Convex Triangular Subdivision Surfaces
with
Bounded Curvature

Charles Loop

June 21, 2000

Technical Report
MSR-TR-2000-71

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

The edge masks for Loop's triangular subdivision surface algorithm are modified resulting in surfaces with bounded curvature and the convex hull property. The new edge masks are derived from a polynomial *mask equation* whose Chebyshev expansion coefficients are closely related to the eigenvalues of the corresponding subdivision matrix. The mask equation is found to satisfy a set of smoothness constraints on these eigenvalues.

1 Introduction

Subdivision surfaces have become a popular geometric modeling primitive for the representation of smooth free-form shapes. Subdivision surfaces easily represent arbitrary topology surfaces and possess a simple, recursive, geometric construction algorithmic. This algorithm takes as input an initial *control mesh* and subdivides the faces and edges to form a new control mesh with more faces, edges, and vertices than the original while leaving the topology unchanged. By applying this procedure recursively, a smooth limit surface is obtained that is topologically equivalent to the original control mesh. This process is illustrated in Figure 1.

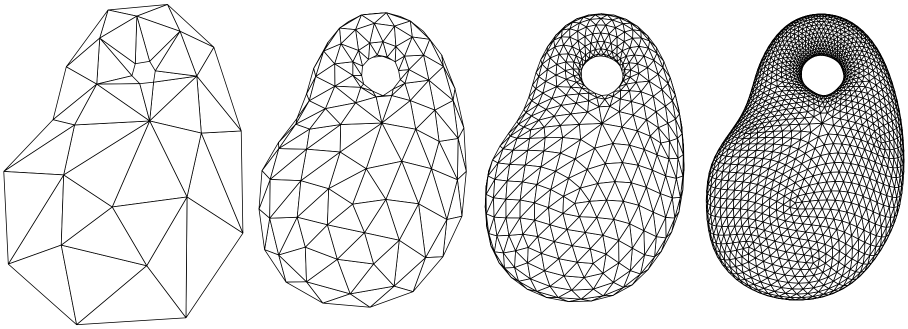

Figure 1: An initial control mesh (far left) and the control meshes obtained by three iterations of Loop's subdivision surface algorithm.

The first subdivision surfaces were developed independently by Doo&Sabin [3] and Catmull&Clark [2]. These methods generalize the subdivision rules of biquadratic and bicubic tensor product B-Spline surfaces respectively. Loop[6] derived a triangular subdivision surface by generalizing the subdivision rules for quartic box splines. These surfaces inherit the smoothness properties of their underlying polynomial splines at all but a finite number of extraordinary vertices. An control mesh vertex is *extraordinary* if it has irregular valance, otherwise it is *ordinary*. Ordinary vertices have valance 4 for rectangle based surfaces, or 6 for triangle based surfaces. The smoothness properties of a subdivision surface at points corresponding to extraordinary vertices are derived from the spectral properties of the subdivision operator[3, 6, 1].

Extraordinary vertices are isolated by expanding layers of *spline rings* as the subdivision level increases. Each spline ring is a collection of polynomial patches that parameterize the surface surrounding an extraordinary vertex. In studying the smoothness of subdivision surfaces at an extraordinary vertex, only a single generic spline ring of valance n need be considered. The subdivision operator takes one spline ring onto the next and is linear. A matrix $\mathbf{A}$ that encodes this operator is known as a *subdivision matrix.* Since subdivision is affine invariant, $\mathbf{A}$ will have a single dominant eigenvalue 1 with corresponding eigenvector $[1, 1, \ldots, 1]$. For symmetric subdivision schemes, e.g. the methods of Doo&Sabin,Catmull&Clark, and Loop, $\mathbf{A}$ will also have a pair of subdominant eigenvalues λ.

Reif[10] derived necessary and sufficient conditions for tangent plane (G^1) continuity by showing that the *characteristic map* defined by the pair of subdominant eigenvectors must be regular and injective. The characteristic map takes individual patch domains into a common xy-plane whose origin corresponds to the extraordinary vertex. Higher order smoothness properties are studied in terms of Taylor expansions over this plane. Necessary conditions for G^k continuity have been given by Prautzsch&Reif[7]. For curvature continuity (G^2), in addition to G^1 continuity, the following must also hold:

- $\mathbf{A}$ has a subsubdominant eigenvalue μ, where $|\mu| = \lambda^2$.
- If $\mathbf{A}v = \mu v$, then the surface defined by $v \in span(x^2, xy, y^2)$.

Prautzsch&Reif use a degree argument to show that no modification to the Catmull&Clark or Loop schemes can be used to construct G^2 continuous surfaces with non-zero curvature at points corresponding to extraordinary vertices.

Despite the apparent failure to realize true curvature continuity, several authors[3, 6, 11] have noted the importance of the ratio $\lambda^2/|\mu|$ in controlling the divergence of the curvature at extraordinary vertices:

1. If $\lambda^2/|\mu| > 1$, then the curvature is unbounded
2. If $\lambda^2/|\mu| = 1$, then the curvature is bounded
3. If $\lambda^2/|\mu| < 1$, then the curvature is zero

Prautzsch&Umlauf[8, 9] exploit property (3) above to create modified versions of the Catmull&Clark and Loop algorithms that result in local flat spots corresponding to extraordinary vertices. Although these schemes are technically G^2, forcing curvatures to zero may not necessarily lead to pleasing shapes. Moreover, these modifications introduce negative weights into the subdivision masks, meaning that the limit surface no longer lies in the convex hull of its control mesh, potentially compromising the utility of this approach.

