Journal of Computational Mathematics
Vol.35, No.1, 2017, 91–120.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1601-m2014-0175

BASES OF BIQUADRATIC POLYNOMIAL SPLINE SPACES OVER HIERARCHICAL T-MESHES*

Fang Deng and Chao Zeng
School of Mathematical Sciences, University of Science and Technology of China, Hefei, China
Email: dengfang@mail.ustc.edu.cn, zengchao@mail.ustc.edu.cn
Meng Wu
School of Mathematics, Hefei University of Technology, Hefei, China
Email: wumeng@mail.ustc.edu.cn
Jiansong Deng
School of Mathematical Sciences, University of Science and Technology of China, Hefei, China
Email: dengjs@ustc.edu.cn

Abstract

Basis functions of biquadratic polynomial spline spaces over hierarchical T-meshes are constructed. The basis functions are all tensor-product B-splines, which are linearly independent, nonnegative and complete. To make basis functions more efficient for geometric modeling, we also give out a new basis with the property of unit partition. Two preliminary applications are given to demonstrate that the new basis is efficient.

Mathematics subject classification: 65D07
Key words: Spline spaces over T-meshes, CVR graph, Basis functions.

1. Introduction

NURBS is a basic tool in surface modeling and isogeometric analysis (IGA). However, NURBS suffer from the weakness that the control points must lie topologically in a tensor-product mesh. If we want to construct a surface which is flat in the most part of the domain, but sharp in a small region, we have to use more control points not only in the sharp region, but also in the flat region to maintain the tensor-product structure. The superfluous control points are a burden. To overcome this limitation, we need splines which can be refined locally. This type of splines are defined on T-meshes which allow T-junctions.

Hierarchical B-splines were first introduced in [13], defined on a nest sequence of hierarchical meshes that can be locally refined, maintain a local tensor product structure and rely on the principle of B-spline subdivision. The hierarchical structure and the linear independence of hierarchical B-splines make it appealing for isogeometric analysis [15, 28]. For the partition of unity, THB-splines were introduced in [14]. However, the completeness of hierarchical B-splines can not be guaranteed generally.

In 2003, T-splines [20] were proposed. T-splines have been widely used in geometric modeling [21, 22] and IGA [1, 2, 5, 24]. However, the use of the T-splines in analysis has exhibited a number of problems that have been addressed by the researchers, such as the linear dependence of blending functions [4]. Analysis-suitable T-splines were introduced in [16, 17, 23] to deal with the problem. The linear independence of analysis-suitable T-splines makes it suitable for

* Received January 6, 2015 / Revised version received December 22, 2015 / Accepted January 14, 2016 / Published online January 18, 2017 /

analysis. However, analysis-suitable T-splines are defined on the analysis-suitable T-meshes. We need to refine more cells to convert a T-mesh to an analysis-suitable T-mesh.

LR B-splines [9] as a collection of hierarchically scaled B-splines are defined on a special type of T-meshes (LR-meshes). LR B-splines are not always linearly independent although there exists an algorithm to check whether the spline functions are linearly independent and convert the spline functions to be linearly independent by inserting more control points.

The splines mentioned above are not defined from the viewpoint of space. Therefore, the linear independence and completeness are two challenges. In 2006, spline spaces over T-meshes [6] were put forward by one of the present authors. This type of splines is defined directly from the viewpoint of spline spaces, dimension formula and basis functions are the main problems. In [6], each function in $\mathbf{S}(m,n,\alpha,\beta,\mathscr{T})$ over a T-mesh $\mathscr{T}$ is a bi-degree (m,n) polynomial in each cell of $\mathscr{T}$ with smoothness order α and β along two directions. A dimension formula is given under the condition of $m \geq 2\alpha+1$ and $n \geq 2\beta+1$. In 2008, polynomial splines over hierarchical T-meshes (PHT-splines) [7], which usually refer to bicubic C^1 polynomial splines over hierarchical T-meshes, were proposed. PHT-splines have been used efficiently and adaptively in isogeometric analysis [25–27, 31] and surface modeling [18, 32]. However, excessive growth of dimension is a challenge in the application of PHT-splines, especially in three-dimensional space. To deal with this problem, we consider the spline spaces $\mathbf{S}(m,n,m-1,n-1,\mathscr{T})$. Unfortunately, [3, 19] observed that the dimension of $\mathbf{S}(m,n,m-1,n-1,\mathscr{T})$ may depend on the geometry of the T-mesh. Thus, more attention is paid to some special classes of T-meshes.

It is noted that [29] gave a general dimension formula of $\mathbf{S}(m,n,m-1,n-1,\mathscr{T})$ over a special T-mesh ((m,n)-subdivided T-mesh). Hierarchical bases of $\mathbf{S}(m,n,m-1,n-1,\mathscr{T})$ over (m,n)-subdivided T-mesh were constructed in [30]. However, hierarchical bases in [30] did not have the properties of non-negativity and partition of unity. [34] gave a dimension formula of $\mathbf{S}(3,3,2,2,\mathscr{T})$ over a more general T-mesh than that in [29], but also a special hierarchical T-mesh. A basis of $\mathbf{S}(3,3,2,2,\mathscr{T})$ was given in [33] under three rules of refinement of the hierarchical T-mesh.

In this paper, we give a basis of $\mathbf{S}(2,2,1,1,\mathscr{T})$ based on the topological explanation of dimension formula in [8]. The basis functions are all tensor-product B-splines, which are linearly independent, nonnegative and complete. As unit partition is an important property for geometry modeling, a new basis with the property of unit partition is given out by an elementary transformation of the previous basis functions. Compared with spline bases in [30], the new basis has good properties of nonnegativity, partition of unity, and smaller local support which may result in a more stable stiffness matrix in numerical solutions of PDE. Also, these properties will facilitate their applications in geometric modeling. In fact, when representing a geometric shape with a linear combination of basis functions satisfying these properties, we can implement local shape control, affine invariance, and variational diminishing. See [11] for details. In geometric modeling, spline functions with C^1 are necessary in some situations.

The remainder of this paper is organized as follows. In Section 2, the spline spaces over T-meshes are reviewed, and several results from paper [8] are listed. In Section 3, the rule of refinement is elaborated. Construction of basis functions is discussed in Section 4. Section 5 gives the proof of the linear independence of the basis. In Section 6, a new basis with unit partition is given. In Section 7, we will present two preliminary applications of the new basis. We end the paper with conclusions and future work in Section 8.

2. T-meshes and Spline Spaces

In this section, the concepts of T-meshes, hierarchical T-meshes and the spline spaces over T-meshes are proposed. The dimension formula of $\mathbf{S}(2,2,1,1,\mathscr{T})$ in [8] is reviewed.

2.1. T-meshes and hierarchical T-meshes

A T-mesh is a rectangular grid that allows T-junctions.

Definition 2.1. ([8]) *Suppose $\mathscr{T}$ is a set of axis-aligned rectangles and the intersection of any two distinct rectangles in $\mathscr{T}$ is either empty or consists of points on the boundaries of the rectangles. Then $\mathscr{T}$ is called a T-mesh. Furthermore, if the entire domain occupied by $\mathscr{T}$ is a rectangle, $\mathscr{T}$ is called a regular T-mesh. If some edges of $\mathscr{T}$ also form a T-mesh $\mathscr{T}^*$, $\mathscr{T}^*$ is called a submesh of $\mathscr{T}$.*

In this paper, we only consider regular T-meshes. We adopt the definitions of vertex, edge, and cell provided in [6]. A grid point in a T-mesh is called a vertex of the T-mesh. If a vertex is on the boundary grid line of the T-mesh, it is called a boundary vertex. Otherwise, it is called an interior vertex. There are two types of interior vertices: crossing vertices and T-vertices. A T-vertex is a vertex of one rectangle that lies in the interior of an edge of another rectangle. A crossing vertex is a vertex of four rectangles. In Fig. 2.1, $b_i, i = 1, \cdots, 10$ are boundary vertices, while $v_i, i = 1, \cdots, 5$ are interior vertices. Among the interior vertices, v_2 is a crossing vertex and the others are T-vertices. An l-edge is a line segment that consists of several edges and whose end points are boundary vertices or T-vertices. It is the longest possible line segment in the mesh. An l-edge is called a boundary l-edge if it consists of several boundary edges, otherwise it is called an interior l-edge. Obviously, a regular T-mesh has four boundary l-edges. In Fig. 2.1, there are five interior l-edges (labeled with dashed lines) and four boundary l-edges (labeled with solid lines).

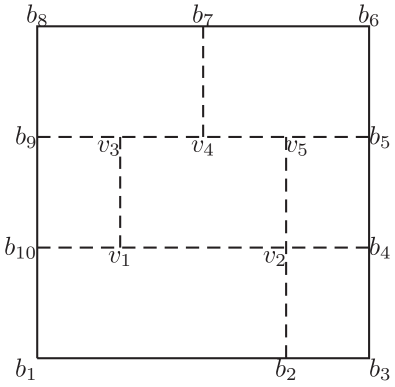


Fig. 2.1. A T-mesh $\mathscr{T}$ with notations.

A hierarchical T-mesh [7] is a special type of T-mesh that has a natural level structure. A hierarchical T-mesh is defined in a recursive manner. We generally start from a tensor-product mesh (level 0) $\mathscr{T}^0$. Then some cells of level k are each divided into 2×2 subcells equally, where the new cells, the new edges and the new vertices are of level $k+1$, the resulting mesh is called $\mathscr{T}^{k+1}$.

Definition 2.2. ([8]) *In forming a hierarchical T-mesh, from level k to level $k+1$, if there exists a cell of level k to be subdivided for which all horizontal and vertical neighboring cells of level k remain unchanged, the cell is called an isolated subdivided cell. Here, we require that at least one neighboring cell exists. Furthermore, if an isolated subdivided cell is not a boundary cell* (*whose four boundary edges are interior edges of the mesh*)*, the cell is called a non-boundary isolated subdivided cell.*

In Fig. 2.2, the cell bounded by lines in yellow is a non-boundary isolated subdivided cell, the cell bounded by lines in red is a boundary isolated subdivided cell, the cell bounded by lines in blue and the cell bounded by lines in green are not isolated subdivided cells.

Given a T-mesh $\mathscr{T}$, the extended T-mesh $\mathscr{T}^\varepsilon$ [8] associated with $\mathscr{T}$ is generated as follows: Produce a tensor-product mesh $\mathscr{M}$ with $2(m+1)$ vertical lines and $2(n+1)$ horizontal lines, the central rectangle of which is identical to the region occupied by $\mathscr{T}$, where (m,n) is the bidegree of the spline space over $\mathscr{T}$. Then extend the edges with one end point on the boundary of $\mathscr{T}$ to the boundary of $\mathscr{M}$.

Given a T-mesh $\mathscr{T}$, By retaining the crossing vertices and the lines with two end points that are both crossing points and removing other vertices and edges in $\mathscr{T}$, we obtain a new graph called a crossing-vertex-relationship graph (CVR graph for short) [8] of $\mathscr{T}$. We designate the CVR graph as $\mathscr{G}$. From the construction of $\mathscr{G}$, we know that $\mathscr{G}$ consists of all the line segments whose two end points are crossing vertices in $\mathscr{T}$ and the vertices of $\mathscr{G}$ are the crossing vertices of $\mathscr{T}$. Fig. 2.3 shows an example. In Fig. 2.3, the extended T-mesh $\mathscr{T}^\varepsilon$ is the extension of $\mathscr{T}$ associated with $\mathbf{S}(2,2,1,1,\mathscr{T})$.

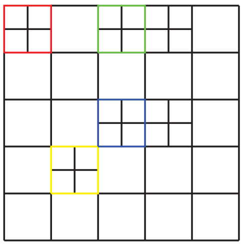


Fig. 2.2. Isolated subdivided cells in a T-mesh $\mathscr{T}$.

