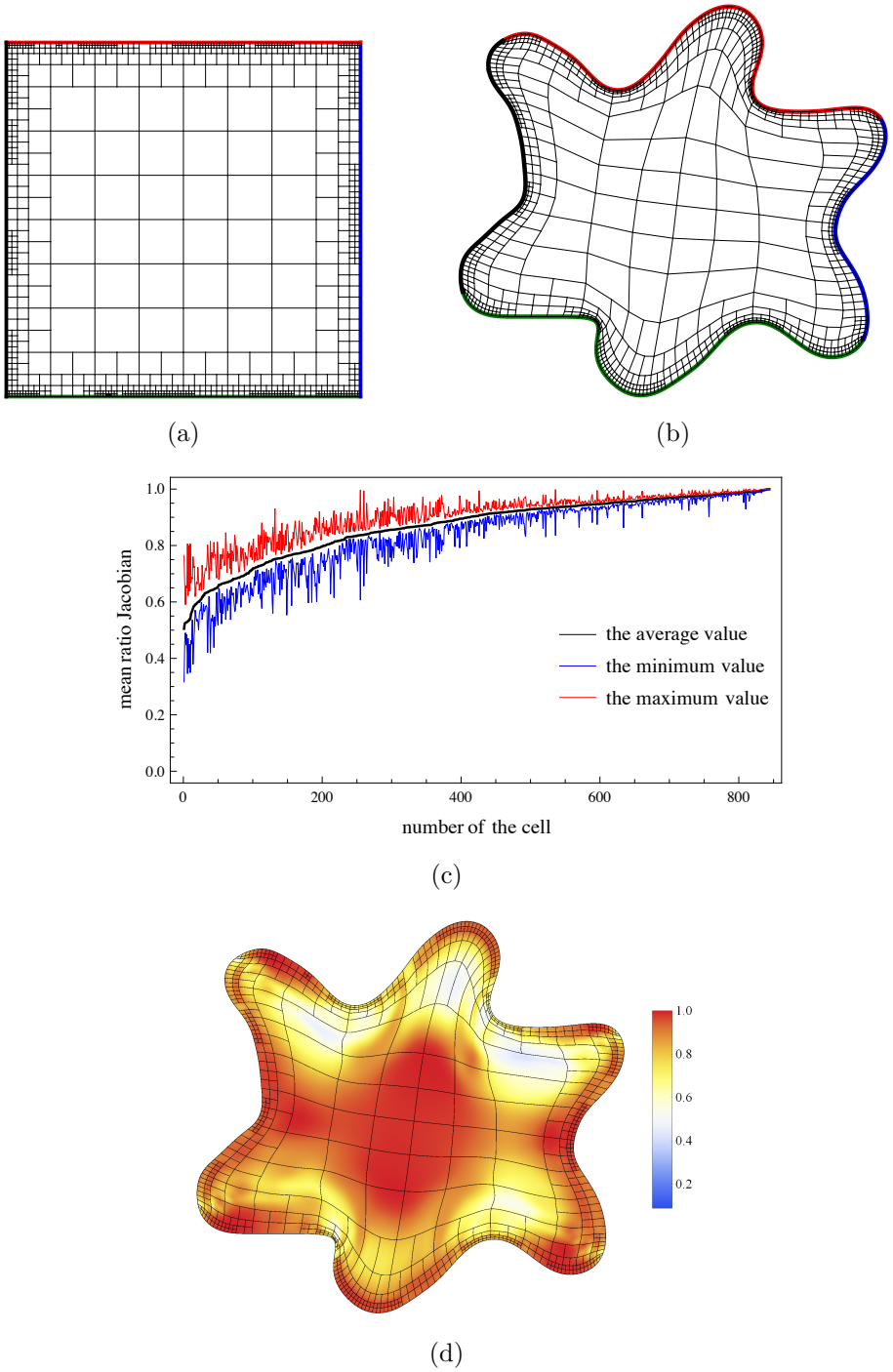


Figura 5.1. Geometría Spot. (a) T-mesh paramétrica. (b) T-mesh física optimizada. (c) Estadísticas del mean ratio Jacobian. (d) Representación T-spline y mapa de color del mean ratio Jacobian.

