Interpolatory Quad/Triangle Subdivision Schemes for Surface Design

Qingtang Jiang[1,*] Baobin Li[2] and Weiwei Zhu[1]

[1] *Dept. of Math and Computer Sci., Univ. of Missouri–St. Louis, St. Louis, MO 63121, U.S.A.*
[2] *School of Information Sci. and Engineering, Graduate Univ. of Chinese Academy of Sci., Beijing, 100871, P. R. China*

Abstract

Recently the study and construction of quad/triangle subdivision schemes have attracted attention. The quad/triangle subdivision starts with a control net consisting of both quads and triangles and produces finer and finer meshes with quads and triangles. The use of the quad/triangle structure for surface design is motivated by the fact that in CAD modelling, the designers often want to model certain regions with quad meshes and others with triangle meshes to get better visual quality of subdivision surfaces. Though the smoothness analysis tool for regular quad/triangle vertices has been established and C^1 and C^2 quad/triangle schemes (for regular vertices) have been constructed, there is no interpolatory quad/triangle schemes available in the literature. The problem for this is probably that since the template sizes of the local averaging rules of interpolatory schemes for either quad subdivision or triangle subdivision are big, an interpolatory quad/triangle scheme will have large sizes of local averaging rule templates. In this paper we consider matrix-valued interpolatory quad/triangle schemes.

In this paper, first we show that both scalar-valued and matrix-valued quad/triangle subdivision scheme can be derived from a nonhomogeneous refinement equation. This observation enables us to treat polynomial reproduction of scalar-valued and matrix-valued quad/triangle schemes in a uniform way. Then, with the result on the polynomial reproduction of matrix-valued quad/triangle schemes provided in our accompanying paper, we obtain in this paper a smoothness estimate for matrix-valued quad/triangle schemes, which extends the smoothness analysis of Levin-Levin from the scalar-valued setting to the matrix-valued setting. Finally, with this smoothness estimate established in this paper, we construct C^1 matrix-valued interpolatory quad/triangle scheme (for regular vertices) with the same sizes of local averaging rule templates as those of Stam-Loop's quad/triangle scheme. We also obtain C^2 matrix-valued interpolatory quad/triangle scheme (for regular vertices) with reasonable sizes of local averaging rule templates.

[*] Research supported by UM Research Board 10/05 and UMSL Research Award 10/06.

Keywords: Quad/triangle subdivision, nonhomogeneous refinement equation, interpolatory quad/triangle scheme, matrix-valued subdivision, polynomial reproduction, smoothness analysis

1 Introduction

Subdivision is a useful efficient algorithm to generate smooth surfaces in the computer aided geometric design. Starting from a control net, a subdivision scheme generates finer and finer meshes that eventually converge to the desirable limiting surface. Conventionally the quadrilateral (quad) subdivision and the triangle subdivision are used for surface subdivision. A quad subdivision starts with a (non-planar) quad control mesh (net) and generates refined meshes consisting of all quads, while a triangle subdivision is based on a triangle control mesh and generates finer and finer meshes consisting of all triangles. For example, Catmull-Clark's scheme [3] and Loop's scheme [29] are the two most commonly used schemes for subdivision. However, the former is for the quad mesh only, while the latter is for the triangle mesh only. But in many applications, it is desirable to have surfaces that have a hybrid quad/triangle net structure so that the surface generated has better visual quality, see e.g. [35] for detailed discussion on this issue. Therefore, the study of the quad/triangle subdivision, whose control net and refined meshes consist of both quads and triangles, has attracted much attention.

Quad/triangle subdivision is first considered in [30]. [35] designs a C^1 quad/triangle scheme, and later, [34] and [26] each provide a C^2 scheme for regular quad/triangle vertices. In all these papers, Catmull-Clark's scheme is used for *quad vertices* (vertices entirely surrounded by quads) and Loop's scheme is used for *triangle vertices* (vertices entirely surrounded by triangles) far away from *quad/triangle vertices* (vertices surrounded by both quad(s) and triangle(s)). In [31], the scheme derived from a 4 directional box-spline is used for quad vertices. Polynomial reproduction and the smoothness of quad/triangle schemes are studied in [25] and [26] resp.

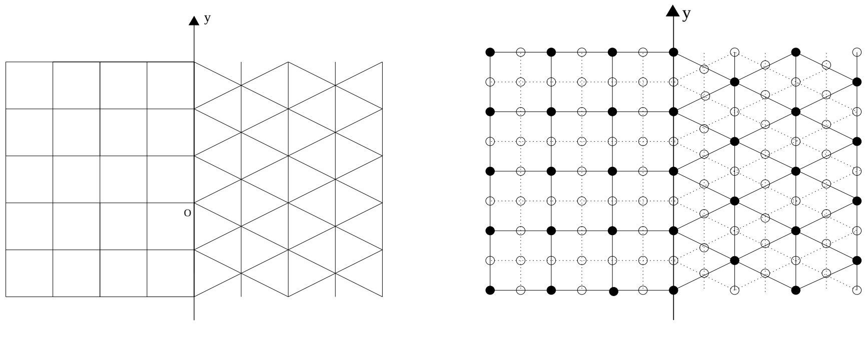

Figure 1: quad/triangle mesh (left) and refined quad/triangle mesh (right)

A subdivision algorithm for a *regular quad/triangle vertex*, a vertex surrounded by 2 adjacent quads and 3 adjacent triangles, can be represented in the parametric plane as that for the origin of the mesh in the left picture of Fig. 1, where all vertices on the y-axis are regular quad/trangle vertices. During the quad/triangle subdivision, "odd" vertices (denoted as $\circ$ in the right picture of Fig. 1) are inserted among the "even" vertices (denoted as $\bullet$), and then, they are connected appropriately such that each quad and triangle in the coarser mesh are split into 4 quads and 4 triangles in the finer mesh, see the right picture of Fig. 1. If the positions of "even" vertices are not changed during the subdivision process, then the subdivision scheme is called an *interpolatory scheme*. Otherwise, it is called an *approximation scheme*. The exact positions of the "odd" and "even" vertices in the finer mesh are given by the local averaging rule.

All the quad/triangle schemes available in the literature are approximating schemes. To construct an interpolatory quad/triangle scheme, one first needs to choose one interpolatory scheme for quad vertices and choose other interpolatory scheme for triangle vertices. Kobbelt's scheme in [17] and the butterfly scheme in [14] are probably the potential interpolatory schemes for quad vertices and for triangle vertices respectively. The problem is that these schemes are C^1 only, which implies that any Kobbelt-butterfly-scheme based interpolatory quad/triangle scheme is at most C^1. In addition, since both Kobbelt's scheme and the butterfly scheme (in particular the former) have big templates, a Kobbelt-butterfly-scheme based interpolatory quad/triangle scheme also has large sizes of templates.

Recently, vector (matrix-valued) subdivision for surface design is studied in [4]-[7], [15]. Matrix-valued subdivision is associated with a refinement equation

$$\boldsymbol{\phi}(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbf{Z}^2} G_{\mathbf{k}}\boldsymbol{\phi}(2\mathbf{x}-\mathbf{k}), \quad \mathbf{x}\in\mathbb{R}^2, \tag{1.1}$$

where $G_{\mathbf{k}}$ are $r\times r$ matrices with finitely many $G_{\mathbf{k}} \neq \mathbf{0}$, and $\boldsymbol{\phi} = [\phi_0, \cdots, \phi_{r-1}]^T$. In this paper for the purpose of simple presentation, we assume $r = 2$. Here we assume $\boldsymbol{\phi}$ (from now on it has two components) satisfies the partition unity property

$$\mathbf{y}_0 \sum_{\mathbf{k}} \boldsymbol{\phi}(\mathbf{x}-\mathbf{k}) = 1, \quad \mathbf{x}\in\mathbb{R}^2, \tag{1.2}$$

with

$$\mathbf{y}_0 = [1, 0]. \tag{1.3}$$

When the matrix-valued refinement equation (1.1) is applied to surface subdivisions, the local averaging rule for either the quad subdivision or triangle subdivision is

$$\mathbf{v}^{\ell+1}_{\mathbf{k}} = \sum_{\mathbf{j}} \mathbf{v}^{\ell}_{\mathbf{j}} G_{\mathbf{k}-2\mathbf{j}}, \quad \ell = 0, 1, \cdots, \tag{1.4}$$

where

$$\mathbf{v}^{\ell}_{\mathbf{k}} =: [v^{\ell}_{\mathbf{k}}, s^{\ell}_{\mathbf{k}}] \tag{1.5}$$

are "row-vectors" with 2 components of points $v^{\ell}_{\mathbf{k}}, s^{\ell}_{\mathbf{k}}$ in $\mathbb{R}^3$, 3×1 vectors. With $\boldsymbol{\phi}$ satisfying (1.2), as in [5, 6, 7], we may use the first components $v^{\ell}_{\mathbf{k}}$ of $\mathbf{v}^{\ell}_{\mathbf{k}}$ to denote the vertices of the subdivision meshes generated after ℓ steps of iterations, with $v^0_{\mathbf{k}}$ being the initial vertices on the control net. The other components $s^0_{\mathbf{k}}$ of $\mathbf{v}^0_{\mathbf{k}}$, can be used to control the surface geometric shape. In [5], the scheme is said to be interpolatory if $v^{\ell+1}_{2\mathbf{k}} = v^{\ell}_{\mathbf{k}}$, namely, the vertices on the coarse mesh remain in the refined mesh. With this definition of interpolatory, C^2 interpolatory quad subdivision schemes (for regular vertices) with the matrix-valued templates having the same sizes as those of Catmull-Clark's scheme and C^2 interpolatory triangle subdivision schemes (for regular vertices) with the matrix-valued templates having the same sizes as those of Loop's scheme are constructed in [5]. For example, with templates in Figs. 2 and 3, where

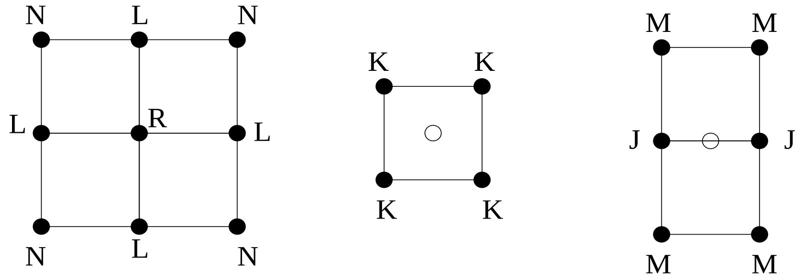

Figure 2: Matrix-valued templates of quad subdivision scheme for "even" vertices (left) and "odd" vertices (right)

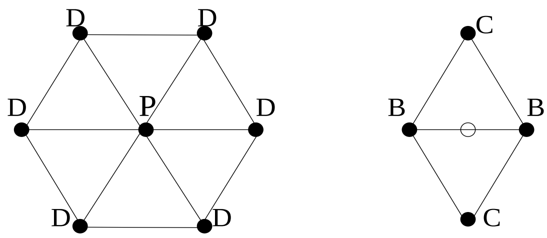

Figure 3: Matrix-valued templates of triangle subdivision scheme for "even" vertices (left) and "odd" vertices (right)

$$
\begin{gathered}
R = \begin{bmatrix} 1, & -\frac{129}{64} \\ 0, & -\frac{43}{128} \end{bmatrix}, \; J = \begin{bmatrix} \frac{3}{8}, & 0 \\ -\frac{11}{128}, & \frac{17}{128} \end{bmatrix}, \; K = \begin{bmatrix} \frac{1}{4}, & 0 \\ -\frac{1}{16}, & \frac{1}{16} \end{bmatrix}, \\
L = \begin{bmatrix} 0, & \frac{99}{256} \\ 0, & -\frac{33}{256} \end{bmatrix}, \; M = \begin{bmatrix} \frac{1}{16}, & 0 \\ -\frac{5}{256}, & -\frac{1}{256} \end{bmatrix}, \; N = \begin{bmatrix} 0, & \frac{15}{128} \\ 0, & -\frac{9}{256} \end{bmatrix},
\end{gathered} \tag{1.6}
$$

and

$$
P = \begin{bmatrix} 1, & -\frac{435}{256} \\ 0, & -\frac{91}{256} \end{bmatrix}, \; B = \begin{bmatrix} \frac{3}{8}, & 0 \\ -\frac{47}{512}, & \frac{69}{512} \end{bmatrix}, \; C = \begin{bmatrix} \frac{1}{8}, & 0 \\ -\frac{17}{512}, & -\frac{5}{512} \end{bmatrix}, \; D = \begin{bmatrix} 0, & \frac{145}{512} \\ 0, & -\frac{45}{512} \end{bmatrix}, \tag{1.7}
$$

one has an interpolatory quad scheme and an interpolatory triangle scheme with the associated refinable function vectors in C^2, see [5] (the matrices in (1.6) are slightly different from those in [5] by left and right multiplications of the matrix $\operatorname{diag}(1, -1)$).

Thus we should design interpolatory quad/triangle schemes based on these two interpolatory schemes. To construct/design C^1 and C^2 interpolatory quad/triangle schemes with matrix-valued templates, we need to generalize the smoothness estimate for quad/triangle schemes in [26] from the scalar-valued setting to the matrix-valued setting. For the smoothness analysis, one of key issues is about the polynomial reproduction of subdivision schemes. As shown in [25], the problem on the polynomial reproduction of a quad/triangle scheme is more complicated than a conventional quad or triangle scheme. On the other hand, the polynomial reproduction of a (conventional) matrix-valued scheme is more complicated than a scalar-valued scheme and there are many papers on this topic, see e.g. [16, 32, 1, 2, 23, 18]. Therefore, the study of polynomial reproduction of matrix-valued quad/triangle schemes is not a straightforward generalization of that of scalar-valued quad/triangle schemes or conventional matrix-valued schemes.