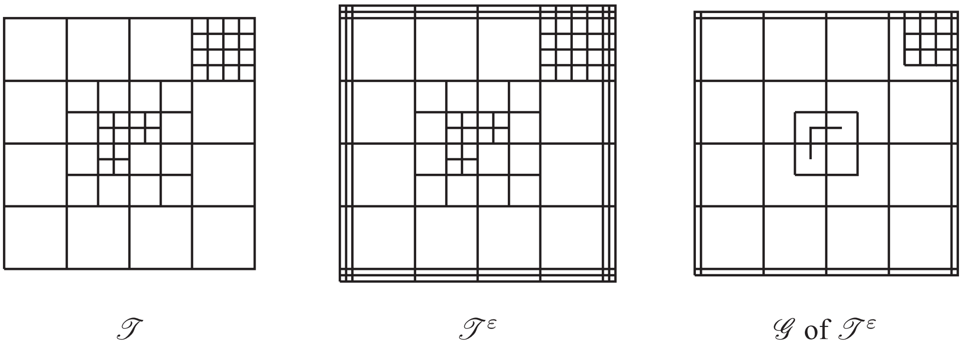

$\mathscr{T}$
$\mathscr{T}^\varepsilon$
$\mathscr{G}$ of $\mathscr{T}^\varepsilon$

Fig. 2.3. A T-mesh $\mathscr{T}$ with its associated extension $\mathscr{T}^\varepsilon$ and the CVR graph $\mathscr{G}$ of $\mathscr{T}^\varepsilon$.

2.2. Spline spaces over T-meshes

Given a T-mesh $\mathscr{T}$, $\mathscr{C}$ is used to denote all of the cells in $\mathscr{T}$, and Ω is used to denote the region occupied by the cells in $\mathscr{T}$. Spline spaces over the T-mesh [6] are defined as follows:

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

where $\mathbb{P}_{mn}$ is the space of the polynomials with bi-degree (m,n), and $C^{\alpha,\beta}(\Omega)$ is the space consisting of all the bivariate functions that are continuous in Ω with order α along the x direction and with order β along the y direction. Setting $(m,n)=(2,2)$ and $(\alpha,\beta)=(1,1)$, we obtain the spline space $\mathbf{S}(2,2,1,1,\mathscr{T})$ that is discussed in this paper.

For a given T-mesh $\mathscr{T}$, we define a spline space over the given T-mesh $\mathscr{T}$ with homogeneous boundary conditions (HBC) [8] as follows:

$$\overline{\mathbf{S}}(m,n,\alpha,\beta,\mathscr{T}) := \Big\{ f(x,y) \in C^{\alpha,\beta}(\mathbb{R}^2) : f(x,y)|_{\phi} \in \mathbb{P}_{mn}, \quad \forall \phi \in \mathscr{C}, \text{ and } f|_{\mathbb{R}^2\setminus\Omega} \equiv 0 \Big\}. \tag{2.2}$$

For the dimension formula of $\mathbf{S}(m,n,\alpha,\beta,\mathscr{T})$ over hierarchical T-mesh $\mathscr{T}$, three theorems in [8] are listed below.

Theorem 2.1. *Given a T-mesh $\mathscr{T}$, assume that $\mathscr{T}^{\varepsilon}$ is its extension associated with $\mathbf{S}(2,2,1,1,\mathscr{T})$ and that Ω is the region occupied by the cells in $\mathscr{T}$. Then,*

$$\mathbf{S}(2,2,1,1,\mathscr{T}) = \overline{\mathbf{S}}(2,2,1,1,\mathscr{T}^{\varepsilon})|_{\Omega}, \tag{2.3}$$

$$\dim \mathbf{S}(2,2,1,1,\mathscr{T}) = \dim \overline{\mathbf{S}}(2,2,1,1,\mathscr{T}^{\varepsilon}). \tag{2.4}$$

With Theorem 2.1, to consider the spline space over a T-mesh, we need only to consider the corresponding spline space with homogeneous boundary conditions over its extended T-mesh. So we only discuss the construction of the spline space $S(2,2,1,1,\mathscr{T})$ with homogeneous boundary conditions in this paper.

Theorem 2.2. *Suppose $\mathscr{T}$ is a hierarchical T-mesh with V^c crossing vertices, E interior l-edges and $\delta-1$ isolated subdivided cells. Then,*

$$\dim \overline{\mathbf{S}}(2,2,1,1,\mathscr{T}) = V^c - E + \delta. \tag{2.5}$$

Theorem 2.3. *Given a hierarchical T-mesh $\mathscr{T}$ with V^c crossing vertices, E interior l-edges and $\delta-1$ isolated subdivided cells, suppose there are $N_{\mathscr{G}}$ cells in its CVR graph $\mathscr{G}$. It follows that*

$$V^c - E + \delta = N_{\mathscr{G}}. \tag{2.6}$$

From Theorems 2.2 and 2.3, we have

Corollary 2.1. *Given a hierarchical T-mesh $\mathscr{T}$ with $N_{\mathscr{G}}$ cells in its CVR graph $\mathscr{G}$. Then*

$$\dim \overline{\mathbf{S}}(2,2,1,1,\mathscr{T}) = N_{\mathscr{G}}. \tag{2.7}$$

In the following, we find a basis of the spline space with the help of Corollary 2.1. The hierarchical T-mesh $\mathscr{T}$ we consider is the extension of some T-mesh $\mathscr{T}'$, so $\mathscr{T}$ is at least a 6 by 6 tensor-product mesh at level 0.

3. The Rule and Algorithm of Refinement

3.1. The rule of the refinement

A tensor-product B-spline in $\mathbf{S}(2,2,1,1,\mathscr{T})$ is defined on a 4×4 tensor-product submesh whose CVR graph is a 2×2 tensor-product mesh. We call the smallest 2×2 tensor-product submesh of $\mathscr{G}$ a basic mesh, where the smallest 2×2 tensor-product submesh $\mathscr{G}^{'}$ of $\mathscr{G}$ means that it does not contain other 2×2 tensor-product submeshes whose grid lines crossing $\mathscr{G}^{'}$. The domain occupied by the cells in a basic mesh is called a basic domain. We denote the basic domain corresponding to a tensor-product B-spline f as $\phi(f)$. In Fig. 3.1, the 2×2 tensor-product mesh bounded by red lines is not a basic mesh, because it contains a 2×2 tensor-product mesh (bounded by green lines) with grid lines crossing it, the 2×2 tensor-product mesh bounded by blue lines is a basic mesh, because it does not contain any 2×2 tensor-product mesh with grid lines crossing it.

Each 4×4 tensor-product submesh of $\mathscr{T}$ corresponds to a basic mesh of $\mathscr{G}$, however, a basic mesh does not necessarily correspond to a 4×4 submesh of $\mathscr{T}$. See Fig. 3.1. For the basic mesh bounded by dashed lines in $\mathscr{G}$, we can not find its corresponding 4×4 tensor-product submesh in $\mathscr{T}$. The reason is that the level-difference of some adjacent cells is greater than 1. In Fig. 3.1, the cell filled with crossed lines is of level 0, while its adjacent cell filled with slashed lines is of level 2, the level difference of the two cells are 2. We have the following lemma.

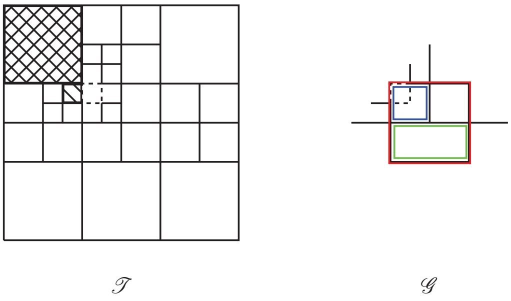


Fig. 3.1. A T-mesh $\mathscr{T}$ with its CVR graph $\mathscr{G}$.

Lemma 3.1. *Given a hierarchical T-mesh $\mathscr{T}$ and its CVR graph $\mathscr{G}$, if the level difference of the adjacent cells is at most one, then each basic mesh of $\mathscr{G}$ corresponds to a 4×4 tensor-product submesh of $\mathscr{T}$.*

Proof. To prove this fact, we classify the basic meshes and consider the basic meshes in the refinement process. For $\mathscr{T}^0$, the fact is obvious. Suppose that the fact is true for $\mathscr{T}^k$, we refine some cells of $\mathscr{T}^k$ and obtain $\mathscr{T}^{k+1}$. With the restriction of the level difference less than 2, all types of basic meshes and their corresponding 4×4 tensor-product meshes in $\mathscr{T}^{k+1}$ are discussed in the following. In Fig. 3.2 and 3.3, the basic meshes are bounded by the dashed lines and the corresponding 4×4 tensor-product meshes are bounded by the dotted lines.

1. The basic mesh of $\mathscr{G}$ corresponds to a single cell in $\mathscr{T}^{k+1}$. See Fig. 3.2. Here we assume edges v_1v_2 and v_1v_4 are of level $k+1$ and edges v_5v_3 and v_3v_6 are of level k:

   (a) If cell 1 is not refined, the situation is as Fig. 3.2a.

   (b) If cell 1 is refined, the situation is as Fig. 3.2b.

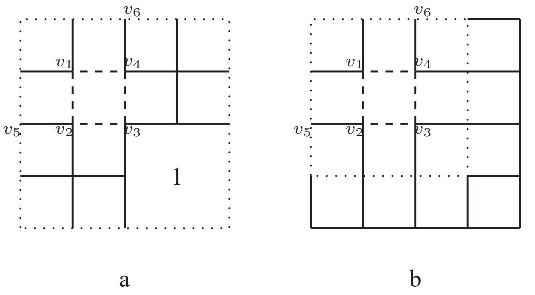


Fig. 3.2. The basic mesh of $\mathscr{G}$ corresponds to a single cell in $\mathscr{T}^{k+1}$.

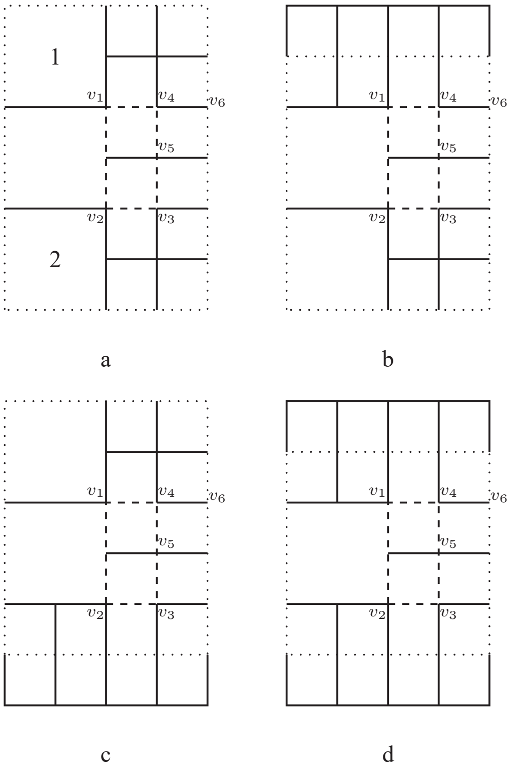


Fig. 3.3. The basic mesh of $\mathscr{G}$ corresponds to two adjacent cells in $\mathscr{T}^{k+1}$.

2. The basic mesh of $\mathscr{G}$ corresponds to two adjacent cells in $\mathscr{T}^{k+1}$. See Fig. 3.3. Here we assume the two adjacent cells of $\mathscr{T}^{k+1}$ are vertically adjacent, edges v_3v_5 and v_5v_4 are of level $k+1$ and edges v_1v_6 and v_1v_2 are of level k. The two adjacent cells which are horizontally adjacent can be discussed similarly:

   (a) If neither of cell 1 and cell 2 are refined, the situation is as Fig. 3.3a.

   (b) If cell 1 is refined, cell 2 is not refined, the situation is as Fig. 3.3b.

   (c) If cell 2 is refined, cell 1 is not refined, the situation is as Fig. 3.3c.

   (d) If cell 1 and cell 2 are both refined, the situation is as Fig. 3.3d.

This completes the proof of the lemma. □

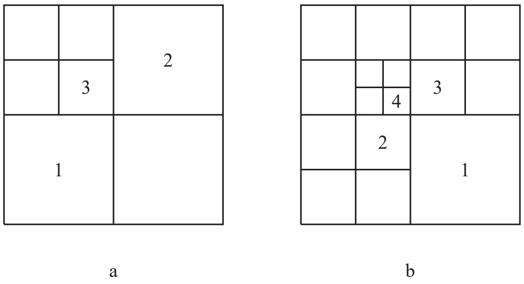


Fig. 3.4. Refinement algorithm for the rule.