Instead of forcing curvatures to zero, we take advantage of property (2) above to develop a modified version of Loop's algorithm that has bounded curvature. Though not strictly G^2, we believe this approach results in improved shapes. Our approach also has the advantage of avoiding negatives weights to preserve the convex hull property.

This paper is organized as follows. In Section 2 we review Loop's algorithm and outline the new modified algorithm. In Section 3 we examine subdivision matrix structure and develop a parameterized subdivision matrix. In Section 4, constraints are placed on this parameterization sufficient for bounded curvature. In Section 5 we present a polynomial mask equation that satisfies these constraints; the mask equation is evaluated at regular intervals to obtain the new edge masks. In Sections 6 and 7 we present results and conclusions.

2 Loop's algorithm

Loop's algorithm subdivides a triangular control mesh by quadrisecting each triangular face as shown in Figure 2. The vertices of the subdivided mesh are found as affine combinations of the vertices of the original control mesh. The weights associated with these affine combinations are called *masks.* The new vertices fall into two classes; those corresponding to vertices of the original control mesh, and those corresponding to edges of the original control mesh. The associated masks are referred to as *vertex masks* and *edge masks* respectively. Figure 3 depicts the two types of masks used in Loop's algorithm.

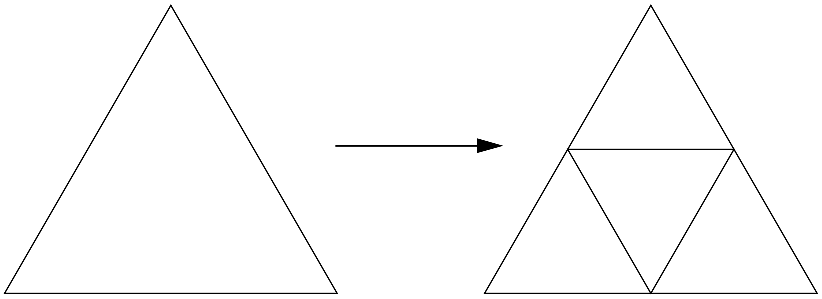

Figure 2: Quadrisection of a single triangular face.

The vertex mask assigns weights to an original mesh vertex v and the set of vertices that share an edge with v. The weights for the vertex mask are

$$\alpha = (\tfrac{3}{8} + \tfrac{1}{4}\cos\tfrac{2\pi}{n})^2 + \tfrac{3}{8} \quad \text{and} \quad \beta = \tfrac{1}{n}\,(1-\alpha),$$

where n is the valance of the vertex. The logic behind this choice of weights is given in Section 4. The edge mask assigns weights to the vertices of the two triangles that share the edge, and are independent of vertex valance.

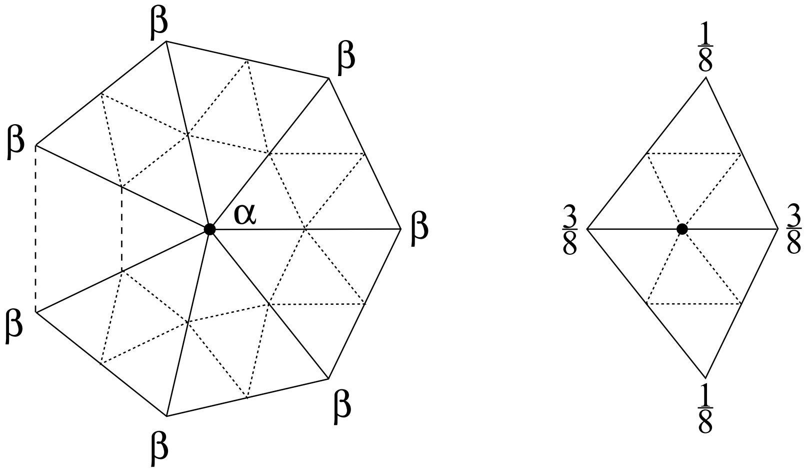

Figure 3: Vertex mask and edge mask used in Loop's algorithm.

Loop's algorithm is a generalization of binary subdivision of C^2 trilaterally symmetric quartic box splines[12]. If the initial control mesh is a regular triangulation (all vertices are valance 6), then the limit surface of Loop's algorithm is a quartic box spline with continuous curvatures. Methods for dealing with boundary curves and various sharp surface features can be found in[5].

2.1 A modified triangular subdivision surface algorithm

We propose a modified version of Loop's triangular subdivision surface algorithm where the edge mask has larger support. The proposed new edge mask is depicted in Figure 4. The mask weights $1-\lambda_0, \gamma_0, \dots, \gamma_{n-1}$ will form a partition of unity, so $\lambda_0 = \sum \gamma_i$.

We point out that such a modified edge mask will be asymmetric with respect to the two incident vertices. That is, the position of a new mesh vertex associated with an edge will be different depending on which of the two incident vertices is used to determine the new edge mask. This ambiguity is handled by examining the valance of the two incident vertices.

- *If both vertices are ordinary, use the original edge mask.*
- *If one vertex is extraordinary, use the edge mask associated with that vertex.*

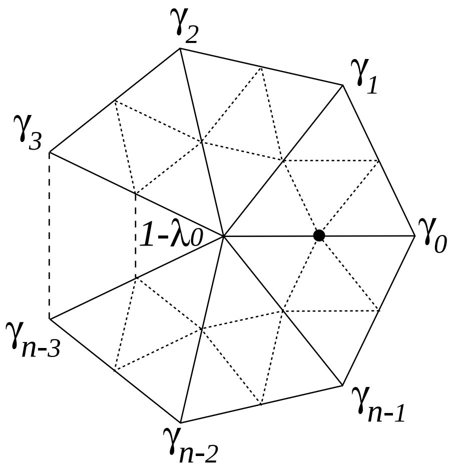

Figure 4: Modified edge mask for the new algorithm.

