Matrix-valued 4-point Spline and 3-point Non-spline Interpolatory Curve Subdivision Schemes

Charles K. Chui[1], Qingtang Jiang[2]

Department of Mathematics and Computer Science
University of Missouri–St. Louis
St. Louis, MO 63121, U.S.A.

Abstract

The objective of this paper is to study and construct matrix-valued templates for interpolatory curve subdivision. Since our investigation of this problem was motivated by the need of such subdivision stencils as boundary templates for interpolatory surface subdivision, we provide both spline and non-spline templates that are necessarily symmetric, due to the lack of direction-orientation in carrying out surface subdivision in general. For example, the minimum-supported Hermite interpolatory C^1 cubic spline curve subdivision scheme, with the skew-symmetric basis function for interpolating first derivatives, does not meet the symmetry specification. Non-spline C^2 interpolatory templates constructed in this paper are particularly important, due to their smaller support needed to minimize undesirable surface oscillations, when adopted as boundary templates for interpolatory C^2 surface subdivision. The curve subdivision templates introduced in this paper are adopted as boundary stencils for interpolatory surface subdivision with matrix-valued templates.

1. Introduction

It is well-known that subdivision schemes provide the most efficient way for generating curves and surfaces, see e. g. [15, 10]. To construct a smooth curve in the plane or in the 3-dimensional space, the subdivision process is carried out by taking weighted averages of the vertices iteratively, using the weights as specified by some “subdivision templates”. The initial vertices, called control points, are vertices of a given (control) polygon, and the number of vertices is increased after each iterative step, thereby generating a sequence of finer and finer polygons that eventually converge to a desirable limiting curve, called subdivision curve. For instance, when a “dyadic” subdivision scheme is employed, a finer polygon with “twice” as many vertices is generated from the previous polygon for each iterative step. This is accomplished by creating one new vertex (called “odd numbered” vertex) between every two consecutive old vertices (called “even numbered” vertices) of the coarser polygon (i.e. the polygon before the iterative step is carried out), while the positions of the old vertices may or may not be updated, again according to

[1] Research supported by ARO Grant #W911NF-07-1-0525. This author is also with the Department of Statistics, Stanford University, Stanford, CA 94305.

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

the subdivision templates. If the even numbered vertices are not to be updated, then the vertices of each coarser polygon are among the vertices of the finer polygon. This subdivision process is then called an *interpolatory subdivision scheme*. Otherwise, it is called an *approximation subdivision scheme*.

The subdivision templates of a "dyadic" curve subdivision scheme are constructed by following some local averaging rule, usually derived from certain refinement equation

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

where $\phi(x)$ is called a refinable function, with dilation factor 2 and corresponding (subdivision) refinement mask $\{p_k\}$, which is necessarily a finite sequence. For any control polygon with vertices v_k^0, an application of the refinement equation (1.1), immediately yields the local averaging rule:

$$v_j^{m+1} = \sum_k v_k^m p_{j-2k}, \quad m = 0, 1, \cdots, \tag{1.2}$$

where, for each $m = 1, 2, \cdots$, the point-set $\{v_k^m\}_k$ denotes the set of vertices obtained after taking m iterations according to this local averaging rule. To facilitate implementational convenience, the local averaging rule can also be described in terms of two subdivision templates, by considering the index j in (1.2) to be even and odd integers, respectively. For example, the local averaging rule for the subdivision mask $\{-\frac{1}{16}, 0, \frac{9}{16}, 1, \frac{9}{16}, 0, -\frac{1}{16}\}$, introduced independently in [4] and [5], can be re-formulated as:

$$\begin{aligned} v_{2i+1}^{m+1} &= \frac{1}{16}(-v_{i-1}^m + 9v_i^m + 9v_{i+1}^m - v_{i+2}^m), \\ v_{2i}^{m+1} &= v_i^m, \quad m \geq 0; \end{aligned}$$

and therefore may be described by the two subdivision templates shown in the middle and right of the three diagrams in Fig. 1. Here, solid circles represent the old vertices (i.e. before the iterative subdivision step is carried out) and the hollow circles represent the new vertices obtained after each iterative subdivision step. Observe that in this example the first subdivision template (shown in the middle) is a single solid circle. This means that the old vertices are not moved, and the subdivision scheme is interpolatory. On the other hand, the second template (shown on the right) consists of four solid circles and one hollow circle (for the new vertex). Hence, this subdivision scheme is called a 4-point interpolatory scheme.

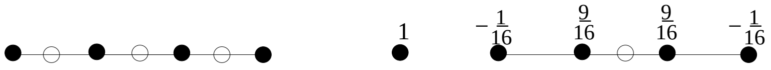

Figure 1: *Templates of 4-point interpolatory subdivision*

As illustrated by the above example, a (scalar) subdivision scheme is interpolatory, if and only if the subdivision mask $\{p_k\}$ satisfies the condition:

$$p_0 = 1, \; p_{2k} = 0, \quad k \neq 0,$$

which, under certain mild condition, is equivalent to the interpolatory property

$$\phi(k) = \delta_k, \quad k \in \mathbb{Z}$$

of the scaling function $\phi(x)$. It is well-known, as proved in [5], that the above 4-point interpolatory scheme is a C^1 scheme, meaning that the (limiting) subdivision curve is C^1 in terms of the parameter on the real line. On the other hand, to construct C^2 interpolatory schemes, it is necessary to admit larger templates (that is, more solid circles in the diagram on the right of Fig. 1), unless a larger integer dilation factor is considered. In this regard, the interested reader is referred to [13] for some 6-point C^2 interpolatory scheme, and [6] for some 4-point C^2 ternary interpolatory scheme. However, the subdivision curves so obtained are not spline curves; that is, the corresponding scaling functions are not (piecewise polynomial) spline functions.

More recently, in the construction of multi-wavelets, the refinement equation (1.1) is extended to vector refinement, which naturally leads to matrix-valued subdivision masks. For example, in [1], we studied matrix-valued interpolatory subdivision schemes for surface design and illustrated that matrix-valued C^2 interpolatory schemes with small subdivision templates can be constructed. In fact, the construction in [1] includes matrix-valued compactly supported spline scaling functions that assure the subdivision surfaces (at regular vertices) to be spline surfaces. The main objective of this paper is to apply the method introduced in [1] to construct matrix-valued 4-point C^2 spline interpolatory schemes and 3-point C^2 (non-spline) interpolatory schemes for curve design. In addition to being interpolatory, the curve subdivision templates constructed in this paper are symmetric, and therefore can be readily adopted as boundary subdivision templates for matrix-valued interpolatory surface subdivision.

This paper is organized as follows. In Section 2, we briefly recall the definition and some basic properties of matrix-valued interpolatory schemes introduced in [1]. Construction of some 4-point C^1 quadratic-spline and C^1 cubic-spline interpolatory schemes, both with symmetric subdivision templates, is demonstrated in Section 3. A 4-point C^2 cubic-spline interpolatory scheme, again with symmetric subdivision templates, is introduced in Section 4. In Section 5, a 3-point C^2 non-spline symmetric interpolatory scheme is presented. Finally, these curve subdivision schemes are adopted as boundary schemes for matrix-valued interpolatory surface subdivision in Section 6.