Given a hierarchical T-mesh $\mathscr{T}$, to guarantee that all the tensor-product B-splines can span the spline space, the level difference of the adjacent cells of $\mathscr{T}$ cannot be more than one. It should be noticed that the constraint is proper in scientific computing [10]. Therefore, to find a basis only consist of tensor-product B-splines, the rule of refinement is : the level difference of the adjacent cells is at most one. In this paper, we only consider the T-meshes which satisfy this rule.

3.2. The algorithm of the Refinement

To guarantee that the level difference is at most one, we should refine more cells. See Fig. 3.4. In Fig. 3.4a, If we want to refine cell 3, because neither of cell 1 and cell 2 are divided, we should refine cell 1 and cell 2 firstly. In Fig. 3.4b, If we want to refine cell 4, because neither of cell 2 and cell 3 are divided, we should refine cell 2 and cell 3 firstly. If we want to refine cell 2 or cell 3, because cell 1 is not divided, we should refine cell 1 firstly. Therefore, to refine cell 4, we should refine cell 1 firstly, then refine cell 2 and cell 3. In the following, we give the algorithm. Algorithm 3.1 is to find a sequence of cells that should be refined to ensure the rule. We use the recursion of functions. Algorithm 3.1 is always guaranteed to terminate, because the worst case is that the final mesh becomes a tensor-product mesh. Algorithm 3.2 is the process of the refinement of a hierarchical T-mesh.

Algorithm 3.1. Finding a sequence of cells for refining.

```
Require: A sequence of cells $C_0$ that will be refined
Ensure: A sequence of cells $C$ that should be refined to ensure the rule
  function RefineListAt ($C_0$)
    Push all elements in $C_0$ to $C$
    for each cell $c$ in $C_0$ do
      if $c$ does not satisfies the rule
        find out a sequence of cells $C_1$ which should be refined
        RefineListAt($C_1$)
      end if
    end for
    erase the repetitive elements in $C$
    sort elements in $C$ according to levels from low to high
    return ($C$)
  end fuction
```

```
Algorithm 3.2. Refinement of a hierarchical T-mesh.
 Require: A hierarchical T-mesh $\mathscr{T}$ and a tolerance $\varepsilon$
 Ensure: A refined hierarchical T-mesh $\mathscr{T}'$
   for each cell $c$ of $\mathscr{T}$ do
     compute the error $e[c]$
     if $e[c] > \varepsilon$ then
       push $c$ to a vector $C_0$
     end if
   end for
   $C \leftarrow$ RefineListAt $(C_0)$
   for each cell $c \in C$ do
     if $c$ has not been refined then
       refine cell $c$
     end if
   end for
   return $\mathscr{T}'$
```

4. Construction of the Basis

In this section, we provide a detailed construction process for the biquadratic hierarchical basis. We use $\mathrm{lev}(\mathscr{T})$ to represent the top level of a T-mesh $\mathscr{T}$. Suppose $\mathrm{lev}(\mathscr{T}) = N$. In the refinement process, we use $\mathscr{T}^k$ to represent the T-mesh of level k. We refine some cells of level k, the new basis functions are called basis functions of level $k+1$.

Two steps are needed for the construction of the basis functions: adding the basis functions of level $k+1$ and deleting some of the basis functions of level k.

4.1. Adding the basis functions of level $k+1$

In Fig. 4.1, if we divide the cell bounded by the dashed lines, cells 1-4 (if they exist) will be influenced.

First, we analyze the CVR graph $\mathscr{G}^{'}$ of the submesh $\mathscr{T}^{'}$ bounded by the dashed lines. If there are new basic meshes in $\mathscr{G}^{'}$, we find their corresponding 4×4 tensor-product submeshes in $\mathscr{T}^{'}$, and add the corresponding tensor-product B-splines to the original basis. Then, we just need to repeat the process for cells 1-4.

We use $lev(f)$ to represent the level of the basis function f and $\mathscr{F}_i^0$ to represent the set of the basis functions of level i. We denote $\mathscr{F}^0 = \bigcup_{i=0}^{N} \mathscr{F}_i^0$, where $\mathscr{F}_0^0$ is the tensor-product B-spline basis defined on $\mathscr{T}^0$.

Remark 4.1. From Fig. 3.2 and Fig. 3.3, we know that the support of a basis function of level k can not contain cells of level $i, i < k-1$.

4.2. Deleting some of the basis functions of level k

We delete the basis functions of level k if the domain of its basic mesh is covered by the domains of the basic meshes of some basis functions of level $k+1$. Fig. 4.2 is a 4×4 tensor-

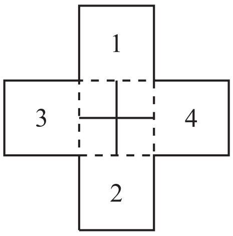


Fig. 4.1. The construction of the basis function of level $k+1$.

product mesh $\mathscr{T}^{'}$ whose corresponding basic mesh is bounded by dashed lines. We denote the basis function of level k defined on Fig. 4.2 as f.

There are three situations that $\phi(f)$ is covered by the domains of the basic meshes of some basis functions of level $k+1$. See Fig. 4.3-4.5, which are three types of CVR graph of the submesh bounded by dashed lines in Fig. 4.2 after that cell 5 is refined.

1. If cells 1-4 are all refined, the CVR graph is as Fig. 4.3, then the domain of the basic mesh of level k is covered by four basic meshes of level $k+1$.

2. If all of the cells 1-4 are refined, the CVR graph is as Fig. 4.4, then the domain of the basic mesh of level k is covered by three basic meshes of level $k+1$.

3. If cell 3 and cell 4, or cell 1 and cell 2 are refined, the CVR graph is as Fig. 4.5, then the domain of the basic mesh of level k is covered by two basic meshes of level $k+1$.

After deleting, we use $\mathscr{F}_i$ to represent the remaining basis functions of level i. The final basis we obtain is denoted by $\mathscr{F} = \bigcup_{i=0}^{N} \mathscr{F}_i$.

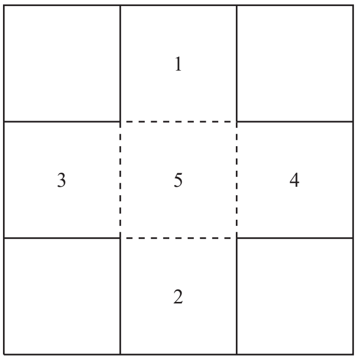


Fig. 4.2. A 4 × 4 tensor-product $\mathscr{T}^{'}$.

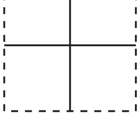


Fig. 4.3. The first type of CVR graph.

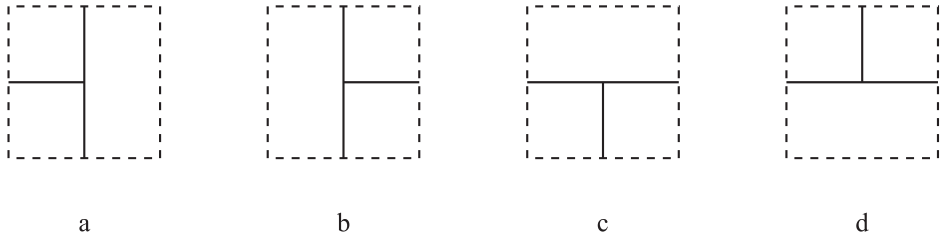


Fig. 4.4. The second type of CVR graph.

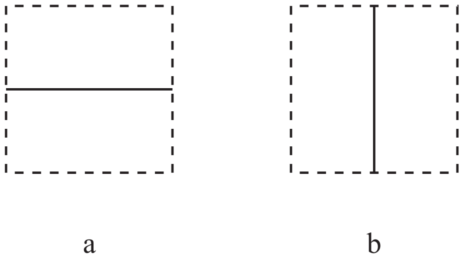


Fig. 4.5. The third type of CVR graph.

Theorem 4.1. *Given a hierarchical T-mesh $\mathscr{T}$, we have*

$$\dim \mathbf{S}(2,2,1,1,\mathscr{T}) = \#\mathscr{F}. \tag{4.1}$$

where $\#\mathscr{F}$ is the number of the elements in $\mathscr{F}$.

Proof. Suppose $\mathscr{G}$ is the CVR graph of $\mathscr{T}$. From Corollary 2.1, we know that $\dim \mathbf{S}(2,2,1,1,\mathscr{T}) = N_{\mathscr{G}}$, where $N_{\mathscr{G}}$ is the number of cells in the CVR graph $\mathscr{G}$ of $\mathscr{T}$. With the restriction that the level difference of the adjacent cells is at most one, the number of cells in $\mathscr{G}$ equals the number of the basic meshes. Because each tensor-product B-spline basis function corresponds to a basic mesh, the theorem is proved. □

4.3. An example

We give an example to illustrate the process of the basis construction. We use $N[x_1,x_2,x_3,x_4](x)$ to denote the quadratic B-spline basis function associated with the knot vector $[x_1,x_2,x_3,x_4]$ in x-direction, $N[y_1,y_2,y_3,y_4](y)$ to denote the quadratic B-spline basis function associated with the knot vector $[y_1,y_2,y_3, y_4]$ in y-direction.

In Fig. 4.6, we use M_i to denote the domain occupied by the cells of the basic mesh, f_i^k to denote the basis function of level k whose corresponding basic domain is labeled with M_i. $\mathscr{T}^0$ is the initial mesh and $\mathscr{G}^0$ is its CVR graph. There are 9 basic meshes in $\mathscr{G}^0$. Therefore, $\#\mathscr{F}_0^0 = 9$. The basis functions are:

$$f_1^0 = N[0,1,2,3](x)N[2,3,4,5](y),\ f_2^0 = N[0,1,2,3](x)N[1,2,3,4](y),\ f_3^0 = N[0,1,2,3](x)N[0,1,2,3](y),$$
$$f_4^0 = N[1,2,3,4](x)N[2,3,4,5](y),\ f_5^0 = N[1,2,3,4](x)N[1,2,3,4](y),\ f_6^0 = N[1,2,3,4](x)N[0,1,2,3](y),$$
$$f_7^0 = N[2,3,4,5](x)N[2,3,4,5](y),\ f_8^0 = N[2,3,4,5](x)N[1,2,3,4](y),\ f_9^0 = N[2,3,4,5](x)N[0,1,2,3](y).$$

Now we refine some cells of level 0 and obtain a new mesh $\mathscr{T}^1$ whose CVR graph is $\mathscr{C}^1$. Six new basic meshes appear in $\mathscr{C}^1$. Therefore, $\#\mathscr{F}_1^0 = 6$. The six basis functions are:

$$f_{10}^1 = N[0,0.5,1,1.5](x)N[1.5,2,2.5,3](y), \quad f_{11}^1 = N[0,0.5,1,1.5](x)N[1,1.5,2,2.5](y),$$
$$f_{12}^1 = N[0.5,1,1.5,2](x)N[1.5,2,2.5,3](y), \quad f_{13}^1 = N[0.5,1,1.5,2](x)N[1,1.5,2,2.5](y),$$
$$f_{14}^1 = N[0,1,2,3](x)N[2,2.5,3,4](y), \qquad f_{15}^1 = N[1,1.5,2,3](x)N[1,2,2.5,3](y).$$

As the domain of basic mesh M_2 is covered by the basic meshes M_{12}, M_{14} and M_{15}, f_2^0 will be deleted. Finally, we have $\mathscr{F}_0 = \{f_1^0, f_3^0, f_4^0, f_5^0, f_6^0, f_7^0, f_8^0, f_9^0\}$, $\mathscr{F}_1 = \{f_{10}^1, f_{11}^1, f_{12}^1, f_{13}^1, f_{14}^1, f_{15}^1\}$. The basis over the T-mesh $\mathscr{T}$ is $\mathscr{F} = \{f_1^0, f_3^0, f_4^0, f_5^0, f_6^0, f_7^0, f_8^0, f_9^0, f_{10}^1, f_{11}^1, f_{12}^1, f_{13}^1, f_{14}^1, f_{15}^1\}$.

