Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 342 (2018) 32–45
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Local refinement for analysis-suitable++ T-splines

Jingjing Zhang[a,b], Xin Li[a,*]

[a] *University of Science and Technology of China, Hefei, Anhui, PR China*
[b] *Hefei Technological University, Hefei, Anhui, PR China*

Received 24 January 2018; received in revised form 15 July 2018; accepted 18 July 2018
Available online 30 July 2018

Abstract

We develop an optimized local refinement algorithm for analysis-suitable++ T-splines (AS++ T-splines) which produces less propagation of control points. We then demonstrate its use as an adaptive framework for isogeometric analysis. AS++ T-splines are a class of T-splines which include analysis-suitable T-spline as a special case, are linearly independent, form a weighted partition of unity and are optimal convergent. These properties, coupled with the local refinement algorithm, make the AS++ T-spline appealing as a better basis both for design and analysis.

Keywords: T-splines; Linear independence; Isogeometric analysis; Analysis-suitable T-splines; Analysis-suitable++ T-splines

1. Introduction

T-splines [1,2] have proved to be an important technology for geometric modeling because of their supporting several very significant operations, such as local refinement [2,3], watertightness and trimmed NURBS conversion [1,4,5]. T-splines are attractive not only in geometric modeling but also in iso-geometric analysis (IGA), which uses the smooth spline basis that defines the geometry as a basis for analysis [6,7]. Although the whole class of T-splines are not suitable as a basis for IGA because of possible linear dependence [8], a mildly topological restricted subset of T-splines, analysis-suitable T-splines (AS T-splines), are optimized to meet the needs both for design and analysis [9,3,10–13]. Thus, the use of T-splines as the basis for IGA has gained widespread attention [14–24].

Analysis-suitable++ T-splines (for short, AS++ T-splines) are T-splines which define on a class of T-meshes with less restriction than AS T-splines, i.e., AS++ T-splines include AS T-splines as a special case. And meanwhile, AS++ T-splines maintain all the good properties as AS T-splines: they are linear independent [25], are NURBS compatible, obey the convex hull property, provide watertight models, and are weighted partition of unity (the constant function belongs to the AS++ T-spline space)—all the important properties for isogeometric analysis.

This paper contains two main contributions. First, we prove that the AS++ T-spline space is closed under all the existing local refinement algorithms, i.e., applying any existing local refinement algorithms in [2,3] on an AS++

* Corresponding author.
E-mail address: lixustc@ustc.edu.cn (X. Li).

https://doi.org/10.1016/j.cma.2018.07.024

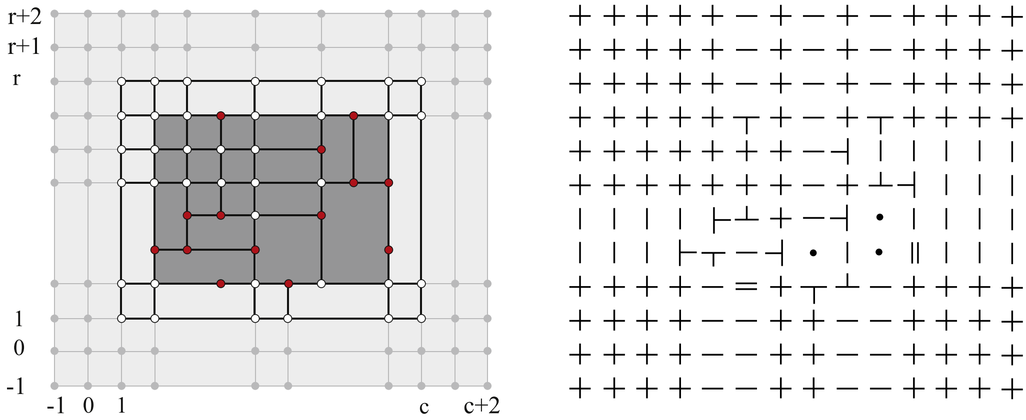

Fig. 1. An example T-mesh and the associated symbolic T-mesh.

T-spline will produce another AS++ T-spline. And second, we develop a highly localized refinement algorithm for AS++ T-splines which tries to minimize the number of additional control points. As the algorithm in [2] is a recursive procedure to refine all the possible influence blending functions without any optimization and the AS++ T-spline spaces are closed under the algorithm in [2], so our new algorithm is an improvement of the algorithm in [2] because our algorithm is an optimized algorithm for all the possible local refinement results in the AS++ T-spline space. The numerical examples and comparisons also indicate that our new algorithm has less propagation than that in [2]. On the other hand, AS++ T-splines have less restrictions than AS T-splines. So the new algorithm is also an improvement of the algorithm in [3]. And the numerical examples show that the new algorithm produces less propagation in most cases.

The rest of the paper is structured as follows. In Section 2, we recall some basic notations for bi-cubic T-splines. In Section 3, we provide the definition of the AS++ T-spline space and prove that the space is closed under the local refinement algorithm in [2]. In Section 4, we give the details of the new optimized local refinement algorithm. The comparisons and the application in the IGA will be discussed in Section 5. Section 6 is the conclusion and future work.

2. T-meshes and T-splines

This section reviews the basic concept for bi-cubic T-splines.

2.1. Index T-mesh

An index T-mesh [14] T for a bi-cubic T-spline is a collection of all the elements of a rectangular partition of the index domain $[-1, c+2] \times [-1, r+2]$, where all rectangle corners (or vertices) have integer coordinates. In the following, (σ_i, τ_i) or $\{\sigma_i\} \times \{\tau_i\}$ denotes a vertex in T, and $[\sigma_j, \sigma_k] \times \{\tau_i\}$ ($\{\sigma_i\} \times [\tau_j, \tau_k]$) denotes a horizontal (vertical) edge or a set of connected horizontal (vertical) edges. Denote $[\sigma_i, \sigma_j] \times [\tau_k, \tau_l]$ ($(\sigma_i, \sigma_j) \times (\tau_k, \tau_l)$) to be a closed (open) face.

A symbolic T-mesh [9] is created by assigning a symbol to each element in a 2-D array formed from the index T-mesh T, see Fig. 1 as an example. The symbol is chosen to match the mesh topology of T. We adopt the symbol '$+$' to indicate valence four vertex, corner vertex, or valence three boundary vertex, symbols '$\vdash$', '$\dashv$', '$\perp$' and '$\top$' to indicate the four possible orientations for the T-junctions, symbols '$\|$' and '$=$' to indicate two possible orientations of the I-junctions, symbols '$|$' and '$-$' to indicate a vertical and a horizontal edge, and symbol '.' to indicate no vertices and edges. In the following, the interior vertices can only be I-junctions, T-junctions and valence four vertices.

For the ith vertex $\mathbf{V}_i = (\sigma_i, \tau_i)$ in the rectangle $[1, c] \times [1, r]$, we define a local index vector $\vec{\sigma}_i \times \vec{\tau}_i$, where $\vec{\sigma}_i = [\sigma_i^0, \ldots, \sigma_i^4]$ and $\vec{\tau}_i = [\tau_i^0, \ldots, \tau_i^4]$. From the vertex, we shoot a ray in both directions traversing the T-mesh and collect a set of knot indices $\{\sigma_i^j\}$ and $\{\tau_i^k\}$ in both directions such that $\sigma_i^2 = \sigma_i$ and $\tau_i^2 = \tau_i$, as shown in Fig. 2.