



ARTICLE

The Weighted Basis for PHT-Splines

Zhiguo Yong¹, Hongmei Kang¹ and Falai Chen^{2,*}

¹School of Mathematical Sciences, Soochow University, Suzhou, 215006, China

²School of Mathematical Sciences, University of Science and Technology of China, Hefei, 230026, China

*Corresponding Author: Falai Chen. Email: chenfl@ustc.edu.cn

Received: 17 October 2022 Accepted: 03 January 2023 Published: 22 September 2023

ABSTRACT

PHT-splines are defined as polynomial splines over hierarchical T-meshes with very efficient local refinement properties. The original PHT-spline basis functions constructed by the truncation mechanism have a decay phenomenon, resulting in numerical instability. The non-decay basis functions are constructed as the B-splines that are defined on the 2×2 tensor product meshes associated with basis vertices in Kang et al., but at the cost of losing the partition of unity. In the field of finite element analysis and topology optimization, forming the partition of unity is the default ingredient for constructing basis functions of approximate spaces. In this paper, we will show that the non-decay PHT-spline basis functions proposed by Kang et al. can be appropriately modified to form a partition of unity. Each non-decay basis function is multiplied by a positive weight to form the weighted basis. The weights are solved such that the sum of weighted bases is equal to 1 on the domain. We provide two methods for calculating weights, based on geometric information of basis functions and the subdivision of PHT-splines. Weights are given in the form of explicit formulas and can be efficiently calculated. We also prove that the weights on the admissible hierarchical T-meshes are positive.

KEYWORDS

PHT-splines; partition of unity; weighted bases; PHT-splines; subdivision

1 Introduction

Polynomial splines over hierarchical T-meshes (PHT-splines for short) [1] are a kind of global C^1 continuous polynomial splines that can be locally refined. PHT-splines possess a very efficient local refinement algorithm and inherit many good properties of T-splines such as adaptivity and locality. PHT-splines are now widely applied in adaptive geometric modeling [2–4], adaptive finite element [5], iso-geometric analysis [6–13] and topology optimization [14].

With further in-depth application, the improvement of PHT-splines has also been underway. One of the limitations of PHT-splines is the restrictions from hierarchical T-meshes, which require the underlying meshes to start with tensor product meshes and be refined by inserting crosses. The generalization of PHT-splines on general T-meshes [15] and modified hierarchical T-meshes (allowing split-in-half in mesh refinement) [16] is precisely aimed at overcoming the restrictions from meshes. Another improvement regard to PHT-splines is the construction of basis functions. The original basis



functions [1] have a decay phenomenon (the function values decay as the level increases) when the level difference is big in the underlying support. The decay of basis functions when applied in iso-geometric analysis leads to ill-conditioned stiffness matrices, and thus unstable numerical solutions. A new set of basis functions is defined by the B-splines that defined on the 2×2 tensor product meshes associated with basis vertices [2] to overcome the decay phenomenon that occurs in basis functions, but at the cost of losing the partition of unity.

The partition of unity is a default ingredient in the construction of approximating functions in finite element methods [17–19]. Moreover, in the field of topology optimization [14], the density variable takes value in the range of $[0, 1]$. The basis functions that form a partition of unity are more suitable for designing density variables that meet the range condition. In this paper, we aim to provide basis functions that form a partition of unity and also have no decay phenomenon, facilitating more effective applications for PHT-splines.

The truncation mechanism is a common way of constructing basis functions for hierarchical spline spaces that form a partition of unity, such as the original PHT-splines [1], THB-splines (the truncated hierarchical B-splines) [20] and truncated T-splines [21]. But as mentioned above, the truncation mechanism causes the decay in basis functions. In this paper, we will show that the basis functions that form the partition of unity and have no decay can be constructed without the truncation mechanism. The weighted bases that satisfy the required properties are defined by the bases constructed in [2] multiplied by positive weights. The weight for each basis function can be computed explicitly and efficiently based on the geometric information of the basis function. Moreover, when the level difference is not greater than 2, the weights can also be computed by the PHT subdivision scheme.

In [22], the authors also proposed a method to define new basis functions to overcome the decay problem and guarantee the partition of unity. However, their method only considers specific cases that the original truncated basis functions have rectangular supports, in which case they are replaced by the associated tensor product B-splines. Instead, our method is more versatile and can be applied to arbitrary T-meshes easily.

The rest of this paper is organized as follows. In Section 2, some basics of PHT-splines are reviewed. In Section 3, we introduce the weighted bases in detail. In Section 4, the subdivision scheme of PHT-splines is introduced as an alternative algorithm for calculating weights. In Section 5, some hierarchical T-meshes are demonstrated to verify the effectiveness of the proposed method. Finally, we conclude the paper with a summary in Section 6.

2 PHT-Splines

In this section, we give a brief review of the definition of PHT-splines. The readers are recommended to refer to [1] for more details.

2.1 Definition of PHT-Splines

Given a T-mesh \mathbb{T} , \mathcal{F} denotes all the cells in \mathbb{T} and Ω denotes the region occupied by \mathcal{F} . The polynomial spline space over \mathbb{T} is defined as

$$\mathcal{S}(m, n, \alpha, \beta, \mathbb{T}) := \{s(x, y) \in C^{\alpha, \beta}(\Omega) | s(x, y)|_{\phi} \in \mathbb{P}_{mn} \text{ for any } \phi \in \mathcal{F}\}, \quad (1)$$

where \mathbb{P}_{mn} is the space of all the polynomials with bi-degree (m, n) , and $C^{\alpha, \beta}(\Omega)$ is the space consisting of all the bivariate functions which are continuous in Ω with order α in the x -direction and with order β in the y -direction. In particular, in the case of $m \geq 2\alpha + 1, n \geq 2\beta + 1$, $\mathcal{S}(m, n, \alpha, \beta, \mathbb{T})$ is well studied and its dimension is given in [23].

PHT is defined by $\mathcal{S}(3, 3, 1, 1, \mathbb{T})$ with \mathbb{T} being a hierarchical T-mesh. Here, a hierarchical T-mesh is a special type of T-mesh that has a natural hierarchical structure and defined in a recursive fashion. One generally starts from a tensor product mesh (level 0). From level k to level $k + 1$, a cell at level k is subdivided into four sub-cells which are cells at level $k + 1$.

The dimension formula of PHT has a concise expression with

$$\mathcal{S}(3, 3, 1, 1, \mathbb{T}) = 4(V^b + V^+), \tag{2}$$

where V^b and V^+ represent the number of boundary vertices and interior crossing vertices in \mathbb{T} respectively. As defined in [1], a boundary vertex or an interior crossing vertex is called a basis vertex. This dimension formula implies that each basis vertex should be associated with four basis functions. Thus, the construction of PHT is to construct four bicubic $C^{1,1}$ continuous functions for each basis vertex so that the functions span the PHT and retain some good properties.

2.2 Basis Construction

The original PHT basis functions are constructed by truncation mechanism in [1] level by level in order to make basis functions vanish at other basis vertices. Fig. 1 shows two typical hierarchical meshes, where the basis functions associated the marked vertices have undesired decay as level increases. In Fig. 1a, the maximum values decay rapidly as the level increases. In Fig. 1b, the maximum value does not decrease, but the basis function decays sharply along the refinement direction as level increases. This decay phenomenon of basis functions is first noted in [2]. The decay phenomenon makes the stiffness matrix have large condition number and thus produces unstable solutions.

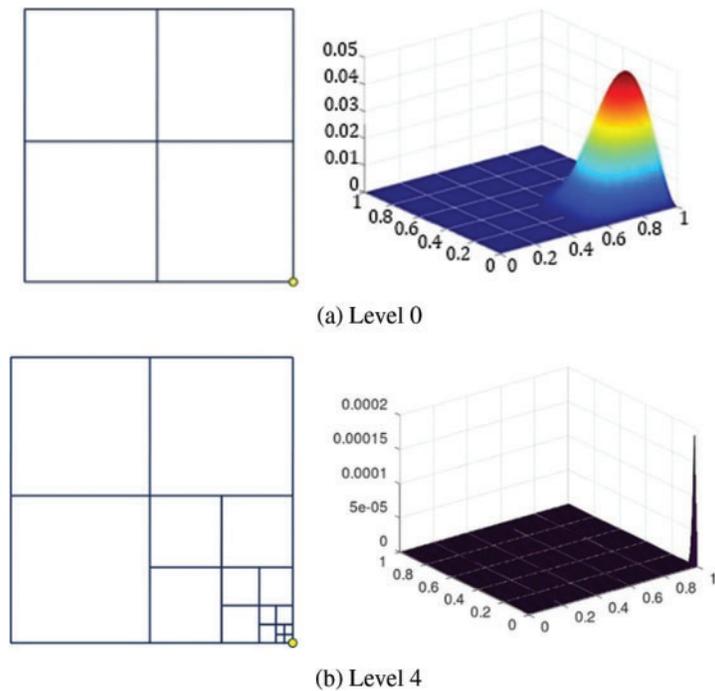


Figure 1: (Continued)

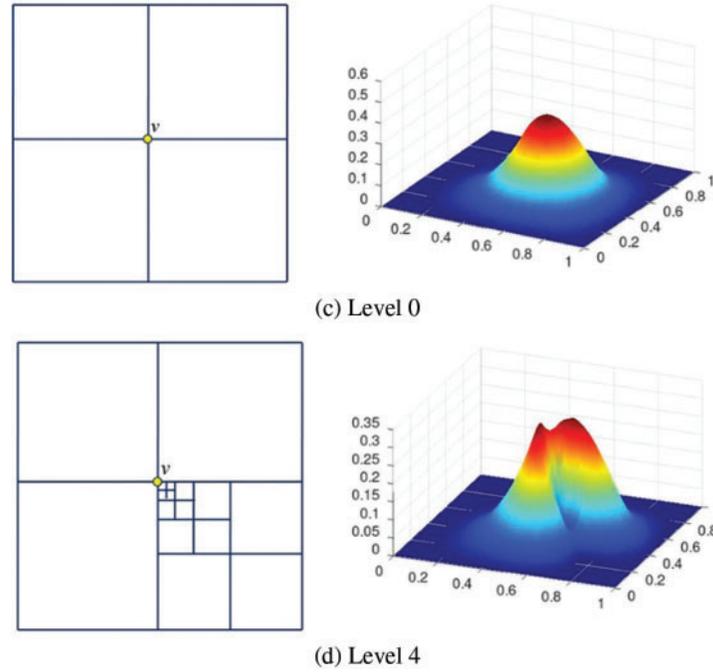


Figure 1: The decay phenomenon of PHT-spline basis functions associated with the vertex marked by a yellow circle

A new basis functions are proposed in [2] to amend the decay phenomenon. In the following, we give a brief review of the construction.