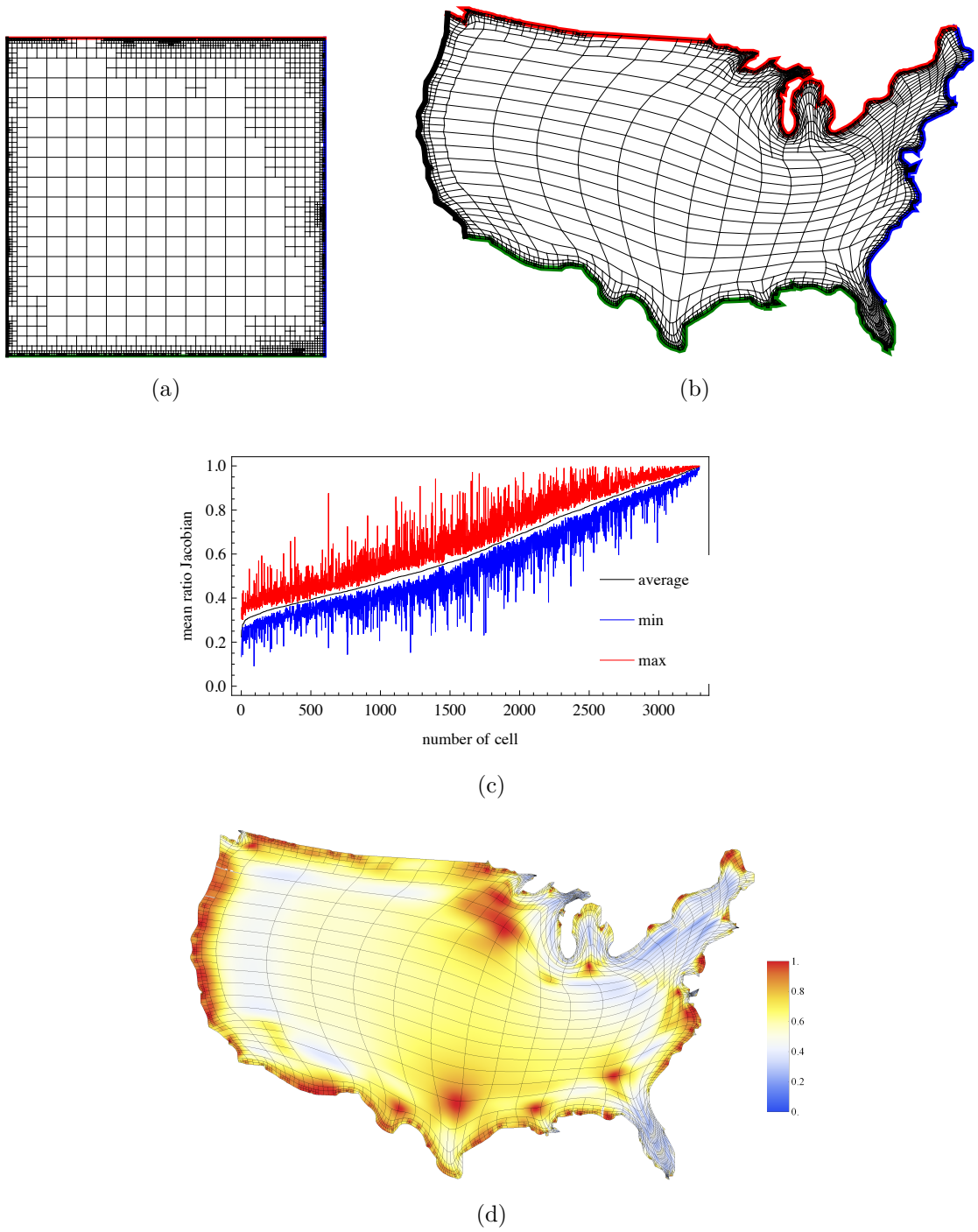
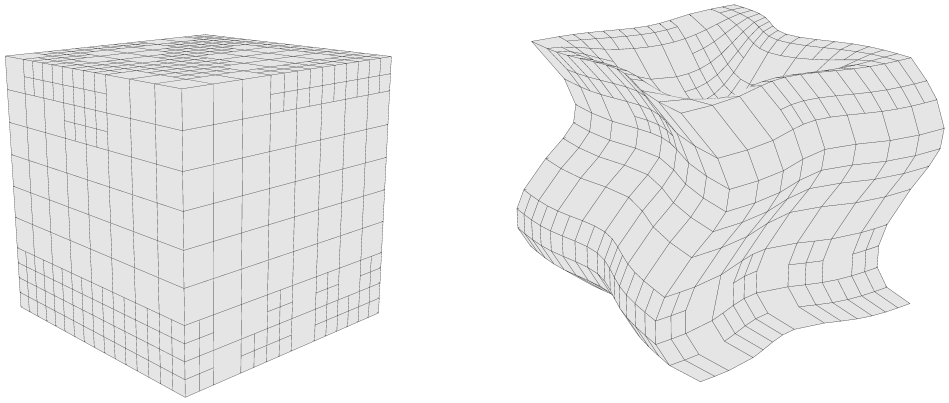
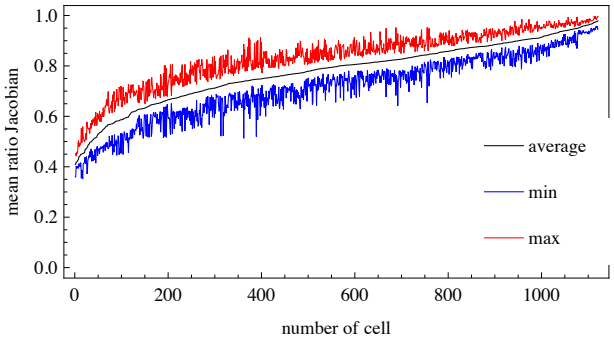


