AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(11): 26767–26802.
DOI: 10.3934/math.20251177
Received: 13 August 2025
Revised: 30 October 2025
Accepted: 05 November 2025
Published: 18 November 2025

Research article

Adaptive fitting with a cubic spline over planar hierarchical quadrilateral meshes

Pengxiao Wang[1] **and Chongjun Li**[2,*]

[1] School of Physical and Mathematical Sciences, Nanjing Tech University, Nanjing 211816, China
[2] School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China

* **Correspondence:** Email: chongjun@dlut.edu.cn.

Abstract: Automatic fitting techniques are required in many industrial applications, for example instrument calibration, data analysis, geometric modeling, and reverse engineering. In this paper, we present a surface construction algorithm for a cubic spline over planar hierarchical quadrilateral meshes. The surface was piecewisely constructed by interpolating the cubic spline surface of the 12 parameters at four vertices on each quadrilateral cell of the hierarchical quadrilateral mesh. For a given hierarchical quadrilateral mesh and geometric information (function values and two first-order partial derivatives) at the corresponding basis vertices of the hierarchical quadrilateral mesh, the surface can be constructed simply. Moreover, we give an adaptively refined surface algorithm for fitting scattered data points based on cubic spline surface construction. The numerical results show that the proposed adaptive algorithm is efficient in fitting scattered data points within a polygonal domain.

Keywords: scattered data fitting; the B-net method; hierarchical quadrilateral mesh; triangulated quadrangulation; surface reconstruction
Mathematics Subject Classification: 65D07, 65D17

1. Introduction

In geometric modeling, the representation of surfaces is one of the most important and interesting research topics. The non-uniform rational B-spline (NURBS) is commonly used in computer-aided design (CAD), computer aided engineering (CAE), and computer-aided manufacturing (CAM) [1]. However, NURBS has the weakness that the control points must lie topologically on a rectangular grid, since NURBS is based on a tensor product structure. Moreover, NURBS models contain a large number of superfluous control points, which are big burdens to modeling systems.

Locally refinable splines are introduced in geometric design and isogeometric analysis to overcome the weakness of B-splines or NURBS in tensor-product form. By allowing T-junctions in the control meshes, T-splines were introduced to remove the redundant control points in geometric modeling [2,3]. The potential of T-splines in isogeometric analysis was reported in [4,5]. To fix the linear dependency problem of T-splines, AST-splines and AS++ T-splines were introduced in [6, 7]. As an extension of tensor-product representations, truncated hierarchical B-splines (THB-splines), which form a convex partition of unity, were introduced by Giannelli et al. in [8,9]. Recently, Pan et al. [10–12] showed how to generate hierarchical T-meshes with associated locally refined B-splines, which possess the property of local linear independence, form a non-negative partition of unity, and span the resulting spaces of C^s smooth polynomial splines of degree $p = 2s + 1$. They are called RMB-splines, which are also a type of locally refined spline. Pan [13] also proposed surface reconstruction based on implicit hierarchical B-splines.

Polynomial splines over hierarchical T-meshes (PHT-splines) were first introduced by Deng et al. [14] in 2008. Theoretic study on the dimension of spline spaces over T-meshes can be found in [15–19]. The basis functions of PHT-splines posses many nice properties like B-spline basis functions, such as linear independency, partition of unity, compact support, non-negativity, etc. Most of all, they have a perfect local refinement property, which makes PHT-splines very useful in adaptive surface modeling. They have been applied in solving partial differential equations, iso-geometric analysis, surface stitching, and implicit surface reconstructions [20–23].

Compared with globally subdivided meshes, hierarchical meshes can decrease the number of unknown quantities of the calculation model while maintaining the accuracy, therefore reducing the demands on the computer system and enhancing the calculation efficiency. Consequently, the hierarchical mesh structure has strong practicality and the PHT-spline could be a useful tool in many areas such as surface modeling [24].

However, computing the basis functions of PHT-splines becomes increasingly complex as the level increases, and they are precomputed and stored in Bézier forms beforehand [14]. Since an evaluation algorithm is fundamental for PHT-spline applications, the memory consumption is very large for models with refined details. Wang et al. [25] proposed a de Boor-like algorithm to evaluate a PHT-spline surface at a given parametric point. This evaluation algorithm basically requires computing the control points of a local tensor-product bicubic B-spline surface at each level, and its computational cost increases with the level. Besides, during the refinement on a T-mesh, a single rectangular cell may be influenced by more than 16 PHT-spline basis functions.

Notice that not all surfaces are suitable to be represented by rectangular surface patches, e.g., for a star model or a V-notched crack model. In this paper, we focus on arbitrary quadrilateral meshes whose valences of the internal vertices can be any value. Compared with original rectangular meshes whose valences of internal vertices are 4, quadrilateral meshes with arbitrary valence are more flexible. This flexibility allows the surface modeling with relatively complicated geometries.

Splines with small polynomial degree could be of interest in many applications. In this paper, we present a new surface construction algorithm for cubic spline surfaces with C^1 smoothness over planar hierarchical quadrilateral meshes. Similar to the finite element method, rather than constructing the basis functions globally, we first construct the basis functions on each cell of the hierarchical meshes and then we find a way to achieve the smoothness between all adjacent cells of the hierarchical meshes. This way of constructing splines over hierarchical meshes can efficiently avoid the storage and calculation problems caused by the complexity of PHT-splines' basis functions during subdivision.

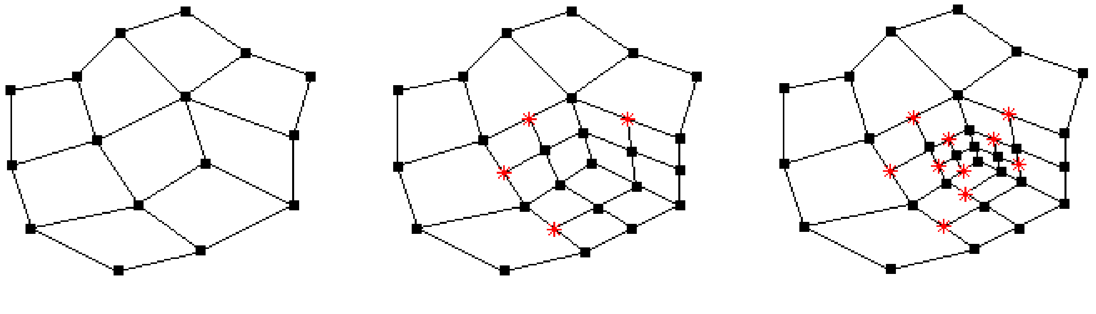
Specifically, we establish Hermite interpolation basis functions of each quadrilateral cell of the planar hierarchical quadrilateral mesh. Through the geometric information transition matrix M, the global smoothness between cells is established. Transferring the geometric information via "points" is the key for constructing splines over planar hierarchical meshes, and the implementation of this algorithm is just a calculation, which is simple and efficient. By this way of transferring the geometric information, new results of splines over planar hierarchical meshes are obtained, which hold theoretical significance in multivariate splines and will also facilitate the integration of engineering CAD design and finite element analysis technology.

Moreover, we give an adaptively fitting algorithm for scattered data points based on the cubic spline surfaces' construction. In this algorithm, the quadrilateral mesh can achieve local refinement adaptively. Furthermore, examples in Section 5 show that our surface fitting algorithm has high precision, low mesh distortion sensitivity, and is suitable for data processing and engineering applications.

The remainder of this paper is organized as follows. We introduce some preliminary knowledge on planar hierarchical quadrilateral mesh in Section 2. In Section 3, we construct a cubic spline surface over planar hierarchical quadrilateral mesh. In Section 4, an adaptive surface reconstruction method for scattered data points by the cubic spline surfaces over hierarchical quadrilateral meshes is given. In Section 5, several numerical experiments unveil that the proposed algorithm can produce high-quality approximation surfaces. Finally, conclusions are drawn in the last section.

2. Planar hierarchical quadrilateral mesh

A planar hierarchical quadrilateral mesh is a quadrilateral grid that allows T-junctions on a 2D plane. It is a special type of quadrilateral grid that has a natural level structure and it is defined recursively. Figure 1 illustrates the process of generating a hierarchical quadrilateral mesh. Generally, we start from an original quadrilateral mesh Q_0, in which the valences of interior vertices can be different. For example, the valences of interior vertices occur as 3, 4, and 5 in Figure 1(a). The elements (including vertices, edges, and cells) of quadrilateral mesh are called the level 0 elements. From level k to level $k+1$, some cells at level k are subdivided into 2×2 uniform subcells, where the new vertices, the new edges, and the new cells are of level $k+1$, and the resulting quadrilateral mesh is called the level $k+1$ quadrilateral mesh, denoted as Q_{k+1}.



(a) Level 0

(b) Level 1

(c) Level 2

Figure 1. A hierarchical quadrilateral mesh's generation (the black squares are basis vertices and the red stars are non-basis vertices).

Similar to hierarchical T-mesh as in [14], the vertices of hierarchical quadrilateral mesh are divided into two kinds: the basis vertex and non-basis vertex. For each vertex of hierarchical quadrilateral mesh, if it is a corner point of all adjacent quadrilateral cells containing it, then it is called a basis vertex; Otherwise, it is a non-basis vertex. Specially, boundary vertices are basis vertices.

If there is a cell at level $k+1$, it must have its father cell at level k. Since a hierarchical quadrilateral mesh is defined recursively, we can find a series of father cells. This property is the same with hierarchical T-mesh.

3. Cubic spline surfaces over hierarchical quadrilateral meshes

The Fraeijs de Veubeke-Sanders (FVS or reduced FVS) element is a classical C^1 (continuously differentiable) finite element employed in the finite element method. The FVS (or reduced FVS) element and its related properties are introduced in [26–28]. In [29], we represented the 12-parameter quadrilateral spline bases by employing the B-net method, which are more convenient for application and calculation. Actually, they are the same as the FVS and the reduced FVS elements.

In this section, we first review on the 12-parameter quadrilateral spline bases presented by the B-net method in [29]. Subsequently, based on the 12-parameter quadrilateral spline bases by the B-net method, we propose a new surface construction algorithm for cubic spline surfaces over hierarchical quadrilateral meshes.

3.1. The B-net method

The B-net method is an important tool for studying the bivariate splines defined on triangulations [30,31]. It is originated from the Bernstein polynomials, and is based on the triangular area coordinates. The computation on derivatives, integrals, and products of the spline functions can be simplified greatly by using their Bézier coefficients on each triangle cell.

There are $(n+2)(n+1)/2$ domain points $\xi_{i,j,k}$ equally located in the triangle $\triangle P_1P_2P_3$ with the area coordinates $(i/n, j/n, k/n)$ (as shown in Figure 2 when $n=3$). The Bernstein polynomials of degree n over the triangle $\triangle P_1P_2P_3$ are defined by

$$B^n_{i,j,k}(\lambda_1,\lambda_2,\lambda_3)=\frac{n!}{i!j!k!}\lambda_1^i\lambda_2^j\lambda_3^k,\ i+j+k=n,\lambda_1,\lambda_2,\lambda_3\geq 0,\ \lambda_1+\lambda_2+\lambda_3=1, \tag{3.1}$$

where $\lambda_1,\lambda_2,\lambda_3$ are the area coordinates. It is easy to know that all the Bernstein polynomials of degree n are linearly independent and they form a basis of $\mathbb{P}_n$. In addition, they satisfy a partition of unity, i.e.,

$$\sum_{i+j+k=n} B^n_{i,j,k}(\lambda_1,\lambda_2,\lambda_3)=(\lambda_1+\lambda_2+\lambda_3)^n=1. \tag{3.2}$$

Hence, a polynomial of degree n can be expressed in the B-net form as

$$p(x,y)=f(\lambda_1,\lambda_2,\lambda_3)=\sum_{i+j+k=n} b_{i,j,k}B^n_{i,j,k}(\lambda_1,\lambda_2,\lambda_3), \tag{3.3}$$

where $b_{i,j,k}$ are the corresponding Bézier coefficients.

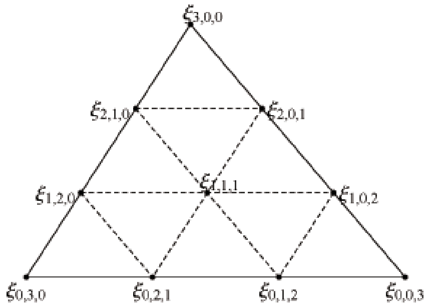
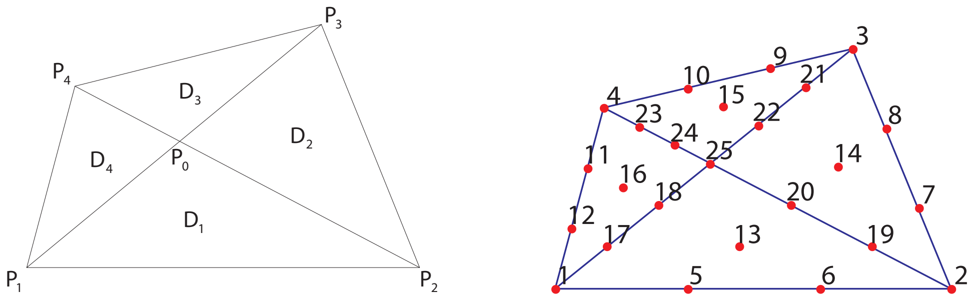


Figure 2. The B-net domain points of degree 3 in a triangle.

For a convex quadrangle, as shown in Figure 3(a), denote the corner nodes by P_1, P_2, P_3, P_4, denote the two diagonals of the quadrilateral by $\overline{P_1P_3}$, $\overline{P_2P_4}$, and denote the intersection of two diagonals $\overline{P_1P_3}$ and $\overline{P_2P_4}$ by P_0. The quadrangle is divided into four subtriangles $D_1, \ldots, D_4$. This triangulated quadrangulation is called FVS triangulation since it was used by Fraeijs de Veubeke and Sander to construct a 16-parameter plate element [26, 27].



(a) (b)

Figure 3. A convex triangulated quadrangle and its 25 domian points of degree 3.

Denote Δ by the above triangulation of region $D_i (i = 1, 2, 3, 4)$, and then a spline space on Δ is defined by

$$S_d^r(\Delta) = \{s \in C^r(\Delta) : s\,|_{D_i} \in \mathbb{P}_d, i = 1, 2, 3, 4\}. \tag{3.4}$$

This means that a spline function in $S_d^r(\Delta)$ is a piecewise polynomial of degree d, and is C^r continuous on the two diagonals $\overline{P_1P_3}$ and $\overline{P_2P_4}$.

3.2. The 16-parameter quadrilateral spline bases

In the triangulated quadrilateral partition of Figure 3(b), the B-net points of the four triangles are reassigned with a unified numbering $b_1, b_2, \ldots, b_{25}$, replacing the previous $b_{i,j,k}$ indices. That is, there are 25 domain points in the triangulated quadrangle, whose indexes are shown in Figure 3(b). The corresponding Bézier coefficients are simply denoted by $b_1, b_2, \ldots, b_{25}$. For the cubic spline space $S_3^1(\Delta)$, the C^1 continuity conditions on the two diagonals are given in (3.5).

$$\begin{cases} b_{17} = a \cdot b_5 + c \cdot b_{12}, \\ b_{18} = a \cdot b_{13} + c \cdot b_{16}, \\ b_{19} = b \cdot b_6 + d \cdot b_7, \\ b_{20} = b \cdot b_{13} + d \cdot b_{14}, \\ b_{21} = a \cdot b_8 + c \cdot b_9, \\ b_{22} = a \cdot b_{14} + c \cdot b_{15}, \\ b_{23} = b \cdot b_{11} + d \cdot b_{10}, \\ b_{24} = b \cdot b_{16} + d \cdot b_{15}, \\ b_{25} = b \cdot b_{18} + d \cdot b_{22}, \\ b_{25} = a \cdot b_{20} + c \cdot b_{24}, \end{cases} \tag{3.5}$$

where $a = \frac{|P_4P_0|}{|P_4P_2|}$, $b = \frac{|P_3P_0|}{|P_3P_1|}$, $c = 1 - a$, $d = 1 - b$.