Definition 1 Suppose v is a basis vertex in \mathbb{T} , the support mesh of v , denoted by T_v , is defined as the minimal 2×2 tensor product mesh, where v is the central vertex and all edges are subordinate to \mathbb{T} .

Note that if v is a boundary vertex, the support mesh is actually a 2×1 , 1×2 or 1×1 tensor product mesh. For any given hierarchical T-mesh, the support mesh of each basis vertex exists and is unique. The support mesh can be found efficiently.

Fig. 2 illustrates the support meshes of six basis vertices including three boundary vertices and three interior crossing vertices, where the six basis vertices are marked by solid circles and the corresponding support meshes are marked by bold lines.

The basis functions associated with a basis vertex are then defined by the four B-spline functions that defined on the support mesh. Specifically, for a basis vertex $v = (s_i, t_j)$, suppose the support mesh is defined by $(s_{i_0}, s_i, s_{i_1}) \times (t_{j_0}, t_j, t_{j_1})$, then the four bases are defined as follows:

$$\begin{aligned}
 b_0(s, t) &= N^3 [s_{i_0}, s_{i_0}, s_i, s_i, s_{i_1}] [s] \times N^3 [t_{j_0}, t_{j_0}, t_j, t_j, t_{j_1}] [t], \\
 b_1(s, t) &= N^3 [s_{i_0}, s_i, s_i, s_{i_1}, s_{i_1}] [s] \times N^3 [t_{j_0}, t_{j_0}, t_j, t_j, t_{j_1}] [t], \\
 b_2(s, t) &= N^3 [s_{i_0}, s_{i_0}, s_i, s_i, s_{i_1}] [s] \times N^3 [t_{j_0}, t_j, t_j, t_{j_1}, t_{j_1}] [t], \\
 b_3(s, t) &= N^3 [s_{i_0}, s_i, s_i, s_{i_1}, s_{i_1}] [s] \times N^3 [t_{j_0}, t_j, t_j, t_{j_1}, t_{j_1}] [t].
 \end{aligned} \tag{3}$$

These four basis functions are linearly independent and have the same support $[s_{i_0}, s_{i_1}] \times [t_{j_0}, t_{j_1}]$. This way of definition has been confirmed in [2] without decay phenomenon.

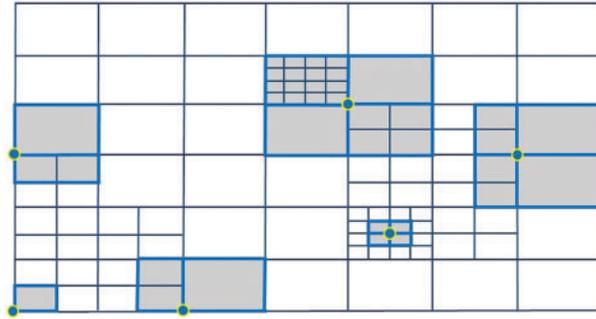


Figure 2: Support meshes of six basis vertices

3 Weighted Bases for PHT-Splines

3.1 Geometric Information

For any basis function $b(s, t)$, we define a geometric information (the function value, the first order partial derivatives and the mixed partial derivative) operator by

$$Lb(s, t) = (b(s, t), b_s(s, t), b_t(s, t), b_{st}(s, t)). \quad (4)$$

For a basis vertex $v = (s_i, t_j)$, the associated four PHT-splines basis functions are defined by (3). We collect the geometric information of b_0, b_1, b_2, b_3 at the basis vertex v into the matrix G ,

$$G = \begin{pmatrix} Lb_0(v) \\ Lb_1(v) \\ Lb_2(v) \\ Lb_3(v) \end{pmatrix} = \begin{pmatrix} \lambda\mu & -3\alpha\mu & -3\beta\lambda & 9\alpha\beta \\ (1-\lambda)\mu & 3\alpha\mu & -3\beta(1-\lambda) & -9\alpha\beta \\ \lambda(1-\mu) & -3\alpha(1-\mu) & 3\beta\lambda & -9\alpha\beta \\ (1-\lambda)(1-\mu) & 3\alpha(1-\mu) & 3\beta(1-\lambda) & 9\alpha\beta \end{pmatrix}, \quad (5)$$

where

$$\alpha = \frac{1}{\Delta s_1 + \Delta s_2}, \beta = \frac{1}{\Delta t_1 + \Delta t_2}, \lambda = \alpha \Delta s_2, \mu = \beta \Delta t_2,$$

$$\Delta s_1 = s_i - s_{i_0}, \Delta s_2 = s_{i_1} - s_i, \Delta t_1 = t_j - t_{j_0}, \Delta t_2 = t_{j_1} - t_j. \quad (6)$$

The sum of the geometric information of the four basis functions b_0, b_1, b_2, b_3 at the associated basis vertex is equal to $(1, 0, 0, 0)$, because the sum of the four basis functions is identically equal to 1. This can also be verified by the column sums in G . As non-vanishing basis functions at basis vertices are not limited to the associated four basis functions, so the basis functions defined in [2] do not form a partition of unity generally.

Suppose that the maximal subdivision level of \mathbb{T} is N and \mathbb{T}_k denotes the hierarchical T-mesh at level $k, k = 0, \dots, N$. Then we have $\mathbb{T} = \mathbb{T}_N$. For a basis vertex v , if $v \in \mathbb{T}_k$ but $v \notin \mathbb{T}_{k-1}$, then it is called a basis vertex at level $k, k = 1, \dots, N$. Specially, every vertex in \mathbb{T}_0 is a basis vertex at level 0. The i th basis vertex at level k is denoted by v_i^k and the four basis functions associated with v_i^k are denoted as $b_{4i+j}^k, j = 0, 1, 2, 3$.

According to the definitions of support meshes and hierarchical T-meshes, the basis function proposed in [2] has the following property:

- A basis vertex at level l can not be in the interior of the support of any basis function at level $k > l$, that is

$$\text{when } l < k, b_{4i+j}^k(v_g^l) = 0, \text{ for } j = 0, 1, 2, 3, i = 1, \dots, n_k, g = 1, \dots, n_l, \quad (7)$$

where n_k and n_l are the number of basis vertices on level k and l , respectively.

- The basis functions b_{4i+j}^k vanish at the basis vertices on level k , except for the vertices associated with them, that is

$$\text{when } l = k, b_{4i+j}^k(v_g^l) = 0, \text{ for } g \neq i, j = 0, 1, 2, 3, k = 0, \dots, N. \quad (8)$$

3.2 Computing Weights

The proposed weighted bases are defined by multiplying the basis functions constructed in [2] by positive weights. This ensures that the weighted bases not only have no decay, but also form the positive partition of unity. In the following, we are going to compute weights $w_{4i+j}^l > 0$ for each basis function such that

$$f(s, t) := \sum_{l=0}^N \sum_{i=1}^{n_l} \sum_{j=0}^3 w_{4i+j}^l b_{4i+j}^l(s, t) \equiv 1, (s, t) \in [0, 1] \times [0, 1]. \quad (9)$$

This problem can be understood as the PHT fitting problem of a constant function. Since a PHT-spline space spans the bicubic polynomial space \mathbb{P}_{33} , the weights w_{4i+j}^l satisfying (9) exist. As described in [1], the control points of the PHT-spline surface, here we call them weights, can be determined based on the geometric information at the basis vertices.

The weights are computed level by level. First, we evaluate the PHT-spline surface at the basis vertices on level 0, that is

$$\sum_{l=0}^N \sum_{i=1}^{n_l} \sum_{j=0}^3 w_{4i+j}^l b_{4i+j}^l(v_g^0) = \sum_{j=0}^3 w_{4g+j}^0 b_{4g+j}^0(v_g^0) = 1, g = 1, \dots, n_0, \quad (10)$$

since the basis functions on level $l > 0$ vanish on any basis vertex on level 0. Furthermore, B-spline basis functions $b_{4g+j}^0, j = 0, 1, 2, 3$ form a partition of unity, thus we obtain $w_{4g+j}^0 = 1$, for $j = 0, 1, 2, 3$ and $g = 1, \dots, n_0$.

For a basis vertex v_g^k on level $k > 0$, the non-vanishing basis functions at v_g^k are composed of two parts: the non-vanishing basis functions at v_g^k on level $l < k$ and the four basis functions associated with v_g^k . The indices of basis vertices associated with the first part are denoted by K ,

$$K = \{j | b_j^l(v_g^k) \neq 0, l < k\},$$

and the weighted sum of the basis functions in K is denoted by $h(s, t)$,

$$h(s, t) = \sum_{l < k, j \in K} w_j^l b_j^l(s, t). \quad (11)$$

Because the geometric information operator is linear, one has

$$\begin{aligned} (Lf)(v_g^k) &= L \left(h(s, t) + \sum_{j=0}^3 w_{4g+j}^k b_{4g+j}^k(s, t) \right) \Big|_{v_g^k} \\ &= Lh(v_g^k) + \sum_{j=0}^3 w_{4g+j}^k Lb_{4g+j}^k(v_g^k). \end{aligned} \quad (12)$$

Thus, the weights of the four bases b_{4g+j}^k are determined by the following linear equations:

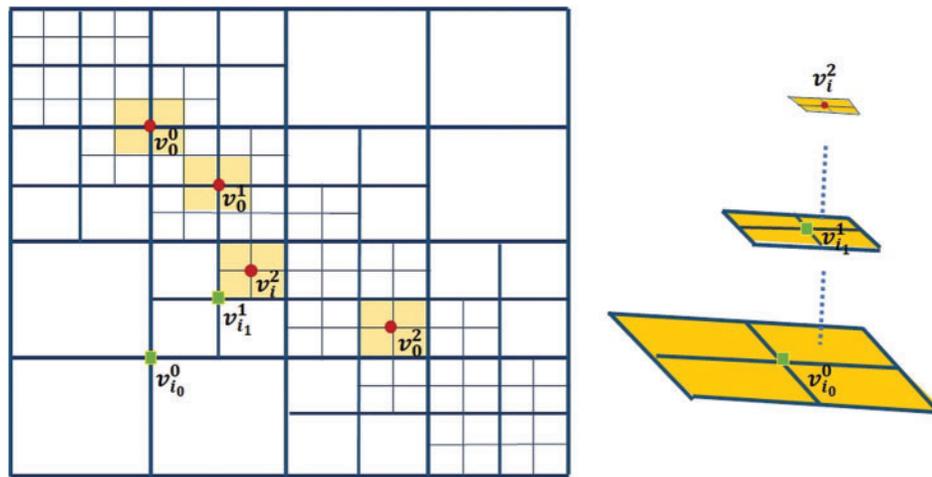
$$G^T \begin{pmatrix} w_{4g}^k \\ w_{4g+1}^k \\ w_{4g+2}^k \\ w_{4g+3}^k \end{pmatrix} = \begin{pmatrix} 1 - h(v_g^k) \\ -\frac{\partial h}{\partial s}(v_g^k) \\ -\frac{\partial h}{\partial t}(v_g^k) \\ -\frac{\partial^2 h}{\partial s \partial t}(v_g^k) \end{pmatrix}, G = \begin{pmatrix} Lb_{4g}^k(v_g^k) \\ Lb_{4g+1}^k(v_g^k) \\ Lb_{4g+2}^k(v_g^k) \\ Lb_{4g+3}^k(v_g^k) \end{pmatrix}. \tag{13}$$

The matrix G is invertible, so the solution of (13) exists and is unique. Specifically, the weights are explicitly expressed by

$$\begin{pmatrix} w_{4g}^k \\ w_{4g+1}^k \\ w_{4g+2}^k \\ w_{4g+3}^k \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{\lambda - 1}{3\alpha} & \frac{\lambda}{3\alpha} & \frac{\lambda - 1}{3\alpha} & \frac{\lambda}{3\alpha} \\ \frac{\mu - 1}{3\beta} & \frac{\mu - 1}{3\beta} & \frac{\mu}{3\beta} & \frac{\mu}{3\beta} \\ -\frac{\lambda + \mu - \lambda\mu - 1}{9\alpha\beta} & -\frac{\lambda - \lambda\mu}{9\alpha\beta} & -\frac{\mu - \lambda\mu}{9\alpha\beta} & \frac{\lambda\mu}{9\alpha\beta} \end{pmatrix}^T \begin{pmatrix} 1 - h(v_g^k) \\ -\frac{\partial h}{\partial s}(v_g^k) \\ -\frac{\partial h}{\partial t}(v_g^k) \\ -\frac{\partial^2 h}{\partial s \partial t}(v_g^k) \end{pmatrix}. \tag{14}$$

The non-vanishing basis functions at v_g^k can be found easily owing to the simple construction. The weights can be computed explicitly based on (14) without the need to solve a linear system of equations.

Fig. 3a shows a typical hierarchical mesh where the diagonal elements are refined. The support meshes of some marked basis vertices are shaded in orange color. We discuss the weights of the basis functions associated with colored vertices.



(a) Diagonal elements are refined

Figure 3: (Continued)

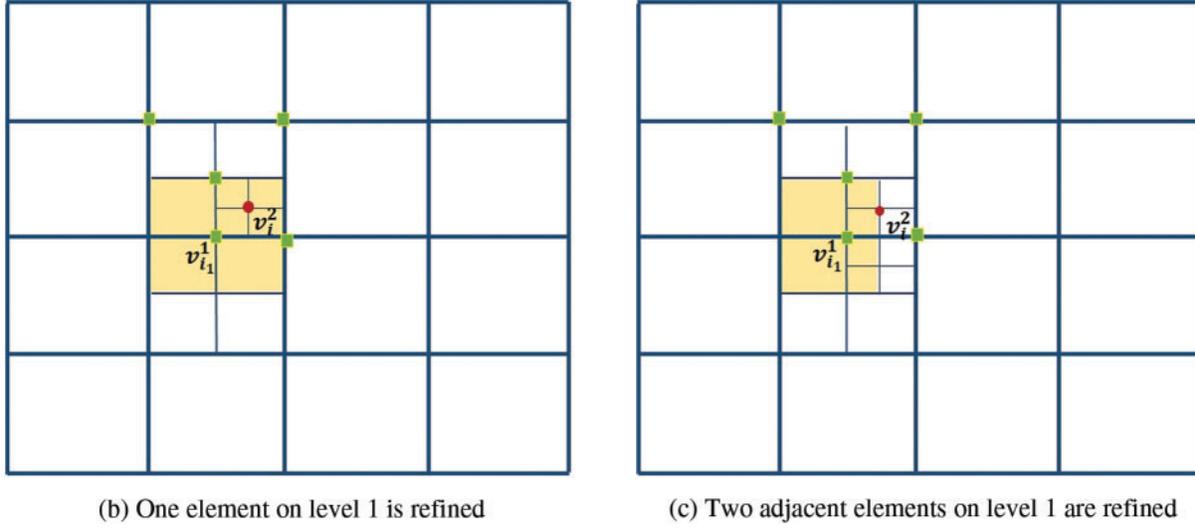


Figure 3: (a) The weights associated v_0^0, v_0^1, v_0^2 (marked by red circles) are all equal to $[1, 1, 1, 1]$. The weights for v_i^2 are equal to $\left[\frac{7}{16}, \frac{13}{16}, \frac{13}{16}, \frac{15}{16}\right]$. There are two non-vanishing basis functions (associated with the vertices marked by green squares) at v_i^2 . (b) One element on level 1 is refined. The weights associated v_i^2 are equal to $[0, 0, 0, 0]$. (c) The weights associated v_i^2 are equal to $[0.5625, 0.1875, 0.1875, 0.0625]$

- The weights of v_0^0, v_0^1 and v_0^2 (marked by red circles) are all equal to $[1, 1, 1, 1]$, since all basis functions are zeros at these three vertices, except for the basis functions associated with themselves, which means $h(s, t) \equiv 0$.
- The weights for $v_{i_1}^1$ are computed as follows: (1) the right term $h(s, t) = \sum_{j=0}^3 w_{4i_0+j}^0 b_{4i_0+j}^0$ since only the support of $v_{i_0}^0$ contains $v_{i_1}^1$; (2) the elements in G are constructed with $\Delta s_1 = \Delta s_2 = 0.125, \Delta t_1 = \Delta t_2 = 0.125, \lambda = \mu = \frac{1}{2}, \alpha = \beta = 4$, based on (6). To sum up,

$$\begin{pmatrix} w_{4i_1}^1 \\ w_{4i_1+1}^1 \\ w_{4i_1+2}^1 \\ w_{4i_1+3}^1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -\frac{1}{24} & \frac{1}{24} & -\frac{1}{24} & \frac{1}{24} \\ -\frac{1}{24} & -\frac{1}{24} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{576} & -\frac{1}{576} & -\frac{1}{576} & \frac{1}{576} \end{pmatrix}. \quad (15)$$

Then, we obtain $\left[w_{4i_1}^1, w_{4i_1+1}^1, w_{4i_1+2}^1, w_{4i_1+3}^1\right] = \left[\frac{7}{16}, \frac{13}{16}, \frac{13}{16}, \frac{15}{16}\right]$.

- The non-vanishing basis functions at v_i^2 (marked by red circles) are denoted by $v_{i_0}^0$ and $v_{i_1}^1$ (marked by green squares). The weighted sum is expressed as

$$h(s, t) = \sum_{j=0}^3 \left(w_{4i_0+j}^0 b_{4i_0+j}^0 + w_{4i_0+j}^1 b_{4i_0+j}^1 \right). \quad (16)$$

Submitting $h(s, t)$ into (14), the weights associated with v_i^2 are equal to $\left[\frac{7}{16}, \frac{13}{16}, \frac{13}{16}, \frac{15}{16} \right]$.

Fig. 3b shows a case where the weights for v_i^2 are equal to $[0, 0, 0, 0]$. The non-vanishing basis functions on F (the refined element) are also the non-vanishing basis functions at v_i^2 . Notice that the weights on level 1 are computed such that the basis functions form a partition of unity, so $h(s, t)|_F \equiv 1$, which means the right term in (14) is a zero vector. Since G is invertible, the weights associated with v_i^2 are thus zeros.

The basis functions correspond to zero weights has no contribution in the entire representation. Notice that zero weights occur if and only if the function g form a partition of unity on the domain occupied by the support mesh of the considered basis vertex. Therefore, to avoid this, the isolated refined elements, like the case in Fig. 3b, are not allowed.

In Fig. 3c, two adjacent elements are refined. The support of $v_{i_1}^1$ is shaded by yellow color in Figs. 3b and 3c. The support of $v_{i_1}^1$ in Fig. 3c becomes smaller, thus $h(s, t)$ do not form a partition of unity on F anymore. Therefore, the weights for v_i^2 are non-zeros.

Based on the above analysis, we made the following assumptions for the hierarchical T-mesh to define positively weighted bases that form a partition of unity.

Definition 2 A hierarchical T-mesh is called an admissible hierarchical T-mesh if there are not any isolated refined elements in the hierarchical T-mesh. If an element on level l is refined, but none of its adjacent elements are refined, it is called an isolated refined element. Here, if two elements of the same level share at least one edge, they are called adjacent elements.

Figs. 3a and 3c are two admissible hierarchical T-meshes, while Fig. 3b is not an admissible hierarchical T-mesh. A hierarchical T-mesh can become an admissible hierarchical T-meshes by performing refinement on adjacent elements of isolated refined elements.

To ensure the refinement is highly localized, we generally require the refinement level between any two neighboring elements cannot be greater than one, which is also required in [18]. Under this requirement, hierarchical T-meshes are admissible.

Proposition 1 There uniquely exists a set of positive weights for the bases constructed in [2] on the admissible hierarchical T-mesh such that the weighted bases form a partition of unity, that is there uniquely exists $w_1, w_2, \dots, w_n > 0$ such that $\sum_{i=1}^n w_i b_i(u, v) \equiv 1$ for $(u, v) \in [a, b] \times [c, d]$.

Proof The existence and uniqueness of the weights w_{4i+j}^l are implied by the linear system (13). In the following, we only prove $w_{4i+j}^l > 0$.

For the basis functions b_i^0 associated with the basis vertices on level 0, the function $h(s, t)$ defined in (11) is equal to zero, i.e., $h(s, t) = 0$, then the weights solved by (13) are all equal to ones.