2. Matrix-valued interpolatory schemes for curve design

Matrix-valued subdivision schemes for curve generation are associated with some vector refinement equation

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

with an r-dimensional vector-valued refinable function $\Phi = [\phi_0, \cdots, \phi_{r-1}]^T$ (also called refinable function vector), and refinement (or two-scale) sequence of r-dimensional square matrices $\{P_k\}$, called a refinement (or subdivision) mask. The refinement equation (2.1) can also be written as $\widehat{\Phi}(\omega) = P(\frac{\omega}{2})\widehat{\Phi}(\frac{\omega}{2})$, where $P(\omega) = \frac{1}{2}\sum_k P_k e^{-ik\omega}$, is called the (two-scale) symbol of Φ. Here and throughout, $\widehat{\Phi} = [\widehat{\phi}_0, \cdots, \widehat{\phi}_{r-1}]^T$, where $\widehat{\phi}_\ell$ denotes the Fourier transform of ϕ_ℓ. In the following, Φ is always assumed to satisfy the property of "partition of unity", meaning that there exists a nonzero constant r-dimensional vector $\mathbf{y}_0$ such that

$$\mathbf{y}_0 \sum_{k\in\mathbf{Z}} \Phi(x-k) = const \neq 0, \quad x \in \mathbb{R}. \tag{2.2}$$

When the vector refinement equation (2.1) is applied to curve subdivision, the local averaging rule (1.2) is extended to the matrix setting:

$$\mathbf{v}_k^{m+1} = \sum_j \mathbf{v}_j^m P_{k-2j}, \quad m = 0, 1, \cdots, \tag{2.3}$$

where

$$\mathbf{v}_k^m =: [v_k^m, s_{k,1}^m \cdots, s_{k,r-1}^m] \tag{2.4}$$

are "row-vectors" with r components of 3×1 vectors $v_k^m, s_{k,i}^m$, $i = 1, \cdots, r-1$. For the vector $\mathbf{y}_0 = [1, 0, \cdots, 0]$, as chosen in our earlier work [1, 2, 3], we may use the first components v_k^m to denote the vertices of the 3-D subdivision polygons generated after the m-th iterative step, with initial vertices v_k^0 being the control points of the curve subdivision. The other components $s_{k,1}^0, \cdots, s_{k,r-1}^0$ of $\mathbf{v}_k^0$, can be used to control the geometric shape of the limiting curve. Then, it can be shown by applying the property of partition of unity, as in [2], that the vertices v_k^m provide an accurate discrete approximation of the target subdivision curve, formulated by the series representation:

$$F(x) = \sum_k v_k^0 \phi_0(x-k) + \sum_k \Big(s_{k,1}^0 \phi_1(x-k) + \cdots + s_{k,r-1}^0 \phi_{r-1}(x-k) \Big).$$

For the matrix-valued surface subdivision, the other components $s_{k,1}^0, \cdots, s_{k,r-1}^0$ of $\mathbf{v}_k^0$ are called the shape control pararemeters in [1, 2, 3]. It was demonstrated in [2] that the shape control parameters can change the shape of the limiting surface significantly.

The local averaging rule for a matrix-valued scheme can also be described in terms of subdivision templates. For example, the two templates for the local averaging rule of any subdivision mask $\{P_k\}_{k=-3,\cdots,3}$ are shown in Fig. 2.

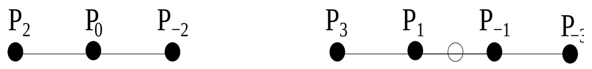

Figure 2: *Templates of matrix-valued 4-point subdivision scheme*

Following [1] in the study of surface subdivision schemes, we have the following definition for matrix-valued interpolatory curve subdivision schemes.

Definition 1 *A matrix-valued (dyadic) subdivision scheme with matrix-valued mask $\{P_k\}$ is said to be interpolatory, if for any given control vectors $\mathbf{v}_k^0$, the first components v_{2k}^m of $\mathbf{v}_{2k}^m$, $m = 1, 2, \cdots$, in (2.3) remain the same as the corresponding first components of coarser vectors v_k^{m-1}; i.e., $v_{2k}^m = v_k^{m-1}$, $m = 1, 2, \cdots$.*

The following result on characterization of interpolatory subdivision masks for the matrix-valued setting can be easily established.

Proposition 1 *A (dyadic) subdivision scheme with matrix-valued mask $\{P_k\}$ is interpolatory if and only if P_j satisfies*

$$P_0 = \begin{bmatrix} 1 & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \cdots & \vdots \\ 0 & * & \cdots & * \end{bmatrix}, \quad P_{2k} = \begin{bmatrix} 0 & * & \cdots & * \\ \vdots & \vdots & \cdots & \vdots \\ 0 & * & \cdots & * \end{bmatrix}, \quad k \neq 0, k \in \mathbb{Z}; \tag{2.5}$$

that is, the first column of P_0 must be the unit vector $[1, 0, \cdots, 0]^T$ and the first columns of the other P_{2k} (with even indices) must be the zero vector.

Let $\Phi = [\phi_0, \phi_1, \cdots, \phi_{r-1}]^T$ be a refinable function vector with subdivision mask $\{P_{\mathbf{k}}\}$. Then under certain mild conditions (see, for example, [1] and [14]), the subdivision scheme associated with $\{P_{\mathbf{k}}\}$ is interpolatory if and only if Φ satisfies:

$$\phi_0(k) = \delta(k), \quad \phi_1(k) = 0, \cdots, \phi_{r-1}(k) = 0, \quad k \in \mathbb{Z}. \tag{2.6}$$

3. Matrix-valued 4-point C^1 quadratic and cubic spline interpolatory scheme

This section is devoted to the study of cardinal spline interpolation by compactly supported, symmetric, and refinable piecewise-polynomial functions, where cardinal interpolation means interpolation of the data δ_k, $k \in \mathbb{Z}$. The statement of this problem itself demands an explanation, since the problem of spline interpolation is well, and perhaps, overly studied in the published literature in both mathematics and engineering (see particularly [9, 11], for quadratic spline interpolation). However, the specification of the problem, as governed by the adjectives, "compactly supported", "symmetric", and "refinable", does impose some challenge, and this particular specification must be met in constructing interpolatory curve subdivision templates, that can be adopted as boundary stencils for interpolatory surface subdivision, (the topic of discussion in Section 6). In particular, in application to connecting individually designed subdivision surface patches in the 3-space, symmetry of the (curve) subdivision templates would compensate for the lack of orientation of the surface subdivision schemes. For example, the minimum-supported Hermite interpolatory C^1 cubic spline curve subdivision scheme, with the skew-symmetric basis function for interpolating first derivatives, does not meet the symmetry specification; and the issue of refinability of basis functions were not considered in the quadratic spline interpolation papers [9, 11].

For spline-based subdivision, we only need to study quadratic and cubic spline interpolation. Let $B_3(x)$ be the normalized C^1 quadratic cardinal B-spline with support [0, 3]. In the following, we consider some spline function φ_0 with support [-1, 1] and symmetric with respect to $x = 0$. (In this regard, observe that the spline function $B_3(2x+1)$ with support $[-\frac{1}{2}, 1]$ lacks symmetry). Let

$$\varphi_0(x) := B_3(2x+2) + B_3(2x+1). \tag{3.1}$$

Note that φ_0 is not refinable, in the sense that $\varphi_0(\frac{x}{2})$ is not a (finite) linear combination of $\varphi_0(x-k), k \in \mathbb{Z}$. Yet we may still write

$$\hat{\varphi}_0(2\omega) = \frac{1}{8}(2 + e^{i\omega} + e^{-i\omega} + 2e^{i\frac{\omega}{2}} + 2e^{-i\frac{\omega}{2}})\hat{\varphi}_0(\omega), \tag{3.2}$$

which follows directly from the definition of φ_0 and the refinement of $B_3(x) : \hat{B}_3(2\omega) = \frac{1}{8}(1 + e^{-i\omega})^3\hat{B}_3(\omega)$. Next, we introduce $\varphi_1(x) = \frac{1}{3}\varphi_0(x) + \frac{2}{3}(\varphi_0(x+\frac{1}{2}) + \varphi_0(x-\frac{1}{2}))$, namely:

$$\varphi_1(x) := B_3(2x+2) + B_3(2x+1) + \frac{2}{3}(B_3(2x+3) + B_3(2x)). \tag{3.3}$$

The Bézier coefficients for φ_0 and φ_1 are shown in Figs. 3 and 4 respectively. In the following proposition, we will see that the vector-valued function $\Phi^a := [\varphi_0, \varphi_1]^T$ is refinable.

