Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2017.0003

Vol. **10**, No. 4, pp. 760-774
November 2017

Evaluation Algorithm of PHT-Spline Surfaces

Zhihua Wang[1,2], Falai Chen[1,*]and Jiansong Deng[1]

[1] *School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, P. R. China*
[2] *School of Mathematics and Computer Science, Anqing Normal University, Anqing 246011, P. R. China*

Received 9 January 2017; Accepted (in revised version) 22 March 2017

Abstract. PHT-splines are a type of polynomial splines over hierarchical T-meshes which posses perfect local refinement property. This property makes PHT-splines useful in geometric modeling and iso-geometric analysis. Current implementation of PHT-splines stores the basis functions in Bézier forms, which saves some computational costs but consumes a lot of memories. In this paper, we propose a de Boor like algorithm to evaluate PHT-splines provided that only the information about the control coefficients and the hierarchical mesh structure is given. The basic idea is to represent a PHT-spline locally in a tensor product B-spline, and then apply the de-Boor algorithm to evaluate the PHT-spline at a certain parameter pair. We perform analysis about computational complexity and memory costs. The results show that our algorithm takes about the same order of computational costs while requires much less amount of memory compared with the Bézier representations. We give an example to illustrate the effectiveness of our algorithm.

AMS subject classifications: TP391

Key words: PHT-splines, hierarchical T-mesh, knot insert, evaluation algorithm.

1. Introduction

In recent years, there have been much interest in locally refinable splines in the community of geometric modeling and iso-geometric analysis. Various such type of splines have been proposed, such as hierarchical B-splines, T-splines, PHT-splines, AST-splines, THB-splines, RHT-splines and LR-splines, etc [1–13, 17]. Compared with tensor product B-splines, locally refinable splines support local editing and eliminate superfluous control points in surface modeling. Furthermore, they are attractive in adaptive finite element methods and iso-geometric analysis.

In this paper, we focus on evaluation of PHT-splines–polynomial splines over hierarchical T-meshes. PHT-splines were firstly introduced by two of the present authors for adaptive surface modeling such as surface fitting, model simplification, etc [5]. The basis

*Corresponding author. *Email address:* chenfl@ustc.edu.cn (F. L. Chen)

 ©2017 Global-Science Press

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

functions of PHT-splines posses many nice properties like B-spline functions, such as linear independency, partition of unity, compact support, nonnegativity, etc. Most of all, the basis functions of PHT-splines have perfect local refinement property, which makes PHT-splines very useful in adaptive surface modeling and iso-geometric analysis. PHT-splines have been successfully applied in solving partial differential equations [14], iso-geometric analysis [15–19], surface stitching [6] and implicit surface reconstructions [20]. Later on, PHT-splines were generalized over general T-meshes with arbitrary topology (GPT-splines, [7]). Theoretic study on the dimension of spline spaces over T-meshes can be found in [21–28].

For the current implementation of PHT-splines, the basis functions are computed and stored in Bézier forms beforehand [5]. The advantage is that computation with such representation is relatively efficient. Yet, the memory consumption is very large, especially for models with fine details and 3D implicit models [5,20]. To overcome such problem, in this paper we extend the de Boor algorithm for tensor-product B-splines to PHT-splines, where only the control coefficients and the hierarchical mesh structure are given. We perform analysis about computational complexity and memory consumption, and the results show that our algorithm takes about the same order of computational costs while requires much less amount of memory compared with the Bézier representations.

The paper is organized as follows. In Section 2, we review some preliminary knowledge about PHT-splines and the detailed basis construction process. Section 3 provides algorithm outline and Section 4 presents detailed algorithm. Section 5 gives an example to illustrate the algorithm. Section 6 analyzes the algorithm complexity together with a comparison with Bézier representations. We conclude the paper in Section 7.

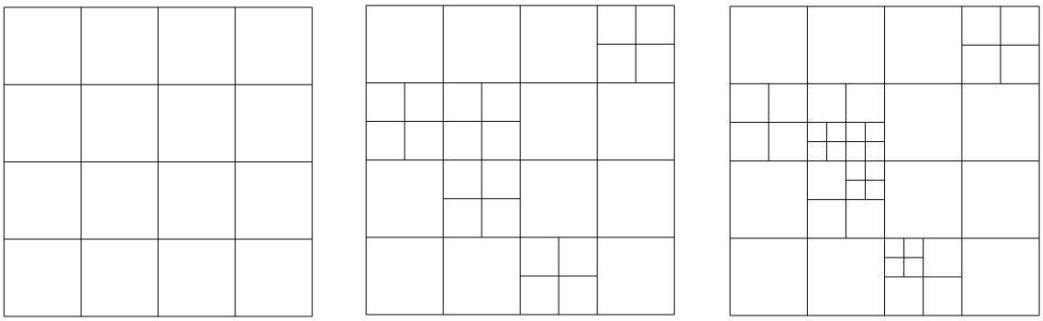
2. PHT-splines

In this section, we recall some preliminary knowledge about PHT-splines and present detailed construction of the basis functions of PHT-splines.

2.1. Hierarchical T-mesh

A T-mesh is a rectangular grid that allows T-junctions on a 2D plane. In this paper, we adopt the same definitions of vertices, edges and cells as as in [5]. A *T-junction* is a vertex that lies in the interior of another edge. A *crossing-vertex* is the intersection of two interior edges. A *hierarchical T-mesh* is a special type of T-mesh that has a natural level structure. It is defined recursively. Generally, we start from a tensor-product mesh $\mathscr{T}_0$, which can be non-uniform and the elements (vertices, edges and cells) of which are called the level 0 elements. From level k to level $k+1$, some cells at level k are subdivided, with each cell is subdivided into 2×2 subcells, where the new vertices, the new edges and the new cells have level $k+1$, and the resulting T-mesh is called the level $k+1$ T-mesh, denoted as $\mathscr{T}_{k+1}$. Fig. 1 illustrates the process of generating a hierarchical T-mesh.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press



(a) (b) (c)

Figure 1: A hierarchical T-mesh. (a): Level 0; (b): Level 1; (c): Level 2.