For an element F in the initial mesh \mathbb{T}_0 , we denote the B-spline functions associated with the four corners of F by $b_{4i_0+j}, b_{4i_1+j}, b_{4i_2+j}, b_{4i_3+j}, j = 0, 1, 2, 3$. Remember that these basis functions are defined over \mathbb{T}_0 , instead of the final hierarchical T-mesh \mathbb{T} . Then, the non-vanishing basis functions on F are only these basis functions, which also form a partition of unity on F . As the refinement proceeds, the

supports of some of these basis functions are changed, and some new basis functions are added. The PHT-spline space defined over \mathbb{T}_0 are contained in the PHT-spline defined over \mathbb{T} , then b_{4i_0+j} , b_{4i_1+j} , b_{4i_2+j} , b_{4i_3+j} can be exactly represented by the basis functions b_{4i+k}^l on level l by knot insertion algorithm, that is, for $j = 0, 1, 2, 3$,

$$\begin{aligned} b_{4i_0+j} &= \sum_{l=0}^N \sum_{i=0}^{n_l} \sum_{k=0}^3 c_{ik}^l b_{4i+k}^l, \quad b_{4i_1+j} = \sum_{l=0}^N \sum_{i=0}^{n_l} \sum_{k=0}^3 d_{ik}^l b_{4i+k}^l, \\ b_{4i_2+j} &= \sum_{l=0}^N \sum_{i=0}^{n_l} \sum_{k=0}^3 e_{ik}^l b_{4i+k}^l, \quad b_{4i_3+j} = \sum_{l=0}^N \sum_{i=0}^{n_l} \sum_{k=0}^3 f_{ik}^l b_{4i+k}^l. \end{aligned} \quad (17)$$

Since

$$\sum_{j=0}^3 b_{4i_0+j} + b_{4i_1+j} + b_{4i_2+j} + b_{4i_3+j} \equiv 1, \text{ on } F,$$

then

$$\sum_{l=0}^N \sum_{i=0}^{n_l} \sum_{k=0}^3 (c_{ik}^l + d_{ik}^l + e_{ik}^l + f_{ik}^l) b_{4i+k}^l \equiv 1 \text{ on } F.$$

This implies that the weights

$$w_{4i+k}^l = c_{ik}^l + d_{ik}^l + e_{ik}^l + f_{ik}^l \geq 0, \quad (18)$$

because c_{ik}^l , d_{ik}^l , e_{ik}^l and f_{ik}^l are the coefficients in the knot insertion algorithm, thus are nonnegative values. On an admissible hierarchical T-mesh, all the weights are non-zeros, thus $w_{4i+k}^l > 0$.

In Fig. 4, we show the PHT surfaces defined over the same hierarchical T-mesh and by the same control points, but using different bases: the proposed weighted bases and non-decay bases [2]. The control points are designed on the plane $z = 1$. We see that the PHT surface defined by weighted basis functions form a partition of unity.

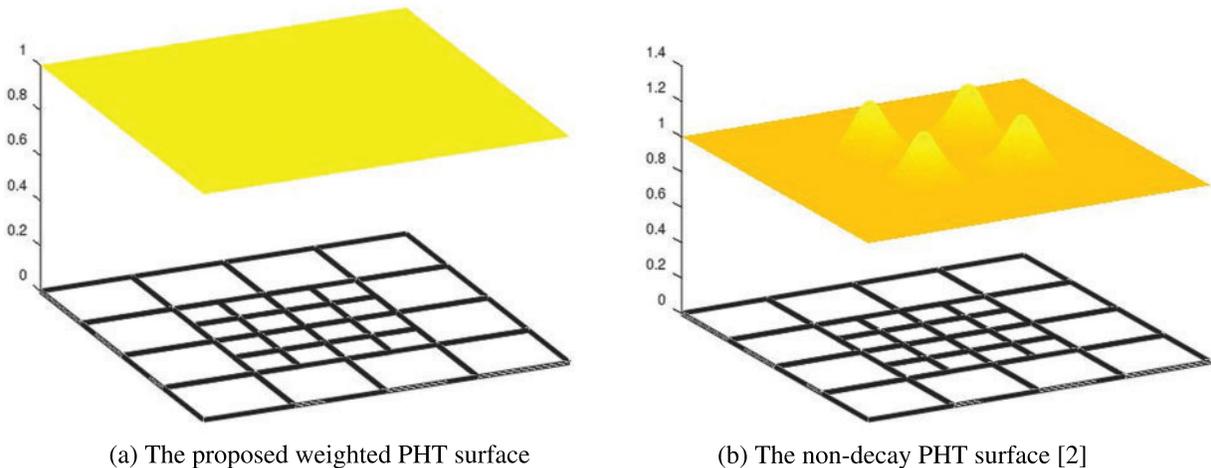


Figure 4: The weighted PHT-splines proposed in this paper form the partition of unity, while the PHT-splines defined by the non-decay bases [2] cannot form a partition of unity

We conclude the method of computing the weights such that the weighted basis functions form a partition of unity in Algorithm 1.

Algorithm 1: Calculation of weights

Input: k -th level hierarchical T-mesh with basis vertices set V

while $v_i \in V$

Find the support mesh of v_i and compute $\alpha, \beta, \lambda, \mu$ based on the formula (6);

Find all the basis functions that are not equal to zeros at v_i and get $h(s, t)$ defined by (11);

Calculate the weight w_{4i+j}^k of v_i according to the formula (14);

Output: A weighted k -th hierarchical T-mesh with basis function that forma partition of unity.

4 Subdivision Scheme for PHT-Splines

Considering a knot vector $\{s_0, s_0, s_1, s_1, s_2, s_2, s_3, s_3\}$, we insert a double knot $s' \in [s_1, s_2]$ in it. Then, we get the following relationship of B-splines before and after s' is inserted based on the B-spline knot insertion algorithm,

$$\begin{pmatrix} N^3[s_0, s_0, s_1, s_1, s_2][s] \\ N^3[s_0, s_1, s_1, s_2, s_2][s] \\ N^3[s_1, s_1, s_2, s_2, s_3][s] \\ N^3[s_1, s_2, s_2, s_3, s_3][s] \end{pmatrix} = W \begin{pmatrix} N^3[s_0, s_0, s_1, s_1, s'][s] \\ N^3[s_0, s_1, s_1, s', s'][s] \\ N^3[s_1, s_1, s', s', s_2][s] \\ N^3[s_1, s', s', s_2, s_2][s] \\ N^3[s', s', s_2, s_2, s_3][s] \\ N^3[s', s_2, s_2, s_3, s_3][s] \end{pmatrix},$$

$$W = \begin{pmatrix} 1 & \alpha_s & \alpha_s \beta_s & 0 & 0 & 0 \\ 0 & 1 - \alpha_s & (2 - \alpha_s - \beta_s) \beta_s & \beta_s^2 & 0 & 0 \\ 0 & 0 & (1 - \beta_s)^2 & (1 - \beta_s)(\beta_s + \gamma_s) & \gamma_s & 0 \\ 0 & 0 & 0 & (1 - \beta_s)(1 - \gamma_s) & 1 - \gamma_s & 1 \end{pmatrix}, \tag{19}$$

where $\alpha_s = \frac{s_2 - s'}{s_2 - s_0}$, $\beta_s = \frac{s_2 - s'}{s_2 - s_1}$ and $\gamma_s = \frac{s_3 - s'}{s_3 - s_1}$.

Based on the above formula, we derive the subdivision scheme for PHT-splines as follows. The support mesh of a basis vertex v_i^k is supposed to be $[s_{i0}, s_i, s_{i1}] \times [t_{j0}, t_j, t_{j1}]$. The knot intervals of the support mesh are denoted by $d_0 = s_i - s_{i0}, d_1 = s_{i1} - s_i, e_0 = t_j - t_{j0}, e_1 = t_{j1} - t_j$. When elements are refined, new basis vertices are added. The new basis vertices lying in the interior of the refined elements are called face vertices, while the new basis vertices lying on the edge of the mesh are called edge vertices. The old basis vertices are called vertex points. Under the assumption that the level difference in an admissible hierarchical T-mesh is not greater than 1, the subdivision scheme of PHT-splines are deduced as follows.

• **Vertex-points**

If the refinement changes the support mesh associated with v_i^k , the control points associated with v_i^k are updated as follows:

$$\begin{pmatrix} P_{4i}^{k+1} \\ P_{4i+1}^{k+1} \\ P_{4i+2}^{k+1} \\ P_{4i+3}^{k+1} \end{pmatrix} = \begin{pmatrix} w_{s1}w_{t1} & (1-w_{s1})w_{t1} & w_{s1}(1-w_{t1}) & (1-w_{s1})(1-w_{t1}) \\ w_{s2}w_{t1} & (1-w_{s2})w_{t1} & w_{s2}(1-w_{t1}) & (1-w_{s2})(1-w_{t1}) \\ w_{s1}w_{t2} & (1-w_{s1})w_{t2} & w_{s1}(1-w_{t2}) & (1-w_{s1})(1-w_{t2}) \\ w_{s2}w_{t2} & (1-w_{s2})w_{t2} & w_{s2}(1-w_{t2}) & (1-w_{s2})(1-w_{t2}) \end{pmatrix} \begin{pmatrix} P_{4i}^k \\ P_{4i+1}^k \\ P_{4i+2}^k \\ P_{4i+3}^k \end{pmatrix}. \quad (20)$$

where $w_{s1} = \frac{\mu_0 d_0 + 2\mu_1 d_1}{2\mu_0 d_0 + 2\mu_1 d_1}$, $w_{s2} = \frac{\mu_1 d_1}{2\mu_0 d_0 + 2\mu_1 d_1}$, $w_{t1} = \frac{\lambda_0 e_0 + 2\lambda_1 e_1}{2\lambda_0 e_0 + 2\lambda_1 e_1}$ and $w_{t2} = \frac{\lambda_1 e_1}{2\lambda_0 e_0 + 2\lambda_1 e_1}$. The parameters $\mu_0, \mu_1, \lambda_0, \lambda_1$ are used to indicate whether the knot intervals d_0, d_1, e_0, e_1 are refined, respectively. The parameter is equal to 1 implies the corresponding knot interval is subdivided. Especially when the parameters $\mu_0, \mu_1, \lambda_0, \lambda_1$ are zeros, control points are not updated, that is $P_{4i+j}^{k+1} = P_{4i+j}^k, j = 0, 1, 2, 3$. Fig. 5 shows different cases of refinement. In Fig. 5a, four elements in the support mesh of v_i^k are refined, now the parameters are set as $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [1, 1, 1, 1]$. In Figs. 5b and 5c, three elements are refined and the corresponding parameters are set as $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [0, 1, 0, 1]$ and $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [1, 0, 1, 0]$. In Figs. 5d and 5e, two elements are refined and the corresponding parameters are set as $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [0, 0, 1, 0]$ and $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [1, 0, 0, 0]$. In Fig. 5f, one element is refined. If v_i^k is on the boundary, $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [0, 1, 0, 1]$. Otherwise, $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [0, 0, 0, 0]$, which means the old vertex v_i^k is not updated.

