Algorithms for Space Mapping Method on Spline Spaces over Modified Hierarchical T-Meshes

Jingjing Liu 1, Li Zhang 2,* and Weihong Zhang 1

1 School of Mathematics and Statistics, Hefei Normal University, Heifei 230601, China
2 School of Mathematics, Hefei University of Technology, Heifei 230000, China
* Correspondence: lizhang@hfut.edu.cn; Tel.: +86-1308-340-6020

Abstract: The space-mapping method provides a novel method for dimension formulae explanation and basis construction for the spline space over hierarchical T-meshes. By the space-mapping method, we provide a unique basis construction framework that incorporates basis modification of the spline space over modified hierarchical T-meshes. The subdivision rules on the modified hierarchical T-meshes are given to prevent the redundant edges that exist on hierarchical T-meshes. In the basis construction framework, we describe the spline-modification mechanism over the modified hierarchical T-mesh when the cells of the corresponding crossing vertex relationship graph (CVR graph) are adjusted. We provide the framework’s algorithms for basis construction and modification. Moreover, we discuss the application of the splines that are constructed by the framework to surface reconstruction with adaptive refinement. In comparison to splines over hierarchical T-meshes, the modified hierarchical T-meshes have fewer cells subdivided when achieving similar accuracy.

Keywords: modified hierarchical T-meshes; space mapping method; subdivision rules; basis modification

MSC: 65D07

1. Introduction

Multivariate splines over tensor-product spaces are one of the most popular tools for the representation of free-form surfaces in computer-aided design (CAD) [1]. More recently, these splines have also been applied for the discretization of partial differential equations in isogeometric analysis (IGA) [2], which uses the same class of functions to describe an object’s geometry and to represent the fields describing physical phenomena to bridge the gap between CAD and computer-aided engineering (CAE). However, the disadvantage of using tensor product splines is that when a knot is inserted, the resulting knot line crosses the whole domain, thus eliminating the possibility of performing local refinement. Because refining the entire mesh leads to a large number of superfluous control points or coefficients, locally refinable multivariate splines have become a popular research topic for design and analysis. In recent years, several locally refinable splines have been widely used in practical problems, such as T-splines, hierarchical B-splines (HB-splines) and truncated hierarchical B-splines (THB-splines), Locally refined B-splines (LR-splines) and polynomial splines over hierarchical T-meshes (PHT-splines) are attractive. We will give a literature review of these splines as follows.

T-splines have overcome the difficulty of local refinement by producing surfaces from control meshes that allow T-junctions [3,4]. T-splines’ potential in IGA has been reported in [5,6]. To ensure linear independence, which is a crucial need for using T-splines in analysis, additional constraints are proposed in [7,8]. Late developments have led to the development of a novel type of spline known as analysis-suitable++ T-splines (AS++ T-splines), which incorporate analysis-suitable T-splines (AS T-splines) as a special instance [9].
HB-splines offer a traditional method for obtaining local refinement in geometric modeling [10]. It has been stated that HB-splines can be applied in geometric modeling and isogeometric analysis [11–14]. As a modification of HB-splines, THB-splines have been proposed for defining locally supported basis functions that hold the partition of unity property [15,16]. THB-splines have been an excellent choice for use in a variety of applications, including isogeometric analysis, computer-aided design, and geometric modeling [16–19].

Locally refined splines (LR splines) can span the complete piecewise polynomial space on the LR mesh and form a nonnegative partition of unity [20]. The LR mesh breaks the tensor-product mesh structure. LR splines in 2D can be constructed in a hierarchical manner [21], and additional properties of LR splines have been discussed in more detail in [22].

PHT splines, which are the bicubic splines over hierarchical T-meshes with $C^1$ continuity, produce nested spline spaces by using a modification mechanism [23]. Works on PHT splines discuss the splines from the perspective of spline spaces, provide the dimension formula [23] and offer the basis construction framework of PHT-splines [24]. Therefore, PHT splines ensure properties such as nonnegativity, partition of unity, linear independence, and completeness. With regard to these excellent properties, PHT splines have been effective in a variety of applications, including the reconstruction of surface models [24–26], the adaptive finite element method for solving elliptic equations [27], and the isogeometric analysis of elastic problem solving [28–30]. Additionally, the PHT splines have been modified as modified PHT splines (MPHT-splines) [30,31]. The construction of PHT splines is made easier by a succinct topological interpretation of the dimensional formula. To do this, [32] introduces the proper crossing vertex relationship graph (CVR graph) and provides a concise topological explanation for the dimension formula of PHT splines. By the CVR graph, the biquadratic and bicubic polynomial basis functions of splines over hierarchical T-meshes have been constructed in [33,34], respectively, and they performed well in the application for geometric modeling and IGA. Unfortunately, the level difference for the hierarchical T-meshes in [33,34] cannot be greater than one. To overcome this restriction, a novel bijective mapping between the biquadratic polynomial spline spaces over the hierarchical T-mesh and the piecewise constant space over the corresponding CVR graph has been proposed in [35]. The bijective mapping offers a succinct topological explanation for the dimension formula of the biquadratic polynomial spline spaces over hierarchical T-meshes and provides a new manner for the basis construction of the biquadratic polynomial spline spaces over hierarchical T-meshes. The basis functions, which are constructed by the space mapping method, are linearly independent, form a partition of unity, and are suitable for CAD [35].

In PHT spline theory, one of the most significant operations is known as the local refinement of PHT spline. Splines that are defined on hierarchical T-meshes allow for the possibility of local refinement. This is because hierarchical T-meshes have a naturally level structure. However, for biquadratic polynomial spline spaces over hierarchical T-meshes, there are numerous redundant edges on the hierarchical T-meshes. In this paper, we give some subdivision rules on the modified hierarchical T-meshes, which are split in half in the mesh-refining process. This is done to prevent the redundant edges that exist on hierarchical T-meshes. In [35], we have used the space mapping method to give the basis construction of the biquadratic polynomial spline space over hierarchical T-meshes. However, for each level space over the hierarchical T-mesh, all of the basis functions need to be reconstructed when their partially supported cells are subdivided. This takes a significant amount of time. In this paper, we provide a unique basis construction framework that incorporates basis modification of the biquadratic polynomial spline space over modified hierarchical T-meshes. This framework is developed by utilizing the work that we have done in [35].

We organize the remainder of the paper as follows. In Section 2, we review some essential conceptions related to hierarchical T-meshes, modified hierarchical T-meshes and spline spaces. In Section 3, we describe the space-mapping method for spline spaces.
over modified hierarchical T-meshes. For the spline spaces over the modified hierarchical
T-meshes, we give the basis construction framework that allows modification in Section 4,
and the corresponding algorithms are also provided. We give the numerical examples
to fit several open surface models in Section 5. In Section 6, our paper closes with some
concluding remarks.

2. Preliminaries

In this section, we will recall some preliminary knowledge that is useful for our
method, such as notations on T-meshes, modified hierarchical T-meshes, and spline space
over T-meshes, and so on.

2.1. T-Meshes

For the T-mesh \( \mathcal{T} \), the definitions of vertices, edges, and cells are the same as that
in [23]. We use Figure 1 to illustrate the corresponding notations as follows.

![Figure 1](image)

The vertices, edges, and cells on \( \mathcal{T} \). The green square vertices denote crossing vertices,
the green triangular vertices denote T-junctions. The gray cell denotes an interior cell, the yellow cell
denotes a boundary cell.

We refer to a grid point in \( \mathcal{T} \) as a vertex of \( \mathcal{T} \). A vertex is referred to as a boundary
vertex if it occupies the boundary grid line of \( \mathcal{T} \); otherwise, it is referred to as an interior
vertex. For example, the blue square vertices denote boundary vertices, the green square
vertices, and the green triangular vertices denote interior vertices. For the interior vertices,
if the valence of an interior vertex is four, it is referred to as a crossing vertex; if the valence
of an interior vertex is three, it is referred to as a T-junction. For example, the green square
vertices denote crossing vertices, and the green triangular vertices denote T-junctions.

We refer to the line segment that connects two adjacent vertices on one grid line as
an edge of \( \mathcal{T} \). An edge is referred to as a boundary edge if it occupies on the boundary of
\( \mathcal{T} \); otherwise, it is referred to as an interior edge. For example, \( b_0b_1 \) denotes a boundary
edge, \( b_1v_0 \) denotes an interior edge. We refer to the edge as an l-edge if it is the longest
possible line segment and its endpoints are either boundary vertices or T-junctions. For
example, \( b_1v_0 \) and \( b_0b_2 \) denote l-edges. We refer to the interior l-edge as a T-l edge [36] if its
two endpoints are both T-junctions; for example, the red segment \( v_2v_3 \) denotes a T-l-edge.
Additionally, we refer to an edge as a c-edge if it is the longest possible line segment and its
inner vertices are all T-junctions; for example, \( v_1b_3 \) denotes a c-edge.