2.2. Splines over T-meshes

In [21], Deng et al. introduced the concept of polynomial splines over T-meshes. Given a T-mesh $\mathscr{T}$, denote Ω as the region occupied by all the cells in $\mathscr{T}$, which is called the domain of $\mathscr{T}$. Define

$$\mathscr{S}(m,n,\alpha,\beta,\mathscr{T}) := \{f(x,y) \in C^{\alpha,\beta}(\Omega) : f(x,y)|_{\phi} \in \mathbb{P}_{m,n}, \quad \forall \phi \in \mathscr{T}\}, \tag{2.1}$$

where $\mathbb{P}_{m,n}$ is the set of all the tensor product polynomials with bi-degree (m,n), $C^{\alpha,\beta}$ is the space consisting of all the bivariate functions in Ω with continuity of order α along the x-direction and with order β along the y-direction, and $\phi \in \mathscr{T}$ means that ϕ is a cell of $\mathscr{T}$. It is obvious that $S(m,n,\alpha,\beta,\mathscr{T})$ is a linear space, and it is called the *spline space over the T-mesh* $\mathscr{T}$. Especially, when $\mathscr{T}$ is a hierarchical T-mesh, then $S(m,n,\alpha,\beta,\mathscr{T})$ is called a *PHT-spline space*, short for polynomial splines over hierarchical T-meshes.

When $m \geq 2\alpha+1$ and $n \geq 2\beta+1$, the dimension of the spline space $\mathscr{S}(m,n,\alpha,\beta,\mathscr{T})$ can be explicitly computed.

Proposition 2.1. ([21]) *Given a regular T-mesh $\mathscr{T}$ and suppose $m \geq 2\alpha+1$ and $n \geq 2\beta+1$. Then*

$$\begin{aligned} dim\ \mathscr{S}(m,n,\alpha,\beta,\mathscr{T}) =& F(m+1)(n+1) - E_h(m+1)(\beta+1) \\ & - E_v(\alpha+1)(n+1) + V(\alpha+1)(\beta+1), \end{aligned} \tag{2.2}$$

where F, E_h, E_v and V are the number of cells, horizontal interior edges, vertical interior edges and interior vertices of $\mathscr{T}$.

A representative spline space is $\mathscr{S}(3,3,1,1,\mathscr{T})$ whose dimension is simply $4(V^b+V^+)$, where V^b, V^+ are respectively the number of boundary vertices and the crossing vertices of $\mathscr{T}$. For this case, the boundary vertices and crossing vertices are called *basis vertices* of the T-mesh $\mathscr{T}$. Note that, according to the dimension formula, each basis vertex should correspond to four basis functions.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

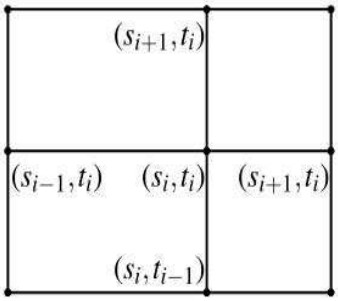


Figure 2: A new basis vertex and its neighboring cells.

2.3. The basis functions of PHT-splines

In this subsection, we briefly recall the construction of the basis functions of PHT-splines. Instead of going to general cases where $m \geq 2\alpha + 1$, $n \geq 2\beta + 1$, we focus our attention on a representative spline space $\mathscr{S}(3, 3, 1, 1, \mathscr{T})$. The details can be found in [5].

The basis functions for the spline space $\mathscr{S}(3, 3, 1, 1, \mathscr{T})$ are constructed level by level. Denote $\mathscr{T}_k$ for the T-mesh at level k. For level 0, the standard tensor product B-spline basis functions (of C^1 continuous bicubic B-spline space) are taken to be the basis functions over the initial mesh $\mathscr{T}_0$, where every basis vertex is associated with four basis functions. Now suppose the basis functions $\{b_j^k(s,t)\}$, $j = 1, \cdots, d_k$ at level k have been constructed, then the basis functions at level $k+1$ consist of two parts. The first part comes from the modification of the basis functions at level k. The second part comes from the basis functions associated with the new basis vertices at level $k+1$.

- *Modification of the basis functions at level k*

  Suppose, among all the cells at level k, the cells $\theta_i^k (i = 1, \cdots, C_k)$ are subdivided. For each j, if the basis function $b_j^k(s,t)$ does not vanish in some cells of $\{\theta_i^k\}$, then subdivide $b_j^k(s,t)$ into these cells at level $k+1$ according to formula (14.15) in [29]. From $\mathscr{T}_k$ to $\mathscr{T}_{k+1}$, some new basis vertices appear. Denote them as ξ_i^{k+1}, $i = 1, \cdots, V_{k+1}$. In each cell at level $k+1$, the 16 Bézier ordinates are divided into four parts. Then all the basis functions $\{b_j^k(s,t)\}$ at level k are modified to $\{\bar{b}_j^k(s,t)\}$ in the following fashion: for each j, reset all the associated Bézier ordinates with the new basis vertices to zero.

- *Adding new basis functions at level k + 1*

  For each new basis vertex (s_i, t_i), its neighboring cells must lie in a position as shown in Fig. 2. Then the four basis functions associated with (s_i, t_i) are defined to be $M_{ik}^3(s)N_{il}^3(t)$, $k, l = 1, 2$, where $M_{i1}^3(s), M_{i2}^3(s)$, $N_{i1}^3(t)$ and $N_{i2}^3(t)$ are respectively the cubic B-spline basis functions associated with the knot vectors $(s_{i-1}, s_{i-1}, s_i, s_i, s_{i+1})$, $(s_{i-1}, s_i, s_i, s_{i+1}, s_{i+1})$, $(t_{i-1}, t_{i-1}, t_i, t_i, t_{i+1})$ and $(t_{i-1}, t_i, t_i, t_{i+1}, t_{i+1})$. And if (s_i, t_i) is a boundary vertex, then either $s_{i-1} = s_i$, or $s_{i+1} = s_i$, or $t_{i-1} = t_i$, or $t_{i-1} = t_i$.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

Now suppose we have constructed a basis $b_{ij}(s,t)$, $i=1,\cdots,n$, $j=1,\cdots,4$ for the spline space $S(3,3,1,1)$, where n is the number of basis vertices, then a PHT-spline surface is defined by

$$\mathbf{P}(s,t)=\sum_{i=1}^{n}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}(s,t),\quad (s,t)\in\Omega, \tag{2.3}$$

where $\mathbf{C}_{ij}$, $i=1,\cdots,n$, $j=1,\cdots,4$ are control points.

In the following sections, we present an algorithm to compute $\mathbf{P}(s,t)$ at a specific pair of parametric values (s_0,t_0).