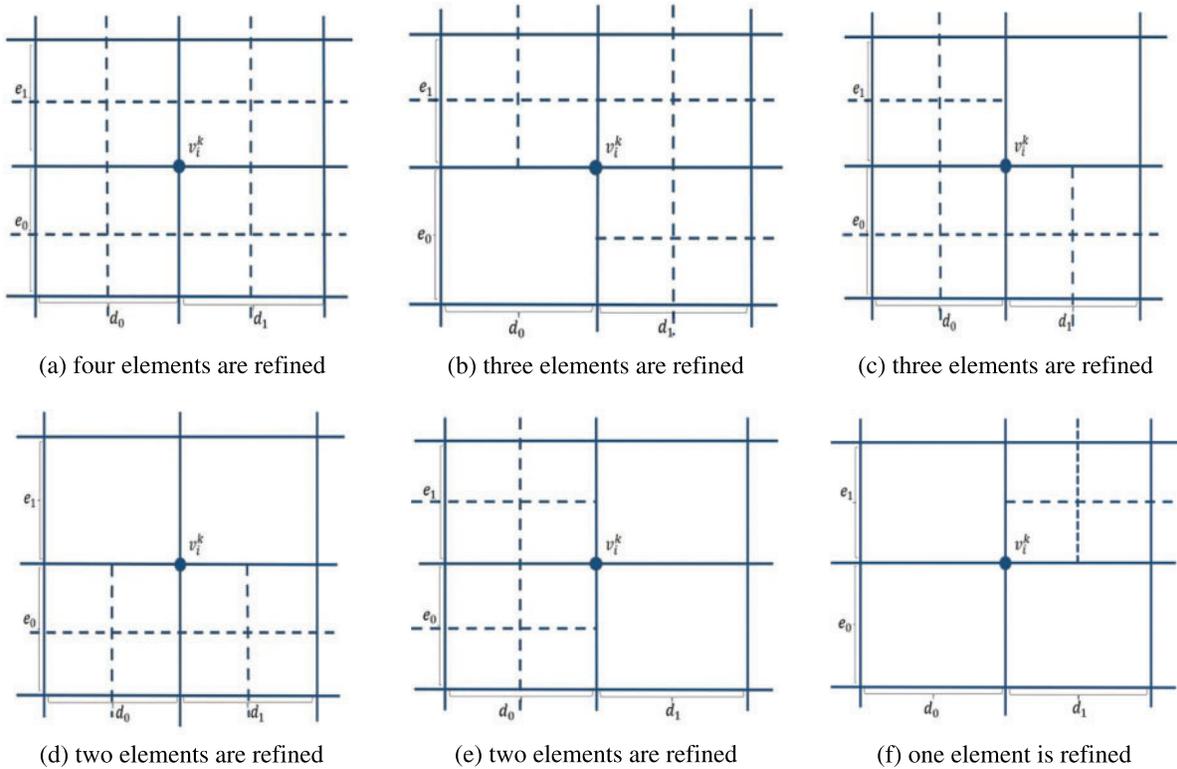


Figure 5: Subdivision scheme of vertex points

• **Edge-points**

Suppose v_i^{k+1} is an edge point lying on the edge with endpoints $v_{i_1}^k$ and $v_{i_2}^k$, referring to Fig. 6. Since the level difference is assumed to be less than 2, $v_{i_1}^k$ and $v_{i_2}^k$ are two basis vertices. As shown in Fig. 6a, the knot intervals for $v_{i_1}^k$ and $v_{i_2}^k$ are denoted by $\{d_0, d_1; e_0, e_1\}$ and $\{d_1, d_2; e_0, e_1\}$, respectively. We have the following edge-points updating formula.

$$\begin{aligned}
 P_{4i}^{k+1} &= \frac{1}{8(d_0 + d_1)(e_0 + e_1)} \left[(3d_0 + 2d_1)(e_0 + 2e_1)P_{4i+1}^k + d_1e_0P_{4i+2}^k + d_1(2e_1 + e_0)P_{4i}^k \right. \\
 &\quad \left. + (3d_0 + 2d_1)e_0P_{4i+3}^k + (d_0 + d_1)(e_0 + 2e_1)P_{4i_2}^k + (d_0 + d_1)e_0P_{4i_2+2}^k \right], \\
 P_{4i+1}^{k+1} &= \frac{1}{8(d_1 + d_2)(e_0 + e_1)} \left[(2d_1 + 3d_2)(e_0 + 2e_1)P_{4i_2}^k + d_1e_0P_{4i_2+3}^k + d_1(e_0 + 2e_1)P_{4i_2+1}^k \right. \\
 &\quad \left. + (2d_1 + 3d_2)e_0P_{4i_2+2}^k + (d_1 + d_2)(e_0 + 2e_1)P_{4i+1}^k + (d_1 + d_2)e_0P_{4i+3}^k \right], \\
 P_{4i+2}^{k+1} &= \frac{1}{8(d_0 + d_1)(e_0 + e_1)} \left[(3d_0 + 2d_1)(2e_0 + e_1)P_{4i+3}^k + d_1e_1P_{4i_1}^k + d_1(2e_0 + e_1)P_{4i+2}^k \right. \\
 &\quad \left. + (3d_0 + 2d_1)e_1P_{4i+1}^k + (d_0 + d_1)e_1P_{4i_2}^k + (d_0 + d_1)(2e_0 + e_1)P_{4i_2+2}^k \right], \\
 P_{4i+3}^{k+1} &= \frac{1}{8(d_1 + d_2)(e_0 + e_1)} \left[(2d_1 + 3d_2)(2e_0 + e_1)P_{4i_2+2}^k + d_1e_1P_{4i_2+1}^k + d_1(2e_0 + e_1)P_{4i_2+3}^k \right. \\
 &\quad \left. + (2d_1 + 3d_2)e_1P_{4i_2}^k + (d_1 + d_2)e_1P_{4i+1}^k + (d_1 + d_2)(2e_0 + e_1)P_{4i+3}^k \right]. \tag{21}
 \end{aligned}$$

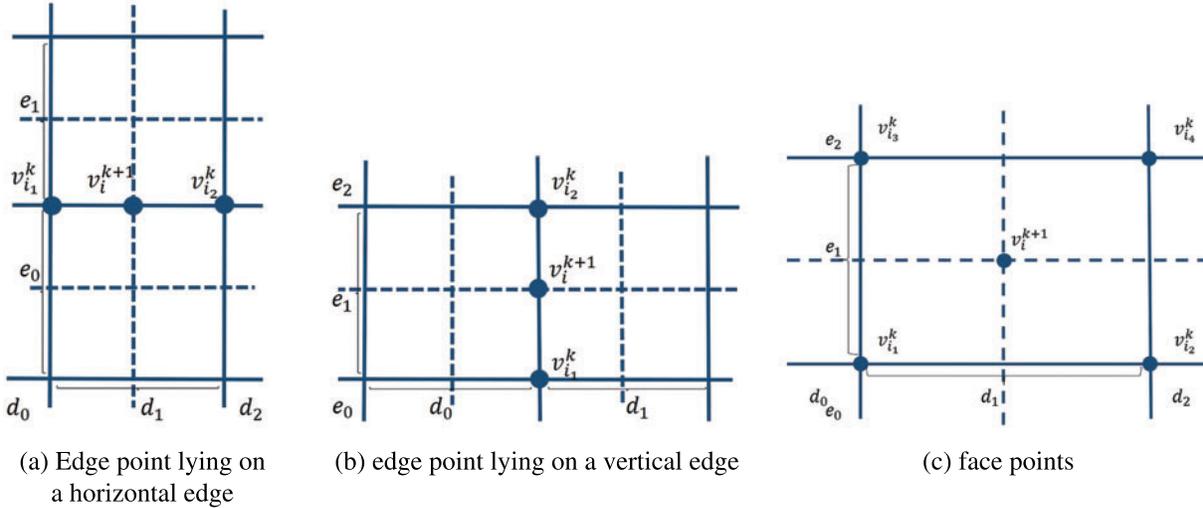


Figure 6: The subdivision scheme of edge points and face points

When the edge point lies on a vertical edge, as shown in Fig. 6b, the update formula is similar to the above one, except that e_i 's and d_i 's are swapped.

• Face-points

Since we assume the level difference in the mesh is not greater than 1, so when an element is refined, the four corners are exactly basis vertices. Suppose the four basis vertices of the refined face F are denoted by $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$. The knot intervals of these four basis vertices are denoted by $\{d_0, d_1, d_2; e_0, e_1, e_2\}$, see Fig. 6c for a reference. Here, we use the parameter $a_j = 1$ and $a_j = 0$ to indicate the support mesh of v_{i_j} is changed and unchanged, respectively. Let $P_{4i+j}^k := a_j P_{4i+j}^k$. We have the following formulas:

$$\begin{aligned}
P_{4i}^{k+1} &= \frac{1}{16(d_0 + d_1)(e_0 + e_1)} \left[d_1 e_1 P_{4i_1}^k + (3d_0 + 2d_1) e_1 P_{4i_1+1}^k + d_1 (3e_0 + 2e_1) P_{4i_1+2}^k \right. \\
&\quad + (3d_0 + 2d_1) (3e_0 + 2e_1) P_{4i_1+3}^k + (d_0 + d_1) e_1 P_{4i_2}^k + (d_0 + d_1) (3e_0 + 2e_1) P_{4i_2+2}^k + d_1 (e_0 + e_1) P_{4i_3}^k \\
&\quad \left. + (3d_0 + 2d_1) (e_0 + e_1) P_{4i_3+1}^k + (d_0 + d_1) (e_0 + e_1) P_{4i_4}^k \right], \\
P_{4i+1}^{k+1} &= \frac{1}{16(d_1 + d_2)(e_0 + e_1)} \left[(2d_1 + 3d_2) e_1 P_{4i_2}^k + d_1 e_1 P_{4i_2+1}^k + (2d_1 + 3d_2) (3e_0 + 2e_1) P_{4i_2+2}^k \right. \\
&\quad + d_1 (3e_0 + 2e_1) P_{4i_2+3}^k + (d_1 + d_2) e_1 P_{4i_1+1}^k + (d_1 + d_2) (3e_0 + 2e_1) P_{4i_1+3}^k \\
&\quad \left. + (2d_1 + 3d_2) (e_0 + e_1) P_{4i_4}^k + d_1 (e_0 + e_1) P_{4i_4+1}^k + (d_1 + d_2) (e_0 + e_1) P_{4i_3+1}^k \right], \\
P_{4i+2}^{k+1} &= \frac{1}{16(d_0 + d_1)(e_1 + e_2)} \left[d_1 (2e_1 + 3e_2) P_{4i_3}^k + d_1 e_1 P_{4i_3+2}^k + (3d_0 + 2d_1) (2e_1 + 3e_2) P_{4i_3+1}^k \right. \\
&\quad + (3d_0 + 2d_1) e_1 P_{4i_3+3}^k + d_1 (e_1 + e_2) P_{4i_1+2}^k + (3d_0 + 2d_1) (e_1 + e_2) P_{4i_1+3}^k + (d_0 + d_1) (2e_1 + 3e_2) P_{4i_4}^k \\
&\quad \left. + (d_0 + d_1) e_1 P_{4i_4+2}^k + (d_0 + d_1) (e_1 + e_2) P_{4i_2+2}^k \right], \tag{22} \\
P_{4i+3}^{k+1} &= \frac{1}{16(d_1 + d_2)(e_1 + e_2)} \left[d_1 (2e_1 + 3e_2) P_{4i_4+1}^k + d_1 e_1 P_{4i_4+3}^k + (2d_1 + 3d_2) e_1 P_{4i_4+2}^k \right. \\
&\quad + (2d_1 + 3d_2) (2e_1 + 3e_2) P_{4i_4}^k + (d_1 + d_2) (2e_1 + 3e_2) P_{4i_3+1}^k + (d_1 + d_2) e_1 P_{4i_3+3}^k + (e_1 + e_2) P_{4i_2+2}^k \\
&\quad \left. + d_1 (e_1 + e_2) P_{4i_2+3}^k + (d_1 + d_2) (e_1 + e_2) P_{4i_1+3}^k \right].
\end{aligned}$$

The weights satisfying (9) can be computed by this subdivision scheme. The control points (weights) at level 0 are all set to be 1. In the following, we take the hierarchical T-mesh shown in Fig. 3a as an example to show how to use subdivision scheme to compute weights.

The weights are computed level by level. Fig. 7 shows the refinement process. The weights of the basis functions on level 0 are set as $[1, 1, 1, 1]$, that is we set $P_{4i+j}^0 = w_{4i+j}^0 = 1$ in the three formulas (20)–(22). Then we compute the weights associated with the basis vertices on level 1, see Fig. 7b. The weights of the edge-point $v_{i_1}^1$ are calculated by (21) with $d_0 = 0.25, d_1 = 0.25, d_2 = 0.25, e_0 = 0.25$, and $e_1 = 0.25$. Consequently, we obtain $[w_{4*i_1}^1, w_{4*i_1+1}^1, w_{4*i_1+2}^1, w_{4*i_1+3}^1] = [1, 1, 1, 1]$. The weights of other edge-points are calculated similarly and are all equal to $[1, 1, 1, 1]$.

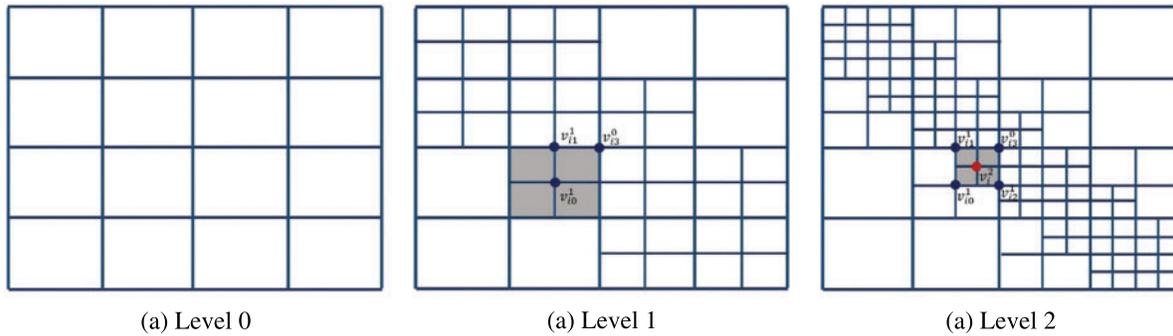


Figure 7: The weights of each level of mesh in Fig. 3a are calculated by the PHT-splines subdivision scheme

For the face point v_{i0}^1 , the indicators a_{js} for the corners v_{i1}^0 , v_{i2}^0 and v_{i3}^0 are all equal to 1, since their support meshes are changed, that is $a_1 = a_2 = a_3 = 1$. While $a_0 = 0$. The knot intervals for this case are set as $d_0 = 0.25, d_1 = 0.25, d_2 = 0.25, e_0 = 0.25, e_1 = 0.25, e_2 = 0.25$. The weights are thus computed based on (22) and are equal to $[w_{4*i1}^1, w_{4*i1+1}^1, w_{4*i1+2}^1, w_{4*i1+3}^1] = \left[\frac{7}{16}, \frac{13}{16}, \frac{13}{16}, \frac{15}{16} \right]$.

For the vertex point v_{i3}^0 , which is also a basis vertex of level 0, is updated based on (20), where $[\mu_0, \mu_1, \lambda_0, \lambda_1] = [1, 1, 1, 1], d_0 = d_1 = e_0 = e_1 = 0.25$ and $P_{4*i3+j}^0 = 1, j = 0, \dots, 3$. The weights of this vertex are all equal to 1. Notice here the support mesh of v_{i0}^0 is not changed, thus the weights are still equal to 1.

On level 2, the mesh is shown in Fig. 7c, the face points and edge points of level 1 now become vertex points. For example, the vertices marked by blue circles are vertex points and updated based on (20). Since all the support meshes of these vertices are not changed, their weights are equal to the weights computed on level 1. For the update of $v_{i1}^1, v_{i2}^1, v_{i3}^1$, because their weights are all $[1, 1, 1, 1]$, combined with sum of the coefficients of the update formula is 1, we easily obtain the updated weights of the three vertex-points are still $[1, 1, 1, 1]$. It is worth noting that if one of the basis functions of the three vertices has a weight other than 1, its weight will change after it is updated.

For the face point v_i^2 , the support of the basis vertex v_{i0}^1 is not changed, so its weight is also set to 0 in the face-points formula, and the weights of the other three vertices are not changed. With their knot intervals $d_0 = 0.125, d_1 = 0.125, d_2 = 0.125, e_0 = 0.125, e_1 = 0.125, e_2 = 0.125$, according to the formula of the face point, the weight of the new face point v_i^2 is also $\left[\frac{7}{16}, \frac{13}{16}, \frac{13}{16}, \frac{15}{16} \right]$.

5 Numerical Experiments

The weighted basis functions not only avoid the decay phenomenon, but also form a partition of unity. The non-decay property reduces the condition number of the stiffness matrix, thus gives better numerical stability. This advantage has been explored in [2] and will not be repeated in this paper. The focus of this paper is to provide a way to efficiently calculate weights for non-decay basis functions so that non-decay PHT splines can be used in specific applications, such as topology optimization [14].

In this section, three typical hierarchical T-meshes demonstrate the validity of the proposed methods. We see that the weights on admissible hierarchical T-meshes are all positive.

The hierarchical T-meshes here are produced in the context of iso-geometric analysis. The Poisson equation is defined as

$$\begin{cases} -\Delta u(x, y) = f(x, y), & (x, y) \in \Omega, \\ u|_{\Gamma} = g, & \Gamma = \partial\Omega. \end{cases} \quad (23)$$

The exact solution $u(x, y)$ is chosen as three functions defined on $[0, 1] \times [0, 1]$. The first one is a C^1 -continuous function defined by

$$u_1(x, y) = \begin{cases} (x - y)^2 (1.0 - x) y, & x > y, \\ -(x - y)^2 (1.0 - y) x, & x < y. \end{cases} \quad (24)$$

The other two are smooth functions with large gradients at some local regions,

$$u_2(x, y) = -\tanh\left(-0.25 + \frac{\sqrt{(x - 0.5)^2 + (y - 0.5)^2 + 0.01}}{0.03}\right), \quad (25)$$

and $u_3(x, y) = 1000x^6y^6(1.0 - x)^2(1.0 - y)^2$. The non-decay PHT-splines are adopted to adaptively solve the Poisson problem. Based on Algorithm 1 or the subdivision scheme, we compute the weights for the non-decay basis functions defined over the hierarchical T-meshes produced during the adaptive solving.

Fig. 8a shows the plot of $u_1(x, y)$. The diagonal elements are refined in each level to capture the feature of the solution. Figs. 8b and 8c show the hierarchical T-meshes at levels 2 and 4, where the weights of the basis vertices are equal to ones, except for the basis vertices that are marked by colored circles. Notice that there are two basis functions with zero weights on level 4 (Fig. 8c), because the underlying refined elements are isolated elements. This can be avoided by a further refinement of the adjacent elements. For this example, the level difference is less than 2. As the level increases, the weights of the old basis vertices are not changed. The weights of new basis vertices on each level have two kinds of values, except for the four ones at the corners.