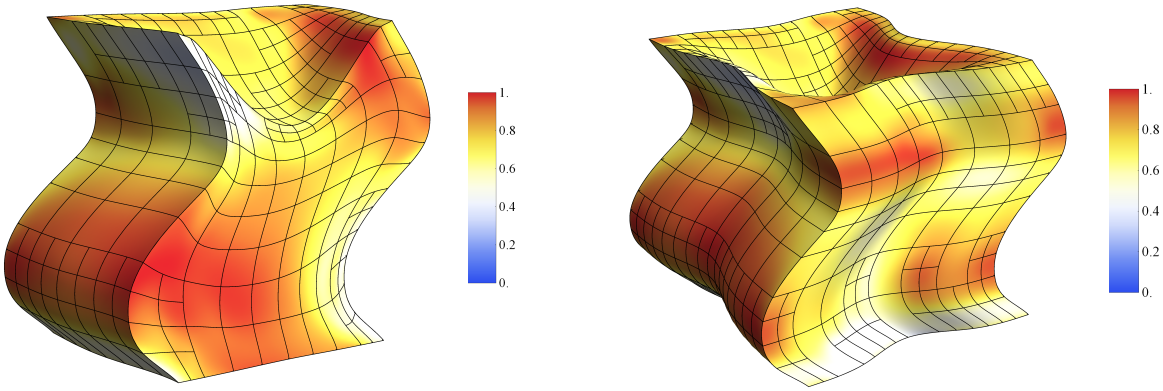
Figura 5.2. Geometría USA. (a) T-mesh paramétrica. (b) T-mesh física optimizada. (c) Estadísticas del mean ratio Jacobian. (d) Representación EP-spline y mapa de color del mean ratio Jacobian.



(a) (b)

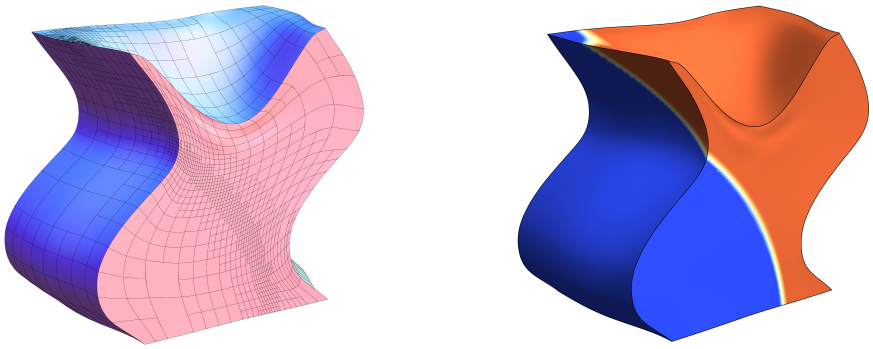


(c)

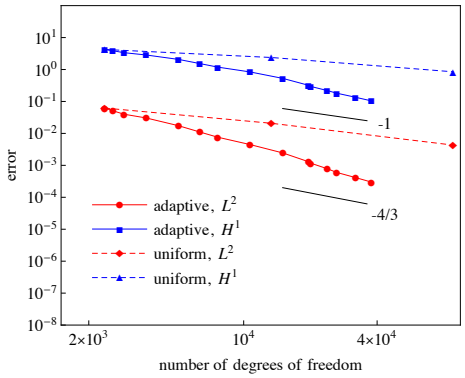


(d) (e)

Figura 5.3. Geometría Cubo deformado. (a) T-mesh paramétrica. (b) T-mesh física optimizada. (c) Estadísticas del mean ratio Jacobian. (d) Representación EP-spline y mapa de color del mean ratio Jacobian en la sección $\xi = 1/2$. (e) Mean ratio Jacobian en toda la geometría.