It is obvious that the dimension of the solution space is 16. For $i = 1, \ldots, 16$, let

$$(b_1, b_2, \cdots, b_{16}) = \boldsymbol{e}_i = (0, \cdots, 0, 1, 0, \cdots, 0), \tag{3.6}$$

where $\boldsymbol{e}_i$ are unit vectors with dimension of 16, e.g., $\boldsymbol{e}_1 = (1, 0, \cdots, 0)$. To solve the linear system of (3.5) for the 25 unknown $b_1, b_2, \ldots, b_{25}$, we can obtain a set of linear independent solution vectors denoted by $\boldsymbol{b}^{(i)}(i = 1, \ldots, 16)$. Here, $\boldsymbol{b}^{(i)}$ are written by row vectors and $\boldsymbol{b}^{(i)} = \{b_j^{(i)}\}_{j=1}^{25}$ $(i = 1, \ldots, 16)$, which are given by the following 16×25 matrix.

$$\begin{pmatrix} \boldsymbol{b}^{(1)} \\ \boldsymbol{b}^{(2)} \\ \vdots \\ \boldsymbol{b}^{(16)} \end{pmatrix} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&a&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&b&0&0&0&0&0&0 \\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&d&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&a&0&0&0&0 \\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&c&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&d&0&0 \\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&b&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&c&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&a&0&b&0&0&0&0&ab \\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&d&0&a&0&0&ad \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&c&0&d&cd \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&c&0&0&0&0&0&b&bc
\end{pmatrix}. \tag{3.7}$$

Denote by $S_1(x, y), S_2(x, y), \ldots, S_{16}(x, y)$ the 16 cubic spline bases, whose Bézier coefficients are $\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(16)}$. These bases posses many nice properties such as linear independency, partition of unity, compact support, nonnegativity, etc.

3.3. The 12-parameter quadrilateral spline bases

By the 16-parameter quadrilateral spline bases presented in (3.7), we introduce a set of Hermite interpolation bases as follows, which interpolates the 12 parameters (function values, two first-order partial derivatives at four vertices) of the quadrilateral cell.

For a convex quadrangle $P_1P_2P_3P_4$, denote the corner nodes by $P_i = (x_i, y_i)$, $i = 1, \ldots, 4$, and the intersection of two diagonals $\overline{P_1P_3}$ and $\overline{P_2P_4}$ by P_0, as shown in Figure 4. The quadrangle is divided into four subtriangles $D_1, \ldots, D_4$.

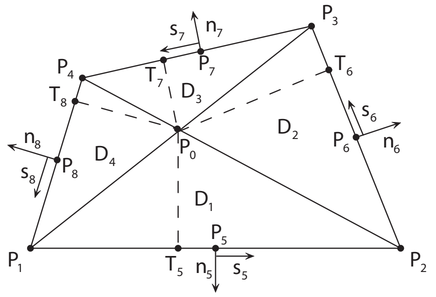


Figure 4. The geometric parameters of a quadrilateral.

Denote the four midpoints on each edge by P_5, P_6, P_7, P_8, and $\boldsymbol{n}_i$ and $\boldsymbol{s}_i (i = 5, \ldots, 8)$ are the unit normal vector and tangent vector, respectively. Let T_5, T_6, T_7, T_8 be the feet of the perpendicular segment from P_0 to each segment, and h_i is the length of perpendicular segment $\overline{P_0T_i}$,

$$h_i = |\overline{P_0T_i}|,\ i = 5, 6, 7, 8. \tag{3.8}$$

Denote $\Delta x_5 = x_2 - x_1$, $\Delta x_6 = x_3 - x_2$, $\Delta x_7 = x_4 - x_3$, $\Delta x_8 = x_1 - x_4$, $\Delta y_5 = y_2 - y_1$, $\Delta y_6 = y_3 - y_2$, $\Delta y_7 = y_4 - y_3$, $\Delta y_8 = y_1 - y_4$, $l_i = \sqrt{\Delta {x_i}^2 + \Delta {y_i}^2}, i = 5, 6, 7, 8$.

For each triangle $D_i (i = 1, \ldots, 4)$, the area coordinate of P_0 is $(1, 0, 0)$. Thus, for each triangle $D_{i-4} (i = 5, \ldots, 8)$, the area coordinates of the foot of the perpendicular T_i is $(0, \alpha_i, \beta_i)$, where $\beta_i = ((x_0 - x_{i-4})\Delta x_i + (y_0 - y_{i-4})\Delta y_i)/l_i^2$, $\alpha_i = 1 - \beta_i$. Therefore, the unit normal vector of each segment can be written by the area coordinates:

$$\boldsymbol{n}_i = ((0, \alpha_i, \beta_i) - (1, 0, 0))/h_i = (-1, \alpha_i, \beta_i)/h_i,\ i = 5, 6, 7, 8. \tag{3.9}$$

According to the constraint that the normal derivative is a linear function and the cubic spline basis functions $S_1(x, y)$, $S_2(x, y)$, $\ldots, S_{16}(x, y)$ are defined by (3.7), we construct the following 12 spline basis functions, which have linear independency, partition of unity, and compact support.

$$\begin{aligned}
\tilde{S}_1 &= S_1 - \tfrac{\beta_8}{2} S_{16} - \tfrac{\alpha_5}{2} S_{13},\\
\tilde{S}_2 &= S_2 - \tfrac{\beta_5}{2} S_{13} - \tfrac{\alpha_6}{2} S_{14},\\
\tilde{S}_3 &= S_3 - \tfrac{\beta_6}{2} S_{14} - \tfrac{\alpha_7}{2} S_{15},\\
\tilde{S}_4 &= S_4 - \tfrac{\beta_7}{2} S_{15} - \tfrac{\alpha_8}{2} S_{16},\\
\tilde{S}_5 &= S_5 + \tfrac{a}{2} S_{16} + \tfrac{a+2\alpha_5-\beta_5}{2} S_{13},\\
\tilde{S}_6 &= S_6 + \tfrac{b}{2} S_{14} + \tfrac{b-\alpha_5+2\beta_5}{2} S_{13},\\
\tilde{S}_7 &= S_7 + \tfrac{d}{2} S_{13} + \tfrac{d+2\alpha_6-\beta_6}{2} S_{14},\\
\tilde{S}_8 &= S_8 + \tfrac{a}{2} S_{15} + \tfrac{a-\alpha_6+2\beta_6}{2} S_{14},\\
\tilde{S}_9 &= S_9 + \tfrac{c}{2} S_{14} + \tfrac{c+2\alpha_7-\beta_7}{2} S_{15},\\
\tilde{S}_{10} &= S_{10} + \tfrac{d}{2} S_{16} + \tfrac{d-\alpha_7+2\beta_7}{2} S_{15},\\
\tilde{S}_{11} &= S_{11} + \tfrac{b}{2} S_{15} + \tfrac{b+2\alpha_8-\beta_8}{2} S_{16},\\
\tilde{S}_{12} &= S_{12} + \tfrac{c}{2} S_{13} + \tfrac{c-\alpha_8+2\beta_8}{2} S_{16}.
\end{aligned} \tag{3.10}$$

Furthermore, based on the geometric parameters of a quadrilateral, the spline basis functions presented in (3.10) can be transformed into the following Hermite interpolation basis functions, which interpolate the 12 parameters at four vertices of each quadrilateral cell.

$$
\begin{aligned}
n_1 &= \tilde{S}_1 + \tilde{S}_{12} + \tilde{S}_5,\\
n_2 &= \tilde{S}_2 + \tilde{S}_6 + \tilde{S}_7,\\
n_3 &= \tilde{S}_3 + \tilde{S}_8 + \tilde{S}_9,\\
n_4 &= \tilde{S}_4 + \tilde{S}_{10} + \tilde{S}_{11},\\
n_1^x &= -\tfrac{\Delta x_8}{3}\tilde{S}_{12} + \tfrac{\Delta x_5}{3}\tilde{S}_5,\\
n_2^x &= -\tfrac{\Delta x_5}{3}\tilde{S}_6 + \tfrac{\Delta x_6}{3}\tilde{S}_7,\\
n_3^x &= -\tfrac{\Delta x_6}{3}\tilde{S}_8 + \tfrac{\Delta x_7}{3}\tilde{S}_9,\\
n_4^x &= -\tfrac{\Delta x_7}{3}\tilde{S}_{10} + \tfrac{\Delta x_8}{3}\tilde{S}_{11},\\
n_1^y &= -\tfrac{\Delta y_8}{3}\tilde{S}_{12} + \tfrac{\Delta y_5}{3}\tilde{S}_5,\\
n_2^y &= -\tfrac{\Delta y_5}{3}\tilde{S}_6 + \tfrac{\Delta y_6}{3}\tilde{S}_7,\\
n_3^y &= -\tfrac{\Delta y_6}{3}\tilde{S}_8 + \tfrac{\Delta y_7}{3}\tilde{S}_9,\\
n_4^y &= -\tfrac{\Delta y_7}{3}\tilde{S}_{10} + \tfrac{\Delta y_8}{3}\tilde{S}_{11}.
\end{aligned}
\tag{3.11}
$$

In [29], the 12-parameter quadrilateral spline bases presented in (3.11) are used for constructing plate elements of the quadrangle in the FEM method, which are denoted by QS12. The bases of QS12 are linear independent and possess the compact support property. Furthermore, Theorem 3.1 shows that QS12 possesses completeness of order 2.

Theorem 3.1 (**[29]**). *Let D be an arbitrary convex quadrilateral domain $P_1P_2P_3P_4$. $n_i(x,y)$, $n_i^x(x,y)$, $n_i^y(x,y)$ $(i = 1,2,3,4)$ are the spline bases given by (3.11). We define the following linear interpolants:*

$$(Nf)(x,y) := \sum_{i=1}^{4}(f(P_i)n_i(x,y) + f_x(P_i)n_i^x(x,y) + f_y(P_i)n_i^y(x,y)), \tag{3.12}$$

where $f_x(P_i), f_y(P_i)$ are the partial derivatives at P_i in regard to x and y, respectively. ***Then for all $f(x,y) \in \mathbb{P}_2$, $(Nf)(x,y) \equiv f(x,y)$, $(x,y) \in D$.***

For an arbitrary hierarchical quadrilateral mesh, suppose θ_1^k is a cell of the quadrilateral mesh at level k, its four vertices are $\{V_i^k\}_{i=1}^4$, where $V_1^k = (a,b)$, $V_2^k = (c,d)$, $V_3^k = (e,f)$, $V_4^k = (g,h)$. As shown in Figure 5, the 12-parameter cubic spline surface on cell θ_1^k can be written as follows.

$$
\begin{aligned}
p(x,y) &= \sum_{i=1}^{4}(p(V_i^k)\, n_i(x,y) + p_x(V_i^k)\, n_i^x(x,y) + p_y(V_i^k)\, n_i^y(x,y))\\
&= \sum_{i=1}^{4}\sum_{j=1}^{3} h_{i,j}^k n_{i,j}^k(x,y),
\end{aligned}
\tag{3.13}
$$

where the geometric information at the four vertices is denoted by

$$
\begin{aligned}
(h_{1,1}^k, h_{1,2}^k, h_{1,3}^k) &= (p(V_1^k), p_x(V_1^k), p_y(V_1^k)),\\
(h_{2,1}^k, h_{2,2}^k, h_{2,3}^k) &= (p(V_2^k), p_x(V_2^k), p_y(V_2^k)),
\end{aligned}
$$

$$(h_{3,1}^k, h_{3,2}^k, h_{3,3}^k) = (p(V_3^k), p_x(V_3^k), p_y(V_3^k)),$$
$$(h_{4,1}^k, h_{4,2}^k, h_{4,3}^k) = (p(V_4^k), p_x(V_4^k), p_y(V_4^k)), \tag{3.14}$$

and the corresponding 12-parameter cubic Hermite interpolation bases are denoted by

$$(n_{1,1}, n_{1,2}, n_{1,3}) = (n_1, n_1^x, n_1^y),$$
$$(n_{2,1}, n_{2,2}, n_{2,3}) = (n_2, n_2^x, n_2^y),$$
$$(n_{3,1}, n_{3,2}, n_{3,3}) = (n_3, n_3^x, n_3^y),$$
$$(n_{4,1}, n_{4,2}, n_{4,3}) = (n_4, n_4^x, n_4^y). \tag{3.15}$$

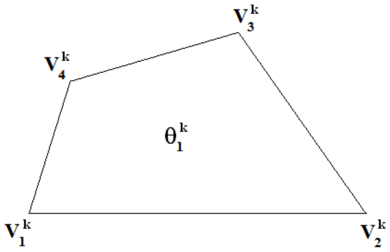


Figure 5. The quadrilateral cell θ_1^k.

In fact, the 12-parameter cubic spline surface over quadrilateral mesh is based on triangulated quadrangulation. Figure 6 shows a quadrangulation and its corresponding triangulated quadrangulation. Since a hierarchical quadrilateral mesh is constructed level by level, suppose there is a cell θ_1^{k-1} at level $k-1$ in a hierarchical quadrilateral mesh as shown in Figure 7(a). Its four vertices are $\{V_i^{k-1}\}(i = 1, \ldots, 4)$ and $V_1^{k-1} = (\overline{a}, \overline{b})$, $V_2^{k-1} = (\overline{c}, \overline{d})$, $V_3^{k-1} = (\overline{e}, \overline{f})$, $V_4^{k-1} = (\overline{g}, \overline{h})$. If θ_1^{k-1} is subdivided into 2×2 uniform subcells $\{\theta_i^k\}(i = 1, \ldots, 4)$ at level k and $V_6^k, V_7^k, V_8^k, V_9^k$ are the middle points of $\overline{V_1^{k-1}V_2^{k-1}}, \overline{V_2^{k-1}V_3^{k-1}}, \overline{V_3^{k-1}V_4^{k-1}}, \overline{V_4^{k-1}V_1^{k-1}}$, respectively. For the 12-parameter cubic spline surface over hierarchical quadrilateral meshes, in order to achieve C^1 (continuously differentiable) at non-basis vertices (i.e., T-junction points) of hierarchical quadrilateral meshes, it is easy to obtain that the function values and the two first-order partial derivatives at $V_6^k, V_7^k, V_8^k, V_9^k$ (denoted by $h_{6,1}^k$, $h_{6,2}^k$, $h_{6,3}^k$ at V_6^k) have the following relation with those at V_1^{k-1}, V_2^{k-1}, V_3^{k-1}, and V_4^{k-1}, respectively.