We refer to each rectangular grid element as a cell of \( \mathcal{T} \). A cell is referred to as an
interior cell if all its edges are interior edges; otherwise, it is referred to as a boundary
cell. For example, the gray cell denotes an interior cell, and the yellow cell denotes a
boundary cell.

2.2. Modified Hierarchical T-Meshes

The modified hierarchical T-meshes can be defined in a recursive manner [31], which
is similar to the hierarchical T-meshes in [24]. \( \mathcal{T}^k \) denotes the mesh at level \( k \), \( \mathcal{T}_k \) denotes
the set consist of all the cells at level \( k \), and the set consists of the cells that are chosen to be
subdivided, denoted as \( \Theta_k \), and \( \Theta_k \subset \mathcal{T}_k \).
1. Start with a tensor product mesh $T^0$.
2. Recursive steps: Subdivide each cell in $\Theta_k$ into $2 \times 2$ equal subcells or two equal subcells to obtain a new T-mesh $T^{k+1}$, $k = 0, 1, \ldots, N$. $T_{k+1}$ denotes the set that consists of all current cells that emerge at level $k+1$ after subdivision.
3. $T = T^{N+1}$.

Figure 2 illustrates the refinement process of a modified hierarchical T-mesh dynamically.

**Figure 2.** An example for the refinement process of a modified hierarchical T-mesh.

2.3. Spline Spaces over T-Meshes

Given a T-mesh $T$, all of the cells in $T$ are denoted as $F$, and the region occupied by the cells in $T$ is denoted as $\Omega$. The spline spaces can be defined as 

$$S(m, n, \alpha, \beta, T) := \{ f(x, y) \in C^{\alpha, \beta}(\Omega) : f(x, y)|_{\phi} \in P_{mn}, \forall \phi \in F \}. \quad (1)$$

Here, $P_{mn}$ denotes the polynomial space with bi-degree $(m, n)$, and $C^{\alpha, \beta}$ denotes the space consisting of all of the bivariate functions. These functions are $\alpha$ order-continuous in the $x$-direction and $\beta$ order-continuous in the $y$-direction on $\Omega$. Obviously, $S(m, n, \alpha, \beta, T)$ should be a linear space.

In [32], the spline spaces over $T$ with homogeneous boundary conditions corresponding to $S(m, n, \alpha, \beta, T)$ have been defined as

$$\mathfrak{S}(m, n, \alpha, \beta, T) := \{ f(x, y) \in C^{\alpha, \beta}(\mathbb{R}^2) : f(x, y)|_{\phi} \in P_{22}, \forall \phi \in F, \text{ and } f|_{\mathbb{R}^2 \setminus \Omega} \equiv 0 \}. \quad (2)$$

As the extended T-mesh [32] will play an important role in our study, we repeat the definition of the extended T-mesh as follows.

**Definition 1.** For the T-mesh $T$ associated with $S(m, n, \alpha, \beta, T)$, the extended T-mesh can be generated in the following manner.

1. Add $m$ edges to the horizontal boundaries in an average manner; add $n$ edges to the vertical boundaries in an average manner.
2. Connect the boundary vertices on $T$ to the outer layer edges.

We denote the extended T-mesh of $T$ associated with $S(m, n, \alpha, \beta, T)$ as $T^\varepsilon$ for convenience.

Figure 3 shows a T-mesh $T$, and its extended T-mesh $T^\varepsilon$ that associates with $S(2, 2, 1, 1, T)$. Ref. [32] gives Theorem 1 to illustrate that the spline space $S(m, n, \alpha, \beta, T)$ is related to $\mathfrak{S}(m, n, \alpha, \beta, T^\varepsilon)$ closely.
Figure 3. A T-mesh $\mathcal{T}$ and its extended T-mesh $\mathcal{T}^e$ associated with $S(2, 2, 1, 1, \mathcal{T})$.

**Theorem 1** ([32]). Given a T-mesh $\mathcal{T}$, $\mathcal{T}^e$ denotes the extended T-mesh associated with $S(m, n, \alpha, \beta, \mathcal{T})$, and $\Omega$ denotes the region occupied by all cells of $\mathcal{T}$. Then,

$$S(m, n, \alpha, \beta, \mathcal{T}) = S(m, n, \alpha, \beta, \mathcal{T}^e)|_\Omega,$$

$$\dim S(m, n, \alpha, \beta, \mathcal{T}) = \dim S(m, n, \alpha, \beta, \mathcal{T}^e).$$

In this paper, we will discuss the subdivision rules and the basis modification framework of $S(m, n, \alpha, \beta, \mathcal{T}^e)$ instead of $S(m, n, \alpha, \beta, \mathcal{T})$, and then use Theorem 1 to obtain the corresponding results of $S(m, n, \alpha, \beta, \mathcal{T})$.

### 3. The Space Mapping Method over the Modified Hierarchical T-Meshes

The space mapping method has been proposed in [35], which gives the conclusion that the biquadratic polynomial spline spaces over the hierarchical T-mesh is isomorphic to the piecewise constant space over the corresponding CVR graph. The bijective mapping in [35] has been used to provide a basis construction method for biquadratic polynomial spline spaces over hierarchical T-meshes. In this part, we discuss the space-mapping method over the modified hierarchical T-meshes as follows.

#### 3.1. CVR Graph of Modified Hierarchical T-Meshes

Because the CVR graph [32,35] is critical in the space-mapping method, it can be defined in a similar manner to that of modified hierarchical T-meshes as follows.

**Definition 2** ([32,35]). Given a modified hierarchical T-mesh $\mathcal{T}$, the CVR graph of $\mathcal{T}$ can be constructed in the following manner.

1. Retain the crossing vertices and the line segments with two endpoints that are crossing vertices.
2. Remove the other edges and vertices from $\mathcal{T}$.

We denote the CVR graph of $\mathcal{T}$ as $\mathcal{G}$ for convenience.

Figure 4 shows an example of a CVR graph. The hierarchical T-mesh can be generated by adding some T-I edges to a modified hierarchical T-mesh, and a T-I edge on hierarchical T-mesh decays to a point on the corresponding CVR graph. Similar to [35], the notations of modified hierarchical T-meshes and the corresponding CVR graph can be defined in Table 1.
Mathematics 2022, 10, 3864

(a) $\mathcal{F}$ associated with $S(2, 2, 1, 1, \mathcal{F})$

(b) The CVR graph $\mathcal{G}$

Figure 4. An example of the modified hierarchical T-mesh and its CVR graph. “•” denotes the domain-center of T-rectangle domain in the red box; “○” denotes the domain-centre of the T-rectangle domain in the green box.

Table 1. Some notations for modified hierarchical T-meshes and the corresponding CVR graphs [35].

| Notations                        | Abbreviations | Definitions                                                                 |
|----------------------------------|---------------|-----------------------------------------------------------------------------|
| Modified hierarchical            |               |                                                                             |
| T-meshes:                        |               |                                                                             |
| P-cell                           | $\mathcal{PC}$| Four corner vertices of the cell $\theta$ are all crossing vertices or boundary vertices, $\theta$ is called a P-cell. |
| T-cell                           |               | At least one of the four corner vertices of the cell $\theta$ is a T-junction, $\theta$ is called a T-cell. |
| T-connected                      |               | If two T-cells have one common T-junction, they are called T-connected.     |
| T-connection                     | $\mathcal{T C}$| The union that every two cells are T-connected.                             |
| P-domain                         | $\mathcal{PD}$| The domain that is occupied by a pure cell.                                 |
| T-connection domain              |               | The domain that is occupied by a T-connection.                             |
| T-rectangle domain               | $\mathcal{TRD}$| The minimal rectangular domain that occupies on a T-connection.              |
| Domain                           | $\mathcal{D}$ | Each T-rectangle-domain or P-domain.                                       |
| Domain-center                    |               | The center of the domain.                                                   |
| one-neighbor cell                |               | For all the cells that are adjacent to the T-connection, the cells with the lowest level are called the one-neighbor cell of the T-connection. |
| The level of the T-connection    |               |                                                                             |
| CVR graph of modified            |               |                                                                             |
| hierarchical T-meshes:           |               |                                                                             |
| g-cell                           |               | Each cell of the CVR graph.                                                |
| P-g-cell                         | $\mathcal{PGC}$| Each g-cell that corresponds to a P-cell in $\mathcal{F}$.                  |
| T-g-cell                         | $\mathcal{TGC}$| Each g-cell that corresponds to a T-connection-domain in $\mathcal{F}$.     |