- *If both vertices are extraordinary, use both each masks and average the result.*

Note that the last case above will only occur during the first level of subdivision, since all extraordinary vertices are surrounded by ordinary vertices after one level. This means that only edges that are incident on an extraordinary vertex will use a modified edge mask. A new edge mask is determined by the valance n of an incident extraordinary vertex; rotated versions of a single edge mask are used to compute all the new edge points adjacent to the extraordinary vertex.

3 Subdivision Matrix Structure

The subdivision matrix, used to analyze the smoothness properties of a subdivision surface at an extraordinary vertex, transforms a spline ring at subdivision level i to a spline ring at level $i+1$. In the case of Loop's algorithm, the spline ring coefficients correspond to three rings of control mesh vertices surrounding an extraordinary vertex, a so-called *3-ring*. In fact, all of the large eigenvalues (those of interest here) of a subdivision matrix can be shown to come from the 1-ring[1]. For this reason, modifications of the type to be considered here need only consider the 1-ring subdivision matrix and its eigen structure.

[1] A 1-ring is a vertex together with all of it's edge sharing neighbor vertices.

3.1 Subdivision matrix parameterization

We begin by parameterizing the 1-ring matrix for a symmetric triangular subdivision surface algorithm. Let n be the valance of a generic extraordinary vertex and let $\alpha, \lambda_0, \lambda_1, \ldots, \lambda_{n-1}$ be a set of scalar values. We define

$$\mathbf{Q} = \left[\begin{array}{c|cccc} \alpha & 1-\alpha & 0 & \cdots & 0 \\ \hline 1-\lambda_0 & \lambda_0 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 & 0 & \cdots \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & \ldots & 0 & \lambda_{n-1} \end{array} \right].$$

Since the subdivision scheme is symmetric, the 1-ring subdivision matrix $\mathbf{S}$ is related to $\mathbf{Q}$ by the following

$$\mathbf{S} = \mathbf{F}\mathbf{Q}\mathbf{F}^{-1},$$

where

$$\mathbf{F} = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \hat{\mathbf{F}} & \\ 0 & & & \end{array} \right] \quad \text{and} \quad \mathbf{F}^{-1} = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \hat{\mathbf{F}}^{-1} & \\ 0 & & & \end{array} \right]$$

are $(n+1) \times (n+1)$ block diagonal matrices whose lower blocks are the $n \times n$ discrete Fourier Transform matrix and corresponding inverse defined

$$\hat{\mathbf{F}}_{j,k} = e^{\frac{2\pi \mathrm{i} jk}{n}}, \quad \hat{\mathbf{F}}^{-1}_{j,k} = \frac{1}{n} e^{\frac{-2\pi \mathrm{i} jk}{n}}, \quad j,k \in 0, \ldots, n-1.$$

The eigen structure of $\mathbf{S}$ is the decomposition

$$\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$$

where Λ is a diagonal matrix of *eigenvalues*; $\mathbf{V}$ and its inverse $\mathbf{V}^{-1}$ are square matrices whose rows (resp. columns) are right (resp. left) *eigenvectors* of $\mathbf{S}$. These matrices are related to our parameterization as follows:

$$\Lambda = \mathbf{G}^{-1}\mathbf{Q}\mathbf{G}, \quad \mathbf{V} = \mathbf{F}\mathbf{G}, \quad \text{and} \quad \mathbf{V}^{-1} = \mathbf{G}^{-1}\mathbf{F}^{-1},$$

where

$$\mathbf{G} = \left[\begin{array}{c|c} \hat{\mathbf{G}} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{n-1} \end{array} \right] \quad \text{and} \quad \mathbf{G}^{-1} = \left[\begin{array}{c|c} \hat{\mathbf{G}}^{-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_{n-1} \end{array} \right]$$

are block diagonal matrices whose lower block $\mathbf{I}_{n-1}$ is the $(n-1) \times (n-1)$ identity matrix, and whose upper 2×2 blocks are defined

$$\hat{\mathbf{G}} = \left[\begin{array}{cc} 1 & \frac{1-\alpha}{\lambda_0 - 1} \\ 1 & 1 \end{array} \right] \quad \text{and} \quad \hat{\mathbf{G}}^{-1} = \frac{1}{2-\alpha-\lambda_0} \left[\begin{array}{cc} 1-\lambda_0 & 1-\alpha \\ \lambda_0 - 1 & 1-\lambda_0 \end{array} \right].$$

The relationships among the various matrices are summarized as follows

$$\mathbf{S} = \mathbf{F}\mathbf{G}\Lambda\mathbf{G}^{-1}\mathbf{F}^{-1} = \mathbf{F}\mathbf{Q}\mathbf{F}^{-1} = \left[\begin{array}{c|ccc} \alpha & \frac{1-\alpha}{n} & \ldots & \frac{1-\alpha}{n} \\ \hline 1-\lambda_0 & & & \\ \vdots & & \hat{\mathbf{S}} & \\ 1-\lambda_0 & & & \end{array}\right], \tag{1}$$

and

$$\Lambda = \mathbf{G}^{-1}\mathbf{F}^{-1}\mathbf{S}\mathbf{F}\mathbf{G} = \mathbf{G}^{-1}\mathbf{Q}\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha+\lambda_0-1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 & \cdots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \ldots & & 0 & \lambda_{n-1} \end{bmatrix} \tag{2}$$

where

$$\hat{\mathbf{S}} = \hat{\mathbf{F}}\mathrm{diag}(\lambda)\hat{\mathbf{F}}^{-1} = \mathrm{circ}(\gamma), \tag{3}$$

and

$$\lambda = [\lambda_0, \ldots, \lambda_{n-1}], \quad \gamma = [\gamma_0, \ldots, \gamma_{n-1}].$$