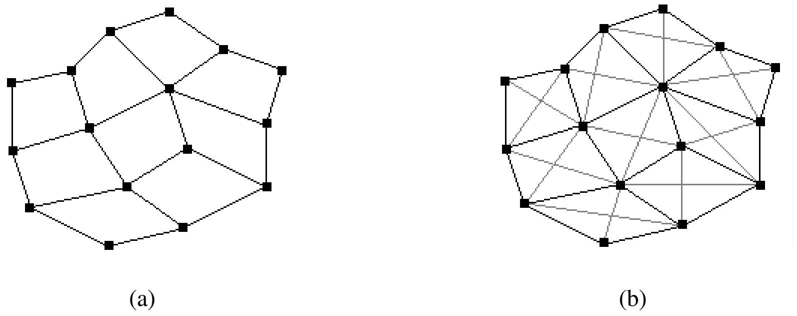

(a) (b)

Figure 6. A quadrangulation and the corresponding triangulated quadrangulation.

$$\left(h_{6,1}^{k},h_{6,2}^{k},h_{6,3}^{k}\right)^{T}=\left(M_1(\overline{a},\overline{b},\overline{c},\overline{d}),M_2(\overline{a},\overline{b},\overline{c},\overline{d})\right)\cdot\left(h_{1,1}^{k-1},h_{1,2}^{k-1},h_{1,3}^{k-1},h_{2,1}^{k-1},h_{2,2}^{k-1},h_{2,3}^{k-1}\right)^{T},\tag{3.16}$$

$$\left(h_{7,1}^{k},h_{7,2}^{k},h_{7,3}^{k}\right)^{T}=\left(M_3(\overline{c},\overline{d},\overline{e},\overline{f}),M_4(\overline{c},\overline{d},\overline{e},\overline{f})\right)\cdot\left(h_{2,1}^{k-1},h_{2,2}^{k-1},h_{2,3}^{k-1},h_{3,1}^{k-1},h_{3,2}^{k-1},h_{3,3}^{k-1}\right)^{T},\tag{3.17}$$

$$\left(h_{8,1}^{k},h_{8,2}^{k},h_{8,3}^{k}\right)^{T}=\left(M_5(\overline{e},\overline{f},\overline{g},\overline{h}),M_6(\overline{e},\overline{f},\overline{g},\overline{h})\right)\cdot\left(h_{3,1}^{k-1},h_{3,2}^{k-1},h_{3,3}^{k-1},h_{4,1}^{k-1},h_{4,2}^{k-1},h_{4,3}^{k-1}\right)^{T},\tag{3.18}$$

$$\left(h_{9,1}^{k},h_{9,2}^{k},h_{9,3}^{k}\right)^{T}=\left(M_7(\overline{g},\overline{h},\overline{a},\overline{b}),M_8(\overline{g},\overline{h},\overline{a},\overline{b})\right)\cdot\left(h_{4,1}^{k-1},h_{4,2}^{k-1},h_{4,3}^{k-1},h_{1,1}^{k-1},h_{1,2}^{k-1},h_{1,3}^{k-1}\right)^{T},\tag{3.19}$$

where $M_1(\overline{a},\overline{b},\overline{c},\overline{d})$, $M_2(\overline{a},\overline{b},\overline{c},\overline{d})$, $M_3(\overline{c},\overline{d},\overline{e},\overline{f})$, $M_4(\overline{c},\overline{d},\overline{e},\overline{f})$, $M_5(\overline{e},\overline{f},\overline{g},\overline{h})$, $M_6(\overline{e},\overline{f},\overline{g},\overline{h})$, $M_7(\overline{g},\overline{h},\overline{a},\overline{b})$, and $M_8(\overline{g},\overline{h},\overline{a},\overline{b})$ are the following matrices:

$$M_1(\bar{a},\bar{b},\bar{c},\bar{d})=\begin{pmatrix}\frac{1}{2} & \frac{(\bar{c}-\bar{a})}{8} & \frac{(\bar{d}-\bar{b})}{8}\\ -\frac{3(\bar{c}-\bar{a})}{2[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & \frac{2(\bar{d}-\bar{b})^2-(\bar{c}-\bar{a})^2}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & -\frac{3(\bar{c}-\bar{a})(\bar{d}-\bar{b})}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]}\\ -\frac{3(\bar{d}-\bar{b})}{2[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & -\frac{3(\bar{c}-\bar{a})(\bar{d}-\bar{b})}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & \frac{2(\bar{c}-\bar{a})^2-(\bar{d}-\bar{b})^2}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]}\end{pmatrix},\tag{3.20}$$

$$M_2(\bar{a},\bar{b},\bar{c},\bar{d})=\begin{pmatrix}\frac{1}{2} & -\frac{(\bar{c}-\bar{a})}{8} & -\frac{(\bar{d}-\bar{b})}{8}\\ \frac{3(\bar{c}-\bar{a})}{2[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & \frac{2(\bar{d}-\bar{b})^2-(\bar{c}-\bar{a})^2}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & -\frac{3(\bar{c}-\bar{a})(\bar{d}-\bar{b})}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]}\\ \frac{3(\bar{d}-\bar{b})}{2[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & -\frac{3(\bar{c}-\bar{a})(\bar{d}-\bar{b})}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]} & \frac{2(\bar{c}-\bar{a})^2-(\bar{d}-\bar{b})^2}{4[(\bar{c}-\bar{a})^2+(\bar{d}-\bar{b})^2]}\end{pmatrix},\tag{3.21}$$

$$M_3(\bar{c},\overline{d},\overline{e},\overline{f})=\begin{pmatrix}\frac{1}{2} & \frac{(\overline{e}-\overline{c})}{8} & \frac{(\overline{f}-\overline{d})}{8}\\ -\frac{3(\overline{e}-\overline{c})}{2[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & \frac{2(\overline{f}-\overline{d})^2-(\overline{e}-\overline{c})^2}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & -\frac{3(\overline{e}-\overline{c})(\overline{f}-\overline{d})}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]}\\ -\frac{3(\overline{f}-\overline{d})}{2[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & -\frac{3(\overline{e}-\overline{c})(\overline{f}-\overline{d})}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & \frac{2(\overline{e}-\overline{c})^2-(\overline{f}-\overline{d})^2}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]}\end{pmatrix},\tag{3.22}$$

$$M_4(\bar{c},\bar{d},\overline{e},\overline{f})=\begin{pmatrix}\frac{1}{2} & -\frac{(\overline{e}-\overline{c})}{8} & -\frac{(\overline{f}-\overline{d})}{8}\\ \frac{3(\overline{e}-\overline{c})}{2[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & \frac{2(\overline{f}-\overline{d})^2-(\overline{e}-\overline{c})^2}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & -\frac{3(\overline{e}-\overline{c})(\overline{f}-\overline{d})}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]}\\ \frac{3(\overline{f}-\overline{d})}{2[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & -\frac{3(\overline{e}-\overline{c})(\overline{f}-\overline{d})}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]} & \frac{2(\overline{e}-\overline{c})^2-(\overline{f}-\overline{d})^2}{4[(\overline{e}-\overline{c})^2+(\overline{f}-\overline{d})^2]}\end{pmatrix},\tag{3.23}$$

$$M_5(\overline{e},\overline{f},\overline{g},\overline{h})=\begin{pmatrix}\frac{1}{2} & \frac{(\overline{g}-\overline{e})}{8} & \frac{(\overline{h}-\overline{f})}{8}\\ -\frac{3(\overline{g}-\overline{e})}{2[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & \frac{2(\overline{h}-\overline{f})^2-(\overline{g}-\overline{e})^2}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & -\frac{3(\overline{g}-\overline{e})(\overline{h}-\overline{f})}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]}\\ -\frac{3(\overline{h}-\overline{f})}{2[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & -\frac{3(\overline{g}-\overline{e})(\overline{h}-\overline{f})}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & \frac{2(\overline{g}-\overline{e})^2-(\overline{h}-\overline{f})^2}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]}\end{pmatrix},\tag{3.24}$$

$$M_6(\overline{e},\overline{f},\overline{g},\overline{h})=\begin{pmatrix}\frac{1}{2} & -\frac{(\overline{g}-\overline{e})}{8} & -\frac{(\overline{h}-\overline{f})}{8}\\ \frac{3(\overline{g}-\overline{e})}{2[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & \frac{2(\overline{h}-\overline{f})^2-(\overline{g}-\overline{e})^2}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & -\frac{3(\overline{g}-\overline{e})(\overline{h}-\overline{f})}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]}\\ \frac{3(\overline{h}-\overline{f})}{2[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & -\frac{3(\overline{g}-\overline{e})(\overline{h}-\overline{f})}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]} & \frac{2(\overline{g}-\overline{e})^2-(\overline{h}-\overline{f})^2}{4[(\overline{g}-\overline{e})^2+(\overline{h}-\overline{f})^2]}\end{pmatrix},\tag{3.25}$$

$$M_7(\overline{g},\overline{h},\overline{a},\overline{b})=\begin{pmatrix} \frac{1}{2} & \frac{(\overline{a}-\overline{g})}{8} & \frac{(\overline{b}-\overline{h})}{8} \\ -\frac{3(\overline{a}-\overline{g})}{2[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & \frac{2(\overline{b}-\overline{h})^2-(\overline{a}-\overline{g})^2}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & -\frac{3(\overline{a}-\overline{g})(\overline{b}-\overline{h})}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} \\ -\frac{3(\overline{b}-\overline{h})}{2[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & -\frac{3(\overline{a}-\overline{g})(\overline{b}-\overline{h})}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & \frac{2(\overline{a}-\overline{g})^2-(\overline{b}-\overline{h})^2}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} \end{pmatrix}, \tag{3.26}$$

$$M_8(\overline{g},\overline{h},\overline{a},\overline{b})=\begin{pmatrix} \frac{1}{2} & -\frac{(\overline{a}-\overline{g})}{8} & -\frac{(\overline{b}-\overline{h})}{8} \\ \frac{3(\overline{a}-\overline{g})}{2[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & \frac{2(\overline{b}-\overline{h})^2-(\overline{a}-\overline{g})^2}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & -\frac{3(\overline{a}-\overline{g})(\overline{b}-\overline{h})}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} \\ \frac{3(\overline{b}-\overline{h})}{2[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & -\frac{3(\overline{a}-\overline{g})(\overline{b}-\overline{h})}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} & \frac{2(\overline{a}-\overline{g})^2-(\overline{b}-\overline{h})^2}{4[(\overline{a}-\overline{g})^2+(\overline{b}-\overline{h})^2]} \end{pmatrix}. \tag{3.27}$$

3.4. Cubic spline surfaces over hierarchical quadrilateral meshes and the evaluation algorithm

If we are provided the geometric information at all basis vertices of hierarchical quadrilateral mesh, the corresponding cubic spline surface can be defined by:

$$P(x,y)=\sum_{i=1}^{\sigma}\sum_{j=1}^{3}\widetilde{h}_{i,j}\widetilde{n}_{i,j}(x,y),(x,y)\in\Omega, \tag{3.28}$$

where $\widetilde{h}_{i,j}, i=1,\dots,\sigma, j=1,\dots,3$, is geometric information at the basis vertices of the hierarchical quadrilateral meshes, $\widetilde{n}_{i,j}(x,y),\ i=1,\dots,\sigma,\ j=1,\dots,3$, are basis functions associated with the basis vertices, and σ is the number of basis vertices.

However, in applications and computations, we need the explicit formulas in regard to the basis functions. In this subsection, we propose a new representation of the cubic spline surface over arbitrary hierarchical quadrilateral mesh, which is based on the geometric information at all basis vertices of quadrilateral mesh. Denote all the cells in the quadrilateral mesh by $\{\theta_l\}_{l=1}^{F}$ and all the vertices by $\{V_i\}_{i=1}^{\omega}$. For each cell θ_l, its four vertices are denoted by $V_1(\theta_l)$, $V_2(\theta_l)$, $V_3(\theta_l)$, $V_4(\theta_l)$. If $V_{i'}(\theta_l)=i$, it means the vertex V_i is the i'-th vertex in the cell θ_l for $i'\in\{1,2,3,4\}$. The cubic spline surface over arbitrary hierarchical quadrilateral mesh is defined by

$$P(x,y)=\sum_{i=1}^{\omega}\sum_{j=1}^{3}\overline{h}_{i,j}\overline{n}_{i,j}(x,y),(x,y)\in\Omega, \tag{3.29}$$

where

$$\overline{n}_{i,j}(s,t)|_{\theta_l}=\begin{cases} n_{i',j}(s,t), & if\ V_{i'}(\theta_l)=i,\\ 0, & otherwise,\end{cases} \qquad l=1,\dots,F,$$

and ω is the number of all vertices of hierarchical quadrilateral mesh. $\overline{h}_{i,j}, i=1,\dots,\omega, j=1,\dots,3$, is geometric information of all vertices of hierarchical quadrilateral mesh and $n_{i',j}(s,t), i'=1,\dots,4, j=1,\dots,3$, are 12-parameter cubic Hermite interpolation bases in regard to the vertices of θ_l (see (3.15) for details).

With the help of the property that the cubic spline surface interpolates geometric information at four vertices of a quadrilateral element, we discuss the cubic spline surface over arbitrary hierarchical quadrilateral mesh. For each quadrilateral element of quadrilateral mesh, for the basis vertex, the surface must interpolate its geometric information; for the non-basis vertex, in order to ensure that the spline function along each quadrilateral segment satisfies the property of C^1, we establish the relation between the geometric information at each non-basis vertex and those of the related vertices of its father element (see (3.16)–(3.19) for details). By searching for all of the father elements, the geometric information at each non-basis vertex can be finally represented by those of the corresponding basis vertices. Thus, we give the evaluation algorithm for a cubic spline surface over arbitrary hierarchical quadrilateral mesh as Algorithm 1.

Algorithm 1 The evaluation algorithm for a cubic spline surface over arbitrary hierarchical quadrilateral mesh.

Input: A cubic spline surface $P(s,t)$ over a given hierarchical quadrilateral mesh with geometric information at the basis vertices and a pair of parametric values (s_0, t_0) inside Q.

Output: The value of the cubic spline surface over the hierarchical quadrilateral mesh $P(s_0, t_0)$.

Step 1. Determine the cell θ_0 containing (s_0, t_0) at level 0. A cubic spline surface over the hierarchical quadrilateral mesh $P_0(s,t)$ is defined as:

$$P_0(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{3} h^0_{i,j} n^0_{i,j}(s,t), \tag{3.30}$$

where $h^0_{i,j}$ is the geometric information of $P(s,t)$ at the four corners $\{V^0_i\}^4_{i=1}$ of θ_0 and $n^0_{i,j}$ are the 12-parameter cubic Hermite interpolation bases defined over θ_0.

Step 2. For $k = 1$ to K, repeat Step 3. Here K is the maximum level of hierarchy associated with the cell which contains (s_0, t_0).

Step 3. Determine the cell θ_k containing (s_0, t_0) at level k. A cubic spline surface over the hierarchical quadrilateral mesh $P_k(s,t)$ is computed:

$$P_k(s,t) := \sum_{i=1}^{4}\sum_{j=1}^{3} h^k_{i,j} n^k_{i,j}(s,t), \tag{3.31}$$

where $n^k_{i,j}$ are the 12-parameter cubic Hermite interpolation bases defined over θ_k as in (3.15), and $h^k_{i,j}$ is geometric information at the four corners $\{V^k_i\}^4_{i=1}$ of θ_k which can be computed by the following rules: if V^k_i is a new basis vertex at level k, then set the corresponding geometric information $h^k_{i,j}$ to be the geometric information of $P(s,t)$ associated with V^k_i; otherwise, just set $h^k_{i,j} = \widehat{h^k_{\tilde{i},j}}$, $\widetilde{i} \in \{6,7,8,9\}$, $j = 1,2,3$, where $\widehat{h^k_{\tilde{i},j}}$ can be computed by one of the formulas of (3.16)–(3.19) according to the different position of the father element θ_{k-1} where the non-basis vertex is located (i.e., the position of V^k_6, V^k_7, V^k_8, V^k_9 relative to the father element θ^{k-1}_1 in Figure 7(b)).

Step 4. Evaluate $P_K(s_0, t_0)$. Output the value.

3.5. The transition matrix M for the geometric information of hierarchical quadrilateral mesh

In this subsection, we provide an algorithm of transition matrix M for the geometric information of hierarchical quadrilateral mesh in Algorithm 2, which can transfer the geometric information at the basis vertices into all vertices of quadrilateral mesh. If the geometric information at the basis vertices of quadrilateral mesh is given, the geometric information at all of the vertices of the quadrilateral mesh can be obtained by the matrix M. Then the cubic spline surface over arbitrary hierarchical quadrilateral mesh can be easily constructed.

Algorithm 2 The algorithm of the transition matrix M for the geometric information of hierarchical quadrilateral mesh.

Input: A hierarchical quadrilateral mesh.

Output: Transition matrix M for the geometric information of the hierarchical quadrilateral mesh.

Step 1. Denote the sets of quadrilateral cells, vertices, and basis vertices by $\mathcal{F}$, D_1, and D_2, and denote the numbers of elements of these sets by F, λ_1, and λ_2, respectively. Set the transition matrix M for the geometric information of the hierarchical quadrilateral mesh as a block matrix with λ_1 rows and λ_2 columns, and each block is a 3×3 sub-matrix.

Step 2. Set $p = 1$. For each quadrilateral cell $\{\theta_l\}_{l=1}^{F}$, set the transition sub-matrix M_l for the geometric information of the quadrilateral element as $I(12, 12)$. We refer to its four vertices as an original vertex set E_l^0. Set $E_l^1 = E_l^0$ and $E_l^2 = \emptyset$. A new vertex set E_l^2 is constructed by repeating the following Steps 2.1 and 2.2 for searching each level of the father element, and the process stops when all elements of E_l^2 are basis vertices.

Step 2.1. For each vertex of E_l^1, we judge:

Case a. If it is a non-basis vertex and it is located in the midpoint of the bottom edge of its father element (as V_6^k in Figure 7(b)), then we add the two related endpoints into E_l^2 (as V_1^{k-1} and V_2^{k-1} in Figure 7(b)). The geometric information at this non-basis vertex and two related endpoints have the relation of (3.16).

Case b. If it is a non-basis vertex and it is located in the midpoint of the right edge of its father element (as V_7^k in Figure 7(b)), then we add the two related endpoints into E_l^2 (as V_2^{k-1} and V_3^{k-1} in Figure 7(b)). The geometric information at this non-basis vertex and two related endpoints have the relation of (3.17).

Case c. If it is a non-basis vertex and it is located in the midpoint of the top edge of its father element (as V_8^k in Figure 7(b)), then we add two related endpoints into E_l^2 (as V_3^{k-1} and V_4^{k-1} in Figure 7(b)). The geometric information at this non-basis vertex and two related endpoints have the relation of (3.18).