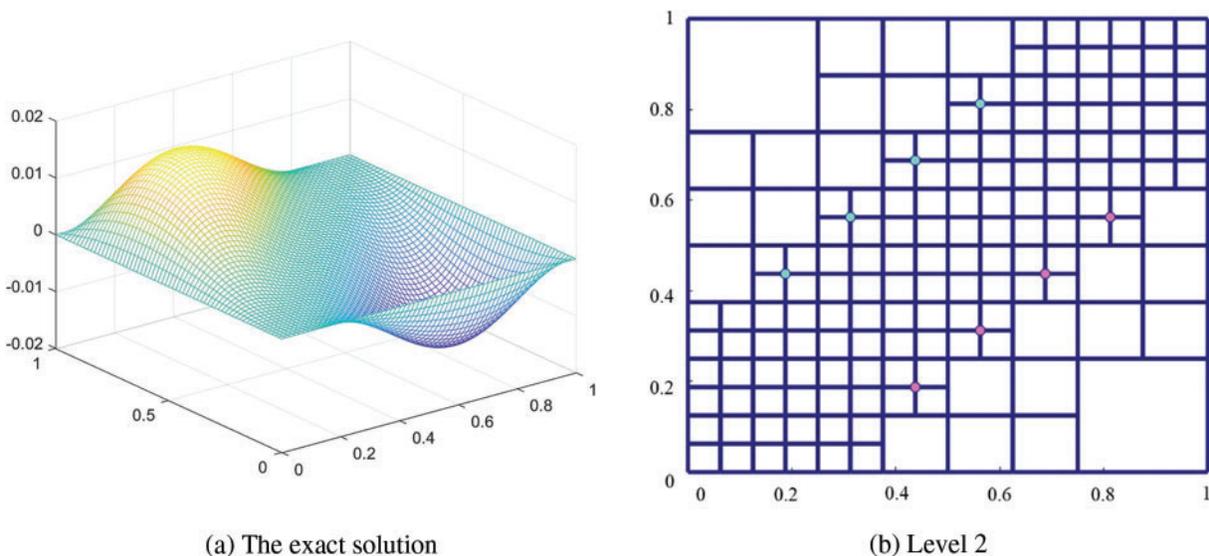


Figure 8: (Continued)

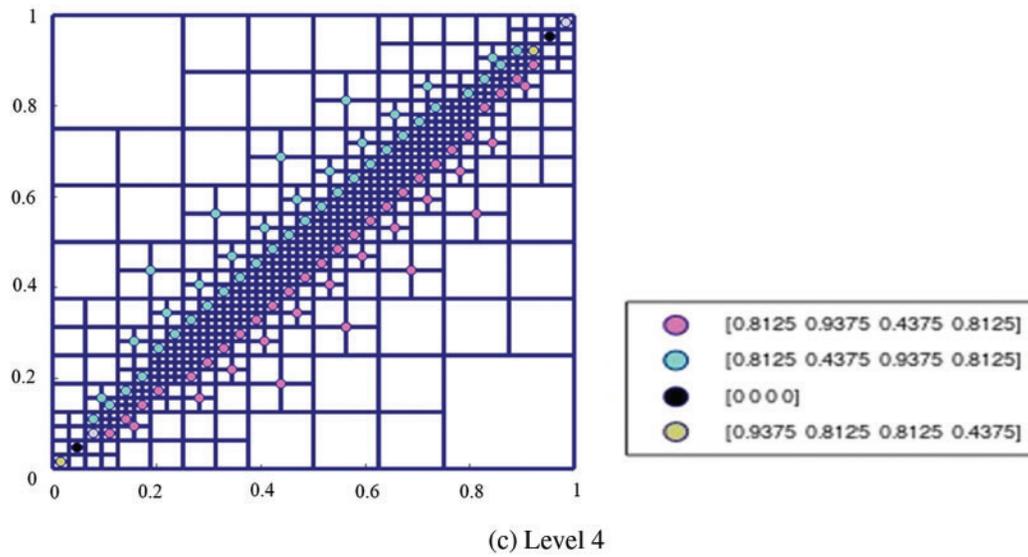


Figure 8: The weights of the basis functions on the hierarchical T-mesh with diagonal elements refined

Fig. 9a shows the plot of $u_2(x, y)$. This function has very large gradients around the circle centered at $(0.5, 0.5)$ and decays very fast from the value 1 to the value -1 . One can easily observe significant refinement around the large gradient area. Figs. 9b and 9c show the hierarchical T-meshes at level 2 and 4, where the weights of the basis vertices are equal to ones, except for the basis vertices that are marked by colored circles. As the level increases, the weights of the old basis vertices are not changed. The weights of new basis vertices on each level have two kinds of values.

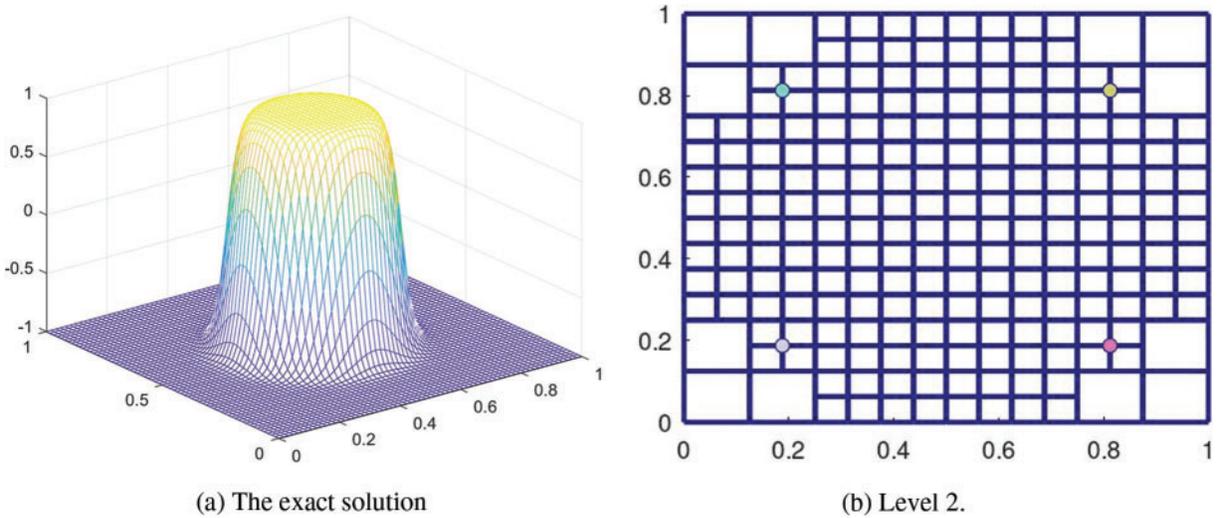


Figure 9: (Continued)

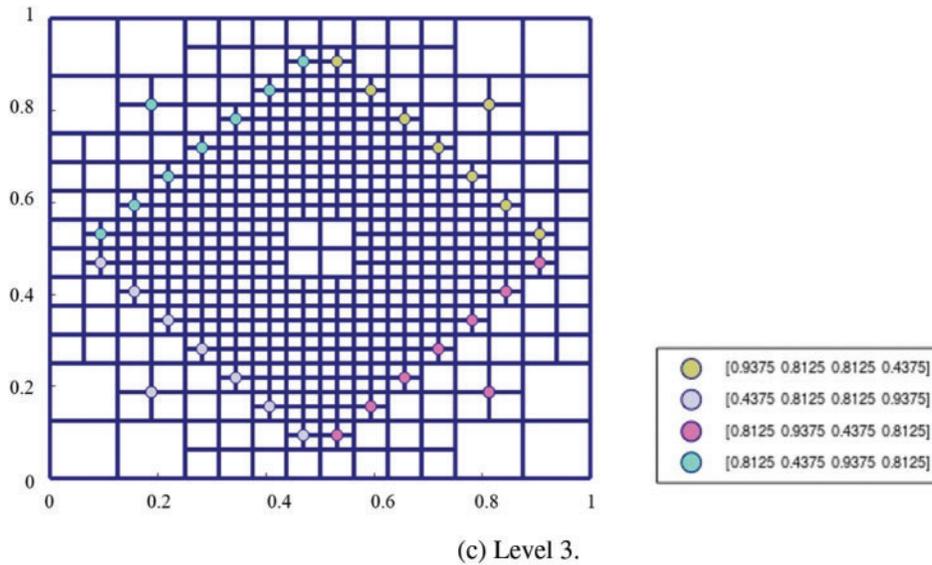


Figure 9: The weights of the basis functions on the hierarchical T-mesh solved from $u_2(x, y)$

Fig. 10a shows the plot of $u_3(x, y)$. For this example, the elements near the right-top corner are refined on each level to get hierarchical T-meshes that have greater level difference. Figs. 10b–10d show the hierarchical T-meshes at level 1, 2, 3, where the weights of the basis vertices are equal to ones, except for the basis vertices that are marked by colored circles. As the level increases, the weights of the old basis vertices are changed. This can be seen from Fig. 10e, where the weights of the basis vertex v_{i0}^1 are changed as the refinement proceeds. Moreover, the closer to the left-down corner of the element, the smaller the weights of the face points, such as the weights of v_{i0}^2 , v_{i0}^3 and v_{i0}^4 . This is caused by a greater level difference.

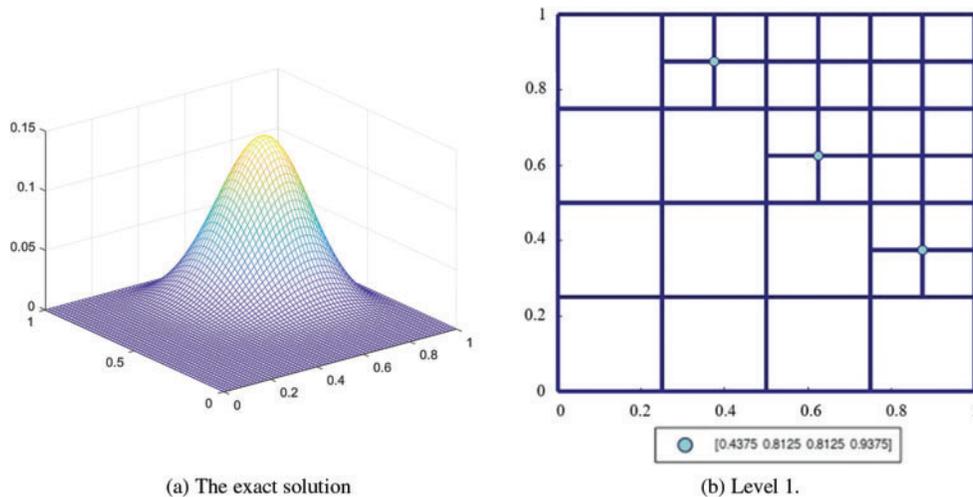


Figure 10: (Continued)