In our study of quad/triagle schemes, we find that these schemes are associated with the so-called nonhomogeneous refinement equations of the form

$$\boldsymbol{\phi}(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbf{Z}^2} A_{\mathbf{k}}\boldsymbol{\phi}(2\mathbf{x}-\mathbf{k}) + N_0(\mathbf{x}), \quad \mathbf{x}\in\mathbb{R}^2, \tag{1.8}$$

where $\boldsymbol{\phi} = [\phi_0, \phi_1, \cdots, \phi_{r'-1}]^T$ for some r' is also called the refinable function vector, and $N_0(\mathbf{x})$ is called the nonhomogeneous term. The nonhomogeneous refinement equation is generalized from their homogeneous counterpart, and it is motivated by the construction of multiwavelets to obtain multi-channel filters with good time-frequency localization and the construction of wavelets on a finite interval (see [36] and [8]). It has been studied by many researchers, see e.g. [12], [37], [13], [19], [20], [27], [28], [38] and [39]. In this paper we show that a quad/triangle subdivision scheme with either scalar-valued templates or matrix-valued templates can be derived from a nonhomogeneous refinement equation. Therefore, the polynomial reproduction of either a scalar-valued or a matrix-valued quad/triangle scheme can be treated in a uniform way. Considering the length of this paper, the detailed discussion on the polynomial reproduction of matrix-valued quad/triangle schemes is presented in our accompanying paper [24], where some related issues such as the analytical expression of the limiting surface are also addressed.

The rest of this paper is organized as following. In Section 2, firstly, we show that a quad/triangle scheme can be derived from a nonhomogeneous refinement equation. After that we recall some results on the polynomial reproduction obtained in [24]. In Section 3, we study the smoothness of matrix-valued quad/triangle schemes. Finally, in Section 4, we use the our smoothness estimate to construct C^1 and C^2 matrix-valued interpolatory quad/triangle schemes (for regular vertices), and then, apply these schemes for surface design.

In this paper we use the following notations. We use Γ_1 and Γ_2 to denote the subsets of $\mathbb{Z}^2$:

$$\begin{aligned} \Gamma_1 &:= \{\mathbf{n} = (n_1, n_2) : n_1 \le -2, n_1, n_2 \in \mathbb{Z}\}, \\ \Gamma_2 &:= \{\mathbf{m} = (m_1, m_2) : m_1 \ge 2, m_1, m_2 \in \mathbb{Z}\}. \end{aligned} \tag{1.9}$$

For $\mathbf{m} = (m_1, m_2) \in \mathbb{Z}^2$, denote

$$\widetilde{\mathbf{m}} := \begin{cases} \mathbf{m}, & \text{if } m_1 \text{ is even} \\ (m_1, m_2 - \frac{1}{2}), & \text{if } m_1 \text{ is odd.} \end{cases} \tag{1.10}$$

In the following, a Greek letter such as α denotes a multi-index $\alpha := (\alpha_1, \alpha_2) \in \mathbb{Z}_+^2$. For $\alpha = (\alpha_1, \alpha_2)$, let $|\alpha| := \alpha_1 + \alpha_2$.

2 Nonhomogeneous refinement equation, quad/triangle schemes and polynomial reproduction

In this section, first, we show that a quad/triangle scheme can be derived from a nonhomogeneous refinement equation. After that we recall the result on the polynomial reproduction obtained in [24].

2.1 Nonhomogeneous refinement equation and quad/triangle schemes

In this subsection, we show that either a scalar-valued or matrix-valued quad/triangular scheme can be derived from a nonhomogeneous refinement equation. For simplicity, let us focus on scalar-valued quad/triangle schemes.

Let $S(\mathbf{x})$ and $T(\mathbf{x})$ be compactly supported functions refinable (with dilation $2I_2$) along lattice $\mathbb{Z}^2$ and $\{(2j,k):\ j,k\in\mathbb{Z}\}\cup\{(2j+1,k+\frac{1}{2}):\ j,k\in\mathbb{Z}\}$ shown on the left and the right of Fig. 4 resp., namely, $S(\mathbf{x})$ and $T(\mathbf{x})$ satisfy

$$S(\mathbf{x})=\sum_{\mathbf{k}\in\mathbf{Z}^2} q_{\mathbf{k}}S(2\mathbf{x}-\mathbf{k}),\quad \mathbf{x}\in\mathbb{R}^2, \tag{2.1}$$

$$T(\mathbf{x})=\sum_{\mathbf{k}\in\mathbf{Z}^2} p_{\mathbf{k}}T(2\mathbf{x}-\widetilde{\mathbf{k}}),\quad \mathbf{x}\in\mathbb{R}^2, \tag{2.2}$$

for constants $q_{\mathbf{k}}, p_{\mathbf{k}}$ with finitely many nonzero, where for $\mathbf{k}\in\mathbb{Z}^2$, $\widetilde{\mathbf{k}}$ is defined in (1.10). For simplicity of presentation of the paper, we assume that $S(\mathbf{x})$ is supported on a neighborhood of the origin consisting of 2-ring quads (the shadowed region on the left of Fig. 4) and $T(\mathbf{x})$ is supported on a neighborhood of the origin consisting of 2-ring triangles (the shadowed region on the right of Fig. 4).

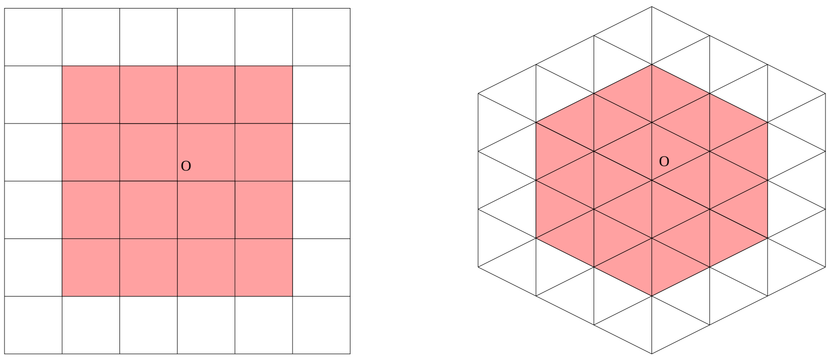

Figure 4: quad grid (left) and triangle grid (right)

As mentioned in §1, a triangle/quad subdivision algorithm for a regular quad/triangle vertex can be represented in the parametric plane with the mesh in the left picture of Fig. 1. Suppose mask $\{q_{\mathbf{k}}\}_{\mathbf{k}}$ is used for quad vertices on the left but far from the y-axis, and $\{p_{\mathbf{k}}\}_{\mathbf{k}}$ is used for triangle vertices on the right but far from the y-axis. Let $\varphi(\mathbf{x})$, $f(\mathbf{x})$ and $g(\mathbf{x})$ be compactly supported functions with $\mathrm{supp}(\varphi)\subset[-2,2]\times\mathbb{R}$, supp$(f) \subset [-3,1] \times \mathbb{R}$, supp$(g) \subset [-1,3] \times \mathbb{R}$. We consider such $\varphi(\mathbf{x}), f(\mathbf{x}), g(\mathbf{x})$ that they, together with $S(\mathbf{x}), T(\mathbf{x})$, generate a sequence of nested spaces of functions. More precisely, let

$$\begin{aligned} V_0 := \{&\varphi(x, y-k),\ f(x, y-k),\ g(x, y-k),\ S(\mathbf{x}-\mathbf{n}), T(\mathbf{x}-\widetilde{\mathbf{m}}) : \\ &k \in \mathbb{Z}, \mathbf{n} \in \Gamma_1, \mathbf{m} \in \Gamma_2\}, \end{aligned} \tag{2.3}$$

and $V_\ell := \{F(\mathbf{x}) :\ F(\frac{\mathbf{x}}{2^\ell}) \in V_0\}$. To assure that $\{V_\ell\}_\ell$ is nested, that is $V_0 \subset V_1 \subset V_2 \subset \cdots$, φ, f, g satisfy the following refinement relation:

$$\begin{aligned} \varphi(\mathbf{x}) = &\sum_{k \in \mathbf{Z}} a_{0,k}\varphi(2x, 2y-k) + \sum_{k \in \mathbf{Z}} a_{1,k} g(2x, 2y-k) + \\ &\sum_{k \in \mathbf{Z}} a_{-1,k} f(2x, 2y-k) + \sum_{\mathbf{n} \in \Gamma_1} a_{\mathbf{n}} S(2\mathbf{x}-\mathbf{n}) + \sum_{\mathbf{m} \in \Gamma_2} a_{\mathbf{m}} T(2\mathbf{x}-\widetilde{\mathbf{m}}), \end{aligned} \tag{2.4}$$

$$f(\mathbf{x}) = \sum_{k \in \mathbf{Z}} b_{0,k}\varphi(2x, 2y-k) + \sum_{k \in \mathbf{Z}} b_{-1,k} f(2x, 2y-k) + \sum_{\mathbf{n} \in \Gamma_1} b_{\mathbf{n}} S(2\mathbf{x}-\mathbf{n}), \tag{2.5}$$

$$g(\mathbf{x}) = \sum_{k \in \mathbf{Z}} d_{0,k}\varphi(2x, 2y-k) + \sum_{k \in \mathbf{Z}} d_{1,k} g(2x, 2y-k) + \sum_{\mathbf{m} \in \Gamma_2} d_{\mathbf{m}} T(2\mathbf{x}-\widetilde{\mathbf{m}}), \tag{2.6}$$

where $\mathbf{x} = (x, y)$, and $a_{\mathbf{n}}, b_{\mathbf{n}}, d_{\mathbf{n}}$ are some numbers (in this paper, we assume only finitely many of them are nonzero).

Denote

$$\Phi := [\varphi,\ f,\ g]^T,$$

and

$$\begin{aligned} N_1(x, y) &:= \sum_{\mathbf{n} \in \Gamma_1} a_{\mathbf{n}} S(2\mathbf{x}-\mathbf{n}) + \sum_{\mathbf{m} \in \Gamma_2} a_{\mathbf{m}} T(2\mathbf{x}-\widetilde{\mathbf{m}}), \\ N_2(x, y) &:= \sum_{\mathbf{n} \in \Gamma_1} b_{\mathbf{n}} S(2\mathbf{x}-\mathbf{n}), \\ N_3(x, y) &:= \sum_{\mathbf{m} \in \Gamma_2} d_{\mathbf{m}} T(2\mathbf{x}-\widetilde{\mathbf{m}}). \end{aligned} \tag{2.7}$$

Then equations (2.4)-(2.6) can be formulated as a nonhomogeneous refinement equation:

$$\Phi(x, y) = \sum_{k \in \mathbf{Z}} H_k \Phi(2x, 2y-k) + N(x, y), \tag{2.8}$$

where

$$N(\mathbf{x}) = [N_1(2\mathbf{x}),\ N_2(2\mathbf{x}),\ N_3(2\mathbf{x})]^T, \tag{2.9}$$

and

$$H_k = \begin{bmatrix} a_{0,k} & a_{-1,k} & a_{1,k} \\ b_{0,k} & b_{-1,k} & 0 \\ d_{0,k} & 0 & d_{1,k} \end{bmatrix}. \tag{2.10}$$

The nonhomogeneous refinement equation (2.8), together with refinement equations (2.1) and (2.2), yields immediately a quad/triangle subdivision algorithm shown below.

Quad/triangle subdivision algorithm For an initial quad/triangle control net with vertices $v^0_{\mathbf{k}}$, $\{H_k\}_k$, $\{q_{\mathbf{k}}\}_{\mathbf{k}}, \{p_{\mathbf{k}}\}_{\mathbf{k}}$ and $a_{\mathbf{k}}, b_{\mathbf{k}}, d_{\mathbf{k}}$ yield a quad/triangle subdivision algorithm with vertices $v_{\mathbf{k}}^{\ell+1}$ on the refined meshes obtained after $\ell+1$ steps of iterations given by

$$[v_{0,j}^{\ell+1}, v_{-1,j}^{\ell+1}, v_{1,j}^{\ell+1}] = \sum_{k\in\mathbf{Z}}[v_{0,k}^{\ell}, v_{-1,k}^{\ell}, v_{1,k}^{\ell}]H_{j-2k},\ j\in\mathbb{Z}, \tag{2.11}$$

$$v_{\mathbf{n}}^{\ell+1} = \sum_{k\in\mathbf{Z}}\left(v_{0,k}^{\ell}a_{\mathbf{n}-(0,2k)} + v_{-1,k}^{\ell}b_{\mathbf{n}-(0,2k)}\right) + \sum_{\mathbf{n}'\in\Gamma_1} v_{\mathbf{n}'}^{\ell}q_{\mathbf{n}-2\mathbf{n}'},\ \mathbf{n}\in\Gamma_1, \tag{2.12}$$

$$v_{\mathbf{m}}^{\ell+1} = \sum_{k\in\mathbf{Z}}\left(v_{0,k}^{\ell}a_{\mathbf{m}-(0,2k)} + v_{1,k}^{\ell}d_{\mathbf{m}-(0,2k)}\right) + \sum_{\mathbf{m}'\in\Gamma_2} v_{\mathbf{m}'}^{\ell}p_{\mathbf{m}-2\widetilde{\mathbf{m}}'},\ \mathbf{m}\in\Gamma_2. \tag{2.13}$$

Here $v_{0,j}^{\ell}$ are the positions of the quad/triangle vertices associated with points $(0, 2^{-\ell}j)$ on the y-axis, $v_{-1,j}^{\ell}$ and $v_{1,j}^{\ell}$ are the positions of the quad vertex and triangle vertex associated with points $(-2^{-\ell}, j2^{-\ell})$ and $(2^{-\ell}, 2^{-\ell}(j-\frac{1}{2}))$ resp., and $v_{\mathbf{n}}^{\ell}, \mathbf{n}\in\Gamma_1$ and $v_{\mathbf{m}}^{\ell}, \mathbf{m}\in\Gamma_2$ are the positions of the quad vertex and triangle vertex associated with points $2^{-\ell}\mathbf{n}$ and $2^{-\ell}\widetilde{\mathbf{m}}$ resp.