For example, in Figure 4a, $\theta_0, \theta_4, \theta_9$ we denote P-cells. The P-g cell $\phi_0$ in Figure 4b corresponds to $\theta_0$ in Figure 4a, the P-g-cell $\phi_2$ in Figure 4b corresponds to $\theta_4$ in Figure 4a, and the P-g-cell $\phi_4$ in Figure 4b corresponds to $\theta_9$ in Figure 4a. $\theta_1, \theta_2, \theta_3, \theta_5, \theta_6, \theta_7,$ and $\theta_8$ in Figure 4a denote T-cells. The T-connection that consists of $\theta_1, \theta_2,$ and $\theta_3$ is denoted as $\mathcal{TC}_0$, the minimal domain that occupies on $\mathcal{TC}_0$ is the T-connection domain of $\mathcal{TC}_0$, in Figure 4a. The domain surrounded by the red line is denoted as the T-rectangle domain of $\mathcal{TC}_0$, which is denoted as $\mathcal{TRD}_0$. $\theta_4$ denotes the one-neighbor cell of $\mathcal{TC}_0$, the level of $\mathcal{TC}_0$ is also the level of $\theta_4$, and the T-g-cell $\phi_1$ in Figure 4b corresponds to the T-connection domain of $\mathcal{TC}_0$. The T-connection that consists of $\theta_5, \theta_6, \theta_7,$ and $\theta_8$ is denoted as $\mathcal{TC}_1$, and the minimal domain that occupies $\mathcal{TC}_1$ is the T-connection domain of $\mathcal{TC}_1$. In Figure 4a, the domain surrounded by the blue square is denoted as the T-rectangle domain of $\mathcal{TC}_1$. $\theta_9$ denotes the one-neighbor cell of $\mathcal{TC}_1$, the level of $\mathcal{TC}_1$ is also the level of $\theta_9$, and the T-g-cell $\phi_3$ in Figure 4b corresponds to the T-connection domain of $\mathcal{TC}_1$. 
3.2. The Isomorphic Property

In [36,37], the dimensions of T-l edges are proposed as follows.

**Theorem 2.** For a T-l edge $E \in \mathcal{T}$, if $E$ only contains $V_0$ interior crossing vertices, then,
\[
\text{dim } W[E] = (V_0 - 2 - 1)^+ := \max(0, V_0 - 2 - 1)
\]
holds for $S(2,2,1,1,\mathcal{T})$.

In particular, for the $S(2,2,1,1,\mathcal{T})$, if $V_0 = 1$, then
\[
\text{dim } W[E] = (V_0 - d - 1)^+ := \max(0, 1 - 2 - 1) = 0,
\]
and the trivial l-edges [36,37] can be defined as follows.

**Definition 3 ([35]).** If there exists only one crossing vertex on the T-l edge $E \in \mathcal{T}$, $E$ is defined as a trivial l-edge associated with $S(2,2,1,1,\mathcal{T})$.

For the hierarchical T-mesh $\mathcal{T}$, $\mathcal{G}$ denotes the CVR graph of $\mathcal{T}$, there exists a bijective mapping between $S(2,2,1,1,\mathcal{T})$ and $S(0,0,-1,-1,\mathcal{G})$, and the two spaces are isomorphic [35]:
\[
S(2,2,1,1,\mathcal{T}) \cong S(0,0,-1,-1,\mathcal{G}).
\]

For a modified hierarchical T-mesh, $\Theta_k$ denotes the subdivided cells whose levels are $k$ first, for each cell $\theta \in \Theta_k$, there are three different subdivision types that tend to choose [31].

- If $\theta$ is subdivided into two equal, vertically adjacent subcells, the subdivision type of $\theta$ is denoted as V.
- If $\theta$ is subdivided into two equal, horizontally adjacent subcells, the subdivision type of $\theta$ is denoted as H.
- If $\theta$ is subdivided into $2 \times 2$ equal subcells, the subdivision type of $\theta$ is denoted as C.

To avoid the trivial l-edges, we can consider our subdivision rules on a modified hierarchical T-mesh as follows.

3.3. Mesh Refinement Rules

Because the subdivision scheme for modified hierarchical T-meshes can avoid the trivial l-edges that appear in hierarchical T-meshes, in this section we give our mesh refinement rules for modified hierarchical T-meshes.

To make use of the subdivision scheme of the modified hierarchical T-meshes, we use Figure 5 to illustrate the mesh refinement rules on a part of modified hierarchical T-mesh $\mathcal{T}_k$.

![Figure 5. Mesh-refinement rules.](image)

In Figure 5a, for $\theta_0 \in \Theta_k$, $\theta_i$, $i = 1, \ldots, 4$ are the cells around $\theta_0$. The refinement rules on $\theta_0$ can be discussed as follows.

- If $\theta_2 \in \Theta_k$ and the subdivision type of $\theta_2$ is C or V, then the subdivision type of $\theta_0$ is V. Figure 5b shows the refinement example.
- If $\theta_i \in \Theta_k$ and the subdivision type of $\theta_i$ is C or V, $i = 2, 4$, then the subdivision type of $\theta_0$ is V.
• If \( \theta_1 \in \Theta_k \) and \( \theta_2 \in \Theta_k \), the subdivision type of \( \theta_1 \) is C or H, the subdivision type of \( \theta_2 \) is C or V, and then the subdivision type of \( \theta_0 \) is C.

If the position of \( \theta_i, i = 1, 2, 3, 4 \) is changed, the refinement rules on \( \theta_0 \) can be obtained similarly.

As a modified hierarchical T-mesh can be converted to a hierarchical T-mesh by adding corresponding trivial l-edges, we give Lemma 1 to illustrate the effect of trivial l-edges on the CVR graph.

**Lemma 1.** Assume that \( \mathcal{T} \) is a hierarchical T-mesh, \( \mathcal{G} \) denotes the CVR graph of \( \mathcal{T} \), the trivial l-edges of \( \mathcal{T} \) correspond to the vertex that is on the hanging edges of \( \mathcal{G} \).

**Proof of Lemma 1.** We use proof by contradiction to prove this lemma. Suppose that \( E \) is an arbitrary trivial l-edge in \( \mathcal{T} \), \( E \) corresponds to the vertex \( V \) in \( \mathcal{G} \), the edge which \( V \) occupies on is denoted as \( L \). We assume that \( L \) is not the hanging edge of \( \mathcal{G} \), and then \( V \) occupies the edge of a g-cell of \( \mathcal{G} \). Then, there are at least two crossing vertices on \( E \). As \( E \) is a trivial l-edge of \( \mathcal{T} \), for the spline space \( S(2, 2, 1, 1, \mathcal{T}) \), \( E \) only contains one interior crossing vertex, this contradicts the assumption. Thus, \( L \) is a hanging edge of \( \mathcal{G} \). The lemma is proven.

For each trivial l-edge \( E \in \mathcal{T} \), as \( \dim W[E] = 0 \), \( E \) is useless for \( S(2, 2, 1, 1, \mathcal{T}) \), from Lemma 1, \( E \) corresponds to a vertex on the hanging edges of \( \mathcal{G} \). We can conclude similarly as in Equation (6) via the relationship between the CVR graph of the hierarchical T-mesh and the CVR graph of its corresponding modified hierarchical T-mesh. Our study continues here, and we can give the proof as follows.