The parameters $\lambda_0, \ldots, \lambda_{n-1}$ are the eigenvalues of $\hat{\mathbf{S}}$, the lower right $n \times n$ block of $\mathbf{S}$ in Equation (1). Note that $\hat{\mathbf{S}}$ is a circulant matrix constructed from the edge mask weights $\gamma_0, \ldots, \gamma_{n-1}$. The parameter α is the central weight of the corresponding vertex mask; this value will differ (in general) from the original value in Loop's algorithm. Equation (2) shows the relationship between the proposed parameterization and the eigenvalues of $\mathbf{S}$.

4 Constraints on the Parameterization

The parameterization introduced in the last section characterizes a large space of possible subdivision matrices. We reduce the size of this space by placing constraints of the parameters $\alpha, \lambda_0, \ldots, \lambda_{n-1}$.

4.1 Real valued

We only consider symmetric subdivision schemes, so $\hat{\mathbf{S}}$ will be a circulant matrix consisting of real values $\gamma_0, \ldots, \gamma_{n-1}$, if the set $\lambda_0, \ldots, \lambda_{n-1}$ is a discrete *even* function. In other words, the $\lambda_i, i > 0$, must be matched pairwise:

$$\lambda_1 = \lambda_{n-1},\ \lambda_2 = \lambda_{n-2},\ \cdots,\ \lambda_{\lfloor \frac{n}{2} \rfloor} = \lambda_{\lceil \frac{n}{2} \rceil}. \tag{4}$$

Note that if n is even, $\lambda_{\frac{n}{2}}$ is unmatched. Assuming real values, Equation (3) can be rewritten

$$\hat{\mathbf{S}} = \mathrm{circ}(\gamma) \quad \text{where} \quad \gamma = \lambda\mathbf{C} \quad \text{and} \quad \mathbf{C}_{i,j} = \tfrac{1}{n}\cos\tfrac{2\pi ij}{n}. \tag{5}$$

4.2 Convergent

We guarantee that the subdivision process will converge to a well defined surface by requiring that all of the eigenvaules of **S** be less than or equal to 1 in absolute value. From Equation (2) the eigenvalues of **S** are

$$1, \alpha + \lambda_0 - 1, \lambda_1, \lambda_2, \ldots, \lambda_{n-1}.$$

The constraints for convergence are therefore

$$|\alpha + \lambda_0 - 1| \leq 1, \tag{6}$$

$$1 \geq |\lambda_i|, \quad i = 1, \ldots, n-1. \tag{7}$$

4.3 Continuous tangent plane

For the subdivision surface to converge to a common tangent plane at each extraordinary vertex, we require that there exist a pair of subdominant eigenvalues such that the characteristic map defined by the corresponding eigenvectors be regular and injective. The characteristic map for triangular subdivision schemes have been extensively studied[4, 13]. We leverage these results by constraining the eigenvalue pair λ_1 and λ_{n-1} to be consistent with Loop's algorithm. Prautzsch&Umlauf take the same approach and prove equivalence of the characteristic map[9]. The constraints for tangent plane continuity are therefore

$$\lambda_1 = \lambda_{n-1} = \tfrac{3}{8} + \tfrac{1}{4}\cos\tfrac{2\pi}{n}, \tag{8}$$

$$\lambda_1 \geq |\lambda_i|, \quad i = 2, \ldots, n-2. \tag{9}$$

Requiring that the characteristic map not deviate from Loop's algorithm is justified in practice since choosing values for λ_1 and λ_{n-1} other than above have lead to undesirable shape artifacts in resulting surfaces.

4.4 Bounded curvature

We ensure the subdivision surface has bounded curvature by requiring that

$$\alpha + \lambda_0 - 1 = \lambda_2 = \lambda_{n-2} = \lambda_1^2, \tag{10}$$

$$\lambda_2 \geq |\lambda_i|, \quad i = 3, \ldots, n-3, \tag{11}$$

where $n \geq 5$ (cases where $n \leq 5$ are handled separately). This will endow **S** with a subsubdominant eigenvalue of geometric and algebraic multiplicity 3. While this multiplicity of the subsubdominant eigenvalue is a stronger condition than required for bounded curvature, it is a necessary condition for non-degenerate curvature continuity (i.e., where the osculating paraboloid can span all quadratics).

Note that Equations (8) and (10) imply (6), therefore tangent plane continuity and bounded curvature imply convergence; the convergent constraints shall not be considered further.

4.5 Convexity

In addition to satisfying the constraints just outlined, we also want to ensure that $\mathbf{S}$ contain only non-negative values. This means that the resulting subdivision algorithm will construct new control mesh vertices as strictly convex combinations of existing control mesh vertices. This will guarantee that the limit surface lies in the convex hull of the control mesh.