3. Algorithm outline

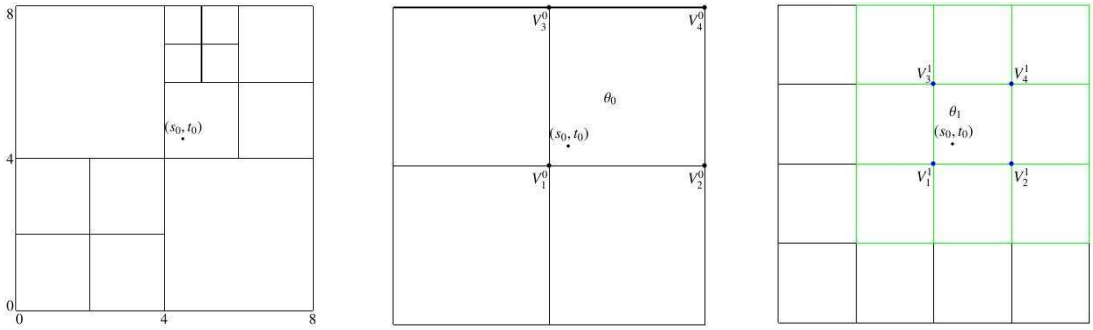
Instead of going to the general PHT-spline spaces $\mathcal{S}(m,n,\alpha,\beta)$ with $m\geq 2\alpha+1$ and $n\geq 2\beta+1$, in this paper, we only present the results for the specific spline space $\mathcal{S}(3,3,1,1)$ for two reasons. First, the algorithm for general cases is totally similar with the algorithm for the spline space $\mathcal{S}(3,3,1,1)$ but is more complicated and tedious to describe. Thus for simplicity we stay with $\mathcal{S}(3,3,1,1)$. Second, the spline space $\mathcal{S}(3,3,1,1)$ has been widely used and is sufficient for many practical applications. In the following, we outline the main procedure of the algorithm in this section before presenting the details in the next section. We start with some notations.

Let $\mathcal{T}$ be a hierarchical T-mesh with maximum level of hierarchy N, and $\mathcal{T}_k$ be the corresponding level k T-mesh. Then, $\mathcal{T}=\mathcal{T}_N$ and the initial mesh $\mathcal{T}_0$ is a tensor product mesh. For a specific pair of parameters (s_0,t_0) inside $\mathcal{T}$, define a new *hierarchical T-mesh $\hat{\mathcal{T}}$ associated with the parameter pair* (s_0,t_0) recursively as follows. First define $\hat{\mathcal{T}}_0=\mathcal{T}_0$. Assume (s_0,t_0) lies in the cell θ_0 in $\hat{\mathcal{T}}_0$, θ_0 is called the *the cell associated with* (s_0,t_0) *at level* 0. The 3×3 tensor product mesh L_0 inside $\hat{\mathcal{T}}_0$ with θ_0 being the central cell is called the *local tensor product mesh associated with* (s_0,t_0) at level 0. Now suppose that at level k, the new hierarchical T-mesh $\hat{\mathcal{T}}_k$, the central cell θ_k and the local tensor product mesh L_k associated with (s_0,t_0) are defined. These terminologies at level $k+1$ are defined as follows. $\hat{\mathcal{T}}_{k+1}$ is the T-mesh obtained by subdividing $\hat{\mathcal{T}}_k$ in the same way as subdividing $\mathcal{T}_k$ to get $\mathcal{T}_{k+1}$, and then extend the T-junctions of the mesh such that a 3×3 local tensor product mesh is formed with the central cell containing (s_0,t_0). The central cell θ_{k+1} is the *cell associated with* (s_0,t_0) *at level* $k+1$, and the 3×3 tensor product mesh L_{k+1} is the *local tensor product mesh associated with* (s_0,t_0) *at level* $k+1$. This process terminates at some maximum level M of hierarchy associated with the cell over $\mathcal{T}$ which contains (s_0,t_0).

Fig. 3 illustrates the construction of the hierarchical T-mesh together with the cells and the local tensor product meshes at different levels associated with parameter pair (s_0,t_0). Note that, if θ_k is at the boundary or the corner, the corresponding local tensor product mesh degenerates to a 2×3 (or 3×2) or 2×2 mesh.

We especially notice that, for some types of hierarchical T-meshes which often appear in scientific computing, the hierarchical T-meshes associated with a parametric pair (s_0,t_0) is exactly the same as the original hierarchical T-meshes. As illustrated in Fig. 4, the local

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press



(a) (b) (c)

Figure 3: (a): A hierarchical T-mesh $\mathscr{T}$ and a parameter pair (s_0, t_0) inside $\mathscr{T}$; (b): The central cell θ_0, the local tensor product mesh and the hierarchical T-mesh $\hat{\mathscr{T}}_0$ associated with (s_0, t_0) at level 0; (c): The central cell θ_1, the local tensor product mesh L_1 (green mesh) and the hierarchial T-mesh $\hat{\mathscr{T}}_1$ associated with (s_0, t_0) at level 1.

tensor product T-mesh L_k associated with (s_0, t_0) at different levels $k = 0, 1, 2$ is contained in the original level k T-mesh $\mathscr{T}_k$, respectively. In this case, $\hat{\mathscr{T}}_k = \mathscr{T}_k$.

Now the basic idea to evaluate a PHT-spline $\mathbf{P}(s, t)$ at a specific pair of values (s_0, t_0) is to express $\mathbf{P}(s, t)$ as a bicubic B-spline surface $\mathbf{P}_k(s, t)$ at the local tensor product mesh L_k associated with (s_0, t_0) recursively from level 0 to some level $M (M \leq N)$. Evaluating the final level bicubic B-spline surface $\mathbf{P}_M(s, t)$ at the parametric values (s_0, t_0) gives the value of $\mathbf{P}(s_0, t_0)$. We use an example to illustrate the detailed process.

Suppose a hierarchical T-mesh $\mathscr{T}$ defined over $[0, 8] \times [0, 8]$ and a pair of parameter values $(s_0, t_0) = (4.5, 4.5)$ are provided as shown in Fig. 3, and a PHT-spline $\mathbf{P}(s, t)$ is defined over $\mathscr{T}$ with given control points. We are going to compute $\mathbf{P}(4.5, 4.5)$.

First, find the cell associated with the parameters $(s_0, t_0) = (4.5, 4.5)$ at level 0–$\theta_0 = [4, 8] \times [4, 8]$. In this case, the local tensor product mesh $L_0 = [0, 8] \times [0, 8]$ whose knot vectors are $(0, 0, 4, 4, 8, 8, 8, 8) \times (0, 0, 4, 4, 8, 8, 8, 8)$. A tensor product bicubic B-spline over L_0 is defined:

$$\mathbf{P}_0(s, t) := \sum_{i=1}^{4} \sum_{j=1}^{4} \mathbf{C}_{ij}^0 M_{ij}^0(s, t)$$

where M_{ij}^0 are the tensor product bicubic B-spline basis functions associated with the knot vectors $(0, 0, 4, 4, 8, 8, 8, 8) \times (0, 0, 4, 4, 8, 8, 8, 8)$, and $\mathbf{C}_{ij}^0 = \mathbf{C}_{ij}$ are the control points corresponding to the four corners $\{V_i^0\}_{i=1}^4$ of the cell θ_0.