**Theorem 3.** Given a modified hierarchical T-mesh \( \mathcal{T} \), assume that \( \mathcal{G} \) is the corresponding CVR graph of \( \mathcal{T} \), then,

\[
S(2, 2, 1, 1, \mathcal{T}) \cong S(0, 0, -1, -1, \mathcal{G}).
\]  

**Proof of Theorem 3.** Assume that \( \mathcal{T}_a \) is denoted as the hierarchical T-mesh and corresponds to \( \mathcal{T} \) by adding the trivial l-edges \( E_i, i = 0, \ldots, L \), \( \mathcal{G}_a \) is denoted as the CVR graph of \( \mathcal{T}_a \). Then,

\[
S(2, 2, 1, 1, \mathcal{T}_a) \cong S(0, 0, -1, -1, \mathcal{G}_a),
\]  

and each basis function of \( S(2, 2, 1, 1, \mathcal{T}_a) \) corresponds to a basis function of \( S(0, 0, -1, -1, \mathcal{G}_a) \) [35]. For each trivial l-edge \( E \in \mathcal{T}_a \), \( \dim W[E] = 0 \); thus, each basis function of \( S(2, 2, 1, 1, \mathcal{T}_a) \) is the same with the corresponding basis function of \( S(2, 2, 1, 1, \mathcal{T}) \). From Lemma 1, \( E \) corresponds to the vertex that is on the hanging edges of \( \mathcal{G} \), in other words, the cells of \( \mathcal{G} \) are the same with the cells of \( \mathcal{G}_a \), and each basis function of \( S(0, 0, -1, -1, \mathcal{G}_a) \) is the same with the corresponding basis function of \( S(0, 0, -1, -1, \mathcal{G}) \). Thus, each basis function of \( S(2, 2, 1, 1, \mathcal{T}) \) corresponds to a basis function of \( S(0, 0, -1, -1, \mathcal{G}) \),

\[
S(2, 2, 1, 1, \mathcal{T}) \cong S(0, 0, -1, -1, \mathcal{G}).
\]

The theorem is proven.

From Theorem 3, we conclude that the trivial l-edges do not contribute anything useful to the mapping; hence, the bijective mapping can be described similar to that of [35] in the following manner.

For convenience, we denote \( S(2, 2, 1, 1, \mathcal{T}) \) as \( S^2(\mathcal{T}) \) in the following part of this paper.
3.4. The Mapping

**Definition 4 ([35])**. Given \( f(x, y) \in \mathcal{S}^2(\mathcal{F}) \), and \( D := [x_0, x_1] \times [y_0, y_1] \) is a rectangular domain. Let \( u = \frac{x-x_0}{x_1-x_0}, v = \frac{y-y_0}{y_1-y_0} \), \( f(x, y) \) can be expressed as

\[
f(x, y) = \sum_{j=0}^{2} \sum_{k=0}^{2} b_{jk} B_j^2(u) B_k^2(v),
\]

where \( B_j^2(u) \) and \( B_k^2(v) \) are Bernstein polynomials, we obtain

\[
f(x, y) = b_{1,1} + \sum_{j=0}^{2} (b_{j,0} - b_{1,1}) B_j^2(u) B_1^2(v) + \sum_{j=0, j \neq 1}^{2} (b_{j,1} - b_{1,1}) B_j^2(u) B_1^2(v) + \sum_{j=0}^{2} (b_{j,2} - b_{1,1}) B_j^2(u) B_2^2(v).
\]

A mapping functional \( \phi \) can be defined as

\[
\phi(f(x, y) : D) = b_{1,1}|D.
\]

Here, \( b_{1,1}|D \) denotes the piecewise constant function with value \( b_{1,1} \) on \( D \).

Given \( f(x, y) \in \mathcal{S}^2(\mathcal{F}) \), \( \text{Sup}(f) \) denotes the support of \( f(x, y) \), we recall the notations and abbreviations in Table 1 as follows: \( \mathcal{PC} \) denotes a P-cell of \( \text{Sup}(f) \), \( \mathcal{PD} \) denotes the P-domain of \( \mathcal{PC} \), \( \mathcal{PGC} \) denotes the P-g cell that corresponds to \( \mathcal{PC} \) in \( \mathcal{G} \), \( \mathcal{TC} \) denotes a T-connection of \( \text{Sup}(f) \), \( \mathcal{TRD} \) denotes the T-rectangle domain of \( \mathcal{TC} \), and \( \mathcal{TGC} \) denotes the T-g cell corresponding to \( \mathcal{TC} \) in \( \mathcal{G} \). The mapping between \( \mathcal{S}^2(\mathcal{F}) \) and \( \mathcal{S}^0(\mathcal{G}) \) can be defined in the following manner.

**Definition 5 ([35])**. The mapping between \( \mathcal{S}^2(\mathcal{F}) \) and \( \mathcal{S}^0(\mathcal{G}) \) can be defined as

\[
\Phi : \mathcal{S}^2(\mathcal{F}) \rightarrow \mathcal{S}^0(\mathcal{G}),
\]

\[
\begin{cases}
\phi(f(x, y)|_{\mathcal{PC}} : \mathcal{PD}) \rightarrow g(x, y)|_{\mathcal{PGC}} \\
\phi(f(x, y)|_{\mathcal{TC}} : \mathcal{TRD}) \rightarrow g(x, y)|_{\mathcal{TGC}},
\end{cases}
\]

where \( f(x, y) \in \mathcal{S}^2(\mathcal{F}) \), \( f(x, y)|_{\mathcal{PC}} \) denotes the representation of \( f(x, y) \) on \( \mathcal{PC} \), \( g(x, y) \in \mathcal{S}^0(\mathcal{G}) \), and \( g(x, y)|_{\mathcal{PGC}} \) denotes the representation of \( g(x, y) \) on \( \mathcal{PGC} \), and \( \mathcal{PGC} \) corresponds to \( \mathcal{PC} \), \( f(x, y)|_{\mathcal{TC}} \) denotes the representation of \( f(x, y) \) on the one-neighbor cell of \( \mathcal{TC} \), and \( g(x, y)|_{\mathcal{TGC}} \) denotes the representation of \( g(x, y) \) on \( \mathcal{TGC} \), and \( \mathcal{TGC} \) corresponds to \( \mathcal{TC} \).

To illustrate the mapping, we give an example from \( \mathcal{S}^2(\mathcal{F}) \) to \( \mathcal{S}^0(\mathcal{G}) \) as follows.

**Example 1**. We present the mapping example for both P-g cells and T-g cells separately because the CVR graph only contains the two types of cells.

For \( p(x, y) \in \mathcal{S}^2(\mathcal{F}) \) in Figure 6a, the B-ordinates on the cells belong to the support of \( p(x, y) \) is shown in Figure 6b. In Figure 6c, \( p(x, y) \) occupies on \( \theta_0 \) is denoted as \( p(x, y)_i, i = 0, 1, \ldots, 4 \), respectively. \( \beta_0 \) is a P-cell in \( \mathcal{F} \), \( \mathcal{PD} \) denotes the P-domain of \( \theta_0 \). The T-connection \( \mathcal{TC} \) is constituted of \( \theta_1, \theta_2, \) and \( \theta_3 \). \( \mathcal{TRD} \) denotes the T-rectangle domain of \( \mathcal{TC} \). As the P-g cell \( \beta_0 \) in Figure 6d corresponds to the P-cell \( \theta_0 \) in Figure 6c, from Definition 5, we obtain

\[
\phi(p_0(x, y) : \mathcal{PD}) \rightarrow 1|_{\beta_0}.
\]
As the T-g cell $\beta_1$ in Figure 6d corresponds to the T-connection domain of $TC$ in Figure 6c and $\theta_4$ is the one-neighbor cell of $TC$, from Definition 5, we obtain

$$\phi(p_4(x, y) : T RD) \rightarrow 0|_{\beta_1}.$$ 

Similarly, we can obtain the mapping results for the other cells on the support of $p(x, y)$.

Figure 6d shows the whole mapping result.

Figure 6. A mapping example from $\mathcal{S}^2(T)$ to $\mathcal{S}^0(\mathcal{G})$. The red box in (e) denotes the T-rectangle domain. 0 and 1 in (e) denote the values on the corresponding g-cells.

3.5. The Inverse Mapping (Basis Function Construction Process)

From Theorem 3, we can construct the basis function by the basis construction process in [35]. $\mathcal{G}^k$ denotes the modified hierarchical T-mesh at level $k$, and $\mathcal{G}^k$ denotes the CVR graph of $\mathcal{G}^k$. We describe the construction process of a basis function $b_{j+1}(x, y) \in \mathcal{S}^2(\mathcal{G}^{k+1})$ as follows.

The inverse mapping is the process for constructing $b_{j+1}$, the tools for evaluating B-ordinates of each basis function of $\mathcal{S}^2(\mathcal{G}^{k+1})$ are T-structures; we recall T-structures in [35] as follows.

In Figure 7, the c-edge $v_0v_m$ and the cells $\theta_0, \ldots, \theta_n$ constitute a T-structure, which is denoted as $T$. $v_0v_m$ denotes the mid-edge of $T$, $v_0, \ldots, v_m$ denote the interior vertices of $T$, $v_0$ and $v_m$ denote the endpoints of $T$, $\theta_0$ denotes the mother cell of $T$, $\theta_5, \ldots, \theta_n$ denote the subcells of $T$, $\theta_0$ denotes the mother cell of $T$, and the level of $T$ is also the level of $\theta_0$. Additionally, $T$ should be a horizontal T-structure.