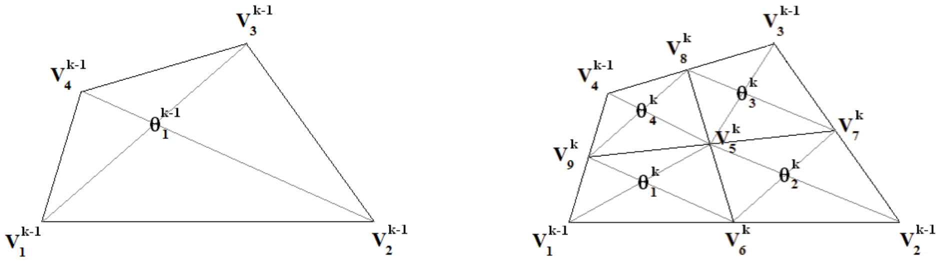
Case d. If it is a non-basis vertex and it is located in the midpoint of the left edge of its father element (as V_9^k in Figure 7(b)), then we add two related endpoints into E_l^2 (as V_4^{k-1} and V_1^{k-1} in Figure 7(b)). The geometric information at this non-basis vertex and two related endpoints have the relation of (3.19).

Case e. If none of the above is true, then we add the vertex itself into E_l^2. The relation of geometric information between a vertex and itself is a unit matrix $I(3, 3)$.

The geometric information's relation between E_l^1 and E_l^2 can be written by a matrix T_p. Set $M_l := M_l \cdot T_p$.

Step 2.2. If there is a non-basis vertex in E_l^2, set $p := p + 1$, $E_l^1 := E_l^2$ and go to Step 2.1; otherwise, go to Step 3.

Step 3. Similar to the finite element method, we assemble the all-element matrix M_l into the global matrix M. Notice that during the assembling of element matrix M_l, we just replace the entries in the right places.



(a) θ_1^{k-1}

(b) Its 2×2 subdivision

Figure 7. The cell θ_1^{k-1} and its 2×2 subdivision.

In the following example, we give the transition matrix M for the geometric information of the special hierarchical quadrilateral mesh in Figure 8.

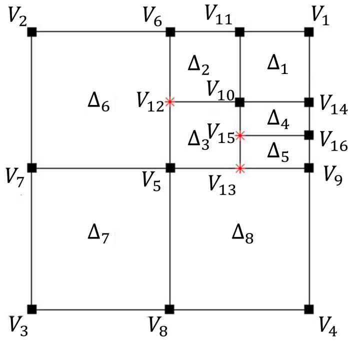


Figure 8. A special hierarchical quadrilateral mesh Q with 3 levels.

Example 3.1. *Figure 8 is a special hierarchical quadrilateral mesh Q defined over* $[0, 1] \times [0, 1]$. *Suppose that each cell of the hierarchical quadrilateral mesh is rectangular and each refinement is uniform. The basis vertices and non-basis vertices are drawn by black squares and red stars, respectively. There are in total 8 rectangular cells, 16 vertices, and 13 basis vertices. That is,* $\mathcal{F} = \{\Delta_1, \ldots, \Delta_8\}$, $D_1 = \{V_1, \ldots, V_{16}\}$, $D_2 = \{V_1, \ldots, V_{10}, V_{11}, V_{14}, V_{16}\}$ *and* $F = 8$, $\lambda_1 = 16$, $\lambda_2 = 13$. *Denote $\mathcal{H}_i$ by the geometric information at vertex V_i:* $\mathcal{H}_i = (h_{i,1}, h_{i,2}, h_{i,3})^T, i = 1, \ldots, 16$.

We just give the transition matrix for the geometric information of the quadrilateral cell $\triangle_4$, i.e., $l = 4$. $E_4^0 = \{V_{14}, V_{10}, V_{15}, V_{16}\}$, *and we set* $E_4^1 = E_4^0$, $E_4^2 = \emptyset$.

(1) For $\triangle_4$ at level 3, its father element at level 2 is the cell with vertices V_{14}, V_{10}, V_{13}, V_9 and it has the domain of $[\frac{3}{4}, 1] \times [\frac{1}{2}, \frac{3}{4}]$. *$V_{15}$ is the midpoint of the left edge of the father element and V_{14}, V_{10}, V_{16} are basis vertices. By Step 2, the current* $E_4^1 = \{V_{14}, V_{10}, V_{15}, V_{16}\}$ *and* $E_4^2 = \{V_{14}, V_{10}, V_{13}, V_{16}\}$. *By (3.19), it can be obtained that*

$$\begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_{15} \\ \mathcal{H}_{16} \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & M_7(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{2}) & M_8(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{2}) & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_{13} \\ \mathcal{H}_{16} \end{pmatrix} = T_1 \begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_{13} \\ \mathcal{H}_{16} \end{pmatrix}. \tag{3.32}$$

(2) Notice that V_{13} is a non-basis vertex in Q, and then we continue seeking the element in the previous generation. Then the father element at level 1 is the cell with vertices V_1, V_6, V_5, V_9 and

it has the domain of $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$. V_{13} *is the midpoint of the bottom edge of the father element and* V_{14}, V_{10}, V_{16} *are basis vertices. The current* $E_4^1 = \{V_{14}, V_{10}, V_{13}, V_{16}\}$ *and* $E_4^2 = \{V_{14}, V_{10}, V_5, V_9, V_{16}\}$*. Thus, it follows that*

$$\begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_{13} \\ \mathcal{H}_{16} \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & M_1(\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}) & M_2(\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}) & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_5 \\ \mathcal{H}_9 \\ \mathcal{H}_{16} \end{pmatrix} = T_2 \begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_5 \\ \mathcal{H}_9 \\ \mathcal{H}_{16} \end{pmatrix}. \tag{3.33}$$

(3) Since all of the vertices in E_4^2 *are basis vertices in* $\mathcal{Q}$*, by substituting (3.33) into (3.32), the transition matrix for the geometric information of* $\triangle_4$ *can be derived by* $M_4 = T_1 T_2$*, i.e.,*

$$\begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_{15} \\ \mathcal{H}_{16} \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & M_7(\frac{3}{4},\frac{3}{4},\frac{3}{4},\frac{1}{2}) & M_8(\frac{3}{4},\frac{3}{4},\frac{3}{4},\frac{1}{2})M_1(\frac{1}{2},\frac{1}{2},1,\frac{1}{2}) & M_8(\frac{3}{4},\frac{3}{4},\frac{3}{4},\frac{1}{2})M_2(\frac{1}{2},\frac{1}{2},1,\frac{1}{2}) & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathcal{H}_{14} \\ \mathcal{H}_{10} \\ \mathcal{H}_5 \\ \mathcal{H}_9 \\ \mathcal{H}_{16} \end{pmatrix}. \tag{3.34}$$

Notice that for the geometric information at each non-basis vertex, we can get the same result by the transition matrices for different quadrilateral elements sharing the vertex. For example, $\mathcal{H}_{15}$ *is the same by the transition matrices for* $\triangle_4$ *or* $\triangle_5$*. By assembling the transition matrices of all elements, the global transition matrix for the geometric information of quadrilateral mesh can be obtained.*

$$\begin{pmatrix} \mathcal{H}_1 \\ \mathcal{H}_2 \\ \mathcal{H}_3 \\ \mathcal{H}_4 \\ \mathcal{H}_5 \\ \mathcal{H}_6 \\ \mathcal{H}_7 \\ \mathcal{H}_8 \\ \mathcal{H}_9 \\ \mathcal{H}_{10} \\ \mathcal{H}_{11} \\ \mathcal{H}_{12} \\ \mathcal{H}_{13} \\ \mathcal{H}_{14} \\ \mathcal{H}_{15} \\ \mathcal{H}_{16} \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & M_8(\frac{1}{2},1,\frac{1}{2},\frac{1}{2}) & M_7(\frac{1}{2},1,\frac{1}{2},\frac{1}{2}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & M_1(\frac{1}{2},\frac{1}{2},1,\frac{1}{2}) & 0 & 0 & 0 & M_2(\frac{1}{2},\frac{1}{2},1,\frac{1}{2}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & M_8(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{2})M_1(\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}) & 0 & 0 & 0 & M_8(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{2})M_2(\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}) & M_7(\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{2}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} \mathcal{H}_1 \\ \mathcal{H}_2 \\ \mathcal{H}_3 \\ \mathcal{H}_4 \\ \mathcal{H}_5 \\ \mathcal{H}_6 \\ \mathcal{H}_7 \\ \mathcal{H}_8 \\ \mathcal{H}_9 \\ \mathcal{H}_{10} \\ \mathcal{H}_{11} \\ \mathcal{H}_{14} \\ \mathcal{H}_{16} \end{pmatrix}. \tag{3.35}$$

For an arbitrary hierarchical quadrilateral mesh, denote by **h** and **d** the vectors of geometric information of all vertices and basis vertices, respectively. According to Algorithm 2, there exists a transition matrix M satisfying

$$\mathbf{h} = M\mathbf{d}. \tag{3.36}$$

It is clear that the matrix M has a full rank of columns. Through the geometric information transition matrix M, the global smoothness between cells is established. Transferring the geometric information via "points" is the key for constructing cubic splines over planar hierarchical meshes, and the implementation of this algorithm is just a calculation, which is simple and efficient. It is easy to represent the cubic spline surface by the geometric information vector $\mathbf{h}$. Since the geometric information at non-basis vertices is determined by relevant basis vertices, the geometric information in vector $\mathbf{d}$ is the true degrees of freedom for cubic spline surface construction.

We now consider the memory requirements for PHT-splines and QS12-splines. Since one basis vertex corresponds to four PHT-spline basis functions, each basis function requires 16 Bézier coefficients for storage. During the refinement, there may be more than 16 non-zero PHT-spline basis functions supported on a single cell. Wang et al. [25] pointed out that if a hierarchical T-mesh contains F cells $\{\theta_i\}_{i=1}^{F}$, and for each cell θ_i, if there are m_i basis vertices corresponding to non-zero basis functions on it, then the memory required for the Bézier coefficients of the PHT-basis functions is $\sum\limits_{i=1}^{F} 64m_i$. It is commonly observed that $m_i \geq 4$ holds for most cells in adaptively refined hierarchical T-meshes. For QS12-splines, each basis function requires 25 Bézier coefficients, and each cell has exactly 12 basis functions, yielding a memory cost of $300F$ for QS12-splines. Consequently, for models requiring high-level subdivision or containing a large number of subcells, the QS12-spline method exhibits a comparatively lower storage requirement.

Like PHT-splines, QS12-splines also possess a local modification property. Altering the geometric information at the control vertex affects only the four adjacent patches, demonstrating an identical locality of influence. In the next section, we show how to reconstruct the geometric information vector $\mathbf{d}$ from scattered data.

4. Reconstruction of cubic spline surfaces over hierarchical quadrilateral meshes

4.1. Cubic spline surface fitting by the least squares method

Since the hierarchical quadrilateral meshes allow T-junctions, cubic spline surfaces defined over these meshes provide a highly flexible representation. In this section, a method for reconstructing cubic spline surfaces over hierarchical quadrilateral meshes is explained.

For an arbitrary hierarchical quadrilateral mesh Q with all vertices $\{V_j\}_{j=1}^{\omega}$, the cubic spline surface $f(x, y)$ defined as $P(s, t)$ in (3.29) can be viewed as the inner product of two vectors $f = \mathbf{n}^T\mathbf{h}$, where $\mathbf{h}$ is a 3ω-dimensional vector that stores the geometric information at all vertices, with its (3j-2)-th to (3j)-th components $\{h_{3j-2}, h_{3j-1}, h_{3j}\} = \{\overline{h}_{j,1}, \overline{h}_{j,2}, \overline{h}_{j,3}\}(j = 1, 2, \ldots, \omega)$ and $\mathbf{n}$ is the vector of 3ω cubic Hermite interpolation bases of 12 parameters with its (3j-2)-th to (3j)-th components $\{n_{3j-2}, n_{3j-1}, n_{3j}\} = \{\overline{n}_{j,1}, \overline{n}_{j,2}, \overline{n}_{j,3}\}(j = 1, 2, \ldots, \omega)$. For a given set of discrete points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{n}$, we have

$$f(\mathbf{p}_i) = \mathbf{n}(\mathbf{p}_i)^T\mathbf{h},\ i = 1, 2, \ldots, n. \tag{4.1}$$

By the definition in (3.29), for each vector $\mathbf{n}(\mathbf{p}_i)$, there are only 12 non-zero entries at the indices of four vertices of the cell containing $\mathbf{p}_i$ in the hierarchical quadrilateral mesh. For other indices, the entries of $\mathbf{n}(\mathbf{p_i})$ are zeros.

Assume that $\mathbf{b}$ is a vector composed of some given function values on $\mathcal{P}$. A good fitting function f is supposed to minimize the errors of the given data. Therefore, the algebraic distance can be defined as a quadratic term:

$$E(\mathbf{h}) = ||N\mathbf{h} - \mathbf{b}||_2^2, \tag{4.2}$$

where the matrix N is constructed of the vectors $\mathbf{n}(\mathbf{p}_i)$ in a row-by-row manner.

Minimization of this quadratic term is widely known as the least squares method. The least squares solution can be computed quite fast by solving a system of linear equations. If the system is well-conditioned, as demonstrated in Examples 5.1–5.3, the least squares solution is a good solution. Unfortunately, it is usually unstable in an adaptive fitting process because its condition number is usually very large for higher refined hierarchical quadrilateral mesh. This is not a coincidence. In the literature of [32, 33], it has been mentioned that the basis functions of PHT-splines reveal a decay phenomenon for refinement of T-meshes, meaning that the finer basis functions approach zero rapidly as the level increases, which leads to the matrices assembled by these PHT-splines' basis functions likely being ill-conditioned. To achieve good fitting solutions in adaptive refinement processes, we use the Tikhonov's regularization as follows:

$$\widehat{E}(\mathbf{h}) = ||N\mathbf{h} - \mathbf{b}||_2^2 + \mu\mathbf{h}^T\mathbf{h}, \tag{4.3}$$

where μ is the regularization parameter.

According to the analysis of Section 3.5, since $f \in S_3^1(Q)$, the geometric information at non-basis vertices is determined by those at basis vertices (denoted by $\mathbf{d}$) and there exists a transition matrix M such that $\mathbf{h} = M\mathbf{d}$. Then (4.3) is equivalent to minimizing the following quadratic function of $\mathbf{d}$,

$$\widehat{E}(\mathbf{d}) = ||NM\mathbf{d} - \mathbf{b}||_2^2 + \mu\mathbf{d}^T M^T M\mathbf{d}. \tag{4.4}$$

Using matrix calculus, the regularization solution can be obtained by

$$\mathbf{d} = (M^T N^T NM + \mu M^T M)^{-1} M^T N^T \mathbf{b}. \tag{4.5}$$

The Tikhonov's regularization ensures that the system remains non-singular for quadrilateral meshes with high refinement levels, which must be solved in the adaptive fitting procedure. In engineering applications, appropriate regularization terms could be selected based on data characteristics such as surface smoothness or sparsity. For instance, the thin-plate energy regularization term is adopted to smooth the surface (see [34]). It is not explored in depth herein and readers could select regularization terms according to their specific needs. In this paper, we present a generalized model for solving surface optimization problems, which mitigates ill-conditioning and enhances numerical stability. A more extensive discussion of alternative methods for reducing the condition number can be found in references [35–37].

In our implementation, we use the generalized cross-validation (GCV) method to compute the regularization parameter μ in our regularized model. For details, see [38]. Since the selection of parameter μ is essential for achieving accurate results and minimizing the number of required iterations, the core principle of GCV lies in selecting an optimal μ to balance model accuracy (goodness-of-fit) and complexity, thereby avoiding overfitting and underfitting. Specifically, the goal of GCV is to identify the μ value that minimizes the generalized cross-validation error (GCV error). That is

$$\mu_{opt} = \arg\min_{\mu} GCV(\mu), \tag{4.6}$$

$$\text{GCV}(\mu) = \frac{||NMd - b||^2}{[trace(I - NM(M^T N^T NM + \mu M^T M)^{-1} M^T N^T)]^2}. \tag{4.7}$$

We utilize the gcv function in Hansen's regularization tools [39], which has become the de facto standard for solving ill-posed problems in MATLAB, to compute the optimal regularization parameter.