The above subdivision algorithm can be obtained by writing a function $F(\mathbf{x})$ in V_0 given by

$$F(\mathbf{x}) = \sum_{k\in\mathbf{Z}}[v_{0,k}^{0}, v_{-1,k}^{0}, v_{1,k}^{0}]\Phi(\mathbf{x}-(0,k)) + \sum_{\mathbf{n}\in\Gamma_1} v_{\mathbf{n}}^{0}S(\mathbf{x}-\mathbf{n}) + \sum_{\mathbf{m}\in\Gamma_2} v_{\mathbf{m}}^{0}T(\mathbf{x}-\widetilde{\mathbf{m}})$$

as

$$F(\mathbf{x}) = \sum_{k\in\mathbf{Z}}[v_{0,k}^{\ell}, v_{-1,k}^{\ell}, v_{1,k}^{\ell}]\Phi(2^{\ell}\mathbf{x}-(0,k)) + \sum_{\mathbf{n}\in\Gamma_1} v_{\mathbf{n}}^{\ell}S(2^{\ell}\mathbf{x}-\mathbf{n}) + \sum_{\mathbf{m}\in\Gamma_2} v_{\mathbf{m}}^{\ell}T(2^{\ell}\mathbf{x}-\widetilde{\mathbf{m}}),$$

for $\ell = 1, 2, \cdots$. Then refinement equations (2.8), (2.1) and (2.2) result in the subdivision algorithm.

Observe that since there are only finitely many $a_{\mathbf{k}}, b_{\mathbf{k}}, d_{\mathbf{k}}$ are nonzero, for $\mathbf{n} = (n_1, n_2)\in\Gamma_1$ and $\mathbf{m} = (m_1, m_2)\in\Gamma_2$ with $-n_1, m_1$ large enough, the first terms in both of (2.12) and (2.13) are zero. Therefore, the averaging rule (2.12) for the vertices on the left but far from the y-axis coincides with the conventional quad subdivision algorithm with mask $\{q_{\mathbf{k}}\}_{\mathbf{k}}$, while the averaging rule (2.13) for the vertices on the right but far from the y-axis is the ordinary triangle subdivision algorithm with mask $\{p_{\mathbf{k}}\}_{\mathbf{k}}$.

The local averaging rule given in (2.11), (2.12) and (2.13) can be represented by templates. For example, if $a_{\mathbf{k}} = 0, b_{\mathbf{k}} = 0, d_{\mathbf{k}} = 0$ for $|k_2| > 2$, then the algorithm (2.11) for "even" quad/triangle vertices $v_{0,2j}^{\ell}$ and "odd" quad/triangle vertices $v_{0,2j+1}^{\ell}$ on the y-axis is given by the templates on the left and right of Fig. 5.

In practice, the symmetry of templates are required. To design a particular quad/triangle scheme, one may start from symmetric templates for the vertices on/near the y-axis. The templates are given by some parameters. Then from the templates, one has the algorithm (2.11)-(2.13) with mask H_k and $a_{\mathbf{k}}, b_{\mathbf{k}}, d_{\mathbf{k}}$ given by parameters. Finally, one determines the parameters by the polynomial reproduction and smoothness of the scheme.

Similarly, a matrix-valued quad/triangle scheme can be derived from a nonhomogeneous refinement equation. Assume that $\{P_{\mathbf{k}}\}_{\mathbf{k}}$ and $\{Q_{\mathbf{k}}\}_{\mathbf{k}}$ be the matrix-valued masks for the quad and triangle schemes resp., and let $S(\mathbf{x}) = [S_1(\mathbf{x}), S_2(\mathbf{x})]^T, T(\mathbf{x}) =$

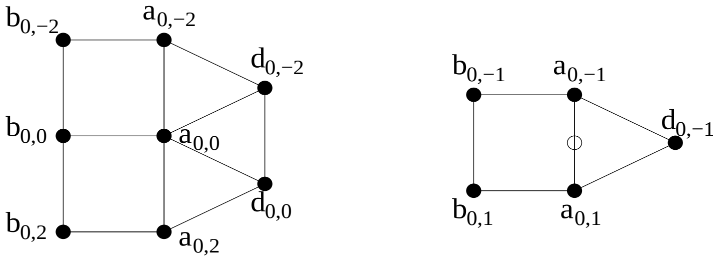

Figure 5: Templates for "even" vertices (left) and "odd" vertices (right) on the y-axis

$[T_1(\mathbf{x}), T_2(\mathbf{x})]^T$ be the associated refinable function vectors. Then a matrix-valued quad/triangle scheme can be derived from

$$\Phi(x, y) = \sum_{k \in \mathbf{Z}} H_k \Phi(2x, 2y - k) + N(x, y), \tag{2.14}$$

where

$$\Phi := [\varphi_0,\ \varphi_1,\ f_0,\ f_1,\ g_0,\ g_1]^T, \tag{2.15}$$

H_k are 6×6 matrices given as in (2.10) for some 2×2 matrices $\mathbf{a_k}, \mathbf{b_k}, \mathbf{d_k}$, and $N(\mathbf{x})$ is given as in (2.9) with each of its components $N_j(\mathbf{x})$ a vector given in (2.7). With

$$\begin{aligned} V_0 := \{&\varphi_j(x, y-k), f_j(x, y-k), g_j(x, y-k), S_j(\mathbf{x} - \mathbf{n}), \\ &T_j(\mathbf{x} - \widetilde{\mathbf{m}}) :\ j = 0, 1, k \in \mathbb{Z}, \mathbf{n} \in \Gamma_1, \mathbf{m} \in \Gamma_2\}, \end{aligned} \tag{2.16}$$

and $V_\ell := \{F(\mathbf{x}) :\ F(\frac{\mathbf{x}}{2^\ell}) \in V_0\}$, the nesting property of $\{V_j\}_j$ yields the matrix-valued quad/triagle subdivision algorithm given by the same formulas as in (2.11)-(2.13) with scalar-valued masks $\{p_\mathbf{k}\}_\mathbf{k}, \{q_\mathbf{k}\}_\mathbf{k}, \{a_\mathbf{k}\}_\mathbf{k}, \{b_\mathbf{k}\}_\mathbf{k}, \{d_\mathbf{k}\}_\mathbf{k}$ replaced by matrix-valued masks $\{P_\mathbf{k}\}_\mathbf{k}, \{Q_\mathbf{k}\}_\mathbf{k}, \{\mathbf{a_k}\}_\mathbf{k}, \{\mathbf{b_k}\}_\mathbf{k}, \{\mathbf{d_k}\}_\mathbf{k}$ resp..

In the rest of this subsection, as examples, we give the masks $\{H_k\}_k, a_\mathbf{k}, b_\mathbf{k}, d_\mathbf{k}$ for the C^1 (approximation) quad/triangle scheme constructed by Stam and Loop in [35] and the C^2 (approximation) quad/triangle scheme constructed by A. levin and D. Levin in [26]. In either case, Catmull-Clark's scheme (for regular vertices) is used for the vertices on the left of the y-axis and Loop's scheme (for regular vertices) is used for the vertices on the right of the y-axis. Namely, $S(\mathbf{x})$ is the bi-cubic spline (the tensor product of the cubic B-spline) and

$$T(\mathbf{x}) = B_{222}(x, y + \frac{x}{2}), \tag{2.17}$$

where B_{222} is the quartic box-spline on the 3-direction mesh. In this case, the nonzero $q_\mathbf{k}, p_\mathbf{k}$ in (2.1) and (2.2) are given resp. by

$$\begin{aligned} &q_{0,0} = \frac{9}{16},\ q_{1,0} = q_{0,1} = q_{-1,0} = q_{0,-1} = \frac{3}{8},\ q_{1,1} = q_{1,-1} = q_{-1,1} = q_{-1,-1} = \frac{1}{4}, \\ &q_{2,0} = q_{-2,0} = q_{0,2} = q_{0,-2} = \frac{3}{32},\ q_{2,2} = q_{-2,2} = q_{2,-2} = q_{-2,-2} = \frac{1}{64}, \\ &q_{2,1} = q_{1,2} = q_{-1,-2} = q_{-2,-1} = q_{2,-1} = q_{-2,1} = q_{-1,2} = q_{1,-2} = \frac{1}{16}, \end{aligned} \tag{2.18}$$

and

$$
\begin{aligned}
&p_{0,0}=\frac{5}{8},\ p_{1,0}=p_{0,1}=p_{-1,0}=p_{-1,1}=p_{0,-1}=p_{1,1}=\frac{3}{8},\\
&p_{2,0}=p_{1,-1}=p_{-1,-1}=p_{-2,0}=p_{-1,2}=p_{1,2}=\frac{1}{8},\\
&p_{2,1}=p_{0,2}=p_{-2,1}=p_{-2,-1}=p_{0,-2}=p_{2,-1}=\frac{1}{16}.
\end{aligned}
\tag{2.19}
$$

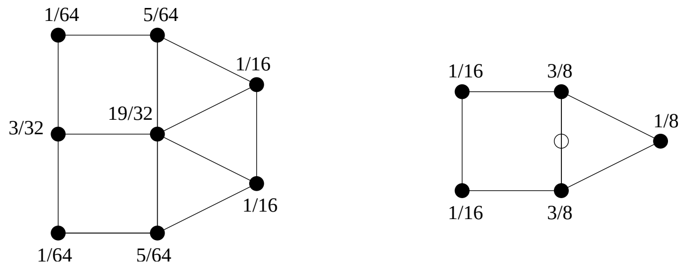

Figure 6: Templates of Stam-Loop's scheme for "even" vertices (left) and "odd" vertices (right) on the y-axis

Stam-Loop's scheme For Stam-Loop's scheme, Catmull-Clark's and Loop's schemes are used for the vertices on the left of the y-axis and on the right of the y-axis resp. The averaging rules for "even" vertices and "odd" vertices on the y-axis are given by the templates on the left and right of Fig. 6 resp. For Stam-Loop's scheme, the corresponding nonzero H_k and nonzero $a_{j,k}, b_{j,k}, d_{-j,k}$ $(j \geq 2)$ are listed as follows:

$$
\begin{aligned}
&H_{-2}=\frac{1}{64}\begin{bmatrix}5&4&0\\1&4&0\\4&0&8\end{bmatrix},\ H_{-1}=\frac{1}{16}\begin{bmatrix}6&4&2\\1&4&0\\2&0&6\end{bmatrix},\\
&H_{0}=\frac{1}{32}\begin{bmatrix}19&12&12\\3&12&0\\2&0&12\end{bmatrix},\ H_{1}=\frac{1}{16}\begin{bmatrix}6&4&6\\1&4&0\\0&0&2\end{bmatrix},\ H_{2}=\frac{1}{64}\begin{bmatrix}5&4&8\\1&4&0\\0&0&0\end{bmatrix},
\end{aligned}
\tag{2.20}
$$

$$
\begin{aligned}
&a_{2,0}=\frac{1}{8},\ a_{2,1}=a_{2,-1}=\frac{1}{16},\ a_{-2,0}=\frac{3}{32},\ a_{-2,1}=a_{-2,-1}=\frac{1}{16},\ a_{-2,2}=a_{-2,-2}=\frac{1}{64},\\
&b_{-2,0}=\frac{9}{16},\ b_{-2,1}=b_{-2,-1}=\frac{3}{8},\ b_{-2,2}=b_{-2,-2}=\frac{3}{32},\ b_{-3,0}=\frac{3}{8},\ b_{-3,1}=b_{-3,-1}=\frac{1}{4},\\
&b_{-3,2}=b_{-3,-2}=\frac{1}{16},\ b_{-4,0}=\frac{3}{32},\ b_{-4,1}=b_{-4,-1}=\frac{1}{16},\ b_{-4,2}=b_{-4,-2}=\frac{1}{64},\\
&d_{2,-1}=\frac{5}{8},\ d_{2,0}=d_{2,-2}=\frac{3}{8},\ d_{2,1}=d_{2,-3}=\frac{1}{16},\\
&d_{3,-1}=d_{3,-2}=\frac{3}{8},\ d_{3,0}=d_{3,-3}=\frac{1}{8},\ d_{4,-1}=\frac{1}{8},\ d_{4,0}=d_{4,-2}=\frac{1}{16}.
\end{aligned}
$$

Levin-Levin's scheme In Levi-Levin's scheme, Catmull-Clark's scheme used for quad vertices on the left of the y-axis, while Loop's scheme is used for triangle vertices

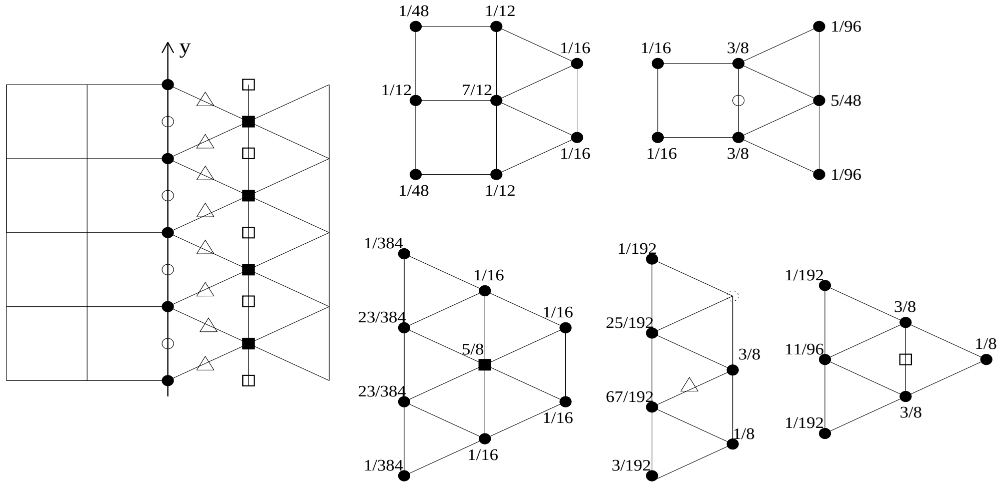

Figure 7: Templates of Levin-Levin's scheme for "even" vertices (top-left) and "odd" vertices (top-right) on the y-axis, for "even" vertices (bottom-left) and "odd" vertices (bottom-middle and bottom-right) on the right and near the y-axis.