(a) (b)



(c)

Figura 5.4. Aplicación de IGA a un problema Poisson 3D. (a) Refinamiento final en el dominio físico. (b) Solución numérica en el dominio físico. (c) Convergencia del error con refinamiento uniforme y refinamiento adaptativo.

6. Conclusiones y líneas futuras

6.1 Conclusiones

En esta tesis se ha propuesto una estrategia para obtener una parametrización spline de alta calidad, tanto para geometrías 2D como 3D, válida para su uso en Análisis Isogeométrico. El método sólo requiere una representación de la frontera de la geometría como dato de entrada y construye automáticamente una transformación spline entre el dominio físico y paramétrico. La clave del método está en definir una transformación isomórfica entre entre la T-mesh paramética y la T-mesh física, encontrando las posiciones óptimas de los nodos del interior de la malla física mediante un algoritmo de desenredo y suavizado. El método fue propuesto inicialmente para la parametrización T-spline de geometrías 2D, donde la optimización de la T-mesh se realiza mediante una descomposición de los elementos en triángulos. Luego, el método se extendió para geometrías 2D y 3D. En este caso, se propone otro algoritmo de optimización de T-mesh basado en una medida de distorsión puntual de los elementos de la T-mesh en lugar de la descomposición en triángulos.

La representación spline de la geometría se calcula imponiendo condiciones de interpolación, utilizando la información proporcionada por la correspondencia uno a uno entre las mallas paramétrica y física. Para medir la calidad de la parametrización se evalúa su mean ratio Jacobian. Además, se detectan las zonas de baja calidad y se realiza un refinamiento adaptativo para aumentar el número de grados de libertad en las zonas de alta distorsión. Esta estrategia permite obtener una parametrización válida para el análisis sin Jacobiano negativo, incluso en dominios complejos.

Para la aplicación de IGA es esencial disponer de un método efectivo para obtener una parametrización volumétrica de la geometría a partir de la descripción de su frontera. Nuestro objetivo en esta tesis ha sido desarrollar un método alternativo para la construcción de una parametrización spline basado en las ideas del método del Mecano.

Las principales partes que se incluyen en esta tesis son:

- descripción de un método para construir una parametrización T-spline de geometrías 2D para su aplicación en IGA;
- un estudio de varias estrategias para la optimización de mallas de hexaedros, basadas en diferentes funciones objetivo;
- una extensión del método para obtener parametrizaciones spline de alta calidad, tanto para geometrías 2D como 3D, donde la optimización T-mesh está basada en una medida de distorsión puntual de los elementos en lugar de realizar una descomposición en símplices;
- se prueba los métodos de parametrización utilizando distintas geometrías. Además se resuelve distintos problemas con IGA empleando las geometrías parametrizadas.

La efectividad del método se probó con distintos tipos de geometrías. Para todas ellas, la estrategia obtiene una parametrización válida para ser utilizada en Análisis

Isogeométrico. En algunas geometrías fue necesario aplicar refinamientos adaptativos para garantizar Jacobiano positivo en todos los puntos de cuadratura. Se resuelven diferentes tipos de problemas usando Análisis Isogeométrico con las geometrías parametrizadas con nuestro método.

En esta tesis también se presenta de forma resumida una estrategia para construir espacios spline polinómicos. Esta estrategia permite definir espacios de spline cúbicas con propiedades adecuadas para el análisis sobre una T-mesh quadtree/octree fuertemente balanceada. Estas bases se emplearon en nuestro método de parametrización 2D/3D.