From Equations (1) and (5), it follows that $\mathbf{S}$ will be non-negative if and only if

$$\gamma_i = [\lambda \mathbf{C}]_i \geq 0, \quad i = 0, \dots, n-1 \tag{12}$$

and

$$0 \leq \alpha \leq 1, \quad \text{and} \quad \lambda_0 \leq 1. \tag{13}$$

We combine (10) and (13) to get

$$\lambda_1^2 \leq \lambda_0 \leq 1. \tag{14}$$

Remark 1 *Equation (10) implies*

$$\alpha = 1 + \lambda_1^2 - \lambda_0.$$

Loop's algorithm obeys this relation and has $\lambda_0 = \frac{5}{8}$ for all n. This leads to Loop's original choice for α. Prautzsch&Umlauf[9] also (indirectly) choose $\lambda_0 = \frac{5}{8}$. Allowing λ_0 to differ from $\frac{5}{8}$ while satisfying (14) is essential in order to avoid negative mask weights. However, $\lambda_0 \approx \frac{5}{8}$ has been observed to minimize curvature distribution artifacts, and is therefore considered desirable.

4.6 Constraint summary

The constraints on on the parameters $\alpha, \lambda_0, \dots, \lambda_{n-1}$ are summarized as follows:

• real valued:	$\lambda_i = \lambda_{n-1}, \quad i = 1, \dots, \lfloor \frac{n}{2} \rfloor$
• continuous tangent plane:	$\lambda_1 = \lambda_{n-1} = \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n}$
• bounded curvature:	$\lambda_2 = \lambda_{n-2} = \lambda_1^2$ $\alpha = 1 + \lambda_2 - \lambda_0$ $\lambda_2 \geq \lvert\lambda_i\rvert, \quad i = 3, \dots, n-3$
• convexity:	$[\lambda \mathbf{C}]_i \geq 0 \quad i = 0, \dots, n-1$ $\lambda_2 \leq \lambda_0 \leq 1$

In the next section, we present our approach to satisfying these constraints simultaneously.

5 A Mask Equation

We want to compute the product $\gamma = \lambda\mathbf{C}$, subject to constraints on the λ's, so that γ is entirely non-negative. Our approach is to convert the mask weights from a discrete set to a continuous function that can be evaluated along a regular interval to recover the γ_i's.

Expanding Equation (12), we can write

$$\gamma_i = \tfrac{1}{n} \sum_{j=0}^{n-1} \lambda_j \cos \tfrac{2\pi ij}{n},$$

or as a continuous function

$$\gamma(\theta) = \tfrac{1}{n} \sum_{j=0}^{n-1} \lambda_j \cos(j\theta)$$

since $\gamma_i = \gamma(\frac{2\pi i}{n}), i = 0, \ldots, n-1$. We make the substitution $\theta = \arccos u$ and define

$$M_n(u) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} x_i T_i(u) \tag{15}$$

where $T_i(u)$ is the i^{th} Chebyshev polynomial; by definition, $T_k(\cos\theta) \equiv \cos(k\theta)$. The x_i's and λ_i's are related by

$$x_i = \tfrac{1}{n} \begin{cases} \lambda_0 & \text{if } i = 0 \\ \lambda_{\frac{n}{2}} & \text{if } i = \frac{n}{2} \text{ and } n \text{ is even} \\ 2\,\lambda_i & \text{otherwise} \end{cases} \tag{16}$$

Example 2 *Consider the regular case $n = 6$. Decomposition of the regular subdivision matrix leads to*

$$\lambda = \{\tfrac{5}{8}, \tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{8}, \tfrac{1}{4}, \tfrac{1}{2}\}.$$

From this

$$\begin{aligned} M_6(u) &= \tfrac{1}{6}\left[\tfrac{5}{8}T_0(u) + T_1(u) + \tfrac{1}{2}T_2(u) + \tfrac{1}{8}T_3(u)\right] \\ &= \tfrac{5}{48} + \tfrac{1}{6}u + \tfrac{1}{12}(2u^2-1) + \tfrac{1}{48}(4u^3 - 3u) \\ &= \tfrac{1}{6}(\tfrac{1}{2}+u)^2(\tfrac{1}{2}+\tfrac{1}{2}u) \end{aligned}$$

We verify this result by evaluating $M_6(\cos\frac{\pi i}{3})\, i = 0, \ldots, 5$ to get the known mask values $\{\frac{3}{8}, \frac{1}{8}, 0, 0, 0, \frac{1}{8}\}$.

We would like a general mask equation for any n. We propose

$$\boxed{M_n(u) = z_0\,(u+z_1)^2\left(\tfrac{1}{2}+\tfrac{1}{2}u\right)^k} \tag{17}$$

where $k = \lfloor \frac{n-4}{2} \rfloor$. This form exposes some useful properties. $M_n(u)$ has a double root at $u = -z_1$ and a root of multiplicity k at $u = -1$. This root structure is the key to demonstrating the claimed convexity result, since any positive value for z_0 in Equation (17) will lead to $M_n(u)$ being non-negative in the interval $-1 \leq u \leq 1$ (i.e., the range of $\cos\theta$).

From Equations (8), (10) and (16) we see that two of the Chebyshev expansion coefficients of $M_n(u)$ must be

$$x_1 = \tfrac{2\lambda_1}{n} \quad \text{and} \quad x_2 = \tfrac{2\lambda_1^2}{n}. \tag{18}$$

By expanding (17) into the Chebyshev basis, we can relate these two equations to the unknowns z_0 and z_1. The resulting system is quadratic in z_1 and solvable using the quadratic formula. Satisfaction of the other eigenvalue constraints turns out to be a property of the particular choice of mask equation (17).