$v^{\ell}_{(m_1,m_2)}$ with $m_1 \geq 3$. The averaging rules for "even" vertices $v^{\ell}_{0,2k}$ (denoted as black disks) "odd" vertices $v^{\ell}_{0,2k+1}$ (denoted hollow circles) on the y-axis, vertices $v^{\ell}_{1,k}$ (denoted as triangles) and vertices v^{ℓ}_{2,m_2} (denoted as squares) are given in Fig. 7. In this case the corresponding nonzero H_k are

$$
\begin{aligned}
H_{-3} &= \frac{1}{192}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}, \ H_{-2} = \frac{1}{192}\begin{bmatrix} 16 & 12 & 3 \\ 4 & 12 & 0 \\ 12 & 0 & 24 \end{bmatrix}, \\
H_{-1} &= \frac{1}{192}\begin{bmatrix} 72 & 48 & 25 \\ 12 & 48 & 0 \\ 20 & 0 & 72 \end{bmatrix}, \ H_{0} = \frac{1}{192}\begin{bmatrix} 112 & 72 & 67 \\ 16 & 72 & 0 \\ 12 & 0 & 72 \end{bmatrix}, \\
H_{1} &= \frac{1}{192}\begin{bmatrix} 72 & 48 & 67 \\ 12 & 48 & 0 \\ 2 & 0 & 24 \end{bmatrix}, \ H_{2} = \frac{1}{192}\begin{bmatrix} 16 & 12 & 25 \\ 4 & 12 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
H_{3} &= \frac{1}{192}\begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ H_{4} = \frac{1}{192}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},
\end{aligned}
\tag{2.21}
$$

and the corresponding nonzero $a_{j,k}, b_{j,k}, d_{j,k}$ are the same as those for the Stam-Loop's scheme except that

$$
a_{2,0} = \frac{11}{96}, \ a_{2,1} = a_{2,-1} = \frac{23}{384}, \ a_{2,2} = a_{2,-2} = \frac{1}{192}, \ a_{2,3} = a_{2,-3} = \frac{1}{384}.
$$

2.2 Polynomial reproduction

Polynomial reproduction of matrix-valued quad/triangle schemes has been studied in [24]. In this subsection, we will include a result from [24], which will be used in this paper.

Recall that for $G(\omega) = \frac{1}{2}\sum_{k\in\mathbf{Z}} G_k e^{-ik\omega}$, where $\{G_k\}_k$ is a 1-D matrix-valued mask with only finitely many of $r' \times r'$ matrices G_k being nonzero, we say $G(\omega)$ has sum rule of order $m+1$ if there exist $1 \times r'$ row constant vectors $\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_m$ with $\mathbf{y}_0 \neq [0, \cdots, 0]$ such that for $j = 0, \cdots, m$,

$$\sum_{n=0}^{j} \left(\begin{array}{c} j \\ n \end{array} \right) 2^n i^{j-n} \mathbf{y}_n (D^{j-n}G)(0) = \mathbf{y}_j, \ \sum_{n=0}^{j} \left(\begin{array}{c} j \\ n \end{array} \right) 2^n i^{j-n} \mathbf{y}_n (D^{j-n}G)(\pi) = 0,$$

where $D^{j-n}G(\omega)$ denotes the $(j-n)$-th derivative of $G(\omega)$.

Let $\{H_k\}_k$ be the 1-D matrix-valued mask for the quad/triangle scheme. Recall that each of H_k is a 6 by 6 matrix. Suppose that $H(\omega) := \frac{1}{2}\sum_{k\in\mathbf{Z}} H_k e^{-ik\omega}$ has sum rule of order $m+1$ with 1×6 vectors $\mathbf{y}_0^0, \mathbf{y}_1^0, \cdots, \mathbf{y}_m^0$. Furthermore, we assume for each j with $1 \leq j \leq m$, $2^j H(\omega)$ has sum rule of order $m+1-j$ with vectors $\mathbf{y}_0^j, \mathbf{y}_1^j, \cdots, \mathbf{y}_{m-j}^j$. Define for $k \in \mathbb{Z}$,

$$\begin{aligned}
Y^{0,l}(k) &= \sum_{n=0}^{l} \left(\begin{array}{c} l \\ n \end{array} \right) k^{l-n} \mathbf{y}_n^0, \ l = 0, 1, \cdots, m, \\
Y^{1,l}(k) &= \sum_{n=0}^{l} \left(\begin{array}{c} l \\ n \end{array} \right) k^{l-n} \mathbf{y}_n^1, \ l = 0, 1, \cdots, m-1, \\
&\vdots \\
Y^{m-1,l}(k) &= \sum_{n=0}^{l} \left(\begin{array}{c} l \\ n \end{array} \right) k^{l-n} \mathbf{y}_n^{m-1}, \ l = 0, 1, \\
Y^{m,0}(k) &= \mathbf{y}_0^m.
\end{aligned}$$

Then we have (see [24])

$$\sum_{k'\in\mathbf{Z}} Y^\alpha(k') H_{k-2k'} = \frac{1}{2^{|\alpha|}} Y^\alpha(k), \ k \in \mathbb{Z}, \ |\alpha| \leq m, \ \alpha \in \mathbb{Z}_+^2. \tag{2.22}$$

Write $Y^\alpha(k)$ as

$$Y^\alpha(k) =: [Y_1^\alpha(k), Y_2^\alpha(k), Y_3^\alpha(k)], \tag{2.23}$$

where $Y_j^\alpha(k), j = 1, 2, 3$ are 1×2 vectors.

Suppose that refinable functions $S(\mathbf{x})$ (for the quad subdivision) and $T(\mathbf{x})$ (for the triangle subdivision) generate polynomials of total degree up to m:

$$\sum_{\mathbf{n}\in\mathbf{Z}^2} \mathbf{u}_\mathbf{n}^\alpha S(\mathbf{x} - \mathbf{n}) = \mathbf{x}^\alpha, \ \mathbf{x} \in \mathbb{R}^2, \ |\alpha| \leq m, \tag{2.24}$$

and

$$\sum_{\mathbf{n}\in\mathbf{Z}^2} \tilde{\mathbf{u}}_\mathbf{n}^\alpha T(\mathbf{x} - \tilde{\mathbf{n}}) = \mathbf{x}^\alpha, \ \mathbf{x} \in \mathbb{R}^2, \ |\alpha| \leq m, \tag{2.25}$$

with 1×2 vectors $\mathbf{u}_\mathbf{n}^\alpha, \tilde{\mathbf{u}}_\mathbf{n}^\alpha$ satisfying

$$\sum_{\mathbf{k}\in\mathbf{Z}^2} \mathbf{u}_\mathbf{k}^\alpha Q_{\mathbf{n}-2\mathbf{k}} = \frac{1}{2^{|\alpha|}} \mathbf{u}_\mathbf{n}^\alpha, \quad \sum_{\mathbf{k}\in\mathbf{Z}^2} \tilde{\mathbf{u}}_\mathbf{k}^\alpha P_{\mathbf{n}-2\mathbf{k}} = \frac{1}{2^{|\alpha|}} \tilde{\mathbf{u}}_\mathbf{n}^\alpha, \mathbf{n} \in \mathbb{Z}^2. \tag{2.26}$$

Theorem 1. *[24] For a quad/triangle scheme with $\{H_k\}_k$, $\{Q_{\mathbf{k}}\}_{\mathbf{k}}$, $\{P_{\mathbf{k}}\}_{\mathbf{k}}$ and $\mathbf{a_k}, \mathbf{b_k}, \mathbf{d_k}$, assume both $\{Q_{\mathbf{k}}\}_{\mathbf{k}}$ and $\{P_{\mathbf{k}}\}_{\mathbf{k}}$ have sum rule of order $m+1$ with vectors $\mathbf{u}^\alpha, \widetilde{\mathbf{u}}^\alpha, |\alpha| \le m$, resp. Suppose for $0 \le j \le m$, $2^j H(\omega)$ has sum rule of order $m+1-j$ with vectors $\mathbf{y}_0^j, \mathbf{y}_1^j, \cdots, \mathbf{y}_{m-j}^j$. Let $\mathbf{u}_{\mathbf{k}}^\alpha$, $\widetilde{\mathbf{u}}_{\mathbf{k}}^\alpha$ and $Y^\alpha(k)$ be the vectors defined above with $Y_j^\alpha(k), j = 1, 2, 3$ given by (2.23). If*

$$\sum_{k\in\mathbf{Z}}\{Y_1^\alpha(k)a_{\mathbf{n}-(0,2k)} + Y_2^\alpha(k)b_{\mathbf{n}-(0,2k)}\} + \sum_{\mathbf{n}'\in\Gamma_1}\mathbf{u}_{\mathbf{n}'}^\alpha Q_{\mathbf{n}-2\mathbf{n}'} = \frac{1}{2^{|\alpha|}}\mathbf{u}_{\mathbf{n}}^\alpha,\ \mathbf{n}\in\Gamma_1, \tag{2.27}$$

and

$$\sum_{k\in\mathbf{Z}}\{Y_1^\alpha(k)\mathbf{a}_{\mathbf{m}-(0,2k)} + Y_3^\alpha(k)\mathbf{d}_{\mathbf{m}-(0,2k)}\} + \sum_{\mathbf{m}'\in\Gamma_2}\widetilde{\mathbf{u}}_{\mathbf{m}'}^\alpha P_{\mathbf{m}-2\widetilde{\mathbf{m}}'} = \frac{1}{2^{|\alpha|}}\widetilde{\mathbf{u}}_{\mathbf{m}}^\alpha,\ \mathbf{m}\in\Gamma_2, \tag{2.28}$$

hold, then V_0 defined by (2.16) reproduces polynomials of total degree up to m:

$$\sum_{k\in\mathbf{Z}} Y^\alpha(k)\Phi(\mathbf{x}-(0,k)) + \sum_{\mathbf{n}\in\Gamma_1}\mathbf{u}_{\mathbf{n}}^\alpha S(\mathbf{x}-\mathbf{n}) + \sum_{\mathbf{m}\in\Gamma_2}\widetilde{\mathbf{u}}_{\mathbf{m}}^\alpha T(\mathbf{x}-\widetilde{\mathbf{m}}) = \mathbf{x}^\alpha,\ |\alpha|\le m,$$

where Φ is defined by (2.15).

Remark 1. *Clearly (2.27) and (2.28) imply that for $|\alpha| \le m$, vectors*

$$\mathbf{l}^\alpha := [\mathbf{u}_{\mathbf{n}}^\alpha, Y_1^\alpha(k), Y_2^\alpha(k), Y_3^\alpha(k), \widetilde{\mathbf{u}}_{\mathbf{m}}^\alpha]_{\mathbf{n}\in\Gamma_1, k\in\mathbf{Z}, \mathbf{m}\in\Gamma_2} \tag{2.29}$$

are the (left) eigenvectors of the subdivision matrix $\mathbf{S}$, corresponding to the eigenvalue of $\frac{1}{2^{|\alpha|}}$. Furthermore, from Theorem 1, we know that if $\mathbf{l}^\alpha$ is the initial vector sequence, then the limiting function is a polynomial of degree $|\alpha|$.

3 Smoothness analysis

In this section, we provide a smoothness estimate on the matrix-valued quad/triangle schemes, a generalization of smoothness estimate in [26] from the scalar-valued setting to the matrix-valued setting.

As in [26], let $X \in \mathbb{R}^2$ be the set of all nodes of the regular unit quad/triangle in the left part of Fig. 1, namely,

$$X = \{(j,k) : k, j \in \mathbb{Z}, j \le 0\} \bigcup \{(2j,k) : j,k\in\mathbb{Z}, j>0\} \bigcup \{(2j+1, k+\frac{1}{2}) : j,k\in\mathbb{Z}, j>0\}.$$

Then, X satisfies

$$\mathcal{E}X = X, \quad 2X \subset X \quad \text{and} \quad \overline{\cup_{n=0}^\infty 2^{-n}X} = \mathbb{R}^2,$$

where $\mathcal{E}$ is the translation operator in the y-axis defined as $\mathcal{E} : (x, y) \to (x, y + 1)$. Clearly, $\mathcal{E}^{-1}$ is also a translation operator in the y-axis given by $\mathcal{E}^{-1} : (x, y) \to (x, y-1)$. Moreover, denote $X^1 = [-1, 1] \times (-\infty, \infty)$,

$$W^- = \left[-1, -\frac{1}{2}\right] \times [0,1], \quad W^+ = \left[\frac{1}{2}, \quad 1\right] \times [0,1], \tag{3.1}$$

and $W = W^- \cup W^+$. Then $X^1 = \cup_{n=0}^{\infty} \cup_{j\in\mathbf{Z}} 2^{-n}\mathcal{E}^j W$.

When we consider the smoothness of the limiting surface on the regular quad/triangle mesh, we need only to consider each component of the limiting surface. Hence, in this section, $\mathbf{v}_{\mathbf{k}}^{\ell} = [v_{\mathbf{k}}^{\ell}, s_{\mathbf{k}}^{\ell}]$ considered above will be a 2×1 vector, namely $v_{\mathbf{k}}^{\ell}, s_{\mathbf{k}}^{\ell}$ are real numbers, not points in $\mathbb{R}^3$. Denote

$$l(X)^2 = \{\mathbf{V} |\ \mathbf{V}:\ X \longrightarrow \mathbb{R} \times \mathbb{R}\}.$$

An element $V = \{[v_{\mathbf{k}}, s_{\mathbf{k}}]\}_{\mathbf{k}\in X}$ of $l(X)^2$ will be called a vector sequence (associated with X).