Next find the cell at level 1 associated with the parameters $(s_0, t_0) = (4.5, 4.5)$–$\theta_1 = [4, 6] \times [4, 6]$ and the local tensor product mesh associated with $(s_0, t_0) = (4.5, 4.5)$–$L_1 = [2, 8] \times [2, 8]$ whose knot vectors are $(2, 2, 4, 4, 6, 6, 8, 8) \times (2, 2, 4, 4, 6, 6, 8, 8)$. Express $\mathbf{P}_0(s, t)$ over L_1 as a bicubic B-spline surface $\hat{\mathbf{P}}_1(s, t)$ by subdividing $\mathbf{P}_0(s, t)$ at $s, t = 2, 6$, and assume

$$\hat{\mathbf{P}}_1(s, t) := \sum_{i=1}^{4} \sum_{j=1}^{4} \hat{\mathbf{C}}_{ij}^1 M_{ij}^1(s, t), \quad (s, t) \in \theta_1,$$

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

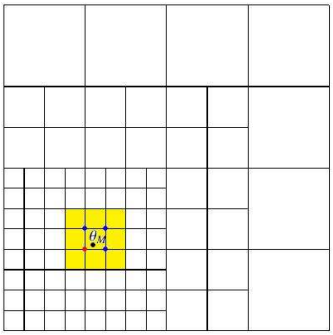


Figure 4: The local tensor product mesh (yellow mesh) is contained in $\mathscr{T}_M$.

where M_{ij}^1 are the tensor product bicubic B-spline basis functions associated with the knot vectors $(2,2,4,4,6,6,8,8)\times(2,2,4,4,6,6,8,8)$, and the control points $\hat{\mathbf{C}}_{ij}^1$ are associated with the four corner vertices $\{V_i^1\}_{i=1}^4$ of θ_1. Now define a tensor product bicubic B-spline surface defined over L_1:

$$\mathbf{P}_1(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{4}\mathbf{C}_{ij}^1 M_{ij}^1(s,t)$$

by modifying the control points of $\hat{\mathbf{P}}_1(s,t)$ as follows. If V_i^1 is a new basis vertex at level 1, then let $\mathbf{C}_{ij}^1$ be the control points of $\mathbf{P}(s,t)$ associated with V_i^1. Otherwise, let $\mathbf{C}_{ij}^1 = \hat{\mathbf{C}}_{ij}^1$, $j = 1,2,3,4$. In our example, V_4^1 is the new basis vertex, so update the corresponding control points with the control points of $\mathbf{P}(s,t)$ associated with V_4^1. The other control points remain unchanged.

Finally evaluate $\mathbf{P}_1(s,t)$ at the parametric values $(s,t) = (4.5,4.5)$ using de Boor's algorithm gives the value $\mathbf{P}(4.5,4.5)$ of the PHT-spline surface.

4. Evaluation algorithm

4.1. Evaluation algorithm

In this section, we introduce the evaluation algorithm for PHT-spline surfaces in detail and verify its correctness.

4.2. Correctness of the algorithm

We claim that the algorithm presented in the last subsection correctly computes the value of the PHT-spline surface $\mathbf{P}(s,t)$ at (s_0,t_0).

Theorem 4.1. *For any $(s,t)\in\theta_M$, $\mathbf{P}(s,t) = \mathbf{P}_M(s,t)$. That is, the algorithm presented in the last subsection correctly computes the value of a PHT-spline surface at a specific parameter pair $(s_0,t_0)\in\theta_M$.*

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

Algorithm 4.1 the evaluation algorithm for PHT-spline surfaces.

Input: A PHT-spline surface $\mathbf{P}(s,t)$ with given control points defined over a hierarchical T-mesh $\mathscr{T}$, and a pair of parametric values (s_0,t_0) inside $\mathscr{T}$.

Output: The value of PHT-spline surface $\mathbf{P}(s,t)$ at (s_0,t_0).

1. Determine the cell θ_0 and the local tensor product mesh L_0 associated with (s_0,t_0) at level 0. A tensor product bicubic B-spline over L_0 is defined:

$$\mathbf{P}_0(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{4} \mathbf{C}_{ij}^0 M_{ij}^0(s,t),$$

where $\mathbf{C}_{ij}^0$ are the control points of $\mathbf{P}(s,t)$ at the four corners $\{V_i^0\}_{i=1}^4$ of θ_0, and M_{ij}^0 are the tensor product B-spline basis functions defined over L_0.

2. For $k=1$ to M, repeat the following step. Here M is the maximum level of hierarchy associated with the cell which contains (s_0,t_0).

3. Determine the cell θ_k and the local tensor product mesh L_k associated with (s_0,t_0) at level k. A tensor product bicubic B-spline surface over L_k is computed:

$$\mathbf{P}_k(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{4} \mathbf{C}_{ij}^k M_{ij}^k(s,t),$$

where M_{ij}^k are the tensor product B-spline basis functions defined over L_k, and $\mathbf{C}_{ij}^k$ are control points at the four corners $\{V_i^k\}_{i=1}^4$ of θ_k which can be computed by subdividing $\mathbf{P}_{k-1}(s,t)$ over L_k. Assume

$$\mathbf{P}_{k-1}(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{4} \hat{\mathbf{C}}_{ij}^k M_{ij}^k(s,t), \quad (s,t)\in\theta_k.$$

If V_i^k is a new basis vertex at level k, then set the corresponding control points $\mathbf{C}_{ij}^k$ to be the control points of $\mathbf{P}(s,t)$ associated with V_i^k. Otherwise, just set $\mathbf{C}_{ij}^k = \hat{\mathbf{C}}_{ij}^k$, $j=1,2,3,4$.

4. Evaluate $\mathbf{P}_M(s_0,t_0)$ using de-Boor's algorithm. Output the value.

Proof. Assume that $\{V_i^k\}_{i=1}^{m_k}$ are all the basis vertices of level k T-mesh $\mathscr{T}_k$, and $\{b_{ij}^k(s,t)\}_{j=1}^4$ are the four basis functions associated with V_i^k. The basis vertices $\{V_i^{k+1}\}_{i=1}^{m_{k+1}}$ at level $k+1$ T-mesh $\mathscr{T}_{k+1}$ consist of two parts: the basis vertices $\{V_i^k\}_{i=1}^{m_k}$ at level k, and the new basis vertices $\{V_i^{k+1}\}_{i=m_k+1}^{m_{k+1}}$. Here we assume $V_i^{k+1} = V_i^k$ for $i=1,2,\cdots,m_k$.