For the modified hierarchical T-mesh $\mathcal{G}^{k+1}$, a T-structure, which is denoted as $T$, consists of one c-edge $E_c$ and all the cells that have at least one common vertex with $E_c$. The interior structure of $T$ is shown in Figure 7. If there is one common interior vertex of a horizontal T-structure $T_h$ and the other vertical T-structure $T_v$, we call $T_h$ and $T_v$ connected.

A T-structure branch, which is denoted as $TSB$, consists of the union that every two T-structures are connected. The level of $TSB$ is the minimal level of all the T-structures in $TSB$. 
From the notations of T-structures and T-structure branches above, several algorithms in [35] are given to evaluate the B-ordinates for the T-cells of the T-connections, we recall the algorithms as follows.

Figure 7. The T-structure $\mathcal{T}^{k+1}$.

Algorithm 1 illustrates the pattern to order the T-structures in a T-structure branch, Algorithm 2 describes the detailed procedures for using T-structures to calculate the B-ordinates of a T-connection. By Algorithms 1 and 2, we can give Algorithm 3 to calculate the B-ordinates for a basis function $b^{k+1}(x,y) \in \mathbb{S}^2(\mathcal{T}^{k+1})$. This is different from the method of searching the domain that supports a basis function in [35]. We can choose the support of a basis function $b^k(x,y) \in \mathbb{S}^2(\mathcal{T}^{k})$ as the domain that is supports $b^{k+1}(x,y)$ directly.

**Algorithm 1:** Ordering the T-structures for each T-structure-Branch of $\mathcal{T}^{k+1}$ [35].

| Input: $\mathcal{TSB}: \{\mathcal{T}_0, \ldots, \mathcal{T}_m\}$ |
|-------------------------------------------------------------|
| **Output:** The T-structure-branch $\mathcal{TSB}_0$ after sorting |
| $\mathcal{TSB}_0 :=$ An empty T-structure-branch ; |
| $\mathcal{TSB}_0 \leftarrow$ the T-structure (T-structures) with the lowest level of $\mathcal{TSB}$ ; |
| Delete the T-structure (T-structures) of $\mathcal{TSB}_0$ from $\mathcal{TSB}$ ; |
| while $|\mathcal{TSB}| > 0$ do |
| for $i = 0, i < |\mathcal{TSB}_0|, i ++$ do |
| $V :=$ An empty T-structure set ; |
| for $j = 0, j < |\mathcal{TSB}|, j ++$ do |
| if $\mathcal{TSB}[i]$ and $\mathcal{TSB}[j]$ are connected then |
| $V \leftarrow \mathcal{TSB}[j]$ ; |
| Delete each T-structure of $V$ from $\mathcal{TSB}$ ; |
| Sort the T-structures of $V$ in descending level order ; |
| $\mathcal{TSB}_0 \leftarrow$ each T-structure of $V$ ; |
Figure 8 shows the construction process for $b^{k+1}(x, y) \in S(2, 2, 1, 1, \mathcal{G}^{k+1})$. The basis construction process can be summarized as follows.

1. **Weights initialization for $b^{k+1}(x, y)$:**

   Given the basis function $q^{k+1}(x, y) \in S_0^{k+1}(\mathcal{G}^{k+1})$ as
   \[
   q^{k+1}(x, y) = \begin{cases} 
   1, & GC \in \mathcal{G}^{k+1} \\
   0, & \text{other g-cells of } \mathcal{G}^{k+1},
   \end{cases}
   \]
by Equation (11), the weight on the domain center can be initialized as
\[
\omega = \begin{cases} 
1, & \left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}\right) \\
0, & \text{other domain-centres},
\end{cases}
\] (12)
where \(\left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}\right)\) denotes the domain center of \(D\) on \(\mathcal{T}^{k+1}\).

2. Obtain the domain \(E\) covers \(\text{Sup}(b^{k+1}(x,y))\).
As the basis cell of each basis function at level \(k+1\) is a part of the basis cell that belongs to a basis function at level \(k\), the support of \(b^{k+1}(x,y)\) is covered by a basis function \(b^k(x,y) \in \mathcal{S}^2(\mathcal{T}^k)\), we adopt the support of \(b^k(x,y)\) as \(E\).

**Figure 8.** Process for basis construction.

3. B-ordinates Calculation.
   (a) Obtain the T-structure-branches corresponding to \(E\). Obviously, \(E\) is referred to as the union of two different types of components: P-cells and T-connections. The center B-ordinate for each P-cell are obtained by Equation (12) directly. Assume there are \(n\) T-connections in \(E\), and they are \(TC_0, TC_1, \ldots, TC_n\), the T-structure-branch that covers \(TC_i\) is denoted as \(T B_i, i = 0, \ldots, n\).
   (b) Order the T-structure branches.
The center B-ordinate for each cell belongs to \(E\) can be evaluated by this T-structure branches as follows. First, we sort the T-structures in each T-structure branch by Algorithm 1, and then sort all of the T-structure branches in descending level order.
   (c) Evaluate the B-ordinates of \(b^{k+1}(x,y)\) via T-structures.
First, we adopt Algorithm 3 for calculating the center B-ordinates on T-cells of each T-connection, and then we evaluate the B-ordinates on the rest edges that are not obtained via the \(C^1\) continuous conditions.

4. Save the B-ordinates in \(\text{Sup}(b^{k+1}(x,y))\).
   If there exists nonzero B-ordinates in \(\theta \in E\), then save the B-ordinates on \(\theta\) into \(\text{Sup}(b^{k+1}(x,y))\).

We give an example to show the process for calculating the B-ordinates for a basis function \(b^{k+1}(x,y) \in \mathcal{S}^2(\mathcal{T}^{k+1})\). We illustrate the above process as follows.

**Example 2.** 1. Weights initialization for \(b^{k+1}(x,y)\). **Figure 9a** shows a part of hierarchical T-mesh. **Figure 9b** shows the corresponding part of its CVR graph. For \(q^{k+1}(x,y) \in \mathcal{S}^0(\mathcal{G}^{k+1})\), the weights of \(b^{k+1}(x,y) \in \mathcal{S}^2(\mathcal{T}^{k+1})\) are initialized by Equation (12), we show the initialized weights in **Figure 9b**.
2. Obtain the domain $E$ covers $\text{Sup}(b^{k+1}(x, y))$. We focus our attention on the part of the mesh that is shown in Figure 9b because the support of the basis function on the cells in Figure 9b covers $\text{Sup}(b^{k+1}(x, y))$.

3. B-ordinates Calculation.
   
   (a) Obtain the T-structure branches corresponding to $E$. In Figure 10, there are three T-structure branches: $T\mathcal{C}_0, T\mathcal{C}_1, \text{and } T\mathcal{C}_2$. $T\mathcal{C}_0$ is constituted of $\theta_0, \theta_1, \text{and } \theta_2$. $T\mathcal{C}_1$ is constituted of $\theta_3, \theta_4, \text{and } \theta_5$. $T\mathcal{C}_1$ corresponds to the T-structure branch $T\mathcal{S}B_i, i = 0, 1, 2$. $T\mathcal{S}B_0$, which is the blue structure in Figure 10, is constituted of $T_0$ (the c-edge $v_0v_2$ and cells adjacent to $v_0v_2$) and $T_1$ (the c-edge $v_1v_3$ and cells adjacent to $v_1v_3$); $T\mathcal{S}B_1$, which is the yellow structure in Figure 10, is constituted of $T_2$ (the c-edge $v_0v_5$ and cells adjacent to $v_0v_5$), and $T_3$ (the c-edge $v_4v_6$ and cells adjacent to $v_4v_6$). $T\mathcal{S}B_2$, which is the green structure in Figure 10, is constituted of $T_4$ (the c-edge $v_7v_8$ and cells adjacent to $v_7v_8$) and $T_5$ (the c-edge $v_8v_9$ and cells adjacent to $v_8v_9$).

   (b) Order the T-structure branches. The level of $T_0$ is lower than the level of $T_1$. Thus, we can order the T-structures in $T\mathcal{S}B_0$ as $T_0, T_1$. Similarly, the T-structures in $T\mathcal{S}B_1$ can be ordered as $T_2, T_3$, the T-structures in $T\mathcal{S}B_2$ can be ordered as $T_4, T_5$.