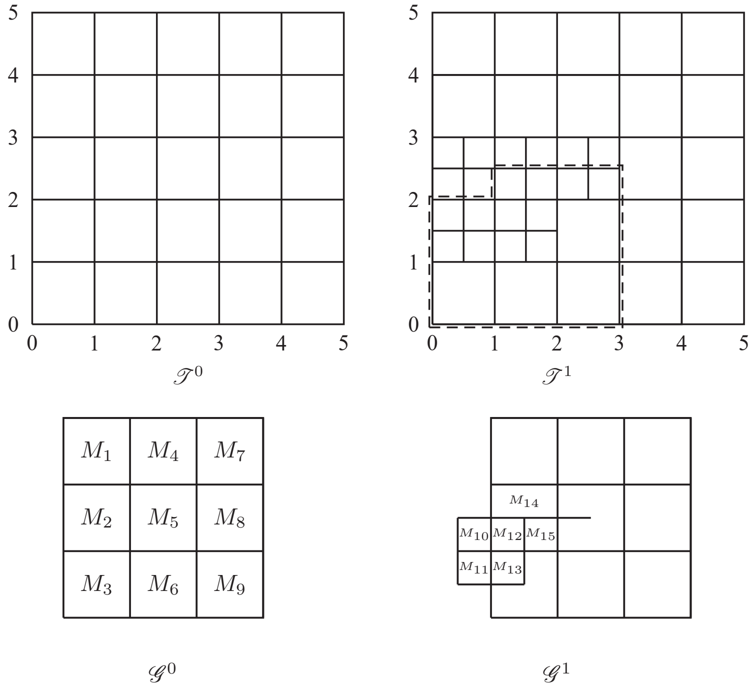


Fig. 4.6. An example for the basis construction.

5. Properties

In this part, we discuss some properties of the basis constructed in Section 4. The support set of any function f is defined as $supp(f) = \{(x,y) \in \Omega : f(x,y) \neq 0\}$, where Ω is the domain occupied by the cells of the T-mesh $\mathscr{T}$.

5.1. Linear independence

We give three lemmas about the basis firstly.

Lemma 5.1. *For two basis functions f_1 and f_2, suppose $\mathscr{T}_1$ and $\mathscr{T}_2$ are the tensor-product meshes which f_1 and f_2 are defined on, respectively. In Fig. 5.1, $\mathscr{T}_1$ and $\mathscr{T}_2$ are bounded by solid lines and dashed lines, respectively. The points A_1, A_2, A_3 and A_4 are the four corner points of $\mathscr{T}_1$. If $supp(f_1) \cap supp(f_2) \neq \emptyset$, $supp(f_1) \nsubseteq supp(f_2)$ and $supp(f_2) \nsubseteq supp(f_1)$, then A_i can not be the corner point of $\mathscr{T}_2$, where $i = 1,2,3,4$.*

Proof. This lemma can be proved by checking all types of the basis functions in Fig. 3.2 and Fig. 3.3. If $lev(f_2) > lev(f_1)$, suppose A_1 is also the corner point of $\mathscr{T}_2$. Then, $supp(f_2) \subseteq supp(f_1)$, a contradiction. Conversely, if $lev(f_2) < lev(f_1)$, then $supp(f_1) \subseteq supp(f_2)$, also a contradiction.

If $lev(f_1) = lev(f_2)$, this conclusion is also correct. We omit the proof here. □

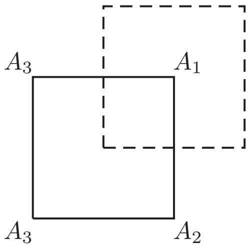


Fig. 5.1. Figure for lemma 5.1.

Lemma 5.2. *Suppose* $\mathcal{F}_k^0 = \{f_1, f_2, \ldots, f_m\}$. *For any basis function* f *of level* $k+1$, *there exists an* $i, i \in \{1, 2, \ldots, m\}$, *such that* $supp(f) \subseteq supp(f_i)$.

Proof. We should check this conclusion for all types of the basis functions in Fig. 3.2 and Fig. 3.3. Here, we only check this lemma for the basis functions in Fig. 3.2.a. The others can be verified similarly.

For Fig. 3.2.a, the set $supp(f)$ consist of four cells of level k which have been illustrated in Fig. 5.2, here we suppose the edges labeled with dashed lines are not boundary edges of the T-mesh $\mathscr{T}$. Each cell of level k must be from a cell of level $k-1$. Therefore, cells 1-4 can be from four different cells of level $k-1$, or two of them are from a same cell of level $k-1$, or cells 1-4 are from a same cell of level $k-1$. The three situations are as follow:

1. If cells 1-4 are from four different cells of level $k-1$, the situation is as Fig. 5.3, where the four cells of level $k-1$ are bounded with dashed lines. The four tensor-product B-splines defined on the mesh in Fig. 5.3 are all basis functions of level k that we are looking for.

2. If two of cells 1-4 are from the same cell of level $k-1$, the situation is as Fig. 5.4, where cell 5 may be divided or not.

   (a) Cell 1 and cell 3 are from a same cell of level $k-1$, and cell 2 and cell 4 are from a same cell of level $k-1$, the situation is as Fig. 5.4a. The tensor-product B-splines defined on the mesh in Fig. 5.4a whose support contains cells 1-4 are the basis functions of level k we are looking for.

   (b) Cell 1 and cell 2 are from a same cell of level $k-1$, and cell 3 and cell 4 are from a same cell of level $k-1$, the situation is as Fig. 5.4b. The tensor-product B-splines defined on the mesh in Fig. 5.4b whose support contains cells 1-4 are the basis functions of level k we are looking for.

3. Cells 1-4 are from a same cell of level $k-1$, the situation is as Fig. 5.5, where cell 5 may be divided or not. The tensor-product B-splines defined on the mesh in Fig. 5.5 whose support contains cells 1-4 are the basis functions of level k we are looking for.

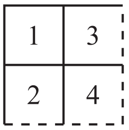


Fig. 5.2. Four cells of level k.

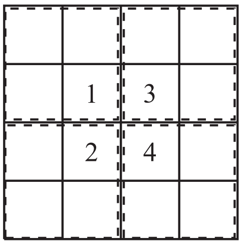


Fig. 5.3. The cells 1-4 are from four different cells of level $k-1$.

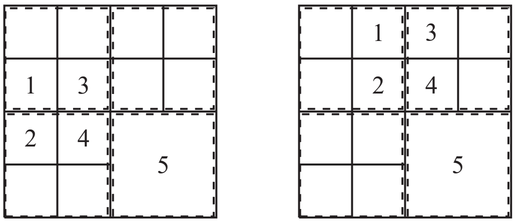


Fig. 5.4. Two of the cells 1-4 are from a same cell of level $k-1$.

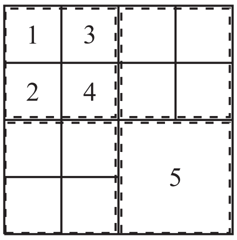


Fig. 5.5. The cells 1-4 are from a same cell of level $k-1$.

For Fig. 3.2b, we can prove similarly. □

For a basis function f of level $k+1$, suppose f_1 is a basis function in $\mathscr{F}_i^0$, $i<k+1$, whose support set contains the support set of f, then we call f_1 a mother function of f. We use $mot_i(f)$ to denote one of the mother functions of level i without distinction.

Lemma 5.3. *Suppose $f \in \mathcal{F}_k$, f is defined on a 4×4 tensor-product mesh whose center cell is denoted as cell C, $f_1, f_2, \ldots, f_m$ are all of the basis functions of level $k+1$ whose support set is a subset of $supp(f)$. If $supp(f)/\bigcup_{i=1}^m supp(f_i) = \emptyset$, then cell C cannot be divided and $f, f_1, f_2, \cdots, f_m$ are linearly independent over cell C.*

Proof. We prove this lemma by checking all types of basis functions in Fig. 3.2 and Fig. 3.3.

For Fig. 3.2.a and Fig. 3.3, the support set of each basis function contains cells of level k which can not be occupied by the support set of basis functions of level $k+1$.

In the following, we consider the basis functions in Fig. 3.2.b. Suppose

$$f = N[x_0, x_2, x_4, x_6](x)N[y_0, y_2, y_4, y_6]\ (y)$$

is defined on the mesh in Fig. 5.6, the basic mesh corresponding to f is bounded by the dashed lines. We prove this lemma by considering the following two cases.

1. The cell bounded by the dashed lines in Fig. 5.6 is divided.
   From Section 4.2, we know $\phi(f)/\bigcup_{i=1}^{m}\phi(f_i) \neq \emptyset$. Therefore, cell 2 and cell 4, or cell 2 and cell 6, or cell 4 and cell 8, or cell 6 and cell 8 cannot be refined (otherwise, f will be deleted according to Section 4.2). We suppose cell 6 and cell 8 are not refined. Then we have a fact: the domain of cell 9 cannot be occupied by $\bigcup_{i=1}^{m} supp(f_i)$ (if f_i exists). We use Fig. 5.7 as an example to verify the fact, other situations can be verified similarly. In Fig. 5.7, there is a new basis of level $k+1$, which is denoted as f_1. The domain of $supp(f)$ is bounded by the dotted lines. Therefore, the fact is true.

2. The cell bounded by the dashed lines in Fig. 5.6 is not divided, that is the center cell C of $supp(f)$ is not divided.
   There is only one case such that $supp(f)/\bigcup_{i=1}^{m} f_i = \emptyset$, see Fig. 5.8. In Fig. 5.8, there are eight basis functions of level $k+1$. We denote the domains occupied by their basic meshes as $M_i, i = 1, 2, \cdots, 8$, and the corresponding basis function as $f_i, i = 1, 2, \cdots, 8$.

$$\begin{aligned}
f_1 &= N[x_0, x_1, x_2, x_4](x)[y_2, y_4, y_5, y_6](y), & f_2 &= N[x_1, x_2, x_4, x_5](x)[y_2, y_4, y_5, y_6](y),\\
f_3 &= N[x_2, x_4, x_5, x_6](x)[y_2, y_4, y_5, y_6](y), & f_4 &= N[x_0, x_1, x_2, x_4](x)[y_1, y_2, y_4, y_5](y),\\
f_5 &= N[x_2, x_4, x_5, x_6](x)[y_1, y_2, y_4, y_5](y), & f_6 &= N[x_0, x_1, x_2, x_4](x)[y_0, y_1, y_2, y_4](y),\\
f_7 &= N[x_1, x_2, x_4, x_5](x)[y_0, y_1, y_2, y_4](y), & f_8 &= N[x_2, x_4, x_5, x_6](x)[y_0, y_1, y_2, y_4](y).
\end{aligned}$$

We denote $\bar{f} = N[x_1, x_2, x_4, x_5](x)[y_1, y_2, y_4, y_5](y)$,

$$\alpha_0 = \frac{x_1 - x_0}{x_4 - x_0}, \quad \alpha_1 = \frac{x_6 - x_5}{x_6 - x_2}, \quad \beta_0 = \frac{y_1 - y_0}{y_4 - y_0}, \quad \beta_1 = \frac{y_6 - y_5}{y_6 - y_2}.$$

Then we have

$$f = \alpha_0\beta_1 f_1 + \beta_1 f_2 + \alpha_1\beta_1 f_3 + \alpha_0 f_4 + \alpha_1 f_5 + \alpha_0\beta_0 f_6 + \beta_0 f_7 + \alpha_1\beta_0 f_8 + \bar{f}. \tag{5.1}$$

Because $f_1, f_2, \cdots, f_8$ and $\bar{f}$ are the tensor-product B-splines defined on the tensor-product mesh with knot vectors $[x_0, x_1, x_2, x_4, x_5, x_6] \times [y_0, y_1, y_2, y_4, y_5, y_6]$, $f_1, f_2, \cdots, f_8$ and $\bar{f}$ are linearly independent on cell C.