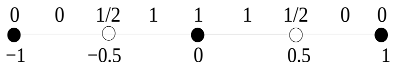

Figure 3: *Support and Bézier coefficients for φ_0*

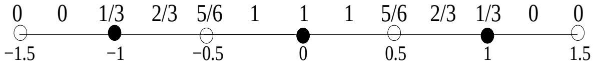

Figure 4: *Support and Bézier coefficients for φ_1*

Proposition 2 *Let φ_0 and φ_1 be the compactly supported C^1 quadratic spline functions defined by (3.1) and (3.3) respectively. Then $\Phi^a = [\varphi_0, \varphi_1]^T$ is refinable with refinement mask $\{P_k\}$ given by*

$$P_0 = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{5}{12} & \frac{1}{4} \end{bmatrix}, \ P_1 = P_{-1} = \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} \end{bmatrix}, \ P_2 = P_{-2} = \begin{bmatrix} 0 & 0 \\ \frac{1}{6} & 0 \end{bmatrix}, \ P_k = \mathbf{0}, |k| > 2. \tag{3.4}$$

We remark that the matrices P_k in (3.4) do not satisfy the conditions in Proposition 1, but they will be later modified to satisfy these conditions.

Proof of Proposition 2. From $\varphi_1(x) = \frac{1}{3}\varphi_0(x) + \frac{2}{3}(\varphi_0(x+\frac{1}{2}) + \varphi_0(x-\frac{1}{2}))$, or equivalently,

$$\hat{\varphi}_1(\omega) = \frac{1}{3}\hat{\varphi}_0(\omega) + \frac{2}{3}(e^{i\frac{\omega}{2}} + e^{-i\frac{\omega}{2}})\hat{\varphi}_0(\omega),$$

and in view of (3.2), we have

$$\begin{aligned}\hat{\varphi}_0(2\omega) &= \frac{1}{8}(2+e^{i\omega}+e^{-i\omega})\hat{\varphi}_0(\omega)+\frac{1}{4}\cdot\frac{3}{2}(\hat{\varphi}_1(\omega)-\frac{1}{3}\hat{\varphi}_0(\omega))\\ &= \frac{1}{8}(1+e^{i\omega}+e^{-i\omega})\hat{\varphi}_0(\omega)+\frac{3}{8}\hat{\varphi}_1(\omega). \qquad (3.5)\end{aligned}$$

On the other hand, since

$$\begin{aligned}&\hat{\varphi}_1(2\omega) = \frac{1}{3}\hat{\varphi}_0(2\omega)+\frac{2}{3}(e^{i\omega}+e^{-i\omega})\hat{\varphi}_0(2\omega)\\ &= \frac{1}{3}(1+2(e^{i\omega}+e^{-i\omega}))\{\frac{1}{8}(1+e^{i\omega}+e^{-i\omega})\hat{\varphi}_0(\omega)+\frac{3}{8}\hat{\varphi}_1(\omega)\}\\ &= \frac{1}{24}(5+3e^{i\omega}+3e^{-i\omega}+2e^{i2\omega}+2e^{-i2\omega})\hat{\varphi}_0(\omega)+\frac{1}{8}(1+2e^{i\omega}+2e^{-i\omega})\hat{\varphi}_1(\omega), (3.6)\end{aligned}$$

It follows, by combining (3.5) and (3.6), that Φ^a is refinable with refinement mask given by (3.4). ◇

The refinement mask of Φ^a yields the subdivision templates for the local averaging rule, as shown in Fig. 5 with $G = P_0, H = P_1, L = P_2$.

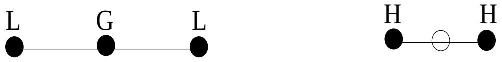

Figure 5: *Local averaging rule*

Since P_0, P_2 do not satisfy (2.5), this scheme is not interpolatory. However, since φ_0 is supported on [-1, 1], we can modify φ_1 by subtracting from $\varphi_1(x)$ a suitable finite linear combination of $\varphi_0(x-k)$. Observe that $\varphi_1(0) = 1$, $\varphi_1(1) = \varphi_1(-1) = \frac{1}{3}$. So, if we let $\widetilde{\varphi}_1(x) := \varphi_1(x) - \varphi_0(x) - \frac{1}{3}(\varphi_0(x-1)+\varphi_0(x+1))$, then $\widetilde{\varphi}_1$ satisfies (2.6). Therefore, $[\varphi_0, \widetilde{\varphi}_1(x)]^T$ will yield a matrix-valued interpolatory scheme, provided that it is refinable. However, although it is indeed refinable, we will consider instead, $\widetilde{\Phi}^a := [\varphi_0+\frac{3}{8}\widetilde{\varphi}_1, \frac{3}{8}\widetilde{\varphi}_1]^T$, in the following, since $\widetilde{\Phi}^a$ satisfies the condition of generalized partition of unity (2.2) with $\mathbf{y}_0 = [1, 0]$.

Proposition 3 *Let φ_0 and φ_1 be the compactly supported C^1 quadratic spline functions defined by (3.1) and (3.3) resp. Set*

$$\begin{aligned}\widetilde{\varphi}_0(x) &= \frac{5}{8}\varphi_0(x)+\frac{3}{8}\varphi_1(x)-\frac{1}{8}(\varphi_0(x-1)+\varphi_0(x+1));\\ \widetilde{\varphi}_1(x) &= \frac{3}{8}(\varphi_1(x)-\varphi_0(x))-\frac{1}{8}(\varphi_0(x-1)+\varphi_0(x+1)).\end{aligned}$$

Then $\widetilde{\Phi}^a = [\widetilde{\varphi}_0, \widetilde{\varphi}_1]^T$ is refinable with refinement mask $\{\widetilde{P}_k\}$ given by

$$\widetilde{P}_0 = \frac{1}{16}\begin{bmatrix}16 & 8\\ 0 & -8\end{bmatrix},\ \widetilde{P}_1 = \widetilde{P}_{-1} = \frac{1}{16}\begin{bmatrix}9 & -1\\ 1 & 7\end{bmatrix}, \qquad (3.7)$$

$$\widetilde{P}_2 = \widetilde{P}_{-2} = \frac{1}{16}\begin{bmatrix}0 & -4\\ 0 & -4\end{bmatrix},\ \widetilde{P}_3 = \widetilde{P}_{-3} = \frac{1}{16}\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix},\ \widetilde{P}_k = \mathbf{0}, |k| > 3.$$

Proof. With $z = e^{-i\omega}$, let

$$U(z) = \begin{bmatrix} \frac{5}{8} - \frac{1}{8}(z + z^{-1}) & \frac{3}{8} \\ -\frac{3}{8} - \frac{1}{8}(z + z^{-1}) & \frac{3}{8} \end{bmatrix}.$$

Then $\widehat{\widetilde{\Phi}^a}(\omega) = U(z)\widehat{\Phi}^a(\omega)$. Thus, the two-scale symbol $\widetilde{P}(\omega)$=$U(z^2)P(\omega)U(z)$ of $\widetilde{\Phi}^a$ is

$$\frac{z^{-3}}{32}\begin{bmatrix} 16z^3 + 9(z^4 + z^2) - (z^6 + 1) & 8z^3 - (z^4 + z^2) - 4(z^5 + z) + (z^6 + 1) \\ (z^4 + z^2) - (z^6 + 1) & -8z^3 + 7(z^4 + z^2) - 4(z^5 + z) + (z^6 + 1) \end{bmatrix}.$$

Therefore, $\widetilde{\Phi}^a$ is refinable with refinable mask given by (3.7). ♢

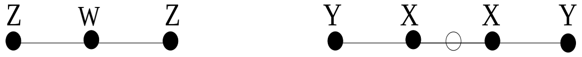

Figure 6: *Matrix-valued 4-point interpolatory scheme*

Clearly, the mask of $\widetilde{\Phi}^a$ immediately yields an interpolatory scheme with templates shown in Fig. 6 with $W = \widetilde{P}_0, X = \widetilde{P}_1, Y = \widetilde{P}_3, Z = \widetilde{P}_2$. We call such a scheme a *matrix-valued 4-point interpolatory scheme.*

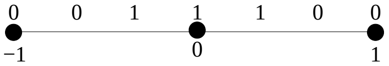

Figure 7: *Support and Bézier coefficients for* f_0

We can also construct the matrix-valued C^1 cubic 4-point interpolatory scheme by the same method. To be more precise, let $f_0(x)$ be the C^1 cubic spline supported on [-1, 1] with its Bézier coefficients shown in Fig. 7. Note that f_0 is not refinable. So, let $f_1(x) = f_0(\frac{x}{2})$, and by applying the C^2-smoothing formula, we have the Bézier representations $f_1(x)$ and $f_1(\frac{x}{2})$, as shown in Fig. 8 and Fig. 9, respectively. By evaluating the Bézier coefficients of $f_1(\frac{x}{2})$ and $f_0(x-k), f_1(x-k)$, we can show that $\Phi^b(x) := [f_0(x), f_1(x)]^T$ is refinable and its refinement mask can be easily computed. Furthermore, we can show that $\widetilde{\Phi}^b(x) := [\widetilde{f}_0(x), \widetilde{f}_1(x)]^T$ yields a matrix-valued 4-point interpolatory scheme, where

$$\begin{cases} \widetilde{f}_0(x) = \frac{1}{3}f_0(x) + \frac{2}{3}f_1(x) - \frac{1}{3}(f_0(x-1) + f_0(x+1)), \\ \widetilde{f}_1(x) = \frac{2}{3}(f_1(x) - f_0(x)) - \frac{1}{3}(f_0(x-1) + f_0(x+1)). \end{cases} \tag{3.8}$$

More precisely, we have the following result.