   (c) Evaluate the B-ordinates of $b^{k+1}(x, y)$ via T-structures. Figure 11 shows the process of using the T-structure-branches that are ordered in Figure 3b to calculate B-ordinates of $b^{k+1}(x, y)$. Figure 12 shows the rest B-ordinates obtained by the $C^1$ continuous conditions.

4. Save the B-ordinates of $\text{Sup}(b^{k+1}(x, y))$. The support of $b^{k+1}(x, y)$ is the cells without the top left cell.

Figure 9. Weights initialization. The 0 and 1 in (a) denote the value on each g-cell that belongs to a part of $\mathcal{S}^k$; The 0 and 1 in (b) denote the corresponding initialized weights.
Figure 10. The T-structure-branches of $\mathcal{E}$. The blue structure denotes T-structure-branch that is constituted of $T_0$ (the c-edge $v_0v_2$ and cells adjacent to $v_0v_2$) and $T_1$ (the c-edge $v_1v_3$ and cells adjacent to $v_1v_3$); The yellow structure denotes T-structure-branch that is constituted of $T_2$ (the c-edge $v_0v_5$ and cells adjacent to $v_0v_5$), and $T_3$ (the c-edge $v_4v_6$ and cells adjacent to $v_4v_6$); The green structure denotes T-structure-branch that is constituted of $T_4$ (the c-edge $v_7v_8$ and cells adjacent to $v_7v_8$) and $T_5$ (the c-edge $v_8v_9$ and cells adjacent to $v_8v_9$).

(a) The B-ordinates calculated by $T_0$ of $TSB_0$

(b) The B-ordinates calculated by $T_1$ of $TSB_0$

Figure 11. Cont.
The B-ordinates calculated by $T_2$ of $TSB_1$ (c)

The B-ordinates calculated by $T_3$ of $TSB_1$ (d)

The B-ordinates calculated by $T_4$ of $TSB_3$ (e)

The B-ordinates calculated by $T_5$ of $TSB_3$ (f)

**Figure 11.** The process of B-ordinates calculation. The values in the highlight domain are the new evaluated B-ordinates on the corresponding domain points.

**Figure 12.** The rest B-ordinates. The values in the highlight domain are the new evaluated B-ordinates on the corresponding domain points.

For the basis function whose basis cell is a P-cell, we give the construction method above.
4. Basis Construction Framework with Modifications

In fact, there are two types of basis cells for the basis functions of \( S^2(\mathcal{T}^{k+1}) \). One corresponds to a P-cell on \( \mathcal{T}^{k+1} \); we can construct such basis functions by the method in Section 3.5. The other corresponds to a T-connection on \( \mathcal{T}^{k+1} \), and we discuss the modification process of such basis function in this section. In addition, we introduce the whole basis construction framework with modifications in this section.

4.1. Basis Modification Method

First, we define basis modification as follows.

**Definition 6.** Given a basis function \( b^k_i(x, y) \) at level \( k \), the set consists of the cells of \( \Theta_k \) that belongs to \( \text{Sup}(b^k_i(x, y)) \) as \( \Theta_k \). For each cell \( \theta \) of \( \Theta_k \), modify \( b^k_i(x, y) \) from level \( k \) to level \( k + 1 \), the modified basis function can be denoted as

\[
b^k_i(x, y) = \mathfrak{M}^{k+1}(b^k_i(x, y)).
\]

The modification process can be described as

1. using 9 B-ordinates to represent \( b^k_i(x, y) \) over \( \theta \);
2. adopting de Casteljau’s algorithm for generating new B-ordinates on \( \theta \) according to the different subdivision types; and
3. modifying the B-ordinates that are associated with \( b^k_i(x, y) \) by Algorithm 4.

**Algorithm 4: Modification of \( b^k_i(x, y) \)**

```
Input: \( b^k_i(x, y) \in S^2(\mathcal{T}^k) \) and \( \Theta_k \)
Output: \( b^k_i(x, y) \)
1. Subdivide the B-ordinates according to the different subdivision type;
2. Obtain \( \text{Sup}(b^k_i(x, y)) \in \mathcal{T}^{k+1} \);
3. Let \( \Xi_k \) be the set consisting of all new domains that belong to \( \text{Sup}(b^k_i(x, y)) \);
4. \( V \leftarrow \) the crossing vertices of \( b^k_i(x, y) \);
5. for \( i = 0; i < |V|; i++ \) do
6. if \( V[i] \) is a conner vertex of one T-connection in \( \text{Sup}(b^k_i(x, y)) \) then
7. if the cells around \( V[i] \) are not subdivided then
8. Reserve the 16 B-ordinates around \( V[i] \);
9. else
10. if the four edges on \( V[i] \) are either the edges or the c-edges of \( \mathcal{T}^k \) then
11. Reserve the 16 B-ordinates around \( V[i] \);
12. Reset all of the weights on domain-centres (without basis-centre of \( b^k_i(x, y) \)) of \( \Xi_k \) to zero;
13. Calculate the rest B-ordinates of \( b^k_i(x, y) \) via T-structure branches;
14. Reset the support of \( b^k_i(x, y) \) \( \in S^2(\mathcal{T}^{k+1}) \) on \( \mathcal{T}^{k+1} \);
```

To demonstrate the basis modification in Definition 6, we provide the following example.

**Example 3.** Figure 13a shows a basis function \( b^k_i(x, y) \in S^2(\mathcal{T}^k) \), the B-ordinates on \( b^k_i(x, y) \) are shown in Figure 13b. First, refine the cells of \( \Theta_k \), according to the different subdivision type, Figure 14 shows the refined mesh. Subdivide the B-ordinates according to the different subdivision types. The B-ordinates subdivision result is also shown in Figure 14.

In Figure 15a, the crossing vertex, which is denoted as the blue square, is subdivided. As the crossing vertex is a corner vertex of a connection, and the four edges are either the edges or the
c-edges of $\mathcal{T}^k$, we can reserve the 16 B-ordinates around the crossing vertex. The subdivision will not influence the B-ordinates on the domain points whose B-ordinates are zero, we can reserve the B-ordinates (B-ordinates in the gray domain) on the crossing vertices in Figure 15a. Then, we need to calculate the B-ordinates on the black dots in Figure 15a. Additionally, we reset the weights on the domain centres in Figure 15b. At last, we can calculate the rest of the B-ordinates on the yellow domain in Figure 15c via Algorithm 3, the modified basis function is shown in Figure 15d. From above, we modify a basis function $b_k^i(x, y) \in S^2(\mathcal{T}^k)$ to $\bar{b}_k^i(s, t) \in S^2(\mathcal{T}^{k+1})$.

(a) $b_k^i(x, y)$

Figure 13. A basis function at level $k$.

(b) The B-ordinates of $b_k^i(x, y)$

Figure 14. The subdivided B-ordinates of $b_k^i(x, y)$. 
Figure 15. Basis modification. The values in the highlight domain of (a) are the reserved B-ordinates. ■ in (a) denotes the crossing vertex. 0 and 1 in (b) denotes the reset weights. The values in the highlight domain of (c) are the new evaluated B-ordinates on the domain points that corresponds to the “●” in (a).

Algorithm 4 gives the process for basis modification from $\mathcal{S}^2(\mathcal{T}^k)$ to $\mathcal{S}^2(\mathcal{T}^{k+1})$, without loss of generality. If there are more than one basis cells at level $k + 1$ appearing in a basis function of $\mathcal{S}^2(\mathcal{T}^k)$, the modification will display similarly.
4.2. Basis Construction Framework

From the above, we give the basis construction and modification framework by Algorithm 5.