6.2 Líneas futuras

Cabe mencionar las siguientes posibles líneas futuras de trabajo:

- Todas las geometrías presentadas en este trabajo han sido parametrizadas utilizando el cuadrado/cubo unitario como espacio paramétrico. Sin embargo, para facilitar la parametrización de geometrías muy complicadas es esencial trabajar con espacios paramétricos más complejos que se ajusten mejor a la geometría. Esto permitiría reducir la alta distorsión que aparece cuando la geometría tiene una forma muy diferente a la del cuadrado/cubo. Para ello, además es necesario elaborar un procedimiento para construir espacios spline sobre este tipo de dominios paramétricos, tratando de conservar la regularidad de las funciones base.

- Un problema similar aparece con geometrías de orden mayor a 0. Es necesario desarrollar una estrategia para definir parametrizaciones spline para este tipo de geometrías.

- Obtener una parametrización 3D global conlleva varias limitaciones importantes dado la naturaleza de producto tensorial de las funciones spline. Primero, para dominios complejos es necesario dividir el objeto y construir la parametrización de varios parches, lo que provoca una pérdida de regularidad en las zonas de unión. Segundo, dada la superficie de un sólido no es trivial encontrar una partición óptima en seis parches correspondientes a las caras del cubo paramétrico. Además, para garantizar regularidad en la parametrización (Jacobiano positivo), los parches deben unirse formando zonas convexas, lo cual puede ser imposible en algunas geometrías. Es muy importante poder diseñar una estrategia para poder resolver todos estos problemas.

Bibliografía

[1] L. Piegl, W. Tiller, The NURBS book, Springer, New York, 1997.

[2] G. Farin, Curves and Surfaces for CAGD: a Practical Guide, 5th Edition, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[3] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Meth. Appl. Mech. Eng. 194 (2005) 4135–4195.

[4] J. Cottrell, A. Reali, Y. Bazilevs, T. Hughes, Isogeometric analysis of structural vibrations, Comput. Meth. Appl. Mech. Eng. 195, Issues 41 43 (2006) 5257–5296.

[5] H. Gómez, V. Calo, Y. Bazilevs, T. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, Comput. Meth. Appl. Mech. Eng. 197, Issues 49 50 (2008) 4333–4352.

[6] H. Gómez, T. J. Hughes, X. Nogueira, V. M. Calo, Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations, Comput. Meth. Appl. Mech. Eng. 199, Issues 25 28 (2010) 1828–1840.

[7] J. Liu, L. Dede, J. A. Evans, M. J. Borden, T. Hughes, Isogeometric analysis of the advective Cahn–Hilliard equation: Spinodal decomposition under shear flow, Journal of Computational Physics 242 (2013) 321–350.

[8] M. Kästner, P. Metsch, R. de Borst, Isogeometric analysis of the Cahn–Hilliard equation – a convergence study, Journal of Computational Physics 305 (2016) 360–371.

[9] F. Auricchio, L. B. D. Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences Vol. 20, No. 11 (2010) 2075–2107.

[10] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, T. J. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Meth. Appl. Mech. Eng. 267 (2013) 170–232.

[11] H. Casquero, L. Liu, Y. Zhang, A. Reali, H. Gómez, Isogeometric collocation using analysis-suitable T-splines of arbitrary degree, Comput. Meth. Appl. Mech. Eng. 301 (2016) 164–186.

[12] C. Anitescu, Y. Jia, Y. J. Zhang, T. Rabczuk, An isogeometric collocation method using superconvergent points, Comput. Meth. Appl. Mech. Eng. 284 (2015) 1073–1097.

[13] Y. Bazilevs, L. Beirão Da Veiga, J. Cottrell, T. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, Mathematical Models and Methods in Applied Sciences 16 (7) (2006) 1031–1090.

[14] L. Beirão da Veiga, A. Buffa, R. J., G. Sangalli, Some estimates for h-p-k-refinement in Isogeometric Analysis, Numer. Math. 118 (2) (2011) 271–305.

[15] A. Tagliabue, L. Dedé, A. Quarteroni, Isogeometric Analysis and error estimates for high order partial differential equations in fluid dynamics, Computers and Fluids 102 (2014) 277–303.

[16] T. J. R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 199, 5-8 (2010) 301–313.

[17] F. Auricchio, F. Calabró, T. J. R. Hughes, A. Reali, G. Sangalli, A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 249-252 (2012) 15–27.

[18] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, T. J. R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Meth. Appl. Mech. Eng. 199 (2010) 357–373.

[19] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric analysis: Toward integration of CAD and FEA, 1st Edition, Wiley Publishing, 2009.