4.2. An adaptive cubic spline surface approximation algorithm over hierarchical quadrilateral mesh

In this section, we use adaptive techniques for the refinement selection of quadrilateral meshes. For the given sampled points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n$ and the corresponding function values $\{g(\mathbf{p}_i)\}_{i=1}^n$ of the target function $g(x, y)$, the adaptive algorithm for cubic spline surface approximation is driven by constructing a series of hierarchical quadrilateral meshes from an initial quadrilateral mesh Q_0, as detailed in Algorithm 3.

Notice that if the number of sample points within a cell is fewer than 12 in Algorithm 3, the cell will not be subdivided. To achieve local refinement, we select an appropriate threshold ε for regions which have significant variations in the function values of scattered data points during numerical experiments. In fact, a smaller threshold ε results in denser refinements.

Algorithm 3 Adaptive cubic spline surface approximation algorithm over hierarchical quadrilateral mesh.

Input: Sample points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^n$, the function values $\{g(\mathbf{p}_i)\}_{i=1}^n$, and threshold ε.

Output: A series of hierarchical quadrilateral meshes generated by the adaptive cubic spline surface approximation algorithm and corresponding approximation surfaces.

Step 1. Set $k = 0$. Construct an initial quadrilateral mesh Q_0 according to the sample points. Calculate the geometric information at all the basis vertices by (4.5), and an initial cubic spline surface $f_0(x, y)$ is determined. Denote the cells in Q_0 by $\{\theta_1^0, \ldots, \theta_{F_0}^0\}$. For each cell θ_l^0 $(l = 1, \ldots, F_0)$, if

$$\max_{\mathbf{p}_i \in \theta_l^0} |f_0(\mathbf{p}_i) - g(\mathbf{p}_i)| \geq \varepsilon, \tag{4.8}$$

then the cell must be labeled and refined in the next step.

Step 2. Subdivide all the labeled cells into 2×2 subcells, add the new sub-cells, and form a hierarchical quadrilateral mesh Q_{k+1} at level $k + 1$.

Step 3. Find out all new basis vertices and non-basis vertices in Q_{k+1}. Calculate the vertex transition matrix M for Q_{k+1} and the geometric information at each basis vertex of Q_{k+1} according to (4.5). Using the relation $\mathbf{h} = M\mathbf{d}$, the geometric information at all vertices can be derived. Subsequently, the cubic spline surface at level $k + 1$ is constructed by (3.29), denoted as $f_{k+1}(x, y)$. Denote the cells in Q_{k+1} by $\{\theta_1^{k+1}, \ldots, \theta_{F_{k+1}}^{k+1}\}$. For each cell $\theta_l^{k+1}(l = 1, \ldots, F_{k+1})$, if $\max_{\mathbf{p}_i \in \theta_l^{k+1}} |f_{k+1}(\mathbf{p}_i) - g(\mathbf{p}_i)| \geq \varepsilon$, then the cell θ_l^{k+1} must be labeled and refined.

Step 4. Set $k := k + 1$ and go to Step 2. The procedure repeats Steps 2, 3, and 4 until no cell requires subdivision or the degree of freedom reaches a preset value.

5. Numerical experiments

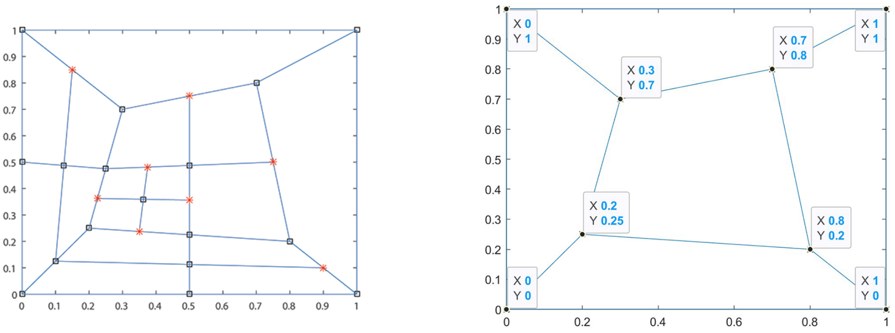
5.1. Numerical experiments without noise

Example 5.1. *Figure 9(a) shows a hierarchical quadrilateral mesh of rectangular domain, which generated from the initial mesh presented in Figure 9(b). We consider an example of cubic spline surface approximation for 400 scattered data points* $\{(x_i, y_i, g_j(x_i, y_i))\}_{i=1}^{400}$, $j = 1, 2$, *which are chosen randomly from the quadratic and cubic polynomials in (5.1) and (5.2). Specially, we choose the regularization parameter* $\mu = 0$.

$$g_1(x, y) = \frac{1}{2}(x^2 + xy + y^2), \tag{5.1}$$

$$g_2(x, y) = \frac{1}{2}(x^3 + xy^2 + y^3). \tag{5.2}$$

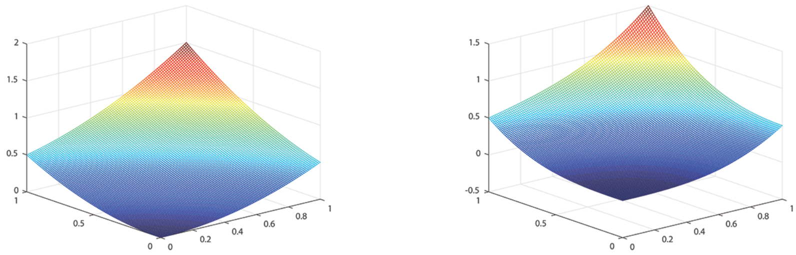
Figures 10 and 11 show the corresponding cubic spline approximation surfaces and errors. This example shows that the cubic spline surface over hierarchical quadrilateral mesh possesses completeness of order 2.



(a)

(b)

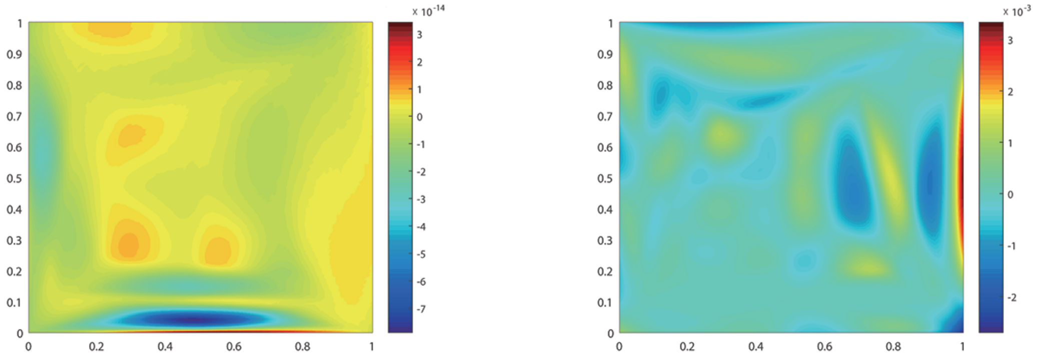
Figure 9. (a) A hierarchical quadrilateral mesh partition of rectangular domain; (b) its initial quadrilateral grid Q_0 with coordinates.



(a)

(b)

Figure 10. The cubic spline surface approximations to data points from g_1 and g_2.

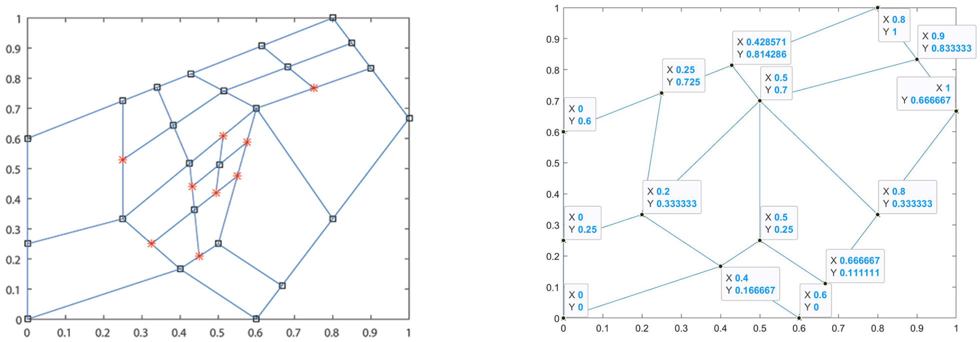


(a) (b)

Figure 11. The corresponding errors of the surfaces in Figure 10 compared by g_1 and g_2.

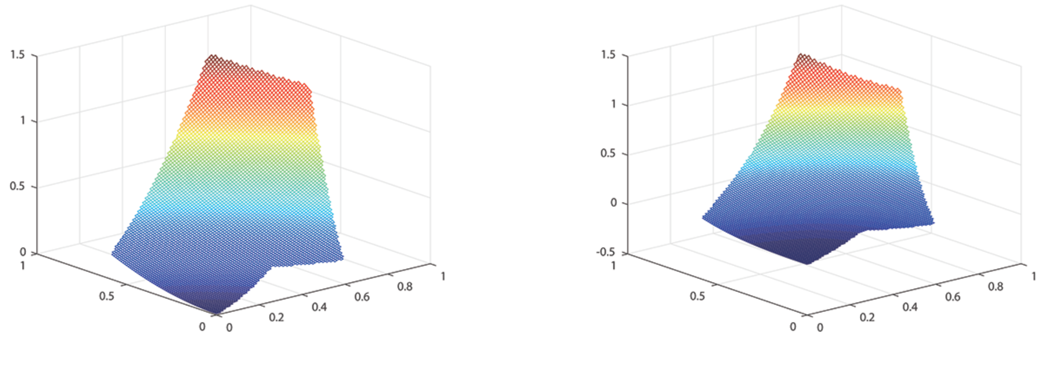
Example 5.2. *Figure 12(a) shows a hierarchical quadrilateral mesh of irregular domain containing a V-notched crack, whose initial quadrilateral grid* Q_0 *with coordinates is illustrated in Figure 12(b). We consider an example of the cubic spline surface for 597 scattered data points* $\{(x_i, y_i, g_j(x_i, y_i))\}_{i=1}^{597}$, $j = 1, 2$, *which are chosen randomly from the quadratic and cubic polynomials in (5.1) and (5.2). Specially, we choose the regularization parameter* $\mu = 0$.

Figures 13 and 14 show the corresponding cubic spline approximation surfaces and errors. This example also shows that the cubic spline surface over hierarchical quadrilateral mesh possesses completeness of order 2. Compared with PHT-spline surface fitting, quadrilateral meshes with arbitrary vertex valence provide greater topological flexibility for geometric modeling within polygonal domains.



(a) (b)

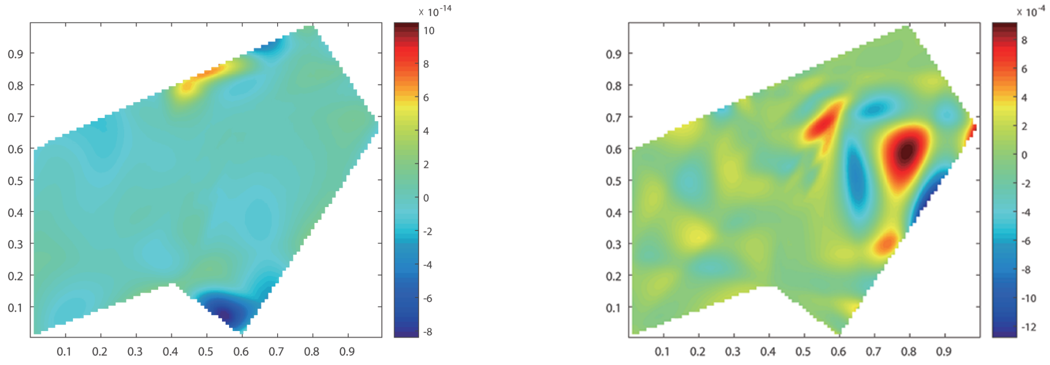
Figure 12. (a) A hierarchical quadrilateral mesh partition of irregular domain; (b) the initial quadrilateral grid Q_0 with coordinates.



(a)

(b)

Figure 13. The cubic spline surface approximations to data points from g_1 and g_2.



(a)

(b)

Figure 14. The corresponding errors of the surfaces in Figure 13 compared by g_1 and g_2.

Example 5.3. *We consider the effects of mesh distortion on cubic spline surface approximation. Figure 15 illustrates a series of distorted hierarchical quadrilateral meshes within a rectangular domain, where the center points are* $(0.5, 0.5) + (\Delta_k, \Delta_k), k = 1, \ldots, 7,$ $\Delta_k = -0.25, -0.2, -0.1, 0, 0.1, 0.2, 0.25,$ *respectively. We give an example of cubic spline surface approximation for 400 scattered data points* $\{(x_i, y_i, g_j(x_i, y_i))\}_{i=1}^{400}, j = 1, 2,$ *which are randomly selected from the quadratic and cubic polynomials in (5.1) and (5.2). Specially, we choose the regularization parameter* $\mu = 0$.

For fitting scattered data points, which are randomly selected from the quadratic and cubic polynomials in (5.1) and (5.2), with $\Delta_k(k = 1, \ldots, 7)$ *varied, Figure 16(a)–(b) present the variations in the fitting errors of cubic spline surfaces over the distorted hierarchical quadrilateral meshes shown in Figure 15. This indicates that mesh distortion has little effect on surface fitting results. That is, our surface fitting algorithm has a low sensitivity to mesh distortion and is suitable for data processing and engineering applications.*

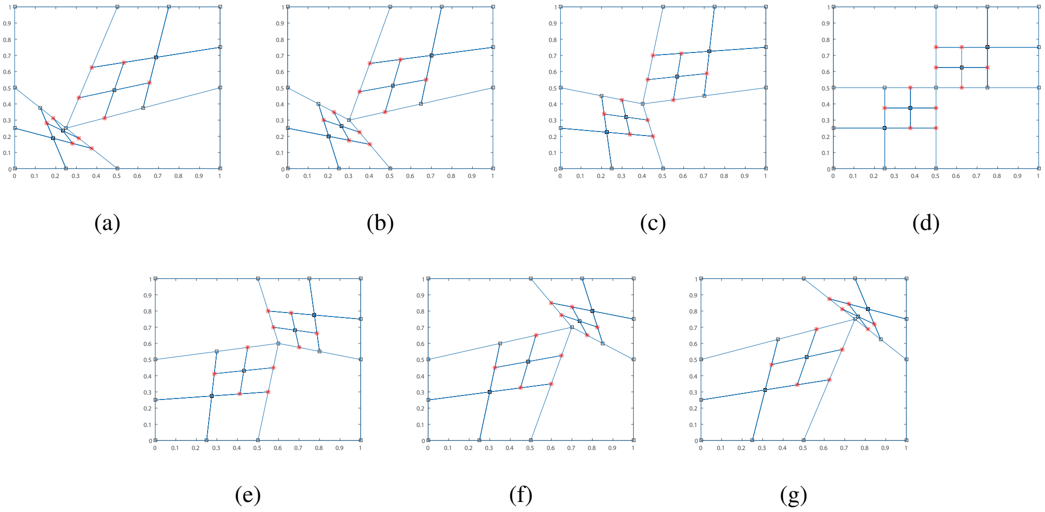
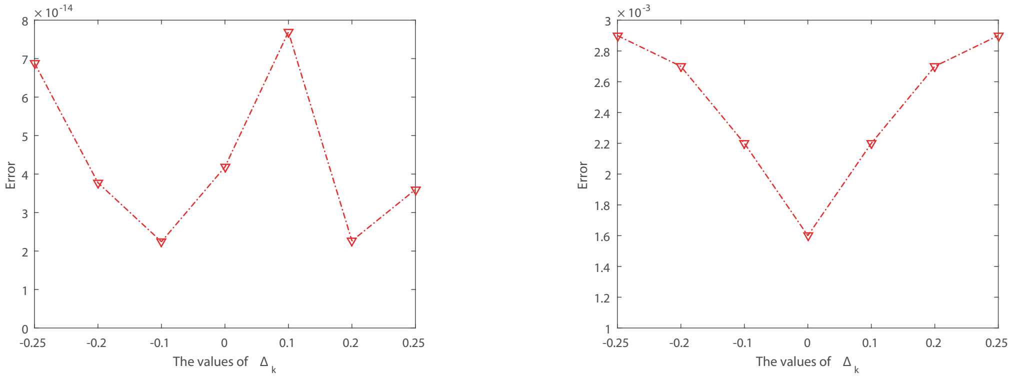


Figure 15. Distorted hierarchical quadrilateral meshes.



(a) The variations of fitting errors of g_1

(b) The variations of fitting errors of g_2

Figure 16. The corresponding errors of cubic spline surfaces over hierarchical quadrilateral meshes in Figure 15 compared by g_1 and g_2.