From (5.1), we have

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \\ f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \alpha_0\beta_1 & \beta_1 & \alpha_1\beta_1 & \alpha_0 & \alpha_1 & \alpha_0\beta_0 & \beta_0 & \alpha_1\beta_0 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \\ \bar{f} \end{pmatrix}. \tag{5.2}$$

Because the coefficient matrix in (5.2) is a lower triangular matrix with diagonal elements all 1, it is full rank. Therefore, $f, f_1, f_2, \cdots, f_8$ are linearly independent on cell C, which proves the lemma. □

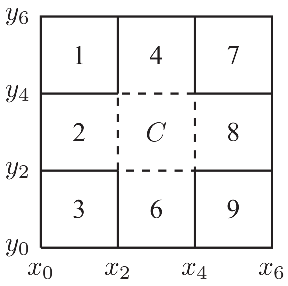


Fig. 5.6. The 4×4 tensor-product mesh which corresponds to f.

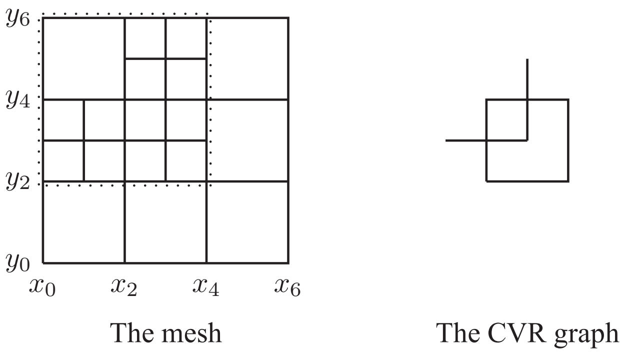


Fig. 5.7. The mesh and the CVR graph after that cell 2, cell 4 and cell 5 in Fig. 5.6 are refined.

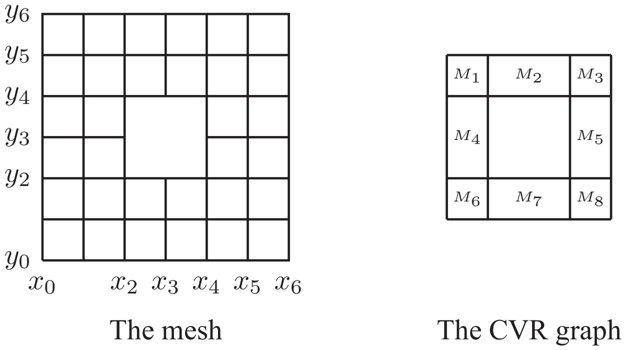


Fig. 5.8. The mesh and the CVR graph after that cells 1-4 and cells 6-9 in Fig. 5.6 are refined.

Theorem 5.1. *The functions in $\mathcal{F}$ are linearly independent.*

Proof. Suppose $\mathcal{F}_0^0 = \{f_1, f_2, \ldots, f_m, f_{m+1}, \ldots, f_n\}, \mathcal{F}_0 = \{f_1, f_2, \ldots, f_m\}$, and $lev(\mathscr{T}) = N$.

For any function f at level $k, k \geq 1$, from lemma 5.2, we know that $supp(f)$ is a subset of the support sets of some functions at level 0.

Suppose

$$\sum_{f\in\mathcal{F}} c(f)f = 0, \qquad c(f) \in \mathbb{R}. \tag{5.3}$$

To prove the theorem, we just need to prove $c(f) = 0$ for all the $f \in \mathcal{F}$.

Let

$$g_i = \begin{cases} c(f_i)f_i + \sum_{k=1}^{N} \sum_{f\in\mathcal{F}_k, mot_0(f)=f_i} c(f)f, & 1 \leqslant i \leqslant m, \\ \sum_{k=1}^{N} \sum_{f\in\mathcal{F}_k, mot_0(f)=f_i} c(f)f, & m+1 \leqslant i \leqslant n. \end{cases}$$

(5.3) becomes

$$\sum_{i=1}^{n} g_i = 0. \tag{5.4}$$

From the definition, we know $supp(g_i) = supp(f_i)$. Consider the bottom-left cell α of the mesh $\mathscr{T}$. Here “bottom-left” means that there is no cell whose bottom edge is below α, and among the cells whose bottom edges are in the same horizontal line as α, α is the leftmost cell. From lemma 5.1, there is only one function $f_j \in \mathcal{F}_0^0$ satisfying $\alpha \in supp(f_j)$, namely $\alpha \in supp(g_j)$. Therefore, g_j is the only non-zero function on α. By (5.4), we obtain $g_j = 0$.

Excluding g_j from (5.4), we consider the bottom-left cell of the region given by $\mathscr{T}\backslash\alpha$. Also from lemma 5.1, there is only one function of $g_k, k \neq j$ that can be non-zero on this cell, and it is zero. Continuing this argument, we obtain $g_l = 0$ for all $1 \leqslant l \leqslant n$.

For each $g_i = 1, 2, \cdots, m$, we consider f_i in the following two cases.

1. $supp(f_i)/\bigcup_{f\in\mathcal{F}_1, mot_0(f)=f_i} supp(f) \neq \emptyset$.
   From the definition, we know that $\bigcup_{f\in\mathcal{F}_{k+1}, mot_0(f)=f_i} supp(f) \subseteq \bigcup_{f\in\mathcal{F}_k, mot_0(f)=f_i} supp(f)$, then we have
   $$supp(f_i)/\bigcup_{k=1}^{N} \bigcup_{f\in\mathcal{F}_k, mot_0(f)=f_i} supp(f) \neq \emptyset.$$
   Because $g_i = 0$, we obtain $c(f_i) = 0$.

2. $supp(f_i)/\bigcup_{f\in\mathcal{F}_1, mot_0(f)=f_i} supp(f) = \emptyset$.
   Suppose f_i is defined on a 4×4 tensor-product mesh with center cell C. With Lemma 5.3, we know cell C is a cell of level 0 that is not divided. With Remark 4.1, we know that the support sets of the basis functions of level $k, k > 1$ can not contain cell C. Therefore, we have
   $$\sum_{k=2}^{N} \sum_{f\in\mathcal{F}_k, mot_0(f)=f_i} c(f)f = 0$$
   on cell C. Because $g_i = 0$, we obtain
   $$c(f_i)f_i + \sum_{f\in\mathcal{F}_1, mot_0(f)=f_i} c(f)f = 0$$

on cell C. With Lemma 5.3, we obtain $c(f_i) = 0$.

Now each g_i becomes:

$$g_i = \sum_{k=1}^{N} \sum_{f \in \mathcal{F}_k, mot_0(f) = f_i} c(f)f.$$

We can repeat the discussion on every g_i, and obtain $c(f) = 0$ for $f \in \mathcal{F}_1$.

Continuing this process, we prove this theorem. □

5.2. Other properties

The basis functions are all tensor-product B-splines, so they inherit some properties, such as nonnegativity and local support. From Theorem 4.1 and the linear independence, we know they are complete. Therefore, $\mathcal{F}$ is a basis of $S(2, 2, 1, 1, \mathscr{T})$.

6. A New Basis with Unit Partition

First, we give the definition of unit partition.

Definition 6.1. *Given a basis $\mathscr{F} = \{g_1, g_2, \cdots, g_n\}$ with homogeneous boundary condition defined on $\mathscr{T}$ which is an extension of a T-mesh $\mathscr{T}'$. If $\sum_{i=1}^{n} g_i = 1$ on Ω, where Ω is the domain occupied by the cells in $\mathscr{T}'$, then we call the basis $\mathscr{F}$ has the property of unit partition.*

Unit partition is an important property in geometry modeling [11]. However, the basis $\mathscr{F}$ constructed in Section 4 does not have this property. In this part, we give a new basis satisfying this property, which is just an elementary transformation of the basis $\mathscr{F}$. The new basis functions are represented as the Bernstein basis.

For the initial tensor-product T-mesh $\mathscr{T}^0$, it is obvious that $\mathscr{F}_0^0$ has the property of unit partition. To ensure the property of unit partition, we need to keep the sum of the basis functions same from level k to level $k+1$. Now we modify $\mathscr{F}_k$ and $\mathscr{F}_{k+1}^0$ as follow:

1. For a function $f \in \mathscr{F}_k$, we use

$$\bar{f} = f - \sum_{f_1 \in \mathscr{F}_{k+1}^0, mot_k(f_1) = f} \alpha(f, f_1) f_1 \tag{6.1}$$

instead of f as the new function, where the coefficient $\alpha(f, f_1)$ is computed using Algorithm 5.1. We denote the set $\{\bar{f}, f \in \mathscr{F}_k\}$ as $\bar{\mathscr{F}}_k$.

2. For a function $f_1 \in \mathscr{F}_{k+1}^0$, we use

$$\bar{f}_1 = \sum_{f \in \mathscr{F}_k^0, mot(f_1) = f} \alpha(f, f_1) f_1 \tag{6.2}$$

instead of f_1 as the new function, where the coefficient $\alpha(f, f_1)$ computed using Algorithm 5.1. We denote the set $\{\bar{f}_1, f_1 \in \mathscr{F}_{k+1}^0\}$ as $\bar{\mathscr{F}}_{k+1}^0$.

The final modified basis is denoted by $\bar{\mathscr{F}} = (\bigcup_{k=0}^{N-1} \bar{\mathscr{F}}_k) \bigcup \bar{\mathscr{F}}_N^0$, where $N = lev(\mathscr{T})$.

Remark 6.1. For a function f in $\bar{\mathscr{F}}_k$, $supp(f) \neq \emptyset$. This follows from Lemma 5.3.

We call the coefficient $\alpha(f, f_1)$ the basic coefficient. In the following section, we will give the algorithm to compute the basic coefficients.

6.1. Algorithm for the computation of the basic coefficients

First, we give a lemma without proof.

Lemma 6.1. *Given a polynomial $f(x, y)$ which is in the form of the Bernstein basis. If the Bezier ordinates in each cell of supp(f) are all nonegative, then the polynomial $f(x, y) \geq 0$.*

Suppose $f_1 \in \mathscr{F}^0_{k+1}$ whose support set is a subset of *supp*(f). In Algorithm 5.1, we find a coefficient α such that all the Bezier ordinates of $f - \alpha f_1$ are nonegative. With Lemma 6.1, we have $f - \alpha f_1 \geq 0$.

```
Algorithm 6.1. Computation of the basic coefficient.
Require: A function $f_1 \in \mathscr{F}^0_{k+1}$ whose support set is a subset of supp($f$), the Bezier ordinates in each cell of supp($f_1$) and supp($f$) are all nonegative
Ensure: A coefficient $\alpha$ satisfies that $f - \alpha f_1 \geq 0$
  function COMPUTECOEFF $f, f_1$
      $M \leftarrow$ the biggest number in the computer
      $\alpha \leftarrow M$
      for each cell $c$ in both supp($f$) and supp($f_1$) do
          for each Bezier ordinate $b \neq 0$ of $f$ and $b_1$ of $f_1$ in cell $c$ do
              if $b_1 = 0$ then
                  $\beta \leftarrow M$
               else
                  $\beta \leftarrow b/b_1$
              end if
              if $\alpha > \beta$ then
                  $\alpha \leftarrow \beta$
              end if
          end for
          if the Bezier ordinates in $c$ are all zeros
              delete $c$ from supp($f$)
          end if
      end for
  end fuction
```

6.2. Algorithm for the construction of the new basis

Algorithm 6.2 is simple process of the construction of the new basis. The new basis is computed level by level. The basis functions in $\mathscr{F}^0_k/\mathscr{F}_k$ will be modified to zeros, this follows Section 4.2. Therefore, $\tilde{\mathscr{F}}_k = \bar{\mathscr{F}}_k$, $\tilde{\mathscr{F}}^0_{k+1} = \bar{\mathscr{F}}^0_{k+1}$, and $\tilde{\mathscr{F}} = \bar{\mathscr{F}}$ in Algorithm 6.2.

```
Algorithm 6.2. A new basis.
Require: The basis $\mathscr{F}$ constructed in Section 4
Ensure: A new basis $\tilde{\mathscr{F}}$
    $N \leftarrow lev(\mathscr{T})$
    $\tilde{\mathscr{F}}_0^0 \leftarrow \mathscr{F}_0^0$
    for level $k$ from 1 to $N$ do
        for each basis function $f_1 \in \mathscr{F}_{k+1}^0$
            $g \leftarrow 0$
            for each function $f \in \tilde{\mathscr{F}}_k^0$ and $mot(f_1) = f$
                $\alpha \leftarrow$ COMPUTECOEFF $f, f_1$
                $f \leftarrow f - \alpha f_1$
                $g \leftarrow g + \alpha f_1$
                $f_1 \leftarrow g$
            end for
            Push $f_1$ to $\tilde{\mathscr{F}}_{k+1}^0$
        end for
        $\tilde{\mathscr{F}}_k = \{f, f \in \tilde{\mathscr{F}}_k^0$ and $supp(f) \neq \emptyset\ \}$
    end for
    $\tilde{\mathscr{F}} = \bigcup_{i=1}^{N-1} \tilde{\mathscr{F}}_k \bigcup \tilde{\mathscr{F}}_N^0$
    return $\tilde{\mathscr{F}}$
```

6.3. Properties of the new basis

In this section, we use three theorems to give out the properties of unity partition, nonegativity and linear independence of the functions in $\bar{\mathscr{F}}$.

