ARTICLE IN PRESS

Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A simple strategy for defining polynomial spline spaces over hierarchical T-meshes ☆

M. Brovka [a,*], J.I. López [a], J.M. Escobar [a], R. Montenegro [a], J.M. Cascón [b]

[a] University of Las Palmas de Gran Canaria, University Institute for Intelligent Systems and Numerical Applications in Engineering (SIANI), Spain

[b] University of Salamanca, Department of Economics and Economic History, Faculty of Economics and Business, Spain

HIGHLIGHTS

- A strategy for defining cubic tensor product spline functions is proposed.
- Simple rules for inferring local knot vectors to define blending functions for a given T-mesh.
- Examples of application of the strategy for adaptive refinement in isogeometric analysis and CAD.

ARTICLE INFO

Keywords:
Isogeometric analysis
Multivariate splines
Local refinement
T-mesh
Nested spaces

ABSTRACT

We present a new strategy for constructing spline spaces over hierarchical T-meshes with quad- and octree subdivision schemes. The proposed technique includes some simple rules for inferring local knot vectors to define C^2-continuous cubic tensor product spline blending functions. Our conjecture is that these rules allow to obtain, for a given T-mesh, a set of linearly independent spline functions with the property that spaces spanned by nested T-meshes are also nested, and therefore, the functions can reproduce cubic polynomials. In order to span spaces with these properties applying the proposed rules, the T-mesh should fulfill the only requirement of being a *0-balanced* mesh. The straightforward implementation of the proposed strategy can make it an attractive tool for its use in geometric design and isogeometric analysis. In this paper we give a detailed description of our technique and illustrate some examples of its application in isogeometric analysis performing adaptive refinement for 2D and 3D problems.

1. Introduction

The main drawback of using B-splines and NURBS for geometric design is the impossibility to perform local refinement. T-splines were introduced by Sederberg et al. [1] as an alternative to NURBS. Based on the idea of admitting meshes with T-junctions and inferring local knot vectors by traversing T-mesh edges, T-splines have provided a promising tool for geometric modeling that allows to perform local refinement without introducing a large number of superfluous control points. Later, in [2], T-splines were incorporated to the framework of isogeometric analysis. Isogeometric analysis (IGA) was introduced in 2005 by Hughes et al. in [3,4]. It has arisen as an attempt to unify the fields of CAD and classical finite element methods. The main idea of IGA consists in using for analysis the same functions that are used in CAD representation of the geometry.

To use spline functions for numerical analysis and obtain a proper convergence behavior, these functions must meet some requirements: linear independence, polynomial reproduction property, local supports and possibility to perform local adaptive refinement. This issue has been the object of numerous research works in recent years.

Analysis-suitable T-splines, proposed by Scott et al. in [5], are a class of T-splines defined over T-meshes that should meet certain topological restrictions formulated in terms of T-junction extensions. Basis functions defined over an extended analysis-suitable T-mesh are linearly independent [6] and form a partition of unity. The refinement algorithm allows to accomplish highly localized refinements and constructs 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 approach to the problem of local enrichment of the approximation space is the hierarchical refinement, originally

☆ This paper has been recommended for acceptance by Matthew L. Staten.

* Corresponding author.
E-mail address: mbrovka@siani.es (M. Brovka).

http://dx.doi.org/10.1016/j.cad.2015.06.008

introduced by Forsey and Bartels in [7] and later developed in [8]. Recently, hierarchical refinement technique in the context of isogeometric analysis was described in [9–11]. This approach is based on a simple and 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. Simplicity of its implementation and straightforward generalization to 3D make it an attractive option for local refinement. However, a drawback 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 [12]. The truncation technique is applied to redefine the function supports and reduce their overlapping.

Other options for performing local refinement of spline spaces are C^1-continuous PHT-splines [13] or local refined splines (LR-splines) [14].