We proceed by showing how to solve for z_0 and z_1 for a particular n. We generalize our notation a bit so that $M_n(u) = M_n^k(u)$ and

$$\begin{aligned} M_n^{j+1}(u) &= z_0\,(u+z_1)^2\left(\tfrac{1}{2}+\tfrac{1}{2}u\right)^{j+1} \\ &= (\tfrac{1}{2}+\tfrac{1}{2}u)M_n^j(u) \end{aligned} \tag{19}$$

and

$$M_n^j(u) = \sum_{i=0}^{j+2} x_i^j T_i(u). \tag{20}$$

Note that superscripts above are used as iteration indices. Substituting (20) into (19) we get

$$M_n^{j+1}(u) = \sum_{i=0}^{j+2} \tfrac{1}{2} x_i^j (T_i(u) + uT_i(u)). \tag{21}$$

By substituting the Chebyshev recursion formula

$$u\,T_i(u) = \tfrac{1}{2}(T_{i-1}(u) + T_{i+1}(u))$$

into (21) we obtain

$$\begin{aligned} M_n^{j+1}(u) = {} & \tfrac{1}{4}x_0^j\,T_{-1}(u) + \tfrac{1}{2}x_0^j\,T_0(u) + \tfrac{1}{4}x_1^j\,T_0(u) \\ & + \sum_{i=1}^{j+1}(\tfrac{1}{4}\,x_{i-1}^j + \tfrac{1}{2}\,x_i^j + \tfrac{1}{4}\,x_{i+1}^j)T_i(u) \\ & + \tfrac{1}{4}x_{j+1}^j\,T_{j+2}(u) + \tfrac{1}{2}x_{j+2}^j\,T_{j+2}(u) + \tfrac{1}{4}x_{j+2}^j\,T_{j+3}(u). \end{aligned} \tag{22}$$

Since $\cos(-\theta) = \cos\theta$, it follows that $T_{-1}(u) = T_1(u)$. We define $x_i^j = 0$, if $i > j+2$, from (22) we derive the following recursion formula for the x_i^j's

$$\left.\begin{array}{l} x_0^j = \frac{1}{2}\, x_0^{j-1} + \frac{1}{4}\, x_1^{j-1} \\ x_1^j = \frac{1}{2}\, x_0^{j-1} + \frac{1}{2}\, x_1^{j-1} + \frac{1}{4}\, x_2^{j-1} \\ x_i^j = \frac{1}{4}\, x_{i-1}^{j-1} + \frac{1}{2}\, x_i^{j-1} + \frac{1}{4}\, x_{i+1}^{j-1}, \quad 2 \le i \le j+2 \end{array}\right\} \qquad (23)$$

From the Chebyshev expansion of $M_n^0(u)$ we have

$$x_0^0 = z_0\,(\tfrac{1}{2} + z_1^2), \quad x_1^0 = 2\, z_0\, z_1, \quad x_2^0 = \tfrac{1}{2}\, z_0, \qquad (24)$$

as initial values for the recursion formulas. The recursion is then evaluated for $j = 1, \ldots, k$ (recall that $k = \lfloor \frac{n-4}{2} \rfloor$) leading to a set of quadratic polynomials of the form

$$x_i^k = z_0(a_i\, z_1^2 + b_i\, z_1 + c_i), \quad i = 0, \ldots, k+2, \qquad (25)$$

where the coefficients a_i, b_i, and c_i are rational and determined solely by the depth of recursion. As a practical matter, the recursion can be run for each of the coefficients a, b, and c separately using the initial values (derived from Equation (24)) $\{1, 0, 0\}$ for the a_i's, $\{0, 2, 0\}$ for the b_i's, and $\{\frac{1}{2}, 0, \frac{1}{2}\}$ for the c_i's.

We combine (25) with Equations (18) to get

$$\tfrac{2\lambda_1}{n} = z_0(a_1\, z_1^2 + b_1\, z_1 + c_1) \quad \text{and} \quad \tfrac{2\lambda_1^2}{n} = z_0(a_2\, z_1^2 + b_2\, z_1 + c_2).$$

These equations are solved by

$$z_1 = \frac{b_2 - b_1\,\lambda_1 \pm \sqrt{(b_1\,\lambda_1 - b_2)^2 - 4\,(a_1\,\lambda_1 - a_2)\,(c_1\,\lambda_1 - c_2)}}{2\,(a_1\,\lambda_1 - a_2)}, \qquad (26)$$

$$z_0 = \frac{2\,\lambda_1}{n\,(a_1\, {z_1}^2 + b_1\, z_1 + c_1)}. \qquad (27)$$

We choose the value of z_1 with the positive square root since this gives a value for λ_0 satisfying Equation (14), and has the property that $\lambda_0 \approx \frac{5}{8}$ (see Remark 1).