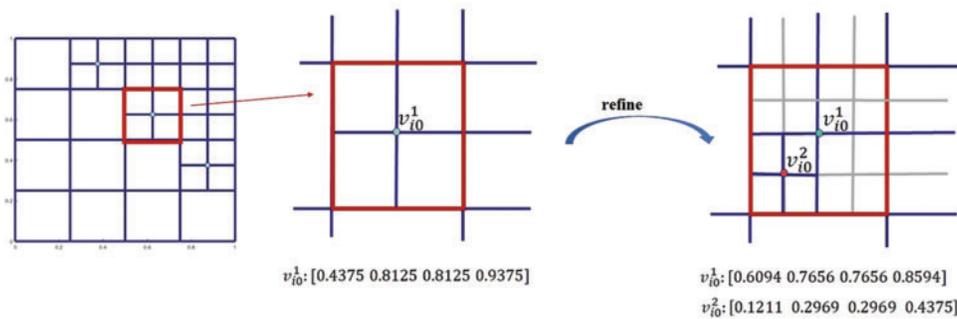
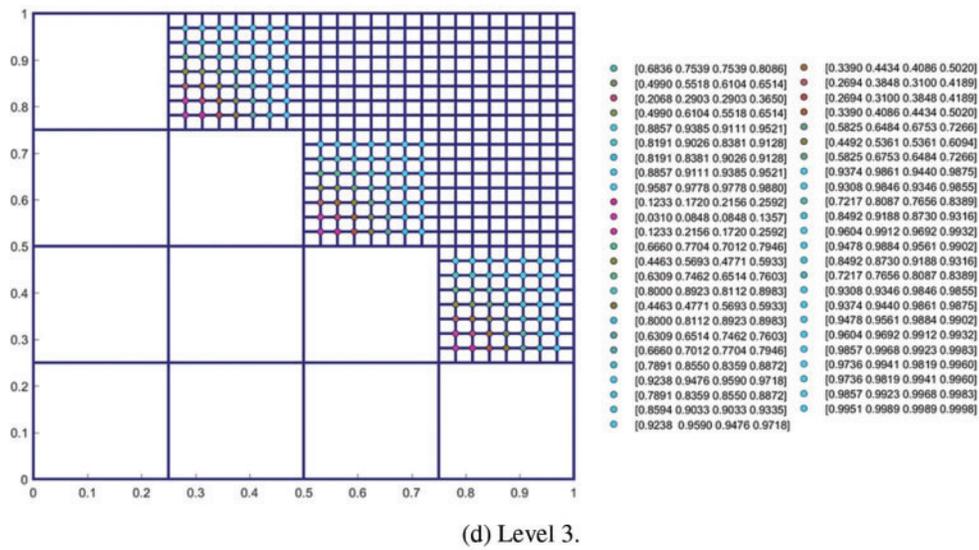
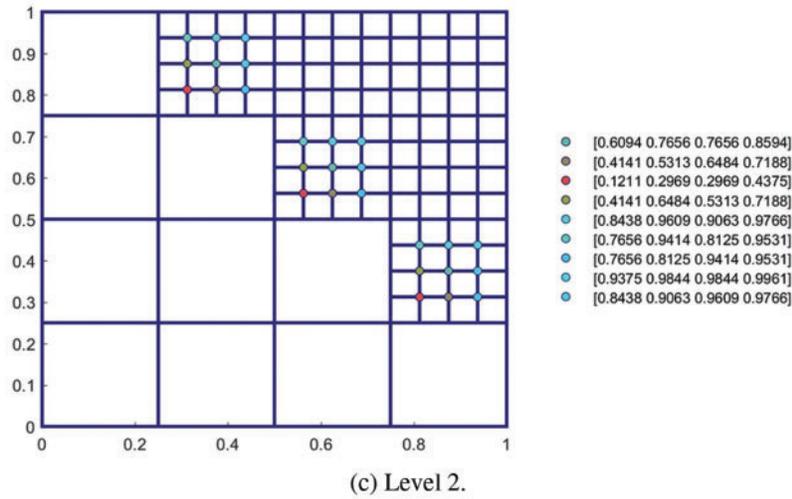
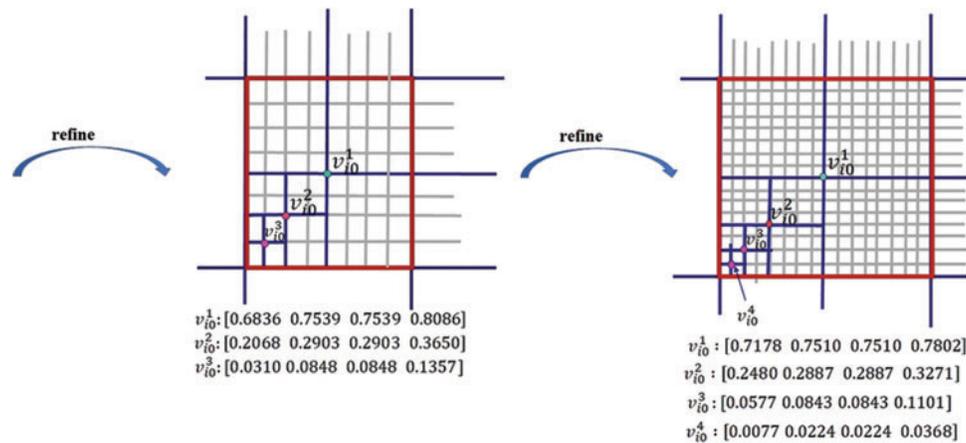


Figure 10: (Continued)



(e) The weights decrease gradually as the associated basis vertices are closer to the left-down corner of the element with red edges

Figure 10: The weights of the basis functions on the hierarchical T-mesh solved from $u_3(x, y)$. The level difference in these meshes is larger than 2

6 Conclusion

In this paper, we proposed two efficient ways of computing weights for the PHT-splines basis functions constructed in [2] such that the weighted basis functions form a partition of unity. The weights are expressed explicitly based on the geometric information of related basis functions or by the subdivision formula of HT-splines. Both algorithms can effectively compute weights. We proved that the weights are positive on the admissible hierarchical T-meshes. Three typical hierarchical T-meshes are demonstrated to verify the effectiveness of the proposed methods.

There are several problems worthy of further discussion. The first one is to derive the subdivision scheme in the case of arbitrary level differences. Another issue worthy of exploring is the application of PHT-splines defined by a weighted basis in topology optimization and isogeometric collocation.

Acknowledgement: The authors thank the reviewers for providing useful comments and suggestion.

Funding Statement: The work was supported by the NSF of China (No. 11801393) and the Natural Science Foundation of Jiangsu Province, China (No. BK20180831).

Author Contributions: Falai Chen contributed to the conception of the study; Zhiguo Yong and Hongmei Kang contributed significantly to data analyses and manuscript writing; Zhiguo Yong performed the experiment.

Availability of Data and Materials: None.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Deng, J., Chen, F., Li, X., Hu, C., Tong, W. et al. (2008). Polynomial splines over hierarchical T-meshes. *Graphical Models*, 70(4), 76–86.
2. Kang, H., Xu, J., Chen, F., Deng, J. (2015). A new basis for PHT-splines. *Graphical Models*, 82, 149–159.
3. Li, X., Deng, J., Chen, F. (2007). Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer*, 23(12), 1027–1033.
4. Wang, J., Yang, Z., Jin, L., Deng, J., Chen, F. (2010). Adaptive surface reconstruction based on implicit PHT-splines. *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling*, pp. 101–110. Haifa, Israel.
5. Tian, L., Chen, F., Du, Q. (2011). Adaptive finite element methods for elliptic equations over hierarchical T-meshes. *Journal of Computational and Applied Mathematics*, 236(5), 878–891.
6. Chan, C. L., Anitescu, C., Rabczuk, T. (2017). Volumetric parametrization from a level set boundary representation with PHT-splines. *Computer-Aided Design*, 82, 29–41.
7. Nguyen-Thanh, N., Kiendl, J., Nguyen-Xuan, H., Wüchner, R., Bletzinger, K. U. et al. (2011). Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48), 3410–3424.
8. Nguyen-Thanh, N., Muthu, J., Zhuang, X., Rabczuk, T. (2014). An adaptive three-dimensional RHT-splines formulation in linear elasto-statics and elasto-dynamics. *Computational Mechanics*, 53(2), 369–385.
9. Nguyen-Thanh, N., Nguyen-Xuan, H., Bordas, S. P. A., Rabczuk, T. (2011). Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(21–22), 1892–1908.
10. Nguyen-Thanh, N., Zhou, K. (2017). Extended isogeometric analysis based on PHT-splines for crack propagation near inclusions. *International Journal for Numerical Methods in Engineering*, 112(12), 1777–1800.
11. Videla, J., Anitescu, C., Khajah, T., Bordas, S. P., Atroshchenko, E. (2019). h- and p-adaptivity driven by recovery and residual-based error estimators for PHT-splines applied to time-harmonic acoustics. *Computers & Mathematics with Applications*, 77(9), 2369–2395.
12. Wang, P., Xu, J., Deng, J., Chen, F. (2011). Adaptive isogeometric analysis using rational PHT-splines. *Computer-Aided Design*, 43(11), 1438–1448.
13. Yang, H., Dong, C. (2019). Adaptive extended isogeometric analysis based on PHT-splines for thin cracked plates and shells with Kirchhoff–Love theory. *Applied Mathematical Modelling*, 76, 759–799.
14. Gupta, A., Mamindlapelly, B., Karuthedath, P. L., Chowdhury, R., Chakrabarti, A. (2022). Adaptive isogeometric topology optimization using PHT splines. *Computer Methods in Applied Mechanics and Engineering*, 395, 114993.
15. Li, X., Deng, J., Chen, F. (2010). Polynomial splines over general T-meshes. *The Visual Computer*, 26(4), 277–286.
16. Ni, Q., Wang, X., Deng, J. (2019). Modified PHT-splines. *Computer Aided Geometric Design*, 73, 37–53.
17. Melenk, J. M., Babuška, I. (1996). The partition of unity finite element method: Basic theory and applications. *Computer Methods Applied Mechanics Engineering*, 139, 289–314.
18. Babuška, I., Melenk, J. M. (1997). The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40, 727–758.
19. Pask, J. E., Sukumar, N. (2017). Partition of unity finite element method for quantum mechanical materials calculations. *Extreme Mechanics Letters*, 11, 8–17.
20. Giannelli, C., Juttler, B., Speleers, H. (2012). THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7), 485–498.
21. Wei, X., Zhang, Y., Liu, L., Hughes, T. J. (2017). Truncated T-splines: Fundamentals and methods. *Computer Methods in Applied Mechanics and Engineering*, 316, 349–372.

22. Zhu, Y., Chen, F. (2017). Modified bases of PHT-splines. *Communications in Mathematics and Statistics*, 5(4), 381–397.
23. Deng, J., Chen, F., Feng, Y. (2006). Dimensions of spline spaces over T-meshes. *Journal of Computational and Applied Mathematics*, 194(2), 267–283.