Now define

$$\mathbf{Q}_k(s,t) := \sum_{i=1}^{m_k}\sum_{j=1}^{4} \mathbf{C}_{ij} b_{ij}^k(s,t), \qquad (s,t)\in\Omega.$$

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

Then

$$\mathbf{Q}_k(s,t)|_{\theta_k} = \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^k(s,t),$$

where the index set I_k is defined by

$$I_k := \{i:\ \text{the support of } b_{ij}^k(s,t) \text{ contains } \theta_k,\ \ j=1,2,3,4.\}.$$

We are going to show that

$$\mathbf{Q}_k(s,t)|_{\theta_k} = \mathbf{P}_k(s,t)|_{\theta_k}, \quad k=0,1,\cdots,M, \tag{4.1}$$

and consequently, for any $(s,t)\in\theta_M$,

$$\mathbf{P}(s,t)|_{\theta_M} = \mathbf{Q}_M(s,t)|_{\theta_M} = \mathbf{P}_M(s,t)|_{\theta_M},$$

which is what we want to prove. Now we prove (4.1) by mathematical induction on k.

For $k=0$, $\mathbf{Q}_0(s,t)$ is a bicubic B-spline surface defined on $\mathcal{T}_0$. Obviously $\mathbf{Q}_0(s,t)|_{\theta_0} = \mathbf{P}_0(s,t)|_{\theta_0}$. Now suppose that (4.1) holds for k. We are going to prove that (4.1) holds for $k+1$.

As $\mathcal{T}_k$ is subdivided to $\mathcal{T}_{k+1}$, the basis functions $\{b_{ij}^k(s,t)\}_{i\in I_k}$ are modified to the basis functions $\{b_{ij}^{k+1}(s,t)\}_{i\in I_k}$ at level $k+1$. By the basis functions construction method described in [5], $b_{ij}^k(s,t)$ is divided into two parts:

$$b_{ij}^k(s,t) = b_{ij}^{k+1}(s,t) + \hat{b}_{ij}^{k+1}(s,t), \quad i\in I_k,\ \ (s,t)\in\theta_k,$$

where $\hat{b}_{ij}^{k+1}(s,t)$ is the part of $b_{ij}^k(s,t)$ whose Bézier ordinates are set to zero when $b_{ij}^k(s,t)$ is modified to $b_{ij}^{k+1}(s,t)$. Then

$$\mathbf{Q}_k(s,t)|_{\theta_k} = \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^k(s,t) = \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^{k+1}(s,t) + \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}\hat{b}_{ij}^{k+1}(s,t). \tag{4.2}$$

For each $i\in I_k$, $\hat{b}_{ij}^{k+1}(s,t)\in \text{span}\{b_{ij}^{k+1}(s,t)\,|\,i\in I_{k+1}\backslash I_k,\ \ j=1,2,3,4\}$ by the definition of $\hat{b}_{ij}^{k+1}(s,t)$. Consequently, the second term in the right handside of (4.2) can be represented as

$$\sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}\hat{b}_{ij}^{k+1}(s,t) = \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}\tilde{\mathbf{C}}_{ij}b_{ij}^{k+1}(s,t),$$

where $\tilde{\mathbf{C}}_{ij}$ $(i\in I_{k+1}\backslash I_k,\ \ j=1,2,3,4)$ is the linear combination of $\{\mathbf{C}_{ij}:\ i\in I_k,\ \ j=1,2,3,4\}$. Thus

$$\mathbf{Q}_k(s,t)|_{\theta_k} = \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^{k+1}(s,t) + \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}\tilde{\mathbf{C}}_{ij}b_{ij}^{k+1}(s,t). \tag{4.3}$$

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

On the other hand,

$$\mathbf{Q}_{k+1}(s,t)|_{\theta_{k+1}} = \sum_{i\in I_{k+1}}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^{k+1}(s,t) = \sum_{i\in I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^{k+1}(s,t) + \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}\mathbf{C}_{ij}b_{ij}^{k+1}(s,t).$$

Hence (noticing that $\theta_{k+1}\subset\theta_k$),

$$(\mathbf{Q}_{k+1}(s,t)-\mathbf{Q}_k(s,t))|_{\theta_{k+1}} = \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}(\mathbf{C}_{ij}-\tilde{\mathbf{C}}_{ij})b_{ij}^{k+1}(s,t). \tag{4.4}$$

Next by subdividing $\mathbf{P}_k(s,t)$ over L_{k+1}, we have

$$\mathbf{P}_k(s,t)|_{\theta_{k+1}} = \sum_{i=1}^{4}\sum_{j=1}^{4}\hat{\mathbf{C}}_{ij}^{k+1}M_{ij}^{k+1}(s,t).$$

Furthermore, by our algorithm we have

$$(\mathbf{P}_{k+1}(s,t)-\mathbf{P}_k(s,t))|_{\theta_{k+1}} = \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}(\mathbf{C}_{ij}-\hat{\mathbf{C}}_{ij}^{k+1})M_{ij}^{k+1}(s,t). \tag{4.5}$$

Based on the basis functions construction algorithm in [5],

$$b_{ij}^{k+1}(s,t) = M_{ij}^{k+1}(s,t), \qquad i\in I_{k+1}\backslash I_k,\ \ j=1,2,3,4.$$

Furthermore, since $\mathbf{Q}_k(s,t)|_{\theta_k} = \mathbf{P}_k(s,t)|_{\theta_k}$, one has

$$\hat{\mathbf{C}}_{ij}^{k+1} = \tilde{\mathbf{C}}_{ij}, \qquad i\in I_{k+1}\backslash I_k \ \text{ and } \ j=1,2,3,4.$$

This can be manifested by the conditions

$$\mathbf{Q}_k(V_i^{k+1}) = \mathbf{P}_k(V_i^{k+1}), \qquad \frac{\partial\mathbf{Q}_k}{\partial s}(V_i^{k+1}) = \frac{\partial\mathbf{P}_k}{\partial s}(V_i^{k+1}),$$
$$\frac{\partial\mathbf{Q}_k}{\partial t}(V_i^{k+1}) = \frac{\partial\mathbf{P}_k}{\partial t}(V_i^{k+1}), \qquad \frac{\partial^2\mathbf{Q}_k}{\partial s\partial t}(V_i^{k+1}) = \frac{\partial\mathbf{P}_k}{\partial s\partial t}(V_i^{k+1}),$$

and the fact that, for any $i\in I_k$ and $j=1,2,3,4$, the function value, the first derivatives and the mixed derivatives of $b_{ij}^{k+1}(s,t)$ vanish at V_i^{k+1}.