For a matrix-valued quad/triangle scheme, let $\mathbf{v}_{\mathbf{k}}^{\ell} = [v_{\mathbf{k}}^{\ell}, s_{\mathbf{k}}^{\ell}], \mathbf{k} \in X$ be the vectors after ℓ steps of subdivision iterations with this scheme from a control vector sequence $\mathbf{V}^0 = \{\mathbf{v}_{\mathbf{k}}^0\}_{\mathbf{k}\in X} \in l(X)^2$. Denote $\mathbf{V}^{\ell} = \{\mathbf{v}_{\mathbf{k}}^{\ell}\}_{\mathbf{k}\in X} \in l(X)^2$. Then,

$$\mathbf{V}^{\ell} = \mathbf{V}^{\ell-1}\mathbf{S},$$

where $\mathbf{S}$ is the subdivision operator (matrix), a linear operator on $l(X)^2$. We call that $\mathbf{S}$ is L_∞-convergent, if for every $\mathbf{V^0} \in l(X)$, there exists $F = [F_1, F_2]$, $F_1, F_2 \in C(\mathbb{R}^2)$ such that $F \not\equiv \mathbf{0}$ for at least one $\mathbf{V}^0$ and that

$$\lim_{n\to\infty} \|\mathbf{V^0 S}^n - F(2^{-n}\cdot)\|_{\infty, X\cap 2^n P} = 0,$$

where P is any open and bounded domain in $\mathbb{R}^2$. F is called the limit function, which is denoted $\mathbf{V^0 S}^\infty = F$. We say that $\mathbf{S}$ is C^m if $\mathbf{V^0 S}^\infty \in C^m(\mathbb{R}^2)$ for any $\mathbf{V^0} \in l(X)^2$, and that $\mathbf{S}$ is $C^{m+\alpha}$ $(0 < \alpha \leq 1)$ if the mth order derivatives of $\mathbf{V^0 S}^\infty$ are Hölder continuous of α for any $\mathbf{V^0} \in l(X)^2$. It was shown in [24] that the second component F_2 of F is the zero function, while the first component F_1 of F is given by

$$\begin{aligned} F_1(\mathbf{x}) = &\textstyle\sum_{k\in\mathbf{Z}} \mathbf{v}_{0,k}^0 [\varphi_0, \varphi_1]^T (x, y-k) + \mathbf{v}_{-1,k}^0 [f_0, f_1]^T (x, y-k) \\ &+ \mathbf{v}_{1,k}^0 [g_0, g_1]^T (x, y-k) + \textstyle\sum_{\mathbf{n}\in\Gamma_1} \mathbf{v}_{\mathbf{n}}^0 S(\mathbf{x} - \mathbf{n}) + \sum_{\mathbf{m}\in\Gamma_2} \mathbf{v}_{\mathbf{m}}^0 T(\mathbf{x} - \widetilde{\mathbf{m}}). \end{aligned} \tag{3.2}$$

The smoothness of $\mathbf{S}$ far away from the y-axis is determined by the smoothness of $S(\mathbf{x})$ and $T(\mathbf{x})$, while that near the y-axis is determined by the smoothness of φ_j, f_j and g_j, $j = 0, 1$. The smoothness of $S(\mathbf{x})$ and $T(\mathbf{x})$ can be determined by their associated $\{Q_{\mathbf{k}}\}_{\mathbf{k}}$ and $\{P_{\mathbf{k}}\}_{\mathbf{k}}$. In the following we assume that the subdivision scheme $\mathbf{S}$ is L_∞-convergent, and that $S(\mathbf{x})$ and $T(\mathbf{x})$ are smooth enough, namely, they are in $C^{m+\alpha}$ when we discuss $C^{m+\alpha}$ smoothness of $\mathbf{S}$. We assume that $\{Q_{\mathbf{k}}\}_{\mathbf{k}}, \{P_{\mathbf{k}}\}_{\mathbf{k}}$ have sum rules order (at least) $m + 1$. Therefore we need only consider its smoothness near the y-axis.

Let $L \subset X$ be the subset of nodes of the unit regular quad/triangle grid around the origin such that the limit (vector-valued) function F takes the (vector-)values in $[-1, 1] \times [0, 1]$ only depending on control nodes in L. Let A be the local subdivision operator taking (vector-)values in L to (vector-)values in L after one subdivision iteration, and let B be the operator taking (vector-)values in L to that in $\mathcal{E}L$. Define $J = \cup_{j\in\mathbf{Z}} \mathcal{E}^j L$, thus, by the definition of L, the limit function F in the strip: $X^1 = [-1, 1] \times (-\infty, \infty)$ only depends on control points in J.

For a control vector sequence $\mathbf{V}^0 = \{\mathbf{v}_{\mathbf{k}}^0\}_{\mathbf{k}}$, denote $\widetilde{\mathbf{V}}^0 = \{\mathbf{v}_{\mathbf{k}}^0\}_{\mathbf{k}\in J}$ associated with J. For $\mathbf{v}_{\mathbf{k}}^{\ell}$, the refined vectors after ℓ steps of subdivision with $\mathbf{S}$, denote $\widetilde{\mathbf{V}}^{\ell} = \{\mathbf{v}_{\mathbf{k}}^{\ell}\}_{\mathbf{k}\in J}$, the vector subsequence of $\mathbf{V}^\ell = \{\mathbf{v}_\mathbf{k}^\ell\}_\mathbf{k}$ restricted to J. Separate $\widetilde{\mathbf{V}}^\ell$ into different groups: $\widetilde{\mathbf{V}}_j^\ell$, $j \in \mathbb{Z}$, where $\widetilde{\mathbf{V}}_j^\ell$ are defined as

$$\widetilde{\mathbf{V}}_j^\ell = \{\mathbf{v}_\mathbf{k}^\ell\}_{\mathbf{k}\in\mathcal{E}^j L},$$

namely, for each j$\in \mathbb{Z}$, $\widetilde{\mathbf{V}}_j^\ell$ is a (finite) subset of $\widetilde{\mathbf{V}}^\ell$ consisting all $\mathbf{v}_\mathbf{k}^\ell$ with $\mathbf{k} \in \mathcal{E}^j L$.

Observe that $\widetilde{\mathbf{V}}_j^1$ can be given as

$$\cdots, \quad \widetilde{\mathbf{V}}_0^1 = \widetilde{\mathbf{V}}_0^0 A, \quad \widetilde{\mathbf{V}}_1^1 = \widetilde{\mathbf{V}}_0^0 B, \quad \widetilde{\mathbf{V}}_2^1 = \widetilde{\mathbf{V}}_1^0 A, \quad \widetilde{\mathbf{V}}_3^1 = \widetilde{\mathbf{V}}_1^0 B, \quad \cdots;$$

while $\widetilde{\mathbf{V}}_j^2$ can be written as

$$\begin{aligned}\cdots, \widetilde{\mathbf{V}}_0^2 &= \widetilde{\mathbf{V}}_1^0 A = \widetilde{\mathbf{V}}_0^0 A^2, \ \widetilde{\mathbf{V}}_1^2 = \widetilde{\mathbf{V}}_1^0 B = \widetilde{\mathbf{V}}_0^0 AB, \\ \widetilde{\mathbf{V}}_2^2 &= \widetilde{\mathbf{V}}_1^1 A = \widetilde{\mathbf{V}}_0^0 BA, \ \widetilde{\mathbf{V}}_3^2 = \widetilde{\mathbf{V}}_1^1 B = \widetilde{\mathbf{V}}_0^0 B^2, \cdots.\end{aligned}$$

In general, we have

$$\widetilde{\mathbf{V}}_{2k}^\ell = \widetilde{\mathbf{V}}_k^{\ell-1} A, \quad \widetilde{\mathbf{V}}_{2k+1}^\ell = \widetilde{\mathbf{V}}_k^{\ell-1} B. \tag{3.3}$$

Thus, with the notation

$$A_{\epsilon_i} = \begin{cases} A & \epsilon_i = 0, \\ B & \epsilon_i = 1, \end{cases}$$

we have

$$\begin{aligned}\widetilde{\mathbf{V}}_j^\ell &= \widetilde{\mathbf{V}}_{k_1}^{\ell-1} A_{\epsilon_{\ell-1}} = \widetilde{\mathbf{V}}_{k_2}^{\ell-2} A_{\epsilon_{\ell-2}} A_{\epsilon_{\ell-1}} = \cdots \\ &= \widetilde{\mathbf{V}}_i^0 A_{\epsilon_0} \cdots A_{\epsilon_{m-1}}\end{aligned} \tag{3.4}$$

for some $k_1, k_2, \cdots, i$, where $\epsilon_j \in \{0,1\}, 0 \le j \le m-1$. Therefore, restricted to J, the subdivision scheme $\mathbf{S}$ is determined by A and B.

For $0 < \alpha \le 1$, define the Hölder constant of a function vector $h = [h_1, h_2]^T$ in a domain $K \in \mathbb{R}^2$ by

$$H(h, \alpha, K) = \sup_{\mathbf{x},\mathbf{y}\in K, \mathbf{x}\neq\mathbf{y}} \frac{\|h(\mathbf{x}) - h(\mathbf{y})\|_\infty}{\|\mathbf{x}-\mathbf{y}\|^\alpha}. \tag{3.5}$$

We need the following lemma proved in [26].

Lemma 1. *[26] Let h be the bounded $C(\mathbb{R}^2)$ function. If for all $j \in \mathbb{Z}$, $n \ge 0$, $H(h, \alpha, 2^{-n}\mathcal{E}^j W) \le c$ for a constant $c > 0$ independent of n and j, then*

$$H(h, \alpha, X^1) < \infty. \tag{3.6}$$

In Lemma 1 and below, c denotes a constant (independent of n and j) which may be different from line to line.

Observe that the the limiting surface near the y-axis is given by the shifts of φ_j, f_j, g_j, $j = 0, 1$ along the y-axis and that each of $\varphi_j(\mathbf{x})$, $f_j(\mathbf{x})$, $g_j(\mathbf{x})$ can be expressed as linear combinations of $\varphi_0(2\mathbf{x}-(0,k)), \varphi_1(2\mathbf{x}-(0,k)), f_0(2\mathbf{x}-(0,k)), f_1(2\mathbf{x}-(0,k)), g_0(2\mathbf{x}-$

$(0,k)), g_1(2\mathbf{x}-(0,k))$ and $S(\mathbf{x}-\mathbf{k}), T(\mathbf{x}-\mathbf{k})$. Thus to consider the smoothness of the limiting function F near the y-axis, we need only consider that on a rectangle: $[-1,1]\times[0,1]$. Hence, in the following we assume the initial vector sequence $\mathbf{V}^0=\{\mathbf{v}^0_{\mathbf{k}}\}_{\mathbf{k}}$ satisfies $\mathbf{v}^0_{\mathbf{k}}=[0,0]$ if $\mathbf{k}\notin L$. This assumption and Lemma 1 imply that we only need to study the behaviors of F in the rectangles, $2^{-n}\mathcal{E}^jW$ with $0\le j\le 2^n$.

First, by the assumption that $\mathbf{S}$ is $C^{m+\alpha}$ away from the y-axis and the linearity and the local support property of $\mathbf{S}$, we have

$$H(D^m\mathbf{V}\mathbf{S}^\infty,\alpha,W)\le c\|\mathbf{V}\|_{\infty,L}, \tag{3.7}$$

for some $c>0$, where D^m denotes the differential operator of order m. And for any domain K and $\lambda>0$, we have

$$H(F(\lambda\cdot),\alpha,K)=\lambda^\alpha H(F,\alpha,\lambda K). \tag{3.8}$$

From (3.8) and $\mathbf{V}\mathbf{S}^\infty=(\mathbf{V}\mathbf{S}^n\mathbf{S}^\infty)(2^n\cdot)$, we have

$$H(D^m\mathbf{V}\mathbf{S}^\infty,\alpha,2^{-n}W)=2^{mn}2^{n\alpha}H(D^m\mathbf{V}\mathbf{S}^n\mathbf{S}^\infty,\alpha,W). \tag{3.9}$$

This, together with (3.7), implies

$$H(D^m\mathbf{V}\mathbf{S}^\infty,\alpha,2^{-n}W)\le 2^{n(m+\alpha)}c\|\mathbf{V}\mathbf{S}^n\|_{\infty,L}. \tag{3.10}$$

Thus, for any j,n, we have

$$H(D^m\mathbf{V}\mathbf{S}^\infty,\alpha,2^{-n}\mathcal{E}^jW)\le 2^{n(m+\alpha)}c\|\mathbf{V}\mathbf{S}^n\|_{\infty,\mathcal{E}^jL}. \tag{3.11}$$

For each $\alpha\in\mathbb{Z}^2_+$ with $|\alpha|\le m$, let $\widetilde{\mathbf{l}}^\alpha$ be the vector consisting of components of $\mathbf{l}^\alpha$ defined (2.29) with indexes lying in L, namely, $\widetilde{\mathbf{l}}^\alpha=\mathbf{l}^\alpha|_L$. Then from Remark 1, we know $\widetilde{\mathbf{l}}^\alpha$ are left eigenvectors of $S|_L$ with eigenvalues $2^{-|\alpha|}$. Denote

$$M=\operatorname{span}\{\widetilde{\mathbf{l}}^\alpha:\quad |\alpha|\le m\}. \tag{3.12}$$

Then for any control vector sequences in M, which is a linear combination of eigenvectors, its corresponding limiting surface on $[-1,1]\times[0,1]$ is a polynomial of (total) degree up to m.