**Algorithm 5:** Basis construction and modification of \( \mathbb{S}^2(\mathcal{F}^{k+1}) \).

**Input:** The set \( B_k \) that is constituted of all the basis functions of \( \mathbb{S}^2(\mathcal{F}^k) \); \( \Theta \)

1; **Output:** The set \( B_{k+1} \) that is constituted of all the basis functions of \( \mathbb{S}^2(\mathcal{F}^{k+1}) \)

2; Obtain \( \mathcal{F}^{k+1} \) by refinement the cells of \( \Theta \) according to the different subdivision type;

3 \( CVR^{k+1} \) ← All the basis cells on \( \mathcal{F}^{k+1} \);

4 \( B^k \) ← All the basis functions of \( \mathbb{S}^2(\mathcal{F}^k) \);

5 **for** int \( i = 0; i < |CVR^{k+1}|; i++ \)** do

6 if \( Sup(B^k[i]) \cap \Theta = \emptyset \) then

7 \( B^{k+1} \leftarrow B^k[i] \);

8 Delete the basis cell of \( B^k[i] \) from \( CVR^{k+1} \);

9 else

10 Modify \( B^k[i] \) as \( \mathbb{M}^{k+1}(B^k[i]) \) via Algorithm 4;

11 \( B^{k+1} \leftarrow \mathbb{M}^{k+1}(B^k[i]) \);

12 Delete the basis cell of \( \mathbb{M}^{k+1}(B^k[i]) \) from \( CVR^{k+1} \);

13 **for** int \( i = 0; i < |CVR^{k+1}|; i++ \)** do

14 if The \( CVR^{k+1}[i] \) corresponds to a P-cell of \( \mathcal{F}^{k+1} \) then

15 Construction the basis function \( B^{k+1}[i] \) via Algorithm 3;

16 \( B^{k+1} \leftarrow B^{k+1}[i] \);

In this section, we present the algorithm for basis modification and discuss the framework for basis construction with modification.

5. Fitting Open Meshes

Surface fitting is an essential method in the field of CAD. In this section, we use several triangular surface models to show the potential of splines generated by our method.

Given the open surface model \( \mathcal{M} \) in 3D space, \( \mathcal{M} \) is constituted of a series of triangles, the vertices on \( \mathcal{M} \) are denoted as \( V_0, V_1, \ldots, V_N \), we parameterize \( \mathcal{M} \) by using the parametrization method in [38]. The parameter mesh is a triangle mesh. The parameter value corresponding to \( V_i \) is denoted as \( V'_i(x_i, y_i), i = 0, \ldots, N \), and the parameter domain is \([0,1] \times [0,1]\). Additionally, the triangle on the parameterized mesh is denoted as \( \Delta \).

5.1. Calculation of Control Points

Different from the least square optimization method in [24,31], we adopt the domain centres for calculating the control points as follows.

Given the basis functions \( b_j(x, y), j = 1, \ldots, m \), let \([x_{in}, x_{in+1}] \times [y_{in}, y_{in+1}] \) be the domain corresponding to basis cell of \( b_j(x, y), j = 1, \ldots, m \), and \((x_m, y_m) = (x_{in} + x_{in+1}, y_{in} + y_{in+1}) / 2 \) denotes the domain centre of \([x_{in}, x_{in+1}] \times [y_{in}, y_{in+1}] \). We calculate the control points by solving the following linear system

\[
S(x_k, y_k) = V'_k, k = 1, 2, \ldots, m.
\]

(13)

Here, \((x_k, y_k) \in \Delta_1: ((x_{k1}, y_{k1}), (x_{k2}, y_{k2}), (x_{k3}, y_{k3})), V'_k = w_{k1}V_{k1} + w_{k2}V_{k2} + w_{k3}V_{k3}, \)
and \(w_{k1}, w_{k2}, w_{k3} \) can be evaluated by \((x_k, y_k) = w_{k1}(x_{k1}, y_{k1}) + w_{k2}(x_{k2}, y_{k2}) + w_{k3}(x_{k3}, y_{k3})\).
5.2. Process of Fitting

With the method of calculating control points, we run the loop of steps 2, steps 3, and steps 4, until the error reaches the given tolerance $\varepsilon$.

1. Given the $4 \times 4$ tensor-product mesh $\mathcal{P}^0$.
2. For the old basis functions, we retain their control points; for the new basis functions of $S^2(\mathcal{P}^k)$, calculate their control points via Section 5.1. The result modified spline space is denoted as $S(x, y)$.
3. $\Theta_k \leftarrow \theta, \text{if} \ max_{(x,y)\in\theta} = \| V(x, y) - S(x, y) \| > \varepsilon$.
4. We use mesh-refinement rules in Section 3.3 to denote the cells in $\Theta_k$. We subdivide the mesh and modify the basis functions via Algorithm 5.

$k \leftarrow k + 1$.

Figures 16–19 illustrate four examples, in which basis functions of spline space over hierarchical T-meshes and modified hierarchical T-meshes are shown. Both basis functions over the two types of meshes provide a good approximation of the original model. However, the number of subdivided cells over modified hierarchical T-meshes are lower than that over hierarchical T-meshes at the same level. The statistical data for comparison are listed in Table 2. The comparisons for the solutions on hierarchical T-meshes and modified hierarchical T-meshes are provided in Figure 20. The comparison shows that the solution on the modified hierarchical T-meshes can achieve similar accuracy with fewer subdivision cells.

Figure 16. Balls.
Figure 17. Bunny head.

(a) Original model
(b) Result of basis functions over hierarchical T-mesh
(c) Result of basis functions over modified hierarchical T-mesh
(d) Result of (b) with hierarchical T-mesh
(e) Result of (c) with modified hierarchical T-mesh

Figure 18. Horse head.

(a) Original model
(b) Result of basis functions over modified hierarchical T-mesh
(c) Result of basis functions over modified hierarchical T-mesh
(d) Result of (b) with hierarchical T-mesh
(e) Result of (c) with modified hierarchical T-mesh
Figure 19. Venus body.

Table 2. Experiment data.

| Hierarchical T-Meshes | Models | Levels | #Subdivision Cells | #Max Error | Modified Hierarchical T-Meshes | #Subdivision Cells | #Max Error |
|-----------------------|--------|--------|--------------------|------------|-------------------------------|--------------------|------------|
| Balls                 | Balls  | 1      | 16                 | 0.169909   | 16                            | 0.169909           |
|                       |        | 3      | 144                | 0.0840613  | 139                           | 0.0840613          |
|                       |        | 5      | 195                | 0.0550255  | 170                           | 0.0550255          |
|                       |        | 7      | 225                | 0.0071944  | 192                           | 0.0071944          |
| Bunny head            | Bunny head | 1 | 16                 | 0.237579   | 16                            | 0.237579           |
|                       |        | 3      | 166                | 0.084061   | 151                           | 0.084061           |
|                       |        | 5      | 358                | 0.161495   | 290                           | 0.159583           |
|                       |        | 7      | 547                | 0.048042   | 433                           | 0.048038           |
|                       |        | 9      | 691                | 0.007814   | 526                           | 0.009279           |
| Horse head            | Horse head | 1 | 16                 | 0.4413     | 16                            | 0.4413             |
|                       |        | 3      | 256                | 0.3082     | 256                           | 0.3082             |
|                       |        | 5      | 1309               | 0.1405     | 1174                          | 0.1405             |
|                       |        | 7      | 2380               | 0.0885     | 2148                          | 0.0885             |
|                       |        | 9      | 3454               | 0.0295     | 3063                          | 0.0295             |
|                       |        | 11     | 3829               | 0.0093     | 3380                          | 0.0093             |
|                       |        | 12     | 3958               | 0.0014     | 3477                          | 0.0014             |
Table 2. Cont.

| Models      | Levels | #Subdivision Cells | #Max Error | Levels | #Subdivision Cells | #Max Error |
|-------------|--------|--------------------|------------|--------|--------------------|------------|
| Venus body  | 1      | 16                 | 0.596      | 1      | 16                 | 0.596      |
|             | 3      | 256                | 0.401      | 3      | 256                | 0.401      |
|             | 5      | 1231               | 0.1521     | 5      | 1129               | 0.1521     |
|             | 7      | 1915               | 0.0755     | 7      | 1708               | 0.0755     |
|             | 9      | 2830               | 0.0199     | 9      | 2501               | 0.0199     |
|             | 11     | 3490               | 0.0064     | 11     | 3055               | 0.0064     |
|             | 12     | 3613               | 0.003      | 12     | 3127               | 0.004      |

Figure 20. Convergence results of basis functions over hierarchical T-meshes and modified hierarchical T-meshes.

6. Conclusions

In this paper, we provide a unique basis construction framework that incorporates basis modification of the spline space over modified hierarchical T-meshes. The subdivision rules on the modified hierarchical T-meshes are also given to prevent the redundant edges that exist on hierarchical T-meshes. Algorithms within the framework make the construction of basis functions easy to operate. In the basis construction framework, the splines over the modified hierarchical T-mesh are modified to accommodate the insertion of the cells on the corresponding CVR graph. The modification process is driven as follows: for each cell belongs to $\Theta_k$, it is subdivided into four subcells or two subcells on $T_{k+1}$. For each basis function at level $k$, if some of the corresponding CVR cells are subdivided, the basis function will be modified as a basis function of level $k+1$. If a basis cell is generated by the subdivision process, we will reconstruct the corresponding basis function. Moreover, the application of the above splines in surface reconstruction with adaptive refinement.
is discussed in detail. In comparison to splines over hierarchical T-meshes, modified hierarchical T-meshes have fewer cells subdivided when achieving similar accuracy.