Thus

$$(\mathbf{P}_{k+1}(s,t)-\mathbf{P}_k(s,t))|_{\theta_{k+1}} = \sum_{i\in I_{k+1}\backslash I_k}\sum_{j=1}^{4}(\mathbf{C}_{ij}-\tilde{\mathbf{C}}_{ij})b_{ij}^{k+1}(s,t). \tag{4.6}$$

Comparing (4.4) with (4.6), one immediately arrives at

$$(\mathbf{Q}_{k+1}(s,t)-\mathbf{Q}_k(s,t))|_{\theta_{k+1}} = (\mathbf{P}_{k+1}(s,t)-\mathbf{P}_k(s,t))|_{\theta_{k+1}},$$

and hence

$$\mathbf{Q}_{k+1}(s,t)|_{\theta_{k+1}} = \mathbf{P}_{k+1}(s,t)|_{\theta_{k+1}}. \tag{4.7}$$

This proves the theorem. □

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

5. Example

In this section, we apply the evaluation algorithm presented in the last section to compute the value of a PHT-spline surface $\mathbf{P}(s,t)$ at $(s,t)=(4.5,4.5)$ as illustrated in Fig. 3. We focus on how to compute the control points of the series of local tensor product bicubic B-splines.

For the first step, $\theta_0=[4,8]\times[4,8]$, $M=1$ and $L_0=[0,8]\times[0,8]$. The local tensor product bicubic B-spline surface $\mathbf{P}_0(s,t)$ defined over L_0 is

$$\mathbf{P}_0(s,t):=\sum_{i=1}^{4}\sum_{j=1}^{4}\mathbf{C}_{ij}^0M_{ij}^0(s,t),$$

where $M_{ij}^0(s,t)$ are the tensor product bicubic B-spline basis functions associated with the knot vectors $(0,0,4,4,8,8,8,8)\times(0,0,4,4,8,8,8,8)$. The control points correspond to the four corner vertices $\{V_i^0\}_{i=1}^4$ of θ_0 have already been given by the input conditions (initial control points), where $V_1^0=(4,4)$, $V_2^0=(8,4)$, $V_3^0=(4,8)$ and $V_4^0=(8,8)$. For the second step, $\theta_1=[4,6]\times[4,6] and L_1=[2,8]\times[2,8]$. The local tensor product bi-cubic B-spline $\mathbf{P}_1(s,t)$ defined over L_1 is

$$\mathbf{P}_1(s,t):=\sum_{i=1}^{4}\sum_{j=1}^{4}\mathbf{C}_{ij}^1M_{ij}^1(s,t),$$

where $M_{ij}^1(s,t)$ are the tensor product bicubic B-spline basis functions associated with the knot vectors $(2,2,4,4,6,6,8,8)\times(2,2,4,4,6,6,8,8)$, and $V_1^1=(4,4)$, $V_2^1=(6,4)$, $V_3^1=(4,6)$, $V_4^1=(6,6)$. The control points $\mathbf{C}_{ij}^1$, $i=4$, $j=1,2,3,4$ are given by the input conditions. And the other control points are computed by subdividing $\mathbf{P}_0(s,t)$ over L_1:

$$\mathbf{C}_{11}^1=\frac{9}{16}\mathbf{C}_{11}^0+\frac{3}{8}\mathbf{C}_{13}^0+\frac{1}{16}\mathbf{C}_{14}^0,\qquad \mathbf{C}_{12}^1=\frac{5}{6}\mathbf{C}_{12}^0+\frac{1}{6}\mathbf{C}_{14}^0,$$

$$\mathbf{C}_{21}^1=\frac{1}{6}\mathbf{C}_{11}^1+\frac{1}{4}\mathbf{C}_{21}^0+\frac{7}{16}\mathbf{C}_{12}^0+\frac{7}{48}\mathbf{C}_{14}^0,\qquad \mathbf{C}_{22}^1=\frac{1}{2}\mathbf{C}_{21}^0+\frac{1}{4}\mathbf{C}_{22}^0+\frac{3}{16}\mathbf{C}_{12}^0+\frac{1}{16}\mathbf{C}_{14}^0,$$

$$\mathbf{C}_{13}^1=\frac{5}{6}\mathbf{C}_{13}^0+\frac{1}{6}\mathbf{C}_{14}^0,\qquad \mathbf{C}_{14}^1=\frac{1}{9}\mathbf{C}_{11}^0+\frac{1}{6}\mathbf{C}_{13}^0+\frac{5}{9}\mathbf{C}_{14}^0+\frac{1}{6}\mathbf{C}_{12}^0,$$

$$\mathbf{C}_{23}^1=\frac{1}{18}\mathbf{C}_{11}^1+\frac{1}{12}\mathbf{C}_{13}^0+\frac{67}{144}\mathbf{C}_{14}^0+\frac{7}{48}\mathbf{C}_{12}^0+\frac{5}{24}\mathbf{C}_{21}^0+\frac{1}{24}\mathbf{C}_{23}^0,$$

$$\mathbf{C}_{24}^1=\frac{1}{16}\mathbf{C}_{12}^0+\frac{9}{48}\mathbf{C}_{14}^0+\frac{5}{12}\mathbf{C}_{21}^0+\frac{5}{24}\mathbf{C}_{23}^0+\frac{1}{12}\mathbf{C}_{22}^0+\frac{1}{24}\mathbf{C}_{24}^0,$$