[20] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parametrization of computational domain in isogeometric analysis: Methods and comparison, Comput. Meth. Appl. Mech. Eng. 200 (2011) 2021–2031.

[21] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis, Comput. Aided Des. 45 (4) (2013) 812–821.

[22] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Variational harmonic method for parameterization of computational domain in 2D isogeometric analysis, in: 12th International Conference on Computer-Aided Design and Computer Graphics, IEEE, Jinan, 2011, pp. 223–228.

[23] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method, Journal of Computational Physics 252 (2013) 275–289.

[24] X. Li, X. Guo, H. Wang, Y. He, X. Gu, H. Qin, Harmonic volumetric mapping for solid modeling applications, in: Proc. of ACM Solid and Physical Modeling Symposium, Association for Computing Machinery, Inc., 2007, pp. 109–120.

[25] T. Martin, E. Cohen, R. Kirby, Volumetric parameterization and trivariate B-spline fitting using harmonic functions, Comput. Aid. Geom. Design 26 (2009) 648–664.

[26] Y. Zhang, W. Wang, T. J. R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Comput. Meth. Appl. Mech. Eng. 249-252 (2012) 185–197.

[27] W. Wang, Y. Zhang, L. Liu, T. J. R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, Computer-Aided Design 45 (2013) 351–360.

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

[29] M. A. Scott, X. Li, T. W. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T-splines, Comput. Meth. Appl. Mech. Eng. 213-216 (2012) 206–222.

[30] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, M. A. Scott, On linear independence of T-spline blending functions, Comput. Aid. Geom. Design 29 (2012) 63–76.

[31] D. R. Forsey, R. H. Bartels, Hierarchical B-spline refinement, Computer Graphics 22(4) (1988) 205–212.

[32] R. Kraft, Surface Fitting and Multiresolution Methods, Vol. 2, Vanderbilt University Press, 1997, Ch. Adaptive and linearly independent multilevel B-splines, pp. 209–218.

[33] A. V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Meth. Appl. Mech. Eng. 200 (2011) 3554–3567.

[34] D. Schillinger, L. Debé, M. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of nurbs, immersed boundary methods, and T-spline CAD surfaces, Comput. Meth. Appl. Mech. Eng. 249–252 (2012) 116–150.

[35] P. B. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical B-spline finite element method, Comput. Meth. Appl. Mech. Eng. 253 (2013) 584–598.

[36] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical T-meshes, Graphical Models 70 (2008) 76–86.

[37] T. Dokken, T. Lyche, K. F. Pettersen, Polynomial splines over locally refined box-partitions, Comput. Aid. Geom. Design 30(3) (2013) 331–356.

[38] C. de Boor, Splines as linear combinations of B-splines. A Survey (1976).

[39] M. Neamtu, Mathematical Methods for Curves and Surfaces, Vanderbilt University, Nashville, TN, USA, 2001, Ch. What is the Natural Generalization of Univariate Splines to Higher Dimensions?, pp. 355–392.

[40] M. Neamtu, Delaunay Configurations and Multivariate Splines: A Generalization of a Result of B. N. Delaunay, Transactions of the American Mathematical Society 359, No. 7 (2007) 2993–3004.

[41] M. Neamtu, Bivariate simplex B-Splines: A New Paradigm, in: In SCCG 01: Proceedings of the 17th Spring conference on Computer graphics, 2001, pp. 71–78.

[42] C. de Boor, K. Höllig, S. Riemenschneider, Box Splines, Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[43] M. J. D. Powell, M. A. Sabin, Piecewise quadratic approximations on triangles, ACM Trans. Math. Softw. 3 (4) (1977) 316–325.

[44] H. Speleers, P. Dierckx, S. Vandewalle, Numerical solution of partial differential equations with powell-sabin splines, J. Comput. Appl. Math. 189 (1-2) (2006) 643–659.

[45] H. Speleers, C. Manni, F. Pelosi, From NURBS to NURPS geometries, Computer Methods in Applied Mechanics and Engineering 255 (2013) 238–254.

[46] N. Jaxon, X. Qian, Isogeometric analysis on triangulations., Computer-Aided Design 46 (2014) 45–57.

[47] J. M. Escobar, J. M. Cascón, E. Rodríguez, R. Montenegro, A new approach to solid modeling with trivariate T-splines based on mesh optimization, Comput. Meth. Appl. Mech. Eng. 200 (2011) 3210–3222.