For the space $l(L)^2$, $l(L)^2=M\bigoplus M^\perp$. $\widetilde{\mathbf{l}}^\alpha, |\alpha|\le m$ and a basis in the $M^\perp$ consist of a basis of $l(L)^2$. Under this basis of $l(L)^2$, the matrix representation of the operator A is

$$\widetilde{A}=\left[\begin{array}{cc} A_{0,0} & A_{0,1}\\ \mathbf{0} & A_{1,1}\end{array}\right], \tag{3.13}$$

where $A_{0,0}=diag(1,2^{-1},2^{-1},\cdots,2^{-m})$. Moreover, it is easy to get that the matrix representation of the operator B can be written as

$$\widetilde{B}=\left[\begin{array}{cc} B_{0,0} & B_{0,1}\\ \mathbf{0} & B_{1,1}\end{array}\right], \tag{3.14}$$

and $B_{0,0}$ is upper triangular matrix with the diagonal entries: $1,2^{-1},2^{-1},\cdots,2^{-m}$.

In the next theorem, as in [26], we give a Hölder smoothness estimate of F in terms of the (L_∞) joint spectral radius of $A_{1,1}, B_{1,1}$. For a finite set $\mathcal{A}$ of operators acting on a fixed finite-dimensional space V, the joint spectral radius $\rho_\infty(\mathcal{A})$ of $\mathcal{A}$ is defined by

$$\rho_\infty(\mathcal{A}) := \lim_{l\to\infty} \|\mathcal{A}^l\|_\infty^{1/l},$$

where

$$\|\mathcal{A}^l\|_\infty := \max\{\|A_1 \cdots A_l\| : A_n \in \mathcal{A}, 1 \le n \le l\}.$$

Here, the operator norm $\|\cdot\|$ is induced by the norm on V and the value of $\rho_\infty(\mathcal{A})$ does not depend on the choice of the norm. The joint spectral radius $\rho_\infty(\mathcal{A})$ was introduced by Rota and Strang [33], and it has been used for the existence and the smoothness analysis of refinable functions, see e.g., [10, 11, 9, 22, 21].

Theorem 2. *Suppose V_0 defined by (2.16) reproduces polynomials of total degree m with (2.27) and (2.28) holding. Assume that $\mathbf{S}$ has $C^{m+\alpha}$ smoothness far away from the y-axis with $0 < \alpha \le 1$. If $\rho_\infty(A_{1,1}, B_{1,1}) < 2^{-(m+\alpha)}$, then $\mathbf{S}$ is $C^{m+\alpha}$.*

Proof. By above discussion, we need only to consider $\mathbf{V}^0 = \{\mathbf{v}^0_{\mathbf{k}}\}_{\mathbf{k}}$ with $\mathbf{v}^0_{\mathbf{k}} = [0, 0]$ if $\mathbf{k} \notin L$. By Lemma 1, it is enough to prove that

$$H(D^m \mathbf{V}\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W) \le c \tag{3.15}$$

for all $0 \le j \le 2^n$, $n \ge 0$.

For the space $l(L)^2$, write $l(L)^2 = M \bigoplus M^\perp$ with M defined in (3.12). With $\widetilde{\mathbf{V}} = \{\mathbf{v}^0_{\mathbf{k}}\}_{\mathbf{k}\in L} \in l(L)^2$, write $\widetilde{\mathbf{V}} = \mathbf{V}^1 + \mathbf{V}^2$, where $\mathbf{V}^1 \in M$ and $\mathbf{V}^2 \in M^\perp$. Thus,

$$H(D^m \widetilde{\mathbf{V}}\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W) \le H(D^m \mathbf{V}^1\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W) + H(D^m \mathbf{V}^2\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W). \tag{3.16}$$

By the definition of the space M, we know that $\mathbf{V}^1\mathbf{S}^\infty$ is a polynomial of total degree no more than m, and hence, its mth order derivatives are either zero or constant. Therefore, $\mathbf{V}^1\mathbf{S}^\infty$ has zero Hölder exponent. Thus if we show

$$H(D^m \mathbf{V}^2\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W) \le c \tag{3.17}$$

for $0 \le j \le 2^n$, $n \ge 0$, then we have (3.15) and hence, we prove Theorem 2.

Next let us prove (3.17). By (3.11), we have

$$H(D^m \mathbf{V}^2\mathbf{S}^\infty, \alpha, 2^{-n}\mathcal{E}^j W) \le 2^{n(m+\alpha)} c \|\mathbf{V}^2\mathbf{S}^n\|_{\infty, \mathcal{E}^j L}. \tag{3.18}$$

Recall that for $\mathbf{V}^2\mathbf{S}^n$, $(\widetilde{\mathbf{V}^2\mathbf{S}^n})_j$ denotes the restriction of $\mathbf{V}^2\mathbf{S}^n$ to $\mathcal{E}^j L$. Then by (3.4), for j with $0 \le j \le 2^n$, we have

$$\begin{aligned}
(\widetilde{\mathbf{V}^2\mathbf{S}^n})_j &= (\widetilde{\mathbf{V}^2})_0 A_{\epsilon_0} A_{\epsilon_1} \cdots A_{\epsilon_{n-1}} \\
&= \mathbf{V}^2 A_{\epsilon_0} A_{\epsilon_1} \cdots A_{\epsilon_{n-1}} \\
&= \mathbf{V}^2 A_{\epsilon_0}|_{M^\perp} A_{\epsilon_1}|_{M^\perp} \cdots A_{\epsilon_{n-1}}|_{M^\perp}.
\end{aligned}$$

By the definition of ∞-norm joint spectral radius, we reach

$$\|\mathbf{V}^2\mathbf{S}^n\|_{\infty, \mathcal{E}^j L} \le c(\rho_\infty(\widetilde{A}|_{M^\perp}, \widetilde{B}|_{M^\perp}) + \epsilon)^n,$$

for some constant c independent of n. Moreover, by the structures of $\widetilde{A}$ and $\widetilde{B}$ as shown in (3.13) and (3.14) respectively, we know

$$\rho_\infty(\widetilde{A}|_{M^\perp}, \widetilde{B}|_{M^\perp}) = \rho_\infty(A_{1,1}, B_{1,1}).$$

Thus the assumption $\rho_\infty(A_{1,1}, B_{1,1}) < 2^{-(m+\alpha)}$ implies

$$\|\mathbf{V}^2\mathbf{S}^n\|_{\infty,\mathcal{E}^j L} \leq c2^{-n(m+\alpha)}.$$

This and (3.18) lead to (3.17), as desired. The proof is completed. □

4 C^1 and C^2 interpolatory quad/triangle schemes for regular vertices

In this section, we construct C^1 and C^2 matrix-valued interpolatory quad/triangle schemes, and apply them for surface design. C^2 interpolatory with templates given in Fig. 2 and C^2 interpolatory scheme with templates given in Fig. 3 are used for quad and triangle vertices not near the y-axis. The procedure to construct quad/triangle schemes is described as follows. First we give (symmetric) interpolatory masks such that the schemes can reproduce polynomials up to certain degrees. The masks are given by some parameters, and hence $A_{1,1}$ and $B_{1,1}$ in (3.13) and (3.14) are also given by parameters. The next step is to choose these parameters such that $\rho_\infty(\{A_{1,1}, B_{1,1}\})$ is as small as possible. It is difficult in general to find the joint spectral radius ρ_∞. Instead, we consider sequences of lower and upper estimates $\underline{\rho}^l_\infty$ and $\overline{\rho}^l_\infty$ which are defined as

$$\begin{aligned}\underline{\rho}^l_\infty &:= \max\{\rho(A_1 \cdots A_l)^{1/l} \;:\; A_n \in \{A_{1,1}, B_{1,1}\}, 1 \leq n \leq l\},\\ \overline{\rho}^l_\infty &:= \max\{\|A_1 \cdots A_l\|_2^{1/l} \;:\; A_n \in \{A_{1,1}, B_{1,1}\}, 1 \leq n \leq l\}\ ,\end{aligned}$$

and satisfy

$$\underline{\rho}^l_\infty \leq \rho_\infty(\{A_{1,1}, B_{1,1}\}) \leq \overline{\rho}^l_\infty,$$

where $\rho(A_1 \cdots A_l)$ means spectral radius of $A_1 \cdots A_l$, $\|\cdot\|_2$ denotes the spectral norm for matrix operators. Set

$$\boldsymbol{\sigma}_\infty := -\log_2 \rho_\infty(\{A_{1,1}, B_{1,1}\}), \quad \overline{\boldsymbol{\sigma}}^l := -\log_2 \underline{\rho}^l_\infty, \quad \underline{\boldsymbol{\sigma}}^l := -\log_2 \overline{\rho}^l_\infty.$$

From Theorem 2, we know the limiting surface $F(\mathbf{x})$ is in $C^{\boldsymbol{\sigma}_\infty - \epsilon}$ for any $\epsilon > 0$. From $\underline{\boldsymbol{\sigma}}^l \leq \boldsymbol{\sigma}_\infty$, we have the estimate: for $\epsilon > 0$,

$$F(\mathbf{x}) \in C^{\underline{\boldsymbol{\sigma}}^l - \epsilon},$$

for any $l > 0$.

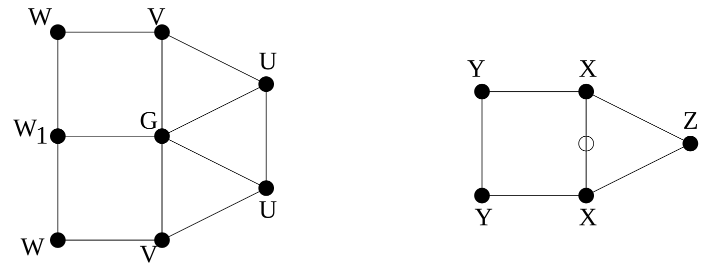

Figure 8: Templates of C^1 interpolatory scheme for "even" vertices (left) and "odd" vertices (right) on the y-axis

C^1-interpolatory scheme We consider a quad/triangle scheme with matrix-valued templates for nodes on the y-axis shown in Fig. 8, where

$$
\begin{aligned}
G &= \begin{bmatrix} 1 & 2\eta_2 - 22\eta_1 - 1 \\ 0 & \frac{1}{4} - 8\eta_1 \end{bmatrix},\ V = \begin{bmatrix} 0 & \frac{1}{2} + \eta_1 - \eta_2 \\ 0 & -\eta_1 \end{bmatrix}, \\
U &= \begin{bmatrix} 0 & 5\eta_1 \\ 0 & -\frac{5}{2}\eta_1 \end{bmatrix},\ W = \begin{bmatrix} 0 & 2\eta_1 \\ 0 & -\eta_1 \end{bmatrix},\ W_1 = \begin{bmatrix} 0 & 6\eta_1 \\ 0 & -3\eta_1 \end{bmatrix}, \\
X &= \begin{bmatrix} \frac{3}{8} & 0 \\ -6\eta_2 & -2\eta_2 \end{bmatrix},\ Y = \begin{bmatrix} \frac{1}{16} & 0 \\ 3\eta_2 - \frac{1}{16} & \eta_2 + \frac{1}{16} \end{bmatrix},\ Z = \begin{bmatrix} \frac{1}{8}, & 0 \\ 6\eta_2 - \frac{1}{8}, & 2\eta_2 + \frac{1}{8} \end{bmatrix},
\end{aligned}
$$

with $\eta_1, \eta_2 \in \mathbb{R}$. This scheme has the same sizes of templates as Stam-Loop's scheme. Because of the special structure of G, U, V, W, W_1, we know this is an interpolatory scheme, namely $v_{2\mathbf{k}}^{\ell+1} = v_{\mathbf{k}}^{\ell}$, where $v_{\mathbf{k}}^{\ell}$ are the first components of the vectors $[v_{\mathbf{k}}^{\ell}, s_{\mathbf{k}}^{\ell}]$ after ℓ steps of subdivision iterations. The associated (vector-valued) basis function $\Phi = [\varphi_0, \varphi_1, f_0, f_1, g_0, g_1]^T$ satisfies the nonhomogeneous refinement equation with nonzero $H_k, \mathbf{a_k}, \mathbf{b_k}, \mathbf{d_k}$ are

$$
H_{-2} = \begin{bmatrix} V, & M, & 0 \\ W, & M, & 0 \\ U, & 0, & C \end{bmatrix}, H_{-1} = \begin{bmatrix} X, & K, & C \\ Y, & K, & 0 \\ Z, & 0, & B \end{bmatrix}, H_0 = \begin{bmatrix} G, & J, & B \\ W_1, & J, & 0 \\ U, & 0, & B \end{bmatrix},
$$

$$
H_1 = \begin{bmatrix} X, & K, & B \\ Y, & K, & 0 \\ 0, & 0, & C \end{bmatrix}, H_2 = \begin{bmatrix} V, & M, & C \\ W, & M, & 0 \\ 0, & 0, & 0 \end{bmatrix},
$$

and

$$
\begin{aligned}
&\mathbf{a}_{2,0} = C,\ \mathbf{a}_{2,1} = \mathbf{a}_{2,-1} = D,\ \mathbf{a}_{-2,0} = L,\ \mathbf{a}_{-2,1} = \mathbf{a}_{-2,-1} = N,\ \mathbf{a}_{-2,2} = \mathbf{a}_{-2,-2} = M, \\
&\mathbf{b}_{-2,0} = R,\ \mathbf{b}_{-2,1} = \mathbf{b}_{-2,-1} = J,\ \mathbf{b}_{-2,2} = \mathbf{b}_{-2,-2} = L, \\
&\mathbf{b}_{-3,0} = J,\ \mathbf{b}_{-3,1} = \mathbf{b}_{-3,-1} = K,\ \mathbf{b}_{-3,2} = \mathbf{b}_{-3,-2} = M, \\
&\mathbf{b}_{-4,0} = L,\ \mathbf{b}_{-4,1} = \mathbf{b}_{-4,-1} = M,\ \mathbf{b}_{-4,2} = \mathbf{b}_{-4,-2} = N, \\
&\mathbf{d}_{2,-1} = P,\ \mathbf{d}_{2,0} = \mathbf{d}_{2,-2} = B,\ \mathbf{d}_{2,1} = \mathbf{d}_{2,-3} = D, \\
&\mathbf{d}_{3,-1} = \mathbf{d}_{3,-2} = B,\ \mathbf{d}_{3,0} = \mathbf{d}_{3,-3} = C,\ \mathbf{d}_{4,-1} = C,\ \mathbf{d}_{4,0} = \mathbf{d}_{4,-2} = D.
\end{aligned}
$$

This scheme reproduces linear polynomials (but not all quadratic polynomials), see [24] for the details. For η_1, η_2 near $(0,0)$, resulting scheme is C^1. On the left of Fig. 9, we show $\underline{\boldsymbol{\sigma}}^{10}$ for different η_1, η_2. We find if we choose

$$\eta_1 = \frac{7}{64}, \ \eta_2 = \frac{1}{128}, \tag{4.1}$$

then $\underline{\boldsymbol{\sigma}}^{10} = 1.4604$. $\underline{\boldsymbol{\sigma}}^{10}$ for (η_1, η_2) near $(\frac{7}{64}, \frac{1}{128})$ is given on the right of Fig. 9. We also calculate that $\underline{\boldsymbol{\sigma}}^{18} = 1.7000$ for the scheme with η_1, η_2 in (4.1). Thus we conclude that this scheme is at least in $C^{1.7000}$. We also observe that for this scheme, $\overline{\boldsymbol{\sigma}}^l = 2.0000, 1 \leq l \leq 18$.