**Author Contributions:** Conceptualization, J.L. and L.Z.; methodology, L.Z.; software, J.L.; validation, J.L. and W.Z.; writing—original draft preparation, J.L. and W.Z.; writing—review and editing, J.L. and W.Z.; project administration, L.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** National Natural Science Foundation of China (No.: 61972131), National key research and development project (No.2018YFB2100301) and Research project of Hefei Normal University (No.2021KJZD20).

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Acknowledgments:** We would like to thank the anonymous referees for providing us with constructive comments and suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Farin, G.; Hoschek, J.; Kim, M.S. *Handbook of Computer Aided Geometric Design*; Elsevier: Amsterdam, The Netherlands, 2002.
2. Hughes, T.; Cottrell, J.; Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* 2005, 194, 4135–4195. [CrossRef]
3. Sederberg, T.W.; Zheng, J.; Bakmoy, A.; Nasri, A. T-Splines and T-NURCCs. *ACM Trans. Graph.* 2003, 22, 477–484. [CrossRef]
4. Sederberg, T.W.; Cardon, D.L.; Finnigan, G.T.; North, N.S.; Zheng, J.; Lyche, T. T-Spline Simplification and Local Refinement. *ACM Trans. Graph.* 2004, 23, 276–283. [CrossRef]
5. Bazilevs, Y.; Calo, V.; Cottrell, J.; Evans, J.; Hughes, T.; Lipton, S.; Scott, M.; Sederberg, T. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Eng.* 2009, 199, 229–263. [CrossRef]
6. Dürr, M.R.; Jüttler, B.; Simeon, B. Adaptive isogeometric analysis by local h-refinement with T-splines. *Comput. Methods Appl. Mech. Eng.* 2010, 199, 264–275. [CrossRef]
7. Li, X.; Zheng, J.; Sederberg, T.W.; Hughes, T.; Scott, M.A. On linear independence of T-spline blending functions. *Comput. Aided Geom. Des.* 2012, 29, 63–76. [CrossRef]
8. Scott, M.; Li, X.; Sederberg, T.; Hughes, T. Local refinement of analysis-suitable T-splines. *Comput. Methods Appl. Mech. Eng.* 2012, 213–216, 206–222. [CrossRef]
9. Li, X.; Zhang, J. AS++ T-splines: Linear independence and approximation. *Comput. Methods Appl. Mech. Eng.* 2018, 333, 462–474. [CrossRef]
10. Forsey, D.R.; Bartels, R.H. Hierarchical B-Spline Refinement. In Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’88, Atlanta, GA, USA, 1–5 August 1988; pp. 205–212.
11. Unther Greiner, G.; Hormann, K. Interpolating and Approximating Scattered 3D-Data with Hierarchical Tensor Product B-Splines; Vanderbilt University Press: Nashville, TN, USA, 1997; pp. 163–172.
12. Schillinger, D.; Dedé, L.; Scott, M.A.; Evans, J.A.; Borden, M.J.; Rank, E.; Hughes, T.J. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Methods Appl. Mech. Eng.* 2012, 249–252, 116–150. [CrossRef]
13. Vuong, A.V.; Giannelli, C.; Jüttler, B.; Simeon, B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* 2011, 200, 3554–3567. [CrossRef]
14. Evans, E.; Scott, M.; Li, X.; Thomas, D. Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* 2015, 284, 1–20. [CrossRef]
15. Giannelli, C.; Jüttler, B.; Speleers, H. THB-splines: The truncated basis for hierarchical splines. *Comput. Aided Geom. Des.* 2012, 29, 485–498. [CrossRef]
16. Giannelli, C.; Jüttler, B.; Speleers, H. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.* 2014, 40, 459–490. [CrossRef]
17. Giannelli, C.; Jüttler, B.; Kleiss, S.K.; Mantzaflaris, A.; Simeon, B.; Špeh, J. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comput. Methods Appl. Mech. Eng.* 2016, 299, 337–365. [CrossRef]
18. Kiss, G.; Giannelli, C.; Zore, U.; Jüttler, B.; Großmann, D.; Barner, J. Adaptive CAD model (re-)construction with THB-splines. *Graph. Model.* 2014, 76, 273–288. [CrossRef]
19. Speleers, H.; Manni, C. Effortless quasi-interpolation in hierarchical spaces. *Numer. Math.* 2016, 132, 155–184. [CrossRef]
20. Dokken, T.; Lyche, T.; Pettersen, K.F. Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Des.* 2013, 30, 331–356. [CrossRef]
21. Bressan, A.; Jüttler, B. A hierarchical construction of LR meshes in 2D. *Comput. Aided Geom. Des.* 2015, 37, 9–24. [CrossRef]
22. Bressan, A. Some properties of LR-splines. *Comput. Aided Geom. Des.* **2013**, *30*, 778–794. [CrossRef]
23. Deng, J.; Chen, F.; Feng, Y. Dimensions of spline spaces over T-meshes. *J. Comput. Appl. Math.* **2006**, *194*, 267–283. [CrossRef]
24. Deng, J.; Chen, F.; Li, X.; Hu, C.; Tong, W.; Yang, Z.; Feng, Y. Polynomial splines over hierarchical T-meshes. *Graph. Model.* **2008**, *70*, 76–86. [CrossRef]
25. Li, X.; Deng, J.; Chen, F. Surface modeling with polynomial splines over hierarchical T-meshes. *Vis. Comput.* **2007**, *23*, 1027–1033. [CrossRef]
26. Wang, J.; Yang, Z.; Jin, L.; Deng, J.; Chen, F. Adaptive surface reconstruction based on implicit PHT-splines. In Proceedings of the 14th ACM Symposium on Solid and Physical Modeling, Haifa, Israel, 1–3 September 2010; pp. 101–110.
27. Tian, L.; Chen, F.; Du, Q. Adaptive finite element methods for elliptic equations over hierarchical T-meshes. *J. Comput. Appl. Math.* **2011**, *236*, 878–891. [CrossRef]
28. Nguyen-Thanh, N.; Nguyen-Xuan, H.; Bordas, S.P.A.; Rabczuk, T. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1892–1908. [CrossRef]
29. Nguyen-Thanh, N.; Kiendl, J.; Nguyen-Xuan, H.; Wüchner, R.; Bletzinger, K.U.; Bazilevs, Y.; Rabczuk, T. Rotation free isogeometric thin shell analysis using PHT-splines. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 3410–3424. [CrossRef]
30. Wang, P.; Xu, J.; Deng, J.; Chen, F. Adaptive isogeometric analysis using rational PHT-splines. *Comput.-Aided Des.* **2011**, *43*, 1438–1448. [CrossRef]
31. Ni, Q.; Wang, X.; Deng, J. Modified PHT-splines. *Comput. Aided Geom. Des.* **2019**, *73*, 37–53. [CrossRef]
32. Deng, J.; Chen, F.; Jin, L. Dimensions of biquadratic spline spaces over T-meshes. *J. Comput. Appl. Math.* **2013**, *238*, 68–94. [CrossRef]
33. Deng, F.; Zeng, C.; Wu, M.; Deng, J. Bases of Biquadratic Polynomial Spline Spaces over Hierarchical T-Meshes. *J. Comput. Math.* **2017**, *35*, 91–120. [CrossRef]
34. Zeng, C.; Deng, F.; Deng, J. Bicubic hierarchical B-splines: Dimensions, completeness, and bases. *Comput. Aided Geom. Des.* **2015**, *38*, 1–23. [CrossRef]
35. Liu, J.; Deng, E.; Deng, J. Space Mapping of Spline Spaces over Hierarchical T-meshes. *arXiv* **2022**, arXiv:2103.11688.
36. Zeng, C.; Deng, F.; Li, X.; Deng, J. Dimensions of biquadratic and bicubic spline spaces over hierarchical T-meshes. *J. Comput. Appl. Math.* **2015**, *287*, 162–178. [CrossRef]
37. Wu, M.; Deng, J.; Chen, F. Dimension of spline spaces with highest order smoothness over hierarchical T-meshes. *Comput. Aided Geom. Des.* **2013**, *30*, 20–34. [CrossRef]
38. Floater, M.S. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des.* **1997**, *14*, 231–250. [CrossRef]