Example 5.4. *We consider an adaptive cubic spline surface approximation for 10,000 scattered data points* $\{(x_i, y_i, g_3(x_i, y_i))\}_{i=1}^{10000}$*, which are selected randomly from the function in (5.3).* Q_0 *is generated by applying a perturbation to a 4* × *4 uniform tensor product grid, whose coordinates are illustrated in Figure 17. The threshold* ε *is* 4.5×10^{-2}*. The error in Table 1 denotes the maximal error with respect to the given function values, i.e., Error* $= \max |g_3(x_i, y_i) - f(x_i, y_i)|$.

$$g_3(x, y) = \tanh\left(\frac{0.25 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{0.03}\right) + 1, \ (x, y) \in [0, 1] \times [0, 1]. \tag{5.3}$$

Figure 18 illustrates a series of hierarchical quadrilateral meshes generated by Algorithm 3. Figures 19 and 20 show the corresponding cubic spline approximation surfaces and errors for five levels. Table 1 presents the degrees of freedom, the regularization parameters μ *obtained by the GCV method, the maximal errors, and the condition numbers.*

Table 1. Results by Algorithm 3 to fitting the data points from $g_3(x, y)$ for each level in Figure 18.

Level	DOF	Parameter μ	Error	Condition number
0	75	0.9650e-01	0.6044	5.3524e+03
1	243	2.0100e-02	0.4870	5.5841e+03
2	711	2.4000e-03	0.2493	4.5049e+04
3	1632	1.2548e-04	0.0991	8.8918e+05
4	3054	4.0149e-05	0.0180	4.4682e+06

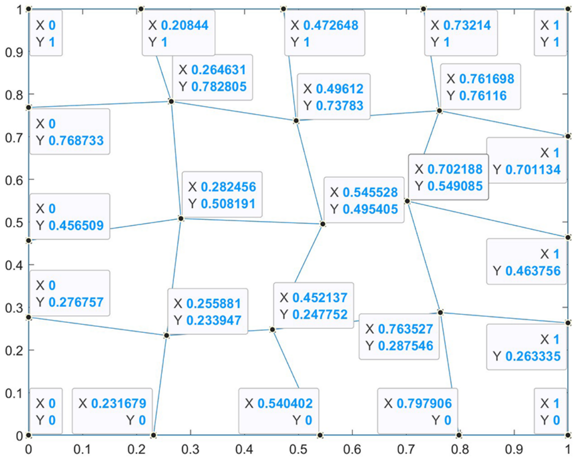
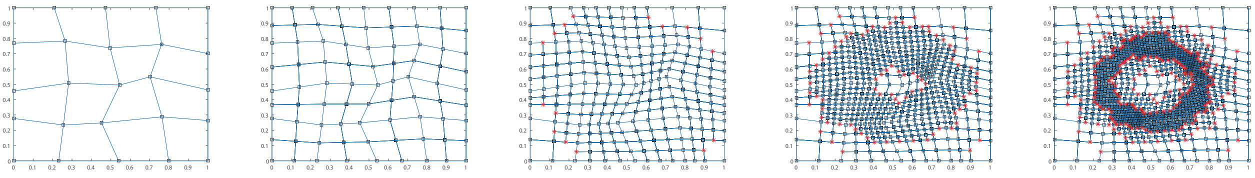
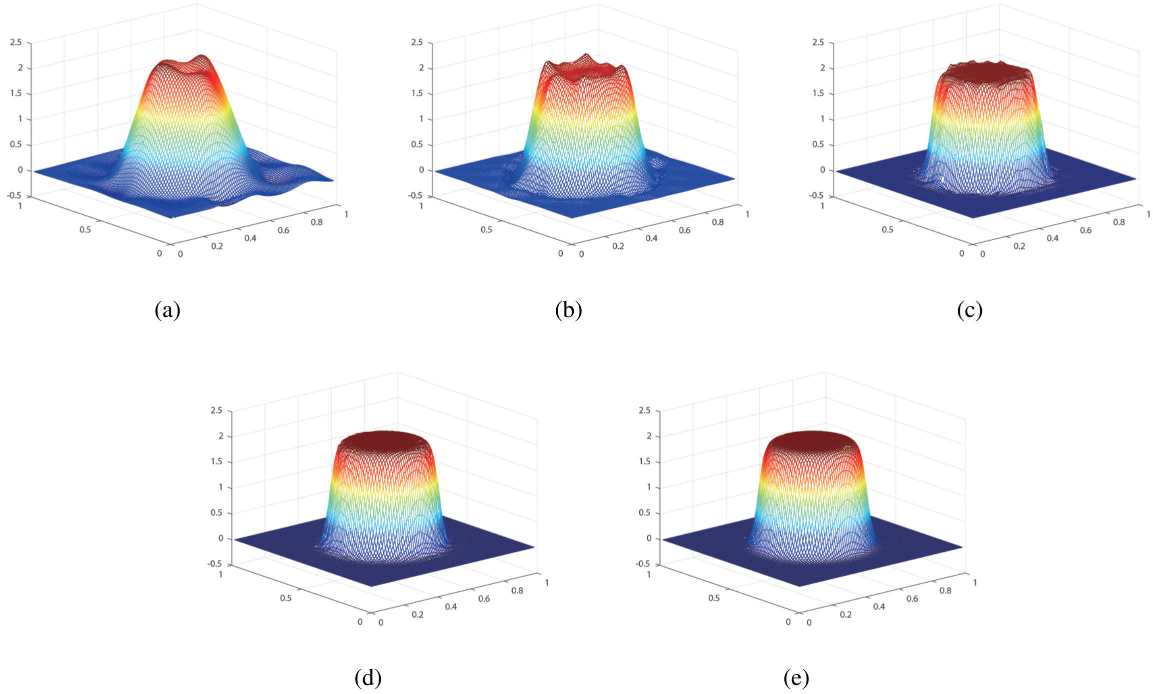


Figure 17. The initial quadrilateral grid Q_0 with coordinates.



(a) (b) (c) (d) (e)

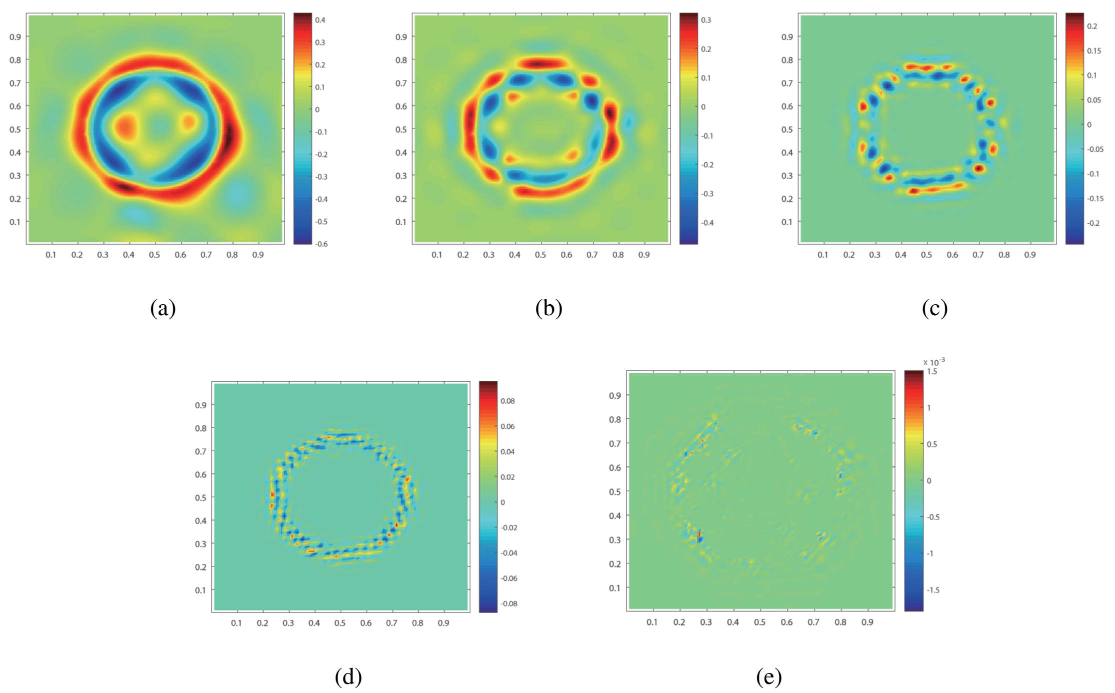
Figure 18. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for scattered data points from $g_3(x, y)$ for all levels.



(a) (b) (c)

(d) (e)

Figure 19. Cubic spline surface approximation to data points from $g_3(x, y)$ for all levels.



(a) (b) (c)

(d) (e)

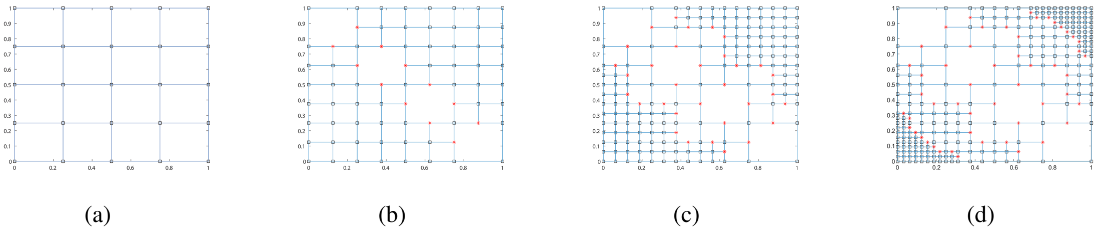
Figure 20. The corresponding errors of the surfaces in Figure 19 compared by $g_3(x, y)$ for all levels.

Example 5.5. *We employ an adaptive cubic spline surface to approximate 10,000 scattered data points,*

$\{(x_i, y_i, g_4(x_i, y_i))\}_{i=1}^{10000}$, *which are uniformly sampled from the function given in (5.4).* $\mathcal{Q}_0$ *is a* 4×4 *uniform tensor product grid. The threshold* ε *is* 0.001. *The error in Table 2 denotes the maximal error with respect to the given function values, i.e., Error* $= \max |g_4(x_i, y_i) - f(x_i, y_i)|$.

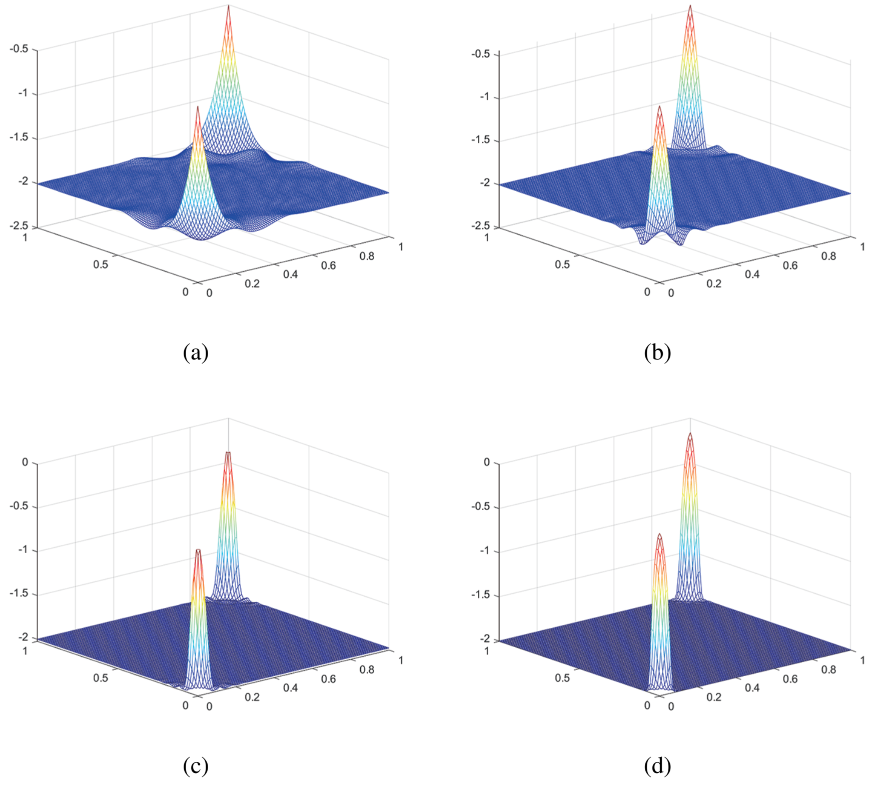
$$g_4(x, y) = -\tanh\left(\frac{x^2 + y^2 - 0.002}{0.002}\right) - \tanh\left(\frac{(x-1)^2 + (y-1)^2 - 0.002}{0.002}\right), \ (x, y) \in [0, 1] \times [0, 1]. \quad (5.4)$$

Figure 21 illustrates a series of hierarchical quadrilateral meshes generated by Algorithm 3 for 4 levels. Figure 22 show the corresponding cubic spline approximation surfaces for 4 levels. Tables 2 and 3 give the the comparison with revised PHT-spline results [24] including degrees of freedom (DOF), the maximal errors, and condition numbers, respectively. Figure 23 shows the computation time per refinement step scaling with degrees of freedom (DOF).



(a) (b) (c) (d)

Figure 21. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for scattered data points from $g_4(x, y)$ for 4 levels.



(a) (b)

(c) (d)

Figure 22. Cubic spline surface approximation to data points from $g_4(x, y)$ for 4 levels.

Table 2. Comparison of Algorithm 3 and PHT-splines in fitting uniformly sampled data points from $g_4(x, y)$.

Model	**Revised PHT-spline**		**Our method**	
Level	DOF	Error	DOF	Error
0	100	0.6857	75	0.6535
1	260	0.3466	183	0.4899
2	500	0.1281	417	0.2044
3	740	0.0045	633	0.0079

Table 3. Comparison of condition numbers for PHT-spline and QS12-spline bases.

Level	PHT-spline bases	QS12-spline bases
0	861.7466	4.6443e+03
1	1.9800e+03	6.7220e+03
2	6.7416e+03	4.3536e+04
3	2.7613e+04	9.1301e+05

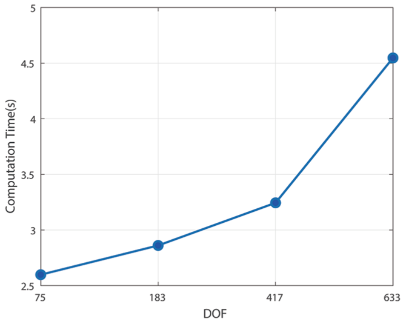
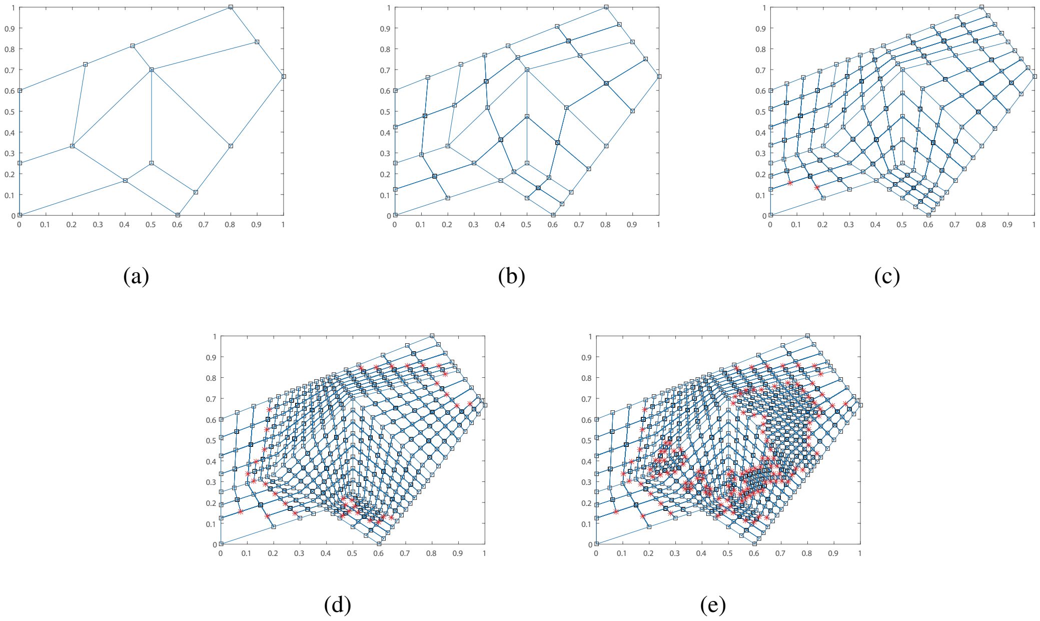


Figure 23. Computation time per refinement step scaling with DOF.