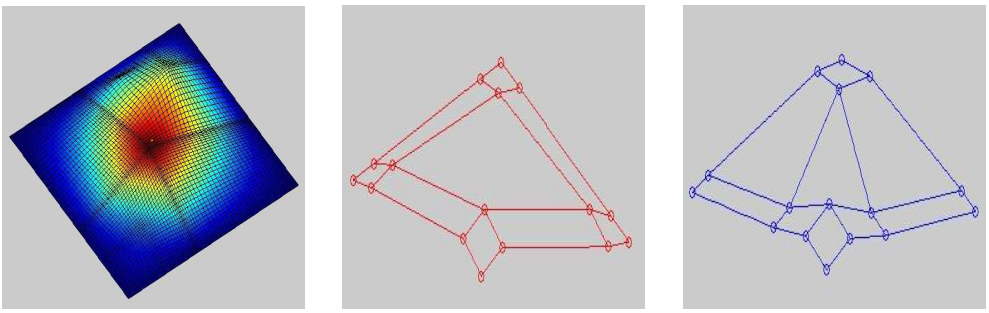
$$\mathbf{C}_{31}^1=\frac{1}{6}\mathbf{C}_{11}^1+\frac{3}{16}\mathbf{C}_{31}^0+\frac{1}{16}\mathbf{C}_{32}^0+\frac{7}{16}\mathbf{C}_{33}^0+\frac{7}{48}\mathbf{C}_{24}^0,$$

$$\mathbf{C}_{32}^1=\frac{1}{18}\mathbf{C}_{11}^1+\frac{1}{12}\mathbf{C}_{12}^0+\frac{67}{144}\mathbf{C}_{14}^0+\frac{21}{144}\mathbf{C}_{13}^0+\frac{1}{16}\mathbf{C}_{31}^0+\frac{9}{48}\mathbf{C}_{32}^0,$$

$$\mathbf{C}_{33}^1=\frac{1}{6}\mathbf{C}_{31}^0+\frac{3}{16}\mathbf{C}_{13}^0+\frac{1}{16}\mathbf{C}_{14}^0+\frac{7}{16}\mathbf{C}_{31}^0+\frac{7}{48}\mathbf{C}_{32}^0,$$

$$\mathbf{C}_{34}^1=\frac{1}{16}\mathbf{C}_{13}^0+\frac{9}{48}\mathbf{C}_{14}^0+\frac{1}{8}\mathbf{C}_{31}^0+\frac{9}{24}\mathbf{C}_{32}^0+\frac{1}{16}\mathbf{C}_{33}^0+\frac{9}{48}\mathbf{C}_{34}^0.$$

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press



(a) (b) (c)

Figure 5: (a) A PHT-surface based on the hierarchical T-mesh shown in Fig.4(a), and the yellow point is the final evaluation result; (b) and (c) are respectively the control nets of the local tensor product B-spline surfaces associated with the parameter pair $(4.5, 4.5)$ at level 0 and 1.

Finally, we evaluate $\mathbf{P}_1(s,t)$ at $(s,t) = (4.5, 4.5)$ using de Boor's algorithm:

$$\begin{aligned}&\mathbf{P}(4.5,4.5)\\ =&\frac{25\mathbf{C}_{11}^1+75\mathbf{C}_{12}^1+75\mathbf{C}_{13}^1+225\mathbf{C}_{14}^1}{16384}+\frac{135\mathbf{C}_{21}^1+135\mathbf{C}_{22}^1+405\mathbf{C}_{23}^1+405\mathbf{C}_{24}^1}{8192}\\ &+\frac{135\mathbf{C}_{31}^1+405\mathbf{C}_{32}^1+135\mathbf{C}_{33}^1+405\mathbf{C}_{34}^1}{8192}+\frac{729\mathbf{C}_{41}^1+729\mathbf{C}_{42}^1+729\mathbf{C}_{43}^1+729\mathbf{C}_{44}^1}{4096}.\end{aligned}$$

6. Complexity analysis

In this section, the computational complexity and the memory cost of our evaluation algorithm of PHT-spline surface are analyzed, together with a comparison with the Bézier representation method proposed in [5], where all the basis functions are computed and stored in Bézier forms beforehand. Because the computational complexity and the memory cost of evaluating PHT-spline surfaces are closely related with the the topological structure of the corresponding hierarchical T-mesh, the algorithm complexity is roughly analyzed.

6.1. Computational complexity

Our evaluation algorithm basically requires to compute 16 control points of a local tensor product bicubic B-spline surface at each level. The computational cost is not more than 376 multiplications at each level. Consequently, we need $376M + 302$ multiplications at most to evaluate a PHT-spline, in which 302 multiplications account for the cost to evaluate $\mathbf{P}_M(s_0, t_0)$ using de-Boor's algorithm. Here M is the maximum hierarchy of the point (s_0, t_0) in the mesh $\mathscr{T}$. In the best case, for example when $M = 0$ or the local structure of the T-mesh $\mathscr{T}$ at (s_0, t_0) is already a tensor product mesh, the complexity of our algorithm just likes to evaluate a tensor-product B-spline using de-Boor's algorithm.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

Table 1: Computational complexity of evaluating PHT-splines.

Our algorithm	Bézier representation method
$376M + 302$ multiplications	$384m + 238$ multiplications

Table 2: Comparison of memory costs.

Our algorithm	Bézier representation method
156 float numbers	$\sum_{i=1}^{n} 64m_i$ float numbers

Next we consider the computational costs of Bézier representation method. For a specific parameter pair (s_0, t_0), suppose there are m basis vertices whose supports contain (s_0, t_0). Then the computational cost is $384m + 238$ multiplications to evaluate a PHT-spline. We summarize the results in the following table.

From Table 1, if $M >> m$, our evaluation algorithm generally costs more computations than the Bézier representation method. However, if M is not too big, our method can be more efficient. For the specific example illustrated in Section 5, the computational costs for the two methods are 678 and 1774 multiplications respectively. Fig. 6 provides an additional example to compare the computational cost of our algorithm with with that of the Bézier representation method. The computational costs for the green, blue and red points in Fig. 6 are 302, 678 and 962 multiplications respectively by using our evaluation algorithm, and are 1774 multiplications by using the Bézier representation method.

6.2. The memory costs

For our evaluation algorithm, we need recursively compute a series of tensor product bicubic B-splines over local tensor product meshes, the storage is the 16 control points of a bicubic B-spline surface. Our algorithm still need 32 intermediate control points and 12 float numbers. Thus for each level, the memory cost is at most 156 float numbers.

The required memory units for Bézier representation method are closely associated with the topological structure of the hierarchical T-mesh $\mathscr{T}$. A basis function is represented by 16 Bézier ordinates in each cell in the support of the basis function. Suppose $\mathscr{T}$ have n cells $\{\theta_i\}_{i=1}^{n} (i = 1, \cdots, n)$, and there are m_i basis vertices that the basis functions associated with them have support on each cell θ_i. Then only storage these Bézier ordinates need $\sum_{i=1}^{n} 64m_i$ float numbers, which is much larger than the memory costs of our method if the T-mesh contains lots of levels and has a complex topology. For the example illustrated in Section 5, the memory costs for the two methods are 156 and 3328 floating numbers respectively. For the example shown in Fig. 6.2, the memory costs for the two methods are 156 and 8960 respectively. We summarize the results in the following table.

7. Conclusions

This will save a lot of space when applying PHT-splines in applications such as surface reconstructions, especially implicit surface reconstruction proposed in [20].