Example 3 *Consider the case $n = 8$. In this case $k = 2$, so the recursion is iterated twice to obtain*

$$a_1 = \tfrac{1}{2},\ b_1 = \tfrac{7}{8},\ c_1 = \tfrac{3}{8}, \quad \textit{and} \quad a_2 = \tfrac{1}{8},\ b_2 = \tfrac{1}{2},\ c_2 = \tfrac{1}{4}.$$

These values, together with $\lambda_1 = \frac{3+\sqrt{2}}{8}$, are substituted into Equations (26) and (27) to get

$$z_0 = \frac{13+9\,\sqrt{2}}{139-23\,\sqrt{2}+9\,\sqrt{235-90\,\sqrt{2}}} \quad \textit{and} \quad z_1 = \frac{11-7\,\sqrt{2}+\sqrt{235-90\,\sqrt{2}}}{8\,\left(1+\sqrt{2}\right)}.$$

Next, we use these values and evaluate Equation (17) at $u = \cos\frac{\pi i}{4}, i = 0, \ldots, 7$ *to get the new edge mask values*

$$\gamma = \{0.32719298,\ 0.15883976,\ 0.01136773,\ 0.00003509,\ 0.0,\ \ldots\}.$$

Table 1 contains values of z_0, z_1 and λ_0 for various valance. It turns out that z_0 is only real for $n < 88$ for the mask equation defined by (17). While being well within the range of valance that might be encountered in practice, this result is unfortunate from a theoretical point of view since we would like a mask equation for arbitrary n.

n	z_0	z_1	λ_0
6	0.1666666666666666	0.5	0.625
7	0.1822391069536521	0.3836916459399613	0.657553
8	0.1287150068251107	0.5943636980509149	0.667678
9	0.1340979275105454	0.5009463415912177	0.679870
10	0.07627676810088113	0.9259323317601788	0.690364
11	0.07837296538039939	0.8307556132822432	0.696764
12	0.03144205431440641	1.798314449246357	0.702710
13	0.03281826302734534	1.648878581962387	0.706602
14	0.004516129601508095	5.91675920648683	0.709627
15	0.005248796858155552	5.225839788173805	0.712187
16	0.002724638755476682	-9.373519013846603	0.713590
17	0.002034597685673819	-10.43053636881855	0.715366
18	0.03235135867010321	-3.292169329750421	0.715876
19	0.02918272720031251	-3.354625797957334	0.717157
20	0.09932352563307439	-2.237006678599589	0.717171
21	0.09254320140201956	-2.252948832692495	0.718125
22	0.2094086213629963	-1.807118001336983	0.717863
23	0.1978762764547314	-1.813255061794855	0.718593
24	0.3682966607544362	-1.577423976164719	0.718179
25	0.3508882240115513	-1.580325255289438	0.718749
26	0.5816437257773352	-1.436384681027866	0.718256
27	0.5572574080584542	-1.43794142531559	0.718710
28	0.8551009549474157	-1.342044855952346	0.718182
29	0.8226566325664633	-1.342956390553909	0.718549
30	1.194337773433554	-1.275151869955513	0.7180109
⋮	⋮	⋮	⋮
86	74.84657993958395	-0.9890416876219642	0.712097
87	73.95905050392309	-0.9890605963795728	0.712110

Table 1: Values of z_0, z_1 and λ_0 for various valance.

Demonstrating that masks generated by Equation (17) satisfy the constraints for bounded curvature are easy to check programatically. In particular, $\lambda_2 \geq |\lambda_i|, i = 3, \ldots, n-3$ has been verified for $n < 88$, this can also be shown inductively using the recursion formulas. Convexity follows from the fact that z_0 is always positive. All of the other constraints are satisfied by construction.

5.1 Cases $n < 6$

Mask equations for the cases where $n = 3, 4, \text{and}, 5$ require special handling since in each case, the previous derivation degenerates in some way.

5.1.1 The case $n = 5$

When $n = 5$, $k = 0$ and the recursion formulas are not iterated. Therefore

$$a_1 = 0,\ b_1 = 2,\ c_1 = 0, \quad \text{and} \quad a_2 = 0,\ b_2 = 0,\ c_2 = \tfrac{1}{2}.$$

Since $a_1 = a_2 = 0$, x_1^0 and x_2^0 are linear, not quadratic, so Equation (26) is not valid for this case. Instead, we must solve the system

$$\tfrac{2\lambda_1}{n} = 2\,z_0\,z_1 \quad \text{and} \quad \tfrac{2\lambda_1^2}{n} = \tfrac{1}{2}\,z_0.$$

This has a single solution

$$z_0 = \tfrac{4\,\lambda_1^2}{n} \quad \text{and} \quad z_1 = \tfrac{1}{4\,\lambda_1}.$$

From this, we get the mask equation

$$M_5(u) = \tfrac{3+\sqrt{5}}{32}\,(\tfrac{5-\sqrt{5}}{5} + u)^2.$$

5.1.2 The case $n = 4$

The case $n = 4$ is similar to the $n = 5$ case with one caveat: the multiplicity of λ_2 is reduced from 3 to 2 since $\frac{n}{2} = 2$, when $n = 4$ (see Equation (16)). We solve a slightly different system

$$\tfrac{2\lambda_1}{n} = 2\,z_0\,z_1 \quad \text{and} \quad \tfrac{\lambda_1^2}{n} = \tfrac{1}{2}\,z_0$$

to get the mask equation

$$M_4(u) = \tfrac{1}{2}\,(\tfrac{1}{2} + \tfrac{3}{8}u)^2.$$

5.1.3 The case $n = 3$

In this case, the multiplicity of λ_2 is reduced from 3 to 1, eliminating λ_2 as an eigenvalue of $\mathbf{S}$. Therefore, the only enforceable bounded curvature constraint is $\alpha + \lambda_0 - 1 = \lambda_1^2$ (which gets enforced by the choice of α). A mask equation that satisfies the remaining parameter constraints is

$$M_3(u) = \tfrac{1}{6}\,(\tfrac{5}{4} + u).$$

This choice coincides with Loop's algorithm for $n = 3$.

6 Results

Figure 5 shows a comparison of the old and new algorithms for a surface containing a valance 13 vertex. In Guassian curvature plot Figure 5a), we see an artifact of divergent curvature associated with Loop's algorithm at an extraordinary vertex. In Figure 5b), this artifact is removed by the new algorithm and replaced by a kind of *spider web* artifact. This artifact indicates a slight flattening of the surface near original control mesh edges but is only noticeable in curvature plots.