Example 5.6. *We consider an adaptive cubic spline surface approximation for 6426 scattered data points* $\{(x_i, y_i, g_3(x_i, y_i))\}_{i=1}^{6426}$*, which are chosen randomly from (5.3). The initial quadrilateral mesh* Q_0 *is an arbitrary quadrilateral grid whose coordinates are the same as Figure 12(b). The threshold* $\varepsilon = 4.5 \times 10^{-2}$*. The error for the approximation function* $f(x, y)$ *is the maximal error to the given function values, i.e., Error* $= \max |g_3(x_i, y_i) - f(x_i, y_i)|$.

Figure 24 illustrates a series of hierarchical quadrilateral meshes generated by Algorithm 3 for 5 levels. Figures 25 and 26 show the corresponding cubic spline approximation surfaces and errors for all levels. Table 4 gives the degrees of freedom, the regularization parameter μ *obtained by the GCV method, the maximal errors, and the condition numbers, respectively.*



(a)

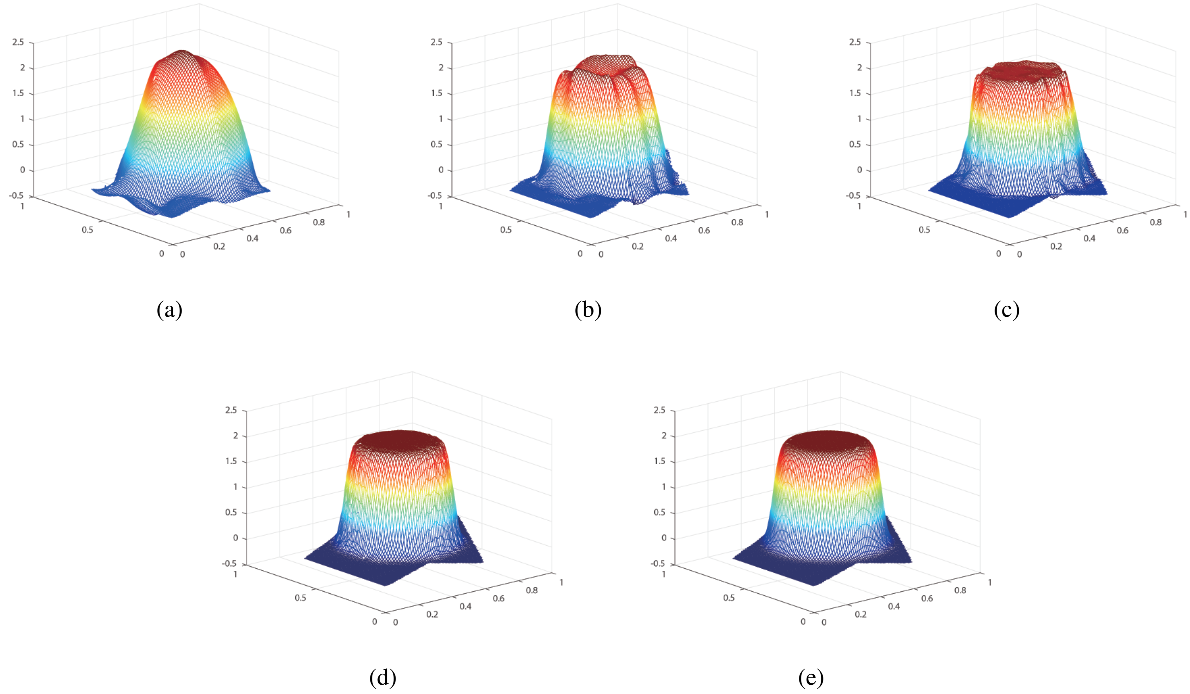
(b)

(c)

(d)

(e)

Figure 24. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for scattered data points from $g_3(x, y)$ for all levels.



(a)

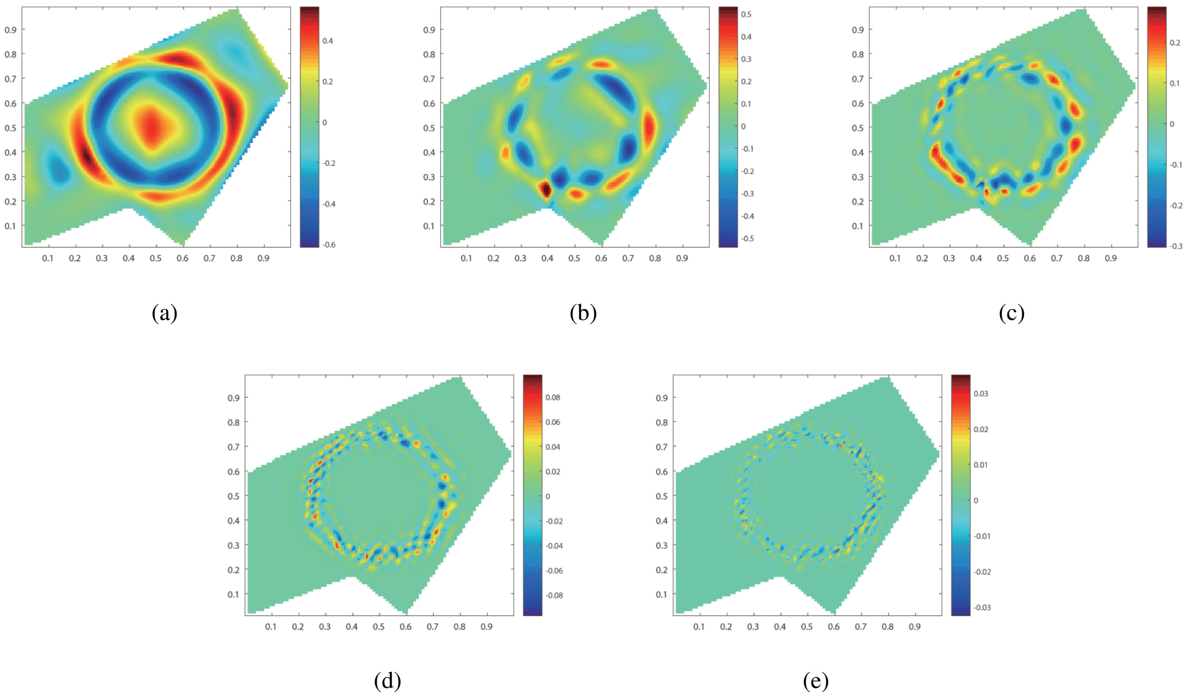
(b)

(c)

(d)

(e)

Figure 25. Cubic spline surface approximation to data points from $g_3(x, y)$ for all levels.



(a)

(b)

(c)

(d)

(e)

Figure 26. The corresponding errors of the surfaces in Figure 25 compared by $g_3(x, y)$ for all levels.

Table 4. Results by Algorithm 3 to fitting the data points from $g_3(x, y)$ for each level in Figure 24.

Level	DOF	Parameter μ	Error	Condition number
0	45	4.2500e-02	0.6162	1.7858e+04
1	135	2.0300e-02	0.5425	1.3574e+04
2	444	4.7000e-03	0.3044	2.2663e+04
3	1206	1.4176e-04	0.0977	6.1372e+05
4	1614	6.7381e-06	0.0351	1.3696e+07

5.2. Numerical experiments with noise

To demonstrate the effectiveness of the adaptive cubic spline surface approximation in Algorithm 3, we perform experiments with the following four functions: Franke's function g_5, a smooth function g_6, and two non-smooth functions g_7 and g_8.

$$\begin{aligned} g_5(x, y) = 0.75\exp[-\frac{(9x-2)^2+(9y-2)^2}{4}] + 0.75\exp[-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}] + \\ 0.5\exp[-\frac{(9x-7)^2+(9y-3)^2}{4}] - 0.2\exp[-(9x-4)^2-(9y-7)^2], \end{aligned} \tag{5.5}$$

$$g_6(x,y) = \frac{1.25+\cos(5.4y)}{6+6(3x-1)^2}, \tag{5.6}$$

$$g_7(x,y) = \begin{cases} \frac{xy}{\sqrt{x^2+y^2}}, & x^2+y^2 \le 1, \\ xy, & x^2+y^2 > 1, \end{cases} \tag{5.7}$$

$$g_8(x,y) = 3 - 3\,|x-y|\,. \tag{5.8}$$

Data points are randomly sampled from each test function $g_j(x, y)$ ($j = 5\ldots 8$) in $\Omega = [0, 1]\times[0, 1]$. Besides, we add noise to the sample data points, i.e., $\{x_i, y_i, g_j(x_i, y_i) + \delta_i\}$, $i = 1, \ldots, 10, 000$. The noise δ_i is generated from pseudo-random values drawn from the standard uniform distribution in the open interval of $\delta * (\max_i |g(x_i, y_i)| - \min_i |g(x_i, y_i)|) * (-1, 1)$, where δ is the noise intensity and $\delta = 0.1\%$ in our experiments. In the following four experiments, the thresholds ε are selected to be 9.0×10^{-2}, 3.0×10^{-3}, 7.0×10^{-3}, and 5.0×10^{-2}. We apply Algorithm 3 to obtain the approximation function f for each test function g. The error is also measured by computing the maximal error between f and g at all sampled data points.

Figures 27–30 present four sets of hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for noisy data points derived from functions $g_j(x, y)$, $j = 5, \ldots, 8$, respectively. Figures 31 and 32 show the corresponding final cubic spline surfaces and their error distributions. Tables 5–8 show the corresponding results by Algorithm 3 for each refinement level.

For scattered data with sharp features, e.g., the scattered data points in $g_8(x, y)$, we can also obtain a good approximation surface by Algorithm 3. In the figures of quadrilateral meshes, we can see the refined cells are consistent with the positions where the sample function values change greatly.

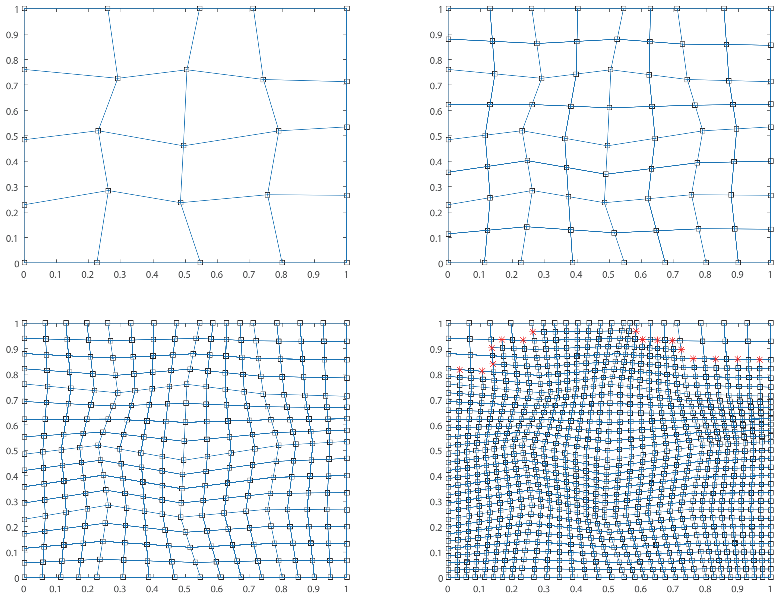


Figure 27. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for noisy data points from $g_5(x, y)$ for 4 levels.

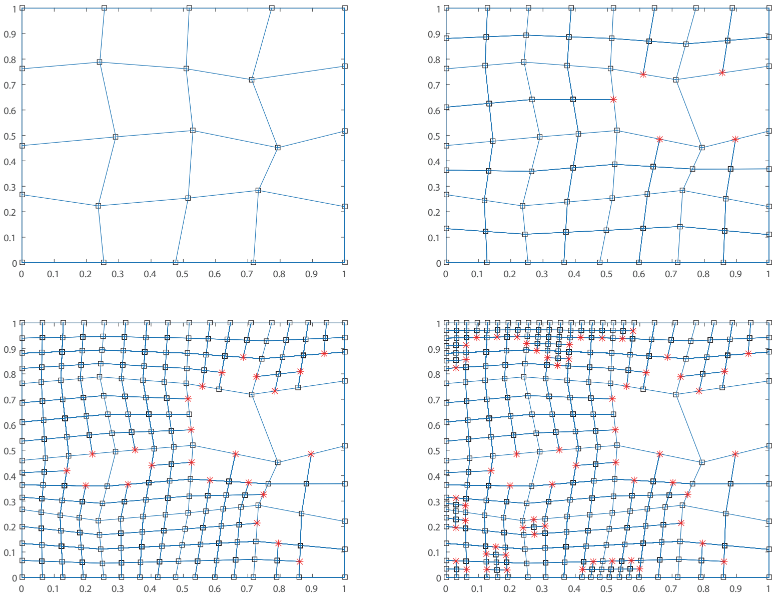


Figure 28. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for noisy data points from $g_6(x, y)$ for 4 levels.

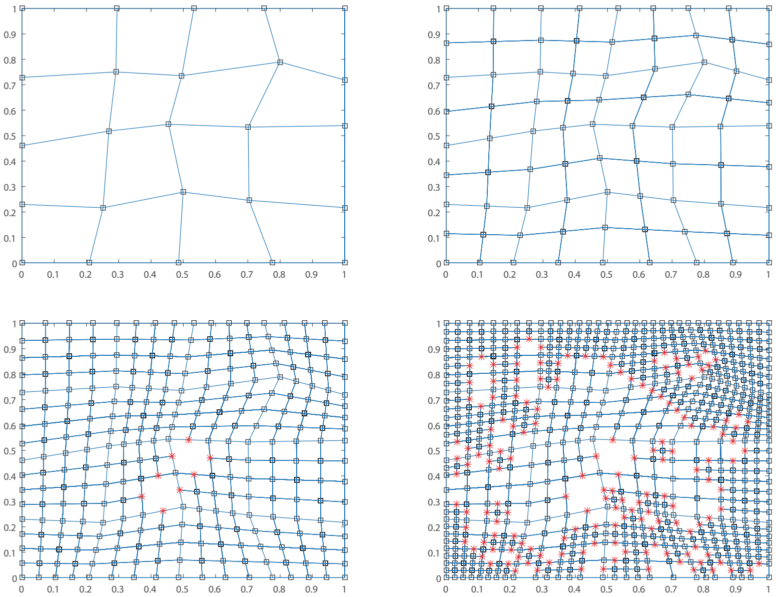


Figure 29. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for noisy data points from $g_7(x, y)$ for 4 levels.

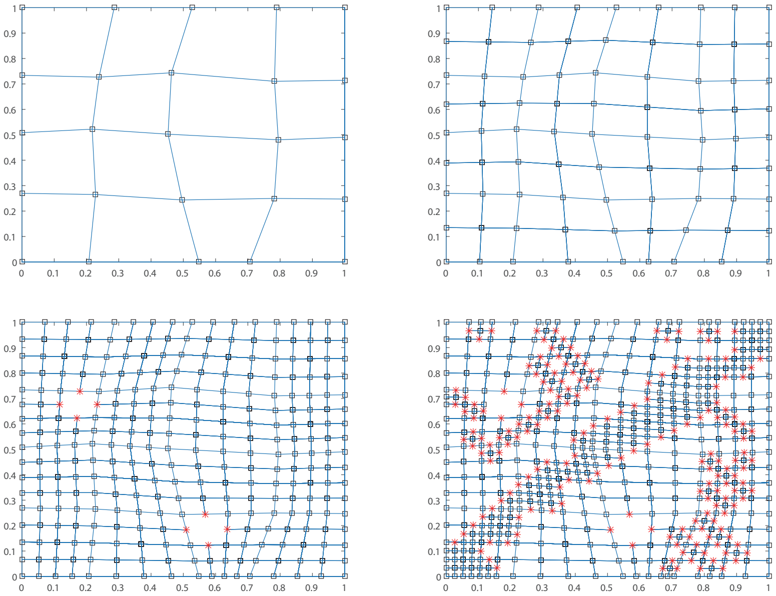


Figure 30. The hierarchical quadrilateral meshes adaptively generated by Algorithm 3 for noisy data points from $g_8(x, y)$ for 4 levels.