[48] J. M. Escobar, R. Montenegro, E. Rodríguez, J. M. Cascón, The meccano method for isogeometric solid modeling and applications, Engineering with Computers 30 (3) (2014) 331–343.

[49] R. Montenegro, J. M. Cascón, E. Rodríguez, J. M. Escobar, G. Montero, The meccano method for automatic three-dimensional triangulation and volume parametrization of complex solids, in: Developments and Applications in Engineering Computational Technology, Saxe-Coburg Publications, Stirling, 2010, pp. 19–48.

[50] M. Brovka, J. I. López, J. M. Escobar, J. M. Cascón, R. Montenegro, A new method for T-spline parameterization of complex 2D geometries, Engineering with Computers 30 (4) (2014) 457–473.

[51] M. Brovka, Construction of polynomial spline spaces over t-meshes for its application in isogeometric analysis, Ph.D. thesis, University of Las Palmas de Gran Canaria (2016).

[52] M. Brovka, J. I. López, J. M. Escobar, R. Montenegro, J. M. Cascón, A simple strategy for defining polynomial spline spaces over hierarchical t- meshes, Computer-Aided Design 72 (2016) 140–156.

[53] J. M. Cascón, R. Montenegro, J. M. Escobar, E. Rodríguez, G. Montero, A new meccano technique for adaptive 3-D triangulation, in: Proc. of the 16th International Meshing Roundtable, Springer, Berlin, 2007, pp. 103–120.

[54] J. M. Cascón, R. Montenegro, J. M. Escobar, E. Rodríguez, G. Montero, The meccano method for automatic tetrahedral mesh generation of complex genus-zero solids, in: Proc. of the 18th International Meshing Roundtable, Springer, Berlin, 2009, pp. 463–480.

[55] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J. M. González-Yuste, Simultaneous untangling and smoothing of tetrahedral meshes, Comput. Meth. Appl. Mech. Eng. 192 (2003) 2775–2787.

[56] J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, J. M. González-Yuste, SUS Code: Simultaneous mesh untangling and smoothing code, http://www.dca.iusiani.ulpgc.es/proyecto2012-2014/html/Software.html (2010).

[57] R. Montenegro, J. M. Cascón, J. M. Escobar, E. Rodríguez, G. Montero, An automatic strategy for adaptive tetrahedral mesh generation, Appl. Num. Math. 59 (2009) 2203–2217.

[58] H. Samet, Foundations of Multidimensional and Metric Data Structures, Morgan Kaufmann Publishers, Burlington, Massachusetts, 2006.

[59] S. A. Coons, Surfaces for computer aided design, Springfield, 1964.

[60] G. Farin, D. Hansford, Discrete Coons patches, Comput. Aid. Geom. Design 16 (1999) 691–700.

[61] P. M. Knupp, Algebraic mesh quality metrics, SIAM J. Sci. Comput. 23 (2001) 193–218.

[62] P. M. Knupp, A method for hexahedral mesh shape optimization, Int. J. Num. Meth. Eng. 58 (2) (2003) 319–332.

[63] L. A. Freitag, P. M. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, Int. J. Num. Meth. Eng. 53 (2002) 1377–1391.

[64] L. A. Freitag, P. Plassmann, Local optimization-based simplicial mesh untangling and improvement, Int. J. Num. Meth. Eng. 49 (2000) 109–125.

[65] J. M. Escobar, R. Montenegro, E. Rodríguez, G. Montero, Simultaneous aligning and smoothing of surface triangulations, Engineering with Computers 27 (2011) 17–29.

[66] J. Dennis, R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Englewood Cliffs, New Jersey, 1983.
URL `http://books.google.es/books?id=RtxcWd0eBD0C`

[67] J. M. Escobar, R. Montenegro, G. Montero, E. Rodríguez, J. M. González-Yuste, Smoothing and local refinement techniques for improving tetrahedral mesh quality, Computers & Structures 83 (2005) 2423–2430.

[68] D. Moore, The cost of balancing generalized quadtrees, in: Proceedings of the Third ACM Symposium on Solid Modeling and Applications, SMA '95, ACM, New York, NY, USA, 1995, pp. 305–312. doi:10.1145/218013.218078.

[69] M. S. Floater, Parametrization and smooth approximation of surface triangulations, Comput. Aid. Geom. Design 14 (1997) 231–250.

INSTITUTO UNIVERSITARIO
SIANI
INGENIERIA COMPUTACIONAL

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es