Evaluation of Piecewise Smooth Subdivision Surfaces

Denis Zorin, Daniel Kristjansson
Media Research Lab, New York University,
719 Broadway, 12th floor,
New York, NY 10003
{dzorin,danielk}mrl.nyu.edu

Abstract

Keywords: Subdivision surfaces, Loop subdivision, surfaces with boundaries, creases, exact evaluation.

In this paper we consider the problem of constant-time evaluation of subdivision surfaces at arbitrary points. Our work extends the work of J. Stam, by considering the subdivision rules for piecewise smooth surfaces with boundaries depending on parameters. The main innovation of this paper is the idea of using a different set of basis vectors for evaluation, which, unlike eigenvectors, depend continuously on the coefficients of the subdivision rules. The advantage of this approach is that it becomes possible to define evaluation for parametric families of rules without considering excessive number of special cases, while improving numerical stability of calculations. We demonstrate how such bases are computed for a particular parametric family of subdivision rules extending Loop subdivision to meshes with boundary, and provide a detailed description of the evaluation algorithms.

‘

1 Introduction

Subdivision surfaces are defined by a set of rules which can be used to generate a sequence of increasingly fine meshes from an initial control mesh. This sequence of meshes converges to a limit surface. In general, it is not possible to evaluate the limit surface in closed form at arbitrary points as it is done with splines. However, the most commonly used approximating schemes are extensions of splines. For such schemes, evaluation is possible with such schemes and can be done with a constant-time algorithm.

There are a number of reasons why direct evaluation is desirable. For example, one may need to compute precisely an integral quantity associated with a surface to perform a finite element simulation, or, to evaluate the point on the surface at an arbitrary domain location for fitting or reparametrization. Furthermore, availability of "black-box" evaluators simplifies inclusion of subdivision in existing systems, avoiding reimplementation of such complex algorithms as surface-surface intersection.

Evaluation of subdivision surfaces was introduced by J. Stam for Catmull-Clark and Loop subdivision [14, 13] for surfaces without boundaries. J. Stam has also implemented evaluation of Catmull-Clark subdivision surfaces with boundary in Alias|Wavefront's Maya [1]. The details of the algorithm for surfaces with boundary remain unpublished.

The basic idea of Stam's approach takes advantage of the fact that the the surface near an extraordinary vertex can be represented as a linear combination of *eigenbasis functions*, which satisfy simple scaling relations, and can be easily evaluated in constant time. To obtain the coefficients in this decomposition, one needs to represent the vector of control points in a neighborhood of an extraordinary vertex in the eigenbasis of the subdivision matrix. As we show in this paper using eigenbasis for evaluation may result in numerically unstable calculations, in particular in the case of rules depending on parameters.

Examples of parametric families of rules include extensions of standard Catmull-Clark and Loop subdivision introduced in [4]. Special rules were designed to create convex and concave corners on the surface boundary, to create creases and to ensure C^1-continuity extraordinary vertices on the boundary. The parameters of the rules affect surface behavior near extraordinary points. While certain values of the parameters are preferable to others it is desirable to be able to perform evaluation without placing any restrictions on the choice of parameters. Typically, it is possible to find parameter values for which the matrix does not have a basis of eigenvectors, and generalized eigenvectors have to be used. Thus, many different cases have to be considered with different bases and evaluation algorithms used in

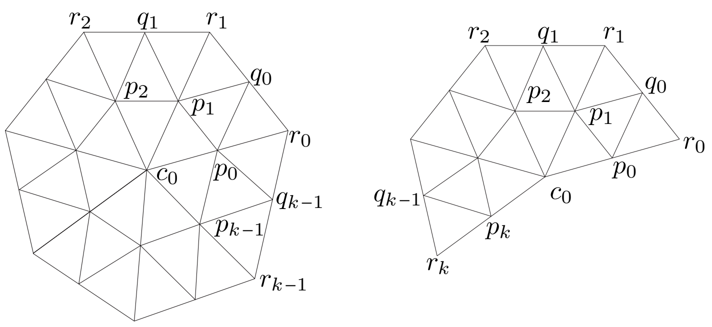

Figure 1: Notation for components of the vectors of control points.

each case, or certain parameter values should be excluded. Furthermore, even the standard Loop and Catmull-Clark rules depend on a parameter: the vertex valence. As a result, while the calculations based on eigenbasis decomposition remain valid, the result is likely to suffer from numerical problems for high valences.

In this paper, we describe a complete evaluation algorithm for piecewise smooth Loop subdivision surfaces introduced in [4], following general idea of Stam, but using a different normal form for the matrix. At the expense of a slightly more complex but still constant time calculation, we are able to consider parametric families of rules in a uniform manner, without treating the cases of derogatory (non-diagonalizable) subdivision matrices as special, as well as increase the numerical stability of the calculations.

Previous work. We have already mentioned the most closely related work of J. Stam, and the paper [4] where the subdivision surfaces that we consider in the paper were introduced. We should mention related work on subdivision schemes for surfaces with boundaries which goes all the way back to early paper on surface subdivision [5, 7, 10, 11]. Most closely related to our work is the thesis of J. Schweitzer [12] and the work of Hoppe et al. [9] where a different set of boundary rules for the Loop scheme was introduced; the differences between our approach and [9] are discussed in [4]. More recently, soft creases were introduced in [6]. Since after a finite number of subdivision steps soft crease schemes start using standard rules, in principle our evaluation scheme also applies.

The fact that generic parametric families of matrices almost always contain derogatory matrices is well known in mathematical literature; the normal form can be regarded as a simple case of the general normal form of families of matrices ([2, 8]).

2 Piecewise Smooth Subdivision

2.1 Subdivision and eigenanalysis

In this section, we briefly state several facts of the theory of subdivision [16], and introduce some basic notation. Consider a vertex v, and let p be the vector of control points in a neighborhood of the vertex (Figure 1). Everywhere in this paper k is used to denote the number of triangles sharing a vertex. Note that this is equal to the number of edges sharing a vertex in the interior case, and one less than the number of edges in the boundary case.

Let S_1 be the matrix of subdivision coefficients relating the vector of control points p^m on subdivision level m to the vector of control points p^{m+1} on a similar neighborhood on the next subdivision level. Suppose the size of the matrix is N. Many properties of the subdivision scheme can be deduced from the eigenstructure of S. We decompose the vector of control points p with respect to the eigenbasis $\{x^i\}$, $i = 0..N-1$, of S: $p = a_0x^0 + a_1x^1 + a_2x^2 + \ldots + a_{N-1}x^{N-1}$. The eigenbasis need not exist in general; in all cases of interest to us it does exist for *single-ring* subdivision matrix that we have described. The situation becomes more complex for the *complete* (double-ring) subdivision matrix S discussed below. Note that we decompose a vector of 3D points: the coefficients a_i are 3D vectors, which are componentwise multiplied with eigenvectors x^i.

In this discussion only, we assume that the eigenvectors x^i are arranged in the order of non-increasing eigenvalues (the order will be different for bases that we consider later). For a convergent scheme, the first eigenvalue λ_0 is 1, and the eigenvector x_0 has all components equal to one; this is also required for invariance with respect to rigid and, more generally, arbitrary affine transformations.

Subdividing the surface m times means that the subdivision matrix is applied m times to the control point vector p.

$$S_1^m p = \lambda_0^m a_0 x^0 + \lambda_1^m a_1 x^1 + \lambda_2^m a_2 x^2 + \cdots \lambda_{N-1}^m a_{N-1} x^{N-1}$$

It is well-known that the limit position of the control point at the center vertex is a_0; the tangent directions at this point are a_1 and a_2. We compute these values using the left eigenvectors of S (i.e., vectors l^i, satisfying $(l^i \cdot x^i) = 1$ and $(l^i \cdot x^j) = 0$ if $i \neq j$): $a_i = (l^i \cdot p)$.

Parameterization over the initial control mesh. Evaluation rules that we define evaluate the surface at arbitrary points; this implies that the surface is parameterized on a domain. Suppose the initial control mesh is a simple polyhedron, i.e., it does not have self-intersections. This assumption is not necessary, but it considerably simplifies the presentation. We will use the initial control mesh as the domain, and the describe the subdivision surface as a function on this domain. Suppose each time we apply the subdivision rules to compute the finer control mesh, we also apply midpoint subdivision to a copy of the initial control polyhedron. Note that each control point that we insert in the mesh using subdivision corresponds to a point in the midpoint-subdivided polyhedron. Another important fact is that midpoint subdivision does not alter the control polyhedron as a set of points. Finally we observe that all vertices inserted by midpoint subdivision are distinct.

As we repeatedly subdivide, we get a mapping from a denser subset of the control polygon to the control points of a finer control mesh. In the limit we get a map from the control polygon to the surface (Figure 2; the surface can be written now as a function $f(u, w, t, j)$ where (u, w, t), $u + w + t = 1$, are barycentric coordinates of a point in a triangle of the control mesh and j is the index of the triangle.

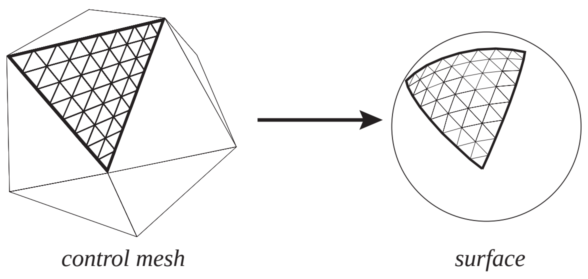

Figure 2: Natural parameterization of the subdivision surface

Complete subdivision matrix. The subdivision matrix we have defined so far does not completely determine the behavior of the surface on any neighborhood of a vertex. To define the surface on a ring of triangles surrounding a vertex one needs to consider a larger neighborhood (a 2-neighborhood in the case of Loop rules) and larger subdivision matrix which we call the *complete* subdivision matrix S; for interior vertices this matrix has size $3k + 1$, for boundary vertices $3k + 3$. As shown in Figure 1, there are three types of control points in the 2-neighborhood, denoted p_i, q_i and r_i respectively. The matrix operates on the complete vector P of control points. We assume that the control points in P are ordered as follows: $P = [c, p_0, p_1, \ldots, q_0, q_1, \ldots, r_0, r_1 \ldots]$.

2.2 Subdivision Rules

We start with a review of the subdivision rules proposed in [4]. We assume that one subdivision step was performed, and no two extraordinary vertices are adjacent. Handling of the first subdivision step is described in [4].

Tagged meshes. Subdivision operates on *tagged meshes*. Edges can be tagged as *crease edges*. We require that all edges on the boundary of the mesh are tagged as crease edges. Some vertices with incident crease edges may be tagged as *corner*, which means that the control point at this vertex should remain fixed. If there are more than two crease edges meeting at a vertex, we require it to be a corner vertex. We do not consider dart vertices, i.e. vertices with a single incident crease edge.

At a corner vertex, the creases meeting at that vertex separate the ring of triangles around the vertex into sectors. We label each sector of the mesh as *convex sector* or *concave sector* indicating how the surface should approach the corner (see [4] for details). We require concave sectors to consist of at least two faces.

In addition to tags, the rules can use a *flatness parameter* for untagged vertices, crease vertices, and for each sector at a corner vertex.

The flatness parameter determines how quickly the surface approaches the tangent plane in the neighborhood of a control point. This parameter is essential to concave corner rules, and if one wants to ensure that the surface is C^2 at an extraordinary vertex, but can be assumed to be zero in all other cases. For concave corners, a reasonable default value is $1/(4\lambda_1(\theta))$ with $\lambda_1(\theta)$ defined below.

Subdivision rules. Here we briefly review these rules for completeness; for details, see [4].

The vertex rules are chosen as follows (Figure 3).

- for crease non-corner vertices we use B-spline subdivision rules;
- for corner vertices the control points remain fixed;
- for regular vertices we use standard Loop rules.

Edge rules. The rules consist of two stages: subdivision and flatness modification. The stencils for the edge rules are shown in Figure 3. Several cases have to be distinguished:

- Vertex inserted on a crease edge. The spline rule (average of adjacent control points) is used.
- Vertex inserted next to a corner vertex, but not on a crease, and the vertex is inside a convex sector. In this case modified Loop rules are used with the parameter $\theta < \pi/k$. Flatness modification is not necessary in this case.
- Vertex inserted next to a corner vertex, but not on a crease, and the vertex is inside a concave sector. In this case we pick $\theta > \pi/k$ and flatness modification is required to ensure C^1-continuity. case. Vertex is inserted next to a non-corner extraordinary crease vertex, not on a crease. $\theta = \pi/k$ is used, flatness modification step can help to improve the surface, but is not necessary to achieve C^1-continuity.
- in all other cases, standard Loop rules are used.

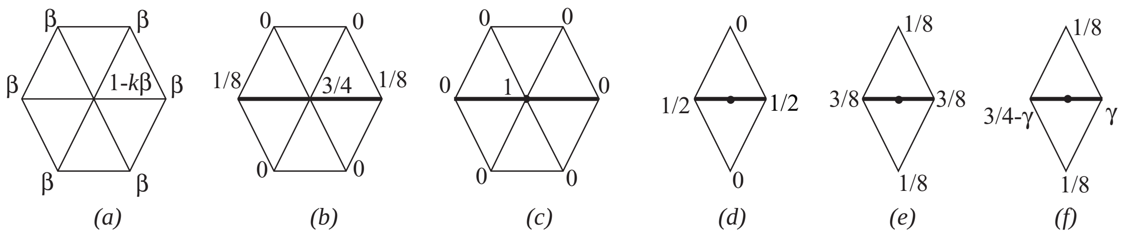

Figure 3: Vertex rules: (a) Standard vertex rule, k is the valence of center vertex. If $k \neq 3$, $\beta = 3/8k$, if $k = 3$, $\beta = 3/16$. (b) Uniform cubic spline rule. (c) Interpolation rule. Edge rules: (d) Uniform cubic spline rule. (e) Standard Loop rule. (f) Modified edge rule, θ is the parameter for different cases, values are specified in the text. In the picture $\gamma = 1/2 - 1/4\cos\theta$.

Stencils used at the first stage of edge subdivision rules are shown in Figure 3 (d) to (f).

The second stage for edge rules is flatness modification, which is required only for concave corner rules in which case it is necessary for C^1-continuity. For corner vertices flatness modification is given by

$$p^{new} = (1-s)p + s(a_0x^0 + a_1x^1 + a_2x^2)$$

It is also useful for smooth crease vertices, in which case we use a rule slightly different from the one described in [4]:

$$p^{new} = (1-s)p + s(a_0x^0 + a_1x^1 + a_2x^2 + a_3x^3)$$

where x_3 is an eigenvector corresponding to the eigenvalue $1/4$; this change has little impact on the surface appearance, but considerably simplifies evaluation (see below).

The subdominant eigenvector configurations are shown in Figure 4 for various cases.

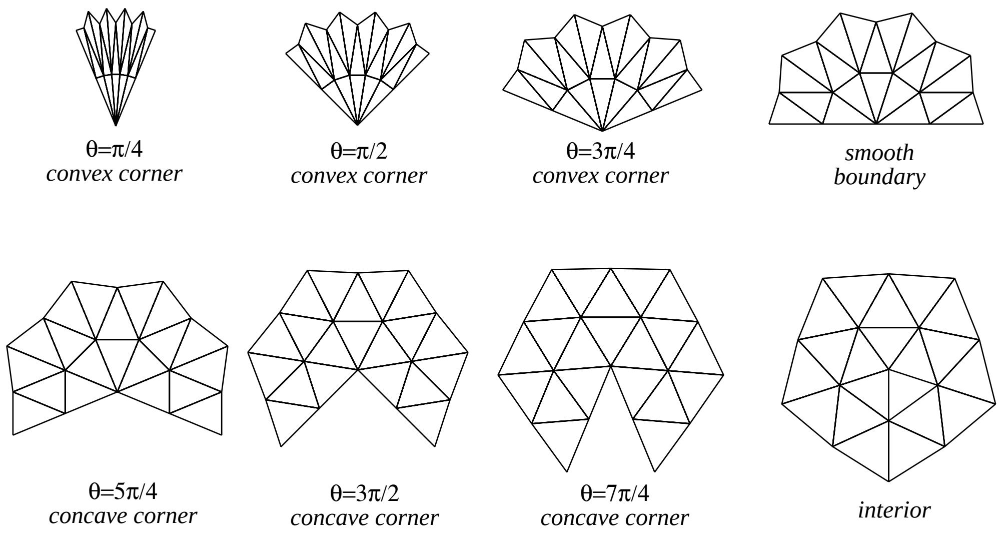

Figure 4: Invariant configurations for various rules and values of parameter θ. The configurations are obtained by using the components of the first subdominant eigenvector as one coordinate, and the components of the second as the second coordinate. Subdominant eigenvalues are $1/2$ in all cases excluding interior vertices, in which case it is $3/8 + (1/4)\cos(2\pi/k)$. All shown cases correspond to $k = 5$.

3 Overview of the Evaluation Algorithm

We start with a precise formulation of the problem. Given a triangle j on the control mesh and barycentric coordinates (u, w, t), $u + w + t = 1$, of a point in the triangle, find the value of the subdivision surface $f(u, w, j)$ at this point.

As we assume that one subdivision step was already performed, only one of the vertices of the triangle j can be extraordinary. If no vertices are extraordinary, one can easily see that after one more subdivision step, the control mesh for the triangle does not contain any extraordinary vertices, and the surface can be evaluated at any point of the triangle directly as shown in Figure 5 (a). If the triangle contains a single extraordinary vertex, we assume that it corresponds to barycentric coordinate t. From now on, we focus our attention on this case. Recall that P denotes the vector of control points in 2-neighborhood of an extraordinary vertex, and S is the complete subdivision matrix acting on P.

The steps of our evaluation algorithm are similar to the steps of the algorithm of [14]. The main difference is in the implementation of the crucial step, namely, evaluation of S^mP.

Consider any point $y = (j, u, w)$ in the domain of the surface different from an extraordinary vertex. The first crucial observation is that there is a sufficiently fine subdivision level m such that the point y is located in a triangle T on level m, and the control mesh of the triangle T does not contain extraordinary vertices. This means that on the triangle T the surface is polynomial and can be evaluated explicitly, including the value at y. Clearly, m is larger for points which are close to extraordinary vertices. It is convenient to think about points in 1-neighborhood of an extraordinary vertex as arranged in layers (Figure 5(b)). For each layer, the number of subdivision steps that have to be applied is fixed.

The second important observation behind the evaluation method is that the control meshes for triangles on level m which are close to an extraordinary vertex can be computed with the help of the complete subdivision matrix,

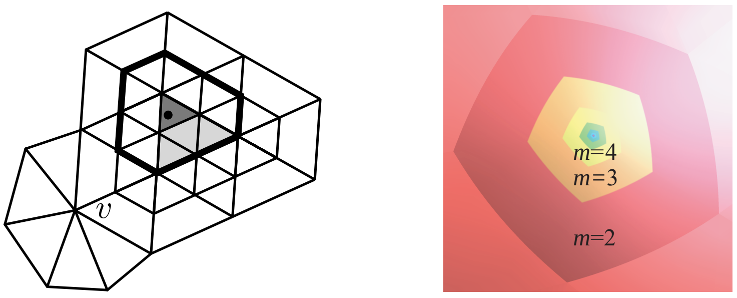

Figure 5: (a) On a triangle with no extraordinary vertices subdivision surface can be evaluated directly after one subdivision step. The initial triangle is light gray, the evaluation point is marked with a dot. The outlined area is the control mesh for the subtriangle where the point is located. Note that a single subdivision is necessary because the control mesh may contain extraordinary vertices such as v. (b) For each layer of patches near an extraordinary vertex a certain number of subdivision steps is required for evaluation.

specifically, using $S^m P$, as defined in Section 2.1. Moreover, $S^m P$ can be computed in constant time. If S is diagonalizable, i.e. $S = R\Lambda R^{-1}$, where the matrix Λ is diagonal, then $S^m P = R\Lambda^m R^{-1} P$. It is easy to see that the time required to evaluate the latter expression is constant, as $\Lambda^m = diag(\lambda_0^m, \dots \lambda_{N-1}^m)$.

Now we describe the steps of the algorithm in greater detail, excluding evaluation of $S^m P$; this part is discussed in Section 4. All other steps of the algorithm do not depend on the choice of subdivision coefficients at or near extraordinary vertices; they depend only on the support of the rules, and on where the special rules are applied.

The following steps are illustrated in Figure 6.

1. Determine the layer number for (u, w): $m = \lfloor -\log_2(u + w) \rfloor$. This means that after m subdivision steps, and conversion of u, w to the barycenteric coordinates (u', w') in the triangle on level m, we get $u' + w' > 1/2$. On level m the point (u, w) is in one of three subtriangles of level $m + 1$ of a triangle with extraordinary vertex, which do not contain the extraordinary vertex. We call the union of these three subtriangles a *trapezoidal domain* T^m. The goal of the next steps is to compute the control mesh for this domain which does not contain the extraordinary vertex, which means we need to subdivide to level $m + 2$.

2. Compute $S^m P$, (Section 4). Extract 10 control points (8 for boundary triangles) as shown in Figure 6.

3. We need to subdivide two more times to obtain a control mesh which has no extraordinary points. First,we use $S^{m+1} P$ to get control points on level $m + 2$ in 2-neighborhood of the extraordinary vertex; then we use regular subdivision to get all other control points for the trapezoidal domain T^m. For trapezoidal domains adjacent to the boundary one can use reflection to obtain missing points ([12]). The resulting control mesh has 28 points.

4. Determine which of the 12 subtriangles of T^m on level $m + 2$ contains (u, w). From the 28-point mesh extract the 12-point control mesh of the subtriangle.

5. Convert (u, w) to the barycentric coordinates of the subtriangle; Use quartic box spline evaluation (Appendix B) to compute the value at (u, w, j).

4 Computing Powers of the Subdivision Matrix

4.1 Alternative to Diagonalization

As it was pointed out in the previous section, the crucial element of the evaluation algorithm is the constant-time evaluation of $S^m P$. For diagonalizable matrices, the standard approach is to use the matrix of eigenvectors R and left eigenvectors L: $S^m = R\Lambda^m L$ where Λ is the diagonalized form of the matrix. As we have mentioned this approach, while working for fixed matrices, may lead to problems if we consider families of matrices. This is a well-known fact, but as it is of fundamental importance to us we consider a simple example to clarify the difficulty.

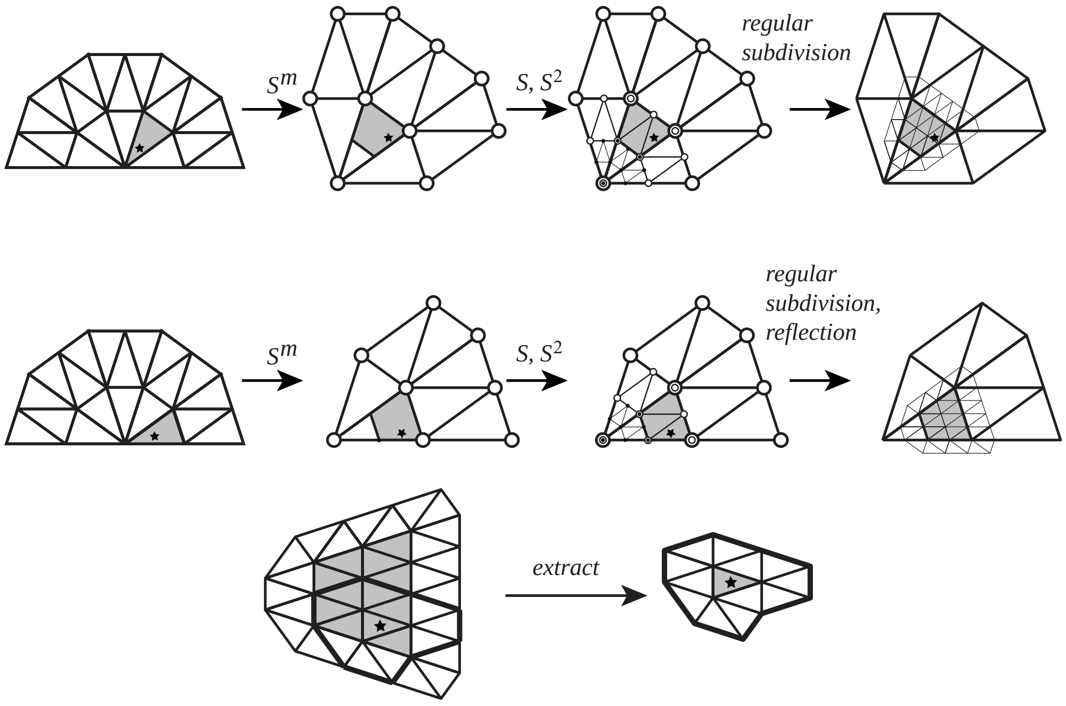

Figure 6: Computing the control mesh for a patch for evaluation. Large empty circles denote control points on level m, smaller circles on level $m+1$, and black dots on level $m+2$.

Consider a 2×2 matrix with one of the eigenvalues depending on a parameter ϵ, converted to diagonal form for $\epsilon \neq 0$:

$$M = \begin{pmatrix} \lambda+\epsilon & 0 \\ 1 & \lambda \end{pmatrix} = \begin{pmatrix} \epsilon & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda+\epsilon & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} 1/\epsilon & 0 \\ -1/\epsilon & 1 \end{pmatrix}$$

Now it is easy to see one aspect if the problem: presence of $1/\epsilon$ for small ϵ potentially introduces numerical problems. Even more importantly for $\epsilon = 0$ we need to use a different approach, as the matrix is no longer diagonalizable. An explicit formula using the basis of generalized eigenvector can be obtained. However, if we have a large matrix, such as subdivision matrix with many eigenvalues depending on parameters handling all special cases when multiple eigenvalues coincide is rather inconvenient.

We propose an alternative approach based on the observation that any basis which allows to compute $S^m P$ in constant time would suffice. Continuing the simple example above, we observe that the m-th power can be computed directly. Indeed, if $\lambda_1 = \lambda + \epsilon$ and $\lambda_2 = \lambda$, then

$$M^m = \begin{pmatrix} \lambda_1^m & 0 \\ \sum_{i=0}^{m-1} \lambda_1^{m-i-1} \lambda_2^i & \lambda_2^m \end{pmatrix} = \begin{pmatrix} \lambda_1^m & 0 \\ (\lambda_2^m - \lambda_1^m)\,/\,(\lambda_2 - \lambda_1) & \lambda_2^m \end{pmatrix}$$

Assuming one of the eigenvalues is sufficiently different from zero (say, λ_1) the expression in the lower left corner of the resulting matrix can be reliably computed as $\lambda_1^{m-1} g(\lambda_2/\lambda_1, m)$ using the function g defined by $g(x, n) = (x^n - 1)/(x - 1)$ for $x \neq 1$, $g(1, n) = 0$

Note that the computation is exactly the same no matter what the Jordan normal form of the matrix is.

In the case of Loop subdivision, this simple observation turns out to be all we need to design an algorithm which can handle parameterized rules in a stable way without excessive number of special cases.

The general idea of the algorithm, which is described in greater detail in the next sections, is the following. One can find a basis computed in a uniform way for any of the three types of extraordinary vertex (interior, boundary corner and boundary smooth) such that the subdivision matrix has block diagonal form, with blocks on the diagonal being either scalars, or 3×3 upper triangular real matrices. In this way, the problem of computing S^m is reduced to the problem of computing B^m for each 3×3 block.

The blocks have the form

$$B = \begin{pmatrix} \lambda(\theta) & 0 & 0 \\ a_1 & \lambda_1 & 0 \\ a_2 & a_3 & \lambda_2 \end{pmatrix}$$

with λ_1 and λ_2 fixed, $\lambda_1 \neq \lambda_2$ and dependence on the parameter θ present only for corner rules.

We observe that a simple change of basis reduces the matrix to the form similar to the matrix M considered above:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & u & 1 \end{pmatrix}; \quad R^{-1}BR = \begin{pmatrix} \lambda(\theta) & 0 & 0 \\ a_1 & \lambda_1 & 0 \\ b_1 & 0 & \lambda_2 \end{pmatrix} = B'$$

for $u = a_3/(\lambda_1 - \lambda_2)$, $b_1 = a_2 - ua_1$. For B' any power can be computed using the function $g(x, m)$ defined above:

$$(B')^m = \begin{pmatrix} \lambda(\theta)^m & 0 & 0 \\ a_1\lambda_1^{m-1} g\left(\lambda(\theta)/\lambda_1, m\right) & \lambda_1^m & 0 \\ b_1\lambda_2^{m-1} g\left(\lambda(\theta)/\lambda_2, m\right) & 0 & \lambda_2^m \end{pmatrix} \tag{4.1}$$

Again, this expression has the same form for all three possible Jordan forms of B.

In the following section we explain in greater detail how this approach is used to compute $S^m p$ for the three types of rules: interior, smooth boundary and corner.

4.2 Reduction of the subdivision matrix to block-diagonal form

Interior rules. This case is most studied and best understood. The standard approach is to use DFT and a permutation to reduce the subdivision matrix to block diagonal form (see for example [15] for details). However the result is a matrix with complex entries, which is computationally inconvenient. It is better to use the real and imaginary parts of the DFT basis to obtain real blocks on the diagonal. The transformation matrix composed out of real and imaginary parts of the columns of the DFT matrix is given by

$$T_{ij} = \begin{cases} 1/\sqrt{2}, & \text{if } j = 0 \\ (-1)^i/\sqrt{2}, & \text{if } j = k/2 \\ \cos\left(2\pi\, ij/k\right), & \text{if } 0 < j < k/2 \\ \sin\left(2\pi\, i(j - \lfloor k/2 \rfloor)/k\right), & \text{if } j < k/2 \end{cases}$$

Note that the second alternative is possible only for even columns. The matrix T is orthogonal up to a factor $2/k$ i.e. $T^{-1} = (2/k)T^T$. After the transformation diag$(1, T, T, T)$ is applied to the subdivision matrix, followed by permutation of entries rearranging the vector of control points in the order $0, 1, k+1, 2k+1, 2, k+2, 2k+2, \ldots$ we obtain $k-1$ block of the form

$$B(c) = \begin{pmatrix} \frac{3}{8} + \frac{1}{4}(2c^2 - 1) & 0 & 0 \\ \frac{3}{4}c & \frac{1}{8} & 0 \\ \frac{c^2}{4} + \frac{1}{2} & \frac{c}{8} & \frac{1}{16} \end{pmatrix}$$

where $c = \cos(j\pi/k)$, $j = 1 \ldots k-1$.

Smooth boundary and corner rules. We consider this case in greater detail, as it was not presented in the literature. We assume that $k > 1$; we consider the case $k = 1$ separately. For $k = 1$ the matrix does not depend on parameter θ and evaluation can be done using the standard eigenbasis approach. The subdivision matrix for a boundary vertex with k adjacent triangles has the following form:

$$\left(\begin{array}{c|cc|ccccc|cccccc|cc|ccccc}
1-2\alpha & \alpha & \alpha & {}&{}&{}&{}&{} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\hline
1/2 & 1/2 & {} & {}&{}&{}&{}&{} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
1/2 & {} & 1/2 & {}&{}&{}&{}&{} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\hline
\epsilon & 1/8 & {} & \gamma & 1/8 & {} & {} & {} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\epsilon & {} & {} & 1/8 & \gamma & 1/8 & {} & {} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\cdot & {} & {} & {} & \cdot & \cdot & \cdot & {} & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\epsilon & {} & {} & {} & {} & 1/8 & \gamma & 1/8 & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\epsilon & {} & 1/8 & {} & {} & {} & 1/8 & \gamma & {}&{}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\hline
1/8 & 3/8 & {} & 3/8 & {} & {} & {} & {} & 1/8 & {}&{}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
1/8 & {} & {} & 3/8 & 3/8 & {} & {} & {} & {} & 1/8 & {}&{}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\cdot & {} & {} & {} & \cdot & \cdot & {} & {} & {} & {} & \cdot & {}&{}&{} & {}&{} & {}&{}&{}&{}&{} \\
\cdot & {} & {} & {} & {} & \cdot & \cdot & {} & {}&{}&{} & \cdot & {}&{} & {}&{} & {}&{}&{}&{}&{} \\
1/8 & {} & {} & {}&{}&{} & 3/8 & 3/8 & {}&{}&{}&{} & 1/8 & {} & {}&{} & {}&{}&{}&{}&{} \\
1/8 & {} & 3/8 & {}&{}&{}&{} & 3/8 & {}&{}&{}&{}&{} & 1/8 & {}&{} & {}&{}&{}&{}&{} \\
\hline
1/8 & 3/4 & {} & {}&{}&{}&{}&{} & {}&{}&{}&{}&{}&{} & 1/8 & {} & {}&{}&{}&{}&{} \\
1/8 & {} & 3/4 & {}&{}&{}&{}&{} & {}&{}&{}&{}&{}&{} & {} & 1/8 & {}&{}&{}&{}&{} \\
\hline
1/16 & 1/16 & {} & 5/8 & 1/16 & {}&{}&{} & 1/16 & 1/16 & {}&{}&{}&{} & {}&{} & 1/16 & {}&{}&{}&{} \\
1/16 & {} & {} & 1/16 & 5/8 & 1/16 & {}&{} & {} & 1/16 & 1/16 & {}&{}&{} & {}&{} & {} & 1/16 & {}&{}&{} \\
\cdot & {}&{} & {}&{} & \cdot & {}&{} & {}&{} & \cdot & \cdot & {}&{} & {}&{} & {}&{} & \cdot & {}&{} \\
1/16 & {}&{} & {}&{} & 1/16 & 5/8 & 1/16 & {}&{}&{} & 1/16 & 1/16 & {} & {}&{} & {}&{}&{} & 1/16 & {} \\
1/16 & {} & 1/16 & {}&{}&{} & 1/16 & 5/8 & {}&{}&{}&{} & 1/16 & 1/16 & {}&{} & {}&{}&{}&{} & 1/16
\end{array}\right) \qquad (4.2)$$

where $\epsilon = 3/4 - \gamma$, $\alpha = 0$ for corner rules and $\alpha = 1/8$ for smooth boundary rules. In block form this matrix can be written as

$$\left(\begin{array}{c|cc|c|c|cc|c}
1-2\alpha & \alpha & \alpha & & & & & \\
\hline
1/2 & 1/2 & {} & & & & & \\
1/2 & {} & 1/2 & & & & & \\
\hline
a_1 & \multicolumn{2}{c|}{A_{10}} & A_{11} & & & & \\
\hline
a_2 & \multicolumn{2}{c|}{A_{20}} & A_{21} & \frac{1}{8}I_k & & & \\
\hline
1/8 & 3/4 & {} & & & 1/8 & {} & \\
1/8 & {} & 3/4 & & & {} & 1/8 & \\
\hline
a_3 & & & A_{31} & A_{32} & & & \frac{1}{16}I_{k-1}
\end{array}\right) \qquad (4.3)$$

The vectors a_1 and a_3 have length $k-1$, the vector a_2 has length k, I_k and I_{k-1} are unit matrices of sizes k and $k-1$. The eigenvalues of the matrix can be separated into the following groups: the eigenvalues of the upper-left 3×3 block, the eigenvalues of the matrix A_{11}, and multiple eigenvalues $1/8$,$1/16$.

However, we are still facing the daunting task of choosing the basis in which the matrix matrix S has a block diagonal form. Most of the calculations are rather tedious and unenlightening, so we provide only a brief outline and the final result in the appendix. Verification of correctness was done using Maple V symbolic algebra system; the worksheet with the calculations is available from the authors.

Construction of a suitable basis proceeds in two steps. First, we include all the eigenvectors of S that have nonzero components on the boundary. After this, we consider the restriction of S to the interior points only, and find a basis in which it has a block diagonal form.

1. There are five vectors with nonzero entries on the boundary; these are obtained from the eigenvectors of the subdivision matrix restricted to the boundary. Assuming the order of entries c, p_0, p_k, r_0, r_k, we choose the following four eigenvectors: $[1,1,1,1,1]$ for eigenvalue 1, $[0,-1,1,-1,2,-2]$ for eigenvalue $1/2$, and $[0,0,0,1,0]$ and $[0,0,0,0,1]$ for eigenvalue $1/8$. In addition, we use $[0,1,1,2,2]$ (eigenvalue $1/2$) for corner rules and $[-1/2,2,2,11/2]$ (eigenvalue $1/4$) for smooth boundary rules. These vectors can be extended to the rest of the control points by solving recurrences following [12]. The corresponding eigenvectors are denoted v_0, v_1, v_2, v_r^0, v_r^k.

2. Once the five eigenvectors listed above are included, we can assume that all control values on the boundary to be zero, and consider the restriction of the subdivision matrix acting on the interior control points only. This matrix has a nicer structure and can be converted to block diagonal form using a system of eigenvectors which does not depend on the matrix, similar to the the DFT columns we use for the interior case. However, now we use two variations of the discrete sine transform: $(k-1)\times(k-1)$ matrix $D^1_{ij} = \sin(ij\theta_k)$, $i,j = 1\dots k-1$, $\theta_k = \pi/k$, and $k \times k$ matrix $D^2_{ij} = \sin((i+1/2)j\theta_k)$, $i = 0\dots k-1, j = 1\dots k-1$, $D^2_{i0} = (-1)^i$. Transformation by the matrix diag(D^1, D^2, D^1) followed by a permutation reduces the restricted subdivision matrix to the block diagonal form, with k 3×3 blocks $B(c,\theta)$ on the diagonal nearly identical to the blocks obtained for interior rules above:

$$B(c,\theta) = \begin{pmatrix} \lambda(\theta) & 0 & 0 \\ \frac{3}{4}c & \frac{1}{8} & 0 \\ \frac{c^2}{4}+\frac{1}{2} & \frac{c}{8} & \frac{1}{16} \end{pmatrix}$$

where $c = \cos(j\theta_k/2)$, with j being the block number, and $\lambda(\theta) = 1/2 + 1/4(\cos j\theta_k - \cos\theta)$, and a single 1×1 block $1/8$, which corresponds to the column D^2_{i0}.

Observe that $B(c,\theta)$ have exactly the form that we considered in Section 4.1; an additional simple transformation reduces each block to the form for which powers can be computed in constant time. In fact, with proper rescaling of the basis vectors the blocks are reduced to the form

$$B(\theta)^{\text{normal}} = \begin{pmatrix} \lambda(\theta) & 0 & 0 \\ 1 & \lambda_1 & 0 \\ 1 & 0 & \lambda_2 \end{pmatrix} \tag{4.4}$$

where $\lambda_1 = 1/8$ and $\lambda_2 = 1/16$. The basis in which the restricted subdivision matrix has this form is a simple combination of the columns of diag(D^1, D^2, D^1); specifically, for block j, $j = 1\dots k-1$, we use three basis vectors v_j^λ, v_j^q and v_j^r, defined in the appendix.

4.3 Putting it all together

Now that we have a set of bases in which our matrices have block-diagonal form with blocks of type (4.4), we can specify the expressions for computing S^m. From (4.1) we see that for a single block

$$B(\theta)^m \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \lambda(\theta)^m a_1 \\ \lambda_1^{m-1} g(\lambda(\theta)/\lambda_1, m) a_1 \lambda_1^m a_2 \\ \lambda_2^{m-1} g(\lambda(\theta)/\lambda_2, m) a_1 \lambda_2^m a_3 \end{bmatrix}$$

This immediately leads to the following complete formulas for computing $S^m p$, keeping in mind that in the boundary case (five boundary and one interior) irregular basis vectors are eigenvectors. Just as in the case of Stam's algorithm we start with computing the decomposition of p with respect to the basis, using corresponding left eigenvectors (see appendix): $a_i = (l_i \cdot P)$, $i = 0\dots 2$, $a_i^\lambda = (l_i^\lambda \cdot P)$, $i = 1\dots k-1$, $a_i^q = (l_i^q \cdot P)$, $i = 0\dots k-1$, $a_i^r = (l_i^r \cdot P)$, $i = 0\dots k-1$. Then

$$P = a_0 v_0 + a_1 a_1 v_1 + a_2 v_2 + a_0^q v_0^q + a_0^r v_0^r + a_k^r v_k^r + \sum_{i=1}^{k-1} a_i^\lambda v_i^\lambda + a_i^q v_i^q + a_i^r v_i^r \tag{4.5}$$

$$
\begin{aligned}
S^m P =& a_0 v_0 + \mu_1^m a_1 a_1 v_1 + \mu_2^m a_2 v_2 + \mu_q^m (a_0^q v_0^q + a_0^r v_0^r + a_k^r v_k^r) \\
&+ \sum_{i=1}^{k-1} \lambda_j^m a_i^\lambda v_i^\lambda + \mu_q^{m-1} \left(\mu_q a_i^q + g\left(\frac{\lambda_i}{\mu_q}, m\right) a_i^\lambda \right) v_i^q + \mu_r^{m-1} \left(\mu_r a_i^r + g\left(\frac{\lambda_i}{\mu_r}, m\right) a_i^\lambda \right) v_i^r
\end{aligned}
\tag{4.6}
$$

where $\mu_1 = 1/2$ $\mu_2 = 1/2$ for corner rules and $\mu_2 = 1/4$ for smooth rules, $\mu_q = 1/8$, $\mu_r = 1/16$.

For the interior points the expressions are slightly different, as there is only one special vector:

$$
\begin{aligned}
S^m P =& a_0 v_0 + \\
&+ \sum_{i=0}^{k-1} \lambda_i^m a_i^\lambda v_i^\lambda + \mu_q^{m-1} \left(\mu_q a_i^q + g\left(\frac{\lambda_i}{\mu_q}, m\right) a_i^\lambda \right) v_i^q + + \mu_r^{m-1} \left(\mu_r a_i^r + g\left(\frac{\lambda_i}{\mu_r}, m\right) a_i^\lambda \right) v_i^r
\end{aligned}
\tag{4.7}
$$

with $\lambda_0 = 1/4 + \gamma - k\beta$, $\mu_2 = 1/8$ and $\mu_3 = 1/16$.

The evaluation process can be thought of as evaluating the surface as a linear combination of simpler surfaces, for which evaluation is easier, specifically, the surfaces that we get when the coordinates of the control points are components of the basis vectors. It is useful to examine these special "basis" surfaces, to understand the relative contribution of different basis vectors in the decomposition. Figures 7 and 8 show two examples of such bases.

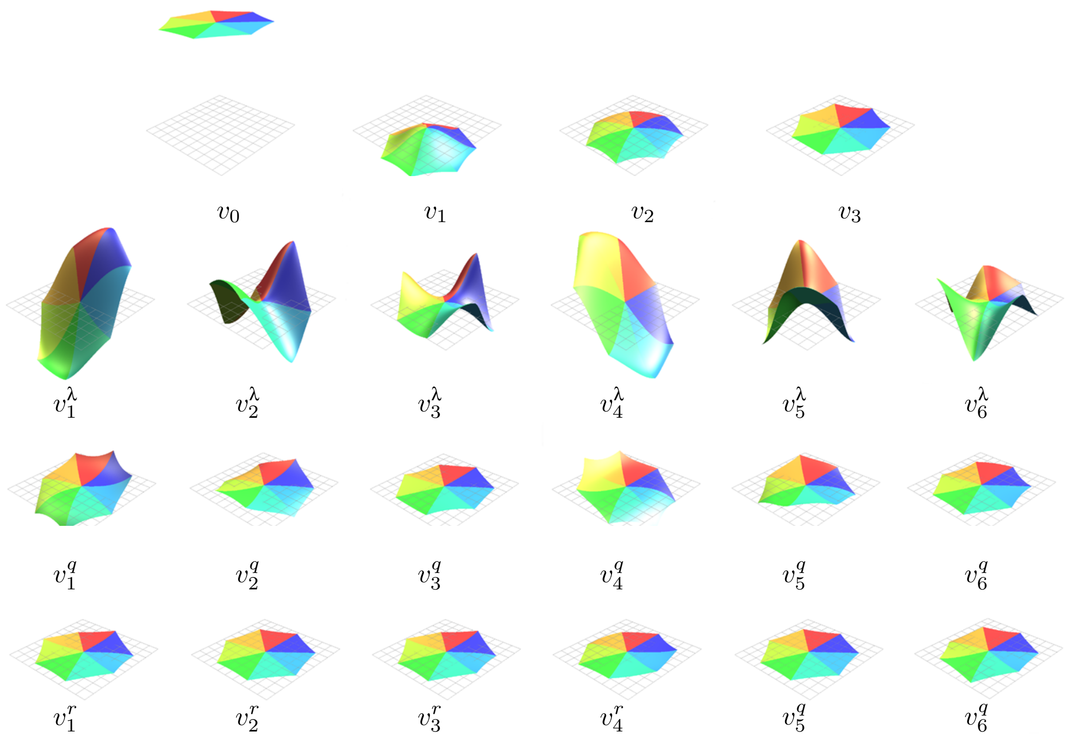

Figure 7: Set of limit functions generated by subdivision from the basis vectors for an interior vertex of with $k = 7$.

4.4 Effect of flatness modification

Flatness modification seemingly may require considerable changes to our evaluation procedure. However, it turns our that for our specific choice of basis, *the only* modification required is scaling of some of the eigenvalues. Here we briefly explain why this is the case, and specify which eigenvalues have to be scaled.

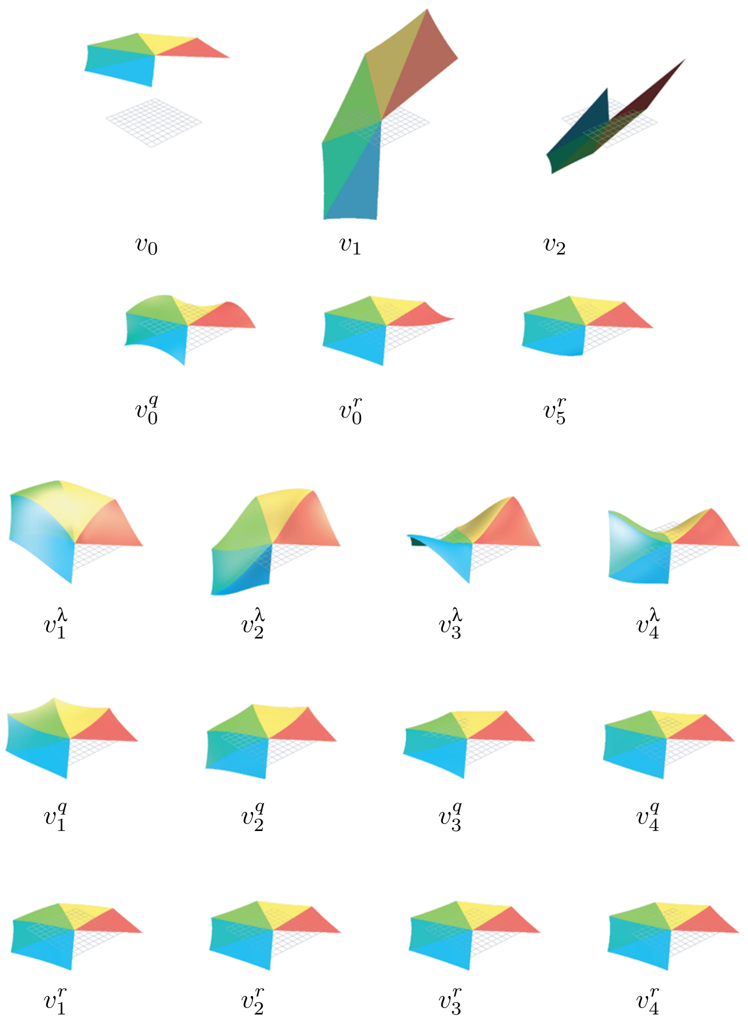

Figure 8: Set of limit functions generated by subdivision from the basis vectors for an interior vertex of with $k = 7$.

First, we note that flatness modification applies only to the control points immediately adjacent to the extraordinary vertex; the modification scales these entries by $(1 - s)$ for all components in the decomposition (4.5) excluding $a_0 v_0$ and two components corresponding to subdominant eigenvalues: $a_1 v_1 + a_2 v_2$ for corner rules, $a_1 v_1 + a_1^\lambda v_1^\lambda$ for smooth boundary rules, $a_1^\lambda v_1^\lambda + a_n^\lambda v_n^\lambda$ for the interior rules with $n = \lfloor k/2 \rfloor + 1$. In addition, the component $a_2 v_2$ does not get scaled by the deviation from [4] we adopted.

We observe that only components that are scaled correspond to v_i^λ, v_i^r, v_j^q $i = 0 \ldots k$ (boundary), $i = 0 \ldots k - 1$ (interior), $j = 0 \ldots k - 1$.

Out of these components, v_j^q and v_i^r have zero entries corresponding to the interior ring. This means scaling has no effect on these components. On the contrary, the only nonzero components of v_i^λ are in the interior ring. Thus scaling the interior ring components is equivalent to scaling the whole vector.

Thus, the net result of the modification is equivalent to scaling the eigenvalues λ_j excluding subdominant by the factor $(1 - s)$. This means that the formulas (4.6) and (4.7) still can be used, with eigenvalue scaling, for both interior and boundary vertices. As we have observed the following eigenvalues need to be scaled by $(1 - s)$:

- interior rules: all λ_j, excluding two subdominant;
- corner rules: all $\lambda_j(\theta)$;

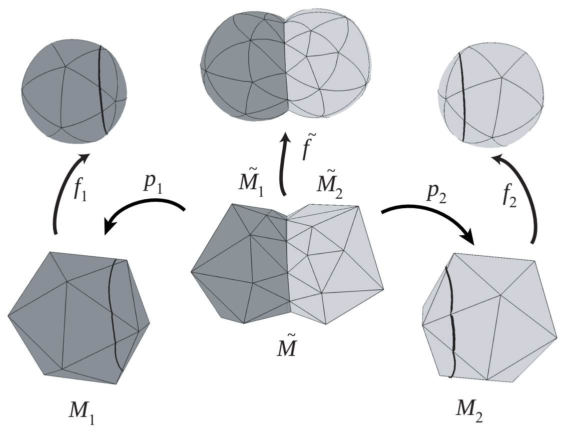

Figure 9: An example application: boolean operations on subdivision surfaces. Maps and domains used in fitting a subdivision surface to the result.

- smooth boundary rules: all λ_j excluding the only subdominant eigenvalue $\lambda_1 = 1/2$.

5 Applications of Evaluation

As we have mentioned direct evaluation of subdivision has a number of applications, including computing quadratures, intersection curves of surfaces and reparametrization.

Here we give a brief overview one of the applications that served as a motivation for our work, and for which we have used our evaluation code. Boolean operations (intersections, unions, differences) on closed surfaces are commonly used in computer-aided design to construct objects. In general, the result of such operations can be computed only approximately, in particular, if the solids are bounded by subdivision surfaces. It is desirable to construct an approximation of the result of a boolean operation represented in the same way as the original surfaces. This means that we need to construct a new control mesh, and fit the surface defined by th new mesh to the parts of old surfaces. In a separate paper [3] we describe in detail how these operations are performed. Once the new mesh is constructed, we end up with the new control mesh with two submeshes $\tilde{M} = \tilde{M}_1 \cup \tilde{M}_2$, old control meshes M_1 and M_2 and maps $p_i : \tilde{M}_i \to M_i$, $i = 1, 2$, extablishing the correspondence. The maps p_i need not map triangles to triangles; a vertex of $\tilde{M}$ can correspond to any point in M. In this setting our goal is to make the parts of the surface f' defined on $\tilde{M}_1$ and $\tilde{M}_2$ as close as possible to f_1 and f_2. This means that we would like to minimize the difference $f' - f_i \circ p_i$ on M_i, $i = 1, 2$ (Figure 9).

To minimize this difference we need at least the ability to evaluate the expression; even if we consider only vertices v of $\tilde{M}$, evaluating the expression requires computing $f_i(p_i(v))$ for any vertex. As there are no restrictions on the location of $p_i(v)$ in M_i, direct evaluation is required. Furthermore, we observe that the surfaces resulting from boolean operations on smooth surfaces are likely to have creases; after two operations corners are likely to appear. Difference operations are likely to produce concave corners (Figure 10).

This means that it is essential to be able to evaluate surfaces with all the features described in this paper.

6 Conclusions

The algorithms for evaluating peicewise smooth surfaces presented in this paper are simpler than those obtained for similar schemes using the standard approach based entirely on eigenanalysis. Remarkably, the fact that the rules of the schemes have two parameters does not have much impact on complexity. At the same time, clearly the algorithms are still quite complex and require substantial implementation effort. We have also applied similar ideas to implement evaluation of piecewice-smooth Catmull-Clark surfaces [4], which will be described in a separate report. One notable omission in this paper is the case of dart vertices, which have a single incident crease edge. The evaluation process for

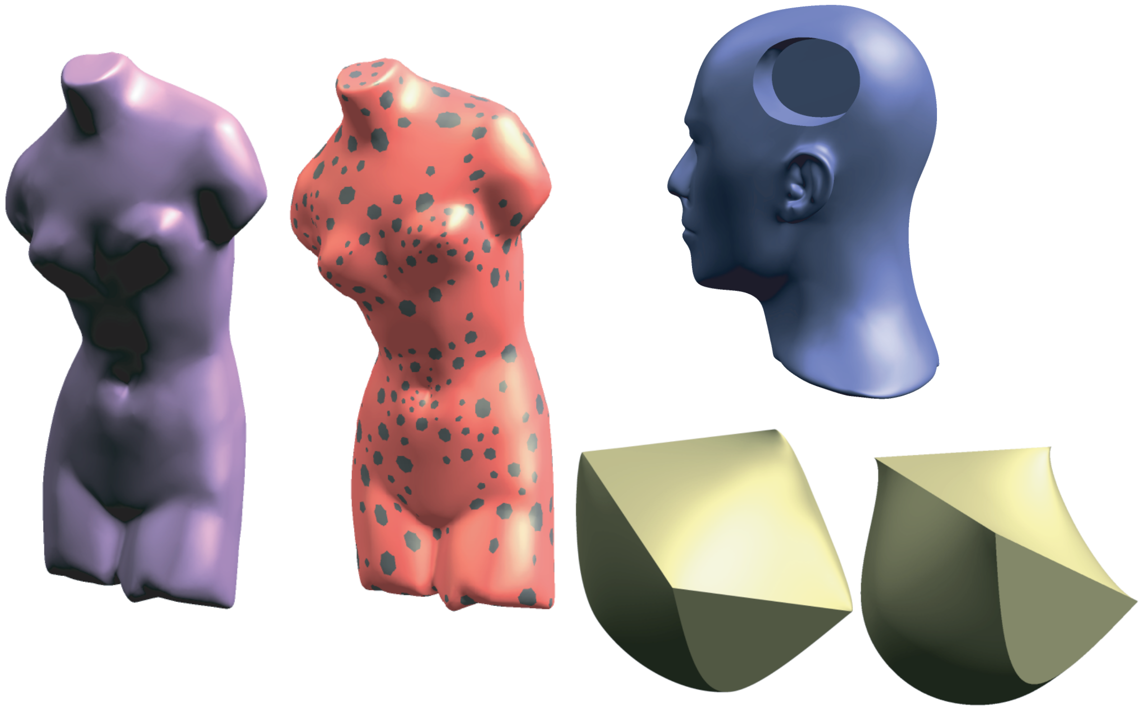

Figure 10: The surfaces shown here were obtained by evaluating the surface at evenly spaced locations, rather than by subdivision. For the Venus model, gray areas indicate neighborhoods of subdivision vertices where computation of the powers of the subdivision matrix was required. The surfaces on the right were obtained as a result of boolean operations; evaluation was used at the fitting stage. Note creases and corners on these surfaces.

dart vertices is similar to the boundary case, and was omitted at the time of writing for technical reasons. Our implementation of the evalation algorithms will include the dart case in the future. The code implementing the algorithms described in the paper is available at the Web site `http://www.mrl.nyu.edu/software/`.

A Left and right basis vectors used in evaluation algorithms

In this appendix we specify complete sets of basis vectors used in our evaluation algorithm. While we made our best to ensure that the numbers given here are correct, we recommend, if at all possible, using the code available from the authors, which was extensively tested for correctness.

A.1 Corner and smooth boundary rules.

Let $\theta_k = \pi/k$. We assume that $k > 1$; the subdivision matrix and eigenbasis for $k = 1$ are described separately. The basis vectors, both left and right, can be separated into two groups: six *irregular* basis vectors, which are eigenvectors, and $3k-3$ basis vectors that have a general form $c_0 = w_0$, $p_0 = w_1$, $p_k = w_2$, $p_i = a\sin(ij\theta_k)$, $q_i = b\sin((i+1/2)j\theta_k)$, $r_i = c\sin(ij\theta_k)$, $j = 1\dots k-1$. Five of the irregular basis vectors are obtained by extension of the eigenvectors for one-dimensional subdivision on the boundary. We call the basis vectors in the second *sine vectors*, as they can be computed using the a variation of the Discrete Sine Transform. Each such eigenvector is characterized by six constants: w_0, w_1, w_2, a, b, c.

Below, when defining the basis vectors, we will define all components only for six irregular eigenvectors. For the sine basis vectors we define only the constants.

The smooth and corner subdivision matrix share many basis vectors, we define them simultaneously. For all p_i and r_i, $i = 0\dots k$ unless otherwise stated. For q_i, $i = 0\dots k-1$.

Define

$$\gamma(\theta) = \frac{1}{2} - \frac{1}{4}\cos(\theta)$$

,

$$\lambda_j(\theta) = \frac{1}{2} - \frac{1}{4}(\cos(\theta) - \cos(j\theta_k))$$

.

For $\lambda \neq 1/8, 1/16$, define

$$r(\lambda,\gamma) = \frac{1}{16\,\lambda - 1}\left(8\left(1 + \frac{3}{8\,\lambda - 1}\right)(\lambda - \gamma) + \frac{6}{8\,\lambda - 1} + 10\right)$$

$$c(\lambda,\gamma) = \frac{1}{16\,\lambda - 1}\left(-\left(1 + \frac{3}{8\,\lambda - 1}\right)(6 - 8\,\gamma) + \frac{2}{8\,\lambda - 1} + 1\right)$$

Irregular right eigenvectors.

- Eigenvalue 1: $c_0 = 1$, $p_i = 1$, $q_i = 1$, $r_i = 1$.
- Eigenvalue $1/2$, first eigenvector (for smooth boundary rule, set $\theta = \pi/k$). $c_0 = 0$,

$$p_i = \frac{\sin((k/2 - i)\theta)}{\sin(k\theta/2)}$$

$$q_i = \frac{\sin((k/2 - i)\theta) + \sin((k/2 - (i+1))\theta)}{\sin(k\theta/2)}$$

$$r_i = r\left(\frac{1}{2}, \gamma(\theta)\right)\frac{\sin((k/2 - i)\theta)}{\sin(k\theta/2)}$$

  $r_0 = -r_k = 2$.
- Smooth boundary rule, eigenvalue $1/4$. Let $\varphi = \arccos(\cos\theta_k - 1)$, $C = (1 + \cos\theta_k)/(4 - 2\cos\theta_k)$. Then $c_0 = -1/2$,

$$p_i = (1 - C)\frac{\cos(i - k/2)\varphi}{\cos(\varphi k/2)} + C$$

$$q_i = 3\left(2C + (1 - C)\frac{\cos(i - k/2)\varphi + \cos(i + 1 - k/2)\varphi}{\cos(\varphi k/2)}\right) - \frac{1}{2}$$

$$\begin{aligned} r_i \quad = \quad & r\left(\frac{1}{4}, \gamma(\theta_k)\right)\left((1 - C)\frac{\cos(i - k/2)\varphi}{\cos(k\varphi/2)} + C\right) \\ & -\frac{1}{2}c\left(\frac{1}{4}, \gamma(\theta_k)\right), \quad i = 1 \ldots k - 1 \end{aligned}$$

  The remaining two components are $r_0 = r_k = 11/2$.
- Corner rule, eigenvalue $1/2$ (second eigenvector): $c_0 = 0$,

$$p_i = \frac{\cos((k/2 - i)\theta)}{\cos(k\theta/2)}$$

$$q_i = \frac{\cos((k/2 - i)\theta) + \cos((k/2 - (i+1))\theta)}{\cos(k\theta/2)}$$

$$r_i = r\left(\frac{1}{2}, \gamma(\theta)\right)\frac{\cos((k/2 - i)\theta)}{\cos(k\theta/2)}$$

  $r_0 = r_k = 2$. Note that we have to assume $\theta \neq \pi/k$, which is required for smoothness at a corner vertex.
- Two remaining eigenvectors obtained by extension from the boundary: the first one: $c_0 =$, $p_i = 0$, $q_i = 0$, $r_0 = 1$, $r_i = 0$, for $i > 0$; the second one: $c_0 =$, $p_i = 0$, $q_i = 0$, $r_k = 1$, $r_i = 0$, for $i < k$.
- Last irregular eigenvector: $c_0 = 0$, $p_i = 0$, $q_i = (-1)^i$, $r_i = 0$.

Sine right basis vectors. For all sine right basis vectors, $w_0 = w_1 = w_2 = 0$.

- First series, correspond to eigenvalues $\lambda_j(\theta)$, $j = 1 \dots k-1$ (for smooth boundary rules, set $\theta = \pi/k$):
$$a^j = 1, \quad b^j = c^j = 0.$$
- Second series, correspond to eigenvalue $1/8$, $k-1$ vectors, for $j = 1 \dots k-1$:
$$a^j = 0, \quad b^j = \frac{3}{4}\cos\frac{j\theta_k}{2}, \quad c^j = \frac{3}{2}\cos^2\frac{j\theta_k}{2}.$$
.
- Third series, correspond to eigenvalue $1/16$, $k-1$ vectors, for $j = 1 \dots k-1$:
$$a^j = b^j = 0, \quad c^j = \frac{1}{2} - \frac{5}{4}\cos^2\frac{j\theta_k}{2}.$$

Left irregular basis vectors. For all left basis vectors, excluding the last one, $p_i = 0$ for $i = 1 \dots k-1$, $q_i = 0$, $r_i = 0$ for $i = 1 \dots k-1$. Thus, we need to define only c_0, p_0, p_k, r_0 and r_k. The first five left eigenvectors coincide with the boundary left eigenvectors.

- Left eigenvector of eigenvalue 1 (corner): $c_0 = 1$, $p_0 = p_k = r_0 = r_k = 0$.
- Left eigenvector of eigenvalue 1 (smooth boundary): $c_0 = 2/3$, $p_0 = p_k = 1/6$, $r_0 = r_k = 0$.
- Left eigenvector of eigenvalue $1/2$ for both smooth boundary and corner rules: $c_0 = 0$, $p_0 = -p_k = 1/2$, $r_0 = r_k = 0$.
- Second left eigenvector of eigenvalue $1/2$ (corner): $c_0 = -1$, $p_0 = p_k = 1/2$, $r_0 = r_k = 0$.
- Left eigenvector of eigenvalue $1/4$ (smooth boundary): $c_0 = -2/3$, $p_0 = p_k = 1/3$, $r_0 = r_k = 0$.
- Left eigenvectors of eigenvalue $1/8$ (smooth boundary). First: $c_0 = 3$, $p_0 = -3$, $p_k = -1$, $r_0 = 1$, $r_k = 0$, Second: $c_0 = 3$, $p_0 = -1$, $p_k = -3$, $r_0 = 0$, $r_k = 1$,
- Left eigenvectors of eigenvalue $1/8$ (corner). First: $c_0 = 1$, $p_0 = -2$, $p_k = 0$, $r_0 = 1$, $r_k = 0$, Second: $c_0 = 1$, $p_0 = 0$, $p_k = -2$, $r_0 = 0$, $r_k = 1$.
- Last irregular eigenvector, eigenvalue $1/8$ (smooth boundary). If k is odd, $c_0 = 3/k$, $p_0 = p_k = -2/k$, $p_i = 0$ for $i = 1 \dots k-1$, $q_i = (-1)^i/k$, $r_i = 0$ for $i = 0 \dots k$. Otherwise, $c_0 = 0$, $p_0 = -p_k = -1/k$, $p_i = 0$ for $i = 1 \dots k-1$, $q_i = (-1)^i/k$, $r_i = 0$ for $i = 0 \dots k$.
- Last irregular eigenvector, eigenvalue $1/8$ (corner). If k is odd, $c_0 = 1/k$, $p_0 = p_k = -1/k$, $p_i = 0$ for $i = 1 \dots k-1$, $q_i = (-1)^i/k$, $r_i = 0$ for $i = 0 \dots k$. Otherwise, $c_0 = 0$, $p_0 = -p_k = -1/k$, $p_i = 0$ for $i = 1 \dots k-1$, $q_i = (-1)^i/k$, $r_i = 0$ for $i = 0 \dots k$.

Sine left basis vectors. Define

$$\begin{aligned}
\sigma_0^j(\theta) &= \frac{\sin j\theta_k}{\cos\theta - \cos j\theta_k} \\
\sigma_1^j(\theta) &= \frac{4\cos^2(\theta/2)\sin(j\theta_k/2)}{\cos\theta - \cos j\theta_k}
\end{aligned}$$

All sine basis vectors are defined by the constants w_0, w_1, w_2, a, b, c. The constants a, b and c are computed in the same way for both smooth boundary and corner rules, w_0, w_1, and w_2 are computed in slightly different ways. We consider the two groups of constants separately. To simplify expressions, we define left basis vectors as $c_0 = (2/k)w_0$, $p_0 = (2/k)w_1$, $p_k = (2/k)w_2$, etc., i.e. factor out $2/k$.

- For the first series of left basis vectors, $j = 1 \dots k-1$, $a^j = 1$, $b^j = 0$, $c^j = 0$.
- For the second series of left basis vectors (eigenvalue $1/8$), $j = 1 \dots k-1$,

$$a^j = 0, \quad b^j = \frac{4}{3}\cos\frac{j\theta_k}{2}, \quad c^j = 0.$$

- For the third series of left basis vectors, (eigenvalue $1/16$), $j = 1 \dots k-1$,

$$a^j = 0, \quad b^j = \frac{-2\cos(\theta_k/2)}{1/2 - (5/4)\cos^2(\theta_k/2)}, \quad c^j = \frac{1}{1/2 - (5/4)\cos^2(\theta_k/2)}.$$

We specify w_0, w_1 and w_2 separately for the smooth boundary and corner rules; they also differ for odd and even vector numbers. First, define two constants:

$$\begin{aligned}
A^{odd} &= a^j\sigma_0^j(0) + b^j\sigma_1^j(0)/2 + c^j\sigma_0^j(0) \\
A^{even} &= a^j\sigma_0^j(\theta) + b^j\sigma_1^j(\theta) + c^j r(1/2, \gamma(\theta))\sigma_0^j(\theta)
\end{aligned}$$

Now consider the four possible cases:

- Both rules, even j: $w_0 = 0$, $w_1 = -w_2 = -A^{even}/2$.
- Smooth boundary rule, odd j. let φ and C be as in the definition of the right eigenvector of eigenvalue $1/4$. Let

$$\begin{aligned}
B = \quad & a^j\left((1-C)\sigma_0^j(\varphi) + c\sigma_0^j(0)\right) \\
& + b^j\left((3c - 1/4)\sigma_1^j(0) + 3(1-C)\sigma_1^j(\varphi)\right) \\
& + c^j\Big(r(1/4, \gamma(\theta))(1-C)\sigma_0^j(\varphi) \\
& + (Cr(1/4,\gamma(\theta)) - c(1/4,\gamma(\theta))/2)\sigma_0^j(0)\Big)
\end{aligned}$$

  Then $w_0 = (2/3)(B - A^{odd})$, $w_1 = w_2 = -A^{odd}/6 - B/3$.
- Corner rule, odd j. In this case, $w1 = -A^{even}/2$, $w_0 = A^{even} - A^{odd}$.

Case $k = 1$**.** For $k = 1$ corners can be only convex. For the corner rules the subdivision matrix and matrices of left and right eigenvectors are:

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 3/8 & 1/8 & 1/8 & 3/8 & 0 & 0 \\ 1/8 & 3/4 & 0 & 0 & 1/8 & 0 \\ 1/8 & 0 & 3/4 & 0 & 0 & 1/8 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 & 1 \\ 1 & -2 & 0 & 0 & 1 & 0 \end{pmatrix}$$

For smooth boundary rules

$$S = \begin{pmatrix} 3/4 & 1/8 & 1/8 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 3/8 & 1/8 & 1/8 & 3/8 & 0 & 0 \\ 1/8 & 3/4 & 0 & 0 & 1/8 & 0 \\ 1/8 & 0 & 3/4 & 0 & 0 & 1/8 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 & -1/2 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1/2 & 1 & 0 & 0 \\ 1 & -2 & 11/2 & 0 & 0 & 1 \\ 1 & 2 & 11/2 & 0 & 1 & 0 \end{pmatrix}, \quad L = \frac{1}{6}\begin{pmatrix} 4 & 1 & 1 & 0 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 & 0 \\ -4 & 2 & 2 & 0 & 0 & 0 \\ -6 & 0 & 0 & 6 & 0 & 0 \\ 18 & -6 & -18 & 0 & 0 & 6 \\ 18 & -18 & -6 & 0 & 6 & 0 \end{pmatrix}$$

A.2 Interior rules

There are four vectors for interior rules with irregular expressions and three families with $k-1$ basis vectors each. We assume that $k \geq 3$ in this case.

Irregular right vectors. In all cases, i varies from 0 to $k-1$.

- Eigenvalue 1: basis vector v_0 has all components equal to 1.
- Eigenvalue $\lambda_0 = 1/4 + \gamma - k\beta$, basis vector v_0^λ: $c_0 = k\beta$, $p_i = \gamma - 3/4$, $q_i = r_i = 0$.
- Eigenvalue $1/8$: basis vector v_0^q: $c_0 = 0$, $p_i = 0$, $q_i = r_i/2 = k\beta/8 - (3/4)(3/4-\gamma)$.
- Eigenvalue $1/16$: basis vector v_0^r: $c_0 = 0$, $p_i = q_i = 0$, $r_i = -3k\beta/16 + (3/4)(3/4-\gamma)$.

Regular right vectors. Here j varies from 1 to $k-1$.

- Eigenvalues $\lambda_j = 3/8 + (1/4)\cos(2\pi j/k)$, basis vectors v_j^λ: $c_0 = q_i = r_i = 0$, $p_i = D^1_{ij}$ (see Section 4 for definition of D^1).
- Eigenvalue $1/8$, basis vectors v_j^q for $j \neq k/2$: $c_0 = p_i = 0$, $q_i = (3c/4)D^2_{ij}$, $r_i = (3c^2/2)D^1_{ij}$, with $c = \cos\left(((j-1)\mathrm{mod}\lfloor k/2\rfloor + 1)\pi/k\right)$. If $j = k/2$, $c_0 = p_i = r - i = 0$, $q_i = (-1)^i$.
- Eigenvalue $1/16$, basis vectors v_j^r: $c_0 = p_i = q_i = 0$, $r_i = (1/2 - 5c^2/4)D^1_{ij}$,

Irregular left vectors. Define $A = k\beta$, $C = 3/4 - \gamma$, $d = 1/(C+A)$, $d' = d/(6C-A)$, $d'' = d/(4C-A)$.

- Eigenvalue 1: basis vector l_0: $c_0 = Cd$, $p_i = dA/k$, $q_i = r_i = 0$.
- Eigenvalue $\lambda_0 = 1/4 + \gamma - k\beta$, basis vector l_0^λ: $c_0 = d$, $p_i = -d/k$, $q_i = r_i = 0$.
- Eigenvalue $1/8$: basis vector l_0^q: $c_0 = 8d'C$, $p_i = 8d'A/k$, $q_i = -8d'(C+A)/k$, $r_i = 0$.
- Eigenvalue $1/16$: basis vector l_0^r: $c_0 = 16Cd''/3$, $p_i = 16Ad''/(3k)$, $q_i = -32d''(C+A)/(3k)$, $r_i = 16d''(C+A)/(3k)$.

Regular left vectors. Let c be defined as for regular right vectors above. The formulas for regular left basis vectors do not apply to $j = k/2$; these need to be treated separately.

- Eigenvalues $\lambda_j = 3/8 + (1/4)\cos(2\pi j/k)$: basis vectors l_j^λ coincide with $(2/k)v_j$.
- Eigenvalue $1/8$, basis vectors l_j^q: $c_0 = p_i = r_i = 0$, $q_i = (8/(3ck))D^2_{ij}$.
- Eigenvalue $1/16$, basis vectors l_j^r: $c_0 = p_i = 0$, $q_i = (2/k)(8c/(5c^2-2))D^2_{ij}$, $r_i = (2/k)(-4/(5c^2-2))D^1_{ij}$.

For $j = k/2$ (possible only for even k), The three eigenvectors l_j^λ, l_j^q, and l_j^r are given by $p_i = (\sqrt{2}/k)(-1)^i$, $q_i = r_i = 0$; $p_i = r_i = 0$, $q_i = (1/k)(-1)^i$; $p_i = q_i = 0$, $r_i = (2\sqrt{2}/k)(-1)^i$. $c_0 = 0$ in all cases.

B Evaluation of quartic box splines

The final evaluation step described in Section 3 is performed using a polynomial basis. We follow the presentation in [12]. At this point, we have a vector V of 12 control points defining the surface on a single triangle on a sufficiently high subdivision level, as well as barycentric coordinates (u, w) in that triangle. We assume that the control points are arranged as shown in Figure 11.

The formula we use is $f(u,w,t) = Bern(u,w,t)QV$, where $t = 1 - u - w$. The evaluation uses the vector of Bernstein monomials

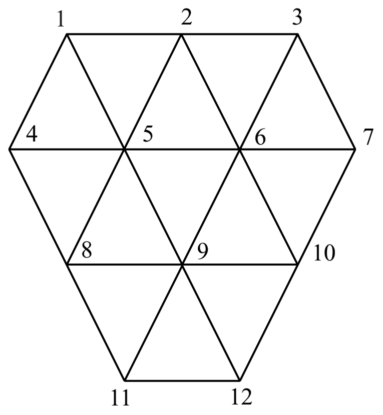

Figure 11: Control points for a single triangle. Barycentric coordinates (u, w, t) correspond to vertices 5, 9 and 6 respectively.

$$\begin{aligned} Bern(u,w,t) = \big[&u^4, 4u^3w, 4u^3t, 6u^2w^2, 12u^2ut, 6u^2t^2, 4uw^3, \\ &12uw^2t, 12uwt^2, 4ut^3, w^4, 4w^3t, 6w^2t^2, 4wt^2, 4wt^3, t^4\big] \end{aligned}$$

and the matrix

$$Q = \frac{1}{24}\begin{pmatrix} 2 & 2 & 0 & 2 & 12 & 2 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 12 & 3 & 0 & 3 & 4 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 12 & 4 & 0 & 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 4 & 0 & 4 & 8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 10 & 6 & 0 & 1 & 6 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 8 & 8 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 3 & 0 & 3 & 12 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 6 & 6 & 0 & 1 & 10 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 6 & 10 & 0 & 0 & 6 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & 4 & 12 & 0 & 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 12 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 3 & 4 & 0 & 1 & 12 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 4 & 8 & 0 & 0 & 8 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 3 & 12 & 1 & 0 & 4 & 3 & 0 & 0 \\ 0 & 2 & 2 & 0 & 2 & 12 & 2 & 0 & 2 & 2 & 0 & 0 \end{pmatrix}$$

References

[1] Alias|wavefront. Maya 3.0, 2000.

[2] V. I. Arnold. *Singularity Theory*. Cambridge University Press, Cambridge, 1981.

[3] H. Biermann, D. Kristjiansson, and D. Zorin. Approximate boolean operations on free-form solids. To appear in SIGGRAPH 2001 Proceedings.

[4] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. *Proceedings of SIGGRAPH 2000*, July 2000.

[5] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.

[6] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. *Proceedings of SIGGRAPH 98*, pages 85–94, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida.

[7] D. Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10(6):356–360, 1978.

[8] A. Edelman, E. Elmroth, and B. K. gström. A geometric approach to perturbation theory of matrices and matrix pencils. part i: Versal deformation. *SIAM Journal on Matrix Analysis and Applications*, 20(3):667–699, 1999.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 19–26, August 1993.

[10] A. H. Nasri. Polyhedral subdivision methods for free-form surfaces. *ACM Trans. Gr.*, 6(1):29–73, January 1987.

[11] A. H. Nasri. Boundary corner control in recursive subdivision surfaces. *Computer Aided Design*, 23(6):405–410, 1991.

[12] J. E. Schweitzer. *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, Seattle, 1996.

[13] J. Stam. Exact evaluation of Loop subdivision surfaces. Unpublished note.

[14] J. Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Proceedings of SIGGRAPH 98*, pages 395–404, July 1998. ISBN 0-89791-999-8. Held in Orlando, Florida.

[15] D. Zorin. A method for analysis of C^1-continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis*, 37(5):1677–1708, 2000.

[16] D. Zorin, P. Schröder, T. DeRose, L. Kobbelt, A. Levin, and W. Sweldens. Subdivision for modeling and animation. SIGGRAPH 2000 Course Notes, ACM SIGGRAPH.