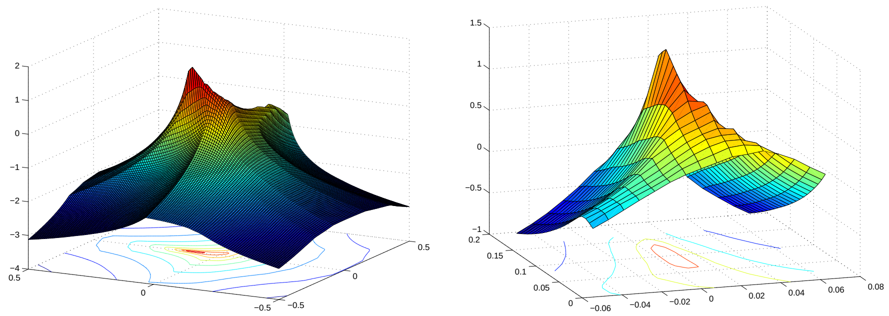

Figure 9: $\underline{\boldsymbol{\sigma}}^{10}$ near the origin (left) and near $\eta_1 = \frac{7}{64}$, $\eta_2 = \frac{1}{128}$ (right)

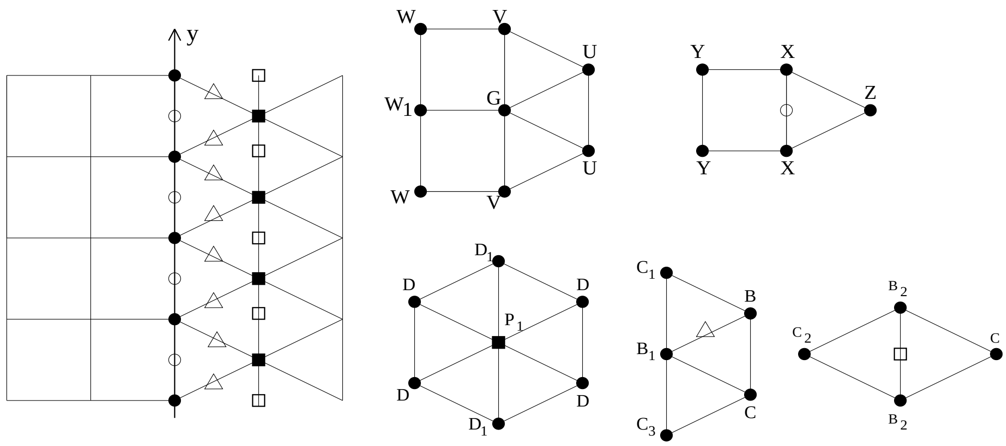

Figure 10: Templates of C^2 interpolatory scheme for "even" vertices (top-left) and "odd" vertices (top-right) on the y-axis, templates for "even" vertices (bottom-left), and "odd" vertices (bottom-middle and bottom-right) on right and near the y-axis

C^2-interpolatory scheme To construct C^2 interpolatory scheme, we consider a scheme with matrix-valued templates for nodes near the y-axis shown in Fig. 10, where

$$W = \begin{bmatrix} 0 & 7t_1 + t_2 \\ 0 & -4t_1 \end{bmatrix}, \ W_1 = \begin{bmatrix} 0, & 6t_2 - 9t_1 \\ 0, & -t_1 - t_2 \end{bmatrix}, \ V = \begin{bmatrix} 0, & t_1 + 5t_2 \\ 0, & -16t_1 \end{bmatrix},$$

$$U=\begin{bmatrix}0, & \frac{5}{2}t_1+4t_2\\ 0, & \frac{29}{2}t_1-4t_2\end{bmatrix},\ G=\begin{bmatrix}1, & -12t_1-26t_2\\ 0, & \frac{1}{4}+2t_1-7t_2\end{bmatrix},\ X=\begin{bmatrix}\frac{3}{8}, & 4t_1\\ -7t_1, & 7t_1\end{bmatrix},$$

$$Y=\begin{bmatrix}\frac{1}{16}, & -2t_1\\ 7t_1-\frac{1}{16} & \frac{1}{8}-21t_1\end{bmatrix},\ Z=\begin{bmatrix}\frac{1}{8}, & -4t_1\\ -\frac{1}{8}, & 36t_1\end{bmatrix},\ B_1=\begin{bmatrix}\frac{11}{32}, & \frac{1}{32}\\ 2t_1-t_2, & t_2\end{bmatrix},$$

$$C_1=\begin{bmatrix}\frac{9}{64}, & -\frac{1}{64}\\ -4t_1, & t_1\end{bmatrix},\ C_2=\begin{bmatrix}\frac{1}{8}, & 0\\ 0, & 0\end{bmatrix},\ C_3=\begin{bmatrix}\frac{1}{64}, & -\frac{1}{64}\\ 2t_1+t_2-\frac{1}{8}, & \frac{1}{8}-t_1-t_2\end{bmatrix},$$

$$B_2=\begin{bmatrix}\frac{3}{8}, & 0\\ -\frac{111}{1024}, & \frac{133}{1024}\end{bmatrix},\quad D_1=\begin{bmatrix}0, & \frac{625}{2048}\\ 0, & 6t_1\end{bmatrix},\quad P_1=\begin{bmatrix}1, & -\frac{1785}{1024}\\ 0, & -\frac{17}{32}-12t_1\end{bmatrix}.$$

The nonzero H_k in the nonhomogeneous refinement equation is

$$H_{-2}=\begin{bmatrix}V, & M, & C_3\\ W, & M, & 0\\ U, & 0, & C\end{bmatrix},H_{-1}=\begin{bmatrix}X, & K, & C_1\\ Y, & K, & 0\\ Z, & 0, & B\end{bmatrix},H_0=\begin{bmatrix}G, & J, & B_1\\ W_1, & J, & 0\\ U, & 0, & B\end{bmatrix},$$

$$H_1=\begin{bmatrix}X, & K, & B_1\\ Y, & K, & 0\\ 0, & 0, & C\end{bmatrix},H_2=\begin{bmatrix}V, & M, & C_1\\ W, & M, & 0\\ 0, & 0, & 0\end{bmatrix},H_3=\begin{bmatrix}0, & 0, & C_3\\ 0, & 0, & 0\\ 0, & 0, & 0\end{bmatrix},$$

and the corresponding nonzero $\mathbf{a}_{j,k},\mathbf{b}_{j,k},\mathbf{d}_{j,k}$ are the same as those for the Example 1 except that

$$\mathbf{a}_{2,0}=C_2,\ \mathbf{d}_{2,0}=\mathbf{d}_{2,-2}=B_2,\ \mathbf{d}_{2,-1}=P_1,\ \mathbf{d}_{2,1}=\mathbf{d}_{2,-3}=D_1.$$

This scheme reproduces all quadratic polynomials, see [24] for the detailed discussion. For t_1,t_2 near $(0,0)$, resulting scheme is C^2. On the left of Fig. 11, we show $\underline{\boldsymbol{\sigma}}^{10}$ for different t_1,t_2. We find if we choose

$$t_1=\frac{1}{128},\ t_2=\frac{27}{256},$$

then $\underline{\boldsymbol{\sigma}}^{10}=2.2338$. $\underline{\boldsymbol{\sigma}}^{10}$ for (t_1,t_2) near $(\frac{1}{128},\frac{27}{256})$ is given on the right of Fig. 11. We also calculate that $\underline{\boldsymbol{\sigma}}^{18}=2.5241$ for this scheme. Thus we conclude that this scheme is at least in $C^{2.5241}$. We also observe that for this scheme, $\overline{\boldsymbol{\sigma}}^{2l-1}=2.8710,\overline{\boldsymbol{\sigma}}^{2l}=2.8691, 1\le l\le 9$.

In the rest of this section, we apply the above C^1 interpolatory quad/triangle scheme to two very simple initial control quad/triangle nets. There are a few extraordinary quad/triangle vertices on these two nets. Recall that a quad/triangle vertex is called regular if it is surrounded by 2 adjacent quads and 3 adjacent triangles. Otherwise it is called an extraordinary quad/triangle vertex. Here we will not provide C^1 schemes for extraordinary quad/triangle vertices, which turns out to be a difficult problem. Instead, in the following we just follow the ideas in [35] to provide a 1-ring interpolatory scheme for extraordinary quad/triangle vertices with its C^1 smoothness not proved. Next, let us state the scheme for extraordinary quad/triangle vertices in [35].

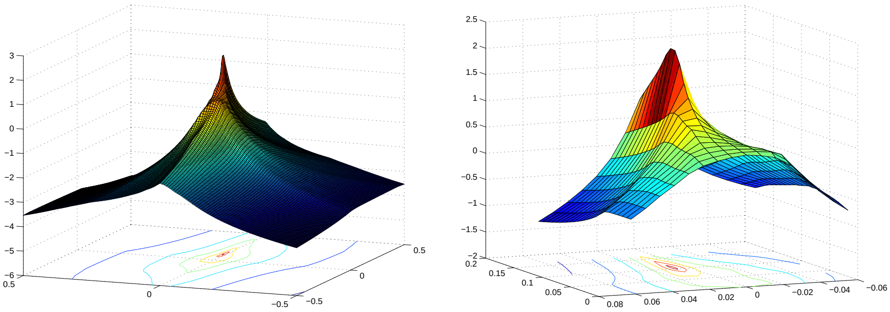

Figure 11: $\underline{\boldsymbol{\sigma}}^{10}$ near the origin (left) and near $t_1 = \frac{1}{128}$, $t_2 = \frac{27}{256}$ (right)

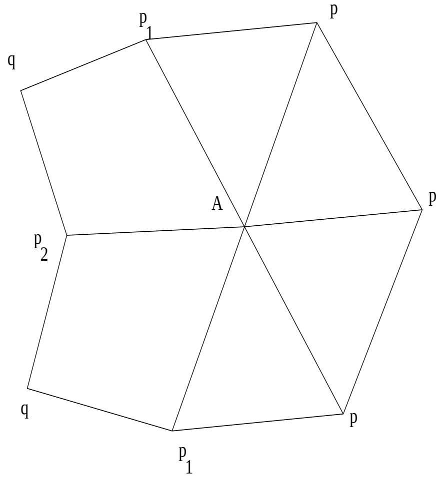

Figure 12: Template of local averaging rule for extraordinary quad/triangle vertices

Let n_e be the number of edges emanating from an extraordinary vertex and let n_q be the number of quads surrounding this vertex. Then the scheme for this extraordinary vertex in [35] is given as follows: the weight A for the extraordinary vertex, the weights p, p_1, p_2 for for the vertex between two triangles, for the vertex between one triangle and one quad, and for the vertex between two quads respectively, the weight q for the vertex which is on one quad and opposite to the extraordinary vertex (see Fig. 12) are given by

$$A = 2\beta + \frac{\beta}{2}n_e + \frac{\beta}{8}n_q, \; p = \frac{\beta}{2}, \; p_1 = \frac{5}{8}\beta, \; p_2 = \frac{3}{4}\beta, \; q = \frac{1}{8}\beta,$$

where β is determined by $2\beta + \beta n_e + \frac{1}{2}\beta n_q = 1$, namely,

$$\beta = \frac{1}{2 + n_e + n_q/2}. \tag{4.2}$$

For the matrix-valued C^1 interpolatory quad/triangle scheme for regular vertices constructed above, the weights A, p, p_1, p_2 and q for extraordinary vertices are 2×2 matrices. Here we choose

$$A = \begin{bmatrix} 1 & \gamma\xi_1 \\ 0 & \gamma\xi_2 \end{bmatrix}, \; p = \frac{\beta}{2}\boldsymbol{\tau}, \; p_1 = \frac{5}{8}\beta\boldsymbol{\tau}, \; p_2 = \frac{3}{4}\beta\boldsymbol{\tau}, \; q = \frac{1}{8}\beta\boldsymbol{\tau},$$