Theorem 6.1. *The functions in $\bar{\mathscr{F}}$ have the property of unit partition.*

Proof. We prove this theorem level by level. For the initial T-mesh $\mathscr{T}_0$ which is a tensor-product mesh, $\bar{\mathscr{F}}_0^0 = \mathscr{F}_0^0$ is a set of tensor-product B-splines that has the property of unit partition. Suppose the new basis $\bar{\mathscr{F}}'$ defined over $\mathscr{T}_k$ also has the property of unit partition, $\bar{\mathscr{F}}$ is the new basis defined over $\mathscr{T}_{k+1}$. With (6.1) and (6.2), we have

$$\begin{aligned}
\sum_{\bar{f}\in\bar{\mathscr{F}}} \bar{f} &= \sum_{i=1}^{k-1} \sum_{\bar{f}\in\bar{\mathscr{F}}_i} \bar{f} + \sum_{\bar{f}\in\bar{\mathscr{F}}_k} \bar{f} + \sum_{\bar{f}_1\in\bar{\mathscr{F}}_{k+1}^0} \bar{f}_1 \\
&= \sum_{i=1}^{k-1} \sum_{\bar{f}\in\bar{\mathscr{F}}_i} \bar{f} + \sum_{f\in\bar{\mathscr{F}}_k^0} \left(f - \sum_{f_1\in\mathscr{F}_{k+1}^0, mot_k(f_1)=f} \alpha(f_1, f) f_1 \right) + \sum_{f_1\in\mathscr{F}_{k+1}^0} \left(\sum_{f\in\bar{\mathscr{F}}_k^0, mot_k(f_1)=f} \alpha(f_1, f) \right) f_1 \\
&= \sum_{i=1}^{k-1} \sum_{\bar{f}\in\bar{\mathscr{F}}_i} \bar{f} + \sum_{f\in\bar{\mathscr{F}}_k^0} \left(f - \sum_{f_1\in\mathscr{F}_{k+1}^0, mot_k(f_1)=f} \alpha(f_1, f) f_1 \right) + \sum_{f\in\bar{\mathscr{F}}_k^0} \left(\sum_{f_1\in\mathscr{F}_{k+1}^0, mot_k(f_1)=f} \alpha(f_1, f) f_1 \right) \\
&= \sum_{i=1}^{k-1} \sum_{\bar{f}\in\bar{\mathscr{F}}_i} \bar{f} + \sum_{f\in\bar{\mathscr{F}}_k^0} f = \sum_{f\in\bar{\mathscr{F}}'} f = 1. \qquad (6.3)
\end{aligned}$$

Theorem 6.2. *The functions in $\bar{\mathscr{F}}$ are all nonegativity.*

Proof. We prove this theorem by checking the function $\bar{f}$ in (6.1) and the function $\bar{f}_1$ in (6.2).

1. Prove the function $\bar{f} \geq 0$ in (6.1).
In Algorithm 5.1, α is the biggest scale such that $b - \alpha b_1 \geq 0$ for each Bezier ordinate b of f and b_1 of f_1 in the common cells of *supp*(f) and *supp*(f_1). With Lemma 6.1, we know that $f - \alpha f_1 \geq 0$. Therefore, the function $\bar{f}$ in (6.1) is nonegative.

2. Prove the function $\bar{f}_1 \geq 0$ in (6.2).
Because all the Bezier ordinates of f and f_1 in Algorithm 5.1 are nonegative, we have $\alpha \geq 0$. Therefore, $\bar{f}_1$ in (6.2) is nonegative. □

Theorem 6.3. *The functions in $\bar{\mathscr{F}}$ are linearly independent.*

Proof. We sort the functions in $\bar{\mathscr{F}}$ according to their levels from low level to high level. From (6.1) and (6.2), we know that the transformation from $\mathscr{F}$ to $\bar{\mathscr{F}}$ is an elementary transformation, and the transformation matrix is an upper triangular matrix which satisfies that the diagonal elements are nonzeros. With Theorem 5.1, the functions in $\bar{\mathscr{F}}$ are also linearly independent.□

6.4. Remark

In this part, we give some remarks about the basis $\bar{\mathscr{F}}$.

Remark 1. The basis functions in $\bar{\mathscr{F}}$ may not be tensor-product B-splines, and the supports of the basis functions are connected which can be concave.

In Section 4.3, suppose $\bar{\mathscr{F}}_0 = \{\bar{f}_1^0, \bar{f}_3^0, \bar{f}_4^0, \bar{f}_5^0, \bar{f}_6^0, \bar{f}_7^0, \bar{f}_8^0, \bar{f}_9^0\}$. The support sets of $f_{10}^1, f_{11}^1, f_{12}^1, f_{13}^1$ and f_{15}^1 are subsets of *supp*(f_3^0). With Algorithm 6.2, we have

$$\begin{aligned}\bar{f}_3^0 &= f_3^0 - \frac{1}{16}f_{10}^1 - \frac{9}{48}f_{11}^1 - \frac{3}{16}f_{12}^1 - \frac{9}{16}f_{13}^1 - \frac{3}{16}f_{15}^1 \\ &= \frac{3}{4}N[0,1,2,3](x)N[0,1,1.5,2](y) + \frac{1}{2}N[1,1.5,2,3](x)N[1,1.5,2,2.5](y). \qquad (6.4)\end{aligned}$$

$\bar{f}_3^0$ is no longer a tensor-product B-spline, and the support set of $\bar{f}_3^0$ is bounded by the dashed lines, which is concave and connected. By considering all the types of basis functions in Fig. 3.2 and Fig. 3.3, we can prove the supports of the basis functions in $\bar{\mathscr{F}}$ are connected. We omit the proof here.

Remark 2. The basis functions in $\bar{\mathscr{F}}$ are not locally linearly independent.

In Fig. 6.1, the domain occupied by the cells in each basic mesh is denoted as $1, 2, \cdots, 11$, we use f_i^k to denote the basis function of level k whose corresponding basic domain is labeled with i, then

$$\mathscr{F} = \{f_1^0, f_2^1, f_3^1, \cdots, f_7^1, f_8^2, f_9^2, f_{10}^2, f_{11}^2\}.$$

After running Algorithm 6.2, the new basis functions

$$\bar{\mathscr{F}} = \{\bar{f}_1^0, \bar{f}_2^1, \bar{f}_3^1, \cdots, \bar{f}_7^1, \bar{f}_8^2, \bar{f}_9^2, \bar{f}_{10}^2, \bar{f}_{11}^2\},$$

where all the supports of basis functions in $\bar{\mathscr{F}}$ except $\bar{f}_2^1$ contain the domain bounded by red lines, that is, there are 10 basis functions in $\bar{\mathscr{F}}$ are nonzeros over the domain bounded by red

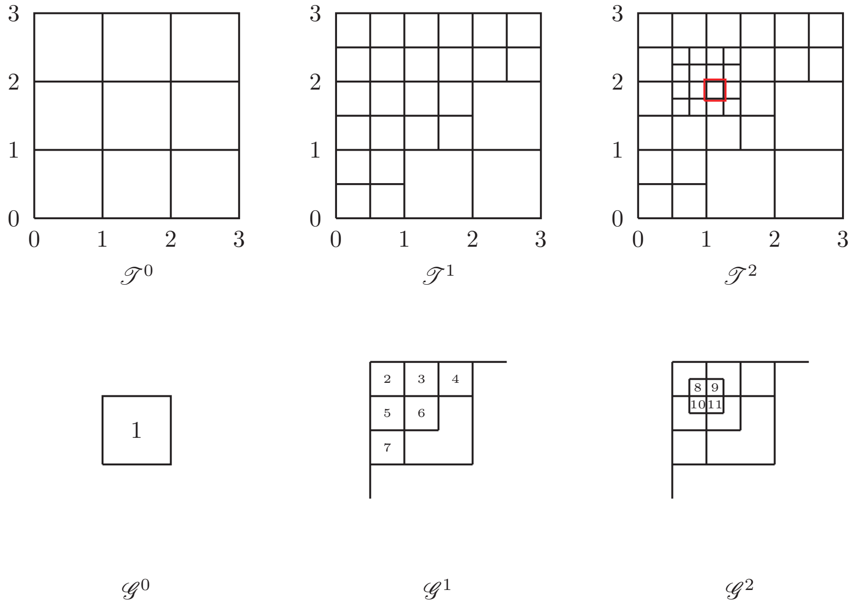


Fig. 6.1. An example which is used to show the basis functions in $\bar{\mathscr{F}}$ are not locally linearly independent.

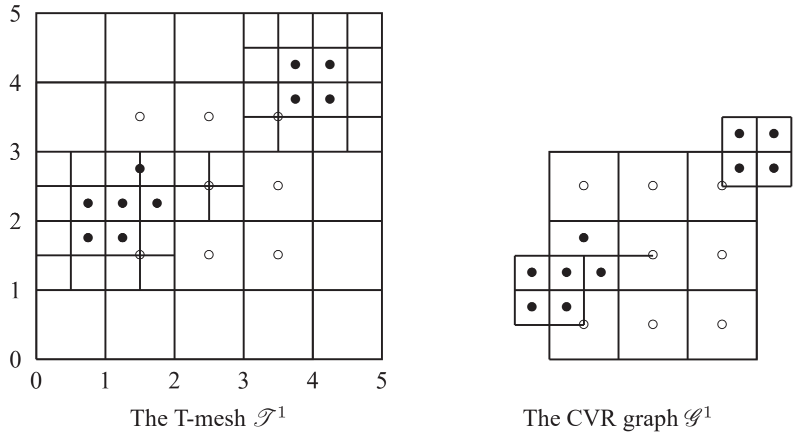


Fig. 6.2. Greville points of the basis $\bar{\mathscr{F}}$.

lines. Because the dimension of biquadratic polynomial space is 9, $\{\bar{f}_1^0, \bar{f}_3^1, \cdots, \bar{f}_7^1, \bar{f}_8^2, \bar{f}_9^2, \bar{f}_{10}^2, \bar{f}_{11}^2\}$ are linearly dependent over the domain bounded by red lines. Therefore, the basis functions in $\bar{\mathscr{F}}$ are not locally linearly independent.

Remark 3. A conjecture about the Greville points of the basis $\bar{\mathscr{F}}$.

With the process of the basis construction, each basis function in $\bar{\mathscr{F}}$ corresponds to a basic mesh which is a 2×2 tensor-product mesh. We conjecture that the center points of the basic meshes can be seen as Greville points. However, we do not know whether the center points combined with control points can give rise to genuine "control points" that have some good properties, such as the convex hull. Fig. 6.2 is an example, where the points label with "○" are the center points of the basic meshes of level 0, and the points label with "●" are the center points of the basic meshes of level 1. The dimension of the splice space defined on Fig. 6.2 is 18.

7. Applications

In this section, we give some numerical applications using the basis constructed in Section 6.

7.1. Isogeometric analysis

In this part, we present two numerical examples in isogeometric analysis. The splines we used to compare are biquadratic tensor-product B-splines defined on uniform tensor-product meshes and splines in [30]. Both examples show that to reach a given accuracy, few degrees of freedom are needed using spline bases in this paper than using the biquadratic tensor-product B-splines defined on uniform tensor-product meshes. Furthermore, compared with spline bases in [30], smaller conditional number of stiffness matrix achieved with spline bases in this paper.