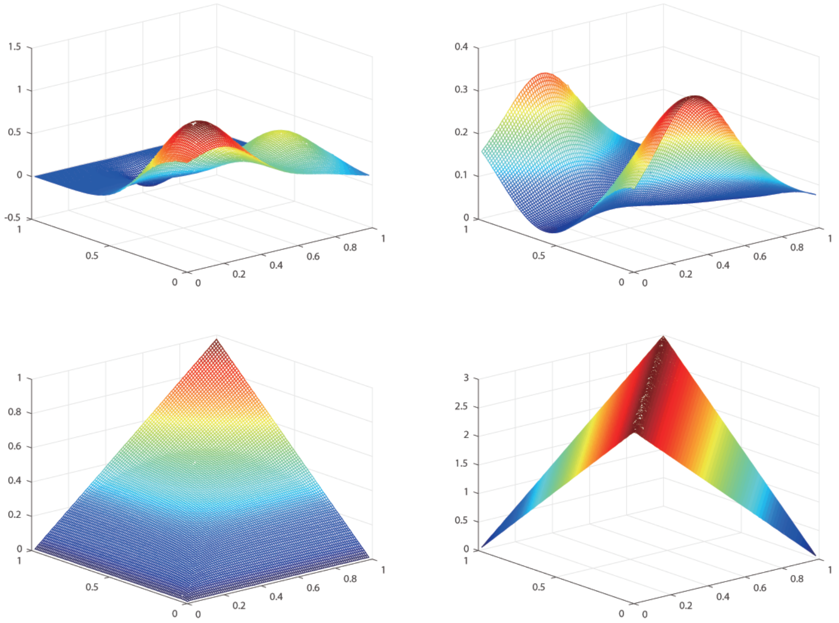


Figure 31. The final adaptive approximations of cubic spline surfaces to fitting the noisy data points from $g_j(x, y)$, $j = 5, \dots, 8$.

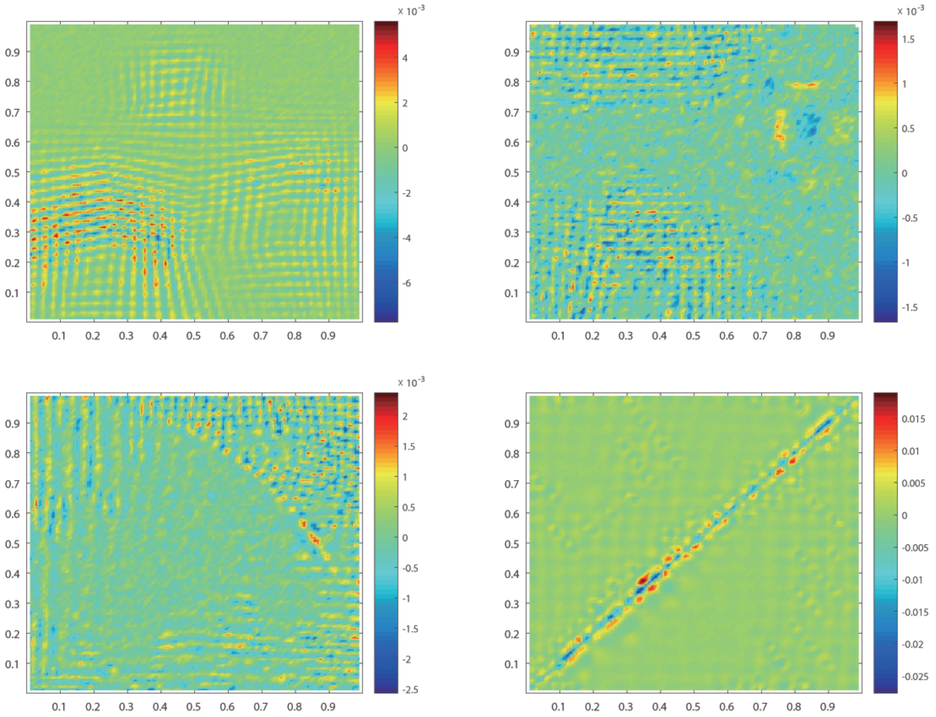


Figure 32. The corresponding errors of the surfaces in Figure 31 compared by $g_j(x, y)$, $j = 5, \dots, 8$.

Table 5. Results by Algorithm 3 to fitting the noisy data points from $g_5(x, y)$ for each level in Figure 27.

Level	DOF	Parameter μ	Error	Condition number
0	75	1.9970e-01	0.0316	9.3245e+03
1	243	4.5300e-02	0.0279	1.3268e+04
2	867	1.0200e-02	0.0242	1.7211e+04
3	3048	1.0536e-04	0.0078	1.4800e+06

Table 6. Results by Algorithm 3 to fitting the noisy data points from $g_6(x, y)$ for each level in Figure 28.

Level	DOF	Parameter μ	Error	Condition number
0	75	2.2700e-01	0.0121	8.2050e+03
1	216	9.4000e-03	0.0035	1.1786e+05
2	615	7.4323e-04	0.0025	1.4284e+06
3	798	5.8134e-05	0.0017	2.0010e+07

Table 7. Results by Algorithm 3 to fitting the noisy data points from $g_7(x, y)$ for each level in Figure 29.

Level	DOF	Parameter μ	Error	Condition number
0	75	1.2750e-01	0.0176	5.5502e+03
1	243	3.1800e-02	0.0147	7.4433e+03
2	837	6.4197e-04	0.0060	1.9209e+05
3	1854	3.6793e-05	0.0026	3.5264e+06

Table 8. Results by Algorithm 3 to fitting the noisy data points from $g_8(x, y)$ for each level in Figure 30.

Level	DOF	Parameter μ	Error	Condition number
0	75	1.046e-01	0.2591	5.1753e+03
1	243	2.3600e-02	0.1391	7.4325e+03
2	837	4.8819e-04	0.0565	1.6808e+05
3	1272	1.5955e-05	0.0277	5.3248e+06

6. Conclusions

In this paper, we present a novel surface construction algorithm based on cubic spline interpolation of geometric information, i.e., function values and two first-order partial derivatives at basis vertices within a hierarchical quadrilateral mesh. First, we introduce an algorithm to compute the transition matrix M, which transfers geometric information from the basis vertices to all mesh vertices. This is the key for constructing cubic splines over planar hierarchical meshes and its algorithmic implementation is just a calculation. Second, we develop an adaptive cubic spline surface refinement algorithm for fitting scattered data points. Numerical experiments demonstrate that the cubic spline surface over hierarchical quadrilateral meshes possesses a completeness of order 2 and exhibits low sensitivity to mesh distortion. In addition, the proposed adaptive algorithm can effectively solve problems of fitting scattered data within polygonal domains.

Author contributions

Pengxiao Wang: Conceptualization, writing—original draft, investigation, software, visualization, funding acquisition; Chongjun Li: Conceptualization, writing—review and editing, formal analysis, supervision. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (No. 21KJB110016) and the Philosophy and Social Science Foundation of the Higher Education Instututions of Jiangsu Province (No. 2022SJYB0217).

Conflict of interest

The authors declare no conflict of interest.

References

1. L. Piegl, W. Tiller, *The NURBS book (2nd ed)*, New York: Springer-Verlag, 1997. Available from: https://link.springer.com/book/10.1007/978-3-642-59223-2
2. T. W. Sederberg, D. L. Cardon, F. G. Thomas, N. S. North, J. M. Zheng , T. Lyche, T-spline simplification and local refinement, *ACM Trans. Graph.*, **23** (2004), 276–283. https://doi.org/10.1145/1015706.1015715
3. T. W. Sederberg, J. M. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, *ACM Trans. Graph.*, **22** (2003), 477–484. https://doi.org/10.1145/882262.882295
4. M. R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, *Comput. Method. Appl. M.*, **199** (2010), 264–275. https://doi.org/10.1016/j.cma.2008.07.012
5. E. J. Evans, M. A. Scott, X. Li, D. C. Thomas, Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis, *Comput. Method. Appl. M.*, **284** (2015), 1–20. https://doi.org/10.1016/j.cma.2014.05.019
6. X. Li, J. J. Zhang, AS++ T-splines: Linear independence and approximation, *Comput. Method. Appl. M.*, **333** (2018), 462–474. https://doi.org/10.1016/j.cma.2018.01.041
7. X. Li, J. M. Zheng, T. W. Sederberg, T. J. R. Hughes, M. A. Scott, On linear independence of T-spline blending functions, *Comput. Aided Geom. D.*, **29**(2012), 63–76. https://doi.org/10.1016/j.cagd.2011.08.005
8. C. Giannelli, B. Jüttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, *Comput. Aided Geom. D.*, **29** (2012), 485–498. https://doi.org/10.1016/j.cagd.2012.03.025
9. C. Giannelli, B. Jüttler, H. Speleers, Strongly stable bases for adaptively refined multilevel spline spaces, *Adv. Comput. Math.*, **40** (2014), 459–490. https://doi.org/10.1007/s10444-013-9315-2
10. M. D. Pan, R. J. Zou, B. Jüttler, Algorithms and data structures for C^s-smooth RMB-splines of Degree $2s + 1$, *Comput. Aided Geom. D.,* **114** (2024), 102389. https://doi.org/10.1016/j.cagd.2024.102389
11. L. Groiss, B. Jüttler, M. D. Pan, Local linear independence of bilinear (and higher degree) B-splines on hierarchical T-meshes, *Comput. Aided Geom. D.,* **103** (2023), 102190. https://doi.org/10.1016/j.cagd.2023.102190

12. L. Groiss, B. Jüttler, M. D. Pan, On Tensor-Product Bases of PHT-Spline Spaces, In *Proceedings of the INdAM Meeting: Approximation Theory and Numerical Analysis Meet Algebra, Geometry, Topology*; Springer Nature: Singapore, 2022; Volume 60, pp. 181–203. https://doi.org/10.1007/978-981-97-6508-9_9
13. M. D. Pan, W. H. Tong, F. L. Chen, Phase-field guided surface reconstruction based on implicit hierarchical B-splines, *Comput. Aided Geom. D.,* **52–53** (2017), 154–169. https://doi.org/10.1016/j.cagd.2017.03.009
14. J. S. Deng, F. L. Chen, X. Li, C. Q. Hu, W. H. Tong, Z. W. Yang, Y. Y. Feng, Polynomial splines over hierarchical T-meshes, *Graph. Models,* **70** (2008), 76–86. http://doi.org/10.1016/j.gmod.2008.03.001
15. J. S. Deng, F. L. Chen, Y. Y. Feng, Dimensions of spline spaces over T-meshes, *J. Comput. Appl. Math.,* **194** (2006), 267–283. https://doi.org/10.1016/j.cam.2005.07.009
16. P. X. Wang, C. J. Li, The instability in the dimensions of spline spaces over T-meshes with nested T-cycles, *Numer. Math. Theor. Meth. Appl.,* **12** (2019), 187–211. https://doi.org/10.4208/nmtma.OA-2017-0110
17. D. Berdinsky, M. J. Oh, T. W. Kim, B. Mourrain, On the problem of instability in the dimension of a spline space over a T-mesh, *Comput. Graph.,* **36** (2012), 507–513. https://doi.org/10.1016/j.cag.2012.03.005
18. B. Mourrain, On the dimension of spline spaces on planar T-meshes, *Math. Comput.,* **286** (2015), 847–871. Available from: `https://www.ams.org/journals/mcom/2014-83-286/S0025-5718-2013-02738-X/S0025-5718-2013-02738-X.pdf`
19. P. X. Wang, The instability in the dimensions of polynomial splines of mixed smoothness over T-meshes, *Mathematics*, **13** (2025), 2886. https://doi.org/10.3390/math13172886
20. L. Tian, F. L. Chen, Q. Du, Adaptive finite element methods for elliptic equations over hierarchical T-meshes, *J. Comput. Appl. Math.,* **236** (2011), 878–891. https://doi.org/10.1016/j.cam.2011.05.016
21. N. Nguyen-Thanh, N. Valizadeh, M. N. Nguyen, H. Nguyen-Xuan, X. Zhuang, P. Areias, et al., An extended isogeometric thin shell analysis based on Kirchhoff–Love theory, *Comput. Method. Appl. M.,* **284** (2015), 265–291. https://doi.org/10.1016/j.cma.2014.08.025
22. X. Li, J. S. Deng, F. L. Chen, Surface modeling with polynomial splines over hierarchical T-meshes, *Visual Comput.,* **23** (2007), 1027–1033. Available from: `https://link.springer.com/article/10.1007/s00371-007-0170-3`
23. J. Wang, Z. Yang, L. Jin, J. Deng, F. Chen, Parallel and adaptive surface reconstruction based on implicit PHT-splines, *Comput. Aided. Geom. D.,* **28** (2011), 463–474. https://doi.org/10.1016/j.cagd.2011.06.004
24. J. L. Xu, Some Problems in isogeometric analysis [PhD dissertation]. Hefei (China): University of Science and Technology of China; 2014. 117p. Chinese. Available from: `https://kns.cnki.net/kcms2/article/abstract?v=9ZlGolWNudLPXc0s654q4_9ys_BWnYZ6jUVeVmbxyIo-xvqCwJRMCH1ZDamepg7bIICd9pCi9JH7X6qgX78YTllUWYLIes9qetP9ron1lDVkkv32WY0BGbg48r-I4IEyed7xeAjHLIIrx4ElFunl8mSOmirq3ffKkrROM_ea1zryxzIkFmuk3DB1Lq25HDkp&uniplatform=NZKPT&language=CHS`

25. Z. H. Wang, F. L. Chen, J. S. Deng, Evaluation algorithm of PHT-spline surfaces, *Numer. Math.-Theory Me.,* **10** (2017), 760–774. https://doi.org/10.4208/nmtma.2017.0003

26. B. F. D. Veubeke, A conforming finite element for plate bending, *Int. J. Solids Struct.,* **4** (1968), 95–108. https://doi.org/10.1016/0020-7683(68)90035-8

27. G. Sander, Bornes supérieures et inférieures dans l'analyse matricielle des plaques en flexion-torsion, *Bull. Soc. Royale Science Liège,* **33** (1964), 456–494.

28. P. G. Ciarlet, *The finite element method for elliptic problems*, Philadelphia: SIAM Publications, 2002. https://doi.org/10.1137/1.9780898719208

29. J. Chen, C. J. Li, The cubic spline Hermite interpolation bases for thin plate bending quadrilateral elements(in Chinese), *Scientia Sinica Math.,* **45** (2015), 1523–1536. https://doi.org/10.1360/n012015-00035

30. J. Chen, C. J. Li, W. J. Chen, A family of spline finite elements, *Comput. Struct.,* **88** (2010), 718–727. https://doi.org/10.1016/j.compstruc.2010.02.011

31. G. Farin, Triangular Bernstein-Bézier patches, *Comput. Aided Geom. D.,* **3** (1986), 83–127. https://doi.org/10.1016/0167-8396(86)90016-6

32. Q. Ni, X. H. Wang, J. S. Deng, Modified basis functions for MPHT-splines, *J. Comput. Appl. Math.,* **375** (2020), 112817. https://doi.org/10.1016/j.cam.2020.112817

33. H. M. Kang, J. L. Xu, F. L. Chen, J. S. Deng, A new basis for PHT-splines, *Graph. Models,* **82** (2015), 149–159. https://doi.org/10.1016/j.gmod.2015.06.011

34. C. Bracco, C. Giannelli, D. Großmann, A. Sestini, Adaptive fitting with THB-splines: Error analysis and industrial applications, *Comput. Aided Geom. D.,* **62** (2018), 239–252. https://doi.org/10.1016/j.cagd.2018.03.026

35. H. W. Lin, Q. Cao, X. T. Zhang, The convergence of least-squares progressive iterative approximation for singular least-squares fitting system, *J. Syst. Sci. Complex,* **31** (2018), 1618–1632. https://doi.org/10.1007/s11424-018-7443-y

36. C. Y. Deng, H. W. Lin, Progressive and iterative approximation for least-squares B-spline curve and surface fitting, *Comput. Aided Des.,* **47** (2014), 32–44. https://doi.org/10.1016/j.cad.2013.08.012

37. N. C. Wu, C. Z. Liu, Asynchronous progressive iterative approximation method for least squares fitting, *Comput. Aided Geom. D.,* **111** (2024), 102295. https://doi.org/10.1016/j.cagd.2024.102295

38. G. Wahba, *Spline Models for Observational Data*, Philadelphia: SIAM Publications, 1990. https://doi.org/10.1137/1.9781611970128

39. P. C. Hansen, REGULARIZATION TOOLS: A Matlab package for analysis and solution of discrete ill-posed problems, *Numer. Algor.,* **6** (1994), 1–35. https://doi.org/10.1007/BF02149761