In the present paper we propose another possible alternative for the construction of spline functions that span spaces with nice properties. The technique we present here is designed for hierarchical T-meshes (multilevel meshes) with a quad- and octree subdivision scheme. This type of meshes can be efficiently implemented with tree data structures [15] which are frequently used in engineering. Due to the elevated complexity of all current strategies, the main goal we pursue here is the simplicity and low computational cost of the implementation, both in 2D and 3D. For that, we have to assume a restriction on the T-mesh. Namely, the T-mesh should fulfill the requirement of being a 0-*balanced* mesh. A balanced mesh condition is usually imposed to have gradual transition from the coarse mesh to the finer zones. In addition, for our technique, this condition is an obligatory prerequisite that the T-mesh should fulfill. Assuming this reasonable restriction over the T-mesh, we can define easily cubic spline functions that span spaces with desirable properties: linear independence, C^2-continuous, cubic polynomial reproduction property, nestedness of spanned spaces and a straightforward implementation. The key of the strategy lies in some simple rules used for inferring local knot vectors for each blending function.

The paper is organized as follows. Some basic concepts about B-splines and T-meshes are given in Section 2. Section 3 includes the general scheme of our strategy and the description of its main stages. In Section 4 we explain in detail the key of our technique, that is, the rules used for inferring function supports in order to span spaces with desirable properties. In Section 5 the properties of the defined functions are given and the issue of support overlapping and sparsity of stiffness matrix is discussed. Computational examples of performing adaptive refinement for 2D and 3D Poisson problems are presented in Section 6. Conclusions are given in Section 7.

2. Basic concepts

2.1. B-spline basis functions

We start with 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

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

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

A knot vector Ξ is called open knot vector if the first and the last knots are repeated $p+1$ times. At each knot of multiplicity m the basis functions are C^{p-m}.

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

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

where coefficients $\mathbf{P}_i \in \mathbb{R}^s$ are called control points, typically $s = 2$ or 3.

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 $I = \{1, 2, \ldots, n_1\} \times \cdots \times \{1, 2, \ldots, n_d\}$.

A B-spline surface (solid) is defined as a linear combination of bivariate (trivariate) B-spline functions

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

where the control points $\mathbf{P}_{\boldsymbol{i}} \in \mathbb{R}^s$ ($s = 2$ or 3) form a control mesh.

For more details about B-splines see [16].

2.2. T-meshes and T-splines

In order to overcome the drawback of tensor product structure, which does not allow to perform local refinement, it is necessary to admit T-junctions in the mesh. The concept *T-junction* is similar to *hanging node* in the classical finite element method. An axes-aligned grid that allows T-junctions is called T-mesh. As was mentioned in the previous section, there are several strategies to define tensor product spline functions over T-meshes and one of these strategies is T-splines. The underlying idea of T-splines consists in defining 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. A local knot vector for each bivariate function B_α is inferred by traversing the T-mesh edges in both parametric directions starting from a vertex v_α of the mesh (the *anchor*), see Fig. 1. For a pair of local knot vectors $\Xi_\alpha^j = \left(\xi_1^j, \xi_2^j, \xi_3^j, \xi_4^j, \xi_5^j\right)$, $j = 1, 2$ the bicubic spline function B_α is defined as $B_\alpha(\xi^1, \xi^2) = B[\Xi_\alpha^1](\xi^1) B[\Xi_\alpha^2](\xi^2)$, where $B[\Xi_\alpha^j](\xi^j)$ is an univariate B-spline corresponding to the knot vector Ξ_α^j. In general, T-spline blending functions do not span a polynomial space. Some additional restrictions on the T-mesh configuration [5] should be satisfied in order to span a polynomial spline space. If these restrictions are not verified, the T-splines should be normalized in order to form a partition of unity. This leads to rational blending functions: $R_\alpha(\xi^1, \xi^2) = \frac{B_\alpha(\xi^1, \xi^2)}{\sum_{\beta \in A} B_\beta(\xi^1, \xi^2)}$, where A is the index set of the basis spanned by the T-mesh. These rational blending functions are capable of reproducing a constant function, but, in general, cannot reproduce a polynomial of a higher order. A T-spline approximation is constructed as a linear combination of all blending functions: $S(\xi^1, \xi^2) = \sum_{\alpha \in A} P_\alpha \, R_\alpha(\xi^1, \xi^2)$.

3. Strategy for the construction of polynomial spline spaces over hierarchical T-meshes

In this section we describe our strategy to define tensor product spline functions over hierarchical T-meshes. The strategy we propose has some similarity with T-splines in as much as we define the blending functions from local knot vectors that are