7.1.1. Stationary heat conduction: L-domain

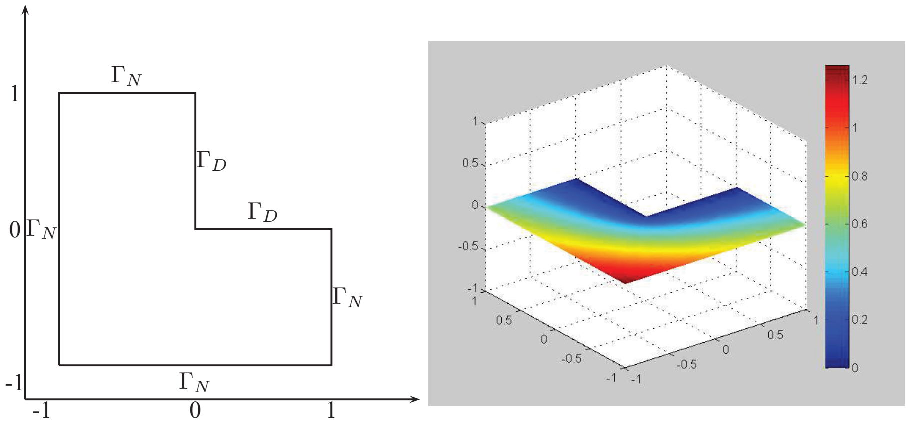
In this part, we try to solve the heat conduction equation with isogeometric analysis. The geometry of the domain is illustrated in Fig. 7.1a, where $\Omega = [-1,1]\times[-1,1]\backslash[0,1]\times[0,1]$. The heat conduction equation is $-\Delta u = 0$ in Ω with a homogeneous condition on Γ_{D}, and Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial f}{\partial \mathbf{n}}$ on Γ_{N}, where $\mathbf{n}$ is outer normal direction, u is an exact solution of the problem and is given here for reference.

$$f = \begin{cases} (x^2+y^2)^{\frac{1}{3}} \sin\left((2\theta-\pi)/3\right), & (x,y)\neq(0,0), \\ 0, & (x,y)=(0,0), \end{cases}$$

where

$$\theta = \begin{cases} \arccos(x/r), & x \le 0 \quad \text{and} \quad y \ge 0, \\ \pi - \arcsin(y/r), & x \le 0 \quad \text{and} \quad y \le 0, \\ 2\pi + \arcsin(y/r), & x \ge 0 \quad \text{and} \quad y \le 0. \end{cases}$$

The exact solution is depicted in Fig. 7.1b. Fig. 7.2 shows the adaptive hierarchical T-meshes after 1, 6, 8, 10 refinements. Fig. 7.3 shows the relative error $(u_{\text{exact}} - u^{\mathrm{h}})/\|u_{\text{exact}}\|_{\mathrm{L}^2}$



a b

Fig. 7.1. Figure a is the setup of the stationary heat conduction, and figure b is the exact solution.

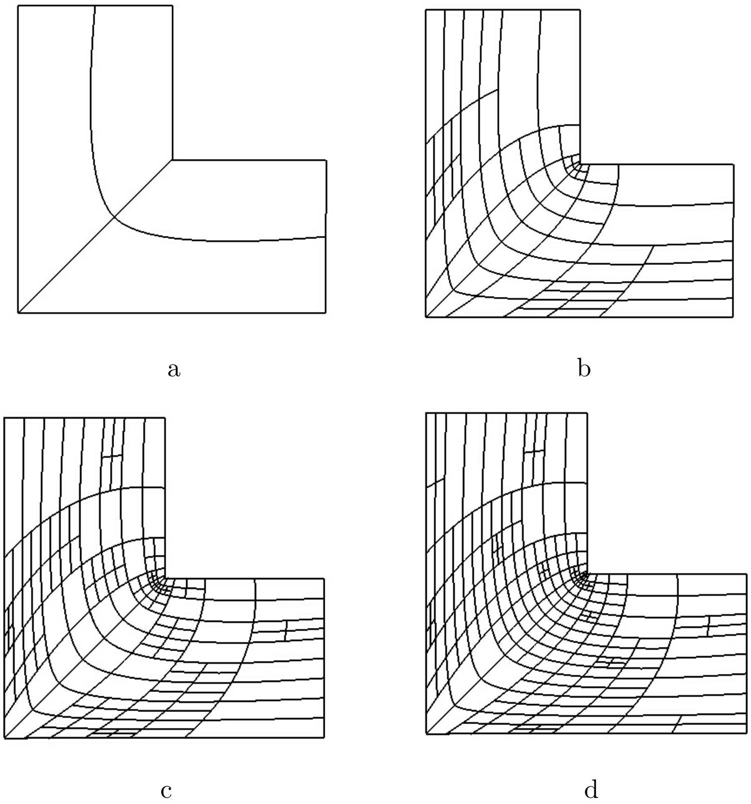


Fig. 7.2. Meshes after 1, 6, 8, 10 refinements.

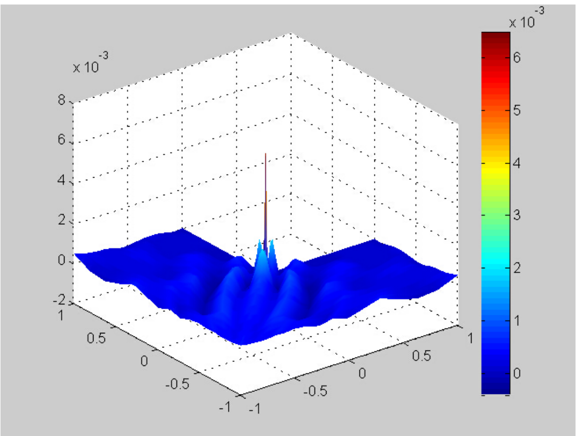


Fig. 7.3. Relative error after 10 refinements.

between the exact solution and the numerical solution associated with the mesh in Fig.7.2c. As the real solution is singular at the origin coordinate, the relative error is larger around the origin coordinate. In Table 7.1, we show the iteration number (n), dimension of spline space, condition number of the stiffness matrix and the error $\|u_{\text{exact}} - u^{\text{h}}\|_{\text{L}^2}/\|u_{\text{exact}}\|_{\text{L}^2}$ for spline bases in this paper, B-splines over uniform tensor-product meshes and spline bases in [30]. Table 7.1 and Fig. 7.4 show that to reach a given tolerance, much fewer degrees of freedom are needed using

Table 7.1: Electron affinity (A), Ionization potential (I), Absolute electroneagativity (χ), Global hardness (η) and Electrophilicity (ω) for monomer and dimers.

Table 1			
n	dim	cond	error
Spline bases in this paper			
1	16	1.74620×10^{1}	4.20730×10^{-2}
2	36	5.18941×10^{1}	9.11038×10^{-3}
6	80	2.17625×10^{2}	1.40437×10^{-3}
8	136	3.04984×10^{2}	6.06022×10^{-4}
10	238	7.34754×10^{2}	1.87078×10^{-4}
B-splines over uniform tensor-product meshes			
1	16	1.74620×10^{1}	4.20730×10^{-2}
2	36	5.18941×10^{1}	9.11038×10^{-3}
3	100	2.28387×10^{2}	1.95270×10^{-3}
4	324	1.00870×10^{3}	4.00000×10^{-4}
5	1156	2.25850×10^{3}	1.58000×10^{-4}
Spline bases in [30]			
1	16	1.74620×10^{1}	4.20730×10^{-2}
2	39	4.767792×10^{2}	8.7421×10^{-3}
7	81	5.43440×10^{3}	2.84351×10^{-3}
12	143	1.62280×10^{4}	5.53105×10^{-4}
15	238	7.57250×10^{4}	3.74046×10^{-4}

the adaptive refinement in this paper than the uniform refinement with biquadratic B-splines. Fig. 7.5 shows that a smaller conditional number obtained using spline bases in this paper than using spline bases in [30] under a given dimension.

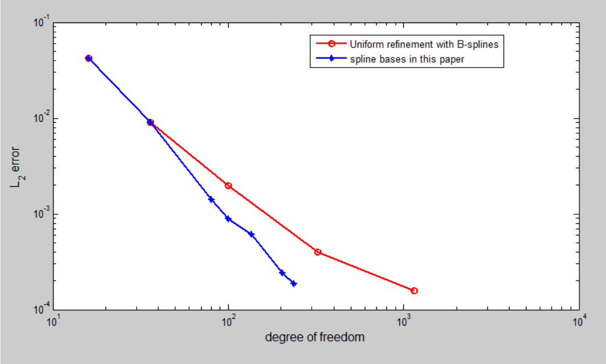


Fig. 7.4. Comparison of the convergence results with spline bases in this paper and uniform refinement.

7.1.2. A two-dimensional elliptic problem defined on $[0,1] \times [0,1]$

In this part, we try to solve a two-dimensional elliptic boundary value problem (BVP)

$$-\Delta u = f, \quad (x,y)\epsilon[0,1]^2$$

with homogeneous conditions. Here $f = -\{10^4[(2x-1)^2(y^2-y)^2 + (x^2-x)^2(2y-1)^2] + 2 \times 10^2(y^2-y+x^2-x)\}e^{10^2(x^2-x)(y^2-y)}$.

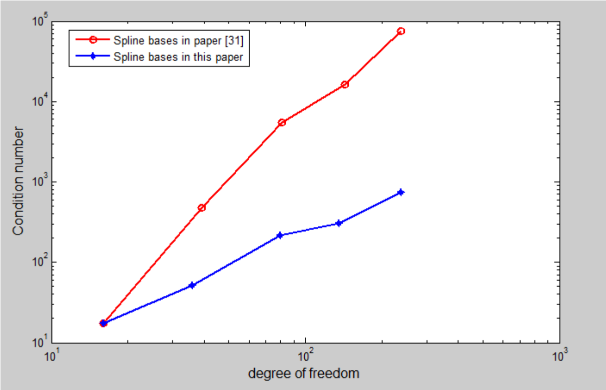


Fig. 7.5. Comparison of condition numbers with spline bases in this paper and spline bases in paper [30].

The exact solution is depicted in Fig. 7.6. Fig. 7.7a shows the relative error $(u_{\text{exact}} - u^{\text{h}})/\|u_{\text{exact}}\|_{L^2}$ between the real solution and the numerical solution after 3 refinements. Fig.7.7b shows the adaptive hierarchical T-meshes after 14 refinements. In Table 7.2 we show the iteration number (n), dimension of spline space, condition number, the error $\|u_{\text{exact}} - u^{\text{h}}\|_{L^2}/\|u_{\text{exact}}\|_{L^2}$

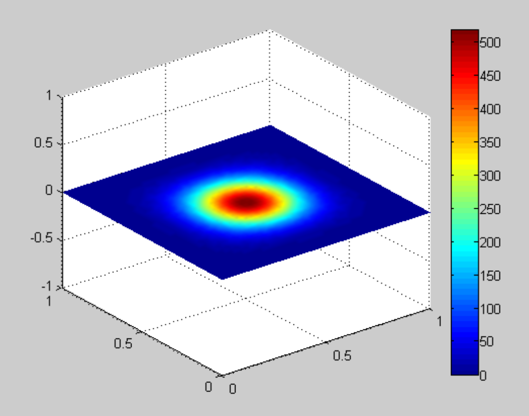
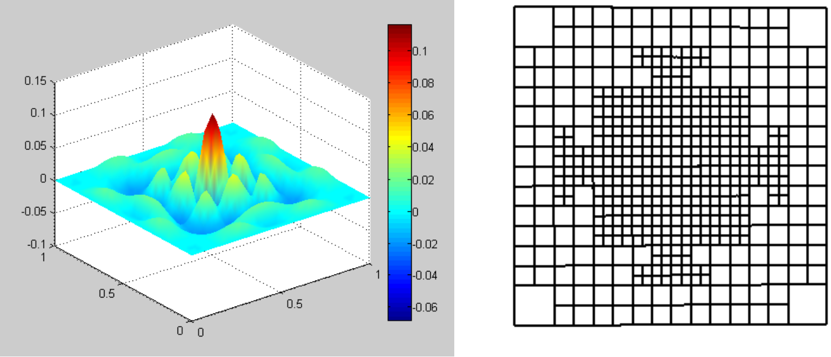


Fig. 7.6. The exact solution.



a b

Fig. 7.7. The relative error after 3 refinements (left) and the mesh after 14 refinement (right).