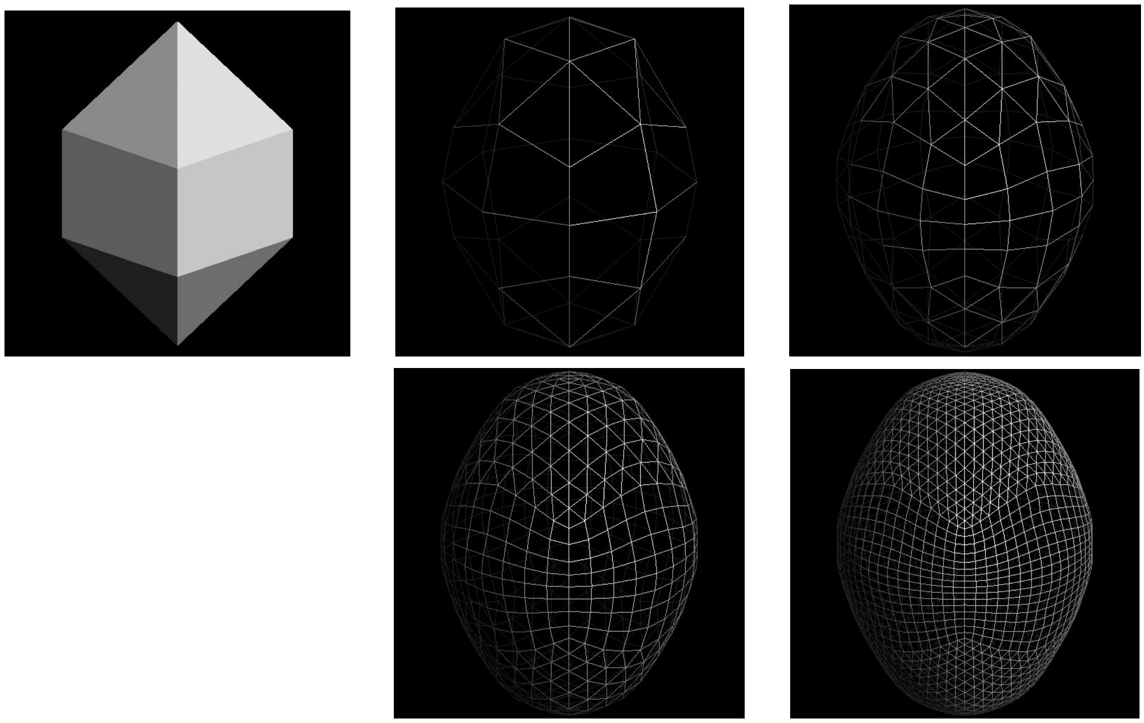

Figure 13: Initial quad/triangle net (top-left) and subdivided quad/triangle meshes

where β is define by (4.2),

$$\boldsymbol{\tau} = \begin{bmatrix} 0, & \xi_1 \\ 0, & \xi_3 \end{bmatrix},$$

for some $\xi_1, \xi_2, \xi_3 \in \mathbb{R}$, and

$$\gamma = -(\frac{\beta}{2}n_e + \frac{3}{8}\beta n_q). \tag{4.3}$$

The choice of γ in (4.3) assures $[\mathbf{y}_0, \mathbf{y}_0, \cdots, \mathbf{y}_0]$ with $\mathbf{y}_0 = [1, 0]$ to be the left 1-eigenvector of the subdivision matrix. There are many different choices of ξ_1, ξ_2, ξ_3. Here we simply choose $\xi_1 = 10, \xi_2 = \frac{14}{11}$ and $\xi_3 = -\frac{7}{2}$ based on the values in the entries of matrices D, J, L given in (1.6) and (1.7).

When we apply this C^1 interpolatory quad/triangle scheme to the initial quad/triangle net in the top-left of Fig. 13 with the shape control parameters $s_k = -2v_k$, we have subdivided polyhedra with iteration steps $1, 2, 3$ and 4 shown in Fig. 13. In this initial net, there are two extraordinary triangle vertices v_k for which the interpolatory local averaging rule for extraordinary triangle vertices in [7] is applied. Fig. 14 shows another example with this interpolatory quad/triagle scheme applied to an initial quad/triangle net (in top-left) with $s_k = t_k v_k$, where $t_k = -\frac{3}{5}$ for 4 vertices on the top quad and 4 vertices on the bottom quad of the net, and $t_k = -\frac{1}{2}$ for other 16 vertices. The choice of s_k is another issue we need to consider. Results on this topic will appear elsewhere.

Acknowledgments. The authors thank two anonymous referees for their valuable suggestions and comments.

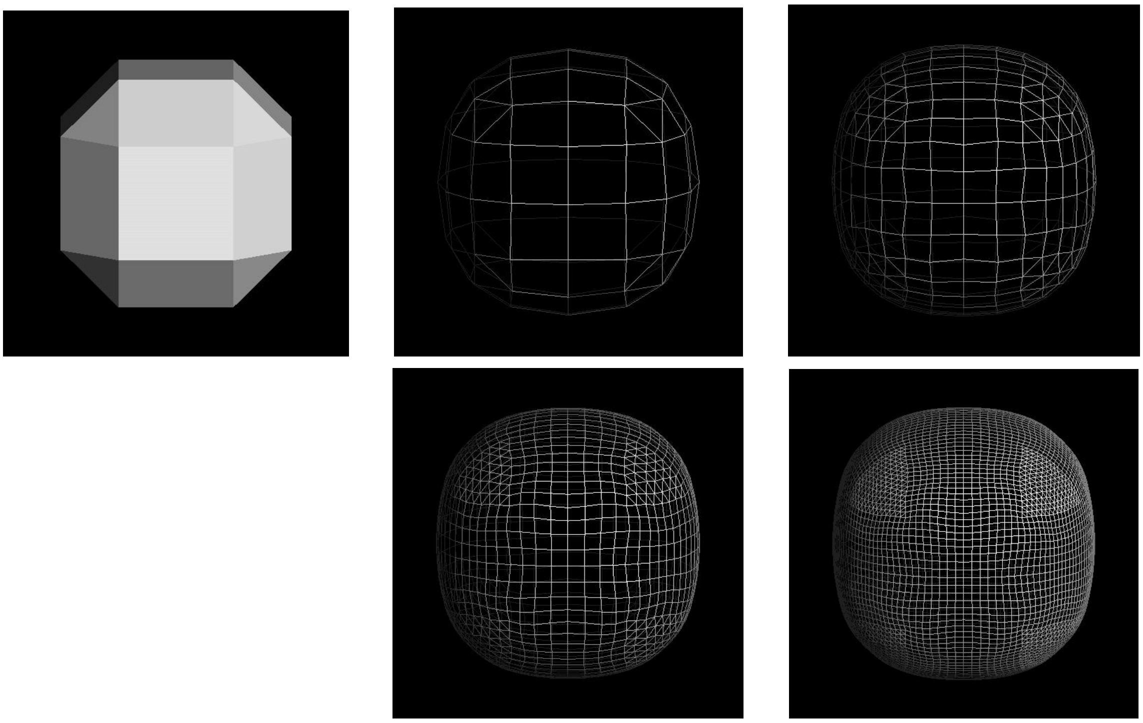

Figure 14: Initial quad/triangle net (top-left) and subdivided quad/triangle meshes

References

[1] C. de Boor, R. DeVore, and A. Ron, Approximation orders of FSI spaces in $L_2(\mathbb{R}^d)$, *Constr. Approx.* **14** (1998), 631–652.

[2] C. Cabrelli, C. Heil, and U. Molter, Accuracy of lattice translates of several multidimensional refinable functions, *J. Approx. Theory* **95** (1998), 5–52.

[3] E. Catmull and J. Clark, Recursively generated b-spline surfaced on arbitrary topological meshes, *Comput. Aided Design* **10** (1978), 350–355.

[4] C.K. Chui and Q.T. Jiang, Surface subdivision schemes generated by refinable bivariate spline function vectors, *Appl. Comput. Harmonic Anal.* **15** (2003), 147–162.

[5] C.K. Chui and Q.T. Jiang, Matrix-valued symmetric templates for interpolatory surface subdivisions I. Regular vertices, *Appl. Comput. Harmonic Anal.* **19** (2005), 303–339.

[6] C.K. Chui and Q.T. Jiang, Matrix-valued subdivision schemes for generating surfaces with extraordinary vertices, *Comput. Aided Geom. Design* **23** (2006), 419–438.

[7] C.K. Chui and Q.T. Jiang, From extension of Loop's approximation scheme to interpolatory subdivisions, *Comput. Aided Geom. Design* **25** (2008), 96–115.

[8] A. Cohen, I. Daubechies, and P. Vial, Wavelets on the interval and fast wavelet transform, *Appl. Comp. Harmonic Anal.* **1** (1993), 54–86.

[9] D. Colella and C. Heil, Characterizations of scaling functions: Continuous solutions, *SIAM J. Matrix Anal. Appl.* **15** (1994), 496–518.

[10] I. Daubechies and J.C. Lagarias, Two-scale difference equations. I. Existence and global regularity of solutions, *SIAM J. Math. Anal.* **22** (1992), 1388–1410.

[11] I. Daubechies and J. Lagarias, Two-scale difference equations II. Local regularity, infinite products of matrices and fractals, *SIAM J. Math. Anal.* **23** (1992), 1031–1079.

[12] T.B. Dinsenbacher and D.P. Hardin, Nonhomogeneous refinement equations, in *Wavelets, Multiwavelets, and their Applications*, A. Aldroubi and E. Lin (eds.), AMS Contemporary Mathematics Series **216**, 1997, pp. 117–128.

[13] T.B. Dinsenbacher and D.P. Hardin, Multivariate nonhomogeneous refinement equations, *J. Fourier Anal. Appl.* **5** (1999), 589–597.

[14] N. Dyn, J.A. Gregory, and D. Levin, A butterfly subdivision scheme for surface interpolation with tension control, *ACM Trans. Graphics* **2** (1990), 160–169.

[15] B. Han, T. Yu, and B. Piper, Multivariate refinable Hermite interpolants, *Math Comput.* **73** (2004), 1913–1935.

[16] C. Heil, G. Strang, and V. Strela, Approximation by translates of refinable functions, *Numer. Math.* **73** (1996), 75-94.

[17] L. Kobbelt, Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Computer Graphics Forum* **15** (1996), 409–420.

[18] R.Q. Jia and Q.T. Jiang, Approximation power of refinable vectors of functions, in: *Wavelet analysis and applications*, Studies Adv. Math., Vol. 25, Amer. Math. Soc., Providence, RI, 2002, pp. 155–178.

[19] R.Q. Jia, Q.T. Jiang, and Z.W. Shen, Distributional solutions of nonhomogeneous discrete and continuous refinement equations, *SIAM J. Math. Anal.* **32** (2000), 420–434.

[20] R.Q. Jia, Q.T. Jiang, and Z.W. Shen, Convergence of cascade algorithms associated with nonhomogeneous refinement equations, *Proc. Amer. Math. Soc.* **129** (2001), 415–427.

[21] R.Q. Jia, K.S. Lau, and D.X. Zhou, L_p solutions of refinement equations, *J. Fourier Anal. Appl.* **7** (2001), 143–167.

[22] R.Q. Jia, S.D. Riemenschneider, and D.X. Zhou, Smoothness of multiple refinable functions and multiple wavelets, *SIAM J. Matrix Anal. Appl.* **21** (1999), 1–28.

[23] Q.T. Jiang, Multivariate matrix refinable functions with arbitrary matrix dilation, *Trans. Amer. Math. Soc.* **351** (1999), 2407–2438.

[24] Q.T. Jiang and B.B. Li, Quad/triangle subdivision, nonhomogeneous refinement equation and polynomial reproduction, preprint, 2008.

[25] A. Levin, Polynomial generation and qusi-interpolation in stationary non-uniform subdivision, *Comput. Aided Geom. Design* **20** (2003), 41–54.

[26] A. Levin and D. Levin, Analysis of quasi-uniform subdivision, *Appl. Comput. Harmon. Anal.* **15** (2003), 18–32.

[27] S. Li, Convergence of cascade algorithms in Sobolev spaces associated with inhomogeneous refinement equations, *J. Approx. Theory* **104** (2000), 153–163.

[28] S. Li, Convergence rates of cascade algorithms associated with nonhomogeneous refinement equations, *J. Math. Anal. Appl.* **292** (2004), 84–95.

[29] C. Loop, Smooth subdivision surfaces based on triangles, University of Utah, Department of Mathematics, Masters Thesis, 1987.

[30] J. Maillot and J. Stam, A unified subdivision scheme for polygonal modeling, *Proceedings of Eurographics 2001, Computer Graphics Forum* **20** (2001), 471–479.

[31] J. Peters and L.-J. Shiue, Combining 4- and 3-direction subdivision, *ACM Trans. Graphics* **23**(2004), 980–1003.

[32] G. Plonka, Approximation order provided by refinable function vectors, *Constr. Approx.* **13** (1997), 221–244.

[33] G.C. Rota and G. Strang, A note on the joint spectral radius, *Indag. Math. (N.S.)* **22** (1960), 379–381.

[34] S. Schaefer and J. Warren, On C^2 triangle/quad subdivision, *ACM Trans. Graphics* **24** (2005), 28–36.

[35] J. Stam and C. Loop, Quad/triangle subdivision, *Computer Graphics Froum* **22** (2003), 1–7.

[36] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, U.S.A., 1996.

[37] G. Strang and D.X. Zhou, Inhomogeneous refinement equations, *J. Fourier Anal. Appl.* **4** (1998), 733–747.

[38] Q.Y. Sun, Nonhomogeneous refinement equation: Existence, regularity and biorthogonality, Unpublished manuscript, 1998.
available at `http://www.math.ucf.edu/~qsun`

[39] Q.Y. Sun, Compactly supported distributional solutions of nonstationary nonhomogeneous refinement equations, *Acta Math. Sin.* (Engl. Ser.) **17** (2001), 1–14.