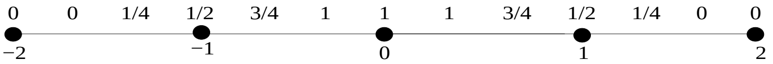

Figure 8: *Support and Bézier coefficients for* $f_1(x)(= f_0(\frac{x}{2}))$

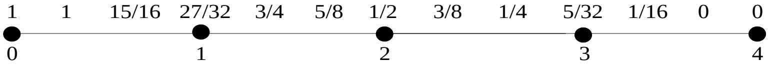

Figure 9: *Bézier coefficients for* $f_1(\frac{x}{2})(= f_0(\frac{x}{4}))$ *on [0, 4]*

Proposition 4 *Suppose* f_0 *is the compactly supported* C^1 *cubic spline functions with Bézier coefficients shown in Fig. 7. Let* $f_1(x) = f_0(\frac{x}{2})$*, and* $\widetilde{f}_0, \widetilde{f}_1$ *be the functions defined by (3.8). Then both* $\Phi^b = [f_0, f_1]^T$ *and* $\widetilde{\Phi}^b = [\widetilde{f}_0, \widetilde{f}_1]^T$ *are refinable, with refinement masks,* $(G_k)_k$ *and* $(\widetilde{G}_k)$*, given respectively by*

$$G_0 = \begin{bmatrix} 0 & 1 \\ -\frac{1}{4} & \frac{3}{4} \end{bmatrix}, \; G_1 = G_{-1} = \begin{bmatrix} 0 & 0 \\ -\frac{7}{32} & \frac{1}{2} \end{bmatrix},$$

$$G_2 = G_{-2} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{8} & \frac{3}{8} \end{bmatrix}, \; G_3 = G_{-3} = \begin{bmatrix} 0 & 0 \\ -\frac{1}{32} & 0 \end{bmatrix}, \; G_k = \mathbf{0}, \; |k| > 3,$$

and

$$\widetilde{G}_0 = \frac{1}{16}\begin{bmatrix} 16 & 4 \\ 0 & -4 \end{bmatrix}, \; \widetilde{G}_1 = \widetilde{G}_{-1} = \frac{1}{16}\begin{bmatrix} 9 & -1 \\ 1 & 7 \end{bmatrix},$$

$$\widetilde{G}_2 = \widetilde{G}_{-2} = \frac{1}{16}\begin{bmatrix} 0 & -2 \\ 0 & -2 \end{bmatrix}, \; \widetilde{G}_3 = \widetilde{G}_{-3} = \frac{1}{16}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \; \widetilde{G}_k = \mathbf{0}, \; |k| > 3.$$

Again, the mask of $\widetilde{\Phi}^b$ yields a matrix-valued 4-point interpolatory scheme with templates shown in Fig. 6, where $W = \widetilde{G}_0, X = \widetilde{G}_1, Y = \widetilde{G}_3, Z = \widetilde{G}_2$.

4. Matrix-valued 4-point C^2 cubic spline interpolatory scheme

In this section we consider C^2 cubic spline functions and show that the above approach can be applied to construct a matrix-valued 4-point C^2 cubic-spline interpolatory scheme. Let $B_4(x)$ be the normalized cardinal C^2 cubic B-spline with support [0, 4], and $\phi_0(x) = \frac{3}{2}B_4(2x+2)$. Then ϕ_0 is supported in [-1, 1] and symmetric with respect to $x = 0$. However, ϕ_0 is not refinable. More precisely, from the definition of ϕ_0 and the refinement of $B_4(x) : \hat{B}_4(2\omega) = \frac{1}{16}(1+e^{-i\omega})^4\hat{B}_4(\omega)$, we have

$$\hat{\phi}_0(2\omega) = \frac{1}{16}(6 + e^{i\omega} + e^{-i\omega})\hat{\phi}_0(\omega) + \frac{1}{4}(e^{i\frac{\omega}{2}} + e^{-i\frac{\omega}{2}})\hat{\phi}_0(\omega). \tag{4.1}$$

Next, we define $\phi_1(x) = \frac{7}{9}\phi_0(x) + \frac{4}{9}(\phi_0(x+\frac{1}{2}) + \phi_0(x-\frac{1}{2}))$, namely:

$$\phi_1(x) := \frac{7}{6}B_4(2x+2) + \frac{2}{3}(B_4(2x+3) + B_4(2x+1)). \tag{4.2}$$

The Bézier coefficients for ϕ_0 and ϕ_1 are shown in Figs. 10 and 11 resp., and the vector-valued function $\Phi^c := [\phi_0, \phi_1]^T$ is refinable. More precisely, we have the following result.

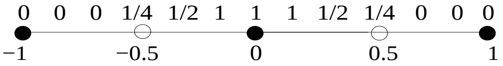

Figure 10: *Support and Bézier coefficients for* ϕ_0

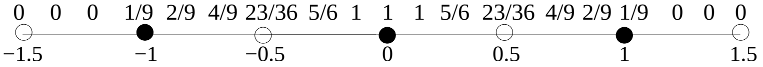

Figure 11: *Support and Bézier coefficients for* ϕ_1

Proposition 5 *Let* $\phi_0 = \frac{3}{2}B_4(2x+2)$ *and* ϕ_1 *be defined by (4.2). Then* $\Phi^c = [\phi_0, \phi_1]^T$ *is refinable and its associated refinement mask* $\{H_k\}$ *is given by*

$$H_0 = \begin{bmatrix} -\frac{1}{8} & \frac{9}{8} \\ \frac{1}{72} & \frac{7}{8} \end{bmatrix}, H_1 = H_{-1} = \begin{bmatrix} \frac{1}{8} & 0 \\ \frac{1}{24} & \frac{1}{2} \end{bmatrix}, H_2 = H_{-2} = \begin{bmatrix} 0 & 0 \\ \frac{1}{18} & 0 \end{bmatrix}, H_k = \mathbf{0}, |k| > 2. \quad (4.3)$$

Proof. From $\phi_1(x) = \frac{7}{9}\phi_0(x) + \frac{4}{9}(\phi_0(x+\frac{1}{2}) + \phi_0(x-\frac{1}{2}))$, we have

$$\hat{\phi}_1(\omega) = \frac{7}{9}\hat{\phi}_0(\omega) + \frac{4}{9}(e^{i\frac{\omega}{2}} + e^{-i\frac{\omega}{2}})\hat{\phi}_0(\omega).$$

This, together with (4.1), leads to

$$\begin{aligned} \hat{\phi}_0(2\omega) &= \frac{1}{16}(6 + e^{i\omega} + e^{-i\omega})\hat{\phi}_0(\omega) + \frac{1}{4}(\frac{9}{4}\hat{\phi}_1(\omega) - \frac{7}{4}\hat{\phi}_0(\omega)) \\ &= \frac{1}{16}(-1 + e^{i\omega} + e^{-i\omega})\hat{\phi}_0(\omega) + \frac{9}{16}\hat{\phi}_1(\omega). \end{aligned} \quad (4.4)$$

On the other hand, we have

$$\begin{aligned} \hat{\phi}_1(2\omega) &= \frac{1}{9}(7 + 4e^{i\omega} + 4e^{-i\omega})\hat{\phi}_0(2\omega) \\ &= \frac{1}{9}(7 + 4e^{i\omega} + 4e^{-i\omega}))\{\frac{1}{16}(-1 + e^{i\omega} + e^{-i\omega})\hat{\phi}_0(\omega) + \frac{9}{16}\hat{\phi}_1(\omega)\} \\ &= \frac{1}{144}(1 + 3e^{i\omega} + 3e^{-i\omega} + 4e^{i2\omega} + 4e^{-i2\omega})\hat{\phi}_0(\omega) + \frac{1}{16}(7 + 4e^{i\omega} + 4e^{-i\omega})\hat{\phi}_1(\omega)(4.5) \end{aligned}$$

From (4.4) and (4.5), we see that Φ^c is refinable and its associated refinement mask is given by (4.3). ♢

The refinement mask of Φ^c yields the templates for the local averaging rule as shown in Fig. 5 with $G = H_0, H = H_1, L = H_2$. In this case the vector $\mathbf{y}_0$ for the partition unity property of Φ^c is $[1, 9]$. In practice, if such a matrix-valued C^2 cubic approximation subdivision scheme is used for curve design, we would recommend the choice of $\Phi^d := [\phi_0 + 9\phi_1,\ -9\phi_0 + 9\phi_1]^T$, since the vector $\mathbf{y}_0$ for the generalized partition of unity property of Φ^d is $[1, 0]$, and Φ^d is refinable with refinement mask, denoted as $(H_k^{(d)})_k$, given by

$$H_0^{(d)} = \begin{bmatrix} \frac{9}{10} & \frac{1}{10} \\ -\frac{1}{10} & -\frac{3}{20} \end{bmatrix}, H_1^{(d)} = H_{-1}^{(d)} = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{3}{8} & \frac{1}{8} \end{bmatrix},$$

$$H_2^{(d)} = H_{-2}^{(d)} = \begin{bmatrix} \frac{1}{20} & -\frac{1}{20} \\ \frac{1}{20} & -\frac{1}{20} \end{bmatrix}, H_k^{(d)} = \mathbf{0}, |k| > 2.$$

The scheme derived from Φ^c is not interpolatory, since H_0, H_2, H_{-2} do not satisfy (2.5). However, we can modify ϕ_1 as in the above section such that the resulting refinable function vector will yields an interpolatory scheme. Observe that $\phi_1(0) = 1, \phi_1(1) = \phi_1(-1) = \frac{1}{9}$. Then

$$\widetilde{\phi}_1(x) := \frac{3}{4}(\phi_1(x) - \phi_0(x)) - \frac{1}{12}(\phi_0(x-1) + \phi_0(x+1)) \tag{4.6}$$

satisfies (2.6). We can show that $[\phi_0, \widetilde{\phi}_1]^T$ is refinable, and therefore, its associated mask yields a matrix-valued interpolatory scheme. However, the vector $\mathbf{y}_0$ for the property of generalized partition of unity (2.2) is [1, 1]. Hence, in the following we consider, instead, $\widetilde{\Phi}^c := [\widetilde{\phi}_0, \widetilde{\phi}_1]^T$ so that the vector $\mathbf{y}_0$ for generalized partition of unity (2.2) of $\widetilde{\Phi}^c$ is $[1, 0]$, where $\widetilde{\phi}_0 = \phi_0 + \widetilde{\phi}_1$, namely,

$$\widetilde{\phi}_0(x) := \frac{1}{4}\phi_0(x) + \frac{3}{4}\phi_1(x) - \frac{1}{12}(\phi_0(x-1) + \phi_0(x+1)). \tag{4.7}$$

Proposition 6 *Suppose $\phi_0 = \frac{3}{2}B_4(2x+2)$ and ϕ_1 is defined by (4.2). Let $\widetilde{\phi}_0$ and $\widetilde{\phi}_1$ be the C^2 cubic spline functions defined by (4.7) and (4.6). Then $\widetilde{\Phi}^c = [\widetilde{\phi}_0, \widetilde{\phi}_1]^T$ is refinable and the refinement mask is given by*

$$\begin{aligned} &\widetilde{H}_0 = \frac{1}{48}\begin{bmatrix} 48 & 12 \\ 0 & -12 \end{bmatrix}, \ \widetilde{H}_1 = \widetilde{H}_{-1} = \frac{1}{48}\begin{bmatrix} 25 & -1 \\ 13 & 11 \end{bmatrix}, \\ &\widetilde{H}_2 = \widetilde{H}_{-2} = \frac{1}{48}\begin{bmatrix} 0 & -6 \\ 0 & -6 \end{bmatrix}, \ \widetilde{H}_3 = \widetilde{H}_{-3} = \frac{1}{48}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \ \widetilde{H}_k = \mathbf{0}, |k| > 3. \end{aligned} \tag{4.8}$$

Proof. Let

$$U_1(z) = \begin{bmatrix} \frac{1}{4} - \frac{1}{12}(z + z^{-1}) & \frac{3}{4} \\ -\frac{3}{4} - \frac{1}{12}(z + z^{-1}) & \frac{3}{4} \end{bmatrix},$$

where $z = e^{-i\omega}$. Then $\widehat{\widetilde{\Phi}^c}(\omega) = U_1(z)\widehat{\Phi}^c(\omega)$. Thus, the two-scale symbol $\widetilde{H}(\omega)$=$U_1(z^2)H(\omega)U_1(z)^{-1}$ of $\widetilde{\Phi}^c$ is

$$\frac{z^{-3}}{96}\begin{bmatrix} 48z^3 + 25(z^4 + z^2) - (z^6 + 1) & 12z^3 - (z^4 + z^2) - 6(z^5 + z) + (z^6 + 1) \\ 13(z^4 + z^2) - (z^6 + 1) & -12z^3 + 11(z^4 + z^2) - 6(z^5 + z) + (z^6 + 1) \end{bmatrix}.$$

Therefore $\widetilde{\Phi}^c$ is refinable with refinement mask given by (4.8), as desired; and $\widetilde{\Phi}^c$ yields a matrix-valued 4-point interpolatory scheme with templates shown in Fig. 6, where $W = \widetilde{H}_0, X = \widetilde{H}_1, Y = \widetilde{H}_3, Z = \widetilde{H}_2$. $\diamondsuit$

5. Matrix-valued 3-point C^2 non-spline interpolatory scheme

In the previous two sections, we have constructed matrix-valued spline interpolatory schemes; namely, schemes with spline functions as refinable functions. In this section, we will construct 3-point non-spline C^2 interpolatory schemes with templates shown in Fig.

12. More precisely, we consider the refinable function vector $\Phi^e(x) := [\phi_0^e(x), \phi_1^e(x)]^T$ that satisfies the refinement relation:

$$\Phi^e(x) = \sum_{k=-2}^{2} Q_k \Phi^e(2x-k),$$

with

$$Q_0 = \begin{bmatrix} 1 & -2t_1 \\ 0 & \frac{1}{4} - 2(t_1+t_2) \end{bmatrix},\ Q_1 = Q_{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{1}{8} & \frac{1}{8} \end{bmatrix},\ Q_2 = Q_{-2} = \begin{bmatrix} 0 & t_1 \\ 0 & t_2 \end{bmatrix},\ t_1, t_2 \in \mathbb{R}.$$

The mask $\{Q_k\}_{k=-2,-1,0,1,2}$ satisfies the so-called "sum rule of order 4" (refer to e. g., [7]), which implies that the integer shifts $\phi_0^e(x-k), \phi_1^e(x-k), k \in \mathbb{Z}$ reproduce polynomials of degrees up to 3. We can determine t_1, t_2 by the Sobolev smoothness estimate formula in [8] for the refinable function vectors so that the resulting ϕ_0^e, ϕ_1^e are C^2. There are many choices of t_1, t_2 for C^2 ϕ_0^e, ϕ_1^e. For example, if we choose

$$t_1 = \frac{21}{64},\ t_2 = -\frac{7}{64},$$

the resulting ϕ_0^e, ϕ_1^e are in the Sobolev space $W^{3.428}(\mathbb{R})$. For such t_1, t_2, the corresponding $Q_k, -2 \le k \le 2$, are $Q_0 = W_1, Q_1 = Q_{-1} = X_1, Q_2 = Q_{-2} = Z_1$, where

$$W_1 = \begin{bmatrix} 1 & -\frac{21}{32} \\ 0 & -\frac{3}{16} \end{bmatrix},\ X_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{1}{8} & \frac{1}{8} \end{bmatrix},\ Z_1 = \begin{bmatrix} 0 & \frac{21}{64} \\ 0 & -\frac{7}{64} \end{bmatrix}. \tag{5.1}$$

Thus, we have a matrix-valued 3-point C^2 interpolatory scheme shown in Fig. 12.

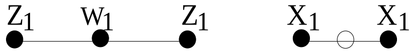

Figure 12: *Matrix-valued 3-point interpolatory scheme*

6. Application to surface design

In this section, we will apply the matrix-valued C^2 interpolatory schemes for curve subdivision constructed in Sections 3-5 as boundary schemes for interpolatory surface design. However, for simplicity, we will only consider the triangular mesh and the interior boundary vertices which are not corner vertices. Furthermore, we will only illustrate our approach by considering the non-spline 1-ring C^2 interpolatory surface subdivision schemes constructed in [1] and [3] for the interior vertices. The subdivision templates for regular vertices are shown in Fig.13, where the matrix-valued weights are given by

$$A = \begin{bmatrix} 1 & -\frac{435}{256} \\ 0 & -\frac{91}{256} \end{bmatrix},\ D = \begin{bmatrix} 0 & \frac{145}{512} \\ 0 & -\frac{45}{512} \end{bmatrix},\ B = \begin{bmatrix} \frac{3}{8} & 0 \\ -\frac{47}{512} & \frac{69}{512} \end{bmatrix},\ C = \begin{bmatrix} \frac{1}{8} & 0 \\ -\frac{17}{512} & -\frac{5}{512} \end{bmatrix}. \tag{6.1}$$

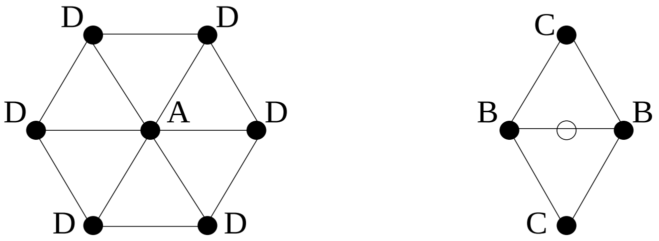

Figure 13: *Templates of 1-ring C^2 interpolatory scheme for regular old ("even") vertices (left) and new ("odd") vertices (right)*

An (interior) boundary vertex on a triangular mesh is called a regular boundary vertex if its valence (number of its neighbors) is 4; otherwise, it is called an extraordinary boundary vertex. We will use the templates of the averaging rule in Fig. 6 for both regular and extraordinary boundary vertices. Using the same templates for both regular and extraordinary boundary vertices has the advantage in connecting two individually designed subdivision surface patches, as pointed out in [15]. Here, we show that the eigenvalues $\lambda_0, \lambda_1, \cdots$ (in the order of non-increasing magnitudes) of the subdivision matrices near boundary extraordinary vertices satisfy certain basic properties. In this regard, we need the following lemma.

Lemma 1 *Let $M_n(s,t)$ be a 3-banded $n \times n$ symmetric matrix defined by*

$$M_n(s,t) = \begin{bmatrix} s & t & & & \\ t & s & t & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & t \\ & & & t & s \end{bmatrix}_{n\times n}, \tag{6.2}$$

where $s, t \in \mathbb{R}$. Then the eigenvalues of $M_n(s,t)$ are given by $s - 2t\cos\frac{j\pi}{n+1}$, $1 \le j \le n$.

The eigenvalues of a matrix of the form (6.2) have been discussed in [12]. For completeness, we will provide the proof of Lemma 1 in the Appendix.

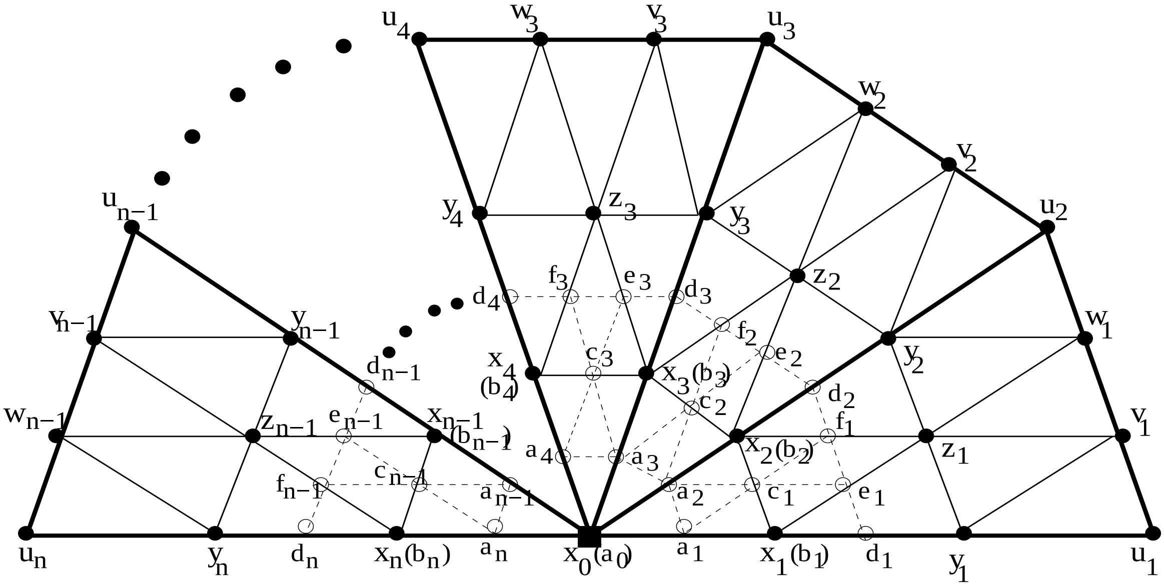

Figure 14: *2-D representation of initial vectors and refined vectors after one step of matrix-valued subdivision iteration near the extraordinary boundary vertex* $\mathbf{x}_0$

Let $\mathbf{x}_0$ be an initial vector associated with an extraordinary boundary vertex with valence n, and let $\mathbf{x}_j, \mathbf{y}_j, \mathbf{u}_j, \mathbf{z}_k, \mathbf{v}_k, \mathbf{w}_k$, $1 \le j \le n, 1 \le k \le n-1$, be the initial vectors associated with vertices in the neighborhood of this extraordinary vertex. Also, let $\mathbf{a}_0, \mathbf{a}_j, \mathbf{b}_j, \mathbf{d}_j, \mathbf{c}_k, \mathbf{e}_k, \mathbf{f}_k$, $1 \le j \le n, 1 \le k \le n-1$, be the corresponding vectors after carrying out one iteration of the subdivision process with the averaging rules given in Figs. 6 and 13. See Fig. 14 for the 2-D representation of these initial vectors and refined vectors.

Consider the subdivision matrix S, defined by

$$\begin{aligned} & \Big[[\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_n, \mathbf{b}_1, \mathbf{b}_n, \mathbf{d}_1, \mathbf{d}_n], [\mathbf{a}_2, \mathbf{a}_3, \cdots, \mathbf{a}_{n-1}], [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{n-1}], [\mathbf{b}_2, \mathbf{b}_3, \cdots, \mathbf{b}_{n-1}], \\ & \qquad [\mathbf{d}_2, \mathbf{d}_3, \cdots, \mathbf{d}_{n-1}], [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{n-1}], [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_{n-1}]\Big] \\ = \; & \Big[[\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_n, \mathbf{y}_1, \mathbf{y}_n, \mathbf{u}_1, \mathbf{u}_n], [\mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_{n-1}], [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{n-1}], [\mathbf{y}_2, \mathbf{y}_3, \cdots, \mathbf{y}_{n-1}], \\ & \qquad [\mathbf{u}_2, \mathbf{u}_3, \cdots, \mathbf{u}_{n-1}], [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n-1}], [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{n-1}]\Big] S. \end{aligned}$$

Then S is an upper-triangular block matrix with nonzero diagonal blocks $S_{j,j}, 0 \le j \le 3$

given by

$$S_{0,0} = \begin{bmatrix} W & X & X & Z & Z & Y & Y \\ Z & X & Y & W & \mathbf{0} & X & \mathbf{0} \\ Z & Y & X & \mathbf{0} & W & \mathbf{0} & X \\ \mathbf{0} & Y & \mathbf{0} & Z & \mathbf{0} & X & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Y & \mathbf{0} & Z & \mathbf{0} & X \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & Y \end{bmatrix}, \; S_{1,1} = \begin{bmatrix} B & C & & & \\ C & B & C & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & C \\ & & & C & B \end{bmatrix}_{(n-2)\times(n-2)},$$

$$S_{2,2} = \begin{bmatrix} C & & & \\ & C & & \\ & & \ddots & \\ & & & C \end{bmatrix}_{(n-1)\times(n-1)}, \; S_{3,3} = \begin{bmatrix} D & & & \\ & D & & \\ & & \ddots & \\ & & & D \end{bmatrix}_{(n-2)\times(n-2)}.$$

Here, $S_{2,2}$ has (nonzero) eigenvalues $\frac{1}{8}$ and $-\frac{5}{512}$, each with multiplicity $n-1$, and $S_{3,3}$ has (nonzero) eigenvalue $-\frac{45}{512}$ with multiplicity $n-2$. For $S_{1,1}$, let us consider $512S_{1,1}$, and introduce the notation

$$\widetilde{B} = 512B = \begin{bmatrix} 192 & 0 \\ -47 & 69 \end{bmatrix}, \; \widetilde{C} = 512C = \begin{bmatrix} 64 & 0 \\ -17 & -5 \end{bmatrix}.$$

Then by interchanging the second and third rows, followed by interchanging the second and third columns of the matrix $512S_{1,1}$, we have the matrix

$$\widetilde{S}_{1,1} = \left[\begin{array}{cccccccccccc} 192 & 64 & 0 & 0 & 0 & 0 & \multicolumn{2}{c}{\mathbf{0}} & \mathbf{0} & \cdots & \mathbf{0} \\ 64 & 192 & 0 & 0 & 64 & 0 & & & & & \\ -47 & -17 & 69 & -5 & 0 & 0 & \multicolumn{2}{c}{\mathbf{0}} & \mathbf{0} & \cdots & \mathbf{0} \\ -17 & -47 & -5 & 69 & -17 & -5 & & & & & \\ 0 & 64 & 0 & 0 & 192 & 0 & 64 & 0 & \mathbf{0} & \cdots & \mathbf{0} \\ 0 & -17 & 0 & -5 & -47 & 69 & -17 & -5 & & & \\ \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\widetilde{B}} & \multicolumn{2}{c}{\widetilde{C}} & \mathbf{0} & \cdots & \mathbf{0} \\ \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\widetilde{C}} & \multicolumn{2}{c}{\widetilde{B}} & \widetilde{C} & \cdots & \mathbf{0} \\ & & & & & & \multicolumn{2}{c}{\ddots} & \ddots & \ddots & \\ & & & & & & & & \ddots & \ddots & \widetilde{C} \\ & & & & & & & & & \widetilde{C} & \widetilde{B} \end{array}\right].$$

Next, by moving the 5th row to become the third row, followed by moving the 5th column to become the third column, of the matrix $\widetilde{S}_{1,1}$, we have the matrix

$$\widetilde{\widetilde{S}}_{1,1} = \left[\begin{array}{cccccccccccc} 192 & 64 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ 64 & 192 & 64 & 0 & 0 & 0 & 0 & 0 & & & & \\ 0 & 64 & 192 & 0 & 0 & 0 & 64 & 0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -47 & -17 & 0 & 69 & -5 & 0 & 0 & 0 & & & & \\ -17 & -47 & -17 & -5 & 69 & -5 & 0 & 0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ 0 & -17 & -47 & 0 & -5 & 69 & -17 & -5 & & & & \\ 0 & 0 & 64 & 0 & 0 & 0 & 192 & 0 & \widetilde{C} & \mathbf{0} & \cdots & \mathbf{0} \\ 0 & 0 & -17 & 0 & 0 & -5 & -47 & 69 & & & & \\ \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\mathbf{0}} & \multicolumn{2}{c}{\widetilde{C}} & \widetilde{B} & \widetilde{C} & \cdots & \mathbf{0} \\ & & & & & & & & \ddots & \ddots & \ddots & \\ & & & & & & & & & \ddots & \ddots & \widetilde{C} \\ & & & & & & & & & & \widetilde{C} & \widetilde{B} \end{array}\right].$$

In general, this procedure is repeated by moving the $(2j+1)$th row to become the $(j+1)$th row, followed by moving the $(2j+1)$th column to become the $(j+1)$th column, for $1 \le j \le n-3$, to arrive at the block matrix:

$$\begin{bmatrix} M_{n-2}(192, 64) & \mathbf{0} \\ M_{n-2}(-47, -17) & M_{n-2}(69, -5) \end{bmatrix},$$

where $M_{n-2}(s,t)$ is the matrix defined by (6.2). Thus, $S_{1,1}$ is similar to

$$\begin{bmatrix} M_{n-2}(\frac{3}{8}, \frac{1}{8}) & \mathbf{0} \\ M_{n-2}(-\frac{47}{512}, -\frac{17}{512}) & M_{n-2}(\frac{69}{512}, -\frac{5}{512}) \end{bmatrix}.$$

Therefore, it follows from Lemma 1, that $S_{1,1}$ has eigenvalues

$$\frac{3}{8} - \frac{1}{4}\cos\frac{j\pi}{n-1}, \quad \frac{69}{512} + \frac{5}{256}\cos\frac{j\pi}{n-1}, \quad 1 \le j \le n-2, \tag{6.3}$$

among which $\frac{3}{8} - \frac{1}{4}\cos\frac{(n-2)\pi}{n-1} = \frac{3}{8} + \frac{1}{4}\cos\frac{\pi}{n-1}$ is the largest in magnitude.

If we apply the matrix-valued 4-point C^2 spline interpolatory scheme for the regular boundary vertices, namely, $X = \widetilde{H}_1, Y = \widetilde{H}_3, W = \widetilde{H}_0, Z = \widetilde{H}_2$ with $\widetilde{H}_k, k = 0, \cdots, 3$, given in (4.8), then the nonzero eigenvalues of $S_{0,0}$ are

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}.$$

Hence, the eigenvalues λ_k of S satisfy $\lambda_0 = 1 > |\lambda_1| \ge |\lambda_2| \ge \cdots$.

Also, if we use the matrix-valued 3-point C^2 non-spline interpolatory scheme for the regular boundary vertices, namely, $X = X_1, Y = \mathbf{0}, W = W_1, Z = Z_1$ with W_1, X_1, Z_1 given in (5.1), then the nonzero eigenvalues of $S_{0,0}$ are

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{16}.$$

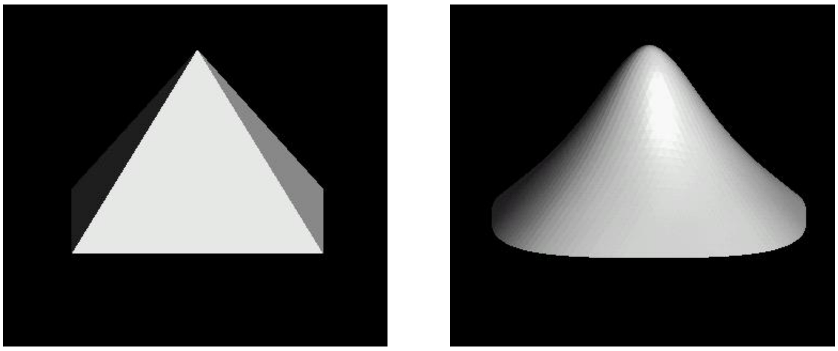

Figure 15: *Initial open mesh (left) and limiting surface (right)*

Thus, the eigenvalues λ_k of S also satisfy $\lambda_0 = 1 > |\lambda_1| \geq |\lambda_2| \geq \cdots$. On the right of Fig. 15, we show the limiting surface by applying this scheme with control parameters $s^0_{\mathbf{k}} = -v^0_{\mathbf{k}}$ to an initial (open) mesh shown on the left of Fig. 15 .

Observe that when $n > 7$, the first two largest eigenvalues $\frac{3}{8} + \frac{1}{4}\cos\frac{\pi}{n-1}$ and $\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n-1}$ of $S_{1,1}$ listed in (6.3) are bigger than $\frac{1}{2}$, an eigenvalue of S. So, it is likely that the resulting surface is not C^1 at boundary extraordinary vertices with large valences. To force the largest eigenvalue of $S_{1,1}$ to be $\frac{1}{2}$, one can use a modified template for the new vertices next to an extraordinary boundary vertex, as suggested in [15]. More precisely, for an extraordinary boundary vertex $\mathbf{x}_0$ with valence n, we may apply the template in Fig. 16 to $\mathbf{a}_2, \mathbf{a}_3, \cdots, \mathbf{a}_{n-1}$ in Fig. 14, where $B_1(n), B_2(n)$ are 2×2 matrices depending on n, and C is given in (6.1).

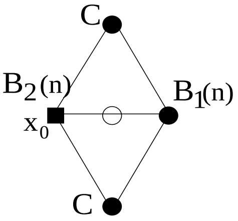

Figure 16: *Template for new ("odd") vertices next to an extraordinary vertex* $\mathbf{x}_0$

In this case the subdivision matrix S is still an upper-triangular block matrix with exactly the same nonzero matrices $S_{j,j}$ except $S_{1,1}$, which is now given by

$$S_{1,1} = \begin{bmatrix} B_2(n) & C & & & \\ C & B_2(n) & C & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & C \\ & & & C & B_2(n) \end{bmatrix}_{(n-2)\times(n-2)} .$$

If we choose

$$B_2(n) = \gamma_n B, \quad B_1(n) = (1-\gamma_n)B,$$

where γ_n is a real number and B is given by (6.1), then the nonzero eigenvalues of $S_{1,1}$ become

$$\frac{3}{8}\gamma_n - \frac{1}{4}\cos\frac{j\pi}{n-1}, \quad \frac{69}{512}\gamma_n + \frac{5}{256}\cos\frac{j\pi}{n-1}, \quad 1 \le j \le n-2.$$

Therefore, if γ_n is so chosen that $\frac{3}{8}\gamma_n + \frac{1}{4}\cos\frac{\pi}{n-1} = \frac{1}{2}$, namely,

$$\gamma_n = \frac{4}{3} - \frac{2}{3}\cos\frac{\pi}{n-1}, \tag{6.4}$$

then $\frac{1}{2}$ is an eigenvalue of $S_{1,1}$ and it is the largest eigenvalue in magnitude of $S_{1,1}$. Hence, with γ_n as given in (6.4), the eigenvalues of S are $1, \frac{1}{2}, \frac{1}{2}, \lambda_3, \lambda_4, \cdots$ with $\frac{1}{2} > |\lambda_3| \ge |\lambda_4| \ge \cdots$ for either the 4-point C^2 spline scheme or 3-point C^2 non-spline scheme to be used as the boundary scheme.

Appendix

Proof of Lemma 1. For $t = 0$, the result is trivial, since $M_n(s,t)$ becomes a diagonal matrix. In the following, we consider the case $t \ne 0$. Let $p_n(\lambda)$ =det$(M_n(s,t) - \lambda I_n)$ be the polynomial in λ with degree n. Clearly, $p_1(\lambda) = s - \lambda$, and with the notation $p_0(\lambda) := 1$, we have

$$p_n = (s-t)p_{n-1} - t^2 p_{n-2}, \quad n \ge 2.$$

To determine the zeros of $p_n(\lambda)$, write $s - \lambda = t(z + z^{-1})$, where $z = e^{-i\theta}$ for some $\theta \in \mathbb{R}\backslash\{0\}$. Then we have

$$p_n = (tz + tz^{-1})p_{n-1} - t^2 p_{n-2}.$$

Therefore,

$$\begin{aligned} &p_n - tz^{-1}p_{n-1} = tzp_{n-1} - t^2p_{n-2} = tz(p_{n-1} - tz^{-1}p_{n-2}) \\ &= (tz)^2(p_{n-2} - tz^{-1}p_{n-3}) = \cdots = (tz)^{n-1}(p_1 - tz^{-1}p_0) \\ &= (tz)^{n-1}(s - \lambda - tz^{-1}) = (tz)^{n-1}(tz + tz^{-1} - tz^{-1}) = (tz)^n. \end{aligned}$$

Hence, we have

$$\begin{aligned} p_n \quad &= tz^{-1}p_{n-1} + (tz)^n = tz^{-1}(tz^{-1}p_{n-2} + (tz)^{n-1}) + (tz)^n \\ &= (tz^{-1})^2 p_{n-2} + t^n z^{n-2} + t^n z^n = \cdots \\ &= (tz^{-1})^n p_0 + t^n z^{n-2(n-1)} + \cdots + t^n z^{n-2} + t^n z^n \\ &= t^n(z^{-n} + z^{n-2(n-1)} + \cdots + z^{n-2} + z^n) \\ &= t^n \frac{z^{n+1} - z^{-(n+1)}}{z - z^1} = t^n \frac{\sin(n+1)\theta}{\sin\theta}. \end{aligned}$$

Clearly, when $\theta = \theta_j = \frac{j\pi}{n+1}, 1 \le j \le n$, we have $\frac{\sin(n+1)\theta}{\sin\theta} = 0$. Thus, $\lambda_j = s - 2t\cos\theta_j$ are the zeros of p_n; and hence, they are the eigenvalues of $M_n(s,t)$. In addition, for $t \ne 0$, the n values $\lambda_j, 1 \le j \le n$ are distinct. Thus, $\lambda_j, 1 \le j \le n$, are the n eigenvalues of $M_n(s,t)$.

References

[1] 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.

[2] 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.

[3] 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.

[4] S. Dubuc, Interpolation through an iterative scheme, *J. Math. Anal. Appl.* 114 (1986), 185–204.

[5] N. Dyn, J.A. Gregory and D. Levin, A 4-point interpolatory subdivision scheme for curve designs, *Comput. Aided Geom. Design* 4 (1987), 257–268.

[6] M.F. Hassan, I. P. Ivrissimitzis, N. A. Dodgson, and M. A. Sabin, An interpolating 4-point C^2 ternary stationary subdivision scheme, *Comput. Aided Geom. Design* 19 (2002), 1–18.

[7] R.Q. Jia, and Q.T. Jiang, Approximation power of refinable vectors of functions, in: *Wavelet Analysis and Applications*, AMS/IP Stud. Adv. Math., Vol. 25, Amer. Math. Soc., Providence, RI, 2002, pp. 155–178.

[8] R.Q. Jia and Q.T. Jiang, Spectral analysis of transition operators and its applications to smoothness analysis of wavelets, *SIAM J. Matrix Anal. Appl.* 24 (2003), 1071–1109.

[9] S.S. Rana, Quadratic spline interpolation, *J. Approx. Theory* 57 (1989), 300–305.

[10] U. Reif and J. Peters, Structural analysis of subdivision surfaces–a summary, in *Topics in Multivariate Approximation and Interpolation*, K. Jetter et al. (Eds.), 2005.

[11] L.L. Schumaker, On shape preserving quadratic spline interpolation, *SIAM J. Numer. Anal.* 20 (1983), 854–864.

[12] J.E. Schweitzer, Analysis and application of subdivision surfaces, Ph.D. Thesis, University of Washington, Dept. of Computer Science and Engineering, Seattle, 1996.

[13] A. Weissman, A 6-point C^2 interpolatory subdivision scheme for curve design, M. Sc. Thesis, Tel-Aviv University, 1989.

[14] D.X. Zhou, Multiple refinable Hermite interpolants, *J. Approx. Theory* 102 (2000), 46–71.

[15] D. Zorin and P. Schröder, A. DeRose, L. Kobbelt, A. Levin, and W. Sweldens, *Subdivision for Modeling and Animation*, SIGGRAPH 2000 Course Notes.