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

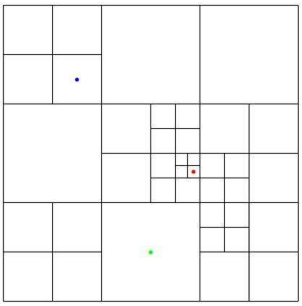


Figure 6: A hierarchical T-mesh $\mathscr{T}$ with maximum hierarchy 3. The hierarchy of the green, blue and red points are 0, 1, and 3 respectively.

References

[1] D. R. Forsey and R. H. Bartels, *Hierarchical B-spline refinement*, Comp. Graph., 22(4) (1988), pp. 205–212.

[2] D. R. Forsey and R. H. Bartels, *Surface fitting with hierarchical splines*, ACM T. Graphic., 14(2) (1995), pp. 134–161.

[3] T. Sederberg, J. Zheng, A. Bakenov and A. Nasri, *T-splines and T-NURCCSs*, ACM T. Graphic., 22(3) (2003), pp. 477–484.

[4] T. Sederberg, D. Cardon, G. Finnigan, N. North, J. Zheng and T. Lyche, *T-spline simplification and local refinement*, ACM T. Graphic., 23(3) (2004), pp. 276–283.

[5] J. S. Deng, F. L. Chen, X. Li, C. Q. Hu, W. H. Tong, Z. W. Yang and Y. Y. Feng, *Polynomial splines over hierarchical T-meshes*, Graph. Models, 70 (2008), pp. 76–86.

[6] X. Li, J. Deng and F. Chen, *Surface modeling with polynomial splines over hierarchical T-meshes*, Visual Comput., 23(12) (2007), pp. 1027–1033.

[7] X. Li, J. S. Deng and F. L. Chen, *Polynomial splines over general T-meshes*, Visual Comput., 26(4) (2010), pp. 277–286.

[8] X. Li and M. A. Scott, Analysis-suitable T-splines: characterization, refineability, and approximation, Math. Mod. Meth. Appl. S., 24(06) (2014), pp. 1141–1164.

[9] M. A. Scott, X. Li, T. W. Sederberg and T. J. R. Hughes, *Local refinement of analysis-suitable T-splines*, Comput. Method. Appl. M., 4 (2012), pp. 206–222.

[10] X. Li, J. M. Zheng, T. W. Sederberg, T. J. R. Hughes and M. A. Scott, *On the linear independence of T-spline blending functions*, Comput. Aided Geom. D., 29 (1) (2012), pp. 63–76.

[11] C. Giannelli, B. Juttler, H. Speleers, *THB-splines: the truncated basis for hierarchical splines*, Comput. Aided Geom. D., 29 (2012), pp. 485–498.

[12] T. Dokken, T. Lyche, K. F. Pettersen, *Polynomial splines over locally refined box-partitions*, Comput. Aided Geom. D., 30 (2013), pp. 331–356.

[13] M. Pan, W. Tong and F. Chen, *Compact implicit surface reconstruction via low-rank tensor approximation*, Computer-Aided Design, 78 (2016), pp. 158–167.

[14] L. Tian, F. Chen and Q. Du, *Adaptive finite element methods for elliptic equations over hierarchical T-meshes*, J. Comput. Appl. Math., 236 (2011), pp. 878–891.

[15] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wuchner, K. U. Bletzinger, Y. Bazilevs and T. Rabczuk, *Rotation free isogeometric thin shell analysis using PHT-splines*, Comput. Method. Appl. M., 2011, pp. 3410–3424.

[16] N. Nguyen-Thanh, H. Nguyen-Xuan, S. P. A. Bordas and T. Rabczuk, *Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids*, Comput.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press

Method. Appl. M., 200 (2011), pp. 1892–1908.
[17] N. Nguyen-Thanh, J. Muthu, X. Zhuang and T. Rabczuk, *An adaptive three dimensional RHT-spline formulation in linear elasto-statics and elasto-dynamics*, Comput. Mech., 3(2) (2014), pp. 369–385.
[18] N. Nguyen-Thanh, N. Valizadeh, M. N. Nguyen, H. Nguyen-Xuan, X. Zhuang, P. Areias, G. Zi, Y. Bazilevs, L. De Lorenzis and T. Rabczuk, *An extended isogeometric thin shell analysis based on Kirchhoff-Love theory*, Comput. Method. Appl. M., 284 2015, pp. 265–291.
[19] P. Wang, J. L. Xu, J. S. Deng and F. L. Chen, *Adaptive isogeometric analysis using rational PHT-splines*, Computer-Aided Design, 43 (2011), pp. 1438–1448.
[20] J. Wang, Z. Yang, L. Jin, J. Deng and F. Chen, *Parallel and adaptive surface reconstruction based on implicit PHT-splines*, Comput. Aided Geom. D., 28(8) (2011), pp. 463–474.
[21] J. Deng, F. Chen and Y. Feng, *Dimensions of spline spaces over T-meshes*, J. Comput. Appl. Math., 194 (2006), pp. 267–283.
[22] B. Mourrain, *On the dimension of spline spaces on planar T-meshes*, Math. Comput., 83 (2014), pp. 847–871.
[23] X. Li and F. Chen, *On the instability in the dimension of splines spaces over T-meshes*, Comput. Aided Geom. D., 28 (2011), pp. 420–426.
[24] J. Deng, F. Chen and L. Jin, *Dimensions of biquadratic spline spaces over T-meshes*, J. Comput. Appl. Math., 238 (2013), pp. 68–94.
[25] M. Wu, J. Deng and F. Chen, *Dimension of spline spaces with highest order smoothness over hierarchical T-meshes*, Comput. Aided Geom. D., 30 (2013), pp. 20–34.
[26] D. Berdinsky, M. Oh, T. Kim and B. Mourrain, *On the problem of instability in the dimension of a spline space over a T-mesh*, Comput. Graph., 36 (2012), pp. 507–513.
[27] C. Zeng, F. Deng, X. Li and J. Deng, *Dimensions of biquadratic and bicubic spline spaces over hierarchical T-meshes*, J. Comput. Appl. Math., 287 (2015), pp. 162–178.
[28] C. Zeng, F. Deng and J. Deng, *Bicubic hierarchical B-splines: Dimensions, completeness, and bases*, Comput. Aided Geom. D., 38 (2015), pp. 1–23.
[29] G. Farin, Curves and Surfaces for CAGD–A Practical Guide, 5th ed., Morgan Kaufmann Publishers, 2002.

https://doi.org/10.4208/nmtma.2017.0003 Published online by Cambridge University Press