Table 7.2: Electron affinity (A), Ionization potential (I), Absolute electroneagativity (χ), Global hardness (η) and Electrophilicity (ω) for monomer and dimers.

Table 2			
n	dim	cond	error
Spline bases in this paper			
1	16	4.97740×10^0	5.97478×10^{-1}
3	40	6.41250×10^0	5.1276×10^{-2}
4	72	8.56920×10^0	1.55392×10^{-2}
7	180	2.27106×10^1	1.65848×10^{-3}
14	496	8.23124×10^1	1.87078×10^{-4}
B-splines over uniform tensor-product meshes			
1	16	4.97740×10^0	5.97478×10^{-1}
2	36	5.0076×10^0	2.35584×10^{-1}
3	100	6.8156×10^0	1.42389×10^{-2}
4	324	2.43688×10^1	1.09188×10^{-3}
5	1156	9.55392×10^1	1.19142×10^{-4}
Spline bases in [30]			
1	16	4.97740×10^0	5.97478×10^{-1}
3	40	5.631579×10^2	5.14273×10^{-2}
4	72	1.18040×10^3	1.61176×10^{-2}
10	194	3.81900×10^4	2.02471×10^{-3}
20	512	8.19940×10^4	3.59690×10^{-4}

for splines bases in this paper, B-splines over uniform tensor-product meshes and spline bases in [30]. Table 7.2 and Fig. 7.8 show that the adaptive refinement in this paper reaches a given tolerance with fewer degrees of freedom than the uniform refinement with biquadratic B-splines over tensor-product meshes. Fig. 7.9 shows that a smaller conditional number achieved using spline bases in this paper than using spline bases in [30] under a given dimension.

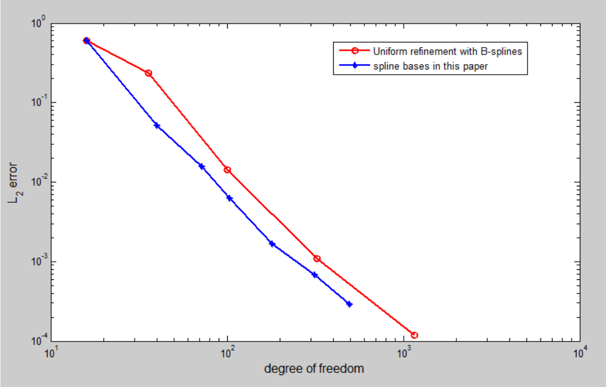


Fig. 7.8. Comparison of the convergence results with spline bases in this paper and uniform refinement.

7.2. Surface fitting

Given an open surface triangulation with vertices $V_j, j = 1, \cdots, N$ in 3D space, the corresponding parameter values $(x_j, y_j), j = 1, \cdots, N$ are obtained using the method of conformal parametrization in [12], and the parameter domain is assumed to be $[0, 1] \times [0, 1]$.

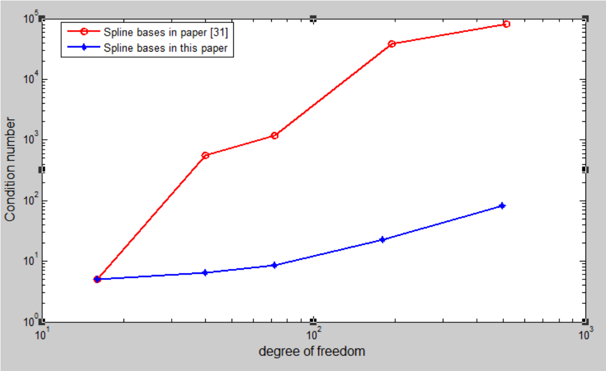


Fig. 7.9. Comparison of condition numbers with spline bases in this paper and spline bases in paper [30].

To construct a spline to fit the given surface, we need only to compute all the control points $P_i, i = 1, \cdots, m$ associated with all the basis functions $b_i, i = 1, \cdots, m$. We denote the fitting spline $\sum_{i=1}^{m} P_i b_i(x, y)$ as $S(x, y)$. To find the control points, we just need to solve an over-determined linear system $S(x_j, y_j) = V(x_j, y_j), j = 1, \cdots, N$ using the least square fitting.

The surface fitting scheme repeats the following two steps until the fitting error in each cell is less than the given tolerance ε:

1. Compute all the control points for all the basis functions.

2. Find all the cells whose errors are greater than the given error tolerance ε, then subdivide these cells into four subcells to form a new mesh, and construct basis functions for the new mesh. We use $\max_{(x,y)\in\mathscr{C}} \|V(x, y) - S(x, y)\|$ to define the fitting error over cell $\mathscr{C}$. In practice, the fitting error is calculated as the maximum of $\|V(u_j, v_j) - S(u_j, v_j)\|$ for certain sample points (u_j, v_j) in $\mathscr{C}$.

Fig. 7.10 is provided to illustrate the above surface fitting scheme.

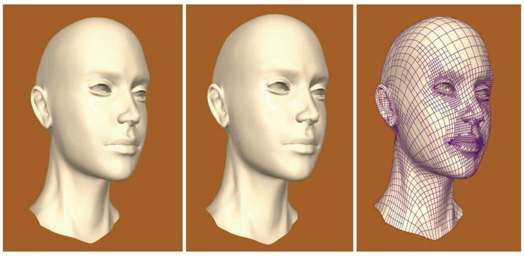


Fig. 7.10. Fitting female head with polynomial splines over a hierarchical T-mesh.

8. Conclusions and Future Studies

Under the rule that the level-difference of the adjacent cells is at most one, a basis of the biquadratic polynomial spline spaces is constructed based on the CVR graphs. The functions in the basis are all tensor-product B-splines, and they have the properties of nonnegativity and local support. However, the basis does not have the property of unit partition. To make the basis functions more efficient for geometric modeling, we also give out a basis with unit partition.

In this paper, we restrict that the level-difference of adjacent cells is one at most, in the future we will discuss the construction of spline spaces with weaker constraint of the level-difference.

For $\mathbf{S}(2,2,1,1,\mathscr{T})$, the dimension equals to the number of cells in the corresponding CVR graph of $\mathscr{T}$. For spline spaces $\mathbf{S}(m,n,m-1,n-1,\mathscr{T})$, in [8] a conjecture was provided, which states that the dimensions of spline spaces over hierarchical T-meshes equal to the dimensions of lower-degree spline spaces over the corresponding CVR-graph. Now we are working on proving this conjecture for bicubic spline spaces, and defining their basis functions according to the structure of CVR graph as well.

Acknowledgement. The authors would like to thank Dr. F. Chen of School of Mathematical Sciences, University of Science and Technology of China for his constructive suggestions in the construction of the basis. The authors are grateful to the referees who address some questions about the basis that we did not consider. The authors are supported by the NSF of China (No. 11371341, No.11626253, No. 11601114, No. 11526069), the Anhui Provincial Natural Science Foundation (No. 1608085QA14).

References

[1] Y. Bazilevs, L.B. Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes, *Math. Models Methods Appl. Sci.*, **16** (2006), 1031-1090.

[2] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg, Isogeometric analysis using T-splines, *Comput. Meth. Appl. Mech. Engrg.*, **199** (2010), 229-263.

[3] 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), 507-513.

[4] A. Buffa, D. Cho, and G. Sangalli, Linear independence of the T-spline blending functions associated with some particular T-meshes, *Comput. Meth. Appl. Mech. Engrg.*, **199** (2010), 1437-1445.

[5] M. Dorfel, B. Juttler, and B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, *Comput. Meth. Appl. Mech. Engrg.*, **199** (2010), 264-275.

[6] J. Deng, F. Chen, and Y. Feng, Dimensions of spline spaces over T-meshes, *J. Comput. Appl. Math.*, **194** (2006), 267-283.

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

[8] J. Deng, F. Chen, and L. Jin, Dimensions of biquadratic spline spaces over T-meshes, *J. Comput. Appl. Math.*, **238** (2013), 68-94.

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

[10] Q. Du, L. T, and X. Zhao, A Convergent Adaptive Finite Element Algorithm for Nonlocal Diffusion and Peridynamic Models, *SIAM J. Numer. Anal.*, **51** (2013), 1211-1234.

[11] G. Farin, Curves and Surfaces for CAGD - A Practical Guide, 5th ed., Morgan-Kaufmann, 2002.

[12] M.S. Floater, Parameterization and smooth approximation of surface triangulations, *Comput. Aided Geom. Des.*, **14** (1997), 231-250.

[13] D.R. Forsey, and R.H. Bartels, Hierarchical B-spline Refinement, *Comput. Graph.*, **22** (1988), 205-212.

[14] C. Giannelli, B. Juttler, and H. Speleers, Thb-splines: The truncated basis for hierarchical splines, *Comput. Aided Geom. Des.*, **29** (2012), 485-498.

[15] R. Kraft, Adaptive and linearly independent multilevel B-splines, In Surface Fitting and Multiresolution Methods, *Vanderbilt University Press*, **2** (1997), 209-218.

[16] X. Li, and M.A. Scott, Analysis-suitable T-splines: characterization, refineability, and approximation, *Math. Models Mehthods Appl.*, **24** (2014), 1141-1164.

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

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

[19] X. Li, and F. Chen, On the instability in the dimension of splines spaces over T-meshes, *Comput. Aided Geom. Des.*, **28** (2011), 420-426.

[20] T.W. Sederberg, J.M. Zheng, A. Bakenov, and A. Nasri, T-splines and T-NURCCs, *ACM Trans. Graph.*, **22** (2003), 477-484.

[21] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J.M. Zheng, and T. Lyche, T-spline simplification and local refinement, *ACM Trans. Graph.*, **23** (2004), 276-283.

[22] T.W. Sederberg, G.T. Finnigan, X. Li, and H. Lin, Watertight trimmed NURBS, *ACM Trans. Graph.*, **27** (2008).

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

[24] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, and T.J.R. Hughes, Isogeometric finite element data structures based on Bzier extraction of T-splines, *Int. J. Numer. Methods Engrg.*, **88** (2011), 126-156.

[25] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wchner, K.U. Bletzinger, Y. Bazilevs, and T. Rabczuka, Rotation free isogeometric thin shell analysis using PHT-splines, *Comput. Meth. Appl. Mech. Engrg.*, **200** (2011), 3410-3424.

[26] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk, Isogeometric finite element analysis using polynomial splines over hierarchical T-meshes, *IOP Conf. Ser. Mater. Sci. Eng.*, **10** (2010), doi:10.1088/1757-899X/10/1/012238.

[27] 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. Meth. Appl. Mech. Engrg.*, **200** (2011), 1892-1908.

[28] A.V. Vuong, C. Giannelli, B. Juttler, and B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, *Comput. Meth. Appl. Mech. Engrg.*, **200** (2011), 3554-3567.

[29] M. Wu, J. Deng, and F. Chen, The dimension of Spline Spaces with Highest Order Smoothness over Hierarchical T-meshes, *Comput. Aided Geom. Des.*, **30** (2013), 20-34.

[30] M. Wu, J.-L. Xu, R.-M. Wang, and Z.-W. Yang, Hierarchical bases of spline spaces with highest order smoothness over hierarchical T-subdivisions, *Comput. Aided Geom. Des.*, **29** (2012), 499-509.

[31] P. Wang, J. Xu, J. Deng, and F. Chen, Adaptive isogeometric analysis using rational PHT-splines, *Comput. Aided Des.*, **43** (2011), 1438-1448.

[32] J. Wang, Z. Yang, L. Jin, J. Deng, and F. Chen, Parallel and adaptive surface reconstruction based on implicit PHT-splines, *Comput. Aided Geom. Des.*, **28** (2011), 463-474.

[33] C. Zeng, F. Deng, J.-S. Deng, Bicubic hierarchical B-splines: Dimensions, completeness, and bases, Comput. Aided Geom. Des., **38** (2015), 1-23.

[34] C. Zeng, F. Deng, X. Li, J.-S. Deng, Dimensions of biquadratic and bicubic spline spaces over hierarchical T-meshes, *J. Comput. Appl. Math.*, **287** (2015), 162-178.