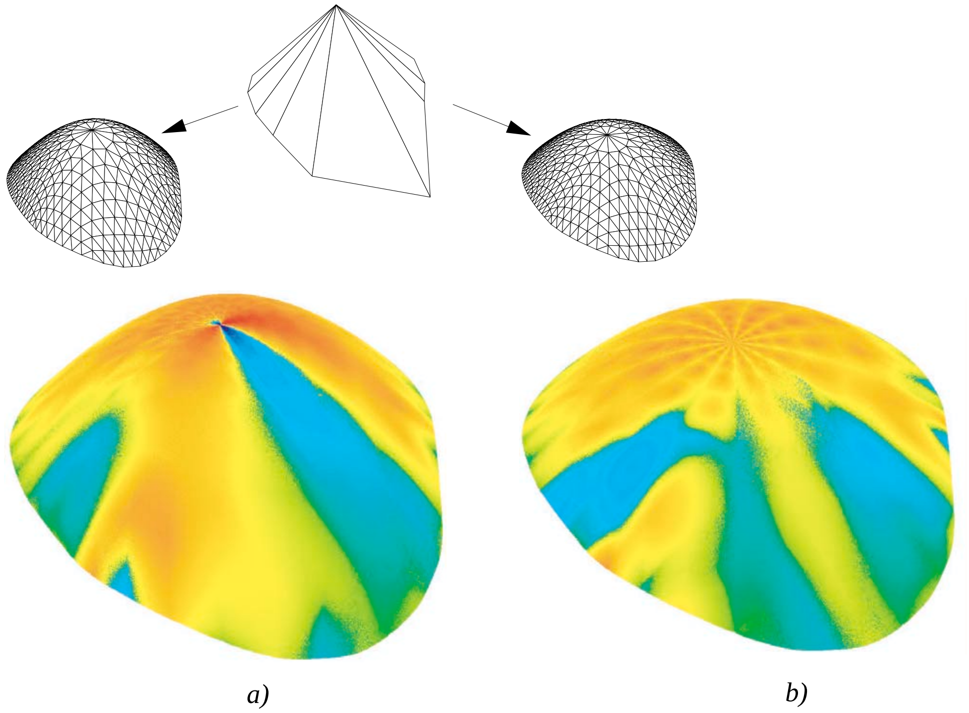

Figure 5: An initial control mesh top, with Loop's algorithm used to generate the surfaces on the right a), and the new modified algorithm used to generate the surfaces on the left b).

7 Conclusions

We have presented a modified version of Loop's triangular subdivision surface algorithm with bounded curvature and non-negative masks. This new algorithm uses an edge mask with larger support than the original whose weights come from a polynomial mask equation. A close relationship between the Chebyshev expansion coefficients of this mask equation and the eigenvalues of the corresponding 1-ring subdivision matrix was established. Using this relationship, constraints on the eigenvalues sufficient for tangent plane continuity and bounded curvature are satisfied by the mask equation. The mask equation itself guarantees convexity of edge masks by its simple root structure that does not allow zero crossings in the interval $-1 \leq u \leq 1$.

The mask equation was chosen to be a product of a degree $\lfloor \frac{n-4}{2} \rfloor$ factor and a quadratic factor. This choice is clearly hueristic, but has the advantage that its two degrees of freedom lead to an analytic solution. The proposed mask equation has the disadvantage that these solutions are only real for $n < 88$. Different mask equations may be possible that are applicable for arbitrary valance. Future work should address this large valance problem. Also to be considered is the quadrilateral case, represented by the Catmull&Clark algorithm.

Acknowledgements

I wish to thank Kirk Olynyk for many helpful discussions and for his insights in deriving the recursion formulas for the Chebyshev expansion coefficients.

References

[1] A.Ball and D. Storry. Conditions for tangent plane continuity over recursively generated b-spline surfaces. *ACM Transactions on Graphics*, 7(2):83–102, 1988.

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

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

[4] G.Umlauf. Analyzing the characteristic map of triangular subdivision schemes. *Constructive Approximation*, 16(1):145–155, 2000.

[5] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH 94 Conference Proceedings*, Annual Conference Series, pages 295–302. ACM SIGGRAPH, Addison Wesley, 1994.

[6] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, 1987.

[7] H. Prautzsch and U. Reif. Necessary conditions for subdivision surfaces. 1996. `http://i33www.ira.uka.de.`

[8] H. Prautzsch and G. Umlauf. A G^2-subdivision algorithm. In G. Farin, H. Bieri, G. Brunnet, and T. DeRose, editors, *Geometric Modelling*, volume 13 of Computing suppl. Springer-Verlag, 1998.

[9] H. Prautzsch and G. Umlauf. Improved triangular subdivision schemes. In F. Wolter and N. Patrikalakis, editors, *Computer Graphics International 1998*, pages 626–632. IEEE Computer Society, June 1998.

[10] U. Reif. A unfied approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design*, 12:153–174, 1995.

[11] U. Reif and P. Schröder. Curvature smoothness of subdivision surfaces. Technical Report TR-00-03, Caltech, 1999.

[12] M. Sabin. *The use of piecewise forms for the numerical representation of shape*. PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary, 1976.

[13] J. Schweitzer. Analysis and application of subdivision surfaces. Technical Report UW-CSE-96-08-02, University